TM,A,B-11

MULTIDRIVE DATA RELIABILITY

MD-11-DZTMH-F

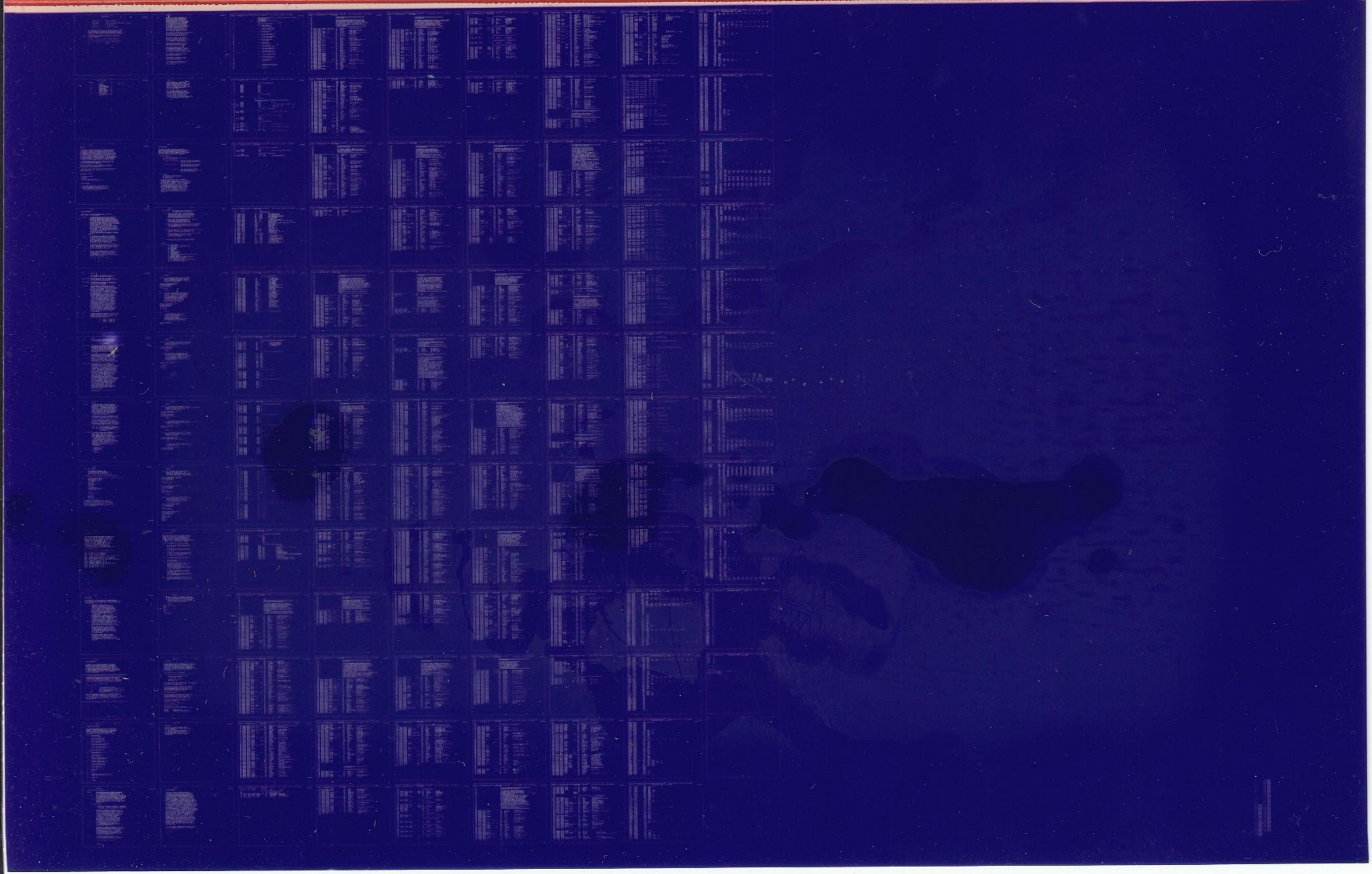
EP-DZTMH-F-DL

FICHE 1 OF 1

JAN 1978

digital

MADE IN USA



IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZTMH-F-D

PRODUCT NAME: TM,A,B-11 MULTIDRIVE DATA RELIABILITY EXERCISER

PRODUCT DATE: 15-NOVEMBER-1977

MAINTAINER: DIAGNOSTIC ENGINEERING

TABLE OF CONTENTS

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 PROCESSER
B. 8K OF CORE
C. TELETYPE
D. TM,A,B-11 TAPE CONTROL UNIT
E. 1 TO 8 TS03 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 CHARACTES 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 REPONSES 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 RENTERED. 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 THERFORE 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 NRZ1
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 CONJUCTION 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 THOUGH 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) CHARATERS 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 RELIABILTY 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:

SW0+SW3: THESE SWITCHES ARE USED TO CONTROL THE SEQUENCE OF MAG TAPE OPERATIONS PREFORMED 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: SW0+SW3

A. SW0=0, SW3=1 WRITE ONLY X RECORDS OF Y CHARACTERS
B. SW0=1, SW3=0 READ ONLY X RECORDS OF Y CHARACTERS
C. SW0=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 INTERECORD GAP IS DONE, A SUSPECTED BAD TAPE SPOT LOGGED AT THIS BLOCK 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 SYSTEMBEFORE 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 ON 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 APPROPIATE 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)
PATTRN = 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  *PATTRN 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  *PATTRN 5
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  *PATTRN  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  *PATTRN 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  *PATTRN 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  *PATTRN R
BN 12*RN 3-4*RS 100*WE
NO INTERRUPT RETURNED
```

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  *PATTRN R
*BN 10 *RN 2-100 *RS 1117 *RE  F***
COMD 1110100110000010
STAT 0011000001000001
WC 0
LPC 337-147
***ORIGINAL ERROR***


UNIT NO. 1 *DEN 3 *PAR 0  *PATTRN R
*BN 10 *RN 2-100 *RS 1117 *RE  F***
COMD 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 *PATTRN R
*BN 2 *RN 370 -461 *RS 2407 *WE
COMD 1110000010000100
STAT 0011000001000001
WC 0
CA 25613 -25613
***ORIGINAL ERROR***

UNIT NO. 0 *DEN 3 *PAR 0 *PATTRN R
*BN 2 *RN 370 -461 *RS 2407 *WE
COMD 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 ITEM8., 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
RDERR:  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 THOUGH 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 THE 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 PREFORMED 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

```
                        .TITLE TM,A,B-11 TS03 OR TU10,N,W  MULTIDRIVE DATA RELIABILITY EXERCISER
                        ;MAINDEC-11-DZTMH-F-D
                        ;15-NOVEMBER-1977
                        ;R. B. BARNES/RON PLATUKIS/R. SOLER
                        .ENABLE ABS,AMA

                        ;CONSOLE SWITCHES**********

                        ;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
                        ;        0=CHECK DATA

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

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

                        ;SW10:  1=DO NOT PRINT ERRORS
                        ;        0=PRINT ERRORS

                        ;SW9:   1=REWIND TAPE
                        ;        0=DO NOT REWIND

                        ;SW8:   1=USE RANDOM DATA
                        ;        0=USE FIXED DATA PATTERN

                        ;SW7:   1=USE RANDOM CHARACTER COUNT
                        ;        0=USE FIXED CHAR COUNT

                        ;SW6:   1=USE RANDOM RECORD COUNT
                        ;        0=USE FIXED RECORD COUNT

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

                        ;SW4:   1=PRINT DROPS AND PICKS
                        ;        0=DO NOT PRINT DROPS AND PICKS

                        ;SW3:   1=DO NOT READ FORWARD
                        ;        0=READ FORWARD

                        ;SW2:    NOT USED

                        ;SW1:   1=INHIBIT WRITE AND READ RETRY
                        ;        0=ENABLE WRITE AND READ RETRY

                        ;SW0:   1=DO NOT WRITE
                        ;        0=WRITE
```

```
                                        ;REGISTER EQUIVS**********

                000000                  R0=%0
                000001                  R1=%1
                000002                  R2=%2
                000003                  R3=%3
                000004                  R4=%4
                000005                  R5=%5
                000006                  SP=%6
                000007                  PC=%7
                000240                  NOP=240

                                        ;TRAP CATCHERS**********

                000000                  .=0
                000042                  .=42
                                .SBTTL  ACT11 HOOKS

                                ;;**************************************************************
                                ;HOOKS REQUIRED BY ACT11
                000042                  $SVPC=.                 ;SAVE PC
                000046                  .=46
000046  004650                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                000052                  .=52
000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
                000042                  .=$SVPC                 ;; RESTORE PC

                                        ;TTY INTERRUPT VECTOR**********

                000060                  .=60
000060  017404                          TTINT                   ;TTY INTERRUPT HANDLER ADDRESS
000062  000000                          0

                                        ;SOFTWARE SWITCH REGISTER LOCATIONS*********

                000174                  .=174
000174  000000                          DISPREG: 0
000176  000000                          SWREG:   0

                                        ;START ADDRESS**********

                000200                  .=200
000200  000137  002772                  JMP     START           ;ENTER PARAMETERS VIA TTY

                000204                  .=204
000204  000137  003124                  JMP     STARTA          ;USE FIXED PARAMETERS; HOLD DATA

                000210                  .=210
000210  005037  013304          CHAIN:  CLR     RDFL
000214  000137  003142                  JMP     STARTE          ;USE FIXED PARAMETERS; NEW DATA
```

```
                                        ;MAG TAPE INTERRUPT VECTOR**********

            000224                      .=224
000224  017466                          MTINT                   ;MAG TAPE INTERRUPT HANDLER ADDRESS
000226  000340                          340

                                        ;AUTO SEQUENCE START**********
            000240                      .=240
000240  005237  021632                  INC     ASEQF           ;SET AUTO SEQUENCE FLAG
000244  000137  003106                  JMP     STAUT           ;GO TO START OF AUTO SEQ
```

```
                        000600                  .=600
                                                ;CONSTANTS**********

000600  172520          MTS:    172520          ;TAPE STATUS REGISTER
000602  172522          MTC:    172522          ;TAPE COMMAND REGISTER
000604  172524          MWC:    172524          ;TAPE CHARACTER COUNT REGISTER
000606  172526          MDA:    172526          ;TAPE DATA ADDRESS REGISTER
000610  172530          MTD:    172530          ;TAPE DATA BUFFER
000612  172532          MTRD:   172532          ;TAPE READ LINES
000614  000224          VECT:   224             ;INTERRUPT VECTOR ADDRESS
000616  000000          UDES:   0               ;UNIT DESCRIPTION (PARITY,DENSITY,UNIT,TRACK)
000620  000100          RCNT:   100             ;RECORD COUNTER
000622  177600          CARCNT: 177600          ;NUMBER OF CHAR (2 - 4000) OCTAL IN TWOS COMPLEMENT
000624  000001          PATRN:  1               ;DATA PATTERN SELECTOR (0 - 20) OCTAL
000626  000002          RDCMD:  2               ;READ COMMAND
000630  000001          SPFLG:  1               ;SINGLE PASS FLAG
000632  000001          RSTAL:  1               ;READ STALL
000634  000001          WSTAL:  1               ;WRITE STALL
000636  000001          TSTAL:  1               ;TURN AROUND STAL
000640  001000          YSTAL:  1000            ;YOZZLE STAL
000642  000100          RCSAV:  100             ;RECORD COUNT SAVE
000644  177600          CCSAV:  -200            ;CHARACTER COUNT SAVE
000646  000000          TMEX:   0               ;TAPE MARK FLAG:   1=TM  0=NO TM
000650  177776          PSW:    177776          ;PROCESSOR STATUS
000652  177570          SWR:    177570          ;CONSOLE SWITCHES
000654  177570          DISPLAY: 177570
000656  177560          TKS:    177560          ;TTY READ STATUS REGISTER
000660  177562          TKB:    177562          ;TTY READ BUFFER
000662  177564          TPS:    177564          ;TTY PUNCH STATUS REGISTER
000664  177566          TPB:    177566          ;TTY PUNCH OUTPUT REGISTER
000666  177550          PRS:    177550          ;H/S READER STATUS REGISTER
000670  177552          PRB:    177552          ;H/S READER BUFFER
000672  153624          RANBAS: 153624          ;RANDOM NUMBER GENERATOR BASE
000674  172520          REGST:  172520          ;STARTING REGISTER ADDRESS
000676  032561          RANSAV: 032561          ;RANDOM NUMBER BUFFER
```

```
                                ;FLAGS AND COUNTERS**********

000700  000000          TINF:   0               ;TTY ENTERY FLAG
000702  000000          TOB:    0               ;TTY OUTPUT BUFFER
000704  000000          TIB:    0               ;TTY INPUT BUFFER
000706  000000          TEMP1:  0               ;TEMP STORAGE
000710  000000          TEMP2:  0               ;TEMP STORAGE
000712  000000          TEMP3:  0               ;TEMP STORAGE
000714  000000          TEMP4:  0               ;TEMP STORAGE
000716  000000          EMADDR: 0               ;ERROR MSG ADDRESS STORAGE
000720  000000          BLCNTR: 0               ;BLOCK COUNTER
000722  000000          BBC:    0               ;BAD RECORD COUNTER
000724  000000          RTRN:   0               ;INTERRUUPT RETURN STORAGE
000726  000000          HDRFL:  0               ;HEADER FLAG
000730  000000          STAL:   0               ;DELAY STORAGE
000732  000000          PFLG:   0               ;PRINT FLAG
000734  000000          UNP:    0               ;UNIT TABLE POINTER
000736  000000          BCNT:   0               ;BIT COUNTER
000740  000000          ERSAV:  0               ;STATUS STORAGE
000742  000000          SERFL:  0               ;STATUS ERROR FLAG
000744  000000          DERFL:  0               ;DATA ERROR FLAG
000746  000000          BTFLG:  0               ;BAD TAPE FLAG
000750  000000          RPCNT:  0               ;REPEAT COUNTER
000752  000000          RTCNT:  0               ;RETRY COUNTER
000754  000000          RTYFL:  0               ;RETRY FLAG
000756  000000          TMFLG:  0               ;TM FLAG
000760  000000          EOTREC: 0               ;END OF TAPE RECORD
000762  000000          BTPT:   0               ;BAD TAPE POINTER
000764  000000          ERTFL:  0               ;ERASE TAPE FLAG
000766  000000          BDPP:   0               ;DROP POINTER
000770  000000          BPKP:   0               ;PICK POINTER
000772  000000          BTSTF:  0               ;BAD TAPE STATISTICS FLAG
000774  000000          RRTYFL: 0               ;READ RETRY FLAG
000776  000000          SEQCT:  0               ;AUTO SEQ PASS COUNT
001000  000000          COUNT:  0
001002  000000          TEMPST: 0
001004  000000          RDSW:   0
001006  000000          DUCTR:  0               ;DROPPED UNIT COUNTER
001010  000000          STCDFL: 0               ;7 TRK CORE DUMP FLAG
```

```
                                        ;UNIT ORDER AND DESCRIPTION TABLE **********

001012  060000            UN1:    60000           ;THIS TABLE IS LOADED
001014  177777            UN2:    -1              ;WITH UNIT NUMBERS AND
001016  177777            UN3:    -1              ;THEIR DESCRIPTIONS IN
001020  177777            UN4:    -1              ;THE ORDER THAT THEY
001022  177777            UN5:    -1              ;WILL BE TESTED
001024  177777            UN6:    -1
001026  177777            UN7:    -1
001030  177777            UN8:    -1
001032  177777            UNX:    -1

                                        ;UNIT DROPS AND PICKS COUNTERS**********

001034  001254            PIK1:   BP00
001036  001274            PIK2:   BP10
001040  001314            PIK3:   BP20
001042  001334            PIK4:   BP30
001044  001354            PIK5:   BP40
001046  001374            PIK6:   BP50
001050  001414            PIK7:   BP60
001052  001434            PIK8:   BP70
001054  001454            DRP1:   BD00
001056  001474            DRP2:   BD10
001060  001514            DRP3:   BD20
001062  001534            DRP4:   BD30
001064  001554            DRP5:   BD40
001066  001574            DRP6:   BD50
001070  001614            DRP7:   BD60
001072  001634            DRP8:   BD70

                                        ;UNIT WRITE ERRORS**********

001074  000000            WTER1:  0
001076  000000            WTER2:  0
001100  000000            WTER3:  0
001102  000000            WTER4:  0
001104  000000            WTER5:  0
001106  000000            WTER6:  0
001110  000000            WTER7:  0
001112  000000            WTER8:  0
                                        ;UNIT READ ERRORS**********

001114  000000            RDER1:  0
001116  000000            RDER2:  0
001120  000000            RDER3:  0
001122  000000            RDER4:  0
001124  000000            RDER5:  0
001126  000000            RDER6:  0
001130  000000            RDER7:  0
001132  000000            RDER8:  0
```

```
                                        ;UNIT DATA ERRORS********************

001134  000000          DATER1: 0
001136  000000          DATER2. 0
001140  000000          DATER3: 0
001142  000000          DATER4: 0
001144  000000          DATER5: 0
001146  000000          DATER6: 0
001150  000000          DATER7: 0
001152  000000          DATER8: 0

                        ;UNIT RETRY COUNTERS**********

001154  000000          RTY1:   0
001156  000000          RTY2:   0
001160  000000          RTY3:   0
001162  000000          RTY4:   0
001164  000000          RTY5:   0
001166  000000          RTY6:   0
001170  000000          RTY7:   0
001172  000000          RTY8:   0

                        ;UNIT SOFT READ ERRORS****************************

001174  000000          GDRTY1: 0
001176  000000          GDRTY2: 0
001200  000000          GDRTY3: 0
001202  000000          GDRTY4: 0
001204  000000          GDRTY5: 0
001206  000000          GDRTY6: 0
001210  000000          GDRTY7: 0
001212  000000          GDRTY8: 0

                        ;UNIT HARD READ ERRORS****************************

001214  000000          BDRTY1: 0
001216  000000          BDRTY2: 0
001220  000000          BDRTY3: 0
001222  000000          BDRTY4: 0
001224  000000          BDRTY5: 0
001226  000000          BDRTY6: 0
001230  000000          BDRTY7: 0
001232  000000          BDRTY8: 0

                        ;UNIT EOT COUNTERS************************************

001234  000000          EOTCT1: 0
001236  000000          EOTCT2: 0
001240  000000          EOTCT3: 0
001242  000000          EOTCT4: 0
001244  000000          EOTCT5: 0
001246  000000          EOTCT6: 0
001250  000000          EOTCT7: 0
001252  000000          EOTCT8: 0
```

```
                                                ;DROPS + PICKS PER CHANNEL PER UNIT. ************************
001254  000000                          BP00:   0
        001274                                  .=.+16
001274  000000                          BP10:   0
        001314                                  .=.+16
001314  000000                          BP20:   0
        001334                                  .=.+16
001334  000000                          BP30:   0
        001354                                  .=.+16
001354  000000                          BP40:   0
        001374                                  .=.+16
001374  000000                          BP50:   0
        001414                                  .=.+16
001414  000000                          BP60:   0
        001434                                  .=.+16
001434  000000                          BP70:   0
        001454                                  .=.+16
001454  000000                          BD00:   0
        001474                                  .=.+16
001474  000000                          BD10:   0
        001514                                  .=.+16
001514  000000                          BD20:   0
        001534                                  .=.+16
001534  000000                          BD30:   0
        001554                                  .=.+16
001554  000000                          BD40:   0
        001574                                  .=.+16
001574  000000                          BD50:   0
        001614                                  .=.+16
001614  000000                          BD60:   0
        001634                                  .=.+16
001634  000000                          BD70:   0
        001654                                  .=.+16

                                                ;UNIT BAD TAPE COUNTER: 16 PER DRIVE

001654  000000                          BT00:   0
        001760                                  .=.+102
001760  000000                          BT01:   0
        002064                                  .=.+102
002064  000000                          BT02:   0
        002170                                  .=.+102
002170  000000                          BT03:   0
        002274                                  .=.+102
002274  000000                          BT04:   0
        002400                                  .=.+102
002400  000000                          BT05:   0
        002504                                  .=.+102
002504  000000                          BT06:   0
        002610                                  .=.+102
002610  000000                          BT07:   0
        002714                                  .=.+102
```

```
                                        ;UNIT BAD TAPE POINTERS**********

002714  001654                  BTADDR. BT00
002716  001760                          BT01
002720  002064                          BT02
002722  002170                          BT03
002724  002274                          BT04
002726  002400                          BT05
002730  002504                          BT06
002732  002610                          BT07

                                        ;DATA PATTERN GENERATORS**********

002734  002734                  DATBL:  .                       ;ENTRY TABLE
002736  012552                  DATA0:  DAT0                    ;EXTERNAL INPUT FROM H/S READER
002740  012754                  DATA1:  DAT1                    ;ALL ONES
002742  012776                  DATA2:  DAT2                    ;ALL ZEROS
002744  013004                  DATA3:  DAT3                    ;WALKING ONE
002746  013032                  DATA4:  DAT4                    ;WALKING ZERO
002750  013044                  DATA5:  DAT5                    ;ALTERNATING ONE/ZERO
002752  013054                  DATA6:  DAT6                    ;ALTERNATING ZERO/ONE
002754  013064                  DATA7:  DAT7                    ;ALTERNATING ONE/ZERO IN ALTERNATING CHARACTERS
002756  013074                  DATA10: DAT10                   ;ALTERNATING ZERO/ONE IN ALTERNATING CHARACTERS
002760  013104                  DATA11: DAT11                   ;ALL BITS 0-377
002762  013126                  DATA12: DAT12                   ;ALL BITS 377-0
002764  013152                  DATA13: DAT13                   ;ALTERNATING CHARACTERS  0 AND 377
002766  013162                  DATA14: DAT14                   ;ALTERNATING CHARACTERS 377 AND 0
002770  013172                  DATA15: DAT15                   ;WALKING ZERO REPEATED FOUR TIMES
```

```
                                        .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).
                                        ;*************************************************************

002772  005037  021632          START:  CLR     ASEQF           ;CLEAR AUTO SEQ FLAG
002776  012737  177570  000652        MOV     #177570,SWR     ;PRESET FOR CONSOLE SWITCHES
003004  005737  000042                TST     @#42            ;SEE IF CHAIN MODE
003010  001436                        BEQ     STAUT           ;IF NOT: BR
003012  012706  000500                MOV     #500,SP         ;SET UP STACK POINTER
003016  012704  023152                MOV     #MSG31,R4
003022  004737  020536                JSR     PC,TTOUT        ;PRINT TITLE
003026  122737  000004  000041        CMPB    #4,@#41         ;SEE IF LOAD MEDIUM
003034  001006                        BNE     1$              ;IF NOT: BR
003036  012704  026124                MOV     #MSG97,R4
003042  004737  020536                JSR     PC,TTOUT        ;PRINT NO TEST LOAD MEDIUM
003046  000137  004622                JMP     REOT8           ;END TEST
003052  012737  000176  000652  1$:   MOV     #176,SWR        ;SET FOR SOFTWARE SWITHCES
003060  012700  001014                MOV     #UN2,R0         ;SET UNIT POINTER
003064  022720  177777          2$:   CMP     #-1,(R0)+       ;SEE IF END OF UNITS
003070  001404                        BEQ     3$              ;IF SO: BR
003072  062737  000401  004716        ADD     #401,REOTC      ;ELSE BUMP UNIT EOT COUNTER
003100  000771                        BR      2$
003102  000137  000210          3$:   JMP     CHAIN           ;GO DO CHAIN START
003106  012737  000001  000700  STAUT: MOV     #1,TINF         ;SET TTY ENTRY FLAG
003114  005037  013304                CLR     RDFL            ;CLEAR RANDOM DATA FLAG
003120  000137  003146                JMP     STARTB
003124  005037  000700          STARTA: CLR     TINF            ;CLEAR TTY ENTRY FLAG
003130  012706  000500                MOV     #500,SP         ;SET STACK POINTER
003134  004737  022046                JSR     PC,SUSWR        ;CHECK FOR SOFTSWR
003140  000451                        BR      STAUTO
003142  005037  000700          STARTE: CLR     TINF            ;CLEAR INPUT FLAG
003146  012700  000702          STARTB: MOV     #TOB,R0
003152  012701  000044                MOV     #44,R1
003156  005020                  STARTO: CLR     (R0)+           ;CLEAR FLAGS AND COUNTERS
003160  005301                        DEC     R1
003162  001375                        BNE     STARTO
003164  012700  000510                MOV     #510,R0         ;SET SIZE OF TABLE
003170  012701  001074                MOV     #WTER1,R1       ;SET START OF TABLE
003174  005021                  STARTC: CLR     (R1)+           ;CLEAR STATISTICS TABLES
003176  005300                        DEC     R0
```

```
 447  003200  001375                        BNE     STARTC          ;CLEAR ALL
 448  003202  012737  177777  012750        MOV     #-1,PATS        ;RESET PATTERN
 449  003210  012737  177777  012752        MOV     #-1,PARS        ;RESET PARITY
 450  003216  012737  000001  000720        MOV     #1,BLCNTR       ;PRESET BLOCK COUNTER
 451  003224  005077  175352                CLR     @MTC
 452  003230  052777  010000  175344        BIS     #10000,@MTC     ;POWER CLEAR CONTROLLER
 453  003236  012706  000500        STARTD: MOV     #500,SP
 454  003242  004737  022046                JSR     PC,SUSWR        ;CHECK FOR SORTSWR
 455  003246  012777  000340  175374  1$:   MOV     #340,@PSW
 456  003254  004737  010722                JSR     PC,TINP         ;GO GET PARAMETERS FROM TTY
 457  003260  004737  004110                JSR     PC,RANSET       ;GO RESET BASE
 458  003264  005000                STAUTO: CLR     R0              ;POINT TO FIRST ENTRY
 459  003266  022737  000176  000652        CMP     #SWREG,SWR      ;TEST FOR SOFTSWR
 460  003274  001005                        BNE     STAROA
 461  003276  005737  000042                TST     @#42            ;SEE IF CHAIN MODE
 462  003302  001002                        BNE     STAROA          ;IF SO: BR
 463  003304  004737  022172                JSR     PC,CNTLU        ;ASK FOR CONTROL SETTINGS
 464  003310  005160  001012        STAROA: COM     UN1(R0)         ;SEE IF LAST ENTRY
 465  003314  001411                        BEQ     STAROB          ;IF SO: BR
 466  003316  005160  001012                COM     UN1(R0)
 467  003322  042760  100200  001012        BIC     #100200,UN1(R0) ;CLEAR EOT/DROPPED FLAG
 468  003330  062700  000002                ADD     #2,R0           ;POINT TO NEXT UNIT ENTRY
 469  003334  000137  003310                JMP     STAROA          ;CONTINUE CLEARING
 470  003340  005160  001012        STAROB: COM     UN1(R0)
 471  003344  013703  004716                MOV     REOTC,R3
 472  003350  000303                        SWAB    R3
 473  003352  110337  004716                MOVB    R3,REOTC        ;RESTORE EOT CNTR
 474  003356  012777  000100  175272  START1: MOV    #100,@TKS       ;SET TTY INTERRUPT ENABLE
 475  003364  013700  000734                MOV     UNP,R0          ;R0 = UNIT TABLE POINTER
 476  003370  005160  001012                COM     UN1(R0)
 477  003374  001407                        BEQ     STAR1B          ;IF LAST UNIT IN STRING: BR
 478  003376  005160  001012                COM     UN1(R0)
 479  003402  016037  001012  000616  STAR1A: MOV    UN1(R0),UDES    ;LOAD NEXT UNIT DESCRIPTION
 480  003410  000137  003542                JMP     START4
 481  003414  005237  000720        STAR1B: INC     BLCNTR          ;BUMP BLOCK COUNTER
 482  003420  005737  021632                TST     ASEQF           ;SEE IF AUTO SEQ
 483  003424  001414                        BEQ     STAR1C          ;IF NOT: BR
 484  003426  023737  000720  021630        CMP     BLCNTR,ABLCNT   ;SEE IF DONE SEQ
 485  003434  001010                        BNE     STAR1C          ;IF NOT: BR
 486  003436  005160  001012                COM     UN1(R0)         ;RESET UNIT TABLE TERMINATOR
 487  003442  012737  000001  000720        MOV     #1,BLCNTR       ;RESET BLOCK COUNTER
 488  003450  005037  000734                CLR     UNP             ;RESET UNIT POINTER
 489  003454  000207                        RTS     PC              ;RETURN TO AUTO SEQ
 490  003456  005037  000734        STAR1C: CLR     UNP
 491  003462  005160  001012                COM     UN1(R0)
 492  003466  005000                        CLR     R0
 493  003470  016037  001012  000616        MOV     UN1(R0),UDES    ;LOAD FIRST UNIT DESCRIPTION
 494  003476  032777  000200  175146        BIT     #200,@SWR       ;SEE IF RANDOM RECORD SIZE
 495  003504  001402                        BEQ     START2          ;IF NOT: BR
 496  003506  004737  010644                JSR     PC,CCNTR        ;GO GENERATE RANDOM CHAR COUNT
 497  003512  032777  000400  175132  START2: BIT    #400,@SWR       ;SEE IF RANDOM DATA
 498  003520  001402                        BEQ     START3          ;IF NOT: BR
 499  003522  004737  013236                JSR     PC,DATR         ;GO GENERATE RANDOM DATA
 500  003526  032777  000100  175116  START3: BIT    #100,@SWR       ;SEE IF RANDOM RECORD COUNT
 501  003534  001402                        BEQ     START4          ;IF NOT: BR
 502  003536  004737  010676                JSR     PC,RCNTR        ;GO GENERATE RANDOM RECORD COUNT
```

```
 503  003542  032760  100000  001012  START4:  BIT   #100000,UN1(R0) ;SEE IF UNIT REACHED EOT OR DROPPED
 504  003550  001404                   BEQ   STAR40                 ;IF NOT: BR
 505  003552  062737  000002  000734           ADD   #2,UNP                 ;POINT TO NEXT UNIT
 506  003560  000676                   BR    START1
 507  003562  013777  000616  175012  STAR40:  MOV   UDES,@MTC              ;SET UNIT NUMBER
 508  003570  004737  021234                   JSR   PC,STDLY               ;GO AWAIT ASSURED STATUS
 509  003574  032777  000001  174776           BIT   #1,@MTS                ;SEE IF TUR
 510  003602  001030                   BNE   STAR46                 ;IF SO: BR
 511  003604  032777  000002  174766           BIT   #2,@MTS                ;SEE IF REWINDING
 512  003612  001414                   BEQ   STAR45                 ;IF NOT: BR
 513  003614  004737  017506                   JSR   PC,PAPRT               ;PRINT HEADER
 514  003620  012704  025503                   MOV   #MSG89,R4
 515  003624  004737  020536                   JSR   PC,TTOUT               ;PRINT REWIND MSG
 516  003630  032777  000001  174742  STAR44:  BIT   #1,@MTS
 517  003636  001774                   BEQ   STAR44                 ;AWAIT REWIND DONE
 518  003640  000137  003664                   JMP   STAR46
 519  003644  004737  017506           STAR45:  JSR   PC,PAPRT               ;PRINT HEADER
 520  003650  012704  023707                   MOV   #MSG49,R4
 521  003654  004737  020536                   JSR   PC,TTOUT               ;PRINT NOT AVAIL
 522  003660  000137  020050                   JMP   DRPDRV                 ;GO DROP DRIVE
 523  003664  005037  001010           STAR46:  CLR   STCDFL                 ;CLEAR 7 TRK CORE DUMP FLAG
 524  003670  032777  000020  174702           BIT   #20,@MTS               ;SEE IF 7 TRK
 525  003676  001411                   BEQ   1$                     ;IF NOT: BR
 526  003700  013704  000616                   MOV   UDES,R4                ;GET UNIT DESCRIPTION
 527  003704  042704  117777                   BIC   #117777,R4             ;MASK DENSITY
 528  003710  022704  060000                   CMP   #60000,R4              ;SEE IF CORE DUMP
 529  003714  001002                   BNE   1$                     ;IF NOT: BR
 530  003716  005237  001010                   INC   STCDFL                 ;ELSE SET FLAG
 531  003722  004737  012370           1$:     JSR   PC,DSUP                ;GO SET UP WRITE DATA
 532  003726  004737  004720                   JSR   PC,RWND                ;REWIND
 533  003732  004737  005254                   JSR   PC,WRITE               ;WRITE
 534  003736  013737  000636  000730           MOV   TSTAL,STAL             ;SET TURN AROUND DELAY
 535  003744  004737  010634                   JSR   PC,STALL               ;DELAY
 536  003750  004737  006626                   JSR   PC,RSEQ                ;GO TO READ SEQUENCER
 537  003754  013737  000636  000730           MOV   TSTAL,STAL             ;SET TURN AROUND DELAY
 538  003762  004737  010634                   JSR   PC,STALL               ;DELAY
 539  003766  032777  000020  174656           BIT   #20,@SWR               ;SEE IF SHOULD PRINT DROPS AND PICK
 540  003774  001410                   BEQ   START5                 ;IF NOT: BR
 541  003776  012700  000001                   MOV   #1,R0                  ;SET RECORD COUNTER TO 1
 542  004002  005237  000772                   INC   BTSTF                  ;SET FOR STAT PRINT ONLY
 543  004006  004737  015370                   JSR   PC,PRSTAT              ;PRINT STATISTICS
 544  004012  005037  000772                   CLR   BTSTF                  ;CLEAR FLAG
 545  004016  017700  174630           START5:  MOV   @SWR,R0                ;LOAD SWR
 546  004022  042700  177762                   BIC   #177762,R0             ;MASK READ/WRITE SWITCHES
 547  004026  022700  000015                   CMP   #15,R0                 ;SEE IF HAVE READ OR WRITE
 548  004032  001424                   BEQ   START8                 ;IF NOT: BR
 549  004034  032777  000001  174536  START6:  BIT   #1,@MTS                ;SEE IF HAVE UNIT READY
 550  004042  001013                   BNE   START7                 ;IF SO: BR
 551  004044  005337  000730                   DEC   STAL
 552  004050  001371                   BNE   START6                 ;DELAY FOR TUR
 553  004052  004737  017506                   JSR   PC,PAPRT               ;PRINT HEADER
 554  004056  012704  023707                   MOV   #MSG49,R4
 555  004062  004737  020536                   JSR   PC,TTOUT               ;PRINT NOT AVAIL
 556  004066  000137  020050                   JMP   DRPDRV                 ;GO DROP DRIVE
 557  004072  062737  000002  000734  START7:  ADD   #2,UNP                 ;POINT TO NEXT UNIT
 558  004100  005077  174476                   CLR   @MTC
```

```
559  004104  000137  003356          START8: JMP     START1          ;CONTINUE
560
561                                          ;RANDOM BASE RESET*********
562
563  004110  012737  153624  000672  RANSET: MOV     #153624,RANBAS  ;RESET BASE
564  004116  012737  032561  000676          MOV     #32561,RANSAV   ;RESET BUFFER
565  004124  013737  000642  000620          MOV     RCSAV,RCNT      ;RESET RECORD COUNT
566  004132  013737  000644  000622          MOV     CCSAV,CARCNT    ;RESET CHAR COUNT
567  004140  000207                          RTS     PC
```

```
                        ;*********************************************************
                        ;REWIND FROM EOT:
                        ;
                        ;WHEN ANY TRANSPORT BEING TESTED REACHES END OF TAPE
                        ;DURING A READ OR WRITE OPERATION, IT WILL BE REWOUND
                        ;AND FLAGGED AS UNAVAILABLE UNTIL ALL AVAILABLE UNITS
                        ;HAVE REACHED EOT AT WHICH TIME ALL TESTING WILL BE RESUMED
                        ;AT A BLOCK COUNT OF ONE (1).  A MESSAGE WILL BE
                        ;PRINTED ON THE SUPERVISORS CONSOLE AS EACH UNIT REACHES
                        ;EOT AND IS REWOUND.
                        ;*********************************************************

004142  013777  000616  174432  REOT:   MOV     UDES,@MTC       ;LOAD COMMAND REGISTER
004150  032777  000010  174422  REOT1:  BIT     #10,@MTS
004156  001374                          BNE     REOT1           ;AWAIT SETTLE DOWN RESET
004160  052777  000017  174414          BIS     #17,@MTC        ;START REWIND
004166  004737  017506                  JSR     PC,PAPRT        ;PRINT HEADER
004172  032737  000004  000746          BIT     #4,BTFLG        ;ERROR DURING RETRY?
004200  001405                          BEQ     1$              ;IF NOT:  BR
004202  012704  025474                  MOV     #MSG88,R4
004206  004737  020536                  JSR     PC,TTOUT        ;PRINT RETRY
004212  000404                          BR      2$
004214  032737  000002  000746  1$:     BIT     #2,BTFLG        ;BACKSPACE ERROR
004222  001405                          BEQ     REOT1C          ;IF NOT:  BR
004224  012704  024312          2$:     MOV     #MSG61,R4       ;POINT TO BACKSPACE ERROR MESSG.
004230  005037  000746                  CLR     BTFLG           ;CLEAR BAD TAPE FLAG
004234  000437                          BR      REOT1B
004236  005737  000746          REOT1C: TST     BTFLG           ;TEST BAD TAPE FLAG
004242  001405                          BEQ     REOT1D          ;IF NOT:  BR
004244  012704  024122                  MOV     #MSG59,R4       ;SET UP BAD TAPE MESSAGE
004250  005037  000746                  CLR     BTFLG           ;CLEAR BAD TAPE FLAG
004254  000427                          BR      REOT1B
004256  005737  021632          REOT1D: TST     ASEQF           ;IS IT AUTO SEQ?
004262  001406                          BEQ     REOT1A          ;IF NOT:  BR
004264  005737  000624                  TST     PATRN           ;IS IT RANDOM DATA?
004270  100403                          BMI     REOT1A          ;IF SO:  BR
004272  012704  025340                  MOV     #MSG87,R4       ;PRINT EARLY ASEQ EOT MESSG.
004276  000416                          BR      REOT1B
004300  012704  022600          REOT1A: MOV     #MSG20,R4
004304  004737  020536                  JSR     PC,TTOUT        ;PRINT EOT MESSAGE
004310  013704  000734                  MOV     UNP,R4
004314  005264  001234                  INC     EOTCT1(R4)      ;BUMP EOT COUNTER
004320  016403  001234                  MOV     EOTCT1(R4),R3
004324  004737  020724                  JSR     PC,OCTP         ;PRINT EOT COUNT
004330  012704  022614                  MOV     #MSG20A,R4
004334  004737  020536          REOT1B: JSR     PC,TTOUT        ;PRINT REWIND MSG
004340  004737  015400                  JSR     PC,PRSTA2       ;PRINT STATS WITHOUT HEADER
004344  032777  000200  174230  REOT2:  BIT     #200,@MTC
004352  001774                          BEQ     REOT2           ;AWAIT CUR
004354  105337  004716                  DECB    REOTC           ;SEE IF LAST UNIT TO REACH EOT
004360  001410                          BEQ     REOT3           ;IF SO:  BR
004362  013700  000734                  MOV     UNP,R0
004366  052760  100000  001012          BIS     #100000,UN1(R0) ;SET EOT FLAG
004374  005726                          TST     (SP)+
004376  000137  004072                  JMP     START7          ;GO TO NEXT UNIT
004402  000337  004716          REOT3:  SWAB    REOTC
```

```
624  004406  013700  004716                MOV     REOTC,R0
625  004412  000337  004716                SWAB    REOTC
626  004416  110037  004716                MOVB    R0,REOTC        ;RESTORE EOT UNIT COUNTER
627  004422  005037  000734                CLR     UNP
628  004426  013700  000734                MOV     UNP,R0          ;POINT TO FIRST UNIT
629  004432  016037  001012  000616 REOT4: MOV     UN1(R0),UDES    ;LOAD UNIT DESCRIPTION
630  004440  032737  000200  000616        BIT     #200,UDES       ;SEE IF UNIT IS DROPPED
631  004446  001034                        BNE     REOT6A          ;IF SO: BR
632  004450  013777  000616  174124        MOV     UDES,@MTC       ;LOAD COMMAND REGISTER
633  004456  032777  000002  174114 REOT5: BIT     #2,@MTS
634  004464  001374                        BNE     REOT5           ;AWAIT RWS RESET
635  004466  032777  000040  174104        BIT     #40,@MTS        ;SEE IF HAVE BOT
636  004474  001012                        BNE     REOT6           ;IF SO: BR
637  004476  012700  000001                MOV     #1,R0
638  004502  004737  017506                JSR     PC,PAPRT        ;PRINT HEADER
639  004506  012704  023661                MOV     #MSG48,R4
640  004512  004737  020536                JSR     PC,TTOUT        ;PRINT BOT ERROR
641  004516  000137  020050                JMP     DRPDRV          ;GO DROP DRIVE
642  004522  032777  000010  174050 REOT6: BIT     #10,@MTS        ;SEE IF SWDN IS RESET
643  004530  001374                        BNE     REOT6           ;IF NOT: AWAIT SWDN RESET
644  004532  042760  100200  001012        BIC     #100200,UN1(R0) ;CLEAR EOT/DROPPED FLAG
645  004540  062737  000002  000734 REOT6A: ADD     #2,UNP
646  004546  013700  000734                MOV     UNP,R0          ;POINT TO NEXT UNIT
647  004552  005160  001012                COM     UN1(R0)         ;SEE IF LAST UNIT
648  004556  001404                        BEQ     REOT7           ;IF SO: BR
649  004560  005160  001012                COM     UN1(R0)
650  004564  000137  004432                JMP     REOT4           ;DO NEXT UNIT
651  004570  005160  001012        REOT7:  COM     UN1(R0)
652  004574  012737  000001  000720        MOV     #1,BLCNTR       ;SET TO BLOCK COUNT 1
653  004602  005037  000734                CLR     UNP
654  004606  005000                        CLR     R0              ;SET TO RESTART WITH FIRST UNIT
655  004610  005726                        TST     (SP)+           ;RESET STACK
656  004612  005737  021632                TST     ASEQF           ;SEE IF AUTO SEQ
657  004616  001401                        BEQ     REOT8           ;IF NOT: BR
658  004620  000207                        RTS     PC              ;RETURN
659  004622  012704  023501        REOT8:  MOV     #MSG39,R4
660  004626  004737  020536                JSR     PC,TTOUT        ;PRINT END OF PASS
661  004632  005737  000630                TST     SPFLG           ;SEE IF SINGLE PASS
662  004636  001412                        BEQ     REOTX           ;IF NOT: BR
663  004640  013704  000042        REOT9:  MOV     @#42,R4
664  004644  001405                        BEQ     HERE            ;IF NOT CHAIN MODE: BR
665  004646  000005                        RESET
666  004650  004714                SENDAD: JSR     PC,(R4)
667  004652  000240                        NOP
668  004654  000240                        NOP
669  004656  000240                        NOP
670  004660  000240                HERE:   NOP
671  004662  000000                REOT10: HALT
672  004664  012706  000500        REOTX:  MOV     #500,SP         ;RESET STACK
673  004670  004737  004110                JSR     PC,RANSET       ;GO RESET RANDOM BASE
674  004674  012737  177777  012750        MOV     #-1,PATS        ;PRESET PATTERN
675  004702  005037  013304                CLR     RDFL            ;CLEAR RANDOM DATA FLAG
676  004706  005037  001006                CLR     DUCTR           ;CLEAR DROPPED UNITER COUNTER
677  004712  000137  003264                JMP     STAUTO          ;RESTART AT BLOCK NUMBER ONE
678  004716  000401                REOTC:  401                     ;EOT UNIT COUNTER(DEFAULT TO ONE UNIT)
```

```
                                        ;*********************************************************
                                        ;REWIND ALL AVAIL TAPES.
                                        ;
                                        ;THIS ROUTINE; ENTERED VIA CONSOLE SWITCH NINE (9),
                                        ;WILL REWIND ALL AVAILABLE TAPES TO BOT NO MATTER
                                        ;WHERE THEY ARE CURRENTLY POSITIONED AND RESUME TESTING
                                        ;AT A BLOCK COUNT OF ONE (1).
                                        ;*********************************************************

004720  032777  001000  173724  RWND:   BIT     #1000,@SWR      ;SEE IF SHOULD REWIND
004726  001001                          BNE     RWNDA           ;IF SO: BR
004730  000207                          RTS     PC              ;ELSE EXIT
004732  005037  000734          RWNDA:  CLR     UNP             ;CLEAR POINTER
004736  000337  004716                  SWAB    REOTC
004742  013700  004716                  MOV     REOTC,R0
004746  000337  004716                  SWAB    REOTC
004752  110037  004716                  MOVB    R0,REOTC        ;RESTORE EOT UNIT COUNTER
004756  013700  000734          RWND0:  MOV     UNP,R0          ;POINT TO UNIT ENTRY
004762  005160  001012                  COM     UN1(R0)         ;SEE IF LAST ENTRY
004766  001424                          BEQ     RWND2           ;IF SO:.BR
004770  005160  001012                  COM     UN1(R0)
004774  016037  001012  000616          MOV     UN1(R0),UDES    ;SET UNIT DESCRIPTION
005002  013777  000616  173572          MOV     UDES,@MTC       ;LOAD COMMAND REGISTER
005010  052777  000017  173564          BIS     #17,@MTC        ;START REWIND
005016  032777  000200  173556  RWND1:  BIT     #200,@MTC
005024  001774                          BEQ     RWND1           ;AWAIT CUR
005026  062737  000002  000734          ADD     #2,UNP          ;BUMP POINTER
005034  000137  004756                  JMP     RWND0           ;DO NEXT UNIT
005040  005160  001012          RWND2:  COM     UN1(R0)
005044  005037  000734                  CLR     UNP             ;CLEAR POINTER
005050  013700  000734          RWND3:  MOV     UNP,R0          ;POINT TO UNIT ENTRY
005054  005160  001012                  COM     UN1(R0)         ;SEE IF LAST ENTRY:
005060  001452                          BEQ     RWNDX           ;IF SO: BR
005062  005160  001012                  COM     UN1(R0)
005066  016037  001012  000616          MOV     UN1(R0),UDES    ;SET UNIT DESCRIPTION
005074  032737  000200  000616          BIT     #200,UDES       ;SEE IF UNIT IS DROPPED
005102  001403                          BEQ     1$              ;IF NOT: BR
005104  005337  004716                  DEC     REOTC           ;ELSE DECREMENT EOT UNIT CNTR
005110  000417                          BR      RWND5
005112  013777  000616  173462  1$:     MOV     UDES,@MTC       ;LOAD COMMAND REGISTER
005120  032777  000002  173452  RWND4:  BIT     #2,@MTS
005126  001374                          BNE     RWND4           ;AWAIT RWS RESET
005130  032777  000040  173442          BIT     #40,@MTS        ;SEE IF HAVE BOT
005136  001411                          BEQ     RWND6           ;IF NOT: BR
005140  032777  000010  173432  1$:     BIT     #10,@MTS        ;SEE IF SDWN SET
005146  001374                          BNE     1$              ;IF SO AWAIT RESET
005150  062737  000002  000734  RWND5:  ADD     #2,UNP          ;BUMP POINTER
005156  000137  005050                  JMP     RWND3           ;DO NEXT UNIT
005162  012700  000001          RWND6:  MOV     #1,R0
005166  004737  017506                  JSR     PC,PAPRT        ;PRINT HEADER
005172  012704  023661                  MOV     #MSG48,R4
005176  004737  020536                  JSR     PC,TTOUT        ;PRINT NO BOT
005202  000137  020050                  JMP     DRPDRV          ;GO DROP DRIVE
005206  005160  001012          RWNDX:  COM     UN1(R0)
005212  005000                          CLR     R0
005214  010037  000734          1$:     MOV     R0,UNP
```

```
   735  005220  016037  001012  000616          MOV     UN1(R0),UDES
   736  005226  032737  100200  000616          BIT     #100200,UDES    ;SEE IF UNIT DROPPED
   737  005234  001403                          BEQ     2$              ;IF NOT: BR
   738  005236  062700  000002                  ADD     #2,R0
   739  005242  000764                          BR      1$
   740  005244  012737  000001  000720  2$:     MOV     #1,BLCNTR
   741  005252  000207                          RTS     PC
```

```
                                        ;*********************************************************
                                        ;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).
                                        ;*********************************************************

005254  032777  000001  173370  WRITE:  BIT     #1,@SWR         ;SEE IF SHOULD WRITE
005262  001076                          BNE     W3A0            ;IF NOT: BR
005264  012737  022460  000716          MOV     #MSG5,EMADDR    ;SET ERROR MSG ADDRESS
005272  005077  173304                  CLR     @MTC
005276  005077  173276                  CLR     @MTS
005302  005037  000760                  CLR     EOTREC          ;CLEAR EOT FLAG
005306  013700  000620                  MOV     RCNT,R0         ;R0=RECORD COUNT
005312  013777  000622  173264  W0:     MOV     CARCNT,@MWC     ;LOAD CHAR COUNT
005320  012777  026204  173260          MOV     #WDATA,@MDA     ;SET DATA ADDR
005326  005737  000764                  TST     ERTFL           ;SEE IF SHOULD ERASE
005332  001406                          BEQ     W0A             ;IF NOT: BR
005334  112777  000014  173240          MOVB    #14,@MTC        ;SET OP-CODE: WRITE W/EXTENDED IRG
005342  005037  000764                  CLR     ERTFL           ;CLEAR ERASE FLAG
005346  000403                          BR      W0B
005350  112777  000004  173224  W0A:    MOVB    #4,@MTC         ;SET WRITE OP COMMAND
005356  012737  005370  000724  W0B:    MOV     #W1A,RTRN       ;SET RETURN ADDRESS
005364  000137  017006                  JMP     TAPG            ;GO EXECUTE COMMAND
005370  005737  017402          W1A:    TST     WEOTF           ;SEE IF EOT FOUND
005374  001413                          BEQ     W1              ;IF NOT:  BR
005376  005037  017402                  CLR     WEOTF           ;CLEAR WRITE EOT FLAG
005402  013701  000620                  MOV     RCNT,R1         ;BUILD SHORTENED RECORD COUNT
005406  160001                          SUB     R0,R1
005410  005201                          INC     R1
005412  010137  000760                  MOV     R1,EOTREC
005416  052737  100000  000760          BIS     #100000,EOTREC  ;SET EOT FLAG
005424  032777  010000  173220  W1:     BIT     #10000,@SWR     ;SEE IF SHOULD CHECK ERRORS
005432  001002                          BNE     W3              ;IF NOT: BR
005434  004737  016036                  JSR     PC,ERCHK        ;GO CHECK ERRORS
005440  013737  000634  000730  W3:     MOV     WSTAL,STAL      ;SET DELAY
005446  004737  010634                  JSR     PC,STALL        ;DELAY
005452  005737  000754                  TST     RTYFL           ;SEE IF RETRY
005456  001401                          BEQ     W3A             ;IF NOT:  BR
005460  000207                  W3A0:   RTS     PC              ;ELSE RETURN TO RETRY ROUTINE
005462  005737  000742          W3A:    TST     SERFL           ;SEE IF WRITE ERROR
```

```
   798  005466  001453                        BEQ     W3D             ;IF NOT: BR
   799  005470  013704  000734                MOV     UNP,R4          ;BUMP WRITE ERROR
   800  005474  005264  001074                INC     WTER1(R4)
   801  005500  005037  000742                CLR     SERFL           ;CLEAR STATUS ERROR FLAG
   802  005504  032777  000002  173140        BIT     #2,@SWR         ;SEE IF RETRY -- SW1
   803  005512  001041                        BNE     W3D             ;IF NOT: BR
   804  005514  042737  072521  000740        BIC     #072521,ERSAV   ;MASK UNRECOVERABLE ERROR
   805  005522  005737  000740                TST     ERSAV           ;SEE IF RETRYABLE ERROR
   806  005526  001411                        BEQ     W3B             ;IF SO: BR
   807  005530  012704  023741                MOV     #MSG52,R4
   808  005534  004737  020536                JSR     PC,TTOUT        ;PRINT NON-RETRYABLE ERROR FLAG
   809  005540  012704  022460                MOV     #MSG5,R4
   810  005544  004737  020536                JSR     PC,TTOUT        ;PRINT WRITE ERROR TAG
   811  005550  000422                        BR      W3D
   812  005552  013704  000734        W3B:    MOV     UNP,R4
   813  005556  005264  001154                INC     RTY1(R4)        ;BUMP RETRY CNTR
   814  005562  032777  002000  173062        BIT     #2000,@SWR      ;SEE IF PRINT ERRORS
   815  005570  001004                        BNE     W3C             ;IF NOT: BR
   816  005572  012704  023761                MOV     #MSG53,R4
   817  005576  004737  020536                JSR     PC,TTOUT        ;PRINT ORIGINAL ERROR TAG
   818  005602  005037  000752        W3C:    CLR     RTCNT           ;CLEAR RETRY NUMBER
   819  005606  005037  000750                CLR     RPCNT           ;CLEAR REPEAT COUNTER
   820  005612  004737  006142                JSR     PC,WRTY         ;GO RETRY WRITE ERROR
   821  005616  005037  000754        W3D:    CLR     RTYFL           ;CLEAR RETRY FLAG
   822  005622  005737  000760                TST     EOTPEC          ;WAS EOT REACHED?
   823  005626  100403                        BMI     WEX             ;IF SO:  BR
   824  005630  005300                        DEC     R0              ;SEE IF DONE ALL
   825  005632  001227                        BNE     W0              ;IF NOT: BR
   826  005634  005200                        INC     R0              ;ADJUST FOR REC NO. IN HEADER
   827  005636  005737  000646        WEX:    TST     TMEX            ;SEE IF TM
   828  005642  001402                        BEQ     WEX1            ;IF NOT:  BR
   829  005644  004737  005712                JSR     PC,WTM          ;WRITE TM
   830  005650  005037  000754        WEX1:   CLR     RTYFL           ;CLEAR RETRY FLAG
   831  005654  005037  000756                CLR     TMFLG           ;CLEAR TM FLAG
   832  005660  005737  000760                TST     EOTREC          ;TEST FOR EOT
   833  005664  100401                        BMI     W4              ;IF SO:  BR
   834  005666  000207        WEX2:   RTS     PC                      ;EXIT
   835  005670  017704  172756        W4:     MOV     @SWR,R4
   836  005674  042704  177767                BIC     #177767,R4      ;CHECK IF WRITE ONLY
   837  005700  022704  000010                CMP     #10,R4
   838  005704  001370                        BNE     WEX2            ;IF NOT:  BR
   839  005706  000137  004142                JMP     REOT            ;GO REWIND ALL AVAIL TAPES
   840
```

```
                                        ;*************************************************************
                                        ;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.
                                        ;*************************************************************

005712  012737  024463  000716  WTM:    MOV     #MSG62,EMADDR   ;POINT TO TM ERROR MSG
005720  005300                          DEC     R0
005722  005237  000756                  INC     TMFLG           ;SET TM FLAG
005726  005077  172652                  CLR     @MWC            ;CLEAR BYTE COUNTER
005732  012777  026204  172646          MOV     #WDATA,@MDA
005740  012777  000006  172634          MOV     #6,@MTC         ;SET TM OP CODE
005746  012737  005760  000724          MOV     #WTM0,RTRN      ;SAVE RETURN ADDRESS
005754  000137  017006                  JMP     TAPG            ;EXECUTE TM COMMAND
005760  032777  010000  172664  WTM0:   BIT     #10000,@SWR     ;SEE IF SHOULD CHECK ERRORS
005766  001062                          BNE     WTM4            ;IF NOT: BR
005770  004737  016036                  JSR     PC,ERCHK        ;CHECK FOR ERRORS
005774  005737  000742                  TST     SERFL           ;SEE IF STATUS ERROR
006000  001455                          BEQ     WTM4            ;IF NOT: BR
006002  005737  000754                  TST     RTYFL           ;SEE IF RETRY
006006  001401                          BEQ     WTM1            ;IF NOT: BR
006010  000207                          RTS     PC              ;ELSE RETURN TO RETRY ROUTINE
006012  013704  000734          WTM1:   MOV     UNP,R4
006016  005264  001074                  INC     WTER1(R4)       ;BUMP WRITE ERROR
006022  032777  000002  172622          BIT     #2,@SWR         ;SEE IF SHOULD RETRY
006030  001041                          BNE     WTM4            ;IF NOT: BR
006032  042737  147377  000740          BIC     #147377,ERSAV   ;MASK UNRECOVERABLE ERROR
006040  005737  000740                  TST     ERSAV           ;SEE IF RECOVERABLE
006044  001411                          BEQ     WTM2            ;IF SO: BR
006046  012704  023741                  MOV     #MSG52,R4
006052  004737  020536                  JSR     PC,TTOUT        ;PRINT UNRETRYABLE TAG
006056  012704  024463                  MOV     #MSG62,R4
006062  004737  020536                  JSR     PC,TTOUT        ;PRINT TM ERROR TAG
006066  000207                          RTS     PC
006070  005037  000750          WTM2:   CLR     RPCNT           ;CLEAR REPEAT CNTR
006074  013704  000734                  MOV     UNP,R4
006100  005264  001154                  INC     RTY1(R4)        ;BUMP RETRY CNTR
006104  005037  000752                  CLR     RTCNT           ;CLEAR RETRY LOOP CNTR
006110  032777  002000  172534          BIT     #2000,@SWR      ;SEE IF PRINT ERRORS
006116  001004                          BNE     WTM3            ;IF NOT: BR
006120  012704  023761                  MOV     #MSG53,R4
006124  004737  020536                  JSR     PC,TTOUT        ;PRINT ORIGINAL ERROR TAG
006130  004737  006142          WTM3:   JSR     PC,WRTY         ;GO DO RETRY
006134  005037  000756          WTM4:   CLR     TMFLG           ;CLEAR TM FLAG
006140  000207                          RTS     PC              ;EXIT
```

```
                                        ;*************************
                                        ;WRITE ERROR RETRY
                                        ;
                                        ;*************************

006142  012737  000001  000754  WRTY:   MOV     #1,RTYFL        ;SET RETRY FLAG
006150  004737  006504          WRTYO:  JSR     PC,WRTSB        ;GO SPACE BACK FOR REPEAT
006154  005737  000756                  TST     TMFLG           ;SEE IF A TM
006160  001003                          BNE     WRTYTM          ;IF SO: BR
006162  004737  005312                  JSR     PC,WO           ;REWRITE RECORD
006166  000402                          BR      WRTYR           ;CONTINUE
006170  004737  005712          WRTYTM: JSR     PC,WTM          ;GO WRITE TM AGAIN
006174  005737  000742          WRTYR:  TST     SERFL           ;REWRITE GOOD?
006200  001027                          BNE     WRTY2           ;IF NOT: BR
006202  005237  000750                  INC     RPCNT           ;BUMP REPEAT COUNTER
006206  022737  000004  000750          CMP     #4,RPCNT        ;SEE IF FOUR GOOD REPEATS
006214  001355                          BNE     WRTYO           ;IF NOT: DO ANOTHER
006216  032777  002000  172426          BIT     #2000,@SWR      ;SEE IF PRINT
006224  001014                          BNE     WRTY1           ;IF NOT: BR
006226  012704  024007                  MOV     #MSG54,R4
006232  004737  020536                  JSR     PC,TTOUT        ;PRINT RECOVERED MESSAGE
006236  012704  024022                  MOV     #MSG55,R4
006242  004737  020536                  JSR     PC,TTOUT        ;PRINT RETRY TAG
006246  013703  000752                  MOV     RTCNT,R3
006252  004737  020724                  JSR     PC,OCTP         ;PRINT RETRY NUMBER
006256  000207                  WRTY1:  RTS     PC              ;RESUME TESTING
006260  032777  002000  172364  WRTY2:  BIT     #2000,@SWR      ;SEE IF PRINT
006266  001024                          BNE     WRTY3           ;IF NOT: BR
006270  012704  024033                  MOV     #MSG56,R4       ;
006274  004737  020536                  JSR     PC,TTOUT        ;PRINT BAD TAPE SUSPECT
006300  012704  024022                  MOV     #MSG55,R4
006304  004737  020536                  JSR     PC,TTOUT        ;PRINT RETRY TAG
006310  013703  000752                  MOV     RTCNT,R3
006314  004737  020724                  JSR     PC,OCTP         ;PRINT RETRY NUMBER
006320  012704  024055                  MOV     #MSG57,R4
006324  004737  020536                  JSR     PC,TTOUT        ;PRINT REPEAT TAG
006330  013703  000750                  MOV     RPCNT,R3
006334  004737  020724                  JSR     PC,OCTP         ;PRINT REPEAT NUMBER
006340  005737  000752          WRTY3:  TST     RTCNT           ;SEE IF FIRST RETRY
006344  001004                          BNE     WRTY3A          ;IF NOT: BR
006346  013704  000734                  MOV     UNP,R4
006352  005364  001074                  DEC     WTER1(R4)       ;DECREMENT WRITE ERROR CNTR
006356  013704  000734          WRTY3A: MOV     UNP,R4          ;GET UNIT NUMBER
006362  016437  002714  000762          MOV     BTADDR(R4),BTPT ;GET ADDRESS OF UNIT BAD TAPE CNTR
006370  017704  172366                  MOV     @BTPT,R4        ;GET COUNTER
006374  005724                          TST     (R4)+           ;SET POINTER OFFSET
006376  010477  172360                  MOV     R4,@BTPT
006402  013703  000762                  MOV     BTPT,R3
006406  060304                          ADD     R3,R4           ;SET ABSOLUTE POINTER
006410  013714  000720                  MOV     BLCNTR,(R4)     ;SET BLOCK NUMBER
006414  062704  000040                  ADD     #40,R4          ;ADD RCNT OFFSET
006420  013714  000620                  MOV     RCNT,(R4)
006424  160014                          SUB     R0,(R4)         ;SET RECORD NUMBER
006426  005214                          INC     (R4)            ;CORRECT RECORD NUMBER
006430  022777  000040  172324          CMP     #40,@BTPT       ;SEE IF TOO MANY BAD SPOTS
006436  001002                          BNE     WRTY4           ;IF NOT: BR
```

```
                                        ;ELSE GO TO BAD TAPE OVERFLOW
WRTY4:  INC     RTCNT                   ;BUMP RETRY COUNTER
        CMP     #4,RTCNT                ;SEE IF DONE 4 RETRIES
        BEQ     WRTY5                   ;IF SO: BR
        MOV     UNP,R4
        INC     RTY1(R4)                ;BUMP RETRY COUNTER
        INC     ERTFL                   ;SET ERASE FLAG
        JMP     WRTYO                   ;DO NEXT RETRY
WRTY5:  JMP     BTUR                    ;ELSE GO TO BAD TAPE UNRECOVERABLE

        ;WRITE RETRY BACKSPACE-ERASE SUBROUTINE

WRTSB:  CLR     SERFL                   ;CLEAR FLAG
        MOV     #-1,@MWC                ;SET FOR 1 RECORD
        MOV     #MSG69,EMADDR
        JSR     PC,SPBK                 ;DO SPACE BACK
        MOV     #MSG5,EMADDR
        BIT     #2,BTFLG                ;SEE IF ERROR ON BACKSPACE
        BEQ     WRTSBO                  ;IF NOT:  BR
        CLR     RTYFL
        CMP     (SP)+,(SP)+             ;RESET STACK
        BIS     #4,BTFLG                ;MARK RETRY ERROR
        JMP     REOT                    ;REWIND AND REMOVE FROM TESTING
WRTSBO: RTS     PC                      ;RETURN

        ;BAD TAPE OVERFLOW SUBROUTINE**********

BTOV:   MOV     UNP,R4
        INC     RTY1(R4)                ;BUMP RETRY COUNTER
        MOV     #1,BTFLG                ;SET BAD TAPE OVERFLOW FLAG
        TST     (SP)+                   ;RESET STACK
        JMP     REOT                    ;GO REWIND AND REMOVE FROM TESTING

        ;BAD TAPE UNRECOVERABLE SUBROUTINE**********

BTUR:   MOV     #MSG58,R4
        JSR     PC,TTOUT                ;PRINT UNRECOVERABLE BAD SPOT MSG
        RTS     PC                      ;RESUME TESTING
```

```
 986
 987                                    ;*********************************************************
 988                                    ;READ SEQUENCER:
 989                                    ;
 990                                    ;THIS ROUTINE IS USED TO DETERMINE THE SEQUENCE
 991                                    ;IN WHICH READ TAPE OPERATIONS ARE TO BE PERFORMED.
 992                                    ;SWITCH THREE (3) DISALLOWS READING.
 993                                    ;IF THE PROGRAM IS BEING RUN IN THE READ ONLY MODE,
 994                                    ;CONSOLE SWITCH ZERO (0) SET TO A ONE (1), THEN SETTING
 995                                    ;CONSOLE SWITCH FOURTEEN (14) WILL CAUSE READING OF
 996                                    ;THE SAME BLOCK OF DATA CONTINUOUSLY,
 997                                    ;WHEN SET TO A ONE (1), AND ALLOW TAPE
 998                                    ;TO READ BLOCKS PROGRESSIVELY WHEN SET TO A ZERO (0).
 999                                    ;*********************************************************
1000
1001  006626  032777  000010  172016  RSEQ:   BIT     #10,@SWR        ;SEE IF SHOULD READ FORWARD
1002  006634  001031                          BNE     RSEX            ;IF NOT: BR
1003  006636  032777  000001  172006          BIT     #1,@SWR         ;SEE IF WRITE
1004  006644  001404                          BEQ     RSFR0A          ;IF SO: BR
1005  006646  032777  040000  171776          BIT     #40000,@SWR     ;SEE IF SHOULD REMAIN IN PLACE
1006  006654  001410                          BEQ     RSFR0           ;IF NOT: BR
1007  006656  004737  010132          RSFR0A: JSR     PC,BKSP         ;GO BACKSPACE TO START
1008  006662  032737  000002  000746          BIT     #2,BTFLG        ;ERROR ON BACKSPACE?
1009  006670  001402                          BEQ     RSFR0           ;IF NOT:  BR
1010  006672  000137  004142                  JMP     REOT            ;REWIND AND REMOVE FROM TESTING
1011  006676  012737  000002  000626  RSFR0:  MOV     #2,RDCMD        ;LOAD READ FORWARD COMMAND
1012  006704  004737  006722                  JSR     PC,READ         ;GO READ FORWARD
1013  006710  032777  040000  171734          BIT     #40000,@SWR     ;SEE IF SHOULD READ SAME BLOCK
1014  006716  001357                          BNE     RSFR0A          ;IF SO: BR
1015  006720  000207                  RSEX:   RTS     PC              ;EXIT
1016
```

```
                                        ;*********************************************************
                                        ;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).
                                        ;*********************************************************

006722  013700  000620          READ:   MOV     RCNT,R0         ;LOAD REC CNTR
006726  012737  022465  000716          MOV     #MSG6,EMADDR    ;SET ERROR MSG ADDRESS
006734  005037  000756                  CLR     TMFLG           ;CLEAR TM FLAG
006740  052777  040000  171644          BIS     #40000,@MTRD    ;SET TO READ LPC ON READ
006746  005077  171630          RD0:    CLR     @MTC
006752  005077  171622                  CLR     @MTS
006756  013777  000622  171620  RD1:    MOV     CARCNT,@MWC     ;LOAD CHAR CNTR
006764  012777  032216  171614  RD1A:   MOV     #RDATA,@MDA     ;LOAD DATA ADDR
006772  053777  000626  171602          BIS     RDCMD,@MTC      ;LOAD READ OP COMMAND
007000  012737  007012  000724          MOV     #RD2,RTRN       ;SET INTERRUPT RETURN ADDRESS
007006  000137  017006                  JMP     TAPG            ;GO EXECUTE TAPE COMMAND
007012  032777  002000  171560  RD2:    BIT     #2000,@MTS      ;SEE IF AT EOT
007020  001405                          BEQ     RD3             ;IF NOT: BR
007022  052737  100000  000760          BIS     #100000,EOTREC  ;MARK EOT FOUND
007030  000137  007350                  JMP     RDEX            ;GO REWIND
007034  032777  000040  171536  RD3:    BIT     #40,@MTS        ;SEE IF AT LOAD POINT
007042  001411                          BEQ     RD4             ;IF NOT: BR
007044  004737  017506                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
007050  012704  022733                  MOV     #MSG22,R4
007054  004737  020536                  JSR     PC,TTOUT        ;PRINT BOT ERROR
007060  000240                          NOP
007062  000137  020050                  JMP     DRPDRV          ;DROP DRIVE
007066  032777  004000  171556  RD4:    BIT     #4000,@SWR      ;SEE IF SHOULD CHECK ERRORS
007074  001037                          BNE     RD5             ;IF NOT: BR
007076  004737  016036                  JSR     PC,ERCHK        ;GO CHECK ERRORS
007102  005737  000742                  TST     SERFL           ;SEE IF STATUS ERROR
007106  001432                          BEQ     RD5             ;IF NOT:  BR
007110  013704  000734                  MOV     UNP,R4
007114  005264  001114                  INC     RDER1(R4)       ;BUMP READ ERROR
007120  032777  000002  171524          BIT     #2,@SWR         ;SEE IF SHOULD DO READ RETRY
007126  001022                          BNE     RD5             ;IF NOT:  BR
007130  017737  171444  000740          MOV     @MTS,ERSAV
007136  042737  073525  000740          BIC     #073525,ERSAV   ;MASK NON-RETRYABLE ERRORS
007144  001411                          BEQ     RD4A            ;IF RETRYABLE:  BR
007146  012704  023741                  MOV     #MSG52,R4
007152  004737  020536                  JSR     PC,TTOUT        ;PRINT NON-RETRYABLE MESSG.
```

```
1073  007156  012704  022465                MOV     #MSG6,R4
1074  007162  004737  020536                JSR     PC,TTOUT        ;PRINT READ ERROR TAG
1075  007166  000402                        BR      RD5
1076  007170  004737  007376        RD4A:   JSR     PC,RRTY         ;DO RETRY
1077  007174  032777  020000  171450 RD5:    BIT     #20000,@SWR     ;SEE IF SHOULD DO DATA CHECK
1078  007202  001007                        BNE     RD6             ;IF NOT; BR
1079  007204  005737  000756                TST     TMFLG           ;IS IT TM?
1080  007210  001004                        BNE     RD6             ;IF SO:  BR
1081  007212  004737  013712                JSR     PC,DCHK         ;GO CHECK DATA
1082  007216  005037  000742                CLR     SERFL           ;CLEAR STATUS ERROR FLAG
1083  007222  004737  012516        RD6:    JSR     PC,DS3          ;CLEAR BUFFER
1084  007226  032777  000040  171416        BIT     #40,@SWR        ;SEE IF SHOULD YOZZLE
1085  007234  001402                        BEQ     RD7             ;IF NOT: BR
1086  007236  004737  007600                JSR     PC,YOZ          ;ELSE GO YOZZLE
1087  007242  013737  000632  000730 RD7:    MOV     RSTAL,STAL      ;SET DELAY
1088  007250  004737  010634                JSR     PC,STALL        ;STALL
1089  007254  005737  000756                TST     TMFLG           ;JUST DONE TM?
1090  007260  001033                        BNE     RDEX            ;IF SO:  BR
1091  007262  005737  000760                TST     EOTREC          ;WAS EOT REACHED
1092  007266  100430                        BMI     RDEX            ;IF SO:  BR
1093  007270  005300                        DEC     R0
1094  007272  001225                        BNE     RD0             ;IF NOT DONE ALL: BR
1095  007274  005200                        INC     R0              ;ADJUST FOR REC NO IN HEADER
1096  007276  005737  000646        RD10:   TST     TMEX            ;EXPECT A TAPE MARK?
1097  007302  001422                        BEQ     RDEX            ;IF NOT:  BR
1098  007304  005300                        DEC     R0              ;ELSE READ TM
1099  007306  012777  177776  171270        MOV     #-2,@MWC        ;SET BYTE COUNT
1100  007314  005737  001010                TST     STCDFL          ;SEE IF 7 TRK CORE DUMP
1101  007320  001402                        BEQ     1$              ;IF NOT: BR
1102  007322  005277  171256                INC     @MWC            ;SET TO ONE CHAR
1103  007326  005237  000756        1$:     INC     TMFLG           ;SET TM FLAG
1104  007332  012737  024567  000716        MOV     #MSG70,EMADDR
1105  007340  042777  040000  171244        BIC     #40000,@MTRD    ;SET TO READ LPC ON READ TM
1106  007346  000606                        BR      RD1A            ;GO READ
1107  007350  005037  000756        RDEX:   CLR     TMFLG
1108  007354  005737  000760                TST     EOTREC          ;WAS EOT REACHED
1109  007360  100005                        BPL     RDEXX           ;IF NOT:  BR
1110  007362  005726                        TST     (SP)+           ;RESET STACK
1111  007364  005037  000760                CLR     EOTREC          ;CLEAR EOT IND.
1112  007370  000137  004142                JMP     REOT            ;GO REWIND
1113  007374  000207                RDEXX:  RTS     PC              ;EXIT
1114
1115
1116
1117                                        ;***********************************************
1118                                        ;READ ERROR RETRY
1119                                        ;
1120                                        ;***********************************************
1121
1122  007376  005237  000774        RRTY:   INC     RRTYFL          ;SET READ RETRY FLAG
1123  007402  032777  002000  171242        BIT     #2000,@SWR      ;SEE IF PRINT?
1124  007410  001004                        BNE     RRTY0           ;IF NOT:  BR
1125  007412  012704  023761                MOV     #MSG53,R4
1126  007416  004737  020536                JSR     PC,TTOUT        ;PRINT ORIGINAL ERROR MESSG
1127  007422  005037  000752        RRTY0:  CLR     RTCNT           ;CLEAR RETRY COUNT
1128  007426  004737  007600        RRTY1:  JSR     PC,YOZ          ;GO REREAD
```

```
1129  007432  005237  000752                 INC     RTCNT           ;BUMP RETRY COUNT
1130  007436  005737  000742                 TST     SERFL           ;SEE IF ERROR?
1131  007442  001431                         BEQ     RRTY4           ;IF NOT:  BR
1132  007444  032777  002000  171200         BIT     #2000,@SWR      ;SEE IF PRINT?
1133  007452  001010                         BNE     RRTY2           ;IF NOT:  BR
1134  007454  012704  024576                 MOV     #MSG71,R4
1135  007460  004737  020536                 JSR     PC,TTOUT        ;PRINT FAILED RETRY MESSG.
1136  007464  013703  000752                 MOV     RTCNT,R3
1137  007470  004737  020724                 JSR     PC,OCTP         ;PRINT RETRY NUMBER
1138  007474  022737  000004  000752 RRTY2:  CMP     #4,RTCNT        ;DONE 4 RETRYS?
1139  007502  001351                         BNE     RRTY1           ;IF NOT:  BR
1140  007504  012704  024624                 MOV     #MSG72,R4
1141  007510  004737  020536                 JSR     PC,TTOUT        ;PRINT SUSPECT HARD ERROR MESSG.
1142  007514  013704  000734         RRTY3:  MOV     UNP,R4
1143  007520  005264  001214                 INC     BDRTY1(R4)      ;BUMP HARD ERROR COUNT
1144  007524  000420                         BR      RRTYX
1145  007526  032777  002000  171116 RRTY4:  BIT     #2000,@SWR      ;SEE IF SHOULD PRINT?
1146  007534  001010                         BNE     RRTY5           ;IF NOT:  BR
1147  007536  012704  024650                 MOV     #MSG73,R4
1148  007542  004737  020536                 JSR     PC,TTOUT        ;TYPE SUCCESSFUL RETRY MESSAGE
1149  007546  013703  000752                 MOV     RTCNT,R3
1150  007552  004737  020724                 JSR     PC,OCTP         ;PRINT RETRY COUNT
1151  007556  013704  000734         RRTY5:  MOV     UNP,R4
1152  007562  005264  001174                 INC     GDRTY1(R4)      ;INCREASE SOFT ERROR COUNT
1153  007566  005037  000774         RRTYX:  CLR     RRTYFL          ;CLEAR RETRY FLAG
1154  007572  004737  022126                 JSR     PC,CKSWR        ;GO CHECK FOR  G
1155  007576  000207                         RTS     PC              ;RETURN
1156
```

```
1157
1158                                    ;*********************************************************
1159                                    ;YOZZLE SUBROUTINE:
1160                                    ;
1161                                    ;THIS SUBROUTINE, ENTERED VIA SWITCH FIVE (5), IS USED TO PERFORM
1162                                    ;A CONTINUOUS READ AND SPACE OVER OF THE CURRENT RECORD ON TAPE.
1163                                    ;FULL STATUS AND DATA CHECKING MAY BE PERFORMED
1164                                    ;OR NOT VIA CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13).
1165                                    ;A SOFTWARE DELAY IS PERFORMED BETWEEN EACH READ
1166                                    ;AND SPACE OPERATION AND MAY BE VARIED BY TYPING
1167                                    ;CNTRL C ON THE TTY AND ENTERING A VALUE IN RESPONSE
1168                                    ;TO THE PRINTED REQUEST.
1169                                    ;*********************************************************
1170
1171  007600  012777  000001  171050  YOZ:    MOV     #1,@TKS         ;SET TTY ENABLE
1172  007606  013737  000640  000730          MOV     YSTAL,STAL
1173  007614  004737  010634                 JSR     PC,STALL        ;DO YOZZLE STALL
1174  007620  012777  177777  170756  YOZO:   MOV     #-1,@MWC        ;SET TO 1 RECORD SPACING
1175  007626  112777  000012  170746  YOZA:   MOVB    #12,@MTC        ;SET TO SPACE REVERSE
1176  007634  012737  007654  000724  YOZB:   MOV     #YOZC,RTRN      ;SET RETURN ADDRESS
1177  007642  012737  177775  000730          MOV     #177775,STAL    ;SET TIME MULTIPLIER
1178  007650  000137  017006                 JMP     TAPG            ;GO YOZZLE
1179  007654  013737  000640  000730  YOZC:   MOV     YSTAL,STAL
1180  007662  004737  010634                 JSR     PC,STALL        ;DO YOZZLE STALL
1181  007666  113777  000626  170706          MOVB    RDCMD,@MTC      ;SET READ COMMAND F OR R
1182  007674  012777  032216  170704          MOV     #RDATA,@MDA     ;SET READ ADDRESS
1183  007702  013777  000622  170674          MOV     CARCNT,@MWC     ;SET CHARACTER COUNT
1184  007710  005737  000756                 TST     TMFLG           ;IS IT A TM?
1185  007714  001410                          BEQ     YOZC1           ;IF NOT:  BR
1186  007716  012777  177776  170660          MOV     #-2,@MWC        ;SET FOR TM
1187  007724  005737  001010                 TST     STCDFL          ;SEE IF 7 TRK CORE DUMP
1188  007730  001402                          BEQ     YOZC1           ;IF NOT: BR
1189  007732  005277  170646                 INC     @MWC            ;SET TO ONE CHARACTER
1190  007736  012737  007750  000724  YOZC1:  MOV     #YOZD,RTRN      ;SET RETURN ADDRESS
1191  007744  000137  017006                 JMP     TAPG            ;GO YOZZLE
1192  007750  032777  004000  170674  YOZD:   BIT     #4000,@SWR      ;SEE IF ERROR CHECK
1193  007756  001002                          BNE     YOZE            ;IF NOT: BR
1194  007760  004737  016036                 JSR     PC,ERCHK        ;ELSE GO CHECK ERRORS
1195  007764  005737  000774          YOZE:   TST     RRTYFL          ;IS IT A READ RETRY?
1196  007770  001401                          BEQ     YOZE1           ;IF NOT:  BR
1197  007772  000207                          RTS     PC
1198  007774  032777  020000  170650  YOZE1:  BIT     #20000,@SWR     ;SEE IF SHOULD CHECK DATA
1199  010002  001002                          BNE     YOZF            ;IF NOT: BR
1200  010004  004737  013712                 JSR     PC,DCHK         ;ELSE GO CHECK DATA
1201  010010  004737  012516          YOZF:   JSR     PC,DS3          ;GO CLEAR DATA AREA
1202  010014  105777  170636                 TSTB    @TKS            ;SEE IF HAVE NEW STALL VALUE
1203  010020  100034                          BPL     YOZG            ;IF NOT: BR
1204  010022  122777  000203  170630          CMPB    #203,@TKB       ;SEE IF CONT C
1205  010030  001030                          BNE     YOZG            ;IF NOT: BR
1206  010032  012704  023606                 MOV     #MSG44,R4
1207  010036  004737  020536                 JSR     PC,TTOUT        ;PRINT YSTALL REQUEST
1208  010042  013703  000640                 MOV     YSTAL,R3
1209  010046  004737  020724                 JSR     PC,OCTP         ;PRINT PRESENT STALL
1210  010052  010037  000712                 MOV     R0,TEMP3        ;SAVE R0(REC CNT)
1211  010056  012705  000640                 MOV     #YSTAL,R5       ;SET ADDRESS OF YSTL
1212  010062  012701  000006                 MOV     #6,R1           ;SET NUMBER OF CHAR TO INPUT
```

```
1213  010066  012702  177777                    MOV     #-1,R2          ;SET MAXIMUM LIMIT
1214  010072  012703  001000                    MOV     #1000,R3        ;SET MINIMUM LIMIT
1215  010076  004737  020272                    JSR     PC,TTR          ;GO GET VALUE
1216  010102  013700  000712                    MOV     TEMP3,R0        ;RESTORE R0(REC CNTR)
1217  010106  000137  007600                    JMP     YOZ             ;RESTART YOZZLE
1218  010112  032777  000040  170532  YOZG:     BIT     #40,@SWR        ;SEE IF SHOULD CONTINUE YOZZLE
1219  010120  001227                            BNE     YOZ             ;IF SO: BR
1220  010122  012777  000100  170526            MOV     #100,@TKS       ;SET TTY INTERRUPT ENABLE
1221  010130  000207                            RTS     PC              ;EXIT
1222
```

```
1223
1224                                    ;*********************************************************
1225                                    ;BACKSPACE SUBROUTINE:
1226                                    ;
1227                                    ;THIS SUBROUTINE IS USED TO PERFORM THE
1228                                    ;BACKSPACE OPERATION REQUIRED BY THE READ
1229                                    ;ROUTINE EITHER FOR READ FORWARD AFTER WRITING,
1230                                    ;OR FOR CONTINUOUS READING OF A DATA BLOCK
1231                                    ;WHEN IN READ ONLY MODE WITH SWITCH FOURTEEN (14)
1232                                    ;SET TO A ONE.
1233                                    ;A CHECK FOR RECORD COUNT ZERO IS MADE AT THE
1234                                    ;END OF THE SPACE OPERATION TO ASSURE THAT PROPER
1235                                    ;TAPE POSITIONING WAS DONE.
1236                                    ;*********************************************************
1237
1238  010132  005037  000726    BKSP:   CLR     HDRFL           ;CLEAR HEADER FLAG
1239  010136  013700  000620    BO.     MOV     RCNT,R0
1240  010142  005100                    COM     R0              ;BUILD SPACE AMOUNT
1241  010144  005200                    INC     R0
1242  010146  005737  000760            TST     EOTREC          ;SEE IF EOT WAS REACHED
1243  010152  001407                    BEQ     BK0             ;IF NOT:  BR
1244  010154  013700  000760            MOV     EOTREC,R0       ;GET SHORTENED BLOCK COUNT
1245  010160  042700  100000            BIC     #100000,R0
1246  010164  005400                    NEG     R0
1247  010166  005037  000760            CLR     EOTREC          ;CLEAR EOT FLAG
1248  010172  010037  000714    BK0.    MOV     R0,TEMP4        ;SAVE BACKSPACE COUNT
1249  010176  005737  000646            TST     TMEX            ;IS THERE A TM?
1250  010202  001520                    BEQ     BOA             ;IF NOT: BR
1251  010204  012737  024472  000716    MOV     #MSG63,EMADDR   ;POINT TO TM SP ERROR MESG
1252  010212  012777  177777  170364    MOV     #-1,@MWC        ;SET FOR 1 RECORD
1253  010220  013700  000620            MOV     RCNT,R0         ;R0=RECORD COUNT
1254  010224  063700  000714            ADD     TEMP4,R0        ;R0=RCNT-BACKSPACE CNT FOR HEADER
1255  010230  004737  010466            JSR     PC,SPBK         ;BACKSPACE OVER TM
1256  010234  032737  000002  000746    BIT     #2,BTFLG        ;WAS THERE AN ERROR
1257  010242  001401                    BEQ     BK1             ;IF NOT: BR
1258  010244  000207                    RTS     PC              ;ELSE RETURN WITH ERROR FLAG SET
1259  010246  017737  170326  000740  BK1:    MOV     @MTS,ERSAV      ;GET STATUS
1260  010254  032737  040000  000740        BIT     #40000,ERSAV    ;IS TM SET
1261  010262  001053                        BNE     BK3             ;IF SET: BR
1262  010264  005737  000726          BK1C: TST     HDRFL           ;ALREADY PRINTED HEADER?
1263  010270  001014                        BNE     BK2             ;IF SO: BR
1264  010272  032777  002000  170352        BIT     #2000,@SWR      ;SHOULD PRINT?
1265  010300  001040                        BNE     BK1B            ;IF NOT:  BR
1266  010302  004737  017506                JSR     PC,PAPRT        ;PRINT HEADER
1267  010306  013704  000716                MOV     EMADDR,R4       ;POINT TO TM SP ERROR
1268  010312  004737  020536                JSR     PC,TTOUT        ;PRINT ERROR
1269  010316  012704  022756                MOV     #MSG23,R4
1270  010322  004737  020536          BK2:  JSR     PC,TTOUT        ;PRINT COMMAND HEADER
1271  010326  017703  170250                MOV     @MTC,R3
1272  010332  005037  000712                CLR     TEMP3
1273  010336  000303                  BK1A: SWAB    R3              ;POSITION MOST SIGNIFICANT
1274  010340  004737  021152                JSR     PC,DOUT         ;PRINT
1275  010344  000303                        SWAB    R3              ;POSITION LEAST SIGNIFICANT
1276  010346  004737  021152                JSR     PC,DOUT         ;PRINT
1277  010352  005737  000712                TST     TEMP3           ;SEE IF PRINTED STATUS
1278  010356  001011                        BNE     BK1B            ;IF SO.  BR
```

```
1279  010360  005237  000712                    INC     TEMP3           ;SET FLAG
1280  010364  012704  023143                    MOV     #MSG30,R4       ;PRINT STATUS HEADER
1281  010370  004737  020536                    JSR     PC,TTOUT
1282  010374  017703  170200                    MOV     @MTS,R3         ;LOAD STATUS
1283  010400  000756                            BR      BK1A            ;GO PRINT STATUS
1284  010402  052737  000002  000746  BK1B:     BIS     #2,BTFLG        ;SET BT FLAG (POSITION ERROR)
1285  010410  000207                            RTS     PC              ;RETURN
1286  010412  042737  142121  000740  BK3:      BIC     #142121,ERSAV   ;LOOK FOR NON-TM ERRORS
1287  010420  001407                            BEQ     BOB             ;IF NOT:  BR
1288  010422  005737  000760                    TST     EOTREC          ;WAS EOT REACHED
1289  010426  001716                            BEQ     BK1C            ;IF NOT:  BR
1290  010430  042737  002000  000740            BIC     #2000,ERSAV     ;CHECK FOR NON-EOT ERRORS
1291  010436  001312                            BNE     BK1C            ;IF ANY: BR
1292
1293  010440  163700  000620          BOB:      SUB     RCNT,R0         ;AGAIN R0=BACKSPACE COUNT
1294  010444  012737  022477  000716  BOA:      MOV     #MSG10,EMADDR   ;POINT TO SE MESSG
1295  010452  005200                            INC     R0              ;R0=BACKSPACE COUNT+1
1296  010454  063700  000620                    ADD     RCNT,R0         ;R0=RCNT-BACKSPACE CNT + 1 FOR HEADER
1297  010460  013777  000714  170116            MOV     TEMP4,@MWC
1298  010466  013737  000636  000730  SPBK:     MOV     TSTAL,STAL
1299  010474  004737  010634                    JSR     PC,STALL        ;DO STALL
1300  010500  005077  170074                    CLR     @MTS
1301  010504  105077  170072                    CLRB    @MTC
1302  010510  052777  000012  170064            BIS     #12,@MTC        ;SET BACKSPACE OP
1303  010516  012737  010536  000724            MOV     #B1,RTRN        ;SET RETURN ADDRESS
1304  010524  012737  177377  000730            MOV     #177377,STAL    ;SET INTERRUPT TIME MULTIPLIER
1305  010532  000137  017006                    JMP     TAPG            ;GO DO SPACE
1306  010536  017701  170042          B1:       MOV     @MWC,R1         ;LOAD SPACE COUNTER
1307  010542  001426                            BEQ     B2              ;IF COUNT IS ZERO:  BR
1308  010544  032777  002000  170100            BIT     #2000,@SWR      ;SEE IF PRINT
1309  010552  001017                            BNE     B1A             ;IF NOT:  BR
1310  010554  004737  017506                    JSR     PC,PAPRT        ;ELSE PRINT SPACE ERROR
1311  010560  013704  000716                    MOV     EMADDR,R4
1312  010564  004737  020536                    JSR     PC,TTOUT
1313  010570  012704  023635                    MOV     #MSG45,R4
1314  010574  004737  020536                    JSR     PC,TTOUT        ;PRINT SPACE COUNT HEADER
1315  010600  005301                            DEC     R1
1316  010602  005101                            COM     R1
1317  010604  010103                            MOV     R1,R3
1318  010606  004737  020724                    JSR     PC,OCTP         ;PRINT NUMBER OF RECORDS LEFT TO SPACE
1319  010612  012737  000002  000746  B1A:      MOV     #2,BTFLG        ;SET BAD TAPE FLAG
1320  010620  013737  000636  000730  B2:       MOV     TSTAL,STAL      ;DO STALL
1321  010626  004737  010634                    JSR     PC,STALL        ;STALL
1322  010632  000207                            RTS     PC              ;EXIT
1323
```

```
                                        ;*********************************************************
                                        ;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.
                                        ;*********************************************************

010634  005337  000730        STALL:    DEC     STAL
010640  001375                        BNE     STALL           ;DELAY
010642  000207                        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.
                                        ;*********************************************************

010644  012701  177760        CCNTR:    MOV     #-20,R1         ;SET HIGH LIMIT
010650  012702  174000                  MOV     #-4000,R2       ;SET LOW LIMIT
010654  004737  020240                  JSR     PC,RANG         ;GO GENERATE NUMBER
010660  013737  000676  000622          MOV     RANSAV,CARCNT   ;SET CHAR COUNT
010666  012737  177777  012750          MOV     #-1,PATS        ;PRESET DATA PATTERN
010674  000207                        RTS     PC              ;EXIT
```

```
                                        ;*********************************************************
                                        ;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.
                                        ;*********************************************************

010676  012702  000001          RCNTR:  MOV     #1,R2           ;SET LOW LIMIT
010702  012701  000500                  MOV     #500,R1         ;SET HIGH LIMIT
010706  004737  020240                  JSR     PC,RANG         ;GO GENERATE NUMBER
010712  013737  000676  000620          MOV     RANSAV,RCNT     ;SET RECORD COUNT
010720  000207                          RTS     PC              ;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.
                                        ;*********************************************************

010722  005737  000700          TINP:   TST     TINF            ;SEE IF SHOULD INPUT FROM TTY
010726  001001                          BNE     TINPA           ;IF SO: BR
010730  000207                          RTS     PC              ;EXIT
010732  005037  000734          TINPA:  CLR     UNP             ;CLEAR TABLE POINTER
010736  005037  004716                  CLR     REOTC           ;CLEAR EOT UNIT COUNTER
010742  012700  000010                  MOV     #10,R0          ;SET SIZE OF TABLE
010746  012701  001012                  MOV     #UN1,R1         ;SET START OF TABLE
010752  005021                  *TINPB: CLR     (R1)+           ;CLEAR TABLE
010754  005300                          DEC     R0              ;SEE IF DONE
010756  001375                          BNE     TINPB           ;IF NOT: BR
010760  005737  021632                  TST     ASEQF           ;SEE IF AUTO SEQUENCE
010764  001405                          BEQ     TINPB1          ;IF NOT:  BR
```

```
1424  010766  012704  024737                  MOV     #MSG77,R4
1425  010772  004737  020536                  JSR     PC,TTOUT        ;PRINT AUTO SEQ PROGRAM NAME
1426  010776  000410                          BR      TINPO
1427  011000  012704  023152          TINPB1. MOV     #MSG31,R4
1428  011004  004737  020536                  JSR     PC,TTOUT        ;PRINT PROGRAM NAME
1429  011010  012704  023267                  MOV     #MSG31A,R4
1430  011014  004737  020536                  JSR     PC,TTOUT        ;PRINT REST OF TITLE
1431  011020  122737  000004  000041  TINPO:  CMPB    #4,@#41         ;SEE IF LOAD MEDIUN
1432  011026  001006                          BNE     1$              ;IF NOT: BR
1433  011030  012704  026124                  MOV     #MSG97,R4
1434  011034  004737  020536                  JSR     PC,TTOUT        ;ELSE PRINT NO TEST
1435  011040  000137  004662                  JMP     REOT10          ;END TEST
1436  011044  012704  025261          1$:     MOV     #MSG84,R4
1437  011050  004737  020536                  JSR     PC,TTOUT        ;REQUEST STARTING REGISTER ADDRESS
1438  011054  013703  000600                  MOV     MTS,R3
1439  011060  004737  020724                  JSR     PC,OCTP         ;PRINT CURRINT REGISTER START
1440  011064  013705  000674                  MOV     REGST,R5        ;SAVE ADDRESS LOCATION
1441  011070  012701  000006                  MOV     #6,R1           ;SET SIZE OF ENTRY
1442  011074  012702  177770                  MOV     #177770,R2      ;SET UPPER LIMIT
1443  011100  012703  170000                  MOV     #170000,R3      ;SET LOWER LIMIT
1444  011104  004737  020272                  JSR     PC,TTR          ;GO GET RESPONSE
1445  011110  012705  000602                  MOV     #MTC,R5         ;SET TABLE BASE
1446  011114  013704  000600                  MOV     MTS,R4          ;GET INITIAL ADDRESS
1447  011120  062704  000002          2$:     ADD     #2,R4           ;BUMP ADDRESS
1448  011124  010425                          MOV     R4,(R5)+        ;FILL TABLE
1449  011126  020527  000614                  CMP     R5,#MTRD+2      ;DONE?
1450  011132  001372                          BNE     2$              ;IF NOT: BR
1451  011134  012704  025304                  MOV     #MSG85,R4
1452  011140  004737  020536                  JSR     PC,TTOUT        ;REQUEST VECTOR ADDR.
1453  011144  013703  000614                  MOV     VECT,R3
1454  011150  004737  020724                  JSR     PC,OCTP         ;PRINT CURRENT VECTOR
1455  011154  012705  000614                  MOV     #VECT,R5        ;SET SAVE LOCATION
1456  011160  012701  000003                  MOV     #3,R1           ;SET SIZE OF RSPONSE
1457  011164  012702  000476                  MOV     #476,R2         ;SET UPPER LIMIT
1458  011170  012703  000060                  MOV     #60,R3          ;SET LOWER LIMIT
1459  011174  004737  020272                  JSR     PC,TTR          ;GO GET RESPONSE
1460  011200  013700  000614                  MOV     VECT,R0         ;GET VECTOR ADDRESS
1461  011204  012720  017466                  MOV     #MTINT,(R0)+    ;LOAD VECTOR WITH HANDLER ADDR.
1462  011210  012710  000340                  MOV     #340,(R0)       ;LOAD PRIORITY LEVEL
1463  011214  005737  021632                  TST     ASEQF           ;SEE IF AUTO SEQ
1464  011220  001403                          BEQ     TINPOO          ;IF NOT: BR
1465  011222  005726                          TST     (SP)+           ;RESET STACK
1466  011224  000137  021260                  JMP     ASEQ            ;GO TO AUTO SEQ
1467  011230  012704  023322  TINPOO:  MOV     #MSG32,R4
1468  011234  004737  020536                  JSR     PC,TTOUT        ;PRINT UNIT NUMBER REQUEST
1469  011240  005037  000710                  CLR     TEMP2           ;CLEAR BUFFER
1470  011244  012705  000710                  MOV     #TEMP2,R5       ;SET UNIT DESCRIPTION BUFFER ADDRESS
1471  011250  012701  000001                  MOV     #1,R1           ;SET NUMBER OF CHARACTERS TO INPUT
1472  011254  012702  000007                  MOV     #7,R2           ;SET MAXIMUM LIMIT
1473  011260  012703  000000                  MOV     #0,R3           ;SET MINIMUM LIMIT
1474  011264  004737  020272                  JSR     PC,TTR          ;GO GET UNIT NUMBER
1475  011270  005737  000706                  TST     TEMP1           ;SEE IF HAVE NEW PARAMETER
1476  011274  001014                          BNE     TINPOB          ;IF SO: BR
1477  011276  005737  000734                  TST     UNP             ;SEE IF FIRST ENTRY
1478  011302  001002                          BNE     TINPOA          ;IF NOT: BR
1479  011304  000137  011230                  JMP     TINPOO          ;ELSE RETRY
```

```
1480  011310  013700  000734          TINPOA: MOV     UNP,R0
1481  011314  012760  177777  001012          MOV     #-1,UN1(R0)     ;SET END UNIT TABLE
1482  011322  000137  011634                  JMP     TINP28          ;GO GET RECORD COUNT
1483  011326  013700  000734          TINPOB: MOV     UNP,R0
1484  011332  042760  003400  001012          BIC     #3400,UN1(R0)   ;CLEAR UNIT NUMBER
1485  011340  012703  000010                  MOV     #10,R3          ;SET ROTATION FACTOR
1486  011344  004737  012342                  JSR     PC,TPOS         ;GO LOAD UNIT NUMBER TO PROPER POSITION
1487  011350  016037  001012  000616          MOV     UN1(R0),UDES    ;SELECT UNIT
1488  011356  013777  000616  167216          MOV     UDES,@MTC       ;LOAD UNIT NUMBER
1489  011364  032777  000100  167206  TINPOC: BIT     #100,@MTS       ;SEE IF UNIT AVAILABLE
1490  011372  001011                          BNE     TINPOD          ;IF R0: BR
1491  011374  005337  000730                  DEC     STAL
1492  011400  001371                          BNE     TINPOC          ;DELAY
1493  011402  012704  023707                  MOV     #MSG49,R4
1494  011406  004737  020536                  JSR     PC,TTOUT        ;PRINT UNIT NOT AVAILABLE
1495  011412  000137  011230                  JMP     TINPOO          ;REDO
1496  011416  032777  000020  167154  TINPOD: BIT     #20,@MTS        ;SEE IF 7 CHANNEL
1497  011424  001404                          BEQ     TINPOE          ;IF NOT: BR
1498  011426  012704  023723                  MOV     #MSG50,R4       ;7 CHANNEL MSG
1499  011432  000137  011442                  JMP     TINPOF
1500  011436  012704  023732          TINPOE: MOV     #MSG51,R4       ;9 CHANNEL MSG
1501  011442  004737  020536          TINPOF: JSR     PC,TTOUT        ;GO PRINT 7 OR 9 CHANNEL
1502  011446  012704  023342          TINP1:  MOV     #MSG33,R4
1503  011452  004737  020536                  JSR     PC.TTOUT        ;PRINT DENSITY REQUEST
1504  011456  005037  000710                  CLR     TEMP2           ;CLEAR BUFFER
1505  011462  012701  000001                  MOV     #1,R1           ;SET NUMBER OF CHARACTERS TO INPUT
1506  011466  012702  000003                  MOV     #3,R2           ;SET MAXIMUM LIMIT
1507  011472  012703  000000                  MOV     #0,R3           ;SET MINIMUM LIMIT
1508  011476  004737  020272                  JSR     PC,TTR          ;GO GET DENSITY
1509  011502  005737  000706                  TST     TEMP1           ;SEE IF HAVE NEW PARAMETER
1510  011506  001407                          BEQ     TINP2           ;IF NOT: BR
1511  011510  042737  060000  000616          BIC     #60000,UDES     ;ELSE CLEAR OLD PARAMETER
1512  011516  012703  000015                  MOV     #15,R3          ;SET POSITION FACTOR
1513  011522  004737  012342                  JSR     PC.TPOS         ;GO LOAD DENSITY INTO PROPER POSITION
1514  011526  012704  023356          TINP2:  MOV     #MSG34,R4
1515  011532  004737  020536                  JSR     PC,TTOUT        ;PRINT PARITY REQUEST
1516  011536  005037  000710                  CLR     TEMP2           ;CLR BUFFER
1517  011542  012701  000001                  MOV     #1,R1           ;SET NUMBER OF CHARACTERS TO INPUT
1518  011546  012702  000001                  MOV     #1,R2           ;SET MAXIMUM LIMIT
1519  011552  012703  000000                  MOV     #0,R3           ;SET MINIMUM LIMIT
1520  011556  004737  020272                  JSR     PC,TTR          ;GO INPUT PARITY
1521  011562  005737  000706                  TST     TEMP1           ;SEE IF HAVE NEW PARAMETER
1522  011566  001407                          BEQ     TINP2A          ;IF NOT: BR
1523  011570  042737  004000  000616          BIC     #4000,UDES      ;ELSE CLEAR OLD PARAMETER
1524  011576  012703  000013                  MOV     #13,R3          ;SET POSITION FACTOR
1525  011602  004737  012342                  JSR     PC,TPOS         ;GO LOAD PARITY TO PROPER POSITION
1526  011606  005237  004716          TINP2A: INC     REOTC           ;BUMP EOT UNIT COUNTER
1527  011612  022737  000016  000734          CMP     #16,UNP         ;SEE IF DONE UNITS
1528  011620  001405                          BEQ     TINP2B          ;IF SO: BR
1529  011622  062737  000002  000734          ADD     #2,UNP          ;POINT TO NEXT UNIT
1530  011630  000137  011230                  JMP     TINPOO          ;ELSE LOOK FOR NEXT UNIT
1531  011634  005037  000734          TINP2B: CLR     UNP             ;CLEAR UNIT POINTER
1532  011640  013700  004716                  MOV     REOTC,R0
1533  011644  000337  004716                  SWAB    REOTC
1534  011650  110037  004716                  MOVB    R0,REOTC        ;SET UNIT EOT COUNTER
1535  011654  012704  023371          TINP3:  MOV     #MSG35,R4
```

```
1536  011660  004737  020536                 JSR     PC,TTOUT        ;PRINT RECORD COUNT REQUEST
1537  011664  013703  000620                 MOV     RCNT,R3
1538  011670  004737  020724                 JSR     PC,OCTP         ;PRINT RECORD COUNT
1539  011674  012705  000620                 MOV     #RCNT,R5        ;SET RECORD COUNT ADDRESS
1540  011700  012701  000006                 MOV     #6,R1           ;SET NUMBER OF CHARACTERS TO INPUT
1541  011704  012702  177777                 MOV     #-1,R2          ;SET MAXIMUM LIMIT
1542  011710  012703  000001                 MOV     #1,R3           ;SET MINIMUM LIMIT
1543  011714  004737  020272                 JSR     PC,TTR          ;GO GET RECORD COUNT
1544  011720  013737  000620  000642         MOV     RCNT,RCSAV      ;SAVE RECORD COUNT
1545  011726  012704  023412                 MOV     #MSG36,R4
1546  011732  004737  020536                 JSR     PC,TTOUT        ;PRINT CHARACTER COUNT REQUEST
1547  011736  005437  000622                 NEG     CARCNT
1548  011742  013703  000622                 MOV     CARCNT,R3
1549  011746  004737  020724                 JSR     PC,OCTP         ;PRINT CHAR COUNT
1550  011752  012705  000622                 MOV     #CARCNT,R5      ;SET CHARACTER COUNT ADDRESS
1551  011756  012701  000006                 MOV     #6,R1           ;SET NUMBER OF CHARACTERS TO INPUT
1552  011762  012702  004000                 MOV     #4000,R2        ;SET MAXIMUM LIMIT
1553  011766  012703  000004                 MOV     #4,R3           ;SET MINIMUM LIMIT
1554  011772  004737  020272                 JSR     PC,TTR          ;GO GET CHARACTER COUNT
1555  011776  005437  000622                 NEG     CARCNT          ;SET TO TWO'S COMPLEMENT
1556  012002  013737  000622  000644         MOV     CARCNT,CCSAV    ;SAVE CHAR COUNT
1557  012010  012704  023436                 MOV     #MSG37,R4       ;PRINT PATTERN NUMBER REQUEST
1558  012014  004737  020536                 JSR     PC,TTOUT
1559  012020  013703  000624                 MOV     PATRN,R3
1560  012024  004737  020724                 JSR     PC,OCTP         ;PRINT PATTERN
1561  012030  005037  012746                 CLR     DOFL            ;CLEAR EXTERNAL DATA FLAG
1562  012034  012705  000624                 MOV     #PATRN,R5       ;SET PATTERN NUMBER ADDRESS
1563  012040  012701  000002                 MOV     #2,R1           ;SET NUMBER OF CHARACTERS TO INPUT
1564  012044  012702  000015                 MOV     #15,R2          ;SET MAXIMUM LIMIT
1565  012050  012703  000000                 MOV     #0,R3           ;SET MINIMUM LIMIT
1566  012054  004737  020272                 JSR     PC,TTR          ;GO GET PATTERN NUMBER
1567  012060  012704  024274                 MOV     #MSG60,R4       ;PRINT TM REQUEST
1568  012064  004737  020536                 JSR     PC,TTOUT
1569  012070  013703  000646                 MOV     TMEX,R3
1570  012074  004737  020724                 JSR     PC,OCTP         ;PRINT TMEX VALUE
1571  012100  012705  000646                 MOV     #TMEX,R5        ;SE TMEX ADDRESS
1572  012104  012701  000001                 MOV     #1,R1           ;SET NUMBER OF CHARACTERS TO INPUT
1573  012110  010102                         MOV     R1,R2           ;SET MAXIMUM LIMIT
1574  012112  005003                         CLR     R3              ;SET MINIMUM LIMIT
1575  012114  004737  020272                 JSR     PC,TTR          ;GO GET RESPONSE
1576  012120  012704  023461                 MOV     #MSG38,R4
1577  012124  004737  020536                 JSR     PC,TTOUT        ;PRINT SINGLE PASS REQUEST
1578  012130  013703  000630                 MOV     SPFLG,R3
1579  012134  004737  020724                 JSR     PC,OCTP         ;PRINT CURRENT FLAG SETTING
1580  012140  012705  000630                 MOV     #SPFLG,R5       ;GET ADDRESS OF FLAG
1581  012144  012701  000001                 MOV     #1,R1           ;SET SIZE OF RESPONSE
1582  012150  012702  000001                 MOV     #1,R2           ;SET UPPER LIMIT
1583  012154  012703  000000                 MOV     #0,R3           ;SET LOWER LIMIT
1584  012160  004737  020272                 JSR     PC,TTR          ;GO GET RESPONSE
1585  012164  012704  023521         TINP4:  MOV     #MSG40,R4
1586  012170  004737  020536                 JSR     PC,TTOUT        ;PRINT READ STALL REQUEST
1587  012174  013703  000632                 MOV     RSTAL,R3
1588  012200  004737  020724                 JSR     PC,OCTP         ;PRINT READ STALL
1589  012204  012705  000632                 MOV     #RSTAL,R5       ;SET READ STALL ADDRESS
1590  012210  012701  000006                 MOV     #6,R1           ;SET NUMBER OF CHARACTERS TO INPUT
1591  012214  012702  177777                 MOV     #-1,R2          ;SET MAXIMUM LIMIT
```

```
1592  012220  012703  000001                MOV     #1,R3           ;SET MINIMUM LIMIT
1593  012224  004737  020272                JSR     PC,TTR          ;GO GET READ STALL
1594  012230  012704  023550                MOV     #MSG41,R4
1595  012234  004737  020536                JSR     PC,TTOUT        ;PRINT WRITE STALL REQUEST
1596  012240  013703  000634                MOV     WSTAL,R3
1597  012244  004737  020724                JSR     PC,OCTP         ;PRINT READ STALL
1598  012250  012705  000634                MOV     #WSTAL,R5       ;SET WRITE STALL ADDRESS
1599  012254  012701  000006                MOV     #6,R1           ;SET NUMBER OF CHARACTERS TO INPUT
1600  012260  012702  177777                MOV     #-1,R2          ;SET MAXIMUM LIMIT
1601  012264  012703  000001                MOV     #1,R3           ;SET MINIMUM LIMIT
1602  012270  004737  020272                JSR     PC,TTR          ;GO GET WRITE STALL
1603  012274  012704  023562                MOV     #MSG42,R4
1604  012300  004737  020536                JSR     PC,TTOUT        ;PRINT TURN AROUND STALL REQUEST
1605  012304  013703  000636                MOV     TSTAL,R3
1606  012310  004737  020724                JSR     PC,OCTP         ;PRINT TA STALL
1607  012314  012705  000636                MOV     #TSTAL,R5       ;SET TURN AROUND STALL ADDRESS
1608  012320  012701  000006                MOV     #6,R1           ;SET NUMBER OF CHARACTERS TO INPUT
1609  012324  012702  177777                MOV     #-1,R2          ;SET MAXIMUM LIMIT
1610  012330  012703  000001                MOV     #1,P3           ;SET MINIMUM LIMIT
1611  012334  004737  020272                JSR     PC,TTR          ;GO GET TURN AROUND STALL
1612  012340  000207                        RTS     PC              ;EXIT
1613
1614                                        ;UNIT DESCRIPTION POSITIONING SUBROUTINE**********
1615
1616  012342  000241                TPOS:   CLC
1617  012344  006137  000710                ROL     TEMP2           ;POSITION CHARACTER
1618  012350  005303                        DEC     R3              ;SEE IF DONE
1619  012352  001373                        BNE     TPOS            ;IF NOT: BR
1620  012354  013700  000734                MOV     UNP,R0          ;LOAD UNIT POINTER
1621  012360  053760  000710  001012        BIS     TEMP2,UN1(R0)   ;LOAD CHARACTER INTO UN1(R0)
1622  012366  000207                        RTS     PC              ;EXIT
1623
```

```
1624
1625                                    ;*********************************************************
1626                                    ;DATA SETUP ROUTINE:
1627                                    ;
1628                                    ;THIS ROUTINE IS USED TO GENERATE INTO THE ENTIRE
1629                                    ;WRITE BUFFER (4000 OCTAL CHARACTERS) THE DATA PATTERN
1630                                    ;SELECTED BY THE OPERATOR.  THESE ARE 20 (8) FIXED
1631                                    ;DATA PATTERNS AVAILABLE AND ONE SELECTION (DATA PATTERN 0)
1632                                    ;WHICH WILL READ ANY PATTERN PRESENTED AT THE
1633                                    ;HIGH SPEED PAPER TAPE READER.  THIS TAPE MUST BE PREPARED
1634                                    ;BY USING THE PROGRAM CALLED DTC.
1635                                    ;RANDOM DATA MAY ALSO BE USED VIA CONSOLE
1636                                    ;SWITCH EIGHT (8).
1637                                    ;THIS ROUTINE IS ALSO USED TO CLEAR OUT THE
1638                                    ;READ BUFFER (4000 OCTAL CHARACTERS) BEFORE EACH
1639                                    ;RECORD IS READ.
1640                                    ;*********************************************************
1641
1642  012370  005737  013304        DSUP:   TST     RDFL            ;SEE IF DID RANDOM DATA
1643  012374  001045                        BNE     DS1             ;IF SO: BR
1644  012376  005737  021632        DSO:    TST     ASEQF           ;SEE IF AUTO SEQ
1645  012402  001406                        BEQ     DSOA            ;IF NOT: BR
1646  012404  005737  000624                TST     PATRN           ;SEE IF AUTO RANDOM
1647  012410  100003                        BPL     DSOA            ;IF NOT: BR
1648  012412  004737  013236                JSR     PC,DATR         ;ELSE GO GENERATE RANDOM DATA
1649  012416  000207                        RTS     PC              ;RETURN
1650  012420  023737  000624  012750 DSOA:   CMP     PATRN,PATS      ;NEW PATTERN?
1651  012426  001014                        BNE     DSOC            ;IF SO: BR
1652  012430  013703  000616                MOV     UDES,R3         ;GET UNIT DESCRIPTION
1653  012434  042703  173777                BIC     #173777,R3      ;MASK PARITY
1654  012440  023703  012752                CMP     PARS,R3         ;SEE IF SAME AS LAST TIME
1655  012444  001404                        BEQ     DSOB            ;IF SO, BR
1656  012446  010337  012752                MOV     R3,PARS         ;SAVE PARITY
1657  012452  004737  013306                JSR     PC,CRCLRC       ;GO GENERATE EXPECTED CRC/LRC
1658  012456  000207                DSOB:   RTS     PC
1659  012460                         DSOC:
1660  012460  012703  026204                MOV     #WDATA,R3       ;R3 = ADDRS OF WRITE BUFFER
1661  012464  013701  000624                MOV     PATRN,R1        ;R1 = PATTERN SELECTOR
1662  012470  010137  012750                MOV     R1,PATS
1663  012474  062701  000(                  ADD     #1,R1           ;BUMP POINTER
1664  012500  000241                        CLC
1665  012502  006101                        ROL     R1              ;MAKE PATTERN SELECTOR EVEN
1666  012504  000171  002?                  JMP     @DATBL(R1)      ;GO GENERATE PATTERN
1667  012510  000240                DS1:    NOP
1668  012512  004737  013306                JSR     PC,CRCLRC
1669  012516  012702  002002        DS3:    MOV     #2002,R2        ;R2=BUFFER SIZE +2
1670  012522  012701  032216                MOV     #RDATA,R1       ;R1=READ DATA START
1671  012526  005021                DS4:    CLR     (R1)+           ;CLEAR BUFFER
1672  012530  005302                        DEC     R2              ;SEE IF DONE ALL
1673  012532  001375                        BNE     DS4             ;IF NOT: BR
1674  012534  013737  000616  012752        MOV     UDES,PARS       ;SET PARITY
1675  012542  042737  173777  012752        BIC     #173777,PARS
1676  012550  000207                        RTS     PC              ;EXIT
1677
1678
```

```
1679
1680                                    ;EXTERNAL DATA INPUT FROM H/S READER (256 CHARACTER MAXIMUM)
1681
1682  012552  005737  012746        DATO:   TST     DOFL            ;SEE IF SHOULD DO EXTERNAL INPUT
1683  012556  001354                        BNE     DS1             ;IF NOT: BR
1684  012560  012737  000001  012746        MOV     #1,DOFL         ;SET EXTERNAL FLAG
1685  012566  005077  166076                CLR     @PRB            ;CLEAR READER BUFFER
1686  012572  005077  166070                CLR     @PRS            ;CLEAR READER STATUS
1687  012576  005037  000706                CLR     TEMP1           ;CLEAR FOR USE AS CHARACTER FLAG
1688  012602  052777  000001  166056  DATOA: BIS     #1,@PRS         ;START READER
1689  012610  005037  000714                CLR     TEMP4
1690  012614  012704  000004                MOV     #4,R4           ;SET UP READER DONE DELAY
1691  012620  032777  000200  166040  DATOB: BIT     #200,@PRS       ;SEE IF DONE
1692  012626  001006                        BNE     1$              ;IF SO : BR
1693  012630  005337  000714                DEC     TEMP4
1694  012634  001371                        BNE     DATOB           ;DELAY FOR READER DONE
1695  012636  005304                        DEC     R4
1696  012640  001367                        BNE     DATOB           ;CONTINUE DELAY
1697  012642  000722                        BR      DS1             ;IF READER NEVER DONE: BR
1698  012644  005001                1$:     CLR     R1              ;CLEAR SAVE LOCATION
1699  012646  117701  166016                MOVB    @PRB,R1         ;SAVE CHARACTER
1700  012652  005737  000706                TST     TEMP1           ;SEE IF HAVE FOUND START CHARACTER
1701  012656  001012                        BNE     DATOC           ;IF SO : BR
1702  012660  105701                        TSTB    R1              ;SEE IF CHARACTER IS 0
1703  012662  001747                        BEQ     DATOA           ;IF SO : BR
1704  012664  012737  000001  000706        MOV     #1,TEMP1        ;ELSE SET CHARACTER FOUND FLAG
1705  012672  010137  000710                MOV     R1,TEMP2        ;SAVE DATA SIZE
1706  012676  010102                        MOV     R1,R2           ;SAVE DATA SIZE
1707  012700  000137  012602                JMP     DATOA           ;GO GET FIRST DATA CHAR
1708  012704  110123                DATOC:  MOVB    R1,(R3)+        ;LOAD BUFFER
1709  012706  005302                        DEC     R2              ;SEE IF READ ALL
1710  012710  001334                        BNE     DATOA           ;IF NOT : BR
1711  012712  012701  026204        DATOD:  MOV     #WDATA,R1       ;R1 = START OF WRITE BUFFER
1712  012716  013702  000710                MOV     TEMP2,R2        ;R2 = SIZE OF DATA FIELD
1713  012722  112123                DATOE:  MOVB    (R1)+,(R3)+     ;REPEAT LOAD OF DATA FIELD
1714  012724  022703  032216                CMP     #RDATA,R3       ;SEE IF DONE
1715  012730  003002                        BGT     DATOF           ;IF NOT: BR
1716  012732  000137  012510                JMP     DS1             ;EXIT
1717  012736  005302                DATOF:  DEC     R2              ;SEE IF AT END OF DATA FIELD
1718  012740  001370                        BNE     DATOE           ;IF NOT : BR
1719  012742  000137  012712                JMP     DATOD           ;ELSE RESTART FILL
1720  012746  000000                DOFL:   0                       ;EXTERNAL DATA FLAG=1 IF ALREADY DONE
1721  012750  177777                PATS:   -1
1722  012752  177777                PARS:   -1
1723
```

```
1724
1725                                        ;ALL ONES**********
1726
1727  012754  012701  177777      DAT1:     MOV     #-1,R1          ;R1=DATA
1728  012760  012702  002002      DAT1A:    MOV     #2002,R2        ;R2=WORD COUNT +2
1729  012764  010123              DAT1B:    MOV     R1,(R3)+        ;LOAD BUFFER
1730  012766  005302                        DEC     R2              ;SEE IF DONE
1731  012770  001375                        BNE     DAT1B           ;IF NOT: BR
1732  012772  000137  012510                JMP     DS1             ;RETURN
1733
1734                                        ;ALL ZEROS**********
1735
1736  012776  005001              DAT2:     CLR     R1              ;R1=DATA
1737  013000  000137  012760                JMP     DAT1A           ;LOAD BUFFER
1738
1739                                        ;WALKING ONE**********
1740
1741  013004  012701  000001      DAT3:     MOV     #1,R1           ;R1=DATA
1742  013010  000241                        CLC
1743  013012  012702  004004      DAT3A:    MOV     #4004,R2        ;R2=CHARACTER COUNT+4
1744  013016  110123              DAT3B:    MOVB    R1,(R3)+        ;LOAD BUFFER
1745  013020  106101                        ROLB    R1              ;SET NEXT CHARACTER
1746  013022  005302                        DEC     R2              ;SEE IF DONE
1747  013024  001374                        BNE     DAT3B           ;IF NOT: BR
1748  013026  000137  012510                JMP     DS1             ;RETURN
1749
1750                                        ;WALKING ZERO**********
1751
1752  013032  012701  000376      DAT4:     MOV     #376,R1         ;R1=START OF DATA
1753  013036  000261                        SEC
1754  013040  000137  013012                JMP     DAT3A           ;LOAD BUFFER
1755
1756                                        ;ALTERNATING ONE/ZERO**********
1757
1758
1759  013044  012701  052525      DAT5:     MOV     #52525,R1       ;R1=DATA
1760  013050  000137  012760                JMP     DAT1A           ;LOAD BUFFER
1761
1762                                        ;ALTERNATING ZERO/ONE**********
1763
1764  013054  012701  125252      DAT6:     MOV     #125252,R1      ;R1=DATA
1765  013060  000137  012760                JMP     DAT1A           ;LOAD BUFFER
1766
1767                                        ;ONE/ZERO IN ALTERNATING CHARACTERS**********
1768
1769  013064  012701  125125      DAT7:     MOV     #125125,R1      ;R1=DATA
1770  013070  000137  012760                JMP     DAT1A           ;LOAD BUFFER
1771
1772                                        ;ZERO/ONE IN ALTERNATING CHARACTERS**********
1773
1774  013074  012701  052652      DAT10:    MOV     #52652,R1       ;R1=DATA
1775  013100  000137  012760                JMP     DAT1A           ;LOAD BUFFER
1776
```

```
1777
1778                                        ;ALL BITS 0-377**********
1779
1780  013104  005001                DAT11:  CLR     R1              ;R1=STARTING DATA
1781  013106  012702  004004                MOV     #4004,R2        ;R2=CHARACTER COUNT+4
1782  013112  110123                DAT11A: MOVB    R1,(R3)+        ;LOAD BUFFER
1783  013114  105201                        INCB    R1              ;BUMP DATA
1784  013116  005302                        DEC     R2              ;SEE IF DONE
1785  013120  001374                        BNE     DAT11A          ;IF NOT: BR
1786  013122  000137  012510                JMP     DS1             ;RETURN
1787
1788                                        ;ALL BITS 377-0**********
1789
1790  013126  012701  000377        DAT12:  MOV     #377,R1         ;R1=STARTING DATA
1791  013132  012702  004004                MOV     #4004,R2        ;R2=CHARACTER COUNT+4
1792  013136  110123                DAT12A: MOVB    R1,(R3)+        ;LOAD BUFFER
1793  013140  105301                        DECB    R1              ;BUMP DATA
1794  013142  005302                        DEC     R2              ;SEE IF DONE
1795  013144  001374                        BNE     DAT12A          ;IF NOT: BR
1796  013146  000137  012510                JMP     DS1             ;RETURN
1797
1798                                        ;ALTERNATING CHARACTERS 0 AND 377**********
1799
1800  013152  012701  000377        DAT13:  MOV     #377,R1         ;R1 = DATA
1801  013156  000137  012760                JMP     DAT1A           ;LOAD BUFFER
1802
1803                                        ;ALTERNATING CHARACTERS 377 AND 0**********
1804
1805  013162  012701  177400        DAT14:  MOV     #177400,R1      ;R1 = DATA
1806  013166  000137  012760                JMP     DAT1A           ;LOAD BUFFER
1807
1808                                        ;WALKING ZERO REPEATED FOUR TIMES**********
1809
1810  013172  012702  002002        DAT15:  MOV     #2002,R2        ;SET NUMBER OF WORDS
1811  013176  012701  177376        DAT15R: MOV     #177376,R1      ;SET START OF DATA
1812  013202  012704  000002        DAT15A: MOV     #2,R4           ;SET NUMBER OF REPEATS
1813  013206  010123                DAT15B: MOV     R1,(R3)+        ;LOAD DATA
1814  013210  005302                        DEC     R2              ;SEE IF DONE
1815  013212  001002                        BNE     DAT15C          ;IF NOT: BR
1816  013214  000137  012510                JMP     DS1             ;RETURN
1817  013220  005304                DAT15C: DEC     R4              ;SEE IF DONE REPEATS
1818  013222  001371                        BNE     DAT15B          ;IF NOT: BR
1819  013224  000261                        SEC
1820  013226  006101                        ROL     R1              ;SET NEXT PATTERN
1821  013230  103764                        BCS     DAT15A          ;SEE IF SHOULD RESTART
1822  013232  000137  013176                JMP     DAT15R          ;IF SO: BR
1823
```

```
1824
1825                                        ;RANDOM DATA GENERATOR SUBROUTINE*********
1826
1827  013236  013704  000622          DATR: MOV     CARCNT,R4       ;SET SIZE OF RECORD
1828  013242  012703  026204                MOV     #WDATA,R3       ;SET ADDRESS OF START OF BUFFER
1829  013246  012701  177777                MOV     #-1,R1          ;SET HIGH LIMIT
1830  013252  005002                         CLR     R2              ;SET LOW LIMIT
1831  013254  004737  020240          DATRO: JSR     PC,RANG         ;GO GENERATE NUMBER
1832  013260  013723  000676                MOV     RANSAV,(R3)+    ;LOAD BUFFER
1833  013264  005204                         INC     R4              ;SEE IF DONE ALL
1834  013266  001372                         BNE     DATRO           ;IF NOT: BR
1835  013270  004737  012510                JSR     PC,DS1          ;GO CHECK FOR 7 CH
1836  013274  012737  000001  013304        MOV     #1,RDFL         ;SET RANDOM DATA FLAG
1837  013302  000207                         RTS     PC              ;EXIT
1838  013304  000000          RDFL:  0                               ;RANDOM DATA SELECT FLAG
```

```
1839
1840                                    ;*************************************************************
1841                                    ;CRC/LRC CHARACTER BUILD;
1842                                    ;
1843                                    ;THIS ROUTINE WILL CONSTRUCT AND SAVE THE EXPECTED
1844                                    ;CRC AND LRC CHARACTERS ACCORDING TO DATA AND
1845                                    ;RECORD SIZE IF OPERATING IN NRZ MODE
1846                                    ;*************************************************************
1847
1848  013306  000240            CRCLRC: NOP
1849  013310  013700  000622    CRLR:   MOV     CARCNT,R0       ;SET RECORD SIZE
1850  013314  005400                    NEG     R0
1851  013316  012701  026204            MOV     #WDATA,R1       ;SET START OF BUFFER
1852  013322  005037  013674            CLR     XORS
1853  013326  111104            CLO:    MOVB    (R1),R4         ;GET CHARACTER
1854  013330  004737  013522            JSR     PC,CLP          ;GO GET PARITY OF CHARACTER
1855  013334  004737  013650            JSR     PC,XOR          ;XOR CHARACTER
1856  013340  000241                    CLC
1857  013342  006004                    ROR     R4              ;ROTATE 1 RIGHT
1858  013344  103014                    BCC     CL2             ;IF NO CARRY: BR
1859  013346  052704  000400            BIS     #400,R4         ;SET BIT NINE
1860  013352  000241                    CLC
1861  013354  010405            CL1:    MOV     R4,R5           ;SAVE CHARACTER
1862  013356  042705  177703            BIC     #177703,R5
1863  013362  005105                    COM     R5
1864  013364  042705  177703            BIC     #177703,R5
1865  013370  042704  000074            BIC     #74,R4
1866  013374  050504                    BIS     R5,R4           ;COMPLEMENT BITS 2,3,4,5
1867  013376  010437  013674    CL2:    MOV     R4,XORS
1868  013402  005300                    DEC     R0
1869  013404  001402                    BEQ     CLLAST          ;IF LAST CHARACTER: BR
1870  013406  000137  013326            JMP     CLO             ;GET NEXT
1871  013412  013704  013674    CLLAST: MOV     XORS,R4
1872  013416  005137  013674            COM     XORS
1873  013422  042737  177050  013674    BIC     #177050,XORS
1874  013430  042704  177727            BIC     #177727,R4      ;COMPLEMENT ALL BUT BITS 3&5
1875  013434  050437  013674            BIS     R4,XORS
1876  013440  013737  013674  013676    MOV     XORS,EXCRC      ;SAVE EXPECTED CRC
1877  013446  013700  000622            MOV     CARCNT,R0
1878  013452  005400                    NEG     R0
1879  013454  012701  026204            MOV     #WDATA,R1       ;DO EXPT LRC
1880  013460  005037  013674            CLR     XORS
1881  013464  111104            CL3:    MOVB    (R1),R4
1882  013466  004737  013522            JSR     PC,CLP          ;GET PARITY
1883  013472  004737  013650            JSR     PC,XOR          ;XOR CHARACTER
1884  013476  005300                    DEC     R0
1885  013500  001371                    BNE     CL3             ;DO ALL FOR LRC
1886  013502  013704  013676            MOV     EXCRC,R4
1887  013506  004737  013650            JSR     PC,XOR          ;XOR CRC TO DATA
1888  013512  013737  013674  013700    MOV     XORS,EXLRC      ;SAVE EXPT LRC
1889  013520  000207                    RTS     PC              ;RETURN
1890  013522  005704            CLP:    TST     R4              ;SEE IF 0 CHAR
1891  013524  001010                    BNE     CLPE            ;IF NOT: BR
1892  013526  032737  004000  000616    BIT     #4000,UDES      ;SEE IF EVEN PARITY
1893  013534  001404                    BEQ     CLPE            ;IF NOT: BR
1894  013536  012704  000420            MOV     #420,R4         ;SET 0 CHAR EVEN PARITY
```

```
1895  013542  005201                        INC     R1              ;BUMP POINTER
1896  013544  000207                        RTS     PC              ;RETURN
1897  013546  005037  013710        CLPE:   CLR     PARCNT          ;CLEAR BIT COUNTER
1898  013552  012703  000010                MOV     #10,R3          ;SET NUMBER OF BITS
1899  013556  032704  000001        CLPO:   BIT     #1,R4           ;SEE IF ONE BIT
1900  013562  001402                        BEQ     CLP1            ;IF NOT: BR
1901  013564  005237  013710                INC     PARCNT          ;BUMP COUNTER
1902  013570  000241                CLP1:   CLC
1903  013572  006004                        ROR     R4              ;ROTATE TO NEXT BIT
1904  013574  005303                        DEC     R3
1905  013576  001367                        BNE     CLPO            ;CONTINUE FOR ALL BITS
1906  013600  112104                        MOVB    (R1)+,R4
1907  013602  042704  177400                BIC     #177400,R4
1908  013606  032737  000001  013710        BIT     #1,PARCNT       ;SEE IF ODD NUMBER OF ONE BITS
1909  013614  001005                        BNE     CLP2            ;IF SO: BR
1910  013616  032737  004000  000616        BIT     #4000,UDES      ;SEE IF SHOULD BE EVEN PARITY
1911  013624  001406                        BEQ     CLP3            ;IF NOT: BR
1912  013626  000207                        RTS     PC              ;ELSE EXIT
1913  013630  032737  004000  000616  CLP2:   BIT     #4000,UDES      ;SEE IF SHOULD BE ODD PARITY
1914  013636  001001                        BNE     CLP3            ;IF NOT: BR
1915  013640  000207                        RTS     PC              ;ELSE EXIT
1916  013642  052704  000400        CLP3:   BIS     #400,R4         ;SET PARITY BIT
1917  013646  000207                        RTS     PC
1918  013650  010446                XOR:    MOV     R4,-(SP)
1919  013652  043716  013674                BIC     XORS,(SP)
1920  013656  040437  013674                BIC     R4,XORS         ;XOR SUBROUTINE: R4 WITH XORS
1921  013662  052637  013674                BIS     (SP)+,XORS
1922  013666  013704  013674                MOV     XORS,R4
1923  013672  000207                        RTS     PC
1924
1925  013674  000000                XORS:   0                       ;XOR SAVE
1926  013676  000000                EXCRC:  0                       ;EXPECTED CRC
1927  013700  000000                EXLRC:  0                       ;EXPECTED LRC
1928  013702  000000                ACTLRC: 0                       ;ACTUAL LRC
1929  013704  000000                LRCSV:  0                       ;LRC SAVE
1930  013706  000000                LRCPT:  0                       ;CRC PRINT FLAG
1931  013710  000000                PARCNT: 0                       ;PARITY COUNTER
1932
```

```
1933
1934                                    ;*********************************************************
1935                                    ;DATA CHECK SUBROUTINE:
1936                                    ;
1937                                    ;THIS SUBROUTINE IS USED TO COMPARE EACH CHARACTER
1938                                    ;OF DATA READ FROM TAPE WITH THE EXPECTED CHARACTER.
1939                                    ;ANY ERROR DETECTED WILL CAUSE CONTROL TO BE
1940                                    ;PASSED TO AN ERROR PRINT SUBROUTINE AND A
1941                                    ;SUBROUTINE TO ACCUMULATE THE NUMBER OF BITS
1942                                    ;DROPPED AND PICKED UP FROM EACH CHARACTER.
1943                                    ;DATA CHECKING MAY BE TERMINATED BY USE OF
1944                                    ;CONSOLE SWITCH THIRTEEN (13).
1945                                    ;*********************************************************
1946
1947  013712  005037  000722       DCHK:   CLR     BBC             ;CLEAR BAD RECORD CNTR
1948  013716  005037  000744               CLR     DERFL           ;CLEAR DATA ERROR FLAG
1949  013722  005037  000726               CLR     HDRFL           ;CLEAR HEADER FLAG
1950  013726  013705  000622               MOV     CARCNT,R5       ;LOAD CHAR COUNT
1951  013732  012701  026204               MOV     #WDATA,R1       ;SET WRITE DATA ADDR
1952  013736  012702  032216               MOV     #RDATA,R2       ;SET READ DATA ADDR
1953  013742  032737  004000  000616       BIT     #4000,UDES      ;SEE IF EVEN PARITY
1954  013750  001435                       BEQ     DF0             ;IF NOT: BR
1955  013752  005737  001010               TST     STCDFL          ;SEE IF 7 TRK CORE DUMP
1956  013756  001032                       BNE     DF0             ;IF SO: BR
1957  013760  012703  000377               MOV     #377,R3
1958  013764  042703  177400               BIC     #177400,R3      ;BACKGROUND DATA MASK
1959  013770  032777  000020  164602       BIT     #20,@MTS        ;SEE IF 7 TRK DRIVE(NORMAL)
1960  013776  001402                       BEQ     DFA             ;IF NOT: BR
1961  014000  042703  000300               BIC     #300,R3         ;MASK FOR 7 TRK NORMAL DATA
1962  014004  130311               DFA:    BITB    R3,(R1)         ;SEE IF ZERO CHARACTER
1963  014006  001404                       BEQ     DFC
1964  014010  005201                       INC     R1              ;BUMP POINTER
1965  014012  005205               DFB:    INC     R5              ;SEE IF DONE
1966  014014  001373                       BNE     DFA             ;IF NOT: BR
1967  014016  000406                       BR      DFD
1968  014020  112721  000020       DFC:    MOVB    #20,(R1)+       ;REPLACE 0 WITH 20
1969  014024  012737  177777  012750       MOV     #-1,PATS        ;SET TO GENERATE NEW PATTERN
1970  014032  000767                       BR      DFB
1971  014034  013705  000622       DFD:    MOV     CARCNT,R5       ;RESET COUNT
1972  014040  012701  026204               MOV     #WDATA,R1       ;RESET ADDRESS
1973  014044  032777  000020  164526 DF0:  BIT     #20,@MTS        ;SEE IF 7 TRACK
1974  014052  001403                       BEQ     DF9             ;IF NOT: BR
1975  014054  005737  001010               TST     STCDFL          ;SEE IF 7 TRK CORE DUMP
1976  014060  001417                       BEQ     DF7             ;IF NOT: BR
1977  014062  122122               DF9:    CMPB    (R1)+,(R2)+     ;SEE IF DATA IS GOOD
1978  014064  001003                       BNE     DF91            ;IF NOT: BR
1979  014066  105037  000722               CLRB    BBC             ;ELSE CLEAR BAD RECORD COUNTER
1980  014072  000407                       BR      DF92
1981  014074  004737  014656       DF91:   JSR     PC,DRPKF        ;GO DO DROPS AND PICKS
1982  014100  004737  014242               JSR     PC,DERR         ;GO PRINT ERROR
1983  014104  012737  000001  000744       MOV     #1,DERFL        ;SET DATA ERROR FLAG
1984  014112  005205               DF92:   INC     R5              ;SEE IF DONE ALL CHARACTERS
1985  014114  001362                       BNE     DF9             ;IF NOT: DO ALL
1986  014116  000432                       BR      DF3
1987  014120  000240               DF7:    NOP
1988  014122  010137  014240               MOV     R1,STAS         ;SAVE CHARACTER ADDRESS
```

```
 1989  014126  117737  000106  014236          MOVB    @STAS,STCS      ;SAVE CHARACTER
 1990  014134  142711  000300                  BICB    #300,(R1)       ;MASK FOR 7 TRACK DRIVE
 1991  014140  122122                          CMPB    (R1)+,(R2)+     ;SEE IF DATA IS GOOD
 1992  014142  001003                          BNE     DF71            ;IF NOT: BR
 1993  014144  105037  000722                  CLRB    BBC             ;CLEAR BAD RECORD COUNTER
 1994  014150  000407                          BR      DF72
 1995  014152  004737  014656          DF71:   JSR     PC,DRPKF        ;GO DO DROPS AND PICKS
 1996  014156  004737  014242                  JSR     PC,DERR         ;GO PRINT ERROR
 1997  014162  012737  000001  000744          MOV     #1,DERFL        ;SET DATA ERROR FLAG
 1998  014170  000240                  DF72:   NOP
 1999  014172  153777  014236  000040          BISB    STCS,@STAS      ;RESET DATA
 2000  014200  005205                          INC     R5              ;SEE IF DONE ALL
 2001  014202  001346                          BNE     DF7             ;IF NOT: DO ALL
 2002  014204  005737  000744          DF3:    TST     DERFL           ;SEE IF HAD DATA ERROR
 2003  014210  001411                          BEQ     DFX             ;IF NOT: BR
 2004  014212  005737  000742                  TST     SERFL
 2005  014216  001006                          BNE     DFX             ;IF NOT DATA ERROR ONLY: BR
 2006  014220  013704  000734                  MOV     UNP,R4
 2007  014224  005264  001134                  INC     DATER1(R4)      ;BUMP DATA ERROR COUNTER
 2008  014230  004737  022126                  JSR     PC,CKSWR        ;CHECK FOR  G
 2009  014234  000207                  DFX:    RTS     PC              ;EXIT
 2010  014236  000000                  STCS:   0                       ;7 TRACK DATA SAVE
 2011  014240  000000                  STAS:   0                       ;7 TRACK ADDRESS SAVE
```

```
2012
2013                                    ;*******************************************************
2014                                    ;DATA ERROR SUBROUTINE:
2015                                    ;
2016                                    ;THIS SUBROUTINE IS USED TO PRINT OUT ANY
2017                                    ;ERRORS FOUND DURING THE DATA CHECK.
2018                                    ;EACH CHARACTER FOUND BAD WILL BE PRINTED
2019                                    ;IN BIT FORMAT ALONG WITH ITS EXPECTED CHARACTER.
2020                                    ;AN ERROR HEADER CONSISTING OF THE UNIT NUMBER,
2021                                    ;BLOCK NUMBER, RECORD NUMBER, SIZE OF RECORD, AND
2022                                    ;ERROR TYPE (READ FORWARD, WRITE, ETC)
2023                                    ;IS PRINTED ONLY ONCE FOR EACH RECORD FOUND BAD:
2024                                    ;A COUNT IS MADE OF THE NUMBER OF SUCCESSIVE BAD
2025                                    ;CHARACTERS, AND IF TEN (10) SUCCESSIVE BAD CHARACTERS
2026                                    ;ARE FOUND IN A SINGLE RECORD, A MESSAGE INDICATING
2027                                    ;A BAD RECORD CONDITION IS PRINTED AND THE NEXT
2028                                    ;TWENTY (20) CHARACTERS ARE SKIPPED BEFORE CHECKING
2029                                    ;IS RESUMED.  IF THE BAD RECORD CONDITION IS FOUND
2030                                    ;THREE TIMES IN A RECORD, ALL REMAINING DATA IS
2031                                    ;SKIPPED EXCEPT THE FINAL TEN (10) CHARACTERS.
2032                                    ;THIS SKIPPING IS OF COURSE ONLY POSSIBLE IN
2033                                    ;RECORDS WHICH CONTAIN A SUFFICIENT NUMBER OF CHARACTERS.
2034                                    ;PRINTING OF ERRORS MAY BE DISALLOWED AT ANY TIME
2035                                    ;BY SETTING CONSOLE SWITCH TEN (10) TO A ONE.
2036                                    ;THE OPERATOR MAY CAUSE THE PROGRAM TO HALT ON ANY ERROR
2037                                    ;BY SETTING CONSOLE SWITCH FIFTEEN (15) TO A ONE.
2038                                    ;*******************************************************
2039
2040  014242  032777  002000  164402  DERR:   BIT     #2000,@SWR      ;SEE IF SHOULD PRINT ERRORS
2041  014250  001402                          BEQ     DERR0           ;IF SO: BR
2042  014252  000137  014400                  JMP     DERR4           ;ELSE SKIP PRINT
2043  014256  005237  000732          DERR0:  INC     PFLG            ;SET PRINT FLAG
2044  014262  005737  000726                  TST     HDRFL           ;SEE IF HAVE PRINTED HEADER
2045  014266  001013                          BNE     DERR0A          ;IF SO: BR
2046  014270  005737  000742                  TST     SERFL           ;ALREADY PRINTED HEADER?
2047  014274  001010                          BNE     DERR0A          ;IF SO:  BR
2048  014276  004737  017506                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
2049  014302  012704  022434                  MOV     #MSG1,R4        ;LOAD ERROR MSG ADDR
2050  014306  004737  020536                  JSR     PC,TTOUT        ;PRINT ERROR
2051  014312  004737  016742                  JSR     PC,FRPRT        ;PRINT F OR R
2052  014316  012704  022453          DERR0A: MOV     #MSG4,R4
2053  014322  004737  020536                  JSR     PC,TTOUT        ;PRINT CHAR NO.  HEADER
2054  014326  010203                          MOV     R2,R3
2055  014330  162703  032216                  SUB     #RDATA,R3       ;POINT TO CHAR
2056  014334  005303                          DEC     R3
2057  014336  004737  020724          DERR0B: JSR     PC,OCTP         ;PRINT CHAR NUMBER
2058  014342  012704  022441                  MOV     #MSG2,R4
2059  014346  004737  020536                  JSR     PC,TTOUT        ;PRINT EXPECTED DATA
2060  014352  114103                          MOVB    -(R1),R3        ;LOAD EXPECTED DATA
2061  014354  004737  021152                  JSR     PC,DOUT         ;GO PRINT CHAR
2062  014360  012704  022446                  MOV     #MSG3,R4
2063  014364  004737  020536                  JSR     PC,TTOUT        ;PRINT RECEIVED DATA
2064  014370  114203          DERR1:  MOVB    -(R2),R3
2065  014372  004737  021152  DERR2:  JSR     PC,DOUT         ;PRINT BAD CHAR
2066  014376  122122          DERR3:  CMPB    (R1)+,(R2)+     ;RESET POINTERS
2067  014400  105237  000722  DERR4:  INCB    BBC             ;BUMP BAD RECORD CNTR
```

```
2068  014404  122737  000010  000722          CMPB    #10,BBC         ;SEE IF BLD BTH
2069  014412  001076                          BNE     DEREX           ;IF NOT: BR
2070  014414  032777  002000  164230          BIT     #2000,@SWR      ;SEE IF PRINT INHIBIT
2071  014422  001004                          BNE     1$              ;IF SO: BR
2072  014424  012704  022534                  MOV     #MSG15,R4
2073  014430  004737  020536                  JSR     PC,TTOUT        ;PRINT BLD BTH
2074  014434  105037  000722          1$:     CLRB    BBC             ;RESET BAD RECORD CNTR
2075  014440  000337  000722                  SWAB    BBC             ;POSITION BLD BTH AMOUNT
2076  014444  105237  000722                  INCB    BBC             ;BUMP AMOUNT
2077  014450  122737  000003  000722          CMPB    #3,BBC          ;SEE IF HAD 3 BLD BTHS
2078  014456  101037                          BHI     DERR4B          ;IF NOT: BR
2079  014460  000337  000722                  SWAB    BBC             ;REPOSITION BBC
2080  014464  022705  177767                  CMP     #177767,R5      ;SEE IF ON LAST EIGHT CHARS
2081  014470  101445                          BLOS    DERR6           ;IF SO: BR
2082  014472  012705  177767                  MOV     #177767,R5      ;SET CHAR CNTR TO 8
2083  014476  013737  000622  000706          MOV     CARCNT,TEMP1    ;LOAD CHAR COUNT
2084  014504  005137  000706                  COM     TEMP1
2085  014510  005237  000706                  INC     TEMP1
2086  014514  162737  000010  000706          SUB     #10,TEMP1       ;POINT TO BUFFER -8
2087  014522  013701  000706                  MOV     TEMP1,R1        ;POINT TO NEXT CHAR
2088  014526  062701  026204                  ADD     #WDATA,R1       ;POINT TO NEXT WRITE CHAR
2089  014532  013702  000706                  MOV     TEMP1,R2        ;POINT TO END OF READ DATA -8 FORWARD
2090  014536  062702  032216                  ADD     #RDATA,R2       ;POINT TO NEXT CHAR
2091  014542  000422                          BR      DEREX           ;EXIT
2092  014544  012702  000010          DERR4A: MOV     #10,R2          ;POINT TO THE END OF READ DATA -8 REVERSE
2093  014550  062702  032216                  ADD     #RDATA,R2       ;POINT TO THE NEXT CHAR
2094  014554  000415                          BR      DEREX           ;EXIT
2095  014556  000337  000722          DERR4B: SWAB    BBC             ;REPOSITION BBC
2096  014562  000241                          CLC
2097  014564  062705  000024                  ADD     #24,R5          ;SKIP 20 CHARS
2098  014570  103405                          BCS     DERR6           ;IF EXCEED RECORD SIZE: BR
2099  014572  062701  000024                  ADD     #24,R1          ;SKIP 20 CHARS
2100  014576  062702  000024          DERR5:  ADD     #24,R2          ;SKIP FORWARD 20 CHARS
2101  014602  000402                          BR      DEREX
2102  014604  012705  177777          DERR6:  MOV     #-1,R5          ;SET TO EOR
2103  014610  032777  100000  164034  DEREX:  BIT     #100000,@SWR    ;SEE IF SHOULD HALT ON ERROR
2104  014616  001412                          BEQ     DEREX1          ;IF NOT: BR
2105  014620  000000                          HALT
2106  014622  005737  000732                  TST     PFLG            ;SEE IF PRINTED
2107  014626  001006                          BNE     DEREX1          ;IF SO: BR
2108  014630  032777  002000  164014          BIT     #2000,@SWR      ;SEE IF SHOULD PRINT
2109  014636  001002                          BNE     DEREX1          ;IF NOT: BR
2110  014640  000137  014256                  JMP     DERR0           ;ELSE PRINT
2111  014644  004737  022126          DEREX1: JSR     PC,CKSWR        ;TEST FOR  G
2112  014650  005037  000732                  CLR     PFLG            ;CLEAR FLAG
2113  014654  000207                          RTS     PC              ;RETURN
2114
```

```
2115
2116                                    ;*************************************************************
2117                                    ;DROPS AND PICKS SUBROUTINE:
2118                                    ;
2119                                    ;THIS SUBROUTINE IS USED TO ACCUMULATE FROM
2120                                    ;EACH BAD DATA CHARACTER FOUND THE NUMBER
2121                                    ;OF BITS WHICH WERE EITHER DROPPED OR PICKED UP.
2122                                    ;TWO COUNTERS  ARE USED TO ACCUMULATE THIS
2123                                    ;INFORMATION AND CAN STORE UP TO 32K DROPS
2124                                    ;OR PICKS BEFORE OVERFLOWING.  IF OVERFLOW IS
2125                                    ;ABOUT TO OCCUR, THESE ACCUMULATORS ARE
2126                                    ;PRINTED IN OCTAL AND RESET TO ZERO.
2127                                    ;THE CONTENTS OF THE ACCUMULATORS MAY BE
2128                                    ;DISPLAYED AT ANY TIME BY SETTING CONSOLE
2129                                    ;SWITCH FOURTEEN TO A ONE (1).  THE PRINTOUT WILL OCCUR
2130                                    ;AT THE END OF THE CURRENT BLOCK CYCLE.
2131                                    ;*************************************************************
2132
2133  014656  005037  000706        DRPKF:  CLR     TEMP1
2134  014662  005037  000710                CLR     TEMP2
2135  014666  005037  000712                CLR     TEMP3
2136  014672  013704  000734                MOV     UNP,R4
2137  014676  016437  001034  000770        MOV     PIK1(R4),BPKP
2138  014704  016437  001054  000766        MOV     DRP1(R4),BDPP
2139  014712  124142                        CMPB    -(R1),-(R2)     ;POINT TO CHAR
2140  014714  112137  000706                MOVB    (R1)+,TEMP1     ;LOAD GOOD CHAR
2141  014720  112237  000710                MOVB    (R2)+,TEMP2     ;LOAD BAD CHAR
2142  014724  004737  014736        DRPK:   JSR     PC,DROP         ;GET DROPS
2143  014730  004737  015156                JSR     PC,PICK         ;GET PICKS
2144  014734  000207                        RTS     PC              ;EXIT
2145  014736  113703  000706        DROP:   MOVB    TEMP1,R3        ;R3 = GOOD CHAR
2146  014742  113704  000710                MOVB    TEMP2,R4        ;R4 = BAD CHAR
2147  014746  140403                DPC:    BICB    R4,R3           ;GET DROPS/PICKS
2148  014750  001001                        BNE     DPCG            ;IF SOME: BR
2149  014752  000207                        RTS     PC              ;RETURN
2150  014754  012737  000010  000736 DPCG:  MOV     #10,BCNT        ;SET NUMBER TO CHECK
2151  014762  132703  000001        DPC0:   BITB    #1,R3           ;SEE IF DROPPED OR PICKED THIS BIT
2152  014766  001455                        BEQ     DPC2            ;IF NOT: BR
2153  014770  105737  000712                TSTB    TEMP3           ;SEE IF ON PICKS
2154  014774  001016                        BNE     DPC1            ;IF SO: BR
2155  014776  005277  163764                INC     @BDPP           ;BUMP DROP CNTR
2156  015002  005777  163760                TST     @BDPP
2157  015006  100045                        BPL     DPC2            ;IF NO OVERFLOW: BR
2158  015010  032777  002000  163634        BIT     #2000,@SWR      ;SEE IF HAVE PRINTED DATA
2159  015016  001402                        BEQ     DPC0A           ;IF SO: BR
2160  015020  004737  017506                JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
2161  015024  004737  015222        DPC0A:  JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
2162  015030  000415                        BR      DPC2A
2163  015032  005277  163732        DPC1:   INC     @BPKP           ;BUMP PICK CNTR
2164  015036  005777  163726                TST     @BPKP           ;SEE IF OVERFLOW
2165  015042  100027                        BPL     DPC2            ;IF NOT: BR
2166  015044  032777  002000  163600        BIT     #2000,@SWR      ;SEE IF HAVE PRINTED DATA
2167  015052  001402                        BEQ     DPC1A           ;IF SO: BR
2168  015054  004737  017506                JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
2169  015060  004737  015222        DPC1A:  JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
2170  015064  013704  000734        DPC2A:  MOV     UNP,R4
```

```
2171  015070  016403  001054                MOV     DRP1(R4),R3     ;SET DROP POINTER
2172  015074  016404  001034                MOV     PIK1(R4),R4     ;SET PICK POINTER
2173  015100  012737  000010  000736        MOV     #10,BCNT        ;SET NUMBER OF BITS
2174  015106  005023                DPC2B:   CLR     (R3)+           ;CLEAR DROPS
2175  015110  005024                         CLR     (R4)+           ;CLEAR PICK
2176  015112  005337  000736                DEC     BCNT            ;SEE IF DONE
2177  015116  001373                        BNE     DPC2B           ;IF NOT: BR
2178  015120  000207                        RTS     PC              ;EXIT
2179  015122  000241                DPC2:    CLC
2180  015124  106003                        RORB    R3              ;GET NEXT BIT
2181  015126  005337  000736                DEC     BCNT            ;SEE IF DONE
2182  015132  001410                        BEQ     DPC3
2183  015134  062737  000002  000770        ADD     #2,BPKP
2184  015142  062737  000002  000766        ADD     #2,BDPP
2185  015150  000137  014762                JMP     DPC0            ;CONTINUE
2186  015154  000207                DPC3:    RTS     PC              ;RETURN
2187  015156  013704  000734                PICK:    MOV     UNP,R4          ;SET UNIT POINTER
2188  015162  016437  001034  000770        MOV     PIK1(R4),BPKP   ;SET PICK POINTER
2189  015170  016437  001054  000766        MOV     DRP1(R4),BDPP   ;SET DROP POINTER
2190  015176  113704  000706                MOVB    TEMP1,R4        ;R4 = GOOD CHAR
2191  015202  113703  000710                MOVB    TEMP2,R3        ;R3 = BAD CHAR
2192  015206  112737  000001  000712        MOVB    #1,TEMP3        ;SET PICK FLAG
2193  015214  004737  014746                JSR     PC,DPC          ;GO CHECK PICKS
2194  015220  000207                        RTS     PC              ;EXIT
2195  015222  012704  023121                DPPRT:   MOV     #MSG26,R4
2196  015226  004737  020536                JSR     PC,TTOUT        ;PRINT DROP HEADER
2197  015232  013704  000734                MOV     UNP,R4
2198  015236  016437  001054  000766        MOV     DRP1(R4),BDPP   ;SET DROP POINTER
2199  015244  016437  001034  000770        MOV     PIK1(R4),BPKP   ;SET PICK POINTER
2200  015252  062737  000016  000766        ADD     #16,BDPP
2201  015260  062737  000016  000770        ADD     #16,BPKP
2202  015266  012737  000010  000736        MOV     #10,BCNT        ;SET NUMBER TO PRINT
2203  015274  017703  163466        DPPRT0:  MOV     @BDPP,R3
2204  015300  004737  020724                JSR     PC,OCTP         ;PRINT DROPS
2205  015304  005337  000736                DEC     BCNT            ;SEE IF DONE
2206  015310  001404                        BEQ     DPPRT1          ;IF NOT: BR
2207  015312  162737  000002  000766        SUB     #2,BDPP         ;BUMP POINTER
2208  015320  000765                        BR      DPPRT0          ;CONTINUE FOR ALL 8 BITS
2209  015322  012737  000010  000736  DPPRT1:  MOV     #10,BCNT        ;SET NUMBER TO PRINT
2210  015330  012704  023132                MOV     #MSG27,R4
2211  015334  004737  020536                JSR     PC,TTOUT        ;PRINT PICK HEADER
2212  015340  017703  163424        DPPRT2:  MOV     @BPKP,R3
2213  015344  004737  020724                JSR     PC,OCTP         ;PRINT PICKS
2214  015350  005337  000736                DEC     BCNT            ;SEE IF DONE
2215  015354  001404                        BEQ     DPPRTX          ;IF SO: BR
2216  015356  162737  000002  000770        SUB     #2,BPKP         ;BUMP POINTER
2217  015364  000765                        BR      DPPRT2          ;CONTINUE FOR ALL 8 BITS
2218  015366  000207                DPPRTX:  RTS     PC              ;RETURN
```

```
2219                                    ;*********************************************************
2220                                    ;STATICSTICS PRINT
2221                                    ;
2222                                    ;THIS SUBROUTINE PRINTS THE ACCUMULATED
2223                                    ;ERROR STATISTICS FOR EACH DRIVE.
2224                                    ;THE ROUTINE CAN BE CALLED TO PRINT
2225                                    ;AT THE END OF EACH BLOCK BY SELECTING
2226                                    ;SW14=1.  THE SUMMARY IS AUTOMATICALLY
2227                                    ;PRINTED FOR A DRIVE WHENEVER A TAPE
2228                                    ;IS REWOUND FROM EOT OR DROPPED.
2229                                    ;*********************************************************
2230
2231  015370  012700  000001   PRSTAT: MOV     #1,R0           ;SET RECORD COUNTER TO 1
2232  015374  004737  017506           JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
2233  015400  004737  015222   PRSTA2: JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
2234  015404  012704  024501           MOV     #MSG64,R4
2235  015410  004737  020536           JSR     PC,TTOUT        ;PRINT WRITE ERROR TAG
2236  015414  013704  000734           MOV     UNP,R4
2237  015420  016403  001074           MOV     WTER1(R4),R3
2238  015424  004737  020724           JSR     PC,OCTP         ;PRINT WRITE ERRORS
2239  015430  012704  024726           MOV     #MSG76,R4
2240  015434  004737  020536           JSR     PC,TTOUT        ;PRINT RETRY TOTAL
2241  015440  013704  000734           MOV     UNP,R4
2242  015444  016403  001154           MOV     RTY1(R4),R3
2243  015450  004737  020724           JSR     PC,OCTP         ;PRINT RETRIES
2244  015454  012704  024512           MOV     #MSG65,R4
2245  015460  004737  020536           JSR     PC,TTOUT        ;PRINT READ ERROR TAG
2246  015464  013704  000734           MOV     UNP,R4
2247  015470  016403  001114           MOV     RDER1(R4),R3
2248  015474  004737  020724           JSR     PC,OCTP         ;PRINT READ ERRORS
2249  015500  012704  024704           MOV     #MSG74,R4
2250  015504  004737  020536           JSR     PC,TTOUT        ;PRINT SOFT ERROR MESSAGE
2251  015510  013704  000734           MOV     UNP,R4
2252  015514  016403  001174           MOV     GDRTY1(R4),R3
2253  015520  004737  020724           JSR     PC,OCTP         ;PRINT SOFT ERROR NUMBER
2254  015524  012704  024715           MOV     #MSG75,R4
2255  015530  004737  020536           JSR     PC,TTOUT        ;PRINT HARD RD ERROR MESSG
2256  015534  013704  000734           MOV     UNP,R4
2257  015540  016403  001214           MOV     BDRTY1(R4),R3
2258  015544  004737  020724           JSR     PC,OCTP         ;PRINT HARD RD ERROR COUNT
2259  015550  012704  024523           MOV     #MSG66,R4
2260  015554  004737  020536           JSR     PC,TTOUT        ;PRINT DATA ERROR TAG
2261  015560  013704  000734           MOV     UNP,R4
2262  015564  016403  001134           MOV     DATER1(R4),R3
2263  015570  004737  020724           JSR     PC,OCTP         ;PRINT DATA ERROR NUMBER
2264  015574  004737  015606           JSR     PC,BTPRT        ;PRINT BAD TAPE STATS
2265  015600  004737  022126           JSR     PC,CKSWR        ;CHECK FOR  G
2266  015604  000207                   RTS     PC              ;RETURN
2267
```

```
2268
2269                                    ;BAD TAPE STATISTICS PRINT**********
2270
2271  015606  005037  000754    BTPRT:  CLR     RTYFL
2272  015612  012704  024534            MOV     #MSG67,R4
2273  015616  004737  020536            JSR     PC,TTOUT        ;DO CR/LF
2274  015622  013704  000734            MOV     UNP,R4
2275  015626  016437  002714  000762    MOV     BTADDR(R4),BTPT ;SET TABLE POINTER
2276  015634  017703  163122            MOV     @BTPT,R3
2277  015640  000241                    CLC
2278  015642  006003                    ROR     R3              ;CORRECT NUMBER
2279  015644  004737  020724            JSR     PC,OCTP         ;PRINT NUMBER OF BAD SPOTS
2280  015650  012704  024536            MOV     #MSG68,R4
2281  015654  004737  020536            JSR     PC,TTOUT        ;PRINT BAD TAPE TAG
2282  015660  005777  163076            TST     @BTPT           ;SEE IF ANY BAD SPOTS
2283  015664  001001                    BNE     BTOV0           ;IF SO: BR
2284  015666  000207                    RTS     PC
2285  015670  013701  000762    BTOV0:  MOV     BTPT,R1         ;SET TABLE POINTER
2286  015674  005721                    TST     (R1)+
2287  015676  005000                    CLR     R0
2288  015700  010003    BTOV1:  MOV     R0,R3
2289  015702  000241                    CLC
2290  015704  006003                    ROR     R3              ;R3=R3/2 FOR CORRECT NUMBER
2291  015706  004737  020724            JSR     PC,OCTP         ;PRINT ENTRY NUMBER
2292  015712  012704  022521            MOV     #MSG13,R4
2293  015716  105724                    TSTB    (R4)+           ;SKIP CR/LF
2294  015720  004737  020536            JSR     PC,TTOUT        ;PRINT BLOCK NUMBER TAG
2295  015724  011103                    MOV     (R1),R3
2296  015726  004737  020724            JSR     PC,OCTP         ;PRINT BLOCK NUMBER
2297  015732  012704  022527            MOV     #MSG14,R4
2298  015736  004737  020536            JSR     PC,TTOUT        ;PRINT RECORD NUMBER TAG
2299  015742  062701  000040            ADD     #40,R1
2300  015746  012103                    MOV     (R1)+,R3
2301  015750  004737  020724            JSR     PC,OCTP         ;PRINT RECORD NUMBER
2302  015754  162701  000040            SUB     #40,R1          ;RESET POINTER FOR BLOCK NUMBER
2303  015760  005720                    TST     (R0)+
2304  015762  020077  162774            CMP     R0,@BTPT        ;SEE IF DONE
2305  015766  001405                    BEQ     BTOV2
2306  015770  012704  024534            MOV     #MSG67,R4
2307  015774  004737  020536            JSR     PC,TTOUT        ;DO CR/LF
2308  016000  000737                    BR      BTOV1           ;CONTINUE
2309  016002  005737  000772    BTOV2:  TST     BTSTF           ;SEE IF STAT ONLY PRINT
2310  016006  001002                    BNE     BTOVX           ;IF SO:  BR
2311  016010  004737  016016            JSR     PC,BTCLR        ;CLEAR TABLE
2312  016014  000207    BTOVX:  RTS     PC                      ;RETURN
2313
2314                                    ;CLEAR BAD TAPE TABLE
2315
2316  016016  012703  000041    BTCLR:  MOV     #41,R3          ;SET SIZE OF TABLE
2317  016022  013704  000762            MOV     BTPT,R4         ;SET POINTER
2318  016026  005024    BTCLR1: CLR     (R4)+           ;CLEAR ENTRY
2319  016030  005303                    DEC     R3              ;DONE?
2320  016032  001375                    BNE     BTCLR1          ;IF NOT: BR
2321  016034  000207                    RTS     PC              ;RETURN
```

```
                                        ;*******************************************************
                                        ;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 STAUTS 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).
                                        ;*******************************************************

016036  013703  000622        ERCHK:  MOV     CARCNT,R3       ;GET CHARACTER COUNT
016042  004737  022126                JSR     PC,CKSWR        ;CHECK FOR  G
016046  005037  000742                CLR     SERFL           ;CLEAR STATUS ERROR FLAG
016052  005403                        NEG     R3
016054  005737  000756                TST     TMFLG           ;A TM OPERATION?
016060  001413                        BEQ     EROA            ;IF NOT:  BR
016062  012703  000002                MOV     #2,R3
016066  005737  001010                TST     STCDFL          ;SEE IF 7 TRK CORE DUMP
016072  001401                        BEQ     1$              ;IF NOT: BR
016074  005303                        DEC     R3              ;SET TO ONE CHARACTER
016076  032777  000004  162476  1$:   BIT     #4,@MTC         ;SEE IF A WRITE TM?
016104  001401                        BEQ     EROA            ;IF NOT: BR
016106  005003                        CLR     R3              ;ELSE CLEAR R3
016110  032777  000004  162464  EROA: BIT     #4,@MTC         ;SEE IF WRITE OP
016116  001404                        BEQ     ERO
016120  062703  026204                ADD     #WDATA,R3
016124  000137  016134                JMP     ER1
016130  062703  032216        ERO:    ADD     #RDATA,R3       ;ADD START OF BUFFER
016134  010337  016740        ER1:    MOV     R3,CADER        ;SAVE EXPT ADDRESS
016140  020377  162442                CMP     R3,@MDA         ;SEE IF ADDRESS OK
016144  001105                        BNE     ER2             ;IF NOT: BR
016146  017703  162432                MOV     @MWC,R3         ;GET CHARACTER COUNT
016152  001102                        BNE     ER2             ;IF NOT ZERO: BR
016154  005037  013706                CLR     LRCPT           ;CLEAR LPC PRINT FLAG
016160  032777  000004  162414        BIT     #4,@MTC         ;A WRITE OP?
016166  001045                        BNE     ER1B            ;IF SO:  BR
016170  032777  000020  162402        BIT     #20,@MTS        ;SEE IF SEVEN TRACK DRIVE
016176  001041                        BNE     ER1B            ;IF SO: BR
016200  013737  013700  013704        MOV     EXLRC,LRCSV     ;SET FOR EXPECTED LPC
016206  005737  000756                TST     TMFLG           ;IS IT A TM?
016212  001404                        BEQ     1$              ;IF NOT:  BR
016214  000432                        BR      ER1B
```

```
2378  016216  012737  000023  013704          MOV     #23,LRCSV       ;USE TM LPC
2379  016224  013704  000616          1$:     MOV     UDES,R4         ;GET UNIT DESCRIPTION
2380  016230  042704  117777                  BIC     #117777,R4      ;MASK DENSITY
2381  016234  022704  060000                  CMP     #60000,R4       ;SEE IF 9 TRK DENSITY AT 800 BPI
2382  016240  001020                          BNE     ER1B            ;IF NOT: BR
2383  016242  017737  162342  013702          MOV     @MTD,ACTLRC     ;GET ACTUAL LPC
2384  016250  032777  020000  162374          BIT     #20000,@SWR     ;SEE IF NO DATA CHECK
2385  016256  001011                          BNE     ER1B            ;IF NOT: BR (ALLOW READ OF UNKNOWN TAPES)
2386  016260  005237  013706                  INC     LRCPT           ;SET LPC PRINT FLAG
2387  016264  042737  177000  013702          BIC     #177000,ACTLRC  ;JUST 9 BITS
2388  016272  023737  013702  013704          CMP     ACTLRC,LRCSV    ;DOES ACTUAL AGREE WITH EXPECTED?
2389  016300  001027                          BNE     ER2             ;IF NOT:  BR
2390  016302  032777  100000  162272  ER1B:   BIT     #100000,@MTC    ;SEE IF HAVE ERROR
2391  016310  001002                          BNE     1$              ;IF SO:  BR
2392  016312  000137  016720                  JMP     EREX
2393  016316  017737  162256  000740  1$:     MOV     @MTS,ERSAV      ;GET STATUS
2394  016324  005737  000756                  TST     TMFLG           ;A TM OPERATION?
2395  016330  001404                          BEQ     ER1A            ;IF NOT:  BR
2396  016332  042737  042125  000740          BIC     #42125,ERSAV    ;IGNORE TM INDICATOR AND WRL
2397  016340  001567                          BEQ     EREX            ;IF NO OTHER ERRORS:  BR
2398  016342  005737  000760          ER1A:   TST     EOTREC          ;IS IT EOT
2399  016346  100004                          BPL     ER2             ;IF NOT:  BR
2400  016350  042737  032125  000740          BIC     #32125,ERSAV    ;IGNORE EOT INDICATOR
2401  016356  001560                          BEQ     EREX            ;IF NO OTHER ERRORS:  BR
2402  016360  005237  000742          ER2:    INC     SERFL           ;SET STATUS ERROR FLAG
2403  016364  032777  002000  162260          BIT     #2000,@SWR      ;SEE IF SHOULD PRINT ERRORS
2404  016372  001411                          BEQ     ER3             ;IF SO: BR
2405  016374  005737  000774                  TST     RRTYFL          ;SEE IF READ RETRY
2406  016400  001404                          BEQ     ER2A            ;IF NOT: BR
2407  016402  022737  000003  000752          CMP     #3,RTCNT        ;SEE IF LAST RETRY
2408  016410  001402                          BEQ     ER3             ;IF SO: BR
2409  016412  000137  016664          ER2A:   JMP     EREXO           ;ELSE EXIT
2410  016416  005237  000732          ER3:    INC     PFLG            ;SET PRINT FLAG
2411  016422  004737  017506                  JSR     PC,PAPRT        ;PRINT HEADER
2412  016426  013704  000716          ER3A:   MOV     EMADDR,R4
2413  016432  004737  020536          ER3B:   JSR     PC,TTOUT        ;PRINT ERROR HEADER
2414  016436  004737  016742                  JSR     PC,FRPRT        ;PRINT F OR R
2415  016442  005037  000712                  CLR     TEMP3
2416  016446  012704  022756                  MOV     #MSG23,R4
2417  016452  004737  020536                  JSR     PC,TTOUT        ;PRINT COMMAND HEADER
2418  016456  017703  162120                  MOV     @MTC,R3
2419  016462  000303          ER7:    SWAB    R3              ;POSITION MOST SIGNIFICANT
2420  016464  004737  021152                  JSR     PC,DOUT         ;PRINT REGISTER
2421  016470  000303                          SWAB    R3              ;POSITION LEAST SIGNIFICANT
2422  016472  004737  021152                  JSR     PC,DOUT         ;PRINT REGISTER
2423  016476  005737  000712                  TST     TEMP3           ;SEE IF PRINTED STATUS REGISTER
2424  016502  001012                          BNE     ER10            ;IF SO: BR
2425  016504  005237  000712                  INC     TEMP3           ;SET FLAG
2426  016510  012704  023143                  MOV     #MSG30,R4
2427  016514  004737  020536                  JSR     PC,TTOUT        ;PRINT STATUS HEADER
2428  016520  017703  162054                  MOV     @MTS,R3         ;LOAD STATUS REGISTER
2429  016524  000137  016462                  JMP     ER7             ;GO PRINT STATUS
2430  016530  012704  023647          ER10:   MOV     #MSG46,R4
2431  016534  004737  020536                  JSR     PC,TTOUT        ;PRINT CHARACTER COUNT HEADER
2432  016540  017703  162040                  MOV     @MWC,R3
2433  016544  005403                          NEG     R3              ;SET TO TRUE VALUE
```

```
2434  016546  004737  020724          JSR     PC,OCTP         ;PRINT CHARACTER COUNT
2435  016552  012704  023654          MOV     #MSG47,R4
2436  016556  004737  020536          JSR     PC,TTOUT        ;PRINT ADDRESS HEADER
2437  016562  017703  162020          MOV     @MDA,R3
2438  016566  004737  020724          JSR     PC,OCTP         ;PRINT ADDRESS
2439  016572  012737  000255  000702  MOV     #255,TOB
2440  016600  004737  020676          JSR     PC,TOG          ;PRINT /
2441  016604  013703  016740          MOV     CADER,R3
2442  016610  004737  020724          JSR     PC,OCTP         ;PRINT EXPT ADDRESS
2443  016614  005737  013706          TST     LRCPT           ;WAS LPC CHECKED?
2444  016620  001421                  BEQ     EREX0           ;IF NOT:  BR
2445  016622  012704  025253          MOV     #MSG83,R4
2446  016626  004737  020536          JSR     PC,TTOUT        ;PRINT LPC TAG
2447  016632  013703  013702          MOV     ACTLRC,R3
2448  016636  004737  020724          JSR     PC,OCTP         ;PRINT ACTUAL  LPC
2449  016642  012737  000255  000702  MOV     #255,TOB
2450  016650  004737  020676          JSR     PC,TOG          ;PRINT -
2451  016654  013703  013704          MOV     LRCSV,R3
2452  016660  004737  020724          JSR     PC,OCTP         ;PRINT EXPECTED LPC
2453  016664  032777  100000  161760  EREX0:  BIT     #100000,@SWR    ;SEE IF STOP ON ERROR
2454  016672  001412                  BEQ     EREX            ;IF NOT: BR
2455  016674  000000                  HALT
2456  016676  005737  000732          TST     PFLG            ;SEE IF PRINT
2457  016702  001006                  BNE     EREX            ;IF SO: BR
2458  016704  032777  002000  161740  BIT     #2000,@SWR      ;SEE IF SHOULD PRINT
2459  016712  001002                  BNE     EREX            ;IF NOT: BR
2460  016714  000137  016416          JMP     ER3             ;PRINT ERROR
2461  016720  004737  022126  EREX:   JSR     PC,CKSWR        ;GO TEST FOR  G
2462  016724  005037  000732          CLR     PFLG            ;CLEAR FLAG
2463  016730  017737  161644  000740  MOV     @MTS,ERSAV      ;SAVE STATUS REGISTER
2464  016736  000207                  RTS     PC              ;RETURN
2465  016740  000000          CADER:  0                       ;EXPT ADDRESS SAVE LOCATION
2466
2467                                  ;*************************************************
2468                                  ;F FOR FORWARD/R FOR REVERSE PRINT SUBROUTINE:
2469                                  ;
2470                                  ;THIS SUBROUTINE IS USED TO PRINT OUT THE
2471                                  ;TAPE DIRECTION USED WHEN ANY ERROR IS
2472                                  ;DETECTED IN STATUS OF READ OR WRITE, DATA, OR
2473                                  ;SPACING OPERATIONS.
2474                                  ;*************************************************
2475
2476  016742  032777  000004  161632  FRPRT:  BIT     #4,@MTC         ;SEE IF WRITE COMMAND
2477  016750  001015                  BNE     FREX            ;IF SO: BR
2478  016752  032737  010000  000626  BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2479  016760  001405                  BEQ     FRO             ;IF NOT: BR
2480  016762  012704  022572          MOV     #MSG17,R4
2481  016766  004737  020536          JSR     PC,TTOUT        ;PRINT R
2482  016772  000404                  BR      FREX
2483  016774  012704  022564  FRO:    MOV     #MSG16,R4
2484  017000  004737  020536          JSR     PC,TTOUT        ;PRINT F
2485  017004  000207          FREX:   RTS     PC              ;EXIT
```

```
2486
2487                                    ;*********************************************************
2488                                    ;TAPE COMMAND EXECUTE SUBROUTINE:
2489                                    ;
2490                                    ;THIS SUBROUTINE IS USED TO EXECUTE THE
2491                                    ;MAG TAPE COMMAND DESCRIBED BY THE READ
2492                                    ;OR WRITE ROUTINE.  THE FINAL COMMAND IS
2493                                    ;SENT TO THE DEVICE REGISTER ALONG WITH THE
2494                                    ;INTERRUPT ENABLE AND GO BITS.
2495                                    ;ONCE THE COMMAND IS ISSUED, AN INTERRUPT
2496                                    ;TIMER IS STARTED AND IF NO INTERRUPT IS RETURNED
2497                                    ;BEFORE TIME OUT OCCURS, AN ERROR WILL BE
2498                                    ;PRINTED AND THE PROGRAM STOPPED.  TESTING MAY
2499                                    ;BE RESUMED BY PRESSING THE CONTINUE BUTTON.
2500                                    ;TWO INTERRUPT HANDLERS ARE USED, ONE FOR MAG TAPE
2501                                    ;AND ANOTHER FOR TELETYPE (TTY).
2502                                    ;UPON RECEIPT OF A MAG TAPE INTERRUPT, HOUSEKEEPING
2503                                    ;IS PERFORMED AND CONTROL RETURNED TO THE CALLING
2504                                    ;ROUTINE (READ,WRITE,ETC).
2505                                    ;RECEIPT OF A TTY INTERRUPT WILL CAUSE THE
2506                                    ;PROGRAM TO CHECK FOR ENTRY OF A CNTRL C CHARACTER.
2507                                    ;IF NOT CNTRL C, THEN CONTINUATION OF WAIT FOR MAG
2508                                    ;TAPE INTERRUPT IS RETURNED.  IF, HOWEVER, THE TTY
2509                                    ;INTERRUPT WAS CAUSED BY ENTRY OF A CNTRL C,
2510                                    ;THEN AT THIS TIME REQUESTS FOR NEW STALL VALUES
2511                                    ;ARE PRINTED AND THE RESPONSES ENTERED.  RESUMPTION
2512                                    ;OF TAPE INTERRUPT WAIT IS THEN RESUMED.
2513                                    ;*********************************************************
2514
2515  017006  005037  000706           TAPG:   CLR     TEMP1
2516  017012  013704  000602                   MOV     MTC,R4          ;GET COMD REGISTER ADDRESS
2517  017016  005204                           INC     R4              ;BUMP TO HIGH BYTE
2518  017020  113714  000617                   MOVB    UDES+1,(R4)     ;LOAD UNIT DESCRIPTION
2519  017024  032777  000200  161550   TAPG0:  BIT     #200,@MTC       ;SEE IF HAVE READY
2520  017032  001035                           BNE     TAPG3           ;IF SO: BR
2521  017034  005237  000706                   INC     TEMP1           ;SEE IF TIMED OUT
2522  017040  001371                           BNE     TAPG0           ;WAIT FOR READY
2523  017042  004737  017506                   JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
2524  017046  032777  000004  161526         BIT     #4,@MTC         ;SEE IF WRITE OP
2525  017054  001405                           BEQ     TAPG1           ;IF NOT: BR
2526  017056  012704  022460                   MOV     #MSG5,R4
2527  017062  004737  020536                   JSR     PC,TTOUT        ;PRINT WRITE ERR
2528  017066  000406                           BR      TAPG2
2529  017070  012704  022465           TAPG1:  MOV     #MSG6,R4
2530  017074  004737  020536                   JSR     PC,TTOUT        ;PRINT READ ERR
2531  017100  004737  016742                   JSR     PC,FRPRT        ;PRINT F OR R
2532  017104  012704  023015           TAPG2:  MOV     #MSG25,R4
2533  017110  004737  020536                   JSR     PC,TTOUT        ;PRINT NO READY ERR
2534  017114  000000                   TAPG2A: HALT
2535  017116  005037  000734                   CLR     UNP             ;RESET UNIT POINTER
2536  017122  000137  003264                   JMP     STAUTO          ;RESTART
2537  017126  000240                   TAPG3:  NOP
2538  017130  000240                           NOP
2539  017132  005037  000706                   CLR     TEMP1           ;SET DELAY
2540  017136  032777  000100  161434   1$:     BIT     #100,@MTS       ;SEE IF SELR
2541  017144  001013                           BNE     2$              ;IF SO. BR
```

```
2542  017146  005237  000706                    INC     TEMP1
2543  017152  001371                            BNE     1$              ;DELAY
2544  017154  004737  017506                    JSR     PC,PAPRT        ;PRINT HEADER
2545  017160  012704  026024                    MOV     #MSG95,R4
2546  017164  004737  020536                    JSR     PC,TTOUT        ;PRINT SELR LOST
2547  017170  000137  020050                    JMP     DRPDRV          ;GO DROP DRIVE
2548  017174  005077  161450          2$:       CLR     @PSW            ;SET TO PRIORITY 0
2549  017200  000240                            NOP
2550  017202  000240                            NOP
2551  017204  052777  000101  161370            BIS     #101,@MTC       ;SET INTERRUPT ENABLE AND GO
2552  017212  012704  020000                    MOV     #20000,R4
2553  017216  005003                            CLR     R3
2554  017220  032777  000004  161354            BIT     #4,@MTC         ;SEE IF WRITE OP
2555  017226  001042                            BNE     TAPG8           ;IF SO GO TO WRITE EOT WATCH
2556  017230  005303                    TAPG4:  DEC     R3
2557  017232  001376                            BNE     TAPG4
2558  017234  005304                            DEC     R4              ;SEE IF TIMED OUT
2559  017236  001374                            BNE     TAPG4
2560  017240  012777  000340  161402    TAPG5:  MOV     #340,@PSW       ;RESET PRIORITY
2561  017246  042777  000100  161326            BIC     #100,@MTC       ;CLEAR INTERRUPT ENABLE
2562  017254  032777  002000  161370            BIT     #2000,@SWR      ;SEE IF SHOULD PRINT ERRORS
2563  017262  001014                            BNE     TAPG6           ;IF NOT: BR
2564  017264  004737  017506                    JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
2565  017270  013704  000716                    MOV     EMADDR,R4
2566  017274  004737  020536                    JSR     PC,TTOUT        ;PRINT ERROR OP
2567  017300  004737  016742                    JSR     PC,FRPRT        ;PRINT F OR R
2568  017304  012704  022765                    MOV     #MSG24,R4
2569  017310  004737  020536                    JSR     PC,TTOUT        ;PRINT NO INTERRUPT
2570  017314  032777  100000  161330    TAPG6:  BIT     #100000,@SWR    ;SEE IF SHOULD HALT ON ERROR
2571  017322  001401                            BEQ     TAPG7           ;IF NOT: BR
2572  017324  000000                            HALT
2573  017326  000240                    TAPG7:  NOP
2574  017330  000177  161370                    JMP     @RTRN           ;RETURN TO CALLING ROUTINE
2575  017334  032777  000010  161236    TAPG8:  BIT     #10,@MTS        ;SEE IF SDWN SET
2576  017342  001012                            BNE     2$              ;IF SO: BR
2577  017344  032777  002000  161226            BIT     #2000,@MTS      ;SEE IF EOT REACHED
2578  017352  001404                            BEQ     1$              ;IF NOT: BR
2579  017354  052737  000001  017402            BIS     #1,WEOTF        ;SET EOT FLAG
2580  017362  000402                            BR      2$
2581  017364  005037  017402          1$:       CLR     WEOTF           ;CLEAR FLAG
2582  017370  005303                    2$:     DEC     R3
2583  017372  001360                            BNE     TAPG8           ;DELAY
2584  017374  005304                            DEC     R4
2585  017376  001356                            BNE     TAPG8           ;DELAY
2586  017400  000717                            BR      TAPG5
2587  017402  000000                    WEOTF:  0
2588
```

```
2589
2590                                    ;TTY INTERRUPT HANDLER**********
2591
2592  017404  012777  000340  161236  TTINT:  MOV     #340,@PSW       ;RESET PSW
2593  017412  005077  161240                  CLR     @TKS            ;CLEAR TTY STATUS
2594  017416  122777  000203  161234          CMPB    #203,@TKB       ;SEE IF CONT C
2595  017424  001404                          BEQ     TTINT0          ;IF SO: BR
2596  017426  004737  022126                  JSR     PC,CKSWR        ;GO CHECK FOR  G
2597  017432  000240                          NOP
2598  017434  000002                          RTI                     ;ELSE RETURN
2599  017436  010037  000712          TTINT0: MOV     R0,TEMP3        ;SAVE R0(REC CNTR)
2600  017442  004737  012164                  JSR     PC,TINP4        ;GO GET STALL VALUES
2601  017446  013700  000712                  MOV     TEMP3,R0        ;RESTORE R0(REC CNTR)
2602  017452  005077  161202                  CLR     @TKB            ;CLEAR TTY BUFFER
2603  017456  012777  000100  161172          MOV     #100,@TKS       ;RESET INTERRUPT ENABLE
2604  017464  000002                          RTI                     ;RETURN
2605
2606                                    ;MAG TAPE INTERRUPT HANDLER**********
2607
2608  017466  022626                  MTINT:  CMP     (SP)+,(SP)+     ;RESET STACK POINTER
2609  017470  042777  000100  161104          BIC     #100,@MTC       ;RESET INTERRUPT ENABLE
2610  017476  000240                          NOP
2611  017500  000240                          NOP
2612  017502  000177  161216                  JMP     @RTRN           ;RETURN
2613
2614                                    ;*****************************************************
2615                                    ;ERROR HEADER PRINT SUBROUTINE:
2616                                    ;
2617                                    ;THIS ROUTINE IS USED TO PFINT OUT A HEADER
2618                                    ;WITH EACH ERROR MESSAGE.  THE PRINT IS IN TWO
2619                                    ;LINES AND CONTAINS THE FOLLOWING INFORMATION.
2620                                    ;LINE 1:  UNIT NUMBER, DATA PATTERN NUMBER
2621                                    ;LINE 2:  CURRENT BLOCK NUMBER, RECORD NUMBER IN
2622                                    ;WHICH THE ERROR OCCURED PLUS THE TOTAL NUMBER
2623                                    ;OF RECORDS IN THIS BLOCK, THE RECORD SIZE (NUMBER
2624                                    ;OF CHARACTERS), AND THE ERROR TYPE (READ,WRITE, SPACE, ETC)
2625                                    ;PLUS THE TAPE DIRECTION (FORWARD OR REVERSE).
2626                                    ;ALL NUMBERS ARE IN OCTAL.
2627                                    ;*****************************************************
2628
2629  017506  012704  022504  PAPRT:  MOV     #MSG11,R4
2630  017512  004737  020536          JSR     PC,TTOUT        ;PRINT UNIT HEADER
2631  017516  013703  000616          MOV     UDES,R3
2632  017522  000303                  SWAB    R3
2633  017524  042703  177770          BIC     #177770,R3
2634  017530  004737  020724          JSR     PC,OCTP         ;PRINT UNIT NUMBER
2635  017534  012704  025564          MOV     #MSG90,R4
2636  017540  004737  020536          JSR     PC,TTOUT        ;PRINT DENSITY TAG
2637  017544  005003                  CLR     R3
2638  017546  032737  020000  000616  BIT     #20000,UDES     ;SEE IF BIT 1 OF DENSITY=1
2639  017554  001401                  BEQ     1$              ;IF NOT: BR
2640  017556  005203                  INC     R3              ;ELSE SET BIT 1
2641  017560  032737  040000  000616  1$:     BIT     #40000,UDES     ;SEE IF BIT 2 OF DENSITY=1
2642  017566  001402                  BEQ     2$              ;IF NOT: BR
2643  017570  052703  000002          BIS     #2,R3           ;ELSE SET BIT 2
2644  017574  004737  020724          2$:     JSR     PC,OCTP         ;PRINT DENSITY SETTING
```

```
 2645  017600  012704  025572                MOV     #MSG91,R4
 2646  017604  004737  020536                JSR     PC,TTOUT        ;PRINT PARITY TAG
 2647  017610  005003                        CLR     R3
 2648  017612  032737  004000  000616        BIT     #4000,UDES      ;SEE IF EVEN PARITY
 2649  017620  001401                        BEQ     3$              ;IF NOT: BR
 2650  017622  005203                        INC     R3              ;ELSE SET TO A ONE
 2651  017624  004737  020724          3$:   JSR     PC,OCTP         ;PRINT PARITY
 2652  017630  012704  025327                MOV     #MSG86,R4
 2653  017634  004737  020536                JSR     PC,TTOUT        ;PRINT PATTRN TAG
 2654  017640  032777  000400  161004        BIT     #400,@SWR       ;SEE IF RANDOM DATA
 2655  017646  001406                        BEQ     PAPRTB          ;IF NOT:  BR
 2656  017650  012737  000122  000702  PAPRTA: MOV     #122,TOB
 2657  017656  004737  020676                JSR     PC,TOG          ;PRINT R
 2658  017662  000412                        BR      PAPRTD
 2659  017664  005737  021632          PAPRTB: TST     ASEQF           ;SEE IF AUTO SEQ
 2660  017670  001403                        BEQ     PAPRTC          ;IF NOT:  BR
 2661  017672  005737  000624                TST     PATRN           ;SEE IF AUTO RANDOM
 2662  017676  100764                        BMI     PAPRTA          ;IF SO:  BR
 2663  017700  013703  000624          PAPRTC: MOV     PATRN,R3
 2664  017704  004737  020724                JSR     PC,OCTP         ;PRINT PATTRN NUMBER
 2665  017710  012704  022521          PAPRTD: MOV     #MSG13,R4
 2666  017714  004737  020536                JSR     PC,TTOUT        ;PRINT BLOCK NO.  HEADER
 2667  017720  013703  000720                MOV     BLCNTR,R3
 2668  017724  004737  020724                JSR     PC.OCTP         ;PRINT NUMBER
 2669  017730  012704  022527                MOV     #MSG14,R4
 2670  017734  004737  020536                JSR     PC,TTOUT        ;PRINT REC NO.  HEADER
 2671  017740  010003                        MOV     R0,R3
 2672  017742  032777  000004  160632        BIT     #4,@MTC         ;SEE IF WRITE OPERATION
 2673  017750  001000                        BNE     PAPRT1          ;IF SO:  BR
 2674  017752  013703  000620          PAPRT1: MOV     RCNT,R3
 2675  017756  160003                        SUB     R0,R3           ;GET RECORD NUMBER
 2676  017760  005203                        INC     R3
 2677  017762  004737  020724          PAPRT2: JSR     PC,OCTP         ;PRINT RECORD NUMBER
 2678  017766  012737  000055  000702        MOV     #55,TOB         ;LOAD DASH (-)
 2679  017774  004737  020676                JSR     PC,TOG          ;PRINT DASH (-)
 2680  020000  013703  000620                MOV     RCNT,R3
 2681  020004  004737  020724                JSR     PC,OCTP         ;PRINT RECORD COUNT
 2682  020010  012704  022472                MOV     #MSG7,R4
 2683  020014  004737  020536                JSR     PC,TTOUT        ;PRINT RECORD SIZE HEADER
 2684  020020  013703  000622                MOV     CARCNT,R3       ;GET CHARACTER COUNT
 2685  020024  005303                        DEC     R3
 2686  020026  005103                        COM     R3              ;REMOVE TWOS COMPLEMENT
 2687  020030  004737  020724                JSR     PC,OCTP         ;PRINT RECORD SIZE
 2688  020034  012737  000001  000726        MOV     #1,HDRFL        ;SET HEADER FLAG
 2689  020042  004737  022126                JSR     PC,CKSWR        ;TEST FOR  G
 2690  020046  000207                        RTS     PC              ;RETURN
 2691
```

```
2692
2693                                    ;DROP UNIT SUBROUTINE**********
2694
2695  020050  000240                DRPDRV. NOP
2696  020052  012777  010000  160522        MOV     #10000,@MTC     ;POWER CLEAR CONTROLLER
2697  020060  012704  025600                MOV     #MSG92,R4
2698  020064  004737  020536                JSR     PC,TTOUT        ;PRINT UNIT DROPPED
2699  020070  013703  000616                MOV     UDES,R3         ;GET UNIT DESCRIPTION
2700  020074  000303                        SWAB    R3
2701  020076  042703  177770                BIC     #177770,R3      ;MASK UNIT NUMBER
2702  020102  004737  020724                JSR     PC,OCTP         ;PRINT DROPPED UNIT NUMBER
2703  020106  012704  025624                MOV     #MSG93,R4
2704  020112  004737  020536                JSR     PC,TTOUT        ;PRINT REST OF MSG
2705  020116  013700  000734                MOV     UNP,R0          ;SET UNIT POINTER
2706  020122  052760  100200  001012        BIS     #100200,UN1(R0) ;SET DROPPED FLAG
2707  020130  005337  004716                DEC     REOTC           ;DECREMENT EOT UNIT COUTER
2708  020134  004737  015370                JSR     PC,PRSTAT       ;PRINT CURRENT STATS
2709  020140  005237  001006                INC     DUCTR           ;BUMP DROPPED UNIT COUNTER
2710  020144  123737  001006  004717        CMPB    DUCTR,REOTC+1   ;SEE IF DROPPED ALL UNITS
2711  020152  103406                        BLO     1$              ;IF NOT: BR
2712  020154  012704  026053                MOV     #MSG96,R4
2713  020160  004737  020536                JSR     PC,TTOUT        ;PRINT ALL DROPPED: STOP
2714  020164  000137  004640                JMP     REOT9           ;GO TO END ROUTINE
2715  020170  000240                1$:     NOP
2716  020172  005000                        CLR     R0
2717  020174  032760  100200  001012  2$:   BIT     #100200,UN1(R0) ;SEE IF ANY DRIVES LEFT IN THIS PASS
2718  020202  001414                        BEQ     3$              ;IF SO: BR
2719  020204  062700  000002                ADD     #2,R0           ;BUMP POINTER
2720  020210  022760  177777  001012        CMP     #-1,UN1(R0)     ;SEE IF LAST ENTRY
2721  020216  001366                        BNE     2$              ;IF NOT: BR
2722  020220  012704  025716                MOV     #MSG94,R4
2723  020224  004737  020536                JSR     PC,TTOUT        ;PRINT NO MORE UNITS
2724  020230  000137  004622                JMP     REOT8           ;GO TO END OF PASS ROUTINE
2725  020234  000137  004072          3$:   JMP     START7          ;GO TO NEXT UNIT
2726
2727                                    ;*************************************************************
2728                                    ;RANDOM NUMBER GENERATOR SUBROUTINE:
2729                                    ;
2730                                    ;THIS SUBROUTINE IS USED TO GENERATE THE RANDOM
2731                                    ;NUMBERS REQUIRED FOR USE AS RANDOM DATA,
2732                                    ;RECORD COUNT, AND CHARACTER COUNT.
2733                                    ;*************************************************************
2734
2735  020240  063737  000676  000672  RANG:  ADD     RANSAV,RANBAS
2736  020246  063737  000672  000676        ADD     RANBAS,RANSAV   ;GET NEW NUMBER
2737  020254  023701  000676                CMP     RANSAV,R1       ;SEE IF NUMBER TOO BIG
2738  020260  101367                        BHI     RANG            ;IF SO: BR
2739  020262  020237  000676                CMP     R2,RANSAV       ;SEE IF NUMBER TOO SMALL
2740  020266  101364                        BHI     RANG            ;IF SO: BR
2741  020270  000207                        RTS     PC              ;EXIT
2742
```

```
                                        ;*********************************************************
                                        ;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
                                        ;*********************************************************

020272  005037  000706          TTR:    CLR     TEMP1           ;CLEAR FIRST CHARACTER FLAG
020276  005000                  TTR0:   CLR     R0
020300  004737  020464          TTR0:   JSR     PC,TTIN         ;GO READ CHARACTER
020304  042737  177600  000704          BIC     #177600,TIB     ;STRIP GARBAGE
020312  122737  000015  000704          CMPB    #15,TIB         ;SEE IF CR
020320  001005                          BNE     TTR1            ;IF NOT: BR
020322  005737  000706                  TST     TEMP1           ;SEE IF FIRST CHARACTER
020326  001446                          BEQ     TTR5            ;IF SO: BR
020330  000137  020422                  JMP     TTR2            ;ELSE GO LOAD VALUE
020334  122737  000060  000704  TTR1:   CMPB    #60,TIB         ;SEE IF CHAR IS LESS THAN 0
020342  101402                          BLOS    TTR1A           ;IF NOT: BR
020344  000137  020446                  JMP     TINER           ;ELSE GO TO ERROR
020350  122737  000070  000704  TTR1A:  CMPB    #70,TIB         ;SEE IF CHAR IS GREATER THAN 7
020356  101002                          BHI     TTR1B           ;IF NOT: BR
020360  000137  020446                  JMP     TINER           ;ELSE GO TO ERROR
020364  005237  000706          TTR1B:  INC     TEMP1           ;SET FIRST CHARACTER FLAG
020370  000241                          CLC
020372  006100                          ROL     R0
020374  000241                          CLC
020376  006100                          ROL     R0              ;SHIFT 3 LEFT
020400  000241                          CLC
020402  006100                          ROL     R0
020404  042737  177770  000704          BIC     #177770,TIB     ;STRIP ASCII
020412  053700  000704                  BIS     TIB,R0          ;LOAD CHARACTER
020416  005301                          DEC     R1              ;SEE IF DONE
020420  001327                          BNE     TTR0            ;IF NOT: BR
020422  020002                  TTR2:   CMP     R0,R2           ;SEE IF EXCEEDED MAXIMUM LIMIT
020424  101402                          BLOS    TTR3            ;IF NOT: BR
020426  000137  020446                  JMP     TINER           ;ELSE GO TO ERROR
020432  020300                  TTR3:   CMP     R3,R0           ;SEE IF BELOW MINIMUM LIMIT
020434  101402                          BLOS    TTR4            ;IF NOT: BR
020436  000137  020446                  JMP     TINER           ;ELSE GO TO ERROR
020442  010015                  TTR4:   MOV     R0,(R5)         ;LOAD VALUE
020444  000207                  TTR5:   RTS     PC              ;EXIT
```

```
2796
2797                                    ;TTY ENTRY ERROR SUBROUTINE**********
2798
2799  020446  012704  023602      TINER:  MOV     #MSG43,R4
2800  020452  004737  020536              JSR     PC,TTOUT        ;PRINT?
2801  020456  162716  000020              SUB     #20,(SP)        ;RESET SP TO START OF VALUE ROUTINE
2802  020462  000207                      RTS     PC              ;REDO VALUE ENTRY
2803
2804                                    ;TTY READ SUBROUTINE**********
2805
2806  020464  005077  160166      TTIN:   CLR     @TKS
2807  020470  005077  160164              CLR     @TKB
2808  020474  005037  000704              CLR     TIB
2809  020500  005277  160152              INC     @TKS
2810  020504  105777  160146      TTIN1:  TSTB    @TKS
2811  020510  100375                      BPL     TTIN1
2812  020512  017737  160142  000704      MOV     @TKB,TIB
2813  020520  105777  160136      TTIN2:  TSTB    @TPS
2814  020524  100375                      BPL     TTIN2
2815  020526  113777  000704  160130      MOVB    TIB,@TPB
2816  020534  000207                      RTS     PC
2817
2818                                    ;TTY OUTPUT SUBROUTINE**********
2819
2820  020536  112437  000702      TTOUT:  MOVB    (R4)+,TOB
2821  020542  122737  000043  000702      CMPB    #43,TOB
2822  020550  001460                      BEQ     TEX
2823  020552  122737  000045  000702      CMPB    #45,TOB
2824  020560  001407                      BEQ     TCRLF
2825  020562  122737  000041  000702      CMPB    #41,TOB
2826  020570  001434                      BEQ     TBELL
2827  020572  004737  020676              JSR     PC,TOG
2828  020576  000757                      BR      TTOUT
2829  020600  112737  000015  000702  TCRLF:  MOVB    #15,TOB
2830  020606  004737  020676              JSR     PC,TOG
2831  020612  012703  000004              MOV     #4,R3
2832  020616  005037  000702      TCRLFA: CLR     TOB
2833  020622  004737  020676              JSR     PC,TOG
2834  020626  005303                      DEC     R3
2835  020630  001372                      BNE     TCRLFA          ;DO FILLERS
2836  020632  112737  000012  000702      MOVB    #12,TOB
2837  020640  004737  020676              JSR     PC,TOG
2838  020644  105737  001004              TSTB    RDSW
2839  020650  100401                      BMI     1$
2840  020652  000731                      BR      TTOUT
2841  020654  005037  001004      1$:     CLR     RDSW
2842  020660  000414                      BR      TEX
2843  020662  112737  000007  000702  TBELL:  MOVB    #7,TOB
2844  020670  004737  020676              JSR     PC,TOG
2845  020674  000720                      BR      TTOUT
2846  020676  105777  157760      TOG:    TSTB    @TPS
2847  020702  100375                      BPL     TOG
2848  020704  113777  000702  157752      MOVB    TOB,@TPB
2849  020712  000207              TEX:    RTS     PC
2850
2851
```

```
                                        ;OCTAL OUTPUT SUBROUTINE***********
                                
020714  012737  000001  021150  OCTPE:  MOV     #1,OFL
020722  000402                  BR      OCTPE1
020724  005037  021150          OCTP:   CLR     OFL             ;CLEAR FLAG FOR LEADING ZERO
020730  010304                  OCTPE1: MOV     R3,R4
020732  001007                  BNE     OCTPO           ;IF NOT ZERO: BR
020734  005737  021150          TST     OFL
020740  001004                  BNE     OCTPO
020742  004737  021130          JSR     PC,OCTPG1       ;ELSE PRINT ZERO
020746  000137  021072          JMP     OCTP3           ;SPACE AND EXIT
020752  032704  100000  OCTPO:  BIT     #100000,R4      ;SEE IF MSD = 1
020756  001406                  BEQ     OCTP1           ;IF NOT: BR
020760  012704  000001          MOV     #1,R4
020764  004737  021106          JSR     PC,OCTPG        ;PRINT 1
020770  000137  021002          JMP     OCTP2
020774  005004          OCTP1:  CLR     R4
020776  004737  021106          JSR     PC,OCTPG        ;PRINT 0
021002  010304          OCTP2:  MOV     R3,R4
021004  006004                  ROR     R4
021006  006004                  ROR     R4
021010  006004                  ROR     R4              ;POSITION DIGIT
021012  006004                  ROR     R4
021014  000304                  SWAB    R4
021016  004737  021106          JSR     PC,OCTPG        ;PRINT DIGIT 2
021022  010304                  MOV     R3,R4
021024  006004                  ROR     R4
021026  000304                  SWAB    R4
021030  004737  021106          JSR     PC,OCTPG        ;PRINT DIGIT 3
021034  010304                  MOV     R3,R4
021036  006104                  ROL     R4
021040  006104                  ROL     R4
021042  000304                  SWAB    R4
021044  004737  021106          JSR     PC,OCTPG        ;PRINT DIGIT 4
021050  010304                  MOV     R3,R4
021052  006004                  ROR     R4
021054  006004                  ROR     R4
021056  006004                  ROR     R4
021060  004737  021106          JSR     PC,OCTPG
021064  010304                  MOV     R3,R4
021066  004737  021106          JSR     PC,OCTPG        ;PRINT DIGIT 5
021072  012737  000240  000702  OCTP3:  MOV     #240,TOB
021100  004737  020676          JSR     PC,TOG          ;PRINT SPACE
021104  000207                  RTS     PC              ;EXIT
021106  042704  177770  OCTPG:  BIC     #177770,R4
021112  001004                  BNE     OCTPGO
021114  005737  021150          TST     OFL
021120  001001                  BNE     OCTPGO
021122  000207                  RTS     PC
021124  005237  021150  OCTPGO: INC     OFL
021130  052704  000260  OCTPG1: BIS     #260,R4
021134  010437  000702          MOV     R4,TOB
021140  004737  020676          JSR     PC,TOG
021144  010304                  MOV     R3,R4
021146  000207                  RTS     PC
021150  000000          OFL:    0                       ;FIRST CHAR FLAG
```

```
2908
2909
2910                                    ;DATA CHARACTER OUTPUT SUBROUTINE***********
2911
2912  021152  005037  000702            DOUT:   CLR     TOB
2913  021156  012704  000010                    MOV     #10,R4          ;SET NUMBER TO PRINT
2914  021162  110337  000702                    MOVB    R3,TOB
2915  021166  105777  157470            DOUT1:  TSTB    @TPS
2916  021172  100375                            BPL     DOUT1
2917  021174  132737  000200  000702          BITB    #200,TOB
2918  021202  001404                            BEQ     DOUT2
2919  021204  012777  000061  157452          MOV     #061,@TPB
2920  021212  000403                            BR      DOUT3
2921  021214  012777  000060  157442  DOUT2:  MOV     #060,@TPB
2922  021222  006137  000702            DOUT3:  ROL     TOB
2923  021226  005304                            DEC     R4
2924  021230  001356                            BNE     DOUT1
2925  021232  000207                            RTS     PC
2926
2927                                    ;ASSURE VALID STATUS DELAY SUBROUTINE********
2928
2929  021234  005777  157352            STDLY:  TST     @MTRD
2930  021240  100775                            BMI     STDLY           ;AWAIT TIMER = 0
2931  021242  005777  157344            1$:     TST     @MTRD
2932  021246  100375                            BPL     1$              ;AWAIT TIMER =1
2933  021250  005777  157336            2$:     TST     @MTRD
2934  021254  100775                            BMI     2$              ;AWAIT TIMER = 0
2935  021256  000207                            RTS     PC              ;EXIT
```

```
                                        ;AUTO SEQUENCE TEST ROUTINE**********
                                
021260  012704  025042          ASEQ:   MOV     #MSG78,R4
021264  004737  020536                  JSR     PC,TTOUT        ;PRINT CONT. REQUEST
021270  013703  021634                  MOV     ASEQCF,R3
021274  004737  020724                  JSR     PC,OCTP         ;PRINT CURRENT VALUE
021300  012705  021634                  MOV     #ASEQCF,R5      ;SET ENTRY ADDRESS
021304  012701  000001                  MOV     #1,R1           ;SET SIZE OF ENTRY
021310  012702  000001                  MOV     #1,R2           ;SET UPPER LIMIT
021314  005003                          CLR     R3              ;SET LOWER LIMIT
021316  004737  020272                  JSR     PC,TTR          ;GET INPUT
                                
021322  004737  021636          ASEQ0:  JSR     PC,HRDS         ;SELECT HARDWARE CONFIGURATION
021326  012704  025057                  MOV     #MSG79,R4
021332  004737  020536                  JSR     PC,TTOUT        ;PRINT DIVIDER
021336  012704  025125                  MOV     #MSG80,R4
021342  004737  020536                  JSR     PC,TTOUT        ;PRINT UNITS NUMBER MESSG.
021346  012700  001012                  MOV     #UN1,R0         ;POINT TOP OF DRIVE TABLE
021352  005710                  ASEQ2:  TST     (R0)            ;SEE IF END
021354  100424                          BMI     AMOD1           ;IF SO: BR
021356  011037  000706                  MOV     (R0),TEMP1      ;GET UNIT DESCRIPTION
021362  113703  000707                  MOVB    TEMP1+1,R3      ;POSITION AND
021366  042703  177770                  BIC     #177770,R3      ;  MASK UNIT NUMBER
021372  004737  020724                  JSR     PC,OCTP         ;PRINT DRIVE TABLE
021376  012704  023732                  MOV     #MSG51,R4       ;PRESET FOR 9 TRK MSG
021402  032710  020000                  BIT     #20000,(R0)     ;SEE IF 7 TRK
021406  001002                          BNE     1$              ;IF NOT: BR
021410  012704  023723                  MOV     #MSG50,R4       ;SET TO 7 TRK MSG
021414  004737  020536          1$:     JSR     PC,TTOUT        ;PRINT TRK MSG
021420  062700  000002                  ADD     #2,R0           ;BUMP POINTER
021424  000752                          BR      ASEQ2           ;DO ALL
021426  005037  000720          AMOD1:  CLR     BLCNTR
                                
021432  004737  004732          AMOD1B: JSR     PC,RWNDA                ;GO REWIND ALL DRIVES
021436  012737  000006  021630          MOV     #6,ABLCNT       ;SET NUMBER OF BLOCKS
021444  012737  174000  000622          MOV     #-4000,CARCNT   ;SET RECORD SIZE
021452  012737  000100  000620          MOV     #100,RCNT       ;SET RECORD COUNT
021460  012737  000003  000624          MOV     #3,PATRN        ;SELECT PATTERN 3
021466  005037  000646                  CLR     TMEX            ;ASSURE NO TM
021472  004737  003264                  JSR     PC,STAUTO       ;GO DO THIS PATTERN
021476  012737  000007  000624          MOV     #7,PATRN        ;SELECT PATTERN 7
021504  004737  003264                  JSR     PC,STAUTO       ;GO DO THIS PATTERN
021510  012737  000011  000624          MOV     #11,PATRN       ;SELECT PATTERN 11
021516  004737  003264                  JSR     PC,STAUTO       ;GO DO THIS PATTERN
021522  012737  177777  021630          MOV     #-1,ABLCNT      ;FORCE TO END OF TAPE
021530  012737  177777  000624          MOV     #-1,PATRN       ;SELECT AUTO RANDOM DATA
021536  012737  152634  000672          MOV     #152634,RANBAS
021544  012737  032561  000676          MOV     #32561,RANSAV   ;RESET RANDOM DATA BASE
021552  004737  003264                  JSR     PC,STAUTO       ;GO DO RANDOM
021556  012704  025057                  MOV     #MSG79,R4
021562  004737  020536                  JSR     PC,TTOUT        ;PRINT DIVIDER
021566  012704  025151          ASEQX:  MOV     #MSG81,R4
021572  004737  020536                  JSR     PC,TTOUT
021576  005737  021634                  TST     ASEQCF          ;SEE IF CONTINUOUS AUTO SEQ
021602  001003                          BNE     ASEQXX          ;IF SO: BR
021604  000000                          HALT
```

```
2992  021606  004737  022126              JSR     PC,CKSWR        ;TEST FOR  G
2993  021612  005237  000776      ASEQXX: INC     SEQCT           ;BUMP PASS COUNT
2994  021616  013703  000776              MOV     SEQCT,R3
2995  021622  004737  020724              JSR     PC,OCTP         ;PRINT PASS COUNT
2996  021626  000635                      BR      ASEQ0
2997  021630  000000              ABLCNT: 0
2998  021632  000000              ASEQF:  0
2999  021634  000000              ASEQCF: 0
3000
3001                                      ;SUBROUTINE TO SELECT AUTO SEQ HARDWARE**********
3002
3003  021636  005003              HRDS:   CLR     R3              ;CLEAR TABLE POINTER
3004  021640  005037  000706              CLR     TEMP1           ;CLEAR UNIT DESCRIPTION HOLDER
3005  021644  005037  000712              CLR     TEMP3           ;UNIT COUNT
3006  021650  005037  004716              CLR     REOTC           ;CLEAR EOT COUNTER
3007  021654  005037  000710              CLR     TEMP2           ;CLEAR UNIT INCREMENT
3008  021660  012777  010000  156714      MOV     #10000,@MTC     ;POWER CLEAR CONTROLLER
3009  021666  113737  000710  000707 HRDS1: MOVB    TEMP2,TEMP1+1   ;POSITION UNIT NUMBER
3010  021674  013777  000706  156700      MOV     TEMP1,@MTC      ;SELECT DRIVE
3011  021702  004737  021234              JSR     PC,STDLY        ;GO ASSURE VALID STATUS
3012  021706  032777  000001  156664      BIT     #1,@MTS         ;SEE IF AVAIL
3013  021714  001421                      BEQ     HRDS2           ;IF NOT: BR
3014  021716  052737  060000  000706      BIS     #60000,TEMP1    ;SET DENSITY AND PARITY
3015  021724  032777  000020  156646      BIT     #20,@MTS        ;SEE IF 7 TRK
3016  021732  001403                      BEQ     1$              ;IF NOT: BR
3017  021734  042737  020000  000706      BIC     #20000,TEMP1    ;ELSE SET TO 7 TRK NORMAL DENSITY
3018  021742  013763  000706  001012 1$:   MOV     TEMP1,UN1(R3)   ;PUT IN TABLE
3019  021750  062703  000002              ADD     #2,R3
3020  021754  005237  000712              INC     TEMP3           ;INCREMENT COUNT
3021
3022  021760  005237  000710      HRDS2:  INC     TEMP2           ;SET FOR NEXT UNIT
3023  021764  022737  000010  000710      CMP     #10,TEMP2       ;DONE?
3024  021772  001335                      BNE     HRDS1           ;IF NOT: BR
3025  021774  005703                      TST     R3              ;FOUND A UNIT?
3026  021776  001007                      BNE     HRDSX           ;IF SO: BR
3027  022000  012704  025177              MOV     #MSG82,R4
3028  022004  004737  020536              JSR     PC,TTOUT        ;TYPE NO UNIT AVAILABLE
3029  022010  000000                      HALT                    ;HALT
3030  022012  000137  003106              JMP     STAUT           ;START AUTO SEQ AGAIN
3031  022016  012763  177777  001012 HRDSX: MOV     #-1,UN1(R3)     ;MARK END OF TABLE
3032  022024  013737  000712  004716      MOV     TEMP3,REOTC     ;SET NUMBER OF UNITS
3033  022032  000337  000712              SWAB    TEMP3
3034  022036  053737  000712  004716      BIS     TEMP3,REOTC     ;SET EOT CNTR
3035  022044  000207                      RTS     PC              ;RETURN
3036
3037
3038  022046  013746  000006      SUSWR:  MOV     @#6,-(SP)               ;SAVE VECTORS
3039  022052  013746  000004              MOV     @#4,-(SP)
3040  022056  012737  022076  000004      MOV     #1$,@#4                 ;SET UP FOR TIMEOUT
3041  022064  022777  177777  156560      CMP     #-1,@SWR                ;REFERENCE HARDWARE SWITCH REGISTER
3042  022072  001402                      BEQ     2$
3043  022074  000407                      BR      3$
3044  022076  022626              1$:     CMP     (SP)+,(SP)+             ;ADJUST STACK
3045  022100  012737  000176  000652 2$:   MOV     #SWREG,SWR              ;POINT TO SOFTWARE SWITCH REG
3046  022106  012737  000174  000654      MOV     #DISPREG,DISPLAY                ;POINT TO SOFT DISPLAY REG
3047  022114  012637  000004          3$: MOV     (SP)+,@#4               ;RESTORE VECTORS
```

```
3048  022120  012637  000006                MOV     (SP)+,@#6
3049  022124  000207                        RTS     PC                      ;RETURN
3050
3051  022126  022737  000176  000652  CKSWR: CMP    #SWREG,SWR              ;SOFTWARE SWITCH REG PRESENT
3052  022134  001036                        BNE     OUT                     ;NO, GET OUT
3053  022136  017737  156516  000704        MOV     @TKB,TIB                ;AND STRIP OFF
3054  022144  042737  177600  000704        BIC     #177600,TIB             ;THE GARBAGE
3055  022152  022737  000007  000704        CMP     #7,TIB          ;IS IT A <G>
3056  022160  001024                        BNE     OUT
3057  022162  012704  026160                MOV     #$CNTG,R4
3058  022166  004737  020536                JSR     PC,TTOUT
3059  022172  012704  026164        CNTLU:  MOV     #$MSWR,R4
3060  022176  004737  020536                JSR     PC,TTOUT
3061  022202  017703  156444                MOV     @SWR,R3
3062  022206  004737  020714                JSR     PC,OCTPE
3063  022212  012704  026174                MOV     #$MNEW,R4
3064  022216  004737  020536                JSR     PC,TTOUT
3065  022222  005037  001002                CLR     @#TEMPST
3066  022226  004737  022234                JSR     PC,$READ                ;GO READ A LINE
3067  022232  000207          OUT:          RTS     PC                      ;RETURN TO MAIN BODY OF PROGRAM
3068
3069  022234  005037  001002        $READ:  CLR     TEMPST
3070  022240  012737  000007  001000        MOV     #7,COUNT
3071  022246  004737  020464        1$:     JSR     PC,TTIN         ;GO  READ A CHARACTER
3072  022252  042737  177600  000704        BIC     #177600,TIB     ;STRIP OFF GARBAGE
3073  022260  122737  000025  000704        CMPB    #25,TIB         ;IS IT A  U?
3074  022266  001002                        BNE     2$              ;BRANCH IF NOT
3075  022270  005726                3$:     TST     (SP)+           ;POP THE STACK
3076  022272  000737                        BR      CNTLU           ;START OVER
3077  022274  122737  000015  000704  2$:    CMPB    #15,TIB         ;IS IT A <CR>?
3078  022302  001013                        BNE     4$              ;BRANCH IF NOT
3079  022304  012737  000200  001004        MOV     #200,RDSW
3080  022312  004737  020600                JSR     PC,TCRLF        ;ECHO IT WITH <LF>
3081  022316  022737  000007  001000        CMP     #7,COUNT        ;WAS IT FIRST CHARACTER
3082  022324  001037                        BNE     7$              ;CHANGE SWR IF NOT FIRST ONE
3083  022326  005726                8$:     TST     (SP)+           ;POP THE STACK
3084  022330  000740                        BR      OUT             ;GET OUT
3085  022332  122737  000060  000704  4$:    CMPB    #60,TIB
3086  022340  003004                        BGT     5$
3087  022342  122737  000067  000704        CMPB    #67,TIB
3088  022350  002005                        BGE     6$
3089  022352  012704  023602          5$:   MOV     #MSG43,R4
3090  022356  004737  020536                JSR     PC,TTOUT
3091  022362  000742                        BR      3$              ;START OVER IF NOT LEGAL CHARACTER
3092  022364  006337  001002          6$:   ASL     TEMPST
3093  022370  006337  001002                ASL     TEMPST
3094  022374  006337  001002                ASL     TEMPST
3095  022400  142737  000060  000704        BICB    #60,TIB         ;GET NITTY-GRITTY
3096  022406  153737  000704  001002        BISB    TIB,TEMPST
3097  022414  005337  001000                DEC     COUNT           ;ONLY WANT 6   DIGITS
3098  022420  001754                        BEQ     5$
3099  022422  000711                        BR      1$
3100  022424  013777  001002  156220  7$:   MOV     TEMPST,@SWR     ;CHANGE SWITCH REGISTER CONTENTS
3101  022432  000735                        BR      8$
3102
3103
```

```
3104
3105                                        ;ERROR MESSAGES**********
3106
3107  022434  042052  020105     043  MSG1:   .ASCII  /*DE #/
3108
3109  022441     045  035507  021440  MSG2:   .ASCII  /%G; #/
3110
3111  022446  041045  020073     043  MSG3:   .ASCII  /%B; #/
3112
3113  022453     045  047103  021440  MSG4:   .ASCII  /%CN #/
3114
3115  022460  053452  020105     043  MSG5:   .ASCII  /*WE #/
3116
3117  022465     052  042522  021440  MSG6:   .ASCII  /*RE #/
3118
3119  022472  051052  020123     043  MSG7:   .ASCII  /*RS #/
3120
3121  022477     052  042523  021440  MSG10:  .ASCII  /*SE #/
3122
3123  022504  022445  052445  044516  MSG11:  .ASCII  /%%%UNIT NO. #/
3124  022512  020124  047516  020056
3125  022520     043
3126
3127  022521     045  041052  020116  MSG13:  .ASCII  /%*BN #/
3128  022526     043
3129
3130  022527     052  047122  021440  MSG14:  .ASCII  /*RN #/
3131
3132  022534  020045  020040  020040  MSG15:  .ASCII  /%          BAD RECORD%%#/
3133  022542  020040  020040  041040
3134  022550  042101  051040  041505
3135  022556  051117  022504  021445
3136
3137  022564  043040  025052  021452  MSG16:  .ASCII  / F****#/
3138
3139  022572  051040  025052  021452  MSG17:  .ASCII  / R****#/
3140
3141  022600  042445  052117  020040  MSG20:  .ASCII  /%EOT   NO. #/
3142  022606  047040  027117  021440
3143  022614  052445  044516  020124  MSG20A: .ASCII  /%UNIT WILL REWIND AND BE%/
3144  022622  044527  046114  051040
3145  022630  053505  047111  020104
3146  022636  047101  020104  042502
3147  022644     045
3148  022645     122  051505  040524          .ASCII  /RESTARTED ON BLOCK ONE%/
3149  022652  052122  042105  047440
3150  022660  020116  046102  041517
3151  022666  020113  047117  022505
3152  022674  044127  047105  040440          .ASCII  /WHEN ALL AVAIL UNITS REACH EOT#/
3153  022702  046114  040440  040526
3154  022710  046111  052440  044516
3155  022716  051524  051040  040505
3156  022724  044103  042440  052117
3157  022732     043
3158
3159
```

```
 3160  022733     045  020441  044441  MSG22:  .ASCII  /%!!!ILLEGAL BOT%%%#/
 3161  022740  046114  043505  046101
 3162  022746  041040  052117  022445
 3163  022754  021445
 3164
 3165  022756  041445  046517  020104  MSG23:  .ASCII  /%COMD #/
 3166  022764     043
 3167
 3168  022765     045  047516  044440  MSG24:  .ASCII  /%NO INTERRUPT RETURNED%#/
 3169  022772  052116  051105  052522
 3170  023000  052120  051040  052105
 3171  023006  051125  042516  022504
 3172  023014     043
 3173
 3174  023015     045  020441  047041  MSG25:  .ASCII  /%!!!NO CONTROLLER READY !!! STOP:%/
 3175  023022  020117  047503  052116
 3176  023030  047522  046114  051105
 3177  023036  051040  040505  054504
 3178  023044  020440  020441  051440
 3179  023052  047524  035120     045
 3180  023057     120  042522  051523          .ASCII  /PRESS CONTINUE TO RESUME TESTING%#/
 3181  023064  041440  047117  044524
 3182  023072  052516  020105  047524
 3183  023100  051040  051505  046525
 3184  023106  020105  042524  052123
 3185  023114  047111  022507     043
 3186
 3187  023121     045  051104  050117  MSG26:  .ASCII  /%DROPS: #/
 3188  023126  035123  021440
 3189
 3190  023132  050045  041511  051513  MSG27:  .ASCII  /%PICKS: #/
 3191  023140  020072     043
 3192
 3193  023143     045  052123  052101  MSG30:  .ASCII  /%STAT #/
 3194  023150  021440
 3195
 3196  023152  022445  046524  040454  MSG31:  .ASCII  /%%TM,A,B-11:TSO3 OR TU10,N,W MULTIDRIVE DATA RELIABILTY EXERCISER (DZTM
 3197  023160  041054  030455  035061
 3198  023166  051524  031460  047440
 3199  023174  020122  052524  030061
 3200  023202  047054  053454  046440
 3201  023210  046125  044524  051104
 3202  023216  053111  020105  040504
 3203  023224  040524  051040  046105
 3204  023232  040511  044502  052114
 3205  023240  020131  054105  051105
 3206  023246  044503  042523  020122
 3207  023254  042050  052132  044115
 3208  023262  043055  022451     043
 3209  023267     105  052116  051105  MSG31A: .ASCII  /ENTER CONDITIONS IN OCTAL%#/
 3210  023274  041440  047117  044504
 3211  023302  044524  047117  020123
 3212  023310  047111  047440  052103
 3213  023316  046101  021445
 3214
 3215  023322  052445  044516  020124  MSG32:  .ASCII  /%UNIT NUMBER = #/
```

```
 3216  023330  052516  041115  051105
 3217  023336  036440  021440
 3218
 3219  023342  042045  047105  044523  MSG33:  .ASCII  /%DENSITY = #/
 3220  023350  054524  036440  021440
 3221
 3222  023356  050045  051101  052111  MSG34:  .ASCII  /%PARITY = #/
 3223  023364  020131  020075     043
 3224
 3225  023371     045  042522  047503  MSG35:  .ASCII  /%RECORD COUNT = #/
 3226  023376  042122  041440  052517
 3227  023404  052116  036440  021440
 3228
 3229  023412  041445  040510  040522  MSG36:  .ASCII  /%CHARACTER COUNT = #/
 3230  023420  052103  051105  041440
 3231  023426  052517  052116  036440
 3232  023434  021440
 3233
 3234  023436  050045  052101  042524  MSG37:  .ASCII  /%PATTERN NUMBER = #/
 3235  023444  047122  047040  046525
 3236  023452  042502  020122  020075
 3237  023460     043
 3238
 3239  023461     045  044523  043516  MSG38:  .ASCII  /%SINGLE PASS = #/
 3240  023466  042514  050040  051501
 3241  023474  020123  020075     043
 3242  023501     041  020441  042445  MSG39:  .ASCII  /!!!%END OF PASS#/
 3243  023506  042116  047440  020106
 3244  023514  040520  051523     043
 3245  023521     045  042445  052116  MSG40:  .ASCII  /%%ENTER STALLS%READ = #/
 3246  023526  051105  051440  040524
 3247  023534  046114  022523  042522
 3248  023542  042101  036440  021440
 3249
 3250  023550  053445  044522  042524  MSG41:  .ASCII  /%WRITE = #/
 3251  023556  036440  021440
 3252
 3253  023562  052045  051125  020116  MSG42:  .ASCII  /%TURN AROUND = #/
 3254  023570  051101  052517  042116
 3255  023576  036440  021440
 3256
 3257  023602  037445  021445          MSG43:  .ASCII  /%?%#/
 3258
 3259  023606  042445  052116  051105  MSG44:  .ASCII  /%ENTER YOZZLE STALL = #/
 3260  023614  054440  055117  046132
 3261  023622  020105  052123  046101
 3262  023630  020114  020075     043
 3263
 3264  023635     045  051105  020122  MSG45:  .ASCII  /%ERR AMT #/
 3265  023642  046501  020124     043
 3266
 3267  023647     045  041527  021440  MSG46:  .ASCII  /%WC #/
 3268
 3269  023654  041445  020101     043  MSG47:  .ASCII  /%CA #/
 3270
 3271  023661     045  020441  047041  MSG48:  .ASCII  /%!!!NO BOT ON REWIND: #/
```

```
3272  023666  020117  047502  020124
3273  023674  047117  051040  053505
3274  023702  047111  035104     043
3275
3276  023707     040  047516  020124  MSG49:  .ASCII  / NOT AVAIL #/
3277  023714  053101  044501  020114
3278  023722     043
3279  023723     055  052067  045522  MSG50:  .ASCII  /-7TRK #/
3280  023730  021440
3281  023732  034455  051124  020113  MSG51:  .ASCII  /-9TRK #/
3282  023740     043
3283  023741     045  047516  035116  MSG52:  .ASCII  /%NON: RETRYABLE #/
3284  023746  042522  051124  040531
3285  023754  046102  020105     043
3286  023761     045  025052  047452  MSG53:  .ASCII  /%***ORIGINAL ERROR***#/
3287  023766  044522  044507  040516
3288  023774  020114  051105  047522
3289  024002  025122  025052     043
3290  024007     045  042522  047503  MSG54:  .ASCII  /%RECOVERED#/
3291  024014  042526  042522  021504
3292  024022  051045  052105  054522  MSG55:  .ASCII  /%RETRY: #/
3293  024030  020072     043
3294  024033     045  052523  050123  MSG56:  .ASCII  /%SUSPECT BAD TAPE#/
3295  024040  041505  020124  040502
3296  024046  020104  040524  042520
3297  024054     043
3298  024055     045  042522  042520  MSG57:  .ASCII  /%REPEAT: #/
3299  024062  052101  020072     043
3300  024067     045  020441  052441  MSG58:  .ASCII  /%!!!UNRECOVERABLE BAD SPOT#/
3301  024074  051116  041505  053117
3302  024102  051105  041101  042514
3303  024110  041040  042101  051440
3304  024116  047520  021524
3305
3306  024122  020445  020441  040502  MSG59:  .ASCII  /%!!!BAD TAPE OVERFLOW/
3307  024130  020104  040524  042520
3308  024136  047440  042526  043122
3309  024144  047514     127
3310  024147     045  040524  042520          .ASCII  /%TAPE WILL BE REWOUND AND REMOVED FROM/
3311  024154  053440  046111  020114
3312  024162  042502  051040  053505
3313  024170  052517  042116  040440
3314  024176  042116  051040  046505
3315  024204  053117  042105  043040
3316  024212  047522     115
3317  024215     045  042524  052123          .ASCII  /%TESTING UNTIL ALL ARE RESTARTED AT BLOCK ONE.#/
3318  024222  047111  020107  047125
3319  024230  044524  020114  046101
3320  024236  020114  051101  020105
3321  024244  042522  052123  051101
3322  024252  042524  020104  052101
3323  024260  041040  047514  045503
3324  024266  047440  042516  021456
3325  024274  052045  050101  020105  MSG60:  .ASCII  /%TAPE MARK = #/
3326  024302  040515  045522  036440
3327  024310  021440
```

```
 3328
 3329  024312  020445  020441  040502  MSG61:  .ASCII  /%!!!BACKSPACE ERROR/
 3330  024320  045503  050123  041501
 3331  024326  020105  051105  047522
 3332  024334     122
 3333  024335     045  040524  042520          .ASCII  /%TAPE WILL BE REWOUND AND REMOVED FROM /
 3334  024342  053440  046111  020114
 3335  024350  042502  051040  053505
 3336  024356  052517  042116  040440
 3337  024364  042116  051040  046505
 3338  024372  053117  042105  043040
 3339  024400  047522  020115
 3340  024404  052045  051505  044524          .ASCII  /%TESTING UNTIL ALL ARE RESTARTED AT BLOCK ONE.#/
 3341  024412  043516  052440  052116
 3342  024420  046111  040440  046114
 3343  024426  040440  042522  051040
 3344  024434  051505  040524  052122
 3345  024442  042105  040440  020124
 3346  024450  046102  041517  020113
 3347  024456  047117  027105     043
 3348  024463     052  042527  052040  MSG62:  .ASCII  /*WE TM#/
 3349  024470  021515
 3350  024472  051452  020105  046524  MSG63:  .ASCII  /*SE TM#/
 3351  024500     043
 3352  024501     045  052127  051105  MSG64:  .ASCII  /%WTERR: #/
 3353  024506  035122  021440
 3354  024512  051045  042504  051122  MSG65:  .ASCII  /%RDERR: #/
 3355  024520  020072     043
 3356  024523     045  052104  051105  MSG66:  .ASCII  /%DTERR: #/
 3357  024530  035122  021440
 3358  024534  021445                  MSG67:  .ASCII  /%#/
 3359  024536  041040  042101  052040  MSG68:  .ASCII  / BAD TAPE SPOTS%#/
 3360  024544  050101  020105  050123
 3361  024552  052117  022523     043
 3362  024557     052  042523  051040  MSG69:  .ASCII  /*SE RTY#/
 3363  024564  054524     043
 3364  024567     052  042522  052040  MSG70:  .ASCII  /*RE TM#/
 3365  024574  021515
 3366  024576  051045  040505  020104  MSG71:  .ASCII  /%READ FAILED--RETRY: #/
 3367  024604  040506  046111  042105
 3368  024612  026455  042522  051124
 3369  024620  035131  021440
 3370  024624  020445  020441  040510  MSG72:  .ASCII  /%!!!HARD READ ERROR#/
 3371  024632  042122  051040  040505
 3372  024640  020104  051105  047522
 3373  024646  021522
 3374  024650  051045  051105  040505  MSG73:  .ASCII  /%REREAD SUCCESSFUL--RETRY: #/
 3375  024656  020104  052523  041503
 3376  024664  051505  043123  046125
 3377  024672  026455  042522  051124
 3378  024700  035131  021440
 3379  024704  020045  047523  052106  MSG74:  .ASCII  /% SOFT: #/
 3380  024712  020072     043
 3381  024715     045  044040  051101  MSG75:  .ASCII  /% HARD: #/
 3382  024722  035104  021440
 3383  024726  020045  052122  054522  MSG76:  .ASCII  /% RTRY: #/
```

```
 3384  024734  020072     043
 3385  024737     045  052045  026115  MSG77:  .ASCII  /%%TM,A,B-11 AUTO SEQUENCE TEST (DZTMH-F)%/
 3386  024744  026101  026502  030461
 3387  024752  040440  052125  020117
 3388  024760  042523  052521  047105
 3389  024766  042503  052040  051505
 3390  024774  020124  042050  052132
 3391  025002  044115  043055  022451
 3392  025010  047105  042524  020122          .ASCII  /ENTER RESPONSES IN OCTAL%#/
 3393  025016  042522  050123  047117
 3394  025024  042523  020123  047111
 3395  025032  047440  052103  046101
 3396  025040  021445
 3397  025042  040445  052125  020117  MSG78:  .ASCII  /%AUTO CONT: #/
 3398  025050  047503  052116  020072
 3399  025056     043
 3400  025057     045  025045  025052  MSG79:  .ASCII  /%%******************/
 3401  025064  025052  025052  025052
 3402  025072  025052  025052  025052
 3403  025100  025052
 3404  025102  025052  025052  025052          .ASCII  /******************%#/
 3405  025110  025052  025052  025052
 3406  025116  025052  025052  022452
 3407  025124     043
 3408  025125     125  044516  051524  MSG80:  .ASCII  /UNITS TO BE TESTED%#/
 3409  025132  052040  020117  042502
 3410  025140  052040  051505  042524
 3411  025146  022504     043
 3412  025151     105  042116  047440  MSG81:  .ASCII  /END OF SEQUENCE  NO. #/
 3413  025156  020106  042523  052521
 3414  025164  047105  042503  020040
 3415  025172  047516  020056     043
 3416  025177     045  020441  047041  MSG82:  .ASCII  /%!!!NO DRIVES AVAILABLE FOR AUTO SEQ--HALT%#/
 3417  025204  020117  051104  053111
 3418  025212  051505  040440  040526
 3419  025220  046111  041101  042514
 3420  025226  043040  051117  040440
 3421  025234  052125  020117  042523
 3422  025242  026521  044055  046101
 3423  025250  022524     043
 3424  025253     045  050114  020103  MSG83:  .ASCII  /%LPC #/
 3425  025260     043
 3426  025261     045  042522  044507  MSG84:  .ASCII  /%REGISTER START = #/
 3427  025266  052123  051105  051440
 3428  025274  040524  052122  036440
 3429  025302  021440
 3430  025304  053045  041505  047524  MSG85:  .ASCII  /%VECTOR ADDRESS = #/
 3431  025312  020122  042101  051104
 3432  025320  051505  020123  020075
 3433  025326     043
 3434  025327     052  040520  052124  MSG86:  .ASCII  /*PATTRN #/
 3435  025334  047122  021440
 3436  025340  050045  042522  040515  MSG87:  .ASCII  /%PREMATURE EOT IN AUTO SEQ/
 3437  025346  052524  042522  042440
 3438  025354  052117  044440  020116
 3439  025362  052501  047524  051440
```

```
 3440  025370  050505
 3441  025372  052045  050101  020105          .ASCII  /%TAPE WILL BE REWOUND AND AUTO SEQUENCE/
 3442  025400  044527  046114  041040
 3443  025406  020105  042522  047527
 3444  025414  047125  020104  047101
 3445  025422  020104  052501  047524
 3446  025430  051440  050505  042525
 3447  025436  041516     105
 3448  025441     045  044527  046114          .ASCII  /%WILL CONINUE ON THIS UNIT#/
 3449  025446  041440  047117  047111
 3450  025454  042525  047440  020116
 3451  025462  044124  051511  052440
 3452  025470  044516  021524
 3453  025474  051040  052105  054522  MSG88:  .ASCII  / RETRY#/
 3454  025502     043
 3455
 3456  025503     045  020441  052441  MSG89:  .ASCII  /%!!!UNIT IS REWINDING; TEST WILL START WHEN DONE#/
 3457  025510  044516  020124  051511
 3458  025516  051040  053505  047111
 3459  025524  044504  043516  020073
 3460  025532  042524  052123  053440
 3461  025540  046111  020114  052123
 3462  025546  051101  020124  044127
 3463  025554  047105  042040  047117
 3464  025562  021505
 3465  025564  042052  047105  021440  MSG90:  .ASCII  /*DEN #/
 3466  025572  050052  051101  021440  MSG91:  .ASCII  /*PAR #/
 3467  025600  020441  022441  042045  MSG92:  .ASCII  /!!!%%DROPPED UNIT: #/
 3468  025606  047522  050120  042105
 3469  025614  052440  044516  035124
 3470  025622  021440
 3471  025624  040445  052124  046505  MSG93:  .ASCII  /%ATTEMPT TO RESTART UNIT WILL BE/
 3472  025632  052120  052040  020117
 3473  025640  042522  052123  051101
 3474  025646  020124  047125  052111
 3475  025654  053440  046111  020114
 3476  025662  042502
 3477  025664  046445  042101  020105          .ASCII  /%MADE AT END OF PASS!!!%%#/
 3478  025672  052101  042440  042116
 3479  025700  047440  020106  040520
 3480  025706  051523  020441  022441
 3481  025714  021445
 3482  025716  020441  022441  047045  MSG94:  .ASCII  /!!!%%NO MORE UNITS TO TEST IN THIS PASS/
 3483  025724  020117  047515  042522
 3484  025732  052440  044516  051524
 3485  025740  052040  020117  042524
 3486  025746  052123  044440  020116
 3487  025754  044124  051511  050040
 3488  025762  051501     123
 3489  025765     045  046101  020114          .ASCII  /%ALL ARE DROPPED OR REWUOND.%%#/
 3490  025772  051101  020105  051104
 3491  026000  050117  042520  020104
 3492  026006  051117  051040  053505
 3493  026014  047525  042116  022456
 3494  026022  021445
 3495  026024  020441  022441  047514  MSG95:  .ASCII  /!!!%LOST SELECT REMOTE#/
```

```
 3496  026032  052123  051440  046105
 3497  026040  041505  020124  042522
 3498  026046  047515  042524     043
 3499  026053     041  020441  040445  MSG96:   .ASCII  /!!!%ALL ARE DROPPED: END OF PASS STOP!!!#/
 3500  026060  046114  040440  042522
 3501  026066  042040  047522  050120
 3502  026074  042105  020072  047105
 3503  026102  020104  043117  050040
 3504  026110  051501  020123  052123
 3505  026116  050117  020441  021441
 3506  026124  020441  041445  047101  MSG97:   .ASCII  /!!%CANNOT TEST LOAD MEDIUM%#/
 3507  026132  047516  020124  042524
 3508  026140  052123  046040  040517
 3509  026146  020104  042515  044504
 3510  026154  046525  021445
 3511  026160  057045  021507          $CNTG:   .ASCII  /% G#/
 3512  026164  022445  053523  036522  $MSWR:   .ASCII  /%%SWR= #/
 3513  026172  021440
 3514  026174  020040  042516  036527  $MNEW:   .ASCII  /  NEW= #/
 3515  026202  021440
 3516                                   .EVEN
 3517  026204  000000                   WDATA:   0                      ;WRITE BUFFER
 3518
 3519          032216                   .=.+4010
 3520  032216  000000                   RDATA:   0                      ;READ BUFFER
 3521
 3522          000001                   .END
```

ABLCNT	021630	484	2970*	2980*	2997#								
ACTLRC	013702	1928#	2383*	2387*	2388	2447							
AMOD1	021426	2955	2967#										
AMOD1B	021432	2969#											
ASEQ	021260	1466	2938#										
ASEQCF	021634	2940	2942	2989	2999#								
ASEQF	021632	118*	411*	482	600	656	1422	1463	1644	2659	2998#		
ASEQX	021566	2987#											
ASEQXX	021612	2990	2993#										
ASEQ0	021322	2948#	2996										
ASEQ2	021352	2954#	2966										
BBC	000722	168#	1947*	1979*	1993*	2067*	2068	2074*	2075*	2076*	2077	2079*	2095*
BCNT	000736	174#	2150*	2173*	2176*	2181*	2202*	2205*	2209*	2214*			
BDPP	000766	186#	2138*	2155*	2156	2184*	2189*	2198*	2200*	2203	2207*		
BDRTY1	001214	286#	1143*	2257									
BDRTY2	001216	287#											
BDRTY3	001220	288#											
BDRTY4	001222	289#											
BDRTY5	001224	290#											
BDRTY6	001226	291#											
BDRTY7	001230	292#											
BDRTY8	001232	293#											
BD00	001454	220	325#										
BD10	001474	221	327#										
BD20	001514	222	329#										
BD30	001534	223	331#										
BD40	001554	224	333#										
BD50	001574	225	335#										
BD60	001614	226	337#										
BD70	001634	227	339#										
BKSP	010132	1007	1238#										
BK0	010172	1243	1248#										
BK1	010246	1257	1259#										
BK1A	010336	1273#	1283										
BK1B	010402	1265	1278	1284#									
BK1C	010264	1262#	1289	1291									
BK2	010322	1263	1270#										
BK3	010412	1261	1286#										
BLCNTR	000720	167#	450*	481*	484	487*	652*	740*	941	2667	2967*		
BPKP	000770	187#	2137*	2163*	2164	2183*	2188*	2199*	2201*	2212	2216*		
BP00	001254	212	309#										
BP10	001274	213	311#										
BP20	001314	214	313#										
BP30	001334	215	315#										
BP40	001354	216	317#										
BP50	001374	217	319#										
BP60	001414	218	321#										
BP70	001434	219	323#										
BTADDR	002714	363#	935	2275									
BTCLR	016016	2311	2316#										
BTCLR1	016026	2318#	2320										
BTFLG	000746	178#	585	590	593*	595	598*	965	969*	977*	1008	1256	1284*	1319*
BTOV	006570	948	975#										
BTOVX	016014	2310	2312#										
BTOV0	015670	2283	2285#										
BTOV1	015700	2288#	2308										

```
BTOV2   016002          2305    2309#
BTPRT   015606          2264    2271#
BTPT    000762           184#    935*    936     938*    939     946    2275*   2276    2282    2285    2304    2317
BTSTF   000772           188#    542*    544*    2309
BTUR    006614           956     983#
BT00    001654           344#    363
BT01    001760           346#    364
BT02    002064           348#    365
BT03    002170           350#    366
BT04    002274           352#    367
BT05    002400           354#    368
BT06    002504           356#    369
BT07    002610           358#    370
B0      010136          1239#
B0A     010444          1250    1294#
B0B     010440          1287    1293#
B1      010536          1303    1306#
B1A     010612          1309    1319#
B2      010620          1307    1320#
CADER   016740          2364*   2441    2465#
CARCNT  000622           132#    566*    771     1043    1183    1364*   1547*   1548    1550    1555*   1556    1827    1849
                        1877    1950    1971    2083    2346    2684    2971*
CCNTR   010644           496    1361#
CCSAV   000644           141#    566    1556*
CHAIN   000210           106#    429
CKSWR   022126          1154    2008    2111    2265    2347    2461    2596    2689    2992    3051#
CLLAST  013412          1869    1871#
CLP     013522          1854    1882    1890#
CLPE    013546          1891    1893    1897#
CLP0    013556          1899#   1905
CLP1    013570          1900    1902#
CLP2    013630          1909    1913#
CLP3    013642          1911    1914    1916#
CL0     013326          1853#   1870
CL1     013354          1861#
CL2     013376          1858    1867#
CL3     013464          1881#   1885
CNTLU   022172           463    3059#   3076
COUNT   001000           191#   3070*   3081    3097*
CRCLRC  013306          1657    1668    1848#
CRLR    013310          1849#
DATA0   002736           375#
DATA1   002740           376#
DATA10  002756           383#
DATA11  002760           384#
DATA12  002762           385#
DATA13  002764           386#
DATA14  002766           387#
DATA15  002770           388#
DATA2   002742           377#
DATA3   002744           378#
DATA4   002746           379#
DATA5   002750           380#
DATA6   002752           381#
DATA7   002754           382#
DATBL   002734           374#    1666
```

DATER1	001134	253#	2007*	2262					
DATER2	001136	254#							
DATER3	001140	255#							
DATER4	001142	256#							
DATER5	001144	257#							
DATER6	001146	258#							
DATER7	001150	259#							
DATER8	001152	260#							
DATR	013236	499	1648	1827#					
DATRO	013254	1831#	1834						
DATO	012552	375	1682#						
DATOA	012602	1688#	1703	1707	1710				
DATOB	012620	1691#	1694	1696					
DATOC	012704	1701	1708#						
DATOD	012712	1711#	1719						
DATOE	012722	1713#	1718						
DATOF	012736	1715	1717#						
DAT1	012754	376	1727#						
DAT1A	012760	1728#	1737	1760	1765	1770	1775	1801	1806
DAT1B	012764	1729#	1731						
DAT10	013074	383	1774#						
DAT11	013104	384	1780#						
DAT11A	013112	1782#	1785						
DAT12	013126	385	1790#						
DAT12A	013136	1792#	1795						
DAT13	013152	386	1800#						
DAT14	013162	387	1805#						
DAT15	013172	388	1810#						
DAT15A	013202	1812#	1821						
DAT15B	013206	1813#	1818						
DAT15C	013220	1815	1817#						
DAT15R	013176	1811#	1822						
DAT2	012776	377	1736#						
DAT3	013004	378	1741#						
DAT3A	013012	1743#	1754						
DAT3B	013016	1744#	1747						
DAT4	013032	379	1752#						
DAT5	013044	380	1759#						
DAT6	013054	381	1764#						
DAT7	013064	382	1769#						
DCHK	013712	1081	1200	1947#					
DEREX	014610	2069	2091	2094	2101	2103#			
DEREX1	014644	2104	2107	2109	2111#				
DERFL	000744	177#	1948*	1983*	1997*	2002			
DERR	014242	1982	1996	2040#					
DERRO	014256	2041	2043#	2110					
DERROA	014316	2045	2047	2052#					
DERROB	014336	2057#							
DERR1	014370	2064#							
DERR2	014372	2065#							
DERR3	014376	2066#							
DERR4	014400	2042	2067#						
DERR4A	014544	2092#							
DERR4B	014556	2078	2095#						
DERR5	014576	2100#							
DERR6	014604	2081	2098	2102#					

```
DFA     014004          1960    1962#   1966
DFB     014012          1965#   1970
DFC     014020          1963    1968#
DFD     014034          1967    1971#
DFX     014234          2003    2005    2009#
DFO     014044          1954    1956    1973#
DF3     014204          1986    2002#
DF7     014120          1976    1987#   2001
DF71    014152          1992    1995#
DF72    014170          1994    1998#
DF9     014062          1974    1977#   1985
DF91    014074          1978    1981#
DF92    014112          1980    1984#
DISPLA  000654           145#   3046*
DISPRE  000174            94#   3046
DOUT    021152          1274    1276    2061    2065    2420    2422    2912#
DOUT1   021166          2915#   2916    2924
DOUT2   021214          2918    2921#
DOUT3   021222          2920    2922#
DPC     014746          2147#   2193
DPCG    014754          2148    2150#
DPCO    014762          2151#   2185
DPCOA   015024          2159    2161#
DPC1    015032          2154    2163#
DPC1A   015060          2167    2169#
DPC2    015122          2152    2157    2165    2179#
DPC2A   015064          2162    2170#
DPC2B   015106          2174#   2177
DPC3    015154          2182    2186#
DPPRT   015222          2161    2169    2195#   2233
DPPRTX  015366          2215    2218#
DPPRTO  015274          2203#   2208
DPPRT1  015322          2206    2209#
DPPRT2  015340          2212#   2217
DROP    014736          2142    2145#
DRPDRV  020050           522     556     641     731     1058    2547    2695#
DRPK    014724          2142#
DRPKF   014656          1981    1995    2133#
DRP1    001054           220#   2138    2171    2189    2198
DRP2    001056           221#
DRP3    001060           222#
DRP4    001062           223#
DRP5    001064           224#
DRP6    001066           225#
DRP7    001070           226#
DRP8    001072           227#
DSUP    012370           531    1642#
DSO     012376          1644#
DSOA    012420          1645    1647    1650#
DSOB    012456          1655    1658#
DSOC    012460          1651    1659#
DS1     012510          1643    1667#   1683    1697    1716    1732    1748    1786    1796    1816    1835
DS3     012516          1083    1201    1669#
DS4     012526          1671#   1673
DUCTR   001006           194#    676*   2709*   2710
DOFL    012746          1561*   1682    1684*   1720#
```

EMADDR	000716	166#	766*	852*	962*	964*	1038*	1104*	1251*	1267	1294*	1311	2412	2565
EOTCT1	001234	297#	609*	610										
EOTCT2	001236	298#												
EOTCT3	001240	299#												
EOTCT4	001242	300#												
EOTCT5	001244	301#												
EOTCT6	001246	302#												
EOTCT7	001250	303#												
EOTCT8	001252	304#												
EOTREC	000760	183#	769*	787*	788*	822	832	1050*	1091	1108	1111*	1242	1244	1247*
		1288	2398											
ERCHK	016036	791	862	1061	1194	2346#								
EREX	016720	2392	2397	2401	2454	2457	2459	2461#						
EREXO	016664	2409	2444	2453#										
ERSAV	000740	175#	804*	805	872*	873	1068*	1069*	1259*	1260	1286*	1290*	2393*	2396*
		2400*	2463*											
ERTFL	000764	185#	773	776*	954*									
ERO	016130	2360	2363#											
EROA	016110	2351	2357	2359#										
ER1	016134	2362	2364#											
ER1A	016342	2395	2398#											
ER1B	016302	2371	2373	2377	2382	2385	2390#							
ER10	016530	2424	2430#											
ER2	016360	2366	2368	2389	2399	2402#								
ER2A	016412	2406	2409#											
ER3	016416	2404	2408	2410#	2460									
ER3A	016426	2412#												
ER3B	016432	2413#												
ER7	016462	2419#	2429											
EXCRC	013676	1876*	1886	1926#										
EXLRC	013700	1888*	1927#	2374										
FREX	017004	2477	2482	2485#										
FRPRT	016742	2051	2414	2476#	2531	2567								
FRO	016774	2479	2483#											
GDRTY1	001174	275#	1152*	2252										
GDRTY2	001176	276#												
GDRTY3	001200	277#												
GDRTY4	001202	278#												
GDRTY5	001204	279#												
GDRTY6	001206	280#												
GDRTY7	001210	281#												
GDRTY8	001212	282#												
HDRFL	000726	170#	1238*	1262	1949*	2044	2688*							
HERE	004660	664	670#											
HRDS	021636	2948	3003#											
HRDSX	022016	3026	3031#											
HRDS1	021666	3009#	3024											
HRDS2	021760	3013	3022#											
LRCPT	013706	1930#	2369*	2386*	2443									
LRCSV	013704	1929#	2374*	2378*	2388	2451								
MDA	000606	126#	772*	856*	1044*	1182*	2365	2437						
MSG1	022434	2049	3107#											
MSG10	022477	1294	3121#											
MSG11	022504	2629	3123#											
MSG13	022521	2292	2665	3127#										
MSG14	022527	2297	2669	3130#										

MSG15	022534	2072	3132#			
MSG16	022564	2483	3137#			
MSG17	022572	2480	3139#			
MSG2	022441	2058	3109#			
MSG20	022600	606	3141#			
MSG20A	022614	612	3143#			
MSG22	022733	1055	3160#			
MSG23	022756	1269	2416	3165#		
MSG24	022765	2568	3168#			
MSG25	023015	2532	3174#			
MSG26	023121	2195	3187#			
MSG27	023132	2210	3190#			
MSG3	022446	2062	3111#			
MSG30	023143	1280	2426	3193#		
MSG31	023152	416	1427	3196#		
MSG31A	023267	1429	3209#			
MSG32	023322	1467	3215#			
MSG33	023342	1502	3219#			
MSG34	023356	1514	3222#			
MSG35	023371	1535	3225#			
MSG36	023412	1545	3229#			
MSG37	023436	1557	3234#			
MSG38	023461	1576	3239#			
MSG39	023501	659	3242#			
MSG4	022453	2052	3113#			
MSG40	023521	1585	3245#			
MSG41	023550	1594	3250#			
MSG42	023562	1603	3253#			
MSG43	023602	2799	3089	3257#		
MSG44	023606	1206	3259#			
MSG45	023635	1313	3264#			
MSG46	023647	2430	3267#			
MSG47	023654	2435	3269#			
MSG48	023661	639	729	3271#		
MSG49	023707	520	554	1493	3276#	
MSG5	022460	766	809	964	2526	3115#
MSG50	023723	1498	2963	3279#		
MSG51	023732	1500	2960	3281#		
MSG52	023741	807	875	1071	3283#	
MSG53	023761	816	886	1125	3286#	
MSG54	024007	911	3290#			
MSG55	024022	913	922	3292#		
MSG56	024033	920	3294#			
MSG57	024055	926	3298#			
MSG58	024067	983	3300#			
MSG59	024122	597	3306#			
MSG6	022465	1038	1073	2529	3117#	
MSG60	024274	1567	3325#			
MSG61	024312	592	3329#			
MSG62	024463	852	877	3348#		
MSG63	024472	1251	3350#			
MSG64	024501	2234	3352#			
MSG65	024512	2244	3354#			
MSG66	024523	2259	3356#			
MSG67	024534	2272	2306	3358#		
MSG68	024536	2280	3359#			

```
MSG69   024557            962    3362#
MSG7    022472           2682    3119#
MSG70   024567           1104    3364#
MSG71   024576           1134    3366#
MSG72   024624           1140    3370#
MSG73   024650           1147    3374#
MSG74   024704           2249    3379#
MSG75   024715           2254    3381#
MSG76   024726           2239    3383#
MSG77   024737           1424    3385#
MSG78   025042           2938    3397#
MSG79   025057           2949    2985    3400#
MSG80   025125           2951    3408#
MSG81   025151           2987    3412#
MSG82   025177           3027    3416#
MSG83   025253           2445    3424#
MSG84   025261           1436    3426#
MSG85   025304           1451    3430#
MSG86   025327           2652    3434#
MSG87   025340            604    3436#
MSG88   025474            587    3453#
MSG89   025503            514    3456#
MSG90   025564           2635    3465#
MSG91   025572           2645    3466#
MSG92   025600           2697    3467#
MSG93   025624           2703    3471#
MSG94   025716           2722    3482#
MSG95   026024           2545    3495#
MSG96   026053           2712    3499#
MSG97   026124            420    1433    3506#
MTC     000602            124#    451*    452*    507*    558*    580*    583*    615     632*    701*    702*    703     718*
                          767*    775*    778*    857*   1041*   1045*   1175*   1181*   1271    1301*   1302*   1445    1488*
                         2356    2359    2370    2390    2418    2476    2516    2519    2524    2551*   2554    2561*   2609*
                         2672    2696*   3008*   3010*
MTD     000610            127#   2383
MTINT   017466            113    1461    2608#
MTRD    000612            128#   1040*   1105*   1449    2929    2931    2933
MTS     000600            123#    509     511     516     524     549     581     633     635     642     719     721     723
                          768*   1042*   1048    1052    1068    1259    1282    1300*   1438    1446    1489    1496    1959
                         1973    2372    2393    2428    2463    2540    2575    2577    3012    3015
MWC     000604            125#    771*    855*    961*   1043*   1099*   1102*   1174*   1183*   1186*   1189*   1252*   1297*
                         1306    2367    2432
NOP   = 000240             67#
OCTP    020724            611     916     925     929    1137    1150    1209    1318    1439    1454    1538    1549    1560
                         1570    1579    1588    1597    1606    2057    2204    2213    2238    2243    2248    2253    2258
                         2263    2279    2291    2296    2301    2434    2438    2442    2448    2452    2634    2644    2651
                         2664    2668    2677    2681    2687    2702    2856#   2941    2959    2995
OCTPE   020714           2854#   3062
OCTPE1  020730           2855    2857#
OCTPG   021106           2866    2869    2876    2880    2885    2890    2892    2896#
OCTPGO  021124           2897    2899    2901#
OCTPG1  021130           2861    2902#
OCTPO   020752           2858    2860    2863#
OCTP1   020774           2864    2868#
OCTP2   021002           2867    2870#
OCTP3   021072           2862    2893#
```

```
OFL     021150          2854* 2856* 2859  2898  2901* 2907#
OUT     022232          3052  3056  3067# 3084
PAPRT   017506           513   519   553   584   638   728  1054  1266  1310  2048  2160  2168  2232
                        2411  2523  2544  2564  2629#
PAPRTA  017650          2656# 2662
PAPRTB  017664          2655  2659#
PAPRTC  017700          2660  2663#
PAPRTD  017710          2658  2665#
PAPRT1  017752          2673  2674#
PAPRT2  017762          2677#
PARCNT  013710          1897* 1901* 1908  1931#
PARS    012752           449* 1654  1656* 1674* 1675* 1722#
PATRN   000624           133#  602  1559  1562  1646  1650  1661  2661  2663  2973* 2976* 2978* 2981*
PATS    012750           448*  674* 1365* 1650  1662* 1721# 1969*
PFLG    000732           172# 2043* 2106  2112* 2410* 2456  2462*
PICK    015156          2143  2187#
PIK1    001034           212# 2137  2172  2188  2199
PIK2    001036           213#
PIK3    001040           214#
PIK4    001042           215#
PIK5    001044           216#
PIK6    001046           217#
PIK7    001050           218#
PIK8    001052           219#
PRB     000670           151# 1685* 1699
PRS     000666           150# 1686* 1688* 1691
PRSTAT  015370           543  2231# 2708
PRSTA2  015400           614  2233#
PSW     000650           143#  455* 2548* 2560* 2592*
RANBAS  000672           152#  563* 2735* 2736  2982*
RANG    020240          1363  1381  1831  2735# 2738  2740
RANSAV  000676           154#  564* 1364  1382  1832  2735  2736* 2737  2739  2983*
RANSET  004110           457   563#  673
RCNT    000620           131#  565* 770   784   943  1037  1239  1253  1293  1296  1382* 1537  1539
                        1544  2674  2680  2972*
RCNTR   010676           502  1379#
RCSAV   000642           140#  565  1544*
RDATA   032216          1044  1182  1670  1714  1952  2055  2090  2093  2363  3520#
RDCMD   000626           134# 1011* 1045  1181  2478
RDER1   001114           241# 1065* 2247
RDER2   001116           242#
RDER3   001120           243#
RDER4   001122           244#
RDER5   001124           245#
RDER6   001126           246#
RDER7   001130           247#
RDER8   001132           248#
RDEX    007350          1051  1090  1092  1097  1107#
RDEXX   007374          1109  1113#
RDFL    013304           106*  431*  675* 1642  1836* 1838#
RDSW    001004           193# 2838  2841* 3079*
RD0     006746          1041# 1094
RD1     006756          1043#
RD1A    006764          1044# 1106
RD10    007276          1096#
RD2     007012          1046  1048#
```

```
RD3     007034          1049    1052#
RD4     007066          1053    1059#
RD4A    007170          1070    1076#
RD5     007174          1060    1063    1067    1075    1077#
RD6     007222          1078    1080    1083#
RD7     007242          1085    1087#
READ    006722          1012    1037#
REGST   000674           153#   1440
REOT    004142           580#    839     970     979    1010    1112
REOTC   004716           427*    471     473*    617*    623*    624     625*    626*    678#    692*    693     694*    695*
                         716*   1416*   1526*   1532    1533*   1534*   2707*   2710    3006*   3032*   3034*
REOTX   004664           662     672#
REOT1   004150           581#    582
REOT1A  004300           601     603     606#
REOT1B  004334           594     599     605     613#
REOT1C  004236           591     595#
REOT1D  004256           596     600#
REOT10  004662           671#   1435
REOT2   004344           615#    616
REOT3   004402           618     623#
REOT4   004432           629#    650
REOT5   004456           633#    634
REOT6   004522           636     642#    643
REOT6A  004540           631     645#
REOT7   004570           648     651#
REOT8   004622           422     657     659#    2724
REOT9   004640           663#   2714
RPCNT   000750           179#    819*    880*    906*    907     928
RRTY    007376          1076    1122#
RRTYFL  000774           189#   1122*   1153*   1195    2405
RRTYX   007566          1144    1153#
RRTY0   007422          1124    1127#
RRTY1   007426          1128#   1139
RRTY2   007474          1133    1138#
RRTY3   007514          1142#
RRTY4   007526          1131    1145#
RRTY5   007556          1146    1151#
RSEQ    006626           536    1001#
RSEX    006720          1002    1015#
RSFR0   006676          1006    1009    1011#
RSFR0A  006656          1004    1007#   1014
RSTAL   000632           136#   1087    1587    1589
RTCNT   000752           180#    818*    883*    915     924     930     949*    950     1127*   1129*   1136    1138    1149
                        2407
RTRN    000724           169#    779*    858*   1046*   1176*   1190*   1303*   2574    2612
RTYFL   000754           181#    794     821*    830*    865     897*    967*   2271*
RTY1    001154           264#    813*    882*    953*    976*   2242
RTY2    001156           265#
RTY3    001160           266#
RTY4    001162           267#
RTY5    001164           268#
RTY6    001166           269#
RTY7    001170           270#
RTY8    001172           271#
RWND    004720           532     688#
RWNDA   004732           689     691#   2969
```

```
RWNDX   005206          711     732#
RWND0   004756          696#    706
RWND1   005016          703#    704
RWND2   005040          698     707#
RWND3   005050          709#    726
RWND4   005120          719#    720
RWND5   005150          717     725#
RWND6   005162          722     727#
SEQCT   000776          190#   2993*   2994
SERFL   000742          176#    797     801*    863     904     960*    1062    1082*   1130    2004    2046    2348*   2402*
SPBK    010466          963    1255    1298#
SPFLG   000630          135#    661    1578    1580
STAL    000730          171#    534*    537*    551*    792*   1087*   1172*   1177*   1179*   1298*   1304*   1320*   1345*
                       1491*
STALL   010634          535     538     793    1088    1173    1180    1299    1321    1345#   1346
START   002772          100     411#
STARTA  003124          103     433#
STARTB  003146          432     438#
STARTC  003174          445#    447
STARTD  003236          453#
STARTE  003142          107     437#
STARTO  003156          440#    442
START1  003356          474#    506     559
START2  003512          495     497#
START3  003526          498     500#
START4  003542          480     501     503#
START5  004016          540     545#
START6  004034          549#    552
START7  004072          550     557#    622    2725
START8  004104          548     559#
STAROA  003310          460     462     464#    469
STAROB  003340          465     470#
STAR1A  003402          479#
STAR1B  003414          477     481#
STAR1C  003456          483     485     490#
STAR40  003562          504     507#
STAR44  003630          516#    517
STAR45  003644          512     519#
STAR46  003664          510     518     523#
STAS    014240         1988*   1989    1999*   2011#
STAUT   003106          119     414     430#    3030
STAUTO  003264          436     458#    677    2536    2975    2977    2979    2984
STCDFL  001010          195#    523*    530*   1100    1187    1955    1975    2353
STCS    014236         1989*   1999    2010#
STDLY   021234          508    2929#   2930    3011
SUSWR   022046          435     454    3038#
SWR     000652          144#    412*    423*    459     494     497     500     539     545     688     764     789     802
                        814     835     860     870     884     909     918    1001    1003    1005    1013    1059    1066
                       1077    1084    1123    1132    1145    1192    1198    1218    1264    1308    2040    2070    2103
                       2108    2158    2166    2384    2403    2453    2458    2562    2570    2654    3041    3045*   3051
                       3061    3100*
SWREG   000176           95#    459    3045    3051
TAPG    017006          780     859    1047    1178    1191    1305    2515#
TAPGO   017024         2519#   2522
TAPG1   017070         2525    2529#
TAPG2   017104         2528    2532#
```

```
TAPG2A  017114          2534#
TAPG3   017126          2520    2537#
TAPG4   017230          2556#   2557    2559
TAPG5   017240          2560#   2586
TAPG6   017314          2563    2570#
TAPG7   017326          2571    2573#
TAPG8   017334          2555    2575#   2583    2585
TBELL   020662          2826    2843#
TCRLF   020600          2824    2829#   3080
TCRLFA  020616          2832#   2835
TEMPST  001002           192#   3065*   3069*   3092*   3093*   3094*   3096*   3100
TEMP1   000706           162#   1475    1509    1521    1687*   1700    1704*   2083*   2084*   2085*   2086*   2087    2089
                        2133*   2140*   2145    2190    2515*   2521*   2539*   2542*   2761*   2767    2776*   2956*   2957
                        3004*   3009*   3010    3014*   3017*   3018
TEMP2   000710           163#   1469*   1470    1504*   1516*   1617*   1621    1705*   1712    2134*   2141*   2146    2191
                        3007*   3009    3022*   3023
TEMP3   000712           164#   1210*   1216    1272*   1277    1279*   2135*   2153    2192*   2415*   2423    2425*   2599*
                        2601    3005*   3020*   3032    3033*   3034
TEMP4   000714           165#   1248*   1254    1297    1689*   1693*
TEX     020712          2822    2842    2849#
TIB     000704           161#   2764*   2765    2770    2773    2783*   2784    2808*   2812*   2815    3053*   3054*   3055
                        3072*   3073    3077    3085    3087    3095*   3096
TINER   020446          2772    2775    2789    2792    2799#
TINF    000700           159#    430*    433*    437*   1412
TINP    010722           456    1412#
TINPA   010732          1413    1415#
TINPB   010752          1419#   1421
TINPB1  011000          1423    1427#
TINPO   011020          1426    1431#
TINPOA  011310          1478    1480#
TINPOB  011326          1476    1483#
TINPOC  011364          1489#   1492
TINPOD  011416          1490    1496#
TINPOE  011436          1497    1500#
TINPOF  011442          1499    1501#
TINPOO  011230          1464    1467#   1479    1495    1530
TINP1   011446          1502#
TINP2   011526          1510    1514#
TINP2A  011606          1522    1526#
TINP2B  011634          1482    1528    1531#
TINP3   011654          1535#
TINP4   012164          1585#   2600
TKB     000660           147#   1204    2594    2602*   2807*   2812    3053
TKS     000656           146#    474*   1171*   1202    1220*   2593*   2603*   2806*   2809*   2810
TMEX    000646           142#    827    1096    1249    1569    1571    2974*
TMFLG   000756           182#    831*    854*    889*    899    1039*   1079    1089    1103*   1107*   1184    2350    2375
                        2394
TOB     000702           160#    438    2439*   2449*   2656*   2678*   2820*   2821    2823    2825    2829*   2832*   2836*
                        2843*   2848    2893*   2903*   2912*   2914*   2917    2922*
TOG     020676          2440    2450    2657    2679    2827    2830    2833    2837    2844    2846#   2847    2894    2904
TPB     000664           149#   2815*   2848*   2919*   2921*
TPOS    012342          1486    1513    1525    1616#   1619
TPS     000662           148#   2813    2846    2915
TSTAL   000636           138#    534     537    1298    1320    1605    1607
TTIN    020464          2763    2806#   3071
TTINT   017404            87    2592#
```

TTINTO	017436	2595	2599#											
TTIN1	020504	2810#	2811											
TTIN2	020520	2813#	2814											
TTOUT	020536	417	421	515	521	555	588	607	613	640	660	730	808	810
		817	876	878	887	912	914	921	923	927	984	1056	1072	1074
		1126	1135	1141	1148	1207	1268	1270	1281	1312	1314	1425	1428	1430
		1434	1437	1452	1468	1494	1501	1503	1515	1536	1546	1558	1568	1577
		1586	1595	1604	2050	2053	2059	2063	2073	2196	2211	2235	2240	2245
		2250	2255	2260	2273	2281	2294	2298	2307	2413	2417	2427	2431	2436
		2446	2481	2484	2527	2530	2533	2546	2566	2569	2630	2636	2646	2653
		2666	2670	2683	2698	2704	2713	2723	2800	2820#	2828	2840	2845	2939
		2950	2952	2964	2986	2988	3028	3058	3060	3064	3090			
TTP	020272	1215	1444	1459	1474	1508	1520	1543	1554	1566	1575	1584	1593	1602
		1611	2761#	2946										
TTP0	020300	2763#	2786											
TTP1	020334	2766	2770#											
TTP1A	020350	2771	2773#											
TTP1B	020364	2774	2776#											
TTR2	020422	2769	2787#											
TTR3	020432	2788	2790#											
TTR4	020442	2791	2793#											
TTR5	020444	2768	2794#											
UDES	000616	130#	479*	493*	507	526	580	629*	630	632	700*	701	713*	714
		718	735*	736	1487*	1488	1511*	1523*	1652	1674	1892	1910	1913	1953
		2379	2518	2631	2638	2641	2648	2699						
UNP	000734	173#	475	488*	490*	505*	557*	608	619	627*	628	645*	646	653*
		691*	696	705*	708*	709	725*	734*	799	812	868	881	932	934
		952	975	1041	1142	1151	1415*	1477	1480	1483	1527	1529*	1531*	1620
		2006	2136	2170	2187	2197	2236	2241	2246	2251	2256	2261	2274	2535*
		2705												
UNX	001032	208#												
UN1	001012	200#	464*	466*	467*	470*	476*	478*	479	486*	491*	493	503	620*
		629	644*	647*	649*	651*	697*	699*	700	707*	710*	712*	713	732*
		735	1418	1481*	1484*	1487	1621*	2706*	2717	2720	2953	3018*	3031*	
UN2	001014	201#	424											
UN3	001016	202#												
UN4	001020	203#												
UN5	001022	204#												
UN6	001024	205#												
UN7	001026	206#												
UN8	001030	207#												
VECT	000614	129#	1453	1455	1460									
WDATA	026204	772	856	1660	1711	1828	1851	1879	1951	1972	2088	2361	3517#	
WEOTF	017402	781	783*	2579*	2581*	2587#								
WEX	005636	823	827#											
WEX1	005650	828	830#											
WEX2	005666	834#	838											
WRITE	005254	533	764#											
WRTSB	006504	898	960#											
WRTSBO	006566	966	971#											
WRTY	006142	820	888	897#										
WRTYR	006174	902	904#											
WRTYTM	006170	900	903#											
WRTY0	006150	898#	908	955										
WRTY1	006256	910	917#											
WRTY2	006260	905	918#											

```
WRTY3   006340            919     930#
WRTY3A  006356            931     934#
WRTY4   006444            947     949#
WRTY5   006500            951     956#
WSTAL   000634            137#    792     1596    1598
WTER1   001074            231#    444      800*    869*    933*   2237
WTER2   001076            232#
WTER3   001100            233#
WTER4   001102            234#
WTER5   001104            235#
WTER6   001106            236#
WTER7   001110            237#
WTER8   001112            238#
WTM     005712            829     852#     903
WTM0    005760            858     860#
WTM1    006012            866     868#
WTM2    006070            874     880#
WTM3    006130            885     888#
WTM4    006134            861     864      871     889#
W0      005312            771#    825      901
W0A     005350            774     778#
W0B     005356            777     779#
W1      005424            782     789#
W1A     005370            779     781#
W3      005440            790     792#
W3A     005462            795     797#
W3A0    005460            765     796#
W3B     005552            806     812#
W3C     005602            815     818#
W3D     005616            798     803      811     821#
W4      005670            833     835#
XOR     013650           1855    1883     1887    1918#
XORS    013674           1852*   1867*    1871    1872*   1873*   1875*   1876    1880*   1888    1919    1920*   1921*   1922
                         1925#
YOZ     007600           1086    1128     1171#   1217    1219
YOZA    007626           1175#
YOZB    007634           1176#
YOZC    007654           1176    1179#
YOZC1   007736           1185    1188     1190#
YOZD    007750           1190    1192#
YOZE    007764           1193    1195#
YOZE1   007774           1196    1198#
YOZF    010010           1199    1201#
YOZG    010112           1203    1205     1218#
YOZO    007620           1174#
YSTAL   000640            139#   1172     1179    1208    1211
$CNTG   026160           3057    3511#
$ENDAD  004650             79     666#
$MNEW   026174           3063    3514#
$MSWR   026164           3059    3512#
$READ   022234           3066    3069#
$SVPC = 000042             77#     82
      = 032220             71#     72#      77      78#     80#     82#     86#     93#     99#     102#    105#    112#    117#
                          120#    310#     312#    314#    316#    318#    320#    322#    324#    326#    328#    330#    332#
                          334#    336#     338#    340#    345#    347#    349#    351#    353#    355#    357#    359#    374
                         3519#
```

```
COMMEN      1#
ENDCOM      1#
ESCAPE      1#
GETPRI      1#
GETSWR      1#
MULT        1#
NEWTST      1#
POP         1#
PUSH        1#
REPORT      1#
SETPRI      1#
SETUP       1#
SKIP        1#
SLASH       1#
STARS       1#      75
SWRSU       1#
TYPBIN      1#
TYPDEC      1#
TYPNAM      1#
TYPNUM      1#
TYPOCS      1#
TYPOCT      1#
TYPTXT      1#
$$ESCA      1#
$$NEWT      1#
$$SKIP      1#
.EQUAT      1#
.HEADE      1#
.KT11       1#
.SETUP      1#
.SWRHI      1#
.$ACT1      1#      6#      73
.$APTB      1#
.$APTH      1#
.$APTY      1#
.$ASTA      1#
.$CATC      1#
.$CMTA      1#
.$DB2D      1#
.$DB20      1#
.$DIV       1#
.$EOP       1#
.$ERRO      1#
.$ERRT      1#
.$MULT      1#
.$POWE      1#
.$RAND      1#
.$RDDE      1#
.$RDOC      1#
.$READ      1#
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB20      1#
.$SCOP      1#
.$SIZE      1#
```

```
.$SUPR      1#
.$TRAP      1#
.$TYPB      1#
.$TYPD      1#
.$TYPE      1#
.$TYPO      1#
.$40CA      1#
.1170       1#


 ABS.   032220     000


ERRORS DETECTED:  0

DZTMHF.BIN,DZTMHF.LST/CRF/SOL/NL:TOC=DZTMHF.SML,DZTMHF.P11
RUN-TIME:  9 13 1 SECONDS
RUN-TIME RATIO:  267/24=10.7
CORE USED:  32K  (63 PAGES)
```

ACO

L