TM03/TE16

DATA RELIABILITY TEST

MD-11-DZTED-A

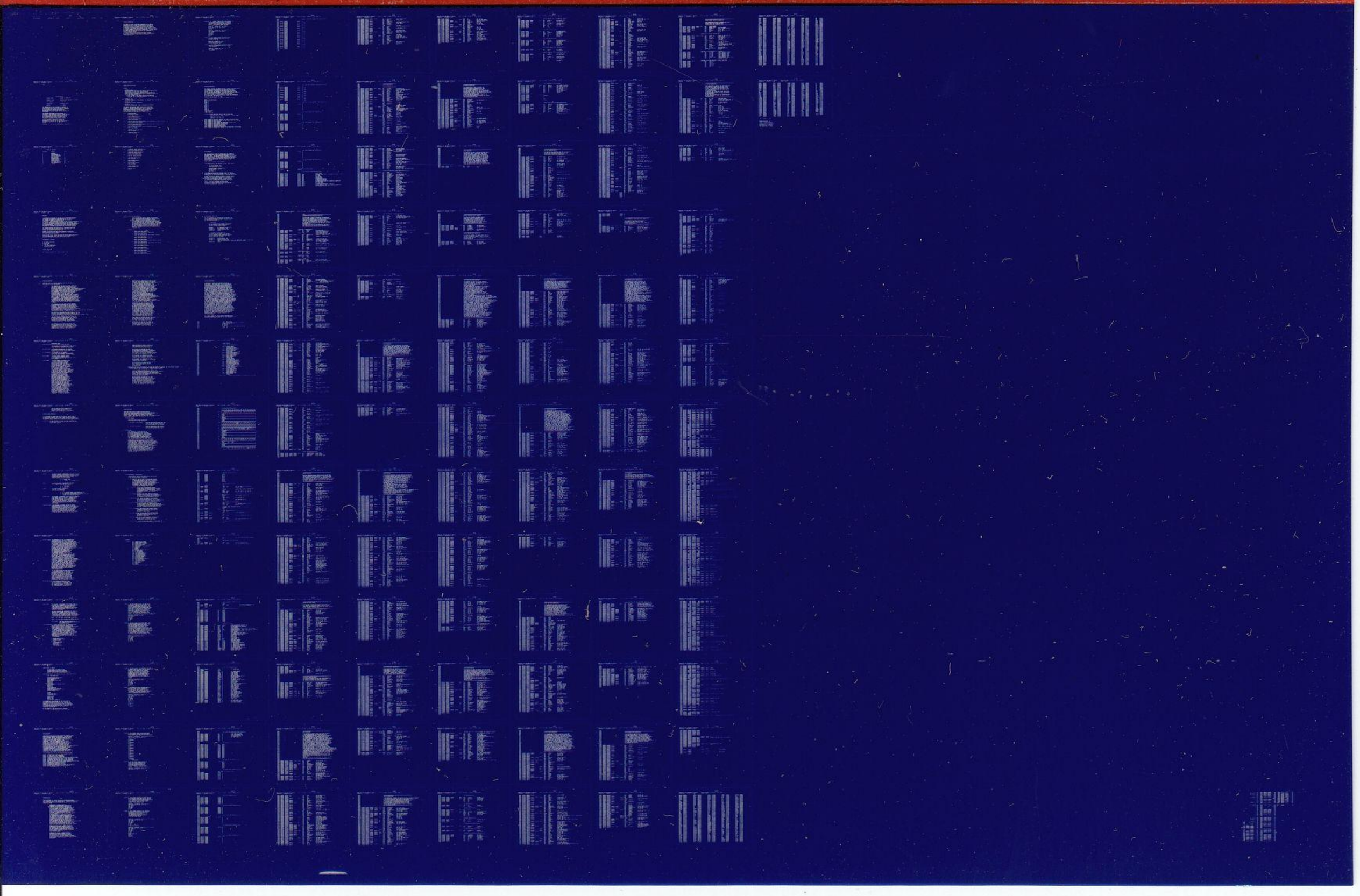
EP-DZTED-A-DL-A

FICHE 1 OF 1

JUN 1977

digital

MADE IN USA



801

EOF1DZTECASE0	00010000	770526	POP10 411	HOR1DZTEDASE0	00010000	770526

.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZTED-A-D

PRODUCT TITLE: TM03/TE16 DATA RELIBILITY PROGRAM

DATE CREATED: 21 FEB 1977

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: J. G. ADAMS

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 THE TE16 MAGNETIC ON A MASSBUS THROUGH THE TM03 MAG TAPE CONTROLLER. ANY COMBINATION OF TM03'S & TE16'S UP TO A MAXIMUM OF 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 RH AND TM03.

HOWEVER; THE RH AND TM03 ARE 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 AS DETECTED BY THE RH OR TM03.

2. REQUIREMENTS (HARDWARE)

A. ANY PDP-11 PROCESSER
B. 8K OF CORE
C. TELETYPE
D. TM03 TAPE CONTROLLER
E. 1 TO 8 MAG TAPE DRIVES
F. MASSBUS CONTROLLER

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING BINARY TAPES

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 RH STARTING ADDRESS, VECTOR ADDRESS, DRIVE NUMBER(TM03 ADDRESS), SLAVE NUMBER, DENSITY,PARITY,FORMAT,RECORD COUNT,CHARACTER COUNT,PATTERN NUMBER,TAPE MARK 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 OF MORE THAN ONE CHARACTER NEED NOT HAVE LEADING ZEROS AND SHOULD BE TERMINATED BY A CARRIAGE RETURN IF LESS THAN THE MAXIMUM NUMBER OF CHARACTERS IS INPUT.

B. 204(8): THIS ADDRESS SHOULD BE USED ANYTIME A RESTART OF THE PROGRAM IS NECESSARY AND THE PARAMETERS ENTERED AT THE INITIAL START OF 200(8) NEED NOT BE CHANGED. ALSO NOTE THAT ANY DATA PATTERN WHICH HAD BEEN GENERATED BY SETTING THE RANDOM DATA SWITCH (CONSOLE SWITCH EIGHT) WILL NOT BE OVERWRITTEN AND THEREFORE IS HELD IN CORE FOR USE UNTIL CONSOLE SWITCH EIGHT(8) IS AGAIN SET AND THAT ALL STATISTICS WILL BE RETAINED.

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 AND ALL STATISTICS ARE CLEARED TO ZERO.

D. 240(8): THIS IS A SPECIAL ADDRESS WHICH WILL CAUSE THE PROGRAM TO EXECUTE A PREDETERMINED TEST PLAN ON ALL AVAILABLE DRIVES AND SLAVES. THE ONLY INPUT REQUIRED BY THE OPERATOR IS A RESPONSE TO REQUESTS FOR THE RH ADDRESS, VECTOR ADDRESS, CONTINUOUS OPERATION OF THE SEQUENCE, AND NRZ ONLY.

E. 300(8): THIS ADDRESS IS TO BE USED AS A RESTART ONLY AND WILL PERFORM JUST AS IN 200(8) EXCEPT THAT THE PARAMETER INPUT LIST IS SHORTENED. THE SHORT PARAMETER LIST CONSISTS OF DRIVE NUMBER, SLAVE NUMBER, DENSITY, PARITY, FORMAT, RECORD COUNT, CHARACTER COUNT, PATTERN, TAPE MARK, AND

INTERCHANGE READ.
**NOTE SEE ALSO SECTION 8-CONSOLE SWITCH SETTINGS

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 RH REGISTER (CS1) AS A SIX DIGIT UNIBUS ADDRESS.

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

DRIVE NUMBER: THE DRIVE NUMBER (MASSBUS ADDRESS OF THE TM03) IS ENTERED AS ONE (1) OCTAL CHARACTER AND MUST BE WITHIN THE LIMITS OF 0 THROUGH 7.

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

THAT AT LEAST ONE SLAVE NUMBER REQUEST
MUST BE ENTERED. IF THE FIRST
REQUEST IS RESPONDED TO BY A CARRIAGE
RETURN, THEN THE REQUEST WILL BE REPEATED.

4.1 AUTOMATIC MODE OPERATION

IF THE PROGRAM IS LOADED AND RUN IN AUTOMATIC (CHAIN) MODE
THE AUTO ACCEPT SEQUENCE TEST PLAN IS RUN. SEE SEC 11. BELOW;
THE SOFTWARE SWR IS INVOKED WITH A SWITCH SETTING OF 100000 (HALT
ON ERROR) IF LOADED VIA ACT11. NO OPERATOR INTERVENTION IS REQUIRED.

**EXCEPTION: IF THIS PROGRAM IS LOADED VIA TMOP CHAIN MODE THE
PROGRAM WILL TEST ALL SLAVES ON THE FIRST AVAILABLE
DRIVE EXCEPT SLAVE 0.

DENSITY: THE DESITY REQUEST IS RESPONDED TO BY ONE (1) OCTAL CHARACTER AND MUST BE WITHIN THE LIMITS OF 0 THRU 4. AS EACH SLAVE NUMBER IS ENTERED, A REQUEST FOR THE OPERATING DENSITY FOR THAT SLAVE IS TYPED. THE RESPONSE MEANINGS ARE AS FOLLOWING:

A. 3 = 800BPI, NRZI
B. 4 = 1600BPI, PE (9 CHANNEL ONLY)

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

FORMAT: THE FORMAT REQUEST IS RESPONDED TO BY TWO (2) CHARACTERS AND SHOULD BE AS FOLLOWS

A. 14 = 9 CHANNEL NORMAL (TWO FRAMES PER WORD)
B. 15 = CORE DUMP (FOUR FRAMES PER WORD)
C. 16 = PDP-15 OR IBM COMPATABLE(TWO FRAMES PER WORD)
(DATA IS BYTE SWAPPED ON TAPE)

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 20 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 15(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 200(8), 210(8), OR 300(8).
WHEN OPERATING IN NRZ MODE (DENSITY 0-3)
THE PROGRAM CONSTRUCTS AND SAVES
BOTH AN EXPECTED CRC CHARACTER AND AN
LRC CHARACTER FOR COMPARISONS WITH
THE HARDWARE GENERATED CHECK CHARACTER
IN BOTH READ AND WRITE.
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-A-D) 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
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.

TAPE MARK: THE TAPE MARK REQUEST IS USED TO DETERMINE
IF THE OPERATOR WISHES TO HAVE EACH
DATA BLOCK SEPERATED BY A TAPE MARK. IF
RESPONDED TO BY A ONE (1) THE TAPE MARK
WILL BE WRITTEN AND WHEN READING WILL
BE EXPECTED AT THE END OF DATA BLOCK.
A ZERO (0) RESPONSE WILL DISALLOW TAPE
MARK. 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.

INTERCHANGE READ: THIS REQUEST IS RESPONDED TO BY
A SINGLE CHARACTER INPUT OF EITHER
ONE (1) OR ZERO (0). A RESPONSE OF ONE (1)
WILL CAUSE ALL READING TO BE DONE IN
THE INTERCHANGE MODE. A ZERO RESPONSE
WILL CAUSE READING IN NORMAL MODE.

SINGLE PASS: THIS REQUEST IS RESPONDED TO BY EITHER A ONE (1) OR A ZERO (0). RESPONSE OF 1, WILL CAUSE THE TEST TO BE STOPPED AFTER THE LAST AVAILABLE DRIVE REACHES END OF TAPE. A RESPONSE OF 0, WILL ALLOW CONTINUOUS RUNNING THROUGH MULTIPLE PASSES. TO RESTART AT END OF PASS, PRESS CONTINUE, OR RESTART AT THE CONSOLE.

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 PARAMTERS EXCEPT FOR THE SLAVE DESCRIPTION VALUES (SLAVE NUMBER, DENSITY, PARITY AND FORMAT) HAVE NOMINAL VALUES ALREADY STORED IN THE PROGRAM. COUNT, CHARACTER COUNT, TAPE MARK 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. TM=0
E. INTERCHANGE READ = 0
F. SINGLE PASS = 0
G. READ STALL = 1
H. WRITE STALL = 1
I. TURN AROUND STALL = 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:

        TE16 TAPE DRIVE TEST

        REGISTER START=172440(172440)
        VECTOR ADDRESS=224(CR)
        DRIVE NUMBER (4)
        SLAVE NUMBER=(5) SN: 5009
        DENSITY=(3)
        PARITY=(0)
        FORMAT=(14)
        SLAVE NUMBER=(2) 9 CHAN SN: 0022
        DENSITY=(3)
        PARITY=(1)
        FORMAT=(15)
        SLAVE NUMBER=(CR)
        RECORD COUNT=100 (500)(CR)
        CHARACTER COUNT=200 (38)?(7)(CR)
        PATTERN NUMBER=1 (22)
        ?
        (6)(CR)
        TM=(0)
        INTERCHANGE READ=(1)
        SINGLE PASS=(0)

        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 SLAVE 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.

NO TAPE MARKS WILL BE WRITTEN AND ALL READING
WILL BE DONE IN INTERCHANGE MODE (MAINT MODE 0001).
```

5. DATA PATTERNS

THERE ARE FIFTEEN 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 (4000 CHARACTERS) WILL BE FILLED WITH THE PATTERN ENTERED SO THAT ANY SIZE RECORD CAN BE USED. (SEE DTC MAINDEC-11-DZTUF-A-D) THE PROGRAM GENERATES A CYLIC REDUNDENCY CHECK CHARACTER (CRC) AND A LONGITUDINAL REDUNDENCY CHECK CHARACTER (LRC) FOR COMPARISONS AGAINST THE CRC AND LRC GENERATED BY THE HARDWARE IN NRZI READS OR WRITES.

THE FOLLOWING IS A LIST OF THE DATA PATTERNS AVAILABLE:

DATA0: EXTERNAL INPUT THRU HIGH SPEED READER (SEE DTC)
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: WALKING ONE/ALL ONE IN ALTERNATING CHARACTERS
DATA11: INCREMENTING CHARACTERS (000-377)
DATA12: DECREMENTING CHARACTERS (377-000)
DATA13: ALTERNATING CHARACTERS OF ALL ZERO AND ALL ONE BITS
DATA14: WALKING ZERO/ALL ZERO IN ALTERNATING CHARACTERS
DATA15: AUTO SEQUENCE PATTERN 0,0,-1,-1,-1,0,0

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 APPROIATE 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 B 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.
THE YOZZLE STALL IS ALSO DYNAMIC AND CAN BE CHANGED BY TYPING A CONTROL B WHILE DOING A YOZZLE. A YOZZLE STALL REQUEST WILL BE PRINTED AND SHOULD BE RESPONDED TO WITH THE DESIRED VALUE.

8. CONSOLE SWITCH SETTINGS

CONTROL:

1) CONTROL G (↑G):
   SELECTS SOFTWARE SWR AND ALLOWS USER TO SELECT NEW SWITCHES.
   THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW=
   WHERE: XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWR.
   AFTER THE 'NEW=' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE
   OF THE FOLLOWING AT THE TTY:
     A) TYPE A NUMBER TO BE LOADED INTO THE SOFTWARE SWR
     B) IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWR
        CONTENTS WILL NOT BE CHANGED.

2) CONTROL A (↑A):
   ALTERNATES USAGE OF THE SWR BETWEEN THE HARDWARE SWR & SOFTWARE SWR.

3) CONTROL B (↑B):
   SEE SECTION 7 DYNAMIC PARAMETERS

4) CONTROL U (↑U):
   DELETES ALL CHARACTERS TYPED IN RESPONSE TO A REQUEST.

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=PRINT READ/WRITE STATISTICS
       0=DO NOT PRINT STATS

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

SW12: 1=DO NOT CHECK WRITE STATUS ERRORS (NOR CLEAR THEM IF THEY DO OCCUR)
       0=CHECK WRITE STATUS ERRORS

SW11: 1=DO NOT CHECK READ STATUS ERRORS (NOR CLEAR THEM IF THEY DO OCCUR)
       0=CHECK READ STATUS ERRORS

SW10: 1=DO NOT PRINT ANY ERRORS (EXCEPT CATASTROPHIC 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=DO WRITE/READ RETRIES
                0=DO NOT RETRY

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

        SW2:    1=DO NOT READ REVERSE
                0=READ REVERSE

        SW1:    1=READ FORWARD FIRST
                0=READ REVERSE FIRST

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

SWITCH EXPLANATION AND EXAMPLES:

SW0-3: 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 REACH EOT, AT WHICH TIME TESTING IS RESUMED ON ALL AVAILABLE UNITS.

EXAMPLES: 0-3

A. SW0=0,SW1=0,SW2=1,SW3=1
WRITE ONLY X RECORDS OF Y CHARACTERS

B. SW0=0,SW1=0,SW2=1,SW3=0
WRITE THEN BACKSPACE AND READ FORWARD X RECORDS

C. SW0=0,SW1=0,SW2=0,SW3=1
WRITE THEN READ REVERSE X RECORDS.

D. SW0=0,SW1=0,SW2=0,SW3=0
WRITE THEN READ REVERSE AND READ FORWARD X RECORDS

E. SW0=0,SW1=1,SW2=0,SW3=0
WRITE THEN BACKSPACE AND READ FORWARD THEN REVERSE

F. SW0=1,SW1=0,SW2=1,SW3=0
READ TAPE FORWARD X RECORDS

G. SW0=1,SW1=0,SW2=0,SW3=1
READ TAPE REVERSE X RECORDS

H. SW0=1,SW1=0,SW2=0,SW3=0
READ TAPE REVERSE THEN FORWARD

I. SW0=1,SW1=1,SW2=0,SW3=0
READ TAPE FORWARD THEN REVERSE

SW4: SWITCH FOUR (4), WHEN SET TO A ONE (1), WILL CAUSE ANY DATA RELATED ERROR TO BE RETRIED. THE WRITE RETRY SCHEME CONSISTS OF REWRITING THE RECORD IN THE SAME SPOT ON TAPE FOUR (4) TIMES. IF ALL FOUR (4) REPEATS ARE SUCCESSFUL, THE RECORD IS CONSIDERED AS RECOVERED, AND A TAPE WRITE ERROR IS LOGGED. IF ANY OF THE FOUR (4) REPEATS IS UNSUCESSFUL, A SKIP ERASE IS DONE, A SUPECTED BAD TAPE SPOT IS 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 SLAVE WILL BE REWOUND AND REMOVED FROM TESTING WITH AN APPROPRIATE MESSAGE PRINTED.
THE READ RETRY SCHEME CONSISTS OF REREADING THE RECORD UP TO EIGHT TIMES. IF ALL EIGHT REREADS ARE BAD, IT IS A HARD ERROR. IF ANY REREAD IS SUCCESSFUL, THIS IS A SOFT ERROR.
IF THE ORIGINAL ERROR IS OF THE NON-RETRYABLE TYPE (IE: ILF,RMR,ILR,NEF,CBUSPE), THE RETRY SCHEME IS NOT ENTERED AND A MESSAGE IS PRINTED.

SW5: SWITCH FIVE (5) WHEN SET DURING A READ FORWARD OR REVERSE WILL CAUSE THE TAPE TO CONTINUOUSLY READ THE CURRENT RECORD BY SPACING EITHER FORWARD OR REVERSE 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 3000 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 (FORWARD OR REVERSE) 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.

**NOTE THAT WHEN SW11 OR 12 ARE SET, NOT ONLY ARE ERRORS NOT CHECKED, BUT THEY ARE NOT CLEARED EITHER.
***THEREFOR USE CAUTION TO ASSURE THAT OPERATIONS ARE NOT UNEXECUTED DUE TO UNCLEARED ERRORS.
****DO NOT SET SW 11 OR 12 TO A ONE (1), DURING A RETRY SEQUENCE.

SW14: SWITCH FOURTEEN (14)WHEN SET TO A ONE (1) WILL PRINT THE ACCUMULATED READ/WRITE STATISTICS FOR THE SELECTED SLAVE UNDER TEST AT THE END OF THE CURRENT BLOCK CYCLE. THE STATISTICS PRINTED ARE THE NUMBER OF BITS DROPPED OR PICKED, THE NUMBER OF RETRIES, WRITE ERRORS, READ ERRORS, AND DATA ERRORS.

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 PROCEEDED BY A TWO LINE HEADER WHICH CONTAINS THE DRIVE NUMBER, SLAVE NUMBER, DENSITY, PARITY, AND FORMAT ON THE FIRST LINE, AND THE BLOCK NUMBER, RECORD NUMBER, RECORD SIZE, AND ERROR TYPE ON THE SECOND.

A. OPERATION ERRORS:

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

1. READ/WRITE STATUS ERRORS: THESE ARE DETECTED BY EITHER THE TM03 ITSELF OR BY THE MASSBUS CONTROLLER. ALL STATUS ERRORS WILL BE REPORTED.

2. TAPE POSITION ERRORS: THESE ARE INDICATED BY AN INCORRECT SPACE OR REWIND OPERATION IN WHICH TAPE POSITION BECOMES UNRELIABLE.

B. DATA ERRORS:

DATA ERRORS WILL OCCUR WHEN TAPE IS BEING READ AND THE DATA FROM TAPE DOES NOT MATCH THE EXPECTED DATA. WHEN READING IN THE REVERSE DIRECTION, THE RECORD NUMBERS WILL BE COUNTED DOWN FROM LAST TO FIRST. THE CHARACTER NUMBERS IN REVERSE READS WILL ALSO BE COUNTED DOWN IN ORDER TO REFLECT TAPE POSITION RATHER THAN THE ORDER TRANSFERRED.

BECAUSE DATA RECORDS CAN BE UP TO FOUR 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 FIVE (25) 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: (CATASTROPHIC)

THESE PRINTOUTS REFLECT THE STATE OF THE TAPE SYSTEM
EITHER BEFORE OR AFTER AN OPERATION

1. EOT: WHEN EOT (END OF TAPE) IS ENCOUNTERED DURING
EITHER A READ OR WRITE, THE CYCLE IS COMPLETED
ON THE SHORTENED BLOCK AFTER WHICH THE SLAVE
WILL BE REWOUND AND FLAGGED AS UNAVAILABLE
FOR TESTING UNTIL ALL SLAVES HAVE REACHED EOT AND
ARE REWOUND. WHEN THE LAST AVAILABLE SLAVE
HAS REACHED EOT AND BEEN REWOUND TO BOT,
TESTING WILL BE RESUMED ON ALL SLAVES.

2. ILLEGAL BOT: WHEN A SLAVE ENCOUNTERS BOT DURING
A READ, WRITE, OR SPACE OPERATION, AN ERROR
IS PRINTED AND THE PROGRAM HALTED. THIS IS
A CATASTROPHIC ERROR. TESTING MAY BE RESUMED
BY PRESSING CONTINUE; BUT A RESTART IS
SUGGESTED.

3. NO INTERRUPT RETURNED: EACH TAPE OPERATION SHOULD BE
TERMINATED BY THE SETTING OF AN INTERRUPT IN
THE CPU. IF NO INTERRUPT IS RETURNED WITHIN
THE APPROPIATE TIME, AN ERROR IS PRINTED.

4. NO MEDIUM ON-LINE: BEFORE AN OPERATION IS ATTEMPTED,
THE TM03 IS CHECKED FOR MOL. IF IT IS NOT
SET, AN ERROR IS PRINTED, AND THE PROGRAM STOPPED.
TESTING MAY BE RESUMED BY PRESSING CONTINUE.

5. NO BOT ON REWIND: AS EACH SLAVE IS REWOUND A CHECK
IS MADE TO ASSURE THAT PROPER POSITION AT BOT
IS ESTABLISHED. IF BOT IS NOT SET UPON COMPLETION OF
A REWIND, AN ERROR IS PRINTED AND THE PROGRAM
WILL HALT. PRESS CONTINUE TO RESUME TESTING.

6. POSITION ERROR: IF POSITION IS LOST DURING A RETRY,
A MESSAGE IS PRINTED, THE TAPE REWOUND,
AND REMOVED FROM TESTING UNTIL ALL ARE
RESTARTED AT BLOCK ONE.

7. BAD TAPE OVERFLOW: IF 20(8) BAD TAPE SPOTS ARE FOUND,
A MESSAGE IS PRINTED, THE TAPE REWOUND,
AND REMOVED FROM TESTING UNITLL ARE
RESTARTED AT BLOCK ONE.

8. HARD READ ERROR: IF ANY HARD READ ERROR IS ENCOUNTERED
DURING A RETRY, A MESSAGE IS PRINTED
REGARDLESS OF THE SETTING OF SW10.

9. NON-RETRYABLE: IF ANY NON-RETRYABLE ERROR IS ENCOUNTERED, A
MESSAGE IS PRINTED REGARDLESS OF THE SETTING OF SW10.

D. EXAMPLES:

GLOSSARY:

BN = CURRENT BLOCK NUMBER
RN = CURRENT RECORD NUMBER
RS = RECORD SIZE, IN FRAMES
WE = WRITE STATUS ERROR
RE = READ STATUS ERROR
SE = SPACE ERROR
TM = TAPE MARK
F = FORWARD
R = REVERSE
CS1 = RH/TE16 CONTROL REGISTER
WC = RH WORD COUNT
BA = RH BUS ADDRESS
FC = TE16 FRAME COUNT
CS2 = RH CONTROLLER STATUS
DS = TE16 DRIVE STATUS
ER = TE16 ERROR REGISTER
AS = ATTENTION SUMMARY
CK = TE16 CHECK CHARACTER
DB = RH DATA BUFFER
MR = TE16 MAINTENENCE REGISTER
DT = TE16 DRIVE TYPE
SN = TE16 SERIAL NUMBER
TC = TE16 TEST CONTROL
*F = DATA FORMAT
*P = PARITY
*D = DENSITY
*PATRN = DATA PATTERN NUMBER (R = RANDOM)

EXAMPLE 1: IN THIS EXAMPLE SLAVE 1 ON TM03 0 WAS
OPERATING AT 1600 BPI IN ODD PARITY USING
THE NINE CHANNEL NORMAL DATA FORMAT. A
WRITE STATUS ERROR WAS DETECTED. THE BAD
STATUS INDICATES THAT AN UNCORRECTABLE DATA
ERROR (BIT 6 OF ER) AND A PE FORMAT ERROR
(BIT 7 OF ER) OCCURED DURING THE WRITE
OPERATION OF THE SIXTH (6) RECORD OF THE
FIFTY (50) RECORDS IN BLOCK (2). THE
SIZE OF THE RECORD WAS TWO HUNDRED (200)
FRAMES. THE CHECK CHARACTER REFLECTS THE BAD TRACK.

```
DRIVE NO. 0 *SLAVE NO. 1 *D 4 *P 0 *F 14 *PATRN 1
*BN 2 *RN 6-50 *RS = 200 *WE
CS1 144260
CS2 100
DS 150640
ER 300
WC 0
CK 4
```

EXAMPLE 2: IN THIS EXAMPLE SLAVE 3 ON TM03 1 WAS
OPERATING AT 800 BPI IN EVEN PARITY USING
THE NINE CHANNEL NORMAL DATA FORMAT. A
READ STATUS ERROR WAS DETECTED DURING
THE REVERSE READ OF THE TENTH (10)
RECORD OF THE 25 RECORDS IN THIS BLOCK (12).
THE SIZE OF THE RECORD IS TWENTY (20) FRAMES.
THE PRINTOUT INDICATES THE DETECTION OF A
VERTICAL PARITY ERROR (VPE: BIT 6 OF ER) AND A
CYCLIC REDUNDENCY ERROR (CRC: BIT 15 OF ER).
THE CRC CHARACTER, AS RECEIVED, IS NOT
AS EXPECTED AND IS PRINTED SHOWING BOTH
THE ACTUAL (FIRST) AND THE EXPECTED (LAST).

```
DRIVE NO. 2 *SLAVE NO. 3 *D 3 *P 1 *F 14 *PATRN 3
*BN 12 *RN 10-25 *RS 20 *RE R
CS1 144276
CS2 100
DS 150600
ER 100100
WC 0
CRC 767-777
```

EXAMPLE 3: IN THIS EXAMPLE, THE HEADER IS THE SAME AS
IN EXAMPLE TWO (2) EXCEPT THAT THE ERROR TYPE
REFLECTS A READ ERROR IN THE FORWARD
DIRECTION. IT IS NORMAL FOR THE SYSTEM
TO DETECT AN ERROR IN THE FORWARD AND
REVERSE DIRECTION AT THE SAME RECORD.
REMEMBER THAT IN REVERSE OPERATIONS THE
RECORD NUMBER IS COUNTED DOWN SO THAT
RECORD NUMBER TEN (10) WILL SHOWN IN
THE PROPER POSITION IN BOTH FORWARD AND
REVERSE.

```
DRIVE NO. 2 *SLAVE NO. 3 *D 3 *P 1 *F 14 *PATRN 2
*BN 12 *RN 10-25 *RS 20 *RE F
CS1 144270
CS2 100
DS 150600
ER 100100
WC 0
CRC 767-777
```

EXAMPLE 4: IN EXAMPLES 2 AND 3 THE READ OPERATION
RESULTED IN BAD STATUS, HOWEVER THE
DATA ASSOCIATED WITH THE OPERATION WAS
NOT BAD (OR WAS NOT CHECKED: SW 13=1).
THIS EXAMPLE (4) SHOWS A PRINTOUT REFLECTING
A READ STATUS ERROR ACCOMPANIED BY BAD
DATA IN CHARACTERS FOUR (4) AND SIX (6).

```
DRIVE NO. 2 *SLAVE NO. 3 *D 3 *P 1 *F 14 *PATRN 2
*BN 12 *RN 10-25 *RS 20 *RE F
CS1 144270
CS2 100
DS 150600
ER 100100
WC 0
CRC 767-777
CN 4
G 11111111
B 10111111
CN 6
G 11111111
B 10111111
```

```
EXAMPLE 5: THIS EXAMPLE SHOWS A READ DATA ERROR
        WHICH OCCURRED, WITHOUT AN ACCOMPANING
        STATUS ERROR, WHICH RESULTED IN A BAD RECORD.

        DRIVE NO. 3 *SLAVE NO. 1 *D 4 *P 0*F 14 *PATRN R
        *BN 100 *RN 66-200 *RS 2000 *DE F
        CN 0
        G 11111111
        B 00000000
        CN 1
        G 11111111
        B 00000000
        CN 2
        G 11111111
        B 00000000
        CN 3
        G 11111111
        B 00000000
        CN 4
        G 11111111
        B 00000000
        CN 5
        G 11111111
        B 00000000
        CN 6
        G 11111111
        B 00000000
        CN 7
        G 11111111
        B 00000000
                BAD RECORD

EXAMPLE 6: THE FOLLOWING EXAMPLE SHOWS THE
        RESULT OF A SPACE OPERATION THAT
        SHOULD HAVE SPACED REVERSE OVER
        AN ENTIRE 100 RECORD BLOCK BUT
        WHICH TERMINATED AT THE END OF 40
        RECORDS.  LEAVING A POSITION ERROR OF 40

        DRIVE NO. 2 *SLAVE NO. 6 *D 2 *P 0 *F 14
        *BN 3 *RN 100-100 *RS 1000 *SE R
        ERR AMT 40
```

EXAMPLE 7: THIS EXAMPLE REFLECTS AN ERROR DETECTED
WHILE WRITING A TAPE MARK (TM) AT THE END
OF THE CURRENT DATA BLOCK PER OPTION RESPONSE
TM=1. NOTE THAT THE TM RECORD NUMBER
IS ONE GREATER THAN THE TOTAL NUMBER OF DATA
RECORDS IN THE CURRENT BLOCK.

```
DRIVE NO. 1 *SLAVE NO. 1 *D 2 *P 0 *F 14
*BN 67 *RN 101-100 *RS 36 *WE TM
CS1 144226
CS2 300
DS 150604
ER 1000
WC 0
```

EXAMPLE 8: THIS EXAMPLE SHOWS TWO (2) PRINTOUTS
REFLECTING A WRITE RETRY WHICH WAS NOT
SUCCESSFUL THE FIRST TIME, BUT WHICH
DID RECOVER ON THE SECOND.
THE UNSUCCESSFUL RETRY IS LOGGED AS
A SUSPECTED BAD TAPE SPOT BY ITS
BLOCK AND RECORD NUMBER.

```
DRIVE NO. 0 *SLAVE NO. 2 *D 4 *P 0 *F 14 *PATRN 6
*BN 2 *RN 12-20 *RS 667 *WE
CS1 144260
CS2 100
DS 150640
ER 100
WC 0
***ORIGINAL ERROR***

DRIVE NO. 0 SLAVE NO. 2 *D 4 *P 0 *F 14 *PATRN 6
*BN 2 *RN 12-20 *RS 667 *WE
CS1 144260
CS2 100
DS 150640
ER 100
WC 0
SUSPECT BAD TAPE
RETRY: 0
REPT: 0
RECOVERED
RETRY: 1
```

EXAMPLE 9: IF , DURING A WRITE RETRY THE BACKSPACE OR THE ERASE OPERATION RESULT IN AN ERROR, THE ERROR WILL BE PRINTED AND THE PROGRAM HALTED. THIS EXAMPLE SHOWS THE ERROR PRINT FOR A SPACE AND AN ERASE (2 EXAMPLES)

```
DRIVE NO. 1 *SLAVE NO. 1 *D 3 *P 0 *F 14
BN 12 *RN 8-64 *RS 500 *SE RTRY
ERR AMT 1

DRIVE NO. 1 *SLAVE NO. 1 *D 3 *P 0 *F 14
*BN 12 *RN 8-64 *RS 500 *ERASE
CS1 144224
CS2 100
DS 150600
ER 400
WC 0
```

EXAMPLE 10: THIS EXAMPLE SHOWS THE PRINTOUT FROM A REWIND OPERATION WHICH DOES NOT HAVE BOT SET AT THE END.

```
DRIVE NO. 2 *SLAVE NO. 3 *D 3 *P 0 *F 14
*BN 66 *RN 15-20 *RS 1000
NOT BOT ON REWIND: HALT
```

EXAMPLE 11: THIS EXAMPLE SHOWS THE PRINTOUT MADE WHEN THERE IS NO INTERRUPT RETURNED AT THE END OF AN OPERATION.

```
DRIVE NO. 7 *SLAVE NO. 7 *D 2 *P 1 *F 14
*BN 1 *RN 25-26 *RS 1200
NO INTERRUPT
```

10. STATISTICS PRINTOUT

THE PROGRAM, THROUGH ITS ERROR CHECKING, IS ABLE TO GATHER CERTAIN
STATISTICS ABOUT THE PERFORMANCE OF EACH UNIT UNDER TEST.
THIS INFORMATION IS PRINTED OUT WHENEVER A UNIT IS REWOUND
FROM END OF TAPE, OR BECAUSE IT IS TO BE REMOVED FROM TESTING
DUE TO SOME CATASTROPHIC ERROR. (POSITION LOST, BAD TAPE OVERFLOW)
THE STATISTICS MAY BE PRINTED AT ANY TIME BY SETTING
SWITCH 14 TO A ONE (1). THIS PRESENTS A PICTURE OF PERFORMANCE
UP TO THIS TIME. THE STATISTICS WILL BE CLEARED UPON REWIND OF THE
UNIT; BUT NOT BY SETTING SW 14.

STATISTICS PRINT EXAMPLE (A HEADER WILL PRECEED THE STATS)

```
DROPS: 0 3 0 0 0 6 45 0
PICKS: 1 0 0 0 0 0 0 2
RETRY: 1
WTERR: 2
REFWD: 3
 SOFT: 2
 HARD: 1
DEFWD: 0
REREV: 4
 SOFT: 1
 HARD: 3
DEREV: 0
2 BAD TAPE SPOTS
0 *BN 1 *RN 2
1 *BN 15 *RN 100

** NOTE ** DROPS AND PICKS REFLECT CORE BIT POSITIONS.
          THE FOLLOWING IS A TABLE OF CORE BITS TO TRACK NUMBER.

          TRACK NO. 7 6 5 3 9 1 8 2
          CORE BIT  7 6 5 4 3 2 1 0

DROPS: NUMBER OF DATA BITS DROPPED: PER CORE BIT(SEE NOTE ABOVE)
PICKS: NUMBER OF DATA BITS PICKED UP: PER CORE BIT(SEE NOTE ABOVE)
RETRY: NUMBER OF WRITE RETRIES
WTERR: NUMBER OF WRITE ERRORS NOT ASSOCIATED WITH BAD TAPE
REFWD: NUMBER OF READ FORWARD STATUS ERRORS
REREV: NUMBER OF READ REVERSE STATUS ERRORS
 SOFT: NUMBER OF RECOVERED READ ERRORS
 HARD: NUMBER OF UNRECOVERED READ ERRORS
DEFWD: NUMBER OF FORWARD DATA ERRORS WITH NO ASSOCIATED STATUS ERROR
DEREV: NUMBER OF REVERSE DATA ERRORS WITH NO ASSOCIATED STATUS ERROR
```

11. AUTO SEQUENCE

THE AUTO SEQUENCE (START AT ADDRESS 240) WILL EXECUTE A PREDETERMINED TEST PLAN ON ALL AVAILABLE SLAVES ON EACH AVAILABLE TM03. THE ONLY OPERATOR RESPONSE IS TO THE TYPED REQUESTS FOR THE RH ADDRESS, VECTOR, CONTINUOUS OR SINGLE CYCLE, AND NRZ ONLY. ALL SWITCHES REMAIN ACTIVE AND MAY BE USED NORMALLY; HOWEVER THE IDEA 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:

```
        TE16 AUTO SEQUENCE TEST
        ENTER CONDITIONS IN OCTAL

        REGISTER START = 172400(172440)
        VECTOR ADDRESS = 224(CR)
        NRZ ONLY: (0)
        AUTO CONT: (1)
```

THIS EXAMPLE SHOWS AN AUTO SEQUENCE START WITH THE RH AT BUS ADDRESS 172440 AND A VECTOR OF 224. ALL AVAILABLE HARDWARE WILL BE TESTED CONTINUOUSLY IN BOTH NRZ AND PE MODE.

AS EACH TM03 AND ITS SLAVES ARE FOUND, A DIVIDER LINE OF ASTERICKS WILL BE PRINTED FOLLOWED BY A PRINTOUT OF THE TM03 AND ITS SLAVES BEING TESTED. AS EACH TM03 AND ITS SLAVES ARE FINISHED, ANOTHER DIVIDER IS PRINTED BEFORE TESTING IS RESUMED ON THE NEXT AVAILABLE DRIVE.

WHEN ALL AVAILABLE HARDWARE HAS BEEN TESTED, A PRINTOUT OF END OF SEQUENCE WILL BE DONE AND THE PROGRAM WILL EITHER HALT (AUTO CONT = 0) OR RESTART WITH THE FIRST AVAILABLE UNIT (AUTO CONT = 1).

AUTO SEQUENCE TEST PLAN:

THE AUTO SEQUENCE WILL EXECUTE BOTH AN NRZ AND A PE CYCLE. EACH CYCLE WILL BE STARTED FROM BOT AND CONSIST OF VARIOUS DATA PATTERNS INTENDED TO BE WORST CASE FOR THAT PARTICULAR MODE.

1. NRZ CYCLE:

SIX (6) BLOCKS OF ONE HUNDRED (100) RECORDS OF FOUR THOUSAND (4000) CHARACTERS FOR EACH OF THE FOUR DATA PATTERNS.

PATTERN 1: ALL ONES DATA IN ALL BYTES
PATTERN 10: WALKING ONE/ALL ONE
PATTERN 14: WALKING ZERO/ALL ZERO
RANDOM DATA: RANDOM

2. PE CYCLE: (IF NRZ ONLY = 0)

SIX BLOCKS OF ONE HUNDRED (100) RECORDS OF FOUR THOUSAND (4000) CHARACTERS EACH FOR EACH OF THREE DATA PATTERNS, THEN RANDOM DATA BLOCKS TO END OF TAPE.

PATTERN 10: WALKING ONE/ALL ONE
PATTERN 14: WALKING ZERO/ALL ZERO
PATTERN 15: THREE (3) 0 CHARACTERS, TWO (2) ALL CHARACTERS, THREE 0 CHARACTERS, THEN COMPLIMENT PATTERN. REPEATED FOR A FULL BUFFER
RANDOM DATA: RANDOM

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 WILL 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.

```
%
1308                                    .LIST   BIN,LOC,SEQ
1309                                    .TITLE TM03/TE16 DATA RELIABILITY PROGRAM
1310                                    ;MAINDEC-11-DZTED-A-D
1311                                    ;21 FEB 1977
1312                                    ;R. BARNES
1313                                    .MCALL  .$ACT11,.$EOP,$SAVE,$RESTORE,$CHAIN
1314                                    .NLIST  MC
1315                                    .LIST   ME
```

```
1316                                    .ENABLE ABS,AMA
1317
1318                                    ;CONSOLE SWITCHES************
1319
1320                                    ;SW15: 1=STOP ON ERROR
1321                                    ;       0=CONTINUE ON ERROR
1322                                    ;SW14: 1=PRINT READ/WRITE STATS
1323                                    ;       0=DO NOT PRINT STATS
1324                                    ;SW13: 1=DO NOT CHECK DATA
1325                                    ;       0=CHECK DATA
1326                                    ;SW12: 1=DO NOT CHECK WRITE ERRORS
1327                                    ;       0=CHECK WRITE ERRORS
1328                                    ;SW11: 1=DO NOT CHECK READ ERRORS
1329                                    ;       0=CHECK READ ERRORS
1330                                    ;SW10: 1=DO NOT PRINT ERRORS
1331                                    ;       0=PRINT ERRORS
1332                                    ;SW9:  1=REWIND TAPE
1333                                    ;       0=DO NOT REWIND
1334                                    ;SW8:  1=USE RANDOM DATA
1335                                    ;       0=USE FIXED DATA PATTERN
1336                                    ;SW7:  1=USE RANDOM CHARACTER COUNT
1337                                    ;       0=USE FIXED CHAR COUNT
1338                                    ;SW6:  1=USE RANDOM RECORD COUNT
1339                                    ;       0=USE FIXED RECORD COUNT
1340                                    ;SW5:  1=YOZZLE ON CURRENT RECORD
1341                                    ;       0=DO NOT YOZZLE
1342                                    ;SW4:  1=DO BOTH READ AND WRITE RETRIES
1343                                    ;       0=INHIBIT RETRIES
1344                                    ;SW3:  1=DO NOT READ FORWARD
1345                                    ;       0=READ FORWARD
1346                                    ;SW2:  1=DO NOT READ REVERSE
1347                                    ;       0=READ REVERSE
1348                                    ;SW1:  1=READ FORWARD FIRST
1349                                    ;       0=READ REVERSE FIRST
1350                                    ;SW0:  1=DO NOT WRITE
1351                                    ;       0=WRITE
1352                                ;IF SWR <15::00> = 177777 OR NOT AVAILABLE USE SOFTWARE SWITCH REGISTER
```

```
1353                                    ;TU16 REGISTER BITS**********
1354
1355                                    ;15 ;14 ;13 ;12 ;11 ;10 ;09 ;08 ;07 ;06 ;05 ;04 ;03 ;02 ;01 ;00 ;
1356                                    ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1357                                ;CS1:;SC ;TRE;MCP;SPR;DNA;PSL;A17;A16;RDY;IE ;FUN;FUN;FUN;FUN;FUN;GO ;
1358                                    ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;
1359                                    ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1360                                ;WC: ; WORD                                                        ;
1361                                    ; COUNT                                                       ;
1362                                    ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1363                                ;BA: ; BUS                                                         ;
1364                                    ; ADDRESS                                                     ;
1365                                    ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1366                                ;FC: ; FRAME                                                       ;
1367                                    ; COUNT                                                       ;
1368                                    ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1369                                ;CS2:;DLT;WCE;UPE;NED;NEM;PGE;MXF;MDP;OR ;IR ;CLR;PAT;BAI;U2 ;U1 ;U0 ;
1370                                    ;   ;   ;   ;   ;   ;   ;   ;ERR;   ;   ;   ;   ;   ;   ;   ;
1371                                    ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1372                                ;DS: ;ATA;ERR;PIP;MOL;WRL;EOT;SPR;DPR;DRY;SSC;PES;SDN;IDB;EOF;BOT;SLA;
1373                                    ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;
1374                                    ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1375                                ;ER: ;CDE;UNS;OPI;DTE;NEF;CS ;FCE;NSG;PEF;INC;DAT;FMT;CNT;RMR;ILR;ILF;
1376                                    ;CRC;   ;   ;   ;   ;ITM;   ;   ;LRC;VPE;BPE;   ;BPE;   ;   ;   ;
1377                                    ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1378                                ;ATS: ; ATTENTION                                                  ;
1379                                    ; SUMMARY                                                     ;
1380                                    ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1381                                ;CC: ; CHECK                                                       ;
1382                                    ; CHARACTER                                                   ;
1383                                    ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1384                                ;DB: ; DATA                                                        ;
1385                                    ; BUFFER                                                      ;
1386                                    ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1387                                ;MR: ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;WRT;MM ;OP ;OP ;OP ;OP ;MM ;
1388                                    ; 7 ; 6 ; 5 ; 4 ; 3 ; 2 ; 1 ; 0 ; P ;CLK;CLK; 4 ; 3 ; 2 ; 1 ;GO ;
1389                                    ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1390                                ;DT: ; DRIVE     ; 7 ;   ;SLV;                                    ;
1391                                    ; TYPE      ;CH ;   ;PR ;                                    ;
1392                                    ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1393                                ;SN: ; SERIAL                                                      ;
1394                                    ; NUMBER                                                      ;
1395                                    ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1396                                ;TC: ;ACL;FCS;TCW;ENA;SPR;DEN;DEN;DEN;FMT;FMT;FMT;FMT;EVN;SSN;SSN;SSN;
1397                                    ;   ;   ;DT ;   ;   ; 4 ; 2 ; 1 ; 8 ; 4 ; 2 ; 1 ;PAR; 4 ; 2 ; 1 ;
1398                                    ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

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

000020                                  .=20
000020  023634                          .WORD   TTOUT           ;SET IOT TRAP TO TTOUT ROUTINE
000022  000340                          .WORD   340             ;PRIORITY LEVEL 7

000004                                  TYPE=IOT                ;EQUATE TYPE TO AN IOT INSTRUCTION
000034                                  .=34
000034  024006                          .WORD   OCTP            ;SET TRAP TRAP TO OCTP ROUTINE
000036  000340                          .WORD   340
104400                                  TYPOCT=TRAP             ;EQUATE TYPOCT TO TRAP INSTRUCTION

                                ;ACT11 HOOK ***********
000040                                  $SVPC=.                 ;SAVE CURRENT LOCATION CTR
000046                                  .=46
000046  005022                          .WORD   $ENDAD          ;SET LOCATION 46
000052                                  .=52
000052  000000                          .WORD   0               ;SET LOCATION 52 = 0
000040                                  .=$SVPC                 ;RESTORE LOCATION CTR

                                        ;TTY INTERRUPT VECTOR**********
000060                                  .=60
000060  021466                          .WORD   TTINT           ;TTY INTERRUPT HANDLER ADDRESS
000062  000340                          .WORD   340             ;PRIORITY LEVEL 7

                                        ;SOFTWARE SWITCH REGISTER*********
                                        ;INVOKED IF SWR <15::00> = 177777 OR NOT AVAILABLE
000176                                  .=176
000176  000000                  SWREG:  .WORD   0

                                        ;START ADDRESS**********
000200                                  .=200
000200  000137  003026                  JMP     START           ;ENTER PARAMETERS VIA TTY

000204                                  .=204
000204  000137  003152                  JMP     STARTC          ;USE FIXED PARAMETERS; HOLD DATA

000210                                  .=210
000210  005037  015022                  CLR     RDFL
000214  000137  003160                  JMP     STARTA          ;USE FIXED PARAMETERS; NEW DATA

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

000224                                  .=224
```

```
1455  000224  021716                  MTINT                   ;MAG TAPE INTERRUPT HANDLER ADDRESS
1456  000226  000340                  340
1457
1458                                  ;AUTO SEQUENCE START**********
1459
1460          000240                  .=240
1461  000240  005237  000736          INC     ASEQF           ;SET AUTO SEQUENCE FLAG
1462  000244  000137  003136          JMP     STAUT           ;GO TO START OF AUTO SEQUENCE
```

```
                                                ;SHORT CONVERSATION RESTART********

                    000300                      .=300
000300  005237  014062                          INC     SCVFL                   ;SET SHORT CONVERSATION FLAG
000304  000137  003026                          JMP     START                   ;ENTER SHORT PARAMETER LIST

                    000510                      .=510
                                                ;TU16 REGISTER EQUIVS**********

000510  172440                          C1:     172440
000512  172442                          MC:     172442
000514  172444                          BA:     172444
000516  172446                          FC:     172446
000520  172450                          CS:     172450
000522  172452                          DS:     172452
000524  172454                          ER:     172454
000526  172456                          AS:     172456
000530  172460                          CC:     172460
000532  172462                          DB:     172462
000534  172464                          MR:     172464
000536  172466                          DT:     172466
000540  172470                          SN:     172470
000542  172472                          TC:     172472

                                                ;CONSTANTS**********

000544  172440                          REGS:   172440          ;STARTING REGISTER ADDRESS (CS1)
000546  000224                          VECT:   224             ;VECTOR ADDRESS (RH INTERRUPT)
000550  000000                          DVN:    0               ;DRIVE NUMBER
000552  000000                          UDES:   0               ;UNIT DESCRIPTION (PARITY,DENSITY,UNIT,FORMAT)
000554  000100                          RCNT:   100             ;RECORD COUNTER
000556  177400                          FMCNT:  177400          ;NUMBER OF CHAR (4 - 4000) OCTAL IN TWOS COMPLEMENT
000560  000001                          PATRN:  1               ;DATA PATTERN SELECTOR (0 - 15) OCTAL
000562  000002                          RDCMD:  2               ;READ COMMAND
000564  000001                          TMEX:   1               ;TAPE MARK FLAG: 1=TM 0=NO TM
000566  000000                          CRCC:   0               ;CRC CORRECTION FLAG (YES=1,NO=0)
000570  000000                          INTRF:  0               ;INTERCHANGE READ 1=YES 0=NO
000572  000000                          SPFLG:  0               ;SINGLE PASS 1=YES 0=NO
000574  000001                          RSTAL:  1               ;READ STALL
000576  000001                          WSTAL:  1               ;WRITE STALL
000600  000001                          TSTAL:  1               ;TURN AROUND STAL
000602  002000                          YSTAL:  2000            ;YOZZLE STAL
000604  000010                          RETRY:  10              ;READ RETRY NUMBER
000606  177776                          PSW:    177776          ;PROCESSOR STATUS
000610  177570                          SWR:    177570          ;CONSOLE SWITCHES
000612  177560                          TKS:    177560          ;TTY READ STATUS REGISTER
000614  177562                          TKB:    177562          ;TTY READ BUFFER
000616  177564                          TPS:    177564          ;TTY PUNCH STATUS REGISTER
000620  177566                          TPB:    177566          ;TTY PUNCH OUTPUT REGISTER
000622  177550                          PRS:    177550          ;H/S READER STATUS REGISTER
000624  177552                          PRB:    177552          ;H/S READER BUFFER
000626  153624                          RANBAS: 153624          ;RANDOM NUMBER GENERATOR BASE
000630  032561                          RANSAV: 032561          ;RANDOM NUMBER BUFFER
000632  000100                          RCSAV:  100             ;RECORD COUNT SAVE
000634  177400                          FCSAV:  177400          ;FRAME COUNT SAVE
```

```
1518
1519                                        ;FLAGS AND COUNTERS**********
1520
1521  000636  000000    TINF:   0           ;TTY ENTRY FLAG
1522  000640            STFLG:
1523  000640  000000    TOB:    0           ;TTY OUTPUT BUFFER
1524  000642  000000    TIB:    0           ;TTY INPUT BUFFER
1525  000644  000000    TEMP1:  0           ;TEMP STORAGE
1526  000646  000000    TEMP2:  0           ;TEMP STORAGE
1527  000650  000000    TEMP3:  0           ;TEMP STORAGE
1528  000652  000000    NRZOF:  0           ;NRZ ONLY FLAG
1529  000654  000000    EMADDR: 0           ;ERROR MSG ADDRESS STORAGE
1530  000656  000000    BLCNTR: 0           ;BLOCK COUNTER
1531  000660  000000    BBC:    0           ;BAD RECORD COUNTER
1532  000662  000000    EOTREC: 0           ;EOT FLAG
1533  000664  000000    RTRN:   0           ;INTERRUUPT RETURN STORAGE
1534  000666  000000    HDRFL:  0           ;HEADER FLAG
1535  000670  000000    STAL:   0           ;DELAY STORAGE
1536  000672  000000    PFLG:   0           ;PRINT FLAG
1537  000674  000000    MTC1:   0           ;MAG TAPE CONT REGISTER BUFFER
1538  000676  000000    UNP:    0           ;UNIT TABLE POINTER
1539  000700  000000    TMFLG:  0           ;TAPE MARK FLAG
1540  000702  000000    RPCNT:  0           ;REPEAT COUNTER
1541  000704  000000    RTCNT:  0           ;RETRY COUNTER
1542  000706  000000    DERFL:  0           ;DATA ERROR FLAG
1543  000710  000000    SERFL:  0           ;STATUS ERROR FLAG
1544  000712  000000    BCNT:   0           ;BIT COUNTER
1545  000714  000000    RTYFL:  0           ;RETRY FLAG
1546  000716  000000    UPS:    0           ;UNIT POINTER SAVE
1547  000720  000000    BDPP:   0           ;BITS DROPPED POINTER
1548  000722  000000    BPKP:   0           ;BITS PICKED POINTER
1549  000724  000000    ERSAV:  0           ;ERROR SAVE LOC
1550  000726  000000    BTFLG:  0           ;BAD TAPE FLAG
1551  000730  000000    BTSTF:  0           ;STATISTIC PRINT FLAG
1552  000732  000000    BTPT:   0           ;BAD TAPE POINTER
1553  000734  000000    ERTFL:  0           ;ERASE FLAG
1554  000736            ENDFLG:
1555  000736  000000    ASEQF:  0           ;AUTO SEQ FLAG
1556  000740  000000    ADRVN:  0           ;UTO SEQ DRIVE NUMBER
1557  000742  000000    ABLCNT: 0           ;AUTO BLOCK COUNTER
1558  000744  000001    ASEQCF: 1           ;AUTO SEQ CONTINUOUS FLAG
```

```
1559
1560                                        ;UNIT ORDER AND DESCRIPTION TABLE **********
1561
1562  000746  000000        UN1:    0             ;THIS TABLE IS LOADED
1563  000750  000000        UN2:    0             ;WITH UNIT NUMBERS AND
1564  000752  000000        UN3:    0             ;THEIR DESCRIPTIONS IN
1565  000754  000000        UN4:    0             ;THE ORDER THAT THEY
1566  000756  000000        UN5:    0             ;WILL BE TESTED
1567  000760  000000        UN6:    0
1568  000762  000000        UN7:    0
1569  000764  000000        UN8:    0
1570  000766  177777        UNX:    -1
1571
1572                                        ;UNIT DROPS AND PICKS POINTERS*********
1573
1574  000770  001210        PIK1:   BP00
1575  000772  001230        PIK2:   BP10
1576  000774  001250        PIK3:   BP20
1577  000776  001270        PIK4:   BP30
1578  001000  001310        PIK5:   BP40
1579  001002  001330        PIK6:   BP50
1580  001004  001350        PIK7:   BP60
1581  001006  001370        PIK8:   BP70
1582  001010  001410        DRP1:   B000
1583  001012  001430        DRP2:   B010
1584  001014  001450        DRP3:   B020
1585  001016  001470        DRP4:   B030
1586  001020  001510        DRP5:   B040
1587  001022  001530        DRP6:   B050
1588  001024  001550        DRP7:   B060
1589  001026  001570        DRP8:   B070
1590
1591                                        ;UNIT BAD TAPE POINTERS**********
1592
1593  001030  001610        BTADDR: BT00
1594  001032  001714                BT01
1595  001034  002020                BT02
1596  001036  002124                BT03
1597  001040  002230                BT04
1598  001042  002334                BT05
1599  001044  002440                BT06
1600  001046  002544                BT07
1601
1602                                        ;UNIT WRITE RETRY COUNTER**********
1603
1604                                ;SET START OF STATISTICS TABLE
1605  001050                        STTBL:
1606  001050  000000        RTY1:   0
1607  001052  000000        RTY2:   0
1608  001054  000000        RTY3:   0
1609  001056  000000        RTY4:   0
1610  001060  000000        RTY5:   0
1611  001062  000000        RTY6:   0
1612  001064  000000        RTY7:   0
1613  001066  000000        RTY8:   0
1614
```

```
1615                                            ;UNIT WRITE ERRORS**********
1616
1617  001070  000000                WTER1:  0
1618  001072  000000                WTER2:  0
1619  001074  000000                WTER3:  0
1620  001076  000000                WTER4:  0
1621  001100  000000                WTER5:  0
1622  001102  000000                WTER6:  0
1623  001104  000000                WTER7:  0
1624  001106  000000                WTER8:  0
1625
1626  ;                             ;UNIT READ FORWARD ERRORS**********
1627
1628  001110  000000                RDER1:  0
1629  001112  000000                RDER2:  0
1630  001114  000000                RDER3:  0
1631  001116  000000                RDER4:  0
1632  001120  000000                RDER5:  0
1633  001122  000000                RDER6:  0
1634  001124  000000                RDER7:  0
1635  001126  000000                RDER8:  0
1636
1637                                            ;UNIT DATA ERRORS FORWARD**********
1638
1639  001130  000000                DATER1: 0
1640  001132  000000                        0
1641  001134  000000                        0
1642  001136  000000                        0
1643  001140  000000                        0
1644  001142  000000                        0
1645  001144  000000                        0
1646  001146  000000                        0
1647
1648                                            ;UNIT READ REVERSE ERRORS**********
1649
1650  001150  000000                RDERR1: 0
1651  001152  000000                        0
1652  001154  000000                        0
1653  001156  000000                        0
1654  001160  000000                        0
1655  001162  000000                        0
1656  001164  000000                        0
1657  001166  000000                        0
1658
1659                                            ;UNIT DATA ERRORS REVERSE**********
1660
1661  001170  000000                DEREV1: 0
1662  001172  000000                        0
1663  001174  000000                        0
1664  001176  000000                        0
1665  001200  000000                        0
1666  001202  000000                        0
1667  001204  000000                        0
1668  001206  000000                        0
```

```
1669                                        ;DROPS + PICKS PER CHANNEL PER UNIT**********
1670
1671  001210  000000                BP00:   0
1672          001230                        .=.+16
1673  001230  000000                BP10:   0
1674          001250                        .=.+16
1675  001250  000000                BP20:   0
1676          001270                        .=.+16
1677  001270  000000                BP30:   0
1678          001310                        .=.+16
1679  001310  000000                BP40:   0
1680          001330                        .=.+16
1681  001330  000000                BP50:   0
1682          001350                        .=.+16
1683  001350  000000                BP60:   0
1684          001370                        .=.+16
1685  001370  000000                BP70:   0
1686          001410                        .=.+16
1687  001410  000000                B000:   0
1688          001430                        .=.+16
1689  001430  000000                B010:   0
1690          001450                        .=.+16
1691  001450  000000                B020:   0
1692          001470                        .=.+16
1693  001470  000000                B030:   0
1694          001510                        .=.+16
1695  001510  000000                B040:   0
1696          001530                        .=.+16
1697  001530  000000                B050:   0
1698          001550                        .=.+16
1699  001550  000000                B060:   0
1700          001570                        .=.+16
1701  001570  000000                B070:   0
1702          001610                        .=.+16
1703
1704
```

```
1705
1706                                        ;UNIT BAD TAPE COUNTER:16 PER SLAVE**********
1707
1708  001610  000000                BTO0:   0
1709          001714                        .=.+102
1710  001714  000000                BTO1:   0
1711          002020                        .=.+102
1712  002020  000000                BTO2:   0
1713          002124                        .=.+102
1714  002124  000000                BTO3:   0
1715          002230                        .=.+102
1716  002230  000000                BTO4:   0
1717          002334                        .=.+102
1718  002334  000000                BTO5:   0
1719          002440                        .=.+102
1720  002440  000000                BTO6:   0
1721          002544                        .=.+102
1722  002544  000000                BTO7:   0
1723          002650                        .=.+102
1724
1725                                        ;UNIT END OF TAPE COUNTERS 1 PER SLAVE*******
1726
1727  002650  000000                EOTCO:  0
1728  002652  000000                        0
1729  002654  000000                        0
1730  002656  000000                        0
1731  002660  000000                        0
1732  002662  000000                        0
1733  002664  000000                        0
1734  002666  000000                        0
1735
1736                                        ;UNIT READ FORWARD SOFT ERROR*******
1737
1738  002670  000000                RFSOFT: 0
1739  002672  000000                        0
1740  002674  000000                        0
1741  002676  000000                        0
1742  002700  000000                        0
1743  002702  000000                        0
1744  002704  000000                        0
1745  002706  000000                        0
1746
1747                                        ;UNIT READ REVERSE SOFT ERROR*******
1748
1749  002710  000000                RRSOFT: 0
1750  002712  000000                        0
1751  002714  000000                        0
1752  002716  000000                        0
1753  002720  000000                        0
1754  002722  000000                        0
1755  002724  000000                        0
1756  002726  000000                        0
1757
```

```
1758
1759                                        ;UNIT READ FORWARD HARD ERROR*******
1760
1761  002730  000000                RFHARD: 0
1762  002732  000000                        0
1763  002734  000000                        0
1764  002736  000000                        0
1765  002740  000000                        0
1766  002742  000000                        0
1767  002744  000000                        0
1768  002746  000000                        0
1769
1770                                        ;UNIT READ REVERSE HARD ERROR*******
1771
1772  002750  000000                RRHARD: 0
1773  002752  000000                        0
1774  002754  000000                        0
1775  002756  000000                        0
1776  002760  000000                        0
1777  002762  000000                        0
1778  002764  000000                        0
1779  002766  000000                        0
1780                                ;SET END OF STATISTICS TABLE
1781  002770                        ENDTBL:
1782
1783                                        ;DATA PATTERN GENERATORS************
1784
1785  002770  002770                DATBL:  .                       ;ENTRY TABLE
1786  002772  014274                DATA0:  DAT0                    ;EXTERNAL INPUT FROM H/S READER(SEE MAINDEC-11-DZTUF)
1787  002774  014434                DATA1:  DAT1                    ;ALL ONES
1788  002776  014454                DATA2:  DAT2                    ;ALL ZEROS
1789  003000  014460                DATA3:  DAT3                    ;WALKING ONE
1790  003002  014504                DATA4:  DAT4                    ;WALKING ZERO
1791  003004  014514                DATA5:  DAT5                    ;ALTERNATING ONE/ZERO
1792  003006  014522                DATA6:  DAT6                    ;ALTERNATING ZERO/ONE
1793  003010  014530                DATA7:  DAT7                    ;ALTERNATING ONE/ZERO IN ALTERNATING CHARACTERS
1794  003012  014556                DATA10: DAT10                   ;WALKING ONE/ALL ONE IN ALTERNATING CHARACTERS
1795  003014  014606                DATA11: DAT11                   ;ALL BITS 0-377
1796  003016  014626                DATA12: DAT12                   ;ALL BITS 377-0
1797  003020  014650                DATA13: DAT13                   ;ALTERNATING CHARACTERS  0 AND 377
1798  003022  014660                DATA14: DAT14                   ;WALKING ZERO/ALL ZERO IN ALTERNATING CHARACTERS
1799  003024  014710                DATA15: DAT15                   ;AUTO SEQUENCE PATTERN 0,0,-1,-1,-1,0,0
1800
```

```
                                        .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 READ WRITE STATS MAY BE PRINTED AT THE END OF
                                        ;EACH TEST CYCLE VIA CONSOLE SWITCH FOURTEEN (14).
                                        ;*********************************************************


                                ;START 200, & 300**********
003026  012706  000500        START:  MOV     #500,SP         ;SET STACK PTR
003032  005037  000736                CLR     ASEQF           ;CLEAR AUTO SEQUENCE FLAG
003036  005027                        CLR     (PC)+           ;;CLEAR CHAIN INDICATOR
003040  000000                CHNFLG: .WORD   0               ;;CHAIN MODE INDICATOR
                                                              ;;1/0 = CHAIN/NOT CHAIN MODE
003042  022737  005022  000042        CMP     #SENDAD,@#42    ;;BRANCH IF LOADED VIA ACT11 CHAIN MODE
003050  001404                        BEQ     50$
003052  005737  000042                TST     @#42            ;;BRANCH IF IN DUMP MODE
003056  001413                        BEQ     52$
003060  000406                        BR      51$
003062  012737  000176  000610 50$:   MOV     #SWREG,SWR      ;;INVOKE SOFTWARE SWR
003070  012777  100000  175512        MOV     #100000,@SWR    ;;WITH HALT ON ERROR SET
003076  005237  003040        51$:    INC     CHNFLG          ;;SET CHNFLG = CHAIN MODE
003102  000137  003126                JMP     3$              ;;GO TO CHAIN ADDRESS
003106                        52$:
003106  122737  000006  000041        CMPB    #6,@#41         ;BRANCH IF LOADED VIA TMDP
003114  001010                        BNE     STAUT
003116  012704  027232                MOV     #MSG120,R4      ;ADVISE USER TO REMOVE TMDP FROM SLAVE
003122  000004                        TYPE
003124  000404                        BR      STAUT
003126  005237  000736        3$:     INC     ASEQF           ;SET AUTO SEQUENCE FLAG
003132  000137  021766                JMP     ASEQQ           ;GO TO AUTO SEQUENCER

                                ;START 240**********
003136  012737  000001  000636 STAUT:  MOV     #1,TINF         ;SET TTY ENTRY FLAG
003144  005037  01502.                CLR     RDFL            ;CLEAR RANDOM DATA FLAG
003150  000405                        BR      STARTB

                                ;START 204**********
003152  005037  000636        STARTC: CLR     TINF            ;CLEAR TTY INPUT FLAG
003156  000432                        BR      STARTD

                                ;START 210**********
003160  005037  000636        STARTA: CLR     TINF            ;CLEAR TTY ENTRY FLAG
003164  012700  000640        STARTB: MOV     #STFLG,R0       ;GET STARTING ADDRESS OF FLAGS
003170  012701  000076                MOV     #ENDFLG-STFLG,R1
003174  105020                1$:     CLRB    (R0)+           ;CLEAR FLAGS AND COUNTERS
003176  005301                        DEC     R1
```

```
1857  003200  001375                          BNE     1$
1858  003202  012706  000500                  MOV     #500,SP         ;SET STACK POINTER
1859  003206  004737  004276                  JSR     PC,RANSET       ;GO RESET RANDOM BASE
1860  003212  012700  001050                  MOV     #STTBL,R0       ;GET STARTING ADDRESS OF STAT TABLE
1861  003216  012701  001720                  MOV     #ENDTBL-STTBL,R1        ;AND # OF BYTES IN TABLE
1862  003222  105020                  2$:     CLRB    (R0)+           ;CLEAR STATISTIC COUNTERS
1863  003224  005301                          DEC     R1
1864  003226  001375                          BNE     2$
1865  003230  012737  177777  014270          MOV     #-1,PATS        ;PRESET PATTERN
1866  003236  012737  000001  000656  STARTE: MOV     #1,BLCNTR       ;PRESET BLOCK COUNTER
1867  003244  013746  000004          STARTD: MOV     @#4,-(SP)       ;SAVE ERROR TRAP VECTOR
1868  003250  013746  000006                  MOV     @#6,-(SP)
1869  003254  022737  000176  000610          CMP     #SWREG,SWR      ;BRANCH IF SOFTWARE SWR
1870  003262  001413                          BEQ     2$              ;ALREADY SELECTED
1871  003264  012737  003310  000004          MOV     #1$,@#4         ;SET TIMEOUT TRAP TO 1$ BELOW
1872  003272  005037  000006                  CLR     @#6
1873  003276  022777  177777  175304          CMP     #177777,@SWR    ;BRANCH IF SWR = 177777 TRAP
1874  003304  001402                          BEQ     2$              ;IF NOT AVAIL (1$) OTHERWISE
1875  003306  000404                          BR      3$              ;GO TO 3$
1876  003310  022626                  1$:     CMP     (SP)+,(SP)+     ;RESET STACK
1877  003312  012737  000176  000610  2$:     MOV     #SWREG,SWR      ;SET SWR = SOFTWARE SWR
1878  003320  012637  000006          3$:     MOV     (SP)+,@#6       ;RESTORE ERROR TRAP
1879  003324  012637  000004                  MOV     (SP)+,@#4
1880  003330  012706  000500                  MOV     #500,SP
1881  003334  004737  012212                  JSR     PC,TINP         ;GO GET PARAMETERS FROM TTY
1882  003340  012777  000040  175152          MOV     #40,@CS         ;INITIALIZE
1883  003346  005000                  STAUTO: CLR     R0              ;POINT TO FIRST ENTRY
1884  003350  022760  177777  000746  1$:     CMP     #-1,UN1(R0)     ;BRANCH IF LAST ENTRY
1885  003356  001406                          BEQ     2$
1886  003360  042760  100000  000746          BIC     #100000,UN1(R0) ;CLEAR EOT FLAG
1887  003366  062700  000002                  ADD     #2,R0           ;POINT TO NEXT UNIT ENTRY
1888  003372  000766                          BR      1$              ;CONTINUE CLEARING
1889  003374  013703  005054          2$:     MOV     REOTC,R3
1890  003400  000303                          SWAB    R3
1891  003402  110337  005054                  MOVB    R3,REOTC        ;RESTORE EOT CNTR
1892  003406  012777  000100  175176  START1: MOV     #100,@TKS       ;SET KEYBOARD IE BIT
1893  003414  013700  000676                  MOV     UNP,R0          ;R0 = UNIT TABLE POINTER
1894  003420  022760  177777  000746  STAR1A: CMP     #-1,UN1(R0)     ;BRANCH IF LAST ENTRY
1895  003426  001404                          BEQ     STAR1B
1896  003430  016037  000746  000552          MOV     UN1(R0),UDES    ;LOAD NEXT UNIT DESCRIPTION
1897  003436  000446                          BR      START4
1898  003440  005237  000656          STAR1B: INC     BLCNTR          ;BUMP BLOCK COUNTER
1899  003444  005737  000736                  TST     ASEQF           ;SEE IF AUTO SEQ
1900  003450  001411                          BEQ     STAP1C          ;IF NOT: BR
1901  003452  023737  000656  000742          CMP     BLCNTR,ABLCNT   ;SEE IF DONE SEQ
1902  003460  001005                          BNE     STAR1C          ;IF NOT: BR
1903  003462  005037  000656                  CLR     BLCNTR          ;RESET BLOCK CNTR
1904  003466  005037  000676                  CLR     UNP             ;RESET UNIT POINTER
1905  003472  000207                          RTS     PC              ;RETURN TO AUTO SEQ
1906  003474  005037  000676          STAR1C: CLR     UNP
1907  003500  005000                          CLR     R0
1908  003502  016037  000746  000552          MOV     UN1(R0),UDES    ;LOAD FIRST UNIT DESCRIPTION
1909  003510  032777  000200  175072          BIT     #200,@SWR       ;SEE IF RANDOM RECORD SIZE
1910  003516  001402                          BEQ     START2          ;IF NOT: BR
1911  003520  004737  012126                  JSR     PC,CCNTR        ;GO GENERATE RANDOM RECORD SIZE
1912  003524  032777  000400  175056  START2: BIT     #400,@SWR       ;SEE IF RANDOM DATA
```

```
1913  003532  001402                          BEQ     START3          ;IF NOT: BR
1914  003534  004737  014760                  JSR     PC,DATR         ;GO GENERATE RANDOM DATA
1915  003540  032777  000100  175042  START3: BIT     #100,@SWR       ;SEE IF RANDOM RECORD COUNT
1916  003546  001402                          BEQ     START4          ;IF NOT: BR
1917  003550  004737  012166                  JSR     PC,RCNTR        ;GO GENERATE RANDOM RECORD COUNT
1918  003554  005760  000746          START4: TST     UNI(R0)         ;SEE IF REACHED EOT
1919  003560  100002                          BPL     START0          ;IF NOT: BR
1920  003562  000137  004264                  JMP     START7          ;ELSE GO TO NEXT UNIT
1921  003566  013777  000550  174724  START0: MOV     DVN,@CS         ;SET DRIVE NUMBER
1922  003574  013777  000552  174740          MOV     UDES,@TC        ;SET UNIT NUMBER
1923  003602  105777  174714                  TSTB    @CS             ;SEE IF UNIT AVAIL
1924  003606  100412                          BMI     START4A         ;IF SO: BR
1925  003610  005337  000670                  DEC     STAL
1926  003614  001357                          BNE     START4          ;AWAIT TUR
1927  003616  004737  022570                  JSR     PC,PAPRT        ;PRINT HEADER
1928  003622  012704  025731                  MOV     #MSG49,R4
1929  003626  000004                          TYPE                    ;TYPE MSG
1930  003630  000000                          HALT                    ;STOP
1931  003632  000750                          BR      START4          ;RETRY
1932  003634  004737  014110          START4A: JSR    PC,DSUP         ;GO SET UP WRITE DATA
1933  003640  004737  005426                  JSR     PC,INIT         ;INIT SLAVE
1934  003644  004737  005056                  JSR     PC,RWND         ;REWIND
1935  003650  004737  005534                  JSR     PC,WRITE        ;WRITE
1936  003654  013737  000600  000670          MOV     TSTAL,STAL      ;SET TURN AROUND DELAY
1937  003662  004737  012116                  JSR     PC,STALL                ;DELAY
1938  003666  004737  007416                  JSR     PC,RSEQ         ;GO TO READ SEQUENCER
1939  003672  013737  000600  000670          MOV     TSTAL,STAL      ;SET TURN AROUND DELAY
1940  003700  004737  012116                  JSR     PC,STALL        ;DELAY
1941  003704  032777  040000  174676          BIT     #40000,@SWR     ;SEE IF SHOULD PRINT STATISTICS
1942  003712  001541                          BEQ     START5          ;IF NOT: BR
1943  003714  012700  000001                  MOV     #1,R0           ;SET RECORD COUNTER TO 1
1944  003720  004737  022570                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
1945  003724  004737  003734                  JSR     PC,STP          ;GO PRINT STATS
1946  003730  000137  004202                  JMP     STPX
1947  003734  004737  017100          STP:    JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
1948  003740  012704  026143                  MOV     #MSG65,R4
1949  003744  000004                          TYPE                    ;TYPE MSG
1950  003746  013704  000676                  MOV     UNP,R4
1951  003752  016403  001050                  MOV     RTYI(R4),R3
1952  003756  104400                          TYPOCT                  ;PRINT RETRIES
1953  003760  012704  026314                  MOV     #MSG73,R4
1954  003764  000004                          TYPE                    ;TYPE MSG
1955  003766  013704  000676                  MOV     UNP,R4
1956  003772  016403  001070                  MOV     WTERI(R4),R3
1957  003776  104400                          TYPOCT                  ;PRINT WRITE ERRORS
1958  004000  012704  026303                  MOV     #MSG72,R4
1959  004004  000004                          TYPE                    ;TYPE MSG
1960  004006  013704  000676                  MOV     UNP,R4
1961  004012  016403  001110                  MOV     RDERI(R4),R3
1962  004016  104400                          TYPOCT                  ;PRINT READ FORWARD ERRORS
1963  004020  012704  027107                  MOV     #MSG113,R4
1964  004024  000004                          TYPE                    ;TYPE MSG
1965  004026  013704  000676                  MOV     UNP,R4
1966  004032  016403  002670                  MOV     RFSOFT(R4),R3
1967  004036  104400                          TYPOCT                  ;PRINT FORWARD SOFT ERRORS
1968  004040  012704  027120                  MOV     #MSG114,R4
```

```
1969  004044  000004                      TYPE                    ;TYPE MSG
1970  004046  013704  000676              MOV     UNP,R4
1971  004052  016403  002730              MOV     RFHARD(R4),R3
1972  004056  104400                      TYPOCT                  ;PRINT HARD FORWARE ERRORS
1973  004060  012704  026374              MOV     #MSG77,R4
1974  004064  000004                      TYPE                    ;TYPE MSG
1975  004066  013704  000676              MOV     UNP,R4
1976  004072  016403  001130              MOV     DATER1(R4),R3
1977  004076  104400                      TYPOCT                  ;PRINT DATA ERROR FORWARD NUMBER
1978  004100  012704  026177              MOV     #MSG68,R4
1979  004104  000004                      TYPE                    ;TYPE MSG
1980  004106  013704  000676              MOV     UNP,R4
1981  004112  016403  001150              MOV     RDERR1(R4),R3
1982  004116  104400                      TYPOCT                  ;PRINT REVESE ERROR NUMBER
1983  004120  012704  027107              MOV     #MSG113,R4
1984  004124  000004                      TYPE                    ;TYPE MSG
1985  004126  013704  000676              MOV     UNP,R4
1986  004132  016403  002710              MOV     RRSOFT(R4),R3
1987  004136  104400                      TYPOCT                  ;PRINT REVERSE SOFT ERROR
1988  004140  012704  027120              MOV     #MSG114,R4
1989  004144  000004                      TYPE                    ;TYPE MSG
1990  004146  013704  000676              MOV     UNP,R4
1991  004152  016403  002750              MOV     RRHARD(R4),R3
1992  004156  104400                      TYPOCT
1993  004160  012704  026363              MOV     #MSG76,R4
1994  004164  000004                      TYPE                    ;TYPE MSG
1995  004166  013704  000676              MOV     UNP,R4
1996  004172  016403  001170              MOV     DEREV1(R4),R3
1997  004176  104400                      TYPOCT                  ;PRINT DATA REVERSE ERROR NUMBER
1998  004200  000207                      RTS     PC              ;RETURN
1999  004202  005237  000730      STPX:   INC     BTSTF           ;SET STAT ONLY PRINT
2000  004206  004737  007326              JSR     PC,BTPRT        ;PRINT BAD TAPE STATS
2001  004212  005037  000730              CLR     BTSTF           ;CLEAR FLAG
2002  004216  017700  174366      START5: MOV     @SWR,R0         ;LOAD SWR
2003  004222  042700  177762              BIC     #177762,R0      ;MASK READ/WRITE SWITCHES
2004  004226  022700  000015              CMP     #15,R0          ;SEE IF HAVE READ OR WRITE
2005  004232  001417                      BEQ     START8          ;IF NOT: BR
2006  004234  105777  174262      START6: TSTB    @DS             ;SEE IF HAVE UNIT READY
2007  004240  100411                      BMI     START7          ;IF SO: BR
2008  004242  005337  000670              DEC     STAL
2009  004246  001372                      BNE     START6          ;DELAY FOR TUR
2010  004250  004737  022570              JSR     PC,PAPRT        ;PRINT HEADER
2011  004254  012704  025731              MOV     #MSG49,R4
2012  004260  000004                      TYPE                    ;TYPE MSG
2013  004262  000000                      HALT                    ;STOP
2014  004264  062737  000002  000676  START7: ADD     #2,UNP          ;POINT TO NEXT UNIT
2015  004272  000137  003406      START8: JMP     START1          ;CONTINUE
2016
2017                                      ;RANDOM BASE RESET**********
2018
2019  004276  012737  153624  000626  RANSET: MOV     #153624,RANBAS  ;RESET BASE
2020  004304  012737  032561  000630          MOV     #32561,RANSAV   ;RESET BUFFER
2021  004312  013737  000632  000554          MOV     RCSAV,RCNT      ;RESET RECORD COUNT
2022  004320  013737  000634  000556          MOV     FCSAV,FMCNT     ;RESET FRAME COUNT
2023  004326  000207                          RTS     PC
2024
```

```
2025                                    ;*********************************************************
2026                                    ;REWIND FROM EOT:
2027                                    ;
2028                                    ;WHEN ANY TRANSPORT BEING TESTED REACHES END OF TAPE
2029                                    ;DURING A READ OR WRITE OPERATION, IT WILL BE REWOUND
2030                                    ;AND FLAGGED AS UNAVAILABLE UNTIL ALL AVAILABLE UNITS
2031                                    ;HAVE REACHED EOT AT WHICH TIME ALL TESTING WILL BE RESUMED
2032                                    ;AT A BLOCK COUNT OF ONE (1).  A MESSAGE WILL BE
2033                                    ;PRINTED ON THE SUPERVISORS CONSOLE AS EACH UNIT REACHES
2034                                    ;EOT AND IS REWOUND.
2035                                    ;*********************************************************
2036
2037  004330  013777  000552  174204  REOT:   MOV     UDES,@TC        ;LOAD TAPE CONTROL REGISTER
2038  004336  012777  000011  174144          MOV     #11,@C1         ;DRIVE CLEAR
2039  004344  105777  174152          1$:     TSTB    @0S             ;WAIT FOR DRY
2040  004350  100375                          BPL     1$
2041  004352  012777  000007  174130          MOV     #7,@C1          ;START REWIND
2042  004360  005737  000726                  TST     BTFLG           ;SEE IF BAD TAPE OVERFLOW REWIND
2043  004364  001004                          BNE     REOT1A          ;IF SO: BR
2044  004366  013700  000662                  MOV     EOTREC,R0
2045  004372  042700  100000                  BIC     #100000,R0      ;SET RECORD NUMBER OF EOT
2046  004376  005037  000662          REOT1A: CLR     EOTREC          ;CLEAR EOT INDICATOR & REC COUNT
2047  004402  004737  022570                  JSR     PC,PAPRT        ;PRINT HEADER
2048  004406  022737  000002  000726          CMP     #2,BTFLG        ;SEE IF POSITION ERROR
2049  004414  001003                          BNE     REOT1B          ;IF NOT: BR
2050  004416  012704  027000                  MOV     #MSG109,R4      ;SET POSITION ERROR MSG
2051  004422  000406                          BR      REOT1F
2052  004424  022737  000001  000726  REOT1B: CMP     #1,BTFLG        ;SEE IF BAD TAPE OVERFLOW
2053  004432  001004                          BNE     REOT1C          ;IF NOT: BR
2054  004434  012704  026626                  MOV     #MSG106,R4      ;SET BAD TAPE OVERFLOW MSG
2055  004440  000004                  REOT1F: TYPE                    ;TYPE MSG
2056  004442  000412                          BR      REOT1E
2057  004444  012704  024636          REOT1C: MOV     #MSG20,R4       ;SET EOT MSG
2058  004450  000004                          TYPE                    ;TYPE MSG
2059  004452  013704  000676                  MOV     UNP,R4
2060  004456  005264  002650                  INC     EOTCO(R4)       ;BUMP CNTR
2061  004462  016403  002650                  MOV     EOTCO(R4),R3
2062  004466  104400                          TYPOCT                  ;PRINT EOT CNTR
2063  004470  012704  026653          REOT1E: MOV     #MSG16A,R4
2064  004474  000004                          TYPE                    ;TYPE MSG
2065  004476  005037  000726                  CLR     BTFLG           ;CLEAR BAD TAPE FLAG
2066  004502  004737  003734                  JSR     PC,STP          ;PRINT STATS
2067  004506  004737  007326                  JSR     PC,BTPRT        ;PRINT BAD TAPE STATS
2068  004512  105777  174004          REOT2:  TSTB    @0S             ;BRANCH IF DRY SET
2069  004516  100414                          BMI     REOT2A
2070  004520  005337  000670                  DEC     STAL
2071  004524  001372                          BNE     REOT2           ;WAIT DRY
2072  004526  012737  024475  000654          MOV     #MSG6,EMADDR
2073  004534  004737  022570                  JSR     PC,PAPRT        ;PRINT HEADER
2074  004540  012704  026105                  MOV     #MSG60,R4
2075  004544  000004                          TYPE                    ;TYPE MSG
2076  004546  000000                          HALT
2077  004550  105337  005054          REOT2A: DECB    REOTC           ;SEE IF LAST UNIT TO REACH EOT
2078  004554  001410                          BEQ     REOT3           ;IF SO: BR
2079  004556  013700  000676                  MOV     UNP,R0
2080  004562  052760  100000  000746          BIS     #100000,UN1(R0) ;SET EOT FLAG
```

```
2081  004570  005726                          TST     (SP)+           ;RESET STACK POINTER
2082  004572  000137  004264                  JMP     START7          ;GO TO NEXT UNIT
2083  004576  000337  005054          REOT3:  SWAB    REOTC
2084  004602  013700  005054                  MOV     REOTC,R0
2085  004606  000337  005054                  SWAB    REOTC
2086  004612  110037  005054                  MOVB    R0,REOTC        ;RESTORE EOT UNIT COUNTER
2087  004616  005037  000676                  CLR     UNP
2088  004622  013700  000676                  MOV     UNP,R0          ;POINT TO FIRST UNIT
2089  004626  016037  000746  000552  REOT4:  MOV     UN1(R0),UDES    ;LOAD UNIT DESCRIPTION
2090  004634  013777  000552  173700          MOV     UDES,@TC        ;LOAD COMMAND REGISTER
2091  004642  032777  020000  173652  REOT5:  BIT     #20000,@DS
2092  004650  001374                          BNE     REOT5           ;AWAIT PIP RESET
2093  004652  032777  000002  173642          BIT     #2,@DS          ;SEE IF HAVE BOT
2094  004660  001012                          BNE     REOT6           ;IF SO: BR
2095  004662  012700  000001                  MOV     #1,R0
2096  004666  004737  022570                  JSR     PC,PAPRT        ;PRINT HEADER
2097  004672  012704  025676                  MOV     #MSG48,R4
2098  004676  000004                          TYPE                    ;TYPE MSG
2099  004700  000000                          HALT
2100  004702  013700  000676                  MOV     UNP,R0
2101  004706  042760  100000  000746  REOT6:  BIC     #100000,UN1(R0) ;CLEAR EOT FLAG
2102  004714  062737  000002  000676          ADD     #2,UNP
2103  004722  013700  000676                  MOV     UNP,R0          ;POINT TO NEXT UNIT
2104  004726  022760  177777  000746          CMP     #-1,UN1(R0)     ;BRANCH IF NOT LAST UNIT
2105  004734  001334                          BNE     REOT4
2106  004736  005037  000676          REOT7:  CLR     UNP             ;CLEAR UNIT POINTER
2107  004742  005037  000636                  CLR     TINF            ;CLEAR TTY INPUT FLAG
2108  004746  005737  000736                  TST     ASEQF           ;SEE IF AUTO SEQ
2109  004752  001402                          BEQ     REOTX           ;IF NOT: BR
2110  004754  005726                          TST     (SP)+           ;RESET STACK POINTER
2111  004756  000207                          RTS     PC              ;RETURN TO AUTO SEQ
2112  004760  004737  004276          REOTX:  JSR     PC,RANSET       ;GO RESET RANDOM BASE
2113  004764  012737  177777  014270          MOV     #-1,PATS        ;PRESET PATTERN
2114  004772  005037  015022                  CLR     RDFL            ;CLEAR RANDOM FLAG
2115  004776  005737  000572                  TST     SPFLG           ;SEE IF SINGLE PASS
2116  005002  001422                          BEQ     REOTXX          ;IF NOT: BR
2117  005004  012704  026506          TEND:   MOV     #MSG100,R4
2118  005010  000004                          TYPE                    ;TYPE MSG
2119  005012  013700  000042                  MOV     @#42,R0         ;GET ACT11 RETURN ADDRESS
2120  005016  001405                          BEQ     HERE            ;BRANCH IF NOT ACT11
2121  005020  000005                          RESET
2122  005022  004710          SENDAD: JSR     PC,(R0)
2123  005024  000240                          NOP
2124  005026  000240                          NOP
2125  005030  000240                          NOP
2126  005032  000240          HERE:   NOP
2127  005034  005737  003040                  TST     CHNFLG          ;BRANCH IF NOT CHAIN MODE
2128  005040  001402                          BEQ     1$
2129  005042  000137  021766                  JMP     ASEQ0           ;RETURN TO AUTO SEQUENCER
2130  005046  000000          1$:     HALT
2131  005050  000137  003236          REOTXX: JMP     STARTE          ;RESTART AT BLOCK NUMBER ONE
2132  005054  000000          REOTC:  0                               ;EOT UNIT COUNTER
```

```
2133                                    ;*********************************************************
2134                                    ;REWIND ALL AVAIL TAPES:
2135                                    ;
2136                                    ;THIS ROUTINE; ENTERED VIA CONSOLE SWITCH NINE (9),
2137                                    ;WILL REWIND ALL AVAILABLE TAPES TO BOT NO MATTER
2138                                    ;WHERE THEY ARE CURRENTLY POSITIONED AND RESUME TESTING
2139                                    ;ON THE CURRENTLY SELECTED UNIT.
2140                                    ;*********************************************************
2141
2142  005056  032777  001000  173524  RWND:   BIT     #1000,@SWR      ;SEE IF SHOULD REWIND
2143  005064  001001                          BNE     RWNDA           ;IF SO: BR
2144  005066  000207                          RTS     PC              ;ELSE EXIT
2145  005070  013737  000676  000716  RWNDA:  MOV     UNP,UPS         ;SAVE UNIT POINTER
2146  005076  005037  000676                  CLR     UNP             ;CLEAR POINTER
2147  005102  005037  000662                  CLR     EOTREC          ;CLEAR EOT FLAG
2148  005106  000337  005054                  SWAB    REOTC
2149  005112  013700  005054                  MOV     REOTC,R0
2150  005116  000337  005054                  SWAB    REOTC
2151  005122  110037  005054                  MOVB    R0,REOTC        ;RESTORE EOT UNIT COUNTER
2152  005126  013700  000676  RWND0:  MOV     UNP,R0          ;POINT TO UNIT ENTRY
2153  005132  022760  177777  000746          CMP     #-1,UN1(R0)     ;BRANCH IF LAST ENTRY
2154  005140  001445                          BEQ     RWND2
2155  005142  005760  000746                  TST     UN1(R0)         ;SEE IF ALREADY REWINDING
2156  005146  100433                          BMI     RWND1A          ;IF SO: BR
2157  005150  016037  000746  000552          MOV     UN1(R0),UDES    ;SET UNIT DESCRIPTION
2158  005156  013777  000552  173356          MOV     UDES,@TC        ;LOAD COMMAND REGISTER
2159  005164  012777  000011  173316          MOV     #11,@C1         ;DRIVE CLEAR
2160  005172  012777  000007  173310          MOV     #7,@C1          ;START REWIND
2161  005200  105777  173316          1$:     TSTB    @DS
2162  005204  100414                          BMI     RWND1A          ;IF DRY: BR
2163  005206  005337  000670                  DEC     STAL
2164  005212  001372                          BNE     1$              ;AWAIT DRY
2165  005214  012737  024475  000654          MOV     #MSG6,EMADDR
2166  005222  004737  022570                  JSR     PC,PAPRT        ;PRINT HEADER
2167  005226  012704  026226                  MOV     #MSG70,R4
2168  005232  000004                          TYPE                    ;TYPE MSG
2169  005234  000000                          HALT
2170  005236  042760  100000  000746  RWND1A: BIC     #100000,UN1(R0) ;CLEAR EOT FLAG
2171  005244  062737  000002  000676          ADD     #2,UNP          ;BUMP POINTER
2172  005252  000725                          BR      RWND0           ;DO NEXT UNIT
2173  005254  005037  000676          RWND2:  CLR     UNP             ;CLEAR POINTER
2174  005260  013700  000676          RWND3:  MOV     UNP,R0          ;POINT TO UNIT ENTRY
2175  005264  022760  177777  000746          CMP     #-1,UN1(R0)     ;BRANCH IF LAST ENTRY
2176  005272  001441                          BEQ     RWNDX
2177  005274  016037  000746  000552          MOV     UN1(R0),UDES    ;SET UNIT DESCRIPTION
2178  005302  013777  000552  173232          MOV     UDES,@TC        ;LOAD COMMAND REGISTER
2179  005310  032777  020000  173204  1$:     BIT     #20000,@DS
2180  005316  001374                          BNE     1$              ;AWAIT PIP RESET
2181  005320  013777  000552  173214          MOV     UDES,@TC        ;LOAD UNIT DESCRIPTION
2182  005326  032777  000002  173166          BIT     #2,@DS          ;SEE IF HAVE BOT
2183  005334  001407                          BEQ     RWND6           ;IF NOT: BR
2184  005336  062737  000002  000676  RWND5:  ADD     #2,UNP          ;BUMP POINTER
2185  005344  012777  000011  173136          MOV     #11,@C1         ;DRIVE CLEAR
2186  005352  000742                          BR      RWND3           ;DO NEXT UNIT
2187  005354  012700  000001          RWND6:  MOV     #1,R0
2188  005360  004737  022570                  JSR     PC,PAPRT        ;PRINT HEADER
```

```
2189  005364  012704  025676                    MOV     #MSG48,R4
2190  005370  000004                            TYPE                    ;TYPE MSG
2191  005372  000000                            HALT
2192  005374  000760                            BR      RWNDS           ;DO NEXT UNIT
2193  005376  013737  000716  000676  RWNDX:    MOV     UPS,UNP         ;RESTORE UNIT POINTER
2194  005404  013700  000676                    MOV     UNP,R0
2195  005410  016037  000746  000552            MOV     UNI(R0),UDES    ;RESET UNIT DESCRIPTION
2196  005416  013777  000552  173116            MOV     UDES,@TC
2197  005424  000207                            RTS     PC              ;RETURN TO TEST
2198
2199
2200                                            ;******************************************************
2201                                            ;INITIALIZE SELECTED SALVE
2202                                            ;THIS ROUTINE REWINDS AND SETS THE PROPER DENSITY IF
2203                                            ;THE DENSITY REQUIRED FOR THE TEST IS DIFFERENT FROM
2204                                            ;THE DENSITY AT WHICH THE SLAVE IS SELECTED.
2205                                            ;******************************************************
2206
2207  005426  013746  000552            INIT:   MOV     UDES,-(SP)      ;GET UNIT DESCRIPTION
2208  005432  013777  000550  173060            MOV     DVN,@CS         ;LOAD DRIVE #
2209  005440  011677  173076                    MOV     (SP),@TC        ;LOAD SLAVE # & SLAVE DESCRIPTION
2210  005444  042716  174377                    BIC     #174377,(SP)    ;CLEAR ALL BUT DENSITY BITS
2211  005450  022726  001400                    CMP     #1400,(SP)+     ;BRANCH IF NOT NRZ
2212  005454  001005                            BNE     1$
2213  005456  032777  000040  173036            BIT     #40,@DS         ;BRANCH IF SLAVE IS IN PE MODE
2214  005464  001422                            BEQ     4$              ;PES = 0
2215  005466  000404                            BR      2$
2216  005470  032777  000040  173024  1$:       BIT     #40,@DS         ;BRANCH IF SLAVE IS IN PE MODE
2217  005476  001015                            BNE     4$              ;PES = 1
2218  005500  012777  000007  173002  2$:       MOV     #7,@C1          ;LOAD REWIND COMMAND
2219  005506  105777  173010          20$:      TSTB    @DS             ;WAIT FOR READY
2220  005512  100375                            BPL     20$
2221  005514  032777  020000  173000  3$:       BIT     #20000,@DS      ;WAIT FOR PIP = 0
2222  005522  001374                            BNE     3$
2223  005524  012777  000011  172756            MOV     #11,@C1         ;CLEAR DRIVE
2224  005532  000207                    4$:     RTS     PC
```

```
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258  005534  032777  000001  173046  WRITE:  BIT     #1,@SWR         ;SEE IF SHOULD WRITE
2259  005542  001402                          BEQ     WRTE
2260  005544  000137  006326                  JMP     WEX             ;IF NOT: BR
2261  005550  013700  000554          WRTE:   MOV     RCNT,R0         ;R0=RECORD COUNT
2262  005554  012737  024470  000654  W0:     MOV     #MSG5,EMADDR    ;SET ERROR MSG ADDRESS
2263  005562  013777  000556  172726          MOV     FMCNT,@FC       ;LOAD CHAR COUNT
2264  005570  012777  027326  172716          MOV     #WDATA,@BA      ;SET DATA ADDR
2265  005576  112737  000060  000674          MOVB    #60,MTC1        ;SET WRITE OP COMMAND
2266  005604  012737  005616  000664          MOV     #W1,RTRN        ;SET RETURN ADDRESS
2267  005612  000137  021054                  JMP     TAPE            ;GO EXECUTE COMMAND
2268  005616  032777  002000  172676  W1:     BIT     #2000,@DS       ;SEE IF EOT
2269  005624  001412                          BEQ     W2              ;IF NOT AT EOT: BR
2270  005626  005737  000662                  TST     EOTREC          ;BRANCH IF WRITTEN PAST EOT
2271  005632  100407                          BMI     W2
2272  005634  010037  000662                  MOV     R0,EOTREC       ;SAVE RECORD COUNT
2273  005640  062737  100001  000662          ADD     #100001,EOTREC  ;SET EOT INDICATOR & ADD 1 TO # OF
2274                                                                  ;RECORDS WRITTEN
2275  005646  012700  000002                  MOV     #2,R0           ;SET TO WRITE 1 LAST RECORD
2276  005652  032777  010000  172730  W2:     BIT     #10000,@SWR     ;SEE IF SHOULD CHECK ERRORS
2277  005660  001002                          BNE     W3              ;IF NOT: BR
2278  005662  004737  017236                  JSR     PC,ERCHK        ;GO CHECK ERRORS
2279  005666  013737  000576  000670  W3:     MOV     WSTAL,STAL      ;SET DELAY
2280  005674  004737  012116                  JSR     PC,STALL        ;DELAY
```

;**********
;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 RESULTS IN
;ANY ERROR CONDITION, A WRITE RETRY OF THAT OPERATION
;MAY BE DONE BY SETTING SWITCH FOUR (4) TO A ONE (1).
;THE RETRY CONSISTS OF A BACKSPACE, ERASE FORWARD, AND
;REWRITE OF THE RECORD. (SEE WRITE RETRY SUBROUTINE)
;AFTER ALL DATA RECORDS IN THE BLOCK HAVE BEEN
;WRITTEN, THE WRITE ROUTINE WILL EXECUTE A WRITE
;TAPE MARK COMMAND IF THE TTY RESPONSE TM=1 WAS
;MADE AT INITIAL START. THE TM IS COUNTED AS TOTAL
;DATA RECORDS PLUS ONE (IE; IF 100 DATA RECORDS TM=RECORD 101)
;IF THE WRITE OPERATION (DATA OR TM) CAUSES THE ELECTED SLAVE
;TO REACH END OF TAPE (EOT) AND THERE IS TO BE NO READING DONE,
;(SW2 AND SW3 SET TO A 1) THEN THE SLAVE IS REWOUND AND
;FLAGGED AS UNAVAILABLE FOR TESTING UNTIL ALL SLAVES HAVE
;REACHED EOT AND BEEN REWOUND AT WHICH TIME TESTING IS
;RESUMED ON ALL AVAILABLE SLAVES.
;WRITE RETRY MAY BE ALLOWED VIA CONSOLE SWITCH FOUR (4).
;ERROR CHECKING MAY BE DISALLOWED VIA CONSOLE SWITCH
;TWELVE (12).
;WRITING TO TAPE MAY BE DISALLOWED VIA CONSOLE SWITCH
;ZERO (0).
;**********

```
2281  005700  005737  000714          TST     RTYFL           ;SEE IF RETRY TIME
2282  005704  001401                  BEQ     W3A             ;IF NOT: BR
2283  005706  000207                  RTS     PC              ;ELSE RETURN
2284  005710  005737  000710  W3A:    TST     SERFL           ;SEE IF WRITE ERROR
2285  005714  001450                  BEQ     W5              ;IF NOT: BR
2286  005716  013704  000676          MOV     UNP,R4
2287  005722  005264  001070          INC     WTER1(R4)       ;BUMP WRITE ERROR
2288  005726  005037  000710          CLR     SERFL           ;CLEAR STATUS ERROR FLAG
2289  005732  032777  000020  172650  BIT     #20,@SWR        ;SEE IF RETRY
2290  005740  001436                  BEQ     W5              ;IF NOT: BR
2291  005742  013703  000724          MOV     ERSAV,R3
2292  005746  042703  102700          BIC     #102700,R3      ;MASK UNRECOVERABLE ERROR
2293  005752  001410                  BEQ     W4              ;IF SO: BR
2294  005754  004737  022570          JSR     PC,PAPRT        ;PRINT HEADER
2295  005760  012704  026405          MOV     #MSG78,R4
2296  005764  000004                  TYPE                    ;TYPE MSG
2297  005766  004737  011236          JSR     PC,NRTP         ;PRINT ER FOR NON-RETRYABLE
2298  005772  000421                  BR      W5
2299  005774  013704  000676  W4:     MOV     UNP,R4
2300  006000  005264  001050          INC     RTY1(R4)        ;BUMP RETRY CNTR
2301  006004  032777  002000  172576  BIT     #2000,@SWR      ;SEE IF PRINT ERRORS
2302  006012  001003                  BNE     W4A             ;IF NOT: BR
2303  006014  012704  026121          MOV     #MSG64,R4
2304  006020  000004                  TYPE                    ;TYPE MSG
2305  006022  005037  000704  W4A:    CLR     RTCNT           ;CLEAR RETRY NUMBER
2306  006026  005037  000702          CLR     RPCNT           ;CLEAR REPEAT COUNTER
2307  006032  004737  006370          JSR     PC,WRTY         ;GO RETRY WRITE ERROR
2308  006036  005037  000714  W5:     CLR     RTYFL           ;CLEAR RETRY COUNTER
2309  006042  005300                  DEC     R0              ;SEE IF DONE ALL
2310  006044  001243                  BNE     W0              ;IF NOT: BR
2311  006046  005737  000564  W6:     TST     TMEX            ;SEE IF TM
2312  006052  001525                  BEQ     WEX             ;IF NOT: BR
2313  006054  005237  000700          INC     TMFLG           ;SET TM FLAG
2314  006060  012737  026026  000654  WTM:    MOV     #MSG54,EMADDR   ;POINT TO TM ERROR MSG
2315  006066  012737  000026  000674          MOV     #26,MTC1        ;SET TM OP CODE
2316  006074  012777  000000  172414          MOV     #0,@FC          ;LOAD FRAME COUNTER
2317  006102  012777  027326  172404          MOV     #WDATA,@BA      ;LOAD BUS ADDRESS
2318  006110  012737  006122  000664          MOV     #WTM0,RTRN      ;SAVE RETURN ADDRESS
2319  006116  000137  021054                  JMP     TAPG            ;WRITE TM
2320  006122  032777  010000  172460  WTM0:   BIT     #10000,@SWR     ;SEE IF SHOULD CHECK ERRORS
2321  006130  001076                  BNE     WEX
2322  006132  032777  000004  172362          BIT     #4,@DS          ;SEE IF TM STATUS
2323  006140  001011                  BNE     WTM1            ;IF SO: BR
2324  006142  012737  027326  020774          MOV     #WDATA,CADER    ;SET EXPT BUS ADDRESS
2325  006150  012737  000001  021002          MOV     #1,DRVER        ;INDICATE ERROR
2326  006156  004737  020056                  JSR     PC,ERPT         ;PRINT TM ERROR
2327  006162  000404                  BR      WTM2
2328  006164  012703  027326  WTM1:   MOV     #WDATA,R3       ;SET EXPT ADDRESS
2329  006170  004737  017332          JSR     PC,ER2          ;GO CHECK FOR OTHER ERRORS
2330  006174  005737  000714  WTM2:   TST     RTYFL           ;SEE IF RETRY
2331  006200  001401                  BEQ     WTM3            ;IF NOT: BR
2332  006202  000207                  RTS     PC              ;ELSE RETURN TO RETRY ROUTINE
2333  006204  005737  000710  WTM3:   TST     SERFL           ;SEE IF WRITE ERROR
2334  006210  001446                  BEQ     WEX             ;IF NOT: BR
2335  006212  013704  000676          MOV     UNP,R4
2336  006216  005264  001070          INC     WTER1(R4)       ;BUMP WRITE ERROR
```

```
2337  006222  032777  000020  172360          BIT     #20,@SWR        ;SEE IF SHOULD RETRY
2338  006230  001436                          BEQ     WEX             ;IF NOT: BR
2339  006232  013703  000724                  MOV     ERSAV,R3
2340  006236  042703  102700                  BIC     #102700,R3      ;MASK UNRECOVERABLE ERROR
2341  006242  001410                          BEQ     WTM4            ;IF SO: BR
2342  006244  004737  022570                  JSR     PC,PAPRT        ;PRINT HEADER
2343  006250  012704  026405                  MOV     #MSG78,R4
2344  006254  000004                          TYPE                    ;TYPE MSG
2345  006256  004737  011236                  JSR     PC,NRTP         ;PRINT ER FOR NON-RETRYABLE
2346  006262  000421                          BR      WEX
2347  006264  005037  000702          WTM4:   CLR     RPCNT           ;CLEAR REPEAT CNTR
2348  006270  013704  000676                  MOV     UNP,R4
2349  006274  005264  001050                  INC     RTY1(R4)        ;BUMP RETRY CNTR
2350  006300  005037  000704                  CLR     RTCNT           ;CLEAR RETRY CNTR
2351  006304  032777  002000  172276          BIT     #2000,@SWR      ;SEE IF PRINT ERRORS
2352  006312  001003                          BNE     WTM4A           ;IF NOT: BR
2353  006314  012704  026121                  MOV     #MSG64,R4
2354  006320  000004                          TYPE                    ;TYPE MSG
2355  006322  004737  006370          WTM4A:  JSR     PC,WRTY         ;GO DO RETRY
2356  006326  005037  000714          WEX:    CLR     RTYFL           ;CLEAR RETRY FLAG
2357  006332  005037  000700                  CLR     TMFLG           ;CLEAR TAPE MARK FLAG
2358  006336  005737  000662                  TST     EOTREC          ;BRANCH IF NOT AT EOT
2359  006342  100011                          BPL     WRWX
2360  006344  017703  172240          WRW:    MOV     @SWR,R3
2361  006350  042703  177763                  BIC     #177763,R3
2362  006354  022703  000014                  CMP     #14,R3          ;SEE IF WRITE ONLY
2363  006360  001002                          BNE     WRWX            ;IF NOT: BR
2364  006362  000137  004330                  JMP     REOT            ;ELSE REWIND
2365  006366  000207          WRWX:   RTS     PC                      ;EXIT
```

```
2366                                            ;*************************
2367                                            ;WRITE ERROR RETRY
2368                                            ;
2369                                            ;*************************
2370
2371  006370  012737  000001  000714  WRTY:   MOV     #1,RTYFL        ;SET RETRY FLAG
2372  006376  004737  006772          WRTY0:  JSR     PC,WRTSB        ;GO SPACE REVERSE FOR REPEAT
2373  006402  005737  000700                  TST     TMFLG           ;SEE IF TAPE MARK TIME
2374  006406  001003                          BNE     WRTYTM          ;IF SO: BR
2375  006410  004737  005554                  JSR     PC,WD           ;REWRITE RECORD
2376  006414  000402                          BR      WRTYR           ;GO ON
2377  006416  004737  006060          WRTYTM: JSR     PC,WTM          ;GO WRITE TAPE MARK AGAIN
2378  006422  005737  000710          WRTYR:  TST     SERFL           ;REWRITE GOOD
2379  006426  001024                          BNE     WRTY2           ;IF NOT: BR
2380  006430  005237  000702                  INC     RPCNT           ;BUMP REPEAT COUNTER
2381  006434  022737  000004  000702          CMP     #4,RPCNT        ;SEE IF FOUR GOOD REPEATS
2382  006442  001355                          BNE     WRTY0           ;IF NOT: REPEAT
2383  006444  032777  002000  172136          BIT     #2000,@SWR      ;SEE IF PRINT
2384  006452  001011                          BNE     WRTY1           ;IF NOT: BR
2385  006454  012704  026613                  MOV     #MSG105,R4
2386  006460  000004                          TYPE                    ;TYPE MSG
2387  006462  012704  026143                  MOV     #MSG65,R4
2388  006466  000004                          TYPE                    ;TYPE MSG
2389  006470  013703  000704                  MOV     RTCNT,R3
2390  006474  104400                          TYPOCT                  ;PRINT RETRY NUMBER
2391  006476  000207          WRTY1:  RTS     PC                      ;RESUME TESTING
2392  006500  013703  000724          WRTY2:  MOV     ERSAV,R3        ;GET ER
2393  006504  005037  000650                  CLR     TEMP3           ;CLEAR RECOVERABLE ERROR INDICATOR
2394  006510  042703  102700                  BIC     #102700,R3      ;MASK RECOVERABLE BITS
2395  006514  001413                          BEQ     WRTY2A          ;IF RECOVERABLE: BR
2396  006516  004737  022570                  JSR     PC,PAPRT        ;PRINT HEADER
2397  006522  012704  026405                  MOV     #MSG78,R4
2398  006526  000004                          TYPE                    ;TYPE MSG
2399  006530  004737  011236                  JSR     PC,NRTP         ;PRINT ER
2400  006534  012737  000001  000650          MOV     #1,TEMP3        ;SET FLAG
2401  006542  000407                          BR      WRTY2B
2402  006544  032777  002000  172036  WRTY2A: BIT     #2000,@SWR      ;SEE IF PRINT
2403  006552  001025                          BNE     WRTY3           ;IF NOT: BR
2404  006554  012704  027032                  MOV     #MSG110,R4
2405  006560  000004                          TYPE                    ;TYPE MSG
2406  006562  012704  026143          WRTY2B: MOV     #MSG65,R4
2407  006566  000004                          TYPE                    ;TYPE MSG
2408  006570  013703  000704                  MOV     RTCNT,R3
2409  006574  104400                          TYPOCT                  ;PRINT RETRY NUMBER
2410  006576  012704  027054                  MOV     #MSG111,R4
2411  006602  000004                          TYPE                    ;TYPE MSG
2412  006604  013703  000702                  MOV     RPCNT,R3
2413  006610  104400                          TYPOCT                  ;PRINT REPEAT NUMBER
2414  006612  005737  000650                  TST     TEMP3           ;SEE IF DID NON-RECOVERABLE
2415  006616  001403                          BEQ     WRTY3           ;IF NOT: BR
2416  006620  005037  000650                  CLR     TEMP3           ;CLEAR FLAG
2417  006624  000207                          RTS     PC              ;EXIT
2418  006626  005737  000704          WRTY3:  TST     RTCNT           ;SEE IF FIRST RETRY
2419  006632  001004                          BNE     WRTY3A          ;IF NOT: BR
2420  006634  013704  000676                  MOV     UNP,R4
2421  006640  005364  001070                  DEC     WTER1(R4)       ;DECREMENT WRITE ERROR CNTR
```

```
2422  006644  013704  000676                WRTY3A: MOV     UNP,R4          ;GET UNIT NUMBER
2423  006650  016437  001030  000732                MOV     BTADDR(R4),BTPT ;GET ADDRESS OF UNIT BAD TAPE CNTR
2424  006656  017704  172050                        MOV     @BTPT,R4        ;GET COUNTER
2425  006662  005724                              TST     (R4)+           ;SET POINTER OFFSET
2426  006664  010477  172042                      MOV     R4,@BTPT
2427  006670  013703  000732                      MOV     BTPT,R3
2428  006674  060304                              ADD     R3,R4           ;SET ABSOLUTE POINTER
2429  006676  013714  000656                      MOV     BLCNTR,(R4)     ;SET BLOCK NUMBER
2430  006702  062704  000040                      ADD     #40,R4          ;ADD RCNT OFFSET
2431  006706  013714  000554                      MOV     RCNT,(R4)
2432  006712  160014                              SUB     R0,(R4)         ;SET RECORD NUMBER
2433  006714  005214                              INC     (R4)            ;CORRECT RECORD NUMBER
2434  006716  022777  000040  172006              CMP     #40,@BTPT       ;SEE IF TOO MANY BAD SPOTS
2435  006724  001002                              BNE     WRTY4           ;IF NOT: BR
2436  006726  000137  007166                      JMP     BTOV            ;ELSE GO TO BAD TAPE OVERFLOW
2437  006732  005237  000704                WRTY4: INC     RTCNT           ;BUMP RETRY COUNTER
2438  006736  022737  000004  000704              CMP     #4,RTCNT        ;SEE IF DONE 4 RETRIES
2439  006744  001410                              BEQ     WRTY5           ;IF SO: BR
2440  006746  013704  000676                      MOV     UNP,R4
2441  006752  005264  001050                      INC     RTY1(R4)        ;BUMP RETRY COUNTER
2442  006756  005237  000734                      INC     ERTFL           ;SET ERASE FLAG
2443  006762  000137  006376                      JMP     WRTY0           ;DO NEXT RETRY
2444  006766  000137  007402                WRTY5: JMP     BTUR            ;ELSE GO TO BAD TAPE UNRECOVERABLE
2445
2446                                              ;WRITE RETRY BACKSPACE-ERASE SUBROUTINE**********
2447
2448  006772  005037  000710                WRTSB: CLR     SERFL           ;CLEAR FLAG
2449  006776  013737  000600  000670              MOV     TSTAL,STAL
2450  007004  004737  012116                      JSR     PC,STALL        ;DO TURN AROUND DELAY
2451  007010  012737  026154  000654              MOV     #MSG66,EMADDR   ;SET ERROR CODE
2452  007016  012777  177777  171472              MOV     #-1,@FC         ;SET TO BACKSPACE 1 RECORD
2453  007024  012777  033334  171462              MOV     #RDATA,@BA      ;SET BA
2454  007032  004737  012046                      JSR     PC,BKRT         ;GO BACKSPACE
2455  007036  005737  000710                      TST     SERFL           ;SEE IF ERROR
2456  007042  001406                              BEQ     WRTSB1          ;IF NOT: BR
2457  007044  012737  000002  000726        WRTSB0: MOV     #2,BTFLG        ;SET FLAG
2458  007052  022626                              CMP     (SP)+,(SP)+     ;RESET STACK
2459  007054  000137  004330                      JMP     REOT            ;GO REWIND AND REMOVE FROM TESTING
2460  007060  005737  000734                WRTSB1: TST     ERTFL           ;SEE IF SHOULD ERASE
2461  007064  001001                              BNE     WRTSB2          ;IF SO: BR
2462  007066  000207                              RTS     PC              ;RETURN
2463  007070  005037  000734                WRTSB2: CLR     ERTFL           ;CLEAR ERASE FLAG
2464  007074  005037  000702                      CLR     RPCNT           ;CLEAR REPEAT CNTR
2465  007100  005037  000710                      CLR     SERFL           ;CLEAR FLAG
2466  007104  012737  026167  000654              MOV     #MSG67,EMADDR   ;SET ERROR CODE
2467  007112  005077  171400                      CLR     @FC             ;CLEAR FRAME COUNT
2468  007116  012737  000024  000674              MOV     #24,MTC1        ;SET ERASE OP-CODE
2469  007124  012777  027326  171362              MOV     #WDATA,@BA      ;SET BA
2470  007132  012737  007144  000664              MOV     #WRTSB3,RTRN    ;SET RETURN ADDRESS
2471  007140  000137  021054                      JMP     TAPG            ;GO ERASE
2472  007144  012703  027326                WRTSB3: MOV     #WDATA,R3       ;SET EXPT BA
2473  007150  004737  017332                      JSR     PC,ER2          ;GO CHECK ERRORS
2474  007154  005737  000710                      TST     SERFL           ;SEE IF ERROR
2475  007160  001737                              BEQ     WRTSB1          ;IF NOT: BR
2476  007162  000137  007044                      JMP     WRTSB0
2477
```

```
2478                                            ;BAD TAPE OVERFLOW SUBROUTINE**********
2479
2480  007166  005037  000714          BTOV:   CLR     RTYFL           ;CLEAR RETRY FLAG
2481  007172  012737  000001  000726          MOV     #1,BTFLG        ;SET BAD TAPE OVERFLOW FLAG
2482  007200  000137  004330                  JMP     REOT            ;GO REWIND AND REMOVE FROM TESTING
2483  007204  013701  000732          BTOVD:  MOV     BTPT,R1         ;SET TABLE POINTER
2484  007210  005721                          TST     (R1)+
2485  007212  005000                          CLR     R0
2486  007214  010003                  BTOV1:  MOV     R0,R3
2487  007216  000241                          CLC
2488  007220  006003                          ROR     R3              ;R3=R3/2 FOR CORRECT NUMBER
2489  007222  104400                          TYPOCT                  ;PRINT ENTRY NUMBER
2490  007224  012704  024565                  MOV     #MSG13+1,R4
2491  007230  000004                          TYPE                    ;TYPE MSG
2492  007232  011103                          MOV     (R1),R3
2493  007234  104400                          TYPOCT                  ;PRINT BLOCK NUMBER
2494  007236  012704  024572                  MOV     #MSG14,R4
2495  007242  000004                          TYPE                    ;TYPE MSG
2496  007244  062701  000040                  ADD     #40,R1          ;SET POINTER OFFSET FOR RECOED NUMBER
2497  007250  012103                          MOV     (R1)+,R3
2498  007252  104400                          TYPOCT                  ;PRINT RECORD NUMBER
2499  007254  162701  000040                  SUB     #40,R1          ;RESET POINTER FOR BLOCK NUMBER
2500  007260  005720                          TST     (R0)+
2501  007262  020077  171444                  CMP     R0,@BTPT        ;SEE IF DONE
2502  007266  001404                          BEQ     BTOV2           ;IF SO: BR
2503  007270  012704  025117                  MOV     #MSG28,R4
2504  007274  000004                          TYPE                    ;TYPE MSG
2505  007276  000746                          BR      BTOV1           ;CONTINUE
2506  007300  005737  000730          BTOV2:  TST     BTSTF           ;SEE IF STAT ONLY PRINT
2507  007304  001007                          BNE     BTOVX           ;IF SO: BR
2508  007306  012703  000041                  MOV     #41,R3          ;SET SIZE OF TABLE
2509  007312  013704  000732                  MOV     BTPT,R4         ;SET POINTER
2510  007316  005024                  BTOV3:  CLR     (R4)+           ;CLEAR TABLE
2511  007320  005303                          DEC     R3              ;SEE IF DONE
2512  007322  001375                          BNE     BTOV3           ;IF NOT: BR
2513  007324  000207                  BTOVX:  RTS     PC              ;RETURN
2514
```

```
2515
2516                                        ;BAD TAPE STATISTIC PRINT********
2517
2518  007326  012704  025117        BTPRT:  MOV     #MSG28,R4
2519  007332  000004                        TYPE                    ;TYPE MSG
2520  007334  013704  000676                MOV     UNP,R4
2521  007340  016437  001030  000732        MOV     BTADDR(R4),BTPT ;SET TABLE POINTER
2522  007346  017703  171360                MOV     @BTPT,R3
2523  007352  000241                        CLC
2524  007354  006003                        ROR     R3              ;CORRECT NUMBER
2525  007356  104400                        TYPOCT                  ;PRINT NUMBER OF BAD SPOTS
2526  007360  012704  027066                MOV     #MSG112,R4
2527  007364  000004                        TYPE                    ;TYPE MSG
2528  007366  005777  171340                TST     @BTPT           ;SEE IF ANY BAD SPOTS
2529  007372  001001                        BNE     BTPRT1          ;IF SO: BR
2530  007374  000207                        RTS     PC              ;ELSE RETURN
2531  007376  000137  007204        BTPRT1: JMP     BTOVO           ;PRINT STATS
2532
2533                                        ;BAD TAPE UNRECOVERABLE SUBROUTINE**********
2534
2535  007402  004737  022570        BTUR:   JSR     PC,PAPRT        ;PRINT HEADER
2536  007406  012704  026714                MOV     #MSG107,R4
2537  007412  000004                        TYPE                    ;TYPE MSG
2538  007414  000207                        RTS     PC              ;RESUME TESTING
2539
```

```
                                        ;*********************************************************
                                        ;READ SEQUENCER:
                                        ;
                                        ;THIS ROUTINE IS USED TO DETERMINE THE SEQUENCE
                                        ;IN WHICH READ TAPE OPERATIONS ARE TO BE PERFORMED.
                                        ;THIS IS NECESSARY WHEN THE UNIT BEING TESTED IS
                                        ;CAPABLE OF READING DATA IN BOTH THE FORWARD AND
                                        ;REVERSE DIRECTIONS.  CONSOLE SWITCHES ONE (1), TWO (2),
                                        ;AND THREE (3) ARE USED TO DETERMINE THE READ SEQUENCE.
                                        ;CONSOLE SWITCH ONE (1) DETERMINES WHETHER TO READ
                                        ;THE BLOCK OF DATA FORWARD FIRST OR REVERSE FIRST.
                                        ;SWITCH TWO (2) DISALLOWS READING IN THE REVERSE
                                        ;DIRECTION AND SWITCH THREE (3) DISALLOWS READING IN
                                        ;THE FORWARD DIRECTION.
                                        ;*********************************************************

007416  012737  000002  000562  RSEQ:  MOV   #2,RDCMD
007424  017704  171160                 MOV   @SWR,R4          ;READ SWITCHES
007430  042704  177763                 BIC   #177763,R4       ;MASK READ BITS & SEE IF BOTH READS
007434  001004                         BNE   RSR              ;IF NOT: BR
007436  032777  000002  171144         BIT   #2,@SWR          ;SEE IF READ REVERSE FIRST
007444  001050                         BNE   RSFR             ;IF NOT: BR
007446  032777  000004  171134  RSR:   BIT   #4,@SWR          ;SEE IF SHOULD READ REVERSE
007454  001005                         BNE   RSF              ;IF NOT: BR
007456  012737  010000  000562         MOV   #10000,RDCMD     ;LOAD READ REVERSE COMMAND
007464  004737  007730                 JSR   PC,READ          ;GO READ REVERSE
007470  032777  000010  171112  RSF:   BIT   #10,@SWR         ;SEE IF SHOULD READ FORWARD
007476  001025                         BNE   RSEX             ;IF NOT: BR
007500  032737  010000  000562         BIT   #10000,RDCMD     ;SEE IF HAVE READ REVERSE
007506  001406                         BEQ   RSF0             ;IF NOT: BR
007510  013737  000600  000670         MOV   TSTAL,STAL
007516  004737  012116                 JSR   PC,STALL         ;DO READ STALL
007522  000406                         BR    RSF1
007524  032777  000001  171056  RSF0:  BIT   #1,@SWR          ;SEE IF WRITE
007532  001002                         BNE   RSF1             ;IF NOT: BR
007534  004737  011672                 JSR   PC,BKSP          ;GO BACKSPACE
007540  012737  000002  000562  RSF1:  MOV   #2,RDCMD         ;LOAD READ FORWARD COMMAND
007546  004737  007730                 JSR   PC,READ          ;GO READ
007552  005737  000662         RSEX:  TST   EOTREC           ;BRANCH IF NOT AT EOT
007556  100002                         BPL   1$
007560  000137  004330                 JMP   REOT             ;ELSE GO TO REWIND
007564  000207                 1$:     RTS   PC               ;EXIT

007566  012737  010000  000562  RSFR:  MOV   #10000,RDCMD
007574  032777  000010  171006         BIT   #10,@SWR         ;SEE IF SHOULD READ FORWARD
007602  001013                         BNE   RSFR1            ;IF NOT: BR
007604  032777  000001  170776         BIT   #1,@SWR          ;SEE IF WRITE
007612  001002                         BNE   RSFR0            ;IF NOT: BR
007614  004737  011672                 JSR   PC,BKSP          ;GO BACKSPACE TO START
007620  012737  000002  000562  RSFR0: MOV   #2,RDCMD         ;LOAD READ FORWARD COMMAND
007626  004737  007730                 JSR   PC,READ          ;GO READ FORWARD
007632  032777  000004  170750  RSFR1: BIT   #4,@SWR          ;SEE IF SHOULD READ REVERSE
007640  001344                         BNE   RSEX             ;IF NOT: BR
007642  032737  010000  000562         BIT   #10000,RDCMD
007650  001005                         BNE   RSFR2            ;IF READ REVERSE: BR
007652  013737  000600  000670         MOV   TSTAL,STAL       ;DO READ STALL
```

```
2596  007660  004737  012116                      JSR     PC,STALL
2597  007664  012737  010000  000562  RSFR2:      MOV     #10000,RDCMD    ;LOAD READ REVERSE
2598  007672  004737  007730                      JSR     PC,READ         ;GO READ REVERSE
2599  007676  005737  000662                      TST     EOTREC          ;SEE IF AT END OF TAPE
2600  007702  100011                              BPL     RSFRX           ;IF NOT: BR
2601  007704  163737  000554  000662              SUB     RCNT,EOTREC
2602  007712  005437  000662                      NEG     EOTREC          ;SET TO PROPER RECORD NUMBER
2603  007716  005237  000662                      INC     EOTREC
2604  007722  000137  004330                      JMP     REOT            ;ELSE GO TO REWIND
2605  007726  000207                      RSFRX:  RTS     PC              ;EXIT
2606
```

```
                                        ;*********************************************************
                                        ;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 AN ERROR IS PRINTED AND THE
                                        ;PROGRAM WILL HALT.  TESTING MAY BE RESUMED BY PRESSING
                                        ;THE CONTINUE SWITCH.
                                        ;IF A TAPE MARK IS EXPECTED (TM=1) THEN THE
                                        ;READ ROUTINE EXPECTS THE FIRST RECORD OF A
                                        ;READ REVERSE TO BE A TM, AND THE LAST RECORD
                                        ;OF A READ FORWARD TO BE A TM.  REMEMBER
                                        ;THAT THE TM ADDS ONE (1) TO THE TOTAL NUMBER
                                        ;OF RECORDS IN A BLOCK.
                                        ;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).
                                        ;*********************************************************

007730  013700  000554          READ:   MOV     RCNT,R0         ;LOAD REC CNTR
007734  005737  000662                  TST     EOTREC          ;SEE IF EOT
007740  100013                          BPL     RDA             ;IF NOT: BR
007742  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ FORWARD
007750  001407                          BEQ     RDA             ;IF SO: BR
007752  042737  100000  000662          BIC     #100000,EOTREC  ;CLEAR FLAG
007760  013703  000662                  MOV     EOTREC,R3       ;GET MODIFIED RECORD COUNT
007764  160300                          SUB     R3,R0           ;SET RECORD AT
007766  005200                          INC     R0              ;SET TO PROPER NUMBER OF RECORDS
007770  012737  024475  000654  RDA:    MOV     #MSG6,EMADDR    ;SET ERROR MSG ADDRESS
007776  005037  000700                  CLR     TMFLG
010002  032737  010000  000562          BIT     #10000,RDCMD
010010  001406                          BEQ     RDO             ;IF READ FORWARD: BR
010012  005737  000564                  TST     TMEX            ;SEE IF TM
010016  001403                          BEQ     RDO             ;IF NOT: BR
010020  005237  000700                  INC     TMFLG           ;SET TM FLAG
010024  005200                          INC     R0
010026  013777  000556  170462  RDO:    MOV     FMCNT,@FC       ;LOAD CHAR CNTR
010034  012777  033334  170452          MOV     #RDATA,@BA      ;LOAD DATA ADDR
010042  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
010050  001417                          BEQ     RD1A            ;IF NOT: BR
010052  013703  000556                  MOV     FMCNT,R3
010056  005103                          COM     R3
010060  032737  000020  000552          BIT     #20,UDES        ;SEE IF CORE DUMP
010066  001402                          BEQ     RD1             ;IF NOT: BR
010070  000241                          CLC
010072  006003                          ROR     R3              ;R3 = FC/2
010074  060377  170414          RD1:    ADD     R3,@BA          ;SET REVERSE BUS ADDRESS
010100  012737  000076  000674          MOV     #76,MTC1        ;SET READ REVERSE
010106  000403                          BR      RD1B
```

```
2663  010110  012737  000070  000674  RD1A:   MOV     #70,MTC1        ;SET READ FORWARD
2664  010116  012737  010130  000664  RD1B:   MOV     #RD2,RTRN       ;SET INTERRUPT RETURN ADDRESS
2665  010124  000137  021054          RD1D:   JMP     1APG            ;GO EXECUTE TAPE COMMAND
2666  010130  032737  010000  000562  RD2:    BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2667  010136  001014                          BNE     RD3             ;IF SO: BR
2668  010140  032777  002000  170354          BIT     #2000,@DS       ;SEE IF EOT
2669  010146  001410                          BEQ     RD3             ;IF NOT: BR
2670  010150  005737  000700                  TST     TMFLG           ;SEE IF TM
2671  010154  001005                          BNE     RD3             ;IF SO: BR
2672  010156  010037  000662                  MOV     R0,EOTREC
2673  010162  052737  100000  000662          BIS     #100000,EOTREC  ;SET EOT FLAG
2674  010170  032777  000002  170324  RD3:    BIT     #2,@DS          ;SEE IF AT LOAD POINT
2675  010176  001410                          BEQ     RD4             ;IF NOT: BR
2676  010200  004737  022570                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
2677  010204  012704  024676                  MOV     #MSG22,R4
2678  010210  000004                          TYPE                    ;TYPE MSG
2679  010212  000000                          HALT
2680  010214  000137  003160                  JMP     STARTA          ;RESTART
2681  010220  032777  004000  170362  RD4:    BIT     #4000,@SWR      ;SEE IF SHOULD CHECK ERRORS
2682  010226  001121                          BNE     RD5             ;IF NOT: BR
2683  010230  005737  000700                  TST     TMFLG
2684  010234  001472                          BEQ     RD4B            ;IF NO TM EXPT: BR
2685  010236  032777  000004  170256          BIT     #4,@DS
2686  010244  001024                          BNE     RD4A            ;IF TM RECVD: BR
2687  010246  012737  033334  020774          MOV     #RDATA,CADER    ;SAVE EXPT BUS ADDRESS
2688  010254  012737  000002  021002          MOV     #2,DRVER        ;SET TM STATUS ERROR FLAG
2689  010262  004737  020056                  JSR     PC,ERPT         ;GO PRINT TM ERROR
2690  010266  013704  000676                  MOV     UNP,R4
2691  010272  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2692  010300  001403                          BEQ     1$              ;IF NOT: BR
2693  010302  005264  001150                  INC     RDERR1(R4)      ;BUMP READ REVERSE ERROR
2694  010306  000502                          BR      RD6
2695  010310  005264  001110          1$:     INC     RDER1(R4)       ;BUMP READ FORWARD ERROR
2696  010314  000477                          BR      RD6
2697  010316  012703  033334          RD4A:   MOV     #RDATA,R3
2698  010322  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2699  010330  001007                          BNE     RD4A0           ;IF SO: BR
2700  010332  032737  002000  000552          BIT     #2000,UDES      ;SEE IF IN PE
2701  010340  001025                          BNE     RD4A2           ;IF SO: BR
2702  010342  062703  000002                  ADD     #2,R3
2703  010346  000422                          BR      RD4A2
2704  010350  013704  000556          RD4A0:  MOV     FMCNT,R4
2705  010354  005104                          COM     R4
2706  010356  032737  000020  000552          BIT     #20,UDES        ;SEE IF CORE DUMP
2707  010364  001402                          BEQ     RD4A1           ;IF NOT: BR
2708  010366  000241                          CLC
2709  010370  006004                          ROR     R4              ;SET TO FC/2
2710  010372  060403          RD4A1:  ADD     R4,R3           ;SET EXPT BUS ADDRESS
2711  010374  042703  000001                  BIC     #1,R3           ;MAKE EXPT ADDRESS EVEN
2712  010400  032737  002000  000552          BIT     #2000,UDES      ;SEE IF IN PE
2713  010406  001002                          BNE     RD4A2           ;IF SO: BR
2714  010410  162703  000002                  SUB     #2,R3
2715  010414  004737  017332          RD4A2:  JSR     PC,ER2
2716  010420  000402                          BR      RD4C
2717  010422  004737  017236          RD4B:   JSR     PC,ERCHK        ;GO CHECK ERRORS
2718  010426  005737  000710          RD4C:   TST     SERFL
```

```
2719  010432  001417                          BEQ     RD5             ;IF NO ERROR: BR
2720  010434  013704  000676                  MOV     UNP,R4
2721  010440  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2722  010446  001003                          BNE     RD4D            ;IF SO: BR
2723  010450  005264  001110                  INC     RDER1(R4)       ;BUMP READ FORWARD ERROR
2724  010454  000402                          BR      RD4E
2725  010456  005264  001150          RD4D:   INC     RDERR1(R4)      ;BUMP READ REVERSE ERROR
2726  010462  004737  010664          RD4E:   JSR     PC,RDRTY        ;GO RETRY
2727  010466  005037  000714                  CLR     RTYFL           ;CLEAR RETRY FLAG
2728  010472  032777  020000  170110  RD5:    BIT     #20000,@SWR     ;SEE IF SHOULD DO DATA CHECK
2729  010500  001005                          BNE     RD6             ;IF NOT; BR
2730  010502  005737  000700                  TST     TMFLG
2731  010506  001002                          BNE     RD6
2732  010510  004737  015366                  JSR     PC,DCHK         ;GO CHECK DATA
2733  010514  005037  000710          RD6:    CLR     SERFL           ;CLEAR STATUS ERROR FLAG
2734  010520  004737  014232                  JSR     PC,DS3          ;CLEAR BUFFER
2735  010524  032777  000040  170056          BIT     #40,@SWR        ;SEE IF SHOULD YOZZLE
2736  010532  001402                          BEQ     RD7             ;IF NOT: BR
2737  010534  004737  011252                  JSR     PC,YOZ          ;ELSE GO YOZZLE
2738  010540  013737  000574  000670  RD7:    MOV     RSTAL,STAL      ;SET DELAY
2739  010546  004737  012116                  JSR     PC,STALL        ;STALL
2740  010552  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2741  010560  001403                          BEQ     RD7A            ;IF NOT: BR
2742  010562  005037  000700                  CLR     TMFLG           ;CLEAR TAPE MARK FLAG
2743  010566  000405                          BR      RD10
2744  010570  005737  000662          RD7A:   TST     EOTREC          ;SEE IF EOT FOUND
2745  010574  100002                          BPL     RD10            ;IF NOT: BR
2746  010576  012700  000001                  MOV     #1,R0           ;SET TO EOT
2747  010602  005300                  RD10:   DEC     R0
2748  010604  001402                          BEQ     RD11            ;IF DONE ALL: BR
2749  010606  000137  010026                  JMP     RD0
2750  010612  032737  010000  000562  RD11:   BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2751  010620  001016                          BNE     RDEX            ;IF SO: BR
2752  010622  005737  000662                  TST     EOTREC          ;SEE IF FOUND EOT
2753  010626  100413                          BMI     RDEX            ;IF SO: BR
2754  010630  005737  000564                  TST     TMEX            ;SEE IF TM EXPECTED
2755  010634  001410                          BEQ     RDEX            ;IF NOT: BR
2756  010636  005737  000700                  TST     TMFLG           ;SEE IF TM FOUND
2757  010642  001005                          BNE     RDEX            ;IF SO: BR
2758  010644  005237  000700                  INC     TMFLG           ;ELSE SET FLAG
2759  010650  005200                          INC     R0              ;SET RECORD COUNT TO ONE
2760  010652  000137  010026                  JMP     RD0             ;GO READ TM
2761  010656  005037  000700          RDEX:   CLR     TMFLG
2762  010662  000207                  RDX:    RTS     PC              ;EXIT
```

```
2763                                    ;*******************************************************
2764                                    ;READ ERROR RETRY SUBROUTINE:
2765                                    ;
2766                                    ;THIS SUBROUTINE WILL RETRY ALL DATA RELATED
2767                                    ;READ ERRORS UP TO EIGHT (8) TIMES.  IF ALL
2768                                    ;FOUR RETRIES ARE BAD, IT IS CONSIDERED
2769                                    ;A HARD ERROR.  IF ANY ARE GOOD, IT IS A
2770                                    ;SOFT ERROR.  RETRIES MAY BE INHIBITED
2771                                    ;VIA SWITCH FOUR (SW4=0: INHIBIT RETRIES)
2772                                    ;*******************************************************
2773
2774  010664  032777  000020  167716  RDRTY:  BIT     #20,@SWR        ;SEE IF RETRY INHIBITED
2775  010672  001001                          BNE     RDRT0           ;IF NOT: BR
2776  010674  000207                          RTS     PC              ;ELSE RETURN
2777  010676  013703  000724          RDRT0:  MOV     ERSAV,R3
2778  010702  042703  102700                  BIC     #102700,R3      ;MARK NON-RECOVERABLE ERROR BITS
2779  010706  001410                          BEQ     RDRT1           ;IF NOT: BR
2780  010710  004737  022570                  JSR     PC,PAPRT        ;PRINT HEADER
2781  010714  012704  026446                  MOV     #MSG79,R4
2782  010720  000004                          TYPE                    ;TYPE MSG
2783  010722  004737  011236                  JSR     PC,NRTP         ;PRINT ER FOR NON-RETRYABLE ERROR
2784  010726  000207          RDRT1A: RTS     PC                      ;RETURN
2785  010730  032777  002000  167652  RDRT1:  BIT     #2000,@SWR      ;SEE IF PRINT INHIBITED
2786  010736  001003                          BNE     RDRT1B          ;IF SO: BR
2787  010740  012704  026121                  MOV     #MSG64,R4
2788  010744  000004                          TYPE                    ;TYPE MSG
2789  010746  005037  000704          RDRT1B: CLR     RTCNT           ;CLEAR RETRY COUNTER
2790  010752  005037  000710          RDRTG:  CLR     SERFL           ;CLEAR STATUS ERROR FLAG
2791  010756  012737  000002  000714          MOV     #2,RTYFL        ;SET READ RETRY FLAG
2792  010764  004737  011252                  JSR     PC,YOZ          ;GO TO YOZZLE TO RETRY READ
2793  010770  005737  000710                  TST     SERFL           ;SEE IF RETRY ERROR
2794  010774  001031                          BNE     RDRT5           ;IF SO: BR
2795  010776  032777  002000  167604          BIT     #2000,@SWR
2796  011004  001011                          BNE     RDRT2
2797  011006  012704  026613                  MOV     #MSG105,R4
2798  011012  000004                          TYPE                    ;TYPE MSG
2799  011014  012704  026143                  MOV     #MSG65,R4
2800  011020  000004                          TYPE                    ;TYPE MSG
2801  011022  013703  000704                  MOV     RTCNT,R3
2802  011026  104400                          TYPOCT                  ;PRINT RETRY NUMBER
2803  011030  013704  000676          RDRT2:  MOV     UNP,R4
2804  011034  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2805  011042  001003                          BNE     RDRT3           ;IF SO: BR
2806  011044  005264  002670                  INC     RFSOFT(R4)      ;ELSO BUMP FORWARD SOFT ERROR COUNTER
2807  011050  000402                          BR      RDRT4
2808  011052  005264  002710          RDRT3:  INC     RRSOFT(R4)      ;BUMP ERRORS SOFT CNTR
2809  011056  000207          RDRT4:  RTS     PC                      ;RETURN
2810  011060  013703  000724          RDRT5:  MOV     ERSAV,R3        ;GET ER
2811  011064  005037  000650                  CLR     TEMP3           ;CLEAR RECOVERABLE ERROR INDICATOR
2812  011070  042703  102700                  BIC     #102700,R3      ;MASK RECOVERABLE BITS
2813  011074  001413                          BEQ     RDRT5A          ;IF RECOVERABLE: BR
2814  011076  004737  022570                  JSR     PC,PAPRT        ;PRINT HEADER
2815  011102  012704  026446                  MOV     #MSG79,R4
2816  011106  000004                          TYPE                    ;TYPE MSG
2817  011110  004737  011236                  JSR     PC,NRTP         ;PRINT ER
2818  011114  012737  000001  000650          MOV     #1,TEMP3        ;SET FLAG
```

```
2819  011122  000404                          BR      RDRT5B
2820  011124  032777  002000  167456  RDRT5A: BIT     #2000,@SWR      ;SEE IF PRINT INHIBITED
2821  011132  001014                          BNE     RDRT6           ;IF SO: BR
2822  011134  012704  026143          RDRT5B: MOV     #MSG65,R4
2823  011140  000004                          TYPE                    ;TYPE MSG
2824  011142  013703  000704                  MOV     RTCNT,R3
2825  011146  104400                          TYPOCT                  ;PRINT RETRY NUMBER
2826  011150  005737  000650                  TST     TEMP3           ;SEE IF DID NON-RECOVERABLE
2827  011154  001403                          BEQ     RDRT6           ;IF NOT: BR
2828  011156  005037  000650                  CLR     TEMP3           ;CLEAR FLAG
2829  011162  000207                          RTS     PC              ;EXIT
2830  011164  005237  000704          RDRT6:  INC     RTCNT
2831  011170  023737  000704  000604          CMP     RTCNT,RETRY     ;SEE IF DONE 8 RETRIES
2832  011176  001265                          BNE     RDRTG           ;IF NOT: BR
2833  011200  012704  027131                  MOV     #MSG115,R4
2834  011204  000004                          TYPE                    ;TYPE MSG
2835  011206  013704  000676                  MOV     UNP,R4
2836  011212  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2837  011220  001003                          BNE     RDRT7           ;IF SO: BR
2838  011222  005264  002730                  INC     RFHARD(R4)      ;BUMP FORWARD HARD ERROR CNTR
2839  011226  000402                          BR      RDRTX
2840  011230  005264  002750          RDRT7:  INC     RRHARD(R4)      ;BUMP REVERSE HARD ERROR CNTR
2841  011234  000207                  RDRTX:  RTS     PC              ;RETURN
2842
2843  011236  013703  000724          NRTP:   MOV     ERSAV,R3        ;GET ER REGISTER
2844  011242  104400                          TYPOCT                  ;PRINT ER
2845  011244  004737  021020                  JSR     PC,FRPRT        ;PRINT F OR R
2846  011250  000207                          RTS     PC              ;RETURN
```

```
                                        ;*********************************************************
                                        ;YOZZLE SUBROUTINE:
                                        ;
                                        ;THIS SUBROUTINE, ENTERED VIA SWITCH FIVE (5), IS USED TO PERFORM
                                        ;A CONTINUOUS READ AND SPACE OVER OF THE CURRENT RECORD ON TAPE.
                                        ;FULL STATUS AND DATA CHECKING MAY BE PERFORMED
                                        ;OR NOT VIA CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13).
                                        ;A SOFTWARE DELAY IS PERFORMED BETWEEN EACH READ
                                        ;AND SPACE OPERATION AND MAY BE VARIED BY TYPING
                                        ;CNTRL C ON THE TTY AND ENTERING A VALUE IN RESPONSE
                                        ;TO THE PRINTED REQUEST.
                                        ;*********************************************************
011252 013737 000602 000670  YOZ:    MOV     YSTAL,STAL
011260 004737 012116                 JSR     PC,STALL        ;DO YOZZLE STALL
011264 012777 177777 167224  YOZO:   MOV     #-1,@FC         ;SET TO 1 RECORD SPACING
011272 032737 010000 000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
011300 001404                        BEQ     YOZA            ;IF NOT: BR
011302 112737 000030 000674          MOVB    #30,MTC1        ;SET TO SPACE FORWARD
011310 000403                        BR      YOZB
011312 112737 000032 000674  YOZA:   MOVB    #32,MTC1        ;SET TO SPACE REVERSE
011320 012737 011340 000664  YOZB:   MOV     #YOZC,RTRN      ;SET RETURN ADDRESS
011326 012737 177775 000670          MOV     #177775,STAL    ;SET TIME MULTIPLIER
011334 000137 021054                 JMP     TAPG            ;GO YOZZLE
011340 005737 000700         YOZC:   TST     TMFLG           ;SEE IF TM
011344 001404                        BEQ     1$              ;IF NOT: BR
011346 012737 040000 000670          MOV     #40000,STAL     ;SET TM STALL
011354 000403                        BR      2$
011356 013737 000602 000670  1$:     MOV     YSTAL,STAL
011364 004737 012116         2$:     JSR     PC,STALL        ;DO YOZZLE STALL
011370 012777 033334 167116          MOV     #RDATA,@BA      ;SET BUS ADDRESS
011376 032737 010000 000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
011404 001416                        BEQ     YOZC1           ;IF NOT: BR
011406 013703 000556                 MOV     FMCNT,R3
011412 005103                        COM     R3
011414 032737 000020 000552          BIT     #20,UDES        ;SEE IF CORE DUMP
011422 001401                        BEQ     YOZC0           ;IF NOT: BR
011424 006203                        ASR     R3              ;R3 = FC/2
011426 060377 167062         YOZC0:  ADD     R3,@BA          ;SET REVERSE BUS ADDRESS
011432 012737 000076 000674          MOV     #76,MTC1        ;SET READ REVERSE
011440 000403                        BR      YOZC2
011442 012737 000070 000674  YOZC1:  MOV     #70,MTC         ;SET READ FORWARD
011450 013777 000556 167040  YOZC2:  MOV     FMCNT,@FC       ;SET CHARACTER COUNT
011456 012737 011470 000664          MOV     #YOZD,RTRN      ;SET RETURN ADDRESS
011464 000137 021054                 JMP     TAPG            ;GO READ
011470 032777 004000 167112  YOZD:   BIT     #4000,@SWR      ;SEE IF SHOULD CHECK ERRORS
011476 001050                        BNE     YOZE            ;IF NOT: BR
011500 005737 000700                 TST     TMFLG           ;SEE IF TAPE MARK TIME
011504 001443                        BEQ     YOZD1           ;IF NOT: BR
011506 032737 010000 000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
011514 001425                        BEQ     YOZDO           ;IF NOT: BR
011516 012703 033334                 MOV     #RDATA,R3
011522 013704 000556                 MOV     FMCNT,R4
011526 005104                        COM     R4
011530 032737 000020 000552          BIT     #20,UDES        ;SEE IF CORE DUMP
011536 001401                        BEQ     YOZD4           ;IF NOT: BR
```

```
2903  011540  006204                          ASR     R4              ;SET TO FC/2
2904  011542  060403                  YOZD4:  ADD     R4,R3           ;SET EXPT BUS ADDRESS
2905  011544  042703  000001                  BIC     #1,R3           ;MAKE EXPT ADDRESS EVEN
2906  011550  032737  002000  000552          BIT     #2000,UDES      ;SEE IF PE
2907  011556  001001                          BNE     YOZD2           ;IF SO: BR
2908  011560  005743                          TST     -(R3)           ;SET EXPT BA
2909  011562  004737  017332          YOZD2:  JSR     PC,ER2          ;GO CHECK ERRORS
2910  011566  000430                          BR      YOZF
2911  011570  012703  033334          YOZD0:  MOV     #RDATA,R3
2912  011574  032737  002000  000552          BIT     #2000,UDES      ;SEE IF PE
2913  011602  001001                          BNE     YOZD3           ;IF SO: BR
2914  011604  005723                          TST     (R3)+           ;SET EXPT BA
2915  011606  004737  017332          YOZD3:  JSR     PC,ER2          ;GO CHECK ERRORS
2916  011612  000416                          BR      YOZF
2917  011614  004737  017236          YOZD1:  JSR     PC,ERCHK        ;ELSE GO CHECK ERRORS
2918  011620  005737  000714          YOZE:   TST     RTYFL           ;SEE IF RETRY
2919  011624  001013                          BNE     YOZG            ;IF SO: BR
2920  011626  032777  020000  166754          BIT     #20000,@SWR     ;SEE IF SHOULD CHECK DATA
2921  011634  001005                          BNE     YOZF            ;IF NOT: BR
2922  011636  005737  000700                  TST     TMFLG           ;SEE IF TAPE MARK
2923  011642  001002                          BNE     YOZF            ;IF SO: BR
2924  011644  004737  015366                  JSR     PC,DCHK         ;ELSE GO CHECK DATA
2925  011650  004737  014232          YOZF:   JSR     PC,DS3          ;GO CLEAR DATA AREA
2926  011654  032777  000040  166726  YOZG:   BIT     #40,@SWR        ;SEE IF SHOULD CONTINUE YOZZLE
2927  011662  001402                          BEQ     YOZH            ;IF NOT: BR
2928  011664  000137  011264                  JMP     YOZD
2929  011670  000207                  YOZH:   RTS     PC              ;EXIT
2930
```

```
                                        ;*********************************************************
                                        ;BACKSPACE SUBROUTINE:
                                        ;
                                        ;THIS SUBROUTINE IS USED TO PERFORM THE
                                        ;BACKSPACE OPERATION REQUIRED BY THE READ
                                        ;ROUTINE FOR READ FORWARD AFTER WRITING.
                                        ;IF A TAPE MARK IS EXPECTED (TM=1) THEN THE SPACE
                                        ;ROUTINE ASSUMES THAT THE TM WILL BE FIRST WHEN
                                        ;BACKSPACING.  THEREFORE TWO OPERATIONS ARE REQUIRED
                                        ;TO SPACE OVER A BLOCK.  FIRST SPACE OVER THE TM, THEN
                                        ;SPACE OVER THE DATA RECORDS.
                                        ;A CHECK FOR RECORD COUNT ZERO IS MADE AT THE
                                        ;END OF THE SPACE OPERATION TO ASSURE THAT PROPER
                                        ;TAPE POSITIONING WAS DONE.
                                        ;*********************************************************
011672  013737  000600  000670  BKSP:   MOV     TSTAL,STAL
011700  004737  012116                  JSR     PC,STALL        ;DO TURN AROUND STALL
011704  012737  024525  000654          MOV     #MSG10,EMADDR
011712  012777  033334  166574          MOV     #RDATA,@BA
011720  005737  000564                  TST     TMEX            ;SEE IF TM
011724  001440                          BEQ     BO              ;IF NOT: BR
011726  012777  177777  166562          MOV     #-1,@FC
011734  012737  000032  000674          MOV     #32,MTC1
011742  012737  011754  000664          MOV     #BKTM,RTRN
011750  000137  021054                  JMP     TAPG            ;SPACE TO TM
011754  032777  010000  166626  BKTM:   BIT     #10000,@SWR     ;SEE IF SHOULD CHECK ERROR
011762  001021                          BNE     BO              ;IF NOT: BR
011764  012737  026035  000654          MOV     #MSG55,EMADDR
011772  032777  000004  166522          BIT     #4,@DS          ;SEE IF TM
012000  001006                          BNE     BKTMO           ;IF SO: BR
012002  012737  033334  020774          MOV     #RDATA,CADER
012010  004737  020056                  JSR     PC,ERPT         ;PRINT ERROR
012014  000404                          BR      BO
012016  012703  033334          BKTMO:  MOV     #RDATA,R3
012022  004737  017332                  JSR     PC,ER2
012026  013700  000554          BO:     MOV     RCNT,R0
012032  005400                          NEG     R0              ;BUILD SPACE AMOUNT
012034  012737  024525  000654          MOV     #MSG10,EMADDR   ;SET ERROR MESG ADDRESS
012042  010077  166450                  MOV     R0,@FC
012046  012737  000032  000674  BKRT:   MOV     #32,MTC1        ;SET SPACE REVERSE
012054  012737  012072  000664          MOV     #B1,RTRN        ;SET RETURN ADDRESS
012062  010037  000670                  MOV     R0,STAL         ;SET INTERRUPT TIME MULTIPLIER
012066  000137  021054                  JMP     TAPG            ;GO DO SPACE
012072  012703  033334          B1:     MOV     #RDATA,R3
012076  004737  017332                  JSR     PC,ER2
012102  013737  000600  000670  B2:     MOV     TSTAL,STAL      ;DO STALL
012110  004737  012116                  JSR     PC,STALL        ;STALL
012114  000207                          RTS     PC              ;EXIT
```

```
                                ;*********************************************************
                                ;STALL ROUTINE:
                                ;
                                ;THIS ROUTINE IS USED TO PROVIDE SOFTWARE DELAYS
                                ;DURING READ, WRITE, TURN AROUND, AND YOZZLE.
                                ;THE DELAY TIMES MAY BE SET BY THE OPERATOR AT
                                ;INITIAL START FROM 200(8) OR MAY BE MODIFIED
                                ;AT ANY TIME BY ENTERING CNTRL C ON THE TTY AND
                                ;INSERTING NEW VALUES IN RESPONSE TO THE REQUEST.
                                ;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.
                                ;*********************************************************
 012116  005337  000670  STALL: DEC     STAL
 012122  001375                 BNE     STALL           ;DELAY
 012124  000207                 RTS     PC              ;EXIT
```

```
3004
3005                                    ;*********************************************************
3006                                    ;RANDOM CHARACTER COUNT GENERATOR:
3007                                    ;
3008                                    ;THIS ROUTINE ENTERED VIA CONSOLE SWITCH
3009                                    ;SEVEN (7) IS USED TO GENERATE A RANDOM
3010                                    ;CHARACTER COUNT FOR EACH DATA BLOCK.
3011                                    ;ALL RECORDS WITHIN A GIVEN BLOCK WILL BE
3012                                    ;THE SAME, BUT EACH BLOCK WILL VARY.
3013                                    ;THE LIMITS ARE TWENTY (20) TO FOUR THOUSAND
3014                                    ;(4000) OCTAL CHARACTERS PER RECORD.
3015                                    ;*********************************************************
3016
3017  012126  012701  177760       CCNTR:  MOV     #-20,R1         ;SET HIGH LIMIT
3018  012132  012702  175000               MOV     #-3000,R2       ;SET LOW LIMIT
3019  012136  004737  023152               JSR     PC,RANG         ;GO GENERATE NUMBER
3020  012142  042737  000001  000630       BIC     #1,RANSAV
3021  012150  013737  000630  000556       MOV     RANSAV,FMCNT    ;SET CHAR COUNT
3022  012156  012737  177777  014270       MOV     #-1,PATS        ;PRESET DATA PATTERN
3023  012164  000207                       RTS     PC              ;EXIT
3024
3025                                    ;*********************************************************
3026                                    ;RANDOM RECORD COUNT GENERATOR:
3027                                    ;
3028                                    ;THIS ROUTINE ENTERED VIA CONSOLE SWITCH SIX (6)
3029                                    ;IS USED TO GENERATE A RANDOM NUMBER OF RECORDS
3030                                    ;FOR EACH BLOCK OF DATA.
3031                                    ;THE LIMITS ARE ONE (1) TO FIVE HUNDRED (500) OCTAL
3032                                    ;RECORDS PER BLOCK.
3033                                    ;*********************************************************
3034
3035  012166  012702  000001       RCNTR:  MOV     #1,R2           ;SET LOW LIMIT
3036  012172  012701  000500               MOV     #500,R1         ;SET HIGH LIMIT
3037  012176  004737  023152               JSR     PC,RANG         ;GO GENERATE NUMBER
3038  012202  013737  000630  000554       MOV     RANSAV,RCNT     ;SET RECORD COUNT
3039  012210  000207                       RTS     PC              ;EXIT
3040
3041
```

```
3042
3043                    ;*********************************************************
3044                    ;TEST CONDITION ENTRY ROUTINE:
3045                    ;
3046                    ;THIS ROUTINE IS USED TO ALLOW THE OPERATOR
3047                    ;TO ENTER, AT THE TTY, THE NECESSARY PARAMETERS
3048                    ;TO RUN THE PROGRAM AS HE WISHES.  THE
3049                    ;ROUTINE IS ONLY ENTERED UPON INITIAL STARTING
3050                    ;FROM LOCATION 200(8).
3051                    ;THE MAIN PURPOSE OF THIS ROUTINE IS TO ESTABLISH
3052                    ;A TABLE OF DEVICES TO BE TESTED.  THIS TABLE
3053                    ;CONSISTS OF AN ENTRY FOR EACH OF ONE (1) TO
3054                    ;EIGHT (8) DEVICES.  EACH ENTRY CONTAINS THE
3055                    ;SLAVE NUMBER, DENSITY, PARITY, AND
3056                    ;FORMAT.  THE INFORMATION IS ENTERED
3057                    ;IN RESPONSE TO PRINTED REQUESTS AT THE TTY.
3058                    ;SLAVES MAY BE ENTERED IN ANY ORDER.  EACH
3059                    ;PARAMETER IS CHECKED FOR LEGALITY BEFORE BEING
3060                    ;SET INTO THE TABLE.
3061                    ;THE DRIVE NUMBER REQUEST WILL ALSO CHECK THE MASSBUS
3062                    ;FOR THE PRESENCE OF THE REQUESTED DRIVE.  IF IT IS NOT FOUND,
3063                    ;A NON-EXIST DRIVE MESSAGE WILL BE PRINTED AND ANOTHER DRIVE
3064                    ;REQUEST MADE.  WHEN THE DRIVE IS FOUND, THE RESPONSE IS STORED
3065                    ;AND CONTROL PASSED TO THE SLAVE SELECT ROUTINE.
3066                    ;THE SLAVE SELECT ROUTINE ALSO CHECKS FOR THE PRESENCE OF THE
3067                    ;SLAVE.  IF IT IS NOT PRESENT, A MESSAGE IS PRINTED AND ANOTHER
3068                    ;REQUEST IS ISSUED.  WHEN THE SELECTED SLAVE IS FOUND TO BE
3069                    ;PRESENT, A MESSAGE IS PRINTED IF IT IS A 7 CHANNEL DRIVE
3070                    ;TO ASSIST IN SELECTING DENSITY, PARITY, AND FORMAT.
3071                    ;UPON COMPLETION OF THE DEVICE TABLE, REQUESTS
3072                    ;ARE PRINTED FOR ENTRY OF THE NUMBER OF CHARACTERS
3073                    ;PER RECORD AND THE NUMBER OF RECORDS PER BLOCK.  THE
3074                    ;NEXT REQUEST IS FOR A PATTERN NUMBER TO BE USED
3075                    ;FOR WRITING AND CHECKING OF READ DATA.
3076                    ;FOLLOWING THE PATTERN REQUEST IS THE TAPE MARK OPTION.
3077                    ;RESPONDING TO THE REQUEST (TM=) WITH A ONE (1)
3078                    ;WILL CAUSE THE PROGRAM TO WRITE A TM AT THE
3079                    ;END OF EACH DATA BLOCK AND TO EXPECT THE
3080                    ;TM TO BE DETECTED IN EITHER READ FORWARD AND REVERSE
3081                    ;OR DURING SPACE OPERATION.  A RESPONSE OF ZERO (TM=0)
3082                    ;DISALLOWS WRITTING OF THE TM AND CAUSES THE READ
3083                    ;AND SPACE ROUTINES TO EXPECT NO TM TO BE PRESENT.
3084                    ;THE LAST REQUESTS ARE FOR ENTRY OF THE DESIRED
3085                    ;WRITE, READ, AND TURN AROUND STALLS.
3086                    ;*********************************************************
3087
3088  012212  005737  000636   TINP:   TST     TINF            ;SEE IF SHOULD INPUT FROM TTY
3089  012216  001002                   BNE     1$              ;IF SO: BR
3090  012220  000137  013700           JMP     TINP4           ;GET SWITCHES
3091  012224  005037  000676   1$:     CLR     UNP             ;CLEAR TABLE POINTER
3092  012230  005037  005054           CLR     REOTC           ;CLEAR EOT UNIT COUNTER
3093  012234  012700  000010           MOV     #10,R0          ;SET SIZE OF TABLE
3094  012240  012701  000746           MOV     #UN1,R1         ;SET START OF TABLE
3095  012244  005021           3$:     CLR     (R1)+           ;CLEAR TABLE
3096  012246  005300                   DEC     R0              ;SEE IF DONE
3097  012250  001375                   BNE     3$              ;IF NOT: BR
```

```
3098  012252  012704  025161                MOV     #MSG31,R4
3099  012256  005737  000736                TST     ASEQF           ;SEE IF AUTO SEQ
3100  012262  001402                        BEQ     4$              ;IF NOT: BR
3101  012264  012704  025121                MOV     #MSG30,R4       ;SET AUTO SEQ HDR
3102  012270  010446              4$:       MOV     R4,-(SP)        ;SAVE ADDRESS OF MESSAGE
3103  012272  000004                        TYPE                    ;TYPE MSG
3104  012274  105036                        CLRB    @(SP)+          ;DO NOT TYPE TITLE ON RESTART
3105  012276  012704  025236                MOV     #MSG31A,R4      ;TYPE INSTRUCTION
3106  012302  000004                        TYPE
3107  012304  105037  025236                CLRB    MSG31A          ;DO NOT TYPE _TARTUP INSTRUCTIONS ON RESTART
3108  012310  005737  014062                TST     SCVFL           ;SEE IF SHORT CONVERSATION
3109  012314  001067                        BNE     6$              ;IF SO: BR
3110  012316  012704  026325                MOV     #MSG74,R4
3111  012322  000004                        TYPE                    ;TYPE MSG
3112  012324  013703  000544                MOV     REGS,R3
3113  012330  104400                        TYPOCT                  ;PRINT CURRENT REG START
3114  012332  012705  000544                MOV     #REGS,R5        ;SAVE ADDRESS LOCATION
3115  012336  012701  000007                MOV     #7,R1           ;SET SIZE OF ENTRY
3116  012342  012702  176400                MOV     #176400,R2      ;SET UPPER LIMIT
3117  012346  012703  172300                MOV     #172300,R3      ;SET LOWER LIMIT
3118  012352  004737  023340                JSR     PC,TTR          ;GO GET RESPONSE
3119  012356  012704  026350                MOV     #MSG75,R4
3120  012362  000004                        TYPE                    ;TYPE MSG
3121  012364  013703  000546                MOV     VECT,R3
3122  012370  104400                        TYPOCT                  ;PRINT CURRENT VECTOR
3123  012372  012705  000546                MOV     #VECT,R5        ;SET SAVE LOCATION
3124  012376  012701  000004                MOV     #4,R1           ;SET SIZE OF ENTRY
3125  012402  012702  000224                MOV     #224,R2         ;SET UPPER LIMIT
3126  012406  012703  000150                MOV     #150,R3         ;SET LOWER LIMIT
3127  012412  004737  023340                JSR     PC,TTR          ;GO GET RESPONSE
3128  012416  013700  000546                MOV     VECT,R0         ;GET VECTOR ADDRESS
3129  012422  012720  021716                MOV     #MTINT,(R0)+    ;LOAD VECTOR WITH HANDLER ADDRESS
3130  012426  012710  000340                MOV     #340,(R0)       ;LOAD PRIORITY LEVEL
3131  012432  013700  000544                MOV     REGS,R0         ;GET STARTING REGISTER ADDRESS
3132  012436  012701  000016                MOV     #16,R1          ;SET NUMBER OF REGISTERS
3133  012442  012702  000510                MOV     #C1,R2          ;GET FIRST ADDRESS LOCATION
3134  012446  010022              5$:       MOV     R0,(R2)+        ;BUILD TABLE OF ADDRESSES
3135  012450  062700  000002                ADD     #2,R0           ;BUMP ADDRESS
3136  012454  005301                        DEC     R1              ;SEE IF DONE
3137  012456  001373                        BNE     5$              ;IF NOT: BR
3138  012460  005737  000736                TST     ASEQF           ;SEE IF AUTO SEQ
3139  012464  001403                        BEQ     6$              ;IF NOT: BR
3140  012466  005726                        TST     (SP)+           ;RESET STACK POINTER
3141  012470  000137  021734                JMP     ASEQ            ;GO TO AUTO SEQUENCE
3142  012474  012777  000040  166016  6$:   MOV     #40,@CS         ;INITIALIZE
3143  012502  012704  025772                MOV     #MSG52,R4
3144  012506  000004                        TYPE                    ;TYPE MSG
3145  012510  012705  000550                MOV     #DVN,R5         ;GET ADDRESS
3146  012514  012701  000002                MOV     #2,R1           ;SET SIZE OF RESPONSE
3147  012520  012702  000007                MOV     #7,R2           ;SET UPPER LIMIT
3148  012524  012703  000000                MOV     #0,R3           ;SET LOWER LIMIT
3149  012530  004737  023340                JSR     PC,TTR          ;GO GET DRIVE NUMBER
3150  012534  013777  000550  165756        MOV     DVN,@CS
3151  012542  005777  165742                TST     @C1             ;ACCESS DRIVE
3152  012546  032777  010000  165744        BIT     #10000,@CS      ;SEE IF NED
3153  012554  001411                        BEQ     TINPO           ;IF NOT: BR
```

```
3154  012556  012704  026262                MOV     #MSG71,R4
3155  012562  000004                       TYPE                    ;TYPE MSG
3156  012564  013704  000510                MOV     C1,R4
3157  012570  005204                       INC     R4
3158  012572  152714  000100                BISB    #100,(R4)       ;CLEAR TRE
3159  012576  000736                       BR      6$              ;RETRY DVN
3160  012600  012704  025323        TINPO:  MOV     #MSG32,R4
3161  012604  000004                       TYPE                    ;TYPE MSG
3162  012606  005037  000646                CLR     TEMP2           ;CLEAR BUFFER
3163  012612  012705  000646                MOV     #TEMP2,R5       ;SET UNIT DESCRIPTION BUFFER ADDRESS
3164  012616  012701  000002                MOV     #2,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3165  012622  012702  000007                MOV     #7,R2           ;SET MAXIMUM LIMIT
3166  012626  012703  000000                MOV     #0,R3           ;SET MINIMUM LIMIT
3167  012632  004737  023340                JSR     PC,TTR          ;GO GET UNIT NUMBER
3168  012636  005737  000644                TST     TEMP1           ;SEE IF HAVE NEW PARAMETER
3169  012642  001012                       BNE     TINPOB          ;IF SO: BR
3170  012644  005737  000676                TST     UNP             ;SEE IF FIRST ENTRY
3171  012650  001753                       BEQ     TINPO
3172  012652  013700  000676                MOV     UNP,R0
3173  012656  012760  177777  000746        MOV     #-1,UN1(R0)     ;SET END UNIT TABLE
3174  012664  000137  013264                JMP     TINP2C          ;GO GET RECORD COUNT
3175  012670  013700  000676        TINPOB: MOV     UNP,R0
3176  012674  042760  000007  000746        BIC     #7,UN1(R0)      ;CLEAR UNIT NUMBER
3177  012702  004737  014074                JSR     PC,TPOS1        ;GO LOAD UNIT NUMBER TO PROPER POSITION
3178  012706  012777  000040  165604        MOV     #40,@CS
3179  012714  013777  000550  165576        MOV     DVN,@CS
3180  012722  016077  000746  165612        MOV     UN1(R0),@TC     ;LOAD UNIT NUMBER
3181  012730  032777  002000  165600 TINPOC: BIT     #2000,@DT       ;SEE  IF SLAVE PRESENT
3182  012736  001004                       BNE     TINPOD          ;IF SO: BR
3183  012740  012704  026050                MOV     #MSG57,R4
3184  012744  000004                       TYPE                    ;TYPE MSG
3185  012746  000714                       BR      TINPO           ;REDO
3186  012750  017703  165562        TINPOD: MOV     @DT,R3          ;GET CONTENTS OF DT REG
3187  012754  042703  000007                BIC     #7,R3           ;CLEAR DRIVE TYPE #
3188  012760  022703  142050                CMP     #142050,R3      ;SEE IF 9TRK TM03,TE16
3189  012764  001410                       BEQ     TINPOE          ;IF SO: BR
3190  012766  012704  025745                MOV     #MSG50,R4       ;ILLEGAL DRIVE TYPE
3191  012772  000004                       TYPE                    ;TYPE MSG
3192  012774  017703  165536                MOV     @DT,R3
3193  013000  042703  000007                BIC     #7,R3           ;CLEAR SLAVE #
3194  013004  104400                       TYPOCT                  ;PRINT DRIVE TYPE REGISTER
3195  013006  012704  024517        TINPOE: MOV     #MSG9,R4
3196  013012  000004                       TYPE                    ;TYPE MSG
3197  013014  017703  165520                MOV     @SN,R3
3198  013020  004737  024322                JSR     PC,$NPT         ;PRINT SERIAL NUMBER
3199  013024  012704  025344        TINP1:  MOV     #MSG33,R4
3200  013030  000004                       TYPE                    ;TYPE MSG
3201  013032  005037  000646                CLR     TEMP2           ;CLEAR BUFFER
3202  013036  012701  000002                MOV     #2,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3203  013042  012702  000004                MOV     #4,R2           ;SET MAXIMUM LIMIT
3204  013046  012703  000003                MOV     #3,R3           ;SET MINIMUM LIMIT
3205  013052  004737  023340                JSR     PC,TTR          ;GO GET DENSITY
3206  013056  005737  000644                TST     TEMP1           ;SEE IF HAVE NEW PARAMETER
3207  013062  001407                       BEQ     TINP2           ;IF NOT: BR
3208  013064  042737  003400  000552        BIC     #3400,UDES      ;ELSE CLEAR OLD PARAMETER
3209  013072  012703  000010                MOV     #10,R3          ;SET POSITION FACTOR
```

```
3210  013076  004737  014064            JSR     PC,TPOS         ;GO LOAD DENSITY INTO PROPER POSITION
3211  013102  012704  025360  TINP2:    MOV     #MSG34,R4
3212  013106  000004                    TYPE                    ;TYPE MSG
3213  013110  005037  000646            CLR     TEMP2           ;CLR BUFFER
3214  013114  012701  000002            MOV     #2,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3215  013120  012702  000001            MOV     #1,R2           ;SET MAXIMUM LIMIT
3216  013124  012703  000000            MOV     #0,R3           ;SET MINIMUM LIMIT
3217  013130  004737  023340            JSR     PC,TTR          ;GO INPUT PARITY
3218  013134  005737  000644            TST     TEMP1           ;SEE IF HAVE NEW PARAMETER
3219  013140  001407                    BEQ     TINP2A          ;IF NOT: BR
3220  013142  042737  000010  000552    BIC     #10,UDES        ;ELSE CLEAR OLD PARAMETER
3221  013150  012703  000003            MOV     #3,R3           ;SET POSITION FACTOR
3222  013154  004737  014064            JSR     PC,TPOS         ;GO LOAD PARITY TO PROPER POSITION
3223  013160  012704  026013  TINP2A:   MOV     #MSG53,R4
3224  013164  000004                    TYPE                    ;TYPE MSG
3225  013166  005037  000646            CLR     TEMP2
3226  013172  012701  000003            MOV     #3,R1
3227  013176  012702  000017            MOV     #17,R2
3228  013202  012703  000000            MOV     #0,R3
3229  013206  004737  023340            JSR     PC,TTR          ;GO GET FORMAT
3230  013212  005737  000644            TST     TEMP1           ;SEE IF NEW PARAMETER
3231  013216  001407                    BEQ     TINP2B          ;IF NOT: BR
3232  013220  042737  000170  000552    BIC     #170,UDES
3233  013226  012703  000004            MOV     #4,R3
3234  013232  004737  014064            JSR     PC,TPOS
3235  013236  005237  005054  TINP2B:   INC     REOTC           ;BUMP EOT UNIT COUNTER
3236  013242  022737  000016  000676    CMP     #16,UNP         ;SEE IF DONE UNITS
3237  013250  001405                    BEQ     TINP2C          ;IF SO: BR
3238  013252  062737  000002  000676    ADD     #2,UNP          ;POINT TO NEXT UNIT
3239  013260  000137  012600            JMP     TINP0           ;ELSE LOOK FOR NEXT UNIT
3240  013264  005037  000676  TINP2C:   CLR     UNP             ;CLEAR UNIT POINTER
3241  013270  013700  005054            MOV     REOTC,R0
3242  013274  000337  005054            SWAB    REOTC
3243  013300  110037  005054            MOVB    R0,REOTC        ;SET UNIT EOT COUNTER
3244  013304  012704  025373  TINP3:    MOV     #MSG35,R4
3245  013310  000004                    TYPE                    ;TYPE MSG
3246  013312  013703  000554            MOV     RCNT,R3
3247  013316  104400                    TYPOCT                  ;PRINT RECORD COUNT
3248  013320  012705  000554            MOV     #RCNT,R5        ;SET RECORD COUNT ADDRESS
3249  013324  012701  000007            MOV     #7,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3250  013330  012702  177777            MOV     #-1,R2          ;SET MAXIMUM LIMIT
3251  013334  012703  000001            MOV     #1,R3           ;SET MINIMUM LIMIT
3252  013340  004737  023340            JSR     PC,TTR          ;GO GET RECORD COUNT
3253  013344  013737  000554  000632    MOV     RCNT,RCSAV      ;SAVE RECORD COUNT
3254  013352  012704  025414            MOV     #MSG36,R4
3255  013356  000004                    TYPE                    ;TYPE MSG
3256  013360  005437  000556            NEG     FMCNT
3257  013364  013703  000556            MOV     FMCNT,R3
3258  013370  104400                    TYPOCT                  ;PRINT CHAR COUNT
3259  013372  012705  000556            MOV     #FMCNT,R5       ;SET CHARACTER COUNT ADDRESS
3260  013376  012701  000007            MOV     #7,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3261  013402  012702  004000            MOV     #4000,R2        ;SET MAXIMUM LIMIT
3262  013406  012703  000004            MOV     #4,R3           ;SET MINIMUM LIMIT
3263  013412  004737  023340            JSR     PC,TTR          ;GO GET CHARACTER COUNT
3264  013416  005437  000556            NEG     FMCNT           ;SET TO TWO'S COMPLIMENT
3265  013422  013737  000556  000634    MOV     FMCNT,FCSAV     ;SAVE FRAME COUNT
```

```
3266  013430  012704  025433          MOV     #MSG37,R4       ;PRINT PATTERN NUMBER REQUEST
3267  013434  000004                  TYPE                    ;TYPE MSG
3268  013436  013703  000560          MOV     PATRN,R3
3269  013442  104400                  TYPOCT                  ;PRINT PATTERN
3270  013444  005037  014432          CLR     DOFL            ;CLEAR EXTERNAL DATA FLAG
3271  013450  012705  000560          MOV     #PATRN,R5       ;SET PATTERN NUMBER ADDRESS
3272  013454  012701  000003          MOV     #3,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3273  013460  012702  000015          MOV     #15,R2          ;SET MAXIMUM LIMIT
3274  013464  012703  000000          MOV     #0,R3           ;SET MINIMUM LIMIT
3275  013470  004737  023340          JSR     PC,TTR          ;GO GET PATTERN NUMBER
3276  013474  012704  026210          MOV     #MSG69,R4
3277  013500  000004                  TYPE                    ;TYPE MSG
3278  013502  013703  000564          MOV     TMEX,R3
3279  013506  104400                  TYPOCT                  ;PRINT CURRENT TM FLAG SETTING
3280  013510  012705  000564          MOV     #TMEX,R5        ;GET TM FLAG ADDRESS
3281  013514  012701  000002          MOV     #2,R1           ;SET SIZE OF RESPONSE
3282  013520  012702  000001          MOV     #1,R2           ;SET UPPER LIMIT
3283  013524  012703  000000          MOV     #0,R3           ;SET LOWER LIMIT
3284  013530  004737  023340          JSR     PC,TTR          ;TM 1=YES
3285  013534  012704  024651          MOV     #MSG21,R4
3286  013540  000004                  TYPE                    ;TYPE MSG
3287  013542  013703  000570          MOV     INTRF,R3
3288  013546  104400                  TYPOCT                  ;PRINT CURRENT SETTING
3289  013550  012705  000570          MOV     #INTRF,R5       ;GET FLAG ADDRESS
3290  013554  012701  000002          MOV     #2,R1           ;SET SIZE OF RESPONSE
3291  013560  012702  000001          MOV     #1,R2           ;SET UPPER LIMIT
3292  013564  012703  000000          MOV     #0,R3           ;SET LOWER LIMIT
3293  013570  004737  023340          JSR     PC,TTR          ;GO GET RESPONSE
3294  013574  012704  025456          MOV     #MSG38,R4
3295  013600  000004                  TYPE                    ;TYPE MSG
3296  013602  013703  000572          MOV     SPFLG,R3
3297  013606  104400                  TYPOCT                  ;PRINT CURRENT SETTING
3298  013610  012705  000572          MOV     #SPFLG,R5       ;SET ADDRESS OF FLAG
3299  013614  012701  000002          MOV     #2,R1           ;SET SIZE OF RESPONSE
3300  013620  012702  000001          MOV     #1,R2           ;SET UPPER LIMIT
3301  013624  012703  000000          MOV     #0,R3           ;SET LOWER LIMIT
3302  013630  004737  023340          JSR     PC,TTR          ;GO GET RESPONSE
3303  013634  012704  025476  TINP3A:  MOV     #MSG39,R4
3304  013640  000004                  TYPE                    ;TYPE MSG
3305  013642  013703  000566          MOV     CRCC,R3
3306  013646  104400                  TYPOCT
3307  013650  012705  000566          MOV     #CRCC,R5
3308  013654  012701  000002          MOV     #2,R1
3309  013660  012702  000001          MOV     #1,R2
3310  013664  012703  000000          MOV     #0,R3
3311  013670  004737  023340          JSR     PC,TTR
3312  013674  004737  023204          JSR     PC,GTSWR        ;GET SWITCHES
3313  013700  005737  014062  TINP4:   TST     SCVFL           ;BRANCH IF SHORT CONVERSATION
3314  013704  001063                  BNE     TINPX
3315  013706  005737  000636  1$:     TST     TINF            ;BRANCH IF NO TTY INPUT
3316  013712  001460                  BEQ     TINPX
3317  013714  012704  025536          MOV     #MSG40,R4
3318  013720  000004                  TYPE                    ;TYPE MSG
3319  013722  013703  000574          MOV     RSTAL,R3
3320  013726  104400                  TYPOCT                  ;PRINT READ STALL
3321  013730  012705  000574          MOV     #RSTAL,R5       ;SET READ STALL ADDRESS
```

```
3322  013734  012701  000007          MOV     #7,R1           ;SET NUMBER OF CHARACTER  TO INPUT
3323  013740  012702  177777          MOV     #-1,R2          ;SET MAXIMUM LIMIT
3324  013744  012703  000001          MOV     #1,R3           ;SET MINIMUM LIMIT
3325  013750  004737  023340          JSR     PC,TTR          ;GO GET READ STALL
3326  013754  012704  025565          MOV     #MSG41,R4
3327  013760  000004                  TYPE                    ;TYPE MSG
3328  013762  013703  000576          MOV     WSTAL,R3
3329  013766  104400                  TYPOCT                  ;PRINT READ STALL
3330  013770  012705  000576          MOV     #WSTAL,R5       ;SET WRITE STALL ADDRESS
3331  013774  012701  000007          MOV     #7,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3332  014000  012702  177777          MOV     #-1,R2          ;SET MAXIMUM LIMIT
3333  014004  012703  000001          MOV     #1,R3           ;SET MINIMUM LIMIT
3334  014010  004737  023340          JSR     PC,TTR          ;GO GET WRITE STALL
3335  014014  012704  025577          MOV     #MSG42,R4
3336  014020  000004                  TYPE                    ;TYPE MSG
3337  014022  013703  000600          MOV     TSTAL,R3
3338  014026  104400                  TYPOCT                  ;PRINT TA STALL
3339  014030  012705  000600          MOV     #TSTAL,R5       ;SET TURN AROUND STALL ADDRESS
3340  014034  012701  000007          MOV     #7,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3341  014040  012702  177777          MOV     #-1,R2          ;SET MAXIMUM LIMIT
3342  014044  012703  000001          MOV     #1,R3           ;SET MINIMUM LIMIT
3343  014050  004737  023340          JSR     PC,TTR          ;GO GET TURN AROUND STALL
3344  014054  005037  014062  TINPX:  CLR     SCVFL           ;CLEAR SHORT CONVERSATION FLAG
3345  014060  000207                  RTS     PC              ;EXIT
3346  014062  000000  SCVFL:  0                               ;SHORT CONVERSATION FLAG
3347
3348                                  ;UNIT DESCRIPTION POSITIONING SUBROUTINE**********
3349
3350  014064  006337  000646  TPOS:   ASL     TEMP2           ;POSITION CHARACTER
3351  014070  005303                  DEC     R3              ;SEE IF DONE
3352  014072  001374                  BNE     TPOS            ;IF NOT: BR
3353  014074  013700  000676  TPOS1:  MOV     UNP,R0          ;LOAD UNIT POINTER
3354  014100  053760  000646  000746  BIS     TEMP2,UN1(R0)   ;LOAD CHARACTER INTO UN1(R0)
3355  014106  000207                  RTS     PC              ;EXIT
3356
```

```
                                        ;*******************************************************
                                        ;DATA SETUP ROUTINE:
                                        ;
                                        ;THIS ROUTINE IS USED TO GENERATE INTO THE ENTIRE
                                        ;WRITE BUFFER (4000 OCTAL CHARACTERS) THE DATA PATTERN
                                        ;SELECTED BY THE OPERATOR.  THERE ARE 15 (8) FIXED
                                        ;DATA PATTERNS AVAILABLE AND ONE SELECTION (DATA PATTERN 0)
                                        ;WHICH WILL READ ANY PATTERN PRESENTED AT THE
                                        ;HIGH SPEED PAPER TAPE READER.  THIS TAPE MUST BE PREPARED
                                        ;BY USING THE PROGRAM CALLED DTC. (MAINDEC-11-DZTUF-A-D)
                                        ;RANDOM DATA MAY ALSO BE USED VIA CONSOLE
                                        ;SWITCH EIGHT (8).
                                        ;THIS ROUTINE IS ALSO USED TO CLEAR OUT THE
                                        ;READ BUFFER (4000 OCTAL CHARACTERS) BEFORE EACH
                                        ;RECORD IS READ.
                                        ;*******************************************************

014110  005737  015022          DSUP:   TST     RDFL            ;SEE IF DID RANDOM DATA
014114  001044                          BNE     DS2A            ;IF NOT: BR
014116  005737  000736          DSO:    TST     ASEQF           ;SEE IF AUTO SEQ
014122  001406                          BEQ     DSOC            ;IF NOT: BR
014124  005737  000560                  TST     PATRN           ;SEE IF AUTO RANDOM
014130  100003                          BPL     DSOC            ;IF NOT: BR
014132  004737  014760                  JSR     PC,DATR         ;ELSE GO GENERATE RANDOM DATA
014136  000207                          RTS     PC              ;RETURN
014140  023737  000560  014270  DSOC:   CMP     PATRN,PATS      ;SEE IF NEW PATTERN
014146  001014                          BNE     DSOA            ;IF SO: BR
014150  013703  000552                  MOV     UDES,R3         ;GET UNIT DESCRIPTION
014154  042703  177767                  BIC     #177767,R3      ;MASK EVEN PARITY
014160  023703  014272                  CMP     PARS,R3         ;SEE IF SAME AS LAST TIME
014164  001404                          BEQ     DSOB            ;IF SO: BR
014166  010337  014272                  MOV     R3,PARS         ;SAVE PARITY
014172  004737  015024                  JSR     PC,CRCLRC       ;GO GENERATE EXPT CRC/LRC
014176  000207                  DSOB:   RTS     PC
014200  012703  027326          DSOA:   MOV     #WDATA,R3       ;R3 = ADDRS OF WRITE BUFFER
014204  013701  000560                  MOV     PATRN,R1        ;R1 = PATTERN SELECTOR
014210  010137  014270                  MOV     R1,PATS
014214  062701  000001                  ADD     #1,R1           ;BUMP POINTER
014220  006301                          ASL     R1              ;MAKE PATTERN SELECTOR EVEN
014222  004771  002770                  JSR     PC,@DATBL(R1)   ;GO GENERATE PATTERN
014226  004737  015024          DS2A:   JSR     PC,CRCLRC       ;GO GENERATE EXPT CRC/LRC
014232  013702  000556          DS3:    MOV     FMCNT,R2        ;R2=BUFFER SIZE
014236  006202                          ASR     R2              ;R2=FRAME CNT/2
014240  012701  033334                  MOV     #RDATA,R1       ;R1=READ DATA START
014244  005021                  DS4:    CLR     (R1)+           ;CLEAR BUFFER
014246  005202                          INC     R2              ;SEE IF DONE ALL
014250  001375                          BNE     DS4             ;IF NOT: BR
014252  013737  000552  014272          MOV     UDES,PARS       ;GET UNIT DESCRIPTION
014260  042737  177767  014272          BIC     #177767,PARS    ;MASK PARITY
014266  000207                          RTS     PC              ;EXIT
014270  177777          PATS:   -1                              ;PATTERN NUMBER SAVE
014272  000000          PARS:   0
```

```
3412
3413                                            ;EXTERNAL DATA INPUT FROM H/S READER (256 CHARACTER MAXIMUM)
3414
3415  014274  005737  014432          DATO:   TST     DOFL            ;SEE IF SHOULD DO EXTERNAL INPUT
3416  014300  001352                  DATOA:  BNE     DS2A            ;IF NOT: BR
3417  014302  012737  000001  014432  DATOB:  MOV     #1,DOFL         ;SET EXTERNAL FLAG
3418  014310  005077  164306          DATOC:  CLR     @PRS            ;CLEAR READER STATUS
3419  014314  005037  000644          DATOD:  CLR     TEMP1           ;CLEAR FOR USE AS CHARACTER FLAG
3420  014320  052777  000001  164274  DATOE:  BIS     #1,@PRS         ;START READER
3421  014326  105777  164270          DATOF:  TSTB    @PRS            ;SEE IF DONE
3422  014332  100375                          BPL     DATOB           ;IF NOT : BR
3423  014334  005001                          CLR     R1              ;CLEAR SAVE LOCATION
3424  014336  117701  164262                  MOVB    @PRB,R1         ;SAVE CHARACTER
3425  014342  005737  000644                  TST     TEMP1           ;SEE IF HAVE FOUND START CHARACTER
3426  014346  001011                          BNE     DATOC           ;IF SO : BR
3427  014350  105701                          TSTB    R1              ;SEE IF CHARACTER IS 0
3428  014352  001762                          BEQ     DATOA           ;IF SO : BR
3429  014354  012737  000001  000644          MOV     #1,TEMP1        ;ELSE SET CHARACTER FOUND FLAG
3430  014362  010137  000646                  MOV     R1,TEMP2        ;SAVE DATA SIZE
3431  014366  010102                          MOV     R1,R2           ;SAVE DATA SIZE
3432  014370  000753                          BR      DATOA           ;GO GET FIRST DATA CHAR
3433  014372  110123          DATOC:  MOVB    R1,(R3)+        ;LOAD BUFFER
3434  014374  005302                          DEC     R2              ;SEE IF READ ALL
3435  014376  001350                          BNE     DATOA           ;IF NOT : BR
3436  014400  012701  027326          DATOD:  MOV     #WDATA,R1       ;R1 = START OF WRITE BUFFER
3437  014404  013702  000646                  MOV     TEMP2,R2        ;R2 = SIZE OF DATA FIELD
3438  014410  112123          DATOE:  MOVB    (R1)+,(R3)+     ;REPEAT LOAD OF DATA FIELD
3439  014412  022703  033334                  CMP     #RDATA,R3       ;SEE IF DONE
3440  014416  003002                          BGT     DATOF           ;IF NOT: BR
3441  014420  000137  014226                  JMP     DS2A            ;EXIT
3442  014424  005302          DATOF:  DEC     R2              ;SEE IF AT END OF DATA FIELD
3443  014426  001370                          BNE     DATOE           ;IF NOT : BR
3444  014430  000763                          BR      DATOD           ;ELSE RESTART FILL
3445  014432  000000          DOFL:   0                               ;EXTERNAL DATA FLAG=1 IF ALREADY DONE
3446
```

```
3447                                        ;ALL ONES**********
3448
3449  014434  012701  177777    DAT1:   MOV     #-1,R1          ;R1=DATA
3450  014440  012702  002002    DAT1A:  MOV     #2002,R2        ;R2=WORD COUNT +2
3451  014444  010123    1$:     MOV     R1,(R3)+        ;LOAD BUFFER
3452  014446  005302            DEC     R2              ;SEE IF DONE
3453  014450  001375            BNE     1$              ;IF NOT: BR
3454  014452  000207            RTS     PC
3455
3456                                        ;ALL ZEROS**********
3457
3458  014454  005001    DAT2:   CLR     R1              ;R1=DATA
3459  014456  000770            BR      DAT1A           ;LOAD BUFFER
3460
3461                                        ;WALKING ONE**********
3462
3463  014460  012701  000001    DAT3:   MOV     #1,R1           ;R1=DATA
3464  014464  000241            CLC
3465  014466  012702  004004    DAT3A:  MOV     #4004,R2        ;R2=CHARACTER COUNT+4
3466  014472  110123    1$:     MOVB    R1,(R3)+        ;LOAD BUFFER
3467  014474  106101            ROLB    R1              ;SET NEXT CHARACTER
3468  014476  005302            DEC     R2              ;SEE IF DONE
3469  014500  001374            BNE     1$              ;IF NOT: BR
3470  014502  000207            RTS     PC
3471
3472                                        ;WALKING ZERO**********
3473
3474  014504  012701  000376    DAT4:   MOV     #376,R1         ;R1=START OF DATA
3475  014510  000261            SEC
3476  014512  000765            BR      DAT3A           ;LOAD BUFFER
3477
3478                                        ;ALTERNATING ONE/ZERO**********
3479
3480
3481  014514  012701  052525    DAT5:   MOV     #52525,R1       ;R1=DATA
3482  014520  000747            BR      DAT1A           ;LOAD BUFFER
3483
3484                                        ;ALTERNATING ZERO/ONE**********
3485
3486  014522  012701  125252    DAT6:   MOV     #125252,R1      ;R1=DATA
3487  014526  000744            BR      DAT1A           ;LOAD BUFFER
3488
3489                                        ;ONE/ZERO IN ALTERNATING WORDS**********
3490
3491  014530  012701  125252    DAT7:   MOV     #125252,R1      ;SET WORD 1
3492  014534  012702  052525            MOV     #52525,R2       ;SET WORD 2
3493  014540  012704  001002            MOV     #1002,R4        ;SET NUMBER OF ENTRIES
3494  014544  010123    1$:     MOV     R1,(R3)+        ;LOAD WORD 1
3495  014546  010223            MOV     R2,(R3)+        ;LOAD WORD 2
3496  014550  005304            DEC     R4              ;SEE IF DONE
3497  014552  001374            BNE     1$              ;IF NOT: BR
3498  014554  000207            RTS     PC
3499
```

```
3510
3511
3512
3513
3514  014606  005001          DAT11:  CLR     R1              ;R1=STARTING DATA
3515  014610  012702  004004          MOV     #4004,R2        ;R2=CHARACTER COUNT+4
3516  014614  110123          1$:     MOVB    R1,(R3)+        ;LOAD BUFFER
3517  014616  105201                  INCB    R1              ;BUMP DATA
3518  014620  005302                  DEC     R2              ;SEE IF DONE
3519  014622  001374                  BNE     1$              ;IF NOT: BR
3520  014624  000207                  RTS     PC              ;RETURN
3521
3522
3523
3524  014626  012701  000377  DAT12:  MOV     #377,R1         ;R1=STARTING DATA
3525  014632  012702  004004          MOV     #4004,R2        ;R2=CHARACTER COUNT+4
3526  014636  110123          1$:     MOVB    R1,(R3)+        ;LOAD BUFFER
3527  014640  105301                  DECB    R1              ;BUMP DATA
3528  014642  005302                  DEC     R2              ;SEE IF DONE
3529  014644  001374                  BNE     1$              ;IF NOT: BR
3530  014646  000207                  RTS     PC              ;RETURN
3531
3532
3533
3534  014650  012701  000377  DAT13:  MOV     #377,R1         ;R1 = DATA
3535  014654  000137  014440          JMP     DAT1A           ;LOAD BUFFER
3536
3537
3538
3539  014660  012702  002002  DAT14:  MOV     #2002,R2        ;SET BUFFER SIZE
3540  014664  012701  000376          MOV     #376,R1         ;SET WALK BASE
3541  014670  000261                  SEC
3542  014672  010113          1$:     MOV     R1,(R3)         ;LOAD WALK BYTE
3543  014674  042723  177400          BIC     #177400,(R3)+   ;CLEAR HIGH BYTE
3544  014700  106101                  ROLB    R1              ;WALK ZERO BIT
3545  014702  005302                  DEC     R2
3546  014704  001372                  BNE     1$              ;FILL BUFFER
3547  014706  000207                  RTS     PC              ;RETURN
3548
```

```
3549                                            ;AUTO SEQUENCE PATTERN*********
3550
3551  014710  012702  000200    DAT15:  MOV     #200,R2         ;SET NUMBER OF ENTRIES
3552  014714  012701  014740    1$:     MOV     #APATS,R1       ;SET START OF PATTERN
3553  014720  012704  000010            MOV     #10,R4          ;SET SIZE OF PATTERN
3554  014724  012123            2$:     MOV     (R1)+,(R3)+     ;FILL BUFFER
3555  014726  005304                    DEC     R4              ;SEE IF DONE PATTERN
3556  014730  001375                    BNE     2$              ;IF NOT: BR
3557  014732  005302                    DEC     R2              ;SEE IF DONE BUFER
3558  014734  001367                    BNE     1$              ;IF NOT: BR
3559  014736  000207                    RTS     PC              ;RETURN
3560
3561  014740  000000    APATS:  0
3562  014742  177400            177400
3563  014744  000377            377
3564  014746  000000            0
3565  014750  177777            -1
3566  014752  000377            377
3567  014754  177400            177400
3568  014756  177777            -1
3569
3570                                            ;RANDOM DATA GENERATOR SUBROUTINE**********
3571
3572  014760  013704  000556    DATR:   MOV     FMCNT,R4        ;SET NUMBER OF FRAMES
3573  014764  012703  027326            MOV     #WDATA,R3       ;SET ADDRESS OF START OF BUFFER
3574  014770  012701  177777            MOV     #-1,R1          ;SET HIGH LIMIT
3575  014774  005002                    CLR     R2              ;SET LOW LIMIT
3576  014776  004737  023152    1$:     JSR     PC,RANG         ;GO GENERATE NUMBER
3577  015002  013723  000630            MOV     RANSAV,(R3)+    ;LOAD BUFFER
3578  015006  005204                    INC     R4              ;SEE IF DONE WHOLE BUFFER
3579  015010  001372                    BNE     1$              ;IF NOT: BR
3580  015012  012737  000001  015022    MOV     #1,RDFL         ;SET RANDOM DATA FLAG
3581  015020  000207                    RTS     PC              ;EXIT
3582  015022  000000            RDFL:   0                       ;RANDOM DATA SELECT FLAG
```

```
3583
3584                                    ;*********************************.**********************
3585                                    ;CRC/LRC CHARACTER BUILD;
3586                                    ;
3587                                    ;THIS ROUTINE WILL CONSTRUCT AND SAVE THE EXPECTED
3588                                    ;CRC AND LRC CHARACTERS ACCORDING TO DATA AND
3589                                    ;RECORD SIZE IF OPERATING IN NRZ MODE
3590                                    ;*************************************************************
3591
3592  015024  013700  000556   CRCLRC: MOV     FMCNT,R0        ;SET RECORD SIZE
3593  015030  005400                   NEG     R0
3594  015032  012701  027326           MOV     #WDATA,R1       ;SET START OF BUFFER
3595  015036  005037  015360           CLR     XORS
3596  015042  111104           CL0:    MOVB    (R1),R4         ;GET CHARACTER
3597  015044  004737  015232           JSR     PC,CLP          ;GO GET PARITY OF CHARACTER
3598  015050  004737  015334           JSR     PC,XOR          ;XOR CHARACTER
3599  015054  000241                   CLC
3600  015056  006004                   ROR     R4              ;ROTATE 1 RIGHT
3601  015060  103014                   BCC     CL2             ;IF NO CARRY: BR
3602  015062  052704  000400           BIS     #400,R4         ;SET BIT NINE
3603  015066  000241                   CLC
3604  015070  010405           CL1:    MOV     R4,R5           ;SAVE CHARACTER
3605  015072  042705  177703           BIC     #177703,R5
3606  015076  005105                   COM     R5
3607  015100  042705  177703           BIC     #177703,R5
3608  015104  042704  000074           BIC     #74,R4
3609  015110  050504                   BIS     R5,R4           ;COMPLIMENT BITS 2,3,4,5
3610  015112  010437  015360   CL2:    MOV     R4,XORS
3611  015116  005300                   DEC     R0
3612  015120  001350                   BNE     CL0             ;BRANCH IF NOT LAST CHAR
3613  015122  013704  015360   CLLAST: MOV     XORS,R4
3614  015126  005137  015360           COM     XORS
3615  015132  042737  177050  015360   BIC     #177050,XORS
3616  015140  042704  177727           BIC     #177727,R4      ;COMPLIMENT ALL BUT BITS 3&5
3617  015144  050437  015360           BIS     R4,XORS
3618  015150  013737  015360  015362   MOV     XORS,EXCRC      ;SAVE EXPECTED CRC
3619  015156  013700  000556           MOV     FMCNT,R0
3620  015162  005400                   NEG     R0
3621  015164  012701  027326           MOV     #WDATA,R1       ;DO EXPT LRC
3622  015170  005037  015360           CLR     XORS
3623  015174  111104           CL3:    MOVB    (R1),R4
3624  015176  004737  015232           JSR     PC,CLP          ;GET PARITY
3625  015202  004737  015334           JSR     PC,XOR          ;XOR CHARACTER
3626  015206  005300                   DEC     R0
3627  015210  001371                   BNE     CL3             ;DO ALL FOR LRC
3628  015212  013704  015362           MOV     EXCRC,R4
3629  015216  004737  015334           JSR     PC,XOR          ;XOR CRC TO DATA
3630  015222  013737  015360  015364   MOV     XORS,EXLRC      ;SAVE EXPT LRC
3631  015230  000207                   RTS     PC              ;RETURN
3632  015232  005704           CLP:    TST     R4              ;SEE IF 0 CHAR
3633  015234  001010                   BNE     CLPE            ;IF NOT: BR
3634  015236  032737  000010  000552   BIT     #10,UDES        ;SEE IF EVEN PARITY
3635  015244  001404                   BEQ     CLPE            ;IF NOT: BR
3636  015246  012704  000420           MOV     #420,R4         ;SET 0 CHAR EVEN PARITY
3637  015252  005201                   INC     R1              ;BUMP POINTER
3638  015254  000207                   RTS     PC              ;RETURN
```

```
3638  015256  005046                  CLPE:   CLR     -(SP)           ;CLEAR WORLING SPACE ON STACK
3639  015260  106304                  1$:     ASLB    R4              ;SHIFT DATA
3640  015262  005516                          ADC     (SP)            ;ADDUP # OF 1 BITS
3641  015264  105704                          TSTB    R4              ;BRANCH IF ALL 0"S LEFT
3642  015266  001374                          BNE     1$
3643  015270  112104                          MOVB    (R1)+,R4
3644  015272  042704  177400                BIC     #177400,R4
3645  015276  106026                          RORB    (SP)+           ;BRANCH IF ODD # OF 1 BITS
3646  015300  103405                          BCS     CLP2
3647  015302  032737  000010  000552        BIT     #10,UDES        ;SEE IF SHOULD BE EVEN PARITY
3648  015310  001406                          BEQ     CLP3            ;IF NOT: BR
3649  015312  000207                          RTS     PC              ;ELSE EXIT
3650  015314  032737  000010  000552  CLP2:  BIT     #10,UDES        ;SEE IF SHOULD BE ODD PARITY
3651  015322  001001                          BNE     CLP3            ;IF NOT: BR
3652  015324  000207                          RTS     PC              ;ELSE EXIT
3653  015326  052704  000400          CLP3:   BIS     #400,R4         ;SET PARITY BIT
3654  015332  000207                          RTS     PC
3655
3656  015334  010446                  XOR:    MOV     R4,-(SP)
3657  015336  043716  015360                BIC     XORS,(SP)
3658  015342  040437  015360                BIC     R4,XORS         ;XOR SUBROUTINE: R4 WITH XORS
3659  015346  052637  015360                BIS     (SP)+,XORS
3660  015352  013704  015360                MOV     XORS,R4
3661  015356  000207                          RTS     PC
3662
3663  015360  000000                  XORS:   0                       ;XOR SAVE
3664  015362  000000                  EXCRC:  0                       ;EXPECTED CRC
3665  015364  000000                  EXLRC:  0                       ;EXPECTED LRC
3666
```

```
3668
3669                                    ;*********************************************************
3670                                    ;DATA CHECK SUBROUTINE:
3671                                    ;
3672                                    ;THIS SUBROUTINE IS USED TO COMPARE EACH CHARACTER
3673                                    ;OF DATA READ FROM TAPE WITH THE EXPECTED CHARACTER.
3674                                    ;ANY ERROR DETECTED WILL CAUSE CONTROL TO BE
3675                                    ;PASSED TO AN ERROR PRINT SUBROUTINE AND A
3676                                    ;SUBROUTINE TO ACCUMULATE THE NUMBER OF BITS
3677                                    ;DROPPED AND PICKED UP FROM EACH CHARACTER.
3678                                    ;THE NUMBER OF READ ERRORS IS ALSO ACCUMULATED.
3679                                    ;DATA CHECKING MAY BE TERMINATED BY USE OF
3680                                    ;CONSOLE SWITCH THIRTEEN (13).
3681                                    ;*********************************************************
3682
3683  015366  005037  000660        DCHK:   CLR     BBC             ;CLEAR BAD RECORD CNTR
3684  015372  005037  000706                CLR     DERFL           ;CLEAR DATA ERROR FLAG
3685  015376  013705  000556                MOV     FMCNT,R5        ;LOAD CHAR COUNT
3686  015402  032737  000020  000552        BIT     #20,UDES        ;SEE IF CORE DUMP
3687  015410  001401                        BEQ     DCHK0           ;IF NOT: BR
3688  015412  006205                        ASR     R5              ;R5 = FC/2
3689  015414  012701  027326        DCHK0:  MOV     #WDATA,R1       ;SET WRITE DATA ADDR
3690  015420  012702  033334                MOV     #RDATA,R2       ;SET READ DATA ADDR
3691  015424  032737  000010  000552        BIT     #10,UDES        ;SEE IF EVEN PARITY
3692  015432  001430                        BEQ     DFOC0           ;IF NOT: BR
3693  015434  032737  000020  000552        BIT     #20,UDES        ;SEE IF CORE DUMP PARITY
3694  015442  001024                        BNE     DFOC0           ;IF SO: BR
3695  015444  032737  002000  000552        BIT     #2000,UDES      ;SEE IF PE MODE
3696  015452  001020                        BNE     DFOC0           ;IF SO: BR
3697  015454  105711                DFOF:   TSTB    (R1)            ;SEE IF 0 CHAR
3698  015456  001404                        BEQ     DFOD            ;IF SO: BR
3699  015460  005201                        INC     R1              ;BUMP POINTER
3700  015462  005205                DFOE:   INC     R5              ;SEE IF DONE
3701  015464  001373                        BNE     DFOF            ;IF NOT: BR
3702  015466  000406                        BR      DFOC            ;ELSE CONTINUE
3703  015470  112721  000020        DFOD:   MOVB    #20,(R1)+       ;SET 20 IN PLACE OF 0
3704  015474  012737  177777  014270        MOV     #-1,PATS        ;SET PATTERN GENERATE FLAG
3705  015502  000767                        BR      DFOE
3706  015504  013705  000556        DFOC:   MOV     FMCNT,R5        ;RESET CHAR CNT
3707  015510  012701  027326                MOV     #WDATA,R1       ;RESET DATA ADDRESS
3708  015514  032737  010000  000562  DFOC0: BIT     #10000,RDCMD    ;SEE IF READ REVERSE
3709  015522  001462                        BEQ     DF0             ;IF NOT: BR
3710  015524  013704  000556        DF08:   MOV     FMCNT,R4        ;GET FRAME COUNT
3711  015530  005404                        NEG     R4              ;SET TO WHOLE NUMBER
3712  015532  032737  000020  000552        BIT     #20,UDES        ;SEE IF CORE DUMP
3713  015540  001402                        BEQ     DF080           ;IF NOT: BR
3714  015542  000241                        CLC
3715  015544  006004                        ROR     R4              ;SET TO FC/2
3716  015546  060401                DF080:  ADD     R4,R1           ;POINT TO START OF WRITE DATA
3717  015550  060402                        ADD     R4,R2           ;POINT TO START OF READ DATA
3718  015552  032737  000001  000556        BIT     #1,FMCNT        ;SEE IF ODD FRAME COUNT
3719  015560  001401                        BEQ     DF0A            ;IF NOT: BR
3720  015562  105722                        TSTB    (R2)+           ;BUMP POINTER
3721  015564  032737  000020  000552  DF0A:  BIT     #20,UDES        ;SEE IF CORE DUMP
3722  015572  001431                        BEQ     DF0A4           ;IF NOT: BR
3723  015574  000241                        CLC
```

```
3724  015576  132742  000001          BITB    #1,-(R2)        ;SEE IF BIT 0 = 1
3725  015602  001401                  BEQ     DFJA0           ;IF NOT: BR
3726  015604  000261                  SEC
3727  015606  106012          DFOA0:  RORB    (R2)
3728  015610  000241                  CLC
3729  015612  132712  000001          BITB    #1,(R2)
3730  015616  001401                  BEQ     DFOA1
3731  015620  000261                  SEC
3732  015622  106012          DFOA1:  RORB    (R2)            ;POSITION BITS FOR REVERSE CORE DUMP
3733  015624  000241                  CLC
3734  015626  132712  000001          BITB    #1,(R2)
3735  015632  001401                  BEQ     DFOA2
3736  015634  000261                  SEC
3737  015636  106012          DFOA2:  RORB    (R2)
3738  015640  000241                  CLC
3739  015642  132712  000001          BITB    #1,(R2)
3740  015646  001401                  BEQ     DFOA3
3741  015650  000261                  SEC
3742  015652  106012          DFOA3:  RORB    (R2)
3743  015654  005202                  INC     R2              ;RESET POINTER
3744  015656  124142          DFOA4:  CMPB    -(R1),-(R2)     ;TEST DATA CHARACTER
3745  015660  001010                  BNE     DF1             ;IF NOT GOOD: BR
3746  015662  105037  000660          CLRB    BBC             ;CLEAR BAD RECORD COUNTER
3747  015666  000411                  BR      DF2
3748  015670  122122          DFO:    CMPB    (R1)+,(R2)+     ;CHECK DATA
3749  015672  001003                  BNE     DF1             ;IF BAD: BR
3750  015674  105037  000660          CLRB    BBC             ;CLEAR BAD RECORD CNTR
3751  015700  000404                  BR      DF2
3752  015702  004737  016514  DF1:    JSR     PC,DRPKF        ;GO GET DROPS AND PICKS
3753  015706  004737  016000          JSR     PC,DERR         ;GO DO PRINT
3754  015712  005205          DF2:    INC     R5              ;BUMP CHAR CNTR
3755  015714  001405                  BEQ     DF3             ;IF DONE ALL: BR
3756  015716  032737  010000  000562  BIT     #10000,RDCMD    ;SEE IF READ REVERSE
3757  015724  001761                  BEQ     DFO             ;IF NOT: BR
3758  015726  000716                  BR      DFOA            ;ELSE CONTINUE READ REV
3759  015730  005037  000666  DF3:    CLR     HDRFL           ;CLEAR HEADER FLAG
3760  015734  005737  000706          TST     DERFL           ;SEE IF HAD DATA ERROR
3761  015740  001416                  BEQ     DFX             ;IF NOT: BR
3762  015742  005737  000710          TST     SERFL
3763  015746  001013                  BNE     DFX             ;IF NOT DATA ERROR ONLY: BR
3764  015750  013704  000676          MOV     UNP,R4
3765  015754  032737  010000  000562  BIT     #10000,RDCMD    ;SEE IF READ REVERSE
3766  015762  001003                  BNE     DF4             ;IF SO: BR
3767  015764  005264  001130          INC     DATER1(R4)      ;BUMP DATA ERROR FORWARD COUNTER
3768  015770  000402                  BR      DFX
3769  015772  005264  001170  DF4:    INC     DEREV1(R4)      ;BUMP REVERSE DATA ERROR
3770  015776  000207          DFX:    RTS     PC              ;EXIT
3771
```

```
3772
3773                                    ;*********************************************************
3774                                    ;DATA ERROR SUBROUTINE:
3775                                    ;
3776                                    ;THIS SUBROUTINE IS USED TO PRINT OUT ANY
3777                                    ;ERRORS FOUND DURING THE DATA CHECK.
3778                                    ;EACH CHARACTER FOUND BAD WILL BE PRINTED
3779                                    ;IN BIT FORMAT ALONG WITH ITS EXPECTED CHARACTER.
3780                                    ;AN ERROR HEADER CONSISTING OF THE UNIT NUMBER,
3781                                    ;BLOCK NUMBER, RECORD NUMBER, SIZE OF RECORD, AND
3782                                    ;ERROR TYPE (READ FORWARD, READ REVERSE, WRITE, ETC)
3783                                    ;IS PRINTED ONLY ONCE FOR EACH RECORD FOUND BAD.
3784                                    ;A COUNT IS MADE OF THE NUMBER OF SUCCESSIVE BAD
3785                                    ;CHARACTERS, AND IF TEN (10) SUCCESSIVE BAD CHARACTERS
3786                                    ;ARE FOUND IN A SINGLE RECORD, A MESSAGE INDICATING
3787                                    ;A BAD RECORD CONDITION IS PRINTED AND THE NEXT
3788                                    ;TWENTY (20) CHARACTERS ARE SKIPPED BEFORE CHECKING
3789                                    ;IS RESUMED.  IF THE BAD RECORD CONDITION IS FOUND
3790                                    ;THREE TIMES IN A RECORD, ALL REMAINING DATA IS
3791                                    ;SKIPPED EXCEPT THE FINAL TEN (10) CHARACTERS.
3792                                    ;THIS SKIPPING IS OF COURSE ONLY POSSIBLE IN
3793                                    ;RECORDS WHICH CONTAIN A SUFFICIENT NUMBER OF CHARACTERS.
3794                                    ;PRINTING OF ERRORS MAY BE DISALLOWED AT ANY TIME
3795                                    ;BY SETTING CONSOLE SWITCH TEN (10) TO A ONE.
3796                                    ;THE OPERATOR MAY CAUSE THE PROGRAM TO HALT ON ANY ERROR
3797                                    ;BY SETTING CONSOLE SWITCH FIFTEEN (15) TO A ONE.
3798                                    ;*********************************************************
3799
3800  016000  032777  002000  162602  DERR:   BIT     #2000,@SWR      ;BRANCH IF NO ERROR
3801  016006  001067                   BNE     DERR4           ;PRINTOUT DESIRED
3802  016010  005237  000672           DERR0:  INC     PFLG            ;SET PRINT FLAG
3803  016014  005737  000666           TST     HDRFL           ;SEE IF HAVE PRINTED HEADER
3804  016020  001007                   BNE     DERR0A          ;IF SO: BR
3805  016022  004737  022570           JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
3806  016026  012704  024444           MOV     #MSG1,R4        ;LOAD ERROR MSG ADDR
3807  016032  000004                   TYPE                    ;TYPE MSG
3808  016034  004737  021020           JSR     PC,FRPRT        ;PRINT F OR R
3809  016040  012704  024463           DERR0A: MOV     #MSG4,R4
3810  016044  000004                   TYPE                    ;TYPE MSG
3811  016046  010203                   MOV     R2,R3
3812  016050  162703  033334           SUB     #RDATA,R3       ;POINT TO CHAR
3813  016054  005303                   DEC     R3
3814  016056  032737  010000  000562   BIT     #10000,RDCMD    ;SEE IF READ REVERSE
3815  016064  001402                   BEQ     DERR0B          ;IF NOT: BR
3816  016066  010503                   MOV     R5,R3           ;GET CHAR NUMBER
3817  016070  005103                   COM     R3
3818  016072  104400           DERR0B: TYPOCT                  ;PRINT CHAR NUMBER
3819  016074  012704  024451           MOV     #MSG2,R4
3820  016100  000004                   TYPE                    ;TYPE MSG
3821  016102  032737  010000  000562   BIT     #10000,RDCMD    ;SEE IF READ REVERSE
3822  016110  001402                   BEQ     DERR0C          ;IF NOT: BR
3823  016112  111103                   MOVB    (R1),R3         ;GET CHAR
3824  016114  000401                   BR      DERR0D
3825  016116  114103           DERR0C: MOVB    -(R1),R3        ;LOAD EXPECTED DATA
3826  016120  004737  024222           DERR0D: JSR     PC,DOUT         ;GO PRINT CHAR
3827  016124  012704  024456           MOV     #MSG3,R4
```

```
3828  016130  000004                          TYPE                    ;TYPE MSG
3829  016132  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
3830  016140  001402                          BEQ     DERR1           ;IF NOT: BR
3831  016142  111203                          MOVB    (R2),R3         ;GET CHAR
3832  016144  000401                          BR      DERR2
3833  016146  114203                  DERR1:  MOVB    -(R2),R3
3834  016150  004737  024222          DERR2:  JSR     PC,DOUT         ;PRINT BAD CHAR
3835  016154  032737  010000  000562          BIT     #10000,RDCMD    ;BRANCH IF NOT READ
3836  016162  001001                          BNE     DERR4           ;REVERSE
3837  016164  122122                  DERR3:  CMPB    (R1)+,(R2)+     ;RESET POINTERS
3838  016166  105237  000660          DERR4:  INCB    BBC             ;BUMP BAD RECORD CNTR
3839  016172  122737  000010  000660          CMPB    #10,BBC         ;SEE IF ILD BTH
3840  016200  001123                          BNE     DEREX           ;IF NOT: BR
3841  016202  032777  002000  162400          BIT     #2000,@SWR      ;SEE IF PRINT INHIBIT
3842  016210  001003                          BNE     1$              ;IF SO: BR
3843  016212  012704  024577                  MOV     #MSG15,R4
3844  016216  000004                          TYPE                    ;TYPE MSG
3845  016220  105037  000660          1$.     CLRB    BBC             ;RESET BAD RECORD CNTR
3846  016224  000337  000660                  SWAB    BBC             ;POSITION BLD BTH AMOUNT
3847  016230  105237  000660                  INCB    BBC             ;BUMP AMOUNT
3848  016234  122737  000003  000660          CMPB    #3,BBC          ;SEE IF HAD 3 BLD BTHS
3849  016242  101054                          BHI     DERR4B          ;IF NOT: BR
3850  016244  000337  000660                  SWAB    BBC             ;REPOSITION BBC
3851  016250  022705  177767                  CMP     #177767,R5      ;SEE IF ON LAST EIGHT CHARS
3852  016254  101473                          BLOS    DERR6           ;IF SO: BR
3853  016256  012705  177767                  MOV     #177767,R5      ;SET CHAR CNTR TO 8
3854  016262  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
3855  016270  001416                          BEQ     DERR4A          ;IF NOT: BR
3856  016272  012701  027326                  MOV     #WDATA,R1       ;GET START OF BUFFER
3857  016276  012702  033334                  MOV     #RDATA,R2       ;GET START OF BUFFER
3858  016302  062701  000010                  ADD     #10,R1
3859  016306  062702  000010                  ADD     #10,R2          ;POINT TO START +10
3860  016312  032737  000001  000556          BIT     #1,FMCNT        ;SEE IF 000 FRAME COUNT
3861  016320  001453                          BEQ     DEREX           ;IF NOT: BR
3862  016322  105722                          TSTB    (R2)+           ;BUMP POINTER
3863  016324  000451                          BR      DEREX
3864  016326  013737  000556  000644  DERR4A: MOV     FMCNT,TEMP1     ;LOAD CHAR COUNT
3865  016334  005137  000644                  COM     TEMP1
3866  016340  005237  000644                  INC     TEMP1
3867  016344  162737  000010  000644          SUB     #10,TEMP1       ;POINT TO BUFFER -8
3868  016352  013701  000644                  MOV     TEMP1,R1        ;POINT TO NEXT CHAR
3869  016356  062701  027326                  ADD     #WDATA,R1       ;POINT TO NEXT WRITE CHAR
3870  016362  013702  000644                  MOV     TEMP1,R2        ;POINT TO END OF READ DATA -8 FORWARD
3871  016366  062702  033334                  ADD     #RDATA,R2       ;POINT TO NEXT CHAR
3872  016372  000426                          BR      DEREX           ;EXIT
3873  016374  000337  000660          DERR4B: SWAB    BBC             ;REPOSITION BBC
3874  016400  000241                          CLC
3875  016402  062705  000024                  ADD     #24,R5          ;SKIP 20 CHARS
3876  016406  103416                          BCS     DERR6           ;IF EXCEED RECORD SIZE: BR
3877  016410  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
3878  016416  001405                          BEQ     DERR5           ;IF NOT: BR
3879  016420  162701  000024                  SUB     #24,R1
3880  016424  162702  000024                  SUB     #24,R2          ;RESET POINTERS
3881  016430  000407                          BR      DEREX
3882  016432  062701  000024          DERR5:  ADD     #24,R1          ;SKIP 20 CHARS
3883  016436  062702  000024                  ADD     #24,R2          ;SKIP FORWARD 20 CHARS
```

```
3884  016442  000402                          BR      DEREX
3885  016444  012705  177777          DERR6:  MOV     #-1,R5          ;SET TO ERR
3886  016450  005777  162134          DEREX:  TST     @SWR            ;BRANCH IF NOT HALT ON ERROR
3887  016454  100012                          BPL     DEREX1
3888  016456  000000                          HALT
3889  016460  005737  000672                  TST     PFLG            ;SEE IF PRINTED
3890  016464  001006                          BNE     DEREX1          ;IF SO: BR
3891  016466  032777  002000  162114          BIT     #2000,@SWR      ;SEE IF SHOULD PRINT
3892  016474  001002                          BNE     DEREX1          ;IF NOT: BR
3893  016476  000137  016010                  JMP     DERR0           ;ELSE PRINT
3894  016502  005037  000672          DEREX1: CLR     PFLG            ;CLEAR FLAG
3895  016506  005237  000706                  INC     DERFL           ;BUMP DATA ERROR FLAG
3896  016512  000207                          RTS     PC              ;RETURN
3897
```

```
3898
3899                                    ;*********************************************************
3900                                    ;DROPS AND PICKS SUBROUTINE:
3901                                    ;
3902                                    ;THIS SUBROUTINE IS USED TO ACCUMULATE FROM
3903                                    ;EACH BAD DATA CHARACTER FOUND THE NUMBER
3904                                    ;OF BITS WHICH WERE EITHER DROPPED OR PICKED UP.
3905                                    ;TWO COUNTERS PER SLAVE ARE USED TO ACCUMULATE THIS
3906                                    ;INFORMATION AND CAN STORE UP TO 32K DROPS
3907                                    ;OR PICKS BEFORE OVERFLOWING.  IF OVERFLOW IS
3908                                    ;ABOUT TO OCCUR, THESE ACCUMULATORS ARE
3909                                    ;PRINTED IN OCTAL AND RESET TO ZERO.
3910                                    ;THE CONTENTS OF THE ACCUMULATORS MAY BE
3911                                    ;DISPLAYED AT ANY TIME BY SETTING CONSOLE
3912                                    ;SWITCH FOURTEEN TO A ONE (1). THE PRINTOUT WILL OCCUR
3913                                    ;AT THE END OF THE CURRENT BLOCK CYCLE.
3914                                    ;*********************************************************
3915
3916  016514  005037  000644           DRPKF:  CLR     TEMP1
3917  016520  005037  000646                   CLR     TEMP2
3918  016524  005037  000650                   CLR     TEMP3
3919  016530  111137  000644                   MOVB    (R1),TEMP1      ;LOAD GOOD CHAR
3920  016534  111237  000646                   MOVB    (R2),TEMP2      ;LOAD BAD CHAR
3921  016540  013704  000676                   MOV     UNP,R4
3922  016544  016437  000770  000722         MOV     PIK1(R4),BPKP
3923  016552  016437  001010  000720         MOV     DRP1(R4),BDPP
3924  016560  032737  010000  000562         BIT     #10000,RDCMD    ;SEE IF READ REVERSE
3925  016566  001005                         BNE     DRPK            ;IF SO: BR
3926  016570  124142                         CMPB    -(R1),-(R2)     ;POINT TO CHAR
3927  016572  112137  000644                 MOVB    (R1)+,TEMP1     ;LOAD GOOD CHAR
3928  016576  112237  000646                 MOVB    (R2)+,TEMP2     ;LOAD BAD CHAR
3929  016602  004737  016614           DRPK:   JSR     PC,DROP         ;GET DROPS
3930  016606  004737  017034                   JSR     PC,PICK         ;GET PICKS
3931  016612  000207                         RTS     PC              ;EXIT
3932  016614  113703  000644           DROP:   MOVB    TEMP1,R3        ;R3 = GOOD CHAR
3933  016620  113704  000646                   MOVB    TEMP2,R4        ;R4 = BAD CHAR
3934  016624  140403                   DPC:    BICB    R4,R3           ;GET DROPS/PICKS
3935  016626  001001                         BNE     DPCG            ;IF SOME: BR
3936  016630  000207                         RTS     PC              ;RETURN
3937  016632  012737  000010  000712   DPCG:   MOV     #10,BCNT        ;SET NUMBER TO CHECK
3938  016640  132703  000001           DPC0:   BITB    #1,R3           ;SEE IF DROPPED OR PICKED THIS BIT
3939  016644  001455                         BEQ     DPC2            ;IF NOT: BR
3940  016646  105737  000650                 TSTB    TEMP3           ;SEE IF ON PICKS
3941  016652  001016                         BNE     DPC1            ;IF SO: BR
3942  016654  005277  162040                 INC     @BDPP           ;BUMP DROP CNTR
3943  016660  005777  162034                 TST     @BDPP
3944  016664  100045                         BPL     DPC2            ;IF NO OVERFLOW: BR
3945  016666  032777  002000  161714         BIT     #2000,@SWR      ;SEE IF HAVE PRINTED DATA
3946  016674  001402                         BEQ     DPC0A           ;IF SO: BR
3947  016676  004737  022570                 JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
3948  016702  004737  017100           DPC0A:  JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
3949  016706  000415                         BR      DPC2A
3950  016710  005277  162006           DPC1:   INC     @BPKP           ;BUMP PICK CNTR
3951  016714  005777  162002                 TST     @BPKP           ;SEE IF OVERFLOW
3952  016720  100027                         BPL     DPC2            ;IF NOT: BR
3953  016722  032777  002000  161660         BIT     #2000,@SWR      ;SEE IF HAVE PRINTED DATA
```

```
3954  016730  001402                          BEQ     DPC1A           ;IF SO: BR
3955  016732  004737  022570                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
3956  016736  004737  017100          DPC1A:  JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
3957  016742  013704  000676          DPC2A:  MOV     UNP,R4
3958  016746  016403  001010                  MOV     DRP1(R4),R3     ;SET DROP POINTER
3959  016752  016404  000770                  MOV     PIK1(R4),R4     ;SET PICK POINTER
3960  016756  012737  000010  000712          MOV     #10,BCNT        ;SET NUMBER OF BITS
3961  016764  005023                  DPC2B:  CLR     (R3)+           ;CLEAR DROPS
3962  016766  005024                          CLR     (R4)+           ;CLEAR PICK
3963  016770  005337  000712                  DEC     BCNT            ;SEE IF DONE
3964  016774  001373                          BNE     DPC2B           ;IF NOT: BR
3965  016776  000207                          RTS     PC              ;EXIT
3966  017000  000241                  DPC2:   CLC
3967  017002  106003                          RORB    R3              ;GET NEXT BIT
3968  017004  005337  000712                  DEC     BCNT            ;SEE IF DONE
3969  017010  001410                          BEQ     DPC3
3970  017012  062737  000002  000722          ADD     #2,BPKP
3971  017020  062737  000002  000720          ADD     #2,BDPP
3972  017026  000137  016640                  JMP     DPC0            ;CONTINUE
3973  017032  000207                  DPC3:   RTS     PC              ;RETURN
3974  017034  013704  000676          PICK:   MOV     UNP,R4          ;GET UNIT POINTER
3975  017040  016437  000770  000722          MOV     PIK1(R4),BPKP   ;SET PICK POINTER
3976  017046  016437  001010  000720          MOV     DRP1(R4),BDPP   ;SET DROP POINTER
3977  017054  113704  000644                  MOVB    TEMP1,R4        ;R4 = GOOD CHAR
3978  017060  113703  000646                  MOVB    TEMP2,R3        ;R3 = BAD CHAR
3979  017064  112737  000001  000650          MOVB    #1,TEMP3        ;SET PICK FLAG
3980  017072  004737  016624                  JSR     PC,DPC          ;GO CHECK PICKS
3981  017076  000207                          RTS     PC              ;EXIT
3982  017100  012704  025075          DPPRT:  MOV     #MSG26,R4
3983  017104  000004                          TYPE                    ;TYPE MSG
3984  017106  013704  000676                  MOV     UNP,R4
3985  017112  016437  001010  000720          MOV     DRP1(R4),BDPP   ;SET DROP POINTER
3986  017120  016437  000770  000722          MOV     PIK1(R4),BPKP   ;SET PICK POINTER
3987  017126  062737  000016  000720          ADD     #16,BDPP
3988  017134  062737  000016  000722          ADD     #16,BPKP
3989  017142  012737  000010  000712          MOV     #10,BCNT        ;SET NUMBER TO PRINT
3990  017150  017703  161544          DPPRT0: MOV     @BDPP,R3
3991  017154  104400                          TYPOCT                  ;PRINT DROPS
3992  017156  005337  000712                  DEC     BCNT            ;SEE IF DONE
3993  017162  001404                          BEQ     DPPRT1          ;IF NOT: BR
3994  017164  162737  000002  000720          SUB     #2,BDPP         ;BUMP POINTER
3995  017172  000766                          BR      DPPRT0          ;CONTINUE FOR ALL 8 BITS
3996  017174  012737  000010  000712  DPPRT1: MOV     #10,BCNT        ;SET NUMBER TO PRINT
3997  017202  012704  025106                  MOV     #MSG27,R4
3998  017206  000004                          TYPE                    ;TYPE MSG
3999  017210  017703  161506          DPPRT2: MOV     @BPKP,R3
4000  017214  104400                          TYPOCT                  ;PRINT PICKS
4001  017216  005337  000712                  DEC     BCNT            ;SEE IF DONE
4002  017222  001404                          BEQ     DPPRTX          ;IF SO: BR
4003  017224  162737  000002  000722          SUB     #2,BPKP         ;BUMP POINTER
4004  017232  000766                          BR      DPPRT2          ;CONTINUE FOR ALL 8 BITS
4005  017234  000207                  DPPRTX: RTS     PC              ;RETURN
```

```
4006
4007                                    ;*********************************************************
4008                                    ;STATUS CHECK SUBROUTINE:
4009                                    ;
4010                                    ;THIS SUBROUTINE IS USED TO PERFORM A CHECK OF
4011                                    ;BOTH THE MASSBUS CONTROLLER (RH11) AND THE TAPE
4012                                    ;CONTROLLER (TM02).  THE RH11 IS CHECKED FOR ERRORS
4013                                    ;AS REFLECTED IN REGISTERS CS1 AND CS2 AND ALSO THAT
4014                                    ;THE BUS ADDRESS (BA) AND WORD COUNT (WC) ARE
4015                                    ;CORRECT.  THE TM02 IS CHECKED FOR DRIVE STATUS (DS),
4016                                    ;DRIVE ERRORS (ER), AND PROPER FRAME COUNT.  THE SPECIAL
4017                                    ;CHECK CHARACTERS (CRC+LRC) ARE ALSO CHECKED WHEN
4018                                    ;APPROPIATE (IE: NRZ READ OR WRITE).  CERTAIN TYPES
4019                                    ;OF DRIVE ERRORS IN PE OPERATION WILL BE ACCOMPANIED
4020                                    ;BY THE DISPLAY OF THE DEAD TRACK REGISTER (CC).  THESE
4021                                    ;TYPES ARE ER BITS 15,10,7,6.  THE PRINTOUTS OF BAD
4022                                    ;CRC,LRC,FC, AND BA WILL SHOW BOTH THE EXPECTED AND
4023                                    ;RECEIVED VALUES (IE: EXPT-RCVD).  ONLY THOSE REGISTERS
4024                                    ;WHICH ARE IN ERROR WILL BE PRINTED AND ALL PRINTOUTS
4025                                    ;ARE IN OCTAL FORMAT WITH NO LEADING ZEROS.  AS IN
4026                                    ;DATA ERRORS, STATUS ERRORS ARE PRECEEDED BY HEADER
4027                                    ;DESCRIBING THE HARDWARE UNDER TEST, THE BLOCKING
4028                                    ;INFORMATION, AND THE ERROR TYPE.
4029                                    ;*********************************************************
4030
4031  017236  013703  000556    ERCHK:  MOV     FMCNT,R3        ;GET FRAME COUNT
4032  017242  032703  000001            BIT     #1,R3           ;SEE IF ODD
4033  017246  001401                    BEQ     ER0             ;IF NOT: BR
4034  017250  005303                    DEC     R3              ;BUMP COUNT
4035  017252  005403            ER0:    NEG     R3
4036  017254  032737  000020  000552    BIT     #20,UDES        ;SEE IF CORE DUMP
4037  017262  001401                    BEQ     ER0B            ;IF NOT: BR
4038  017264  006203                    ASR     R3              ;SET TO FC/2
4039  017266  032737  000010  000674    BIT     #10,MTC1        ;SEE IF WRITE OP
4040  017274  001414                    BEQ     ER1             ;IF SO: BR
4041  017276  032737  010000  000562    BIT     #10000,RDCMD
4042  017304  001405                    BEQ     ER0A
4043  017306  012703  033334            MOV     #RDATA,R3
4044  017312  162703  000002            SUB     #2,R3           ;SET POINTER
4045  017316  000405                    BR      ER2
4046  017320  062703  033334    ER0A:   ADD     #RDATA,R3       ;BUILD EXPT READ ADDRESS
4047  017324  000402                    BR      ER2
4048  017326  062703  027326    ER1:    ADD     #WDATA,R3       ;BUILD EXPT WRITE ADDRESS
4049  017332  010337  020774    ER2:    MOV     R3,CADER        ;SAVE ADDRESS
4050  017336  012704  000007            MOV     #7,R4
4051  017342  012701  020776            MOV     #BAER,R1
4052  017346  005021            ER2A0:  CLR     (R1)+           ;CLEAR FLAGS
4053  017350  005304                    DEC     R4
4054  017352  001375                    BNE     ER2A0
4055  017354  020377  161134            CMP     R3,@BA          ;SEE IF ADDRESS OK
4056  017360  001402                    BEQ     ER2A1           ;IF SO: BR
4057  017362  005237  020776            INC     BAER            ;SET BUS ADDRESS ERROR
4058  017366  032737  000010  000674    BIT     #10,MTC1        ;SEE IF WRITE OPER
4059  017374  001006                    BNE     ER2B            ;IF NOT: BR
4060  017376  005777  161114    ER2A:   TST     @FC             ;SEE IF FC=0
4061  017402  001441                    BEQ     ER3             ;IF SO: BR
```

```
4062  017404  005237  021004                    INC     FCER            ;SET FC ERROR
4063  017410  000436                            BR      ER3
4064  017412  032737  000040  000674  ER2B:     BIT     #40,MTC1        ;SEE IF SPACE OPER
4065  017420  001766                            BEQ     ER2A            ;IF SO: BR
4066  017422  005737  000700                    TST     TMFLG           ;SEE IF TM TIME
4067  017426  001011                            BNE     ER2D            ;IF SO: BR
4068  017430  013703  000556                    MOV     FMCNT,R3
4069  017434  005403                            NEG     R3              ;R3 = EXPT RECORD SIZE
4070  017436  020377  161054          ER2C:     CMP     R3,@FC          ;SEE IF FC = EXPT
4071  017442  001421                            BEQ     ER3             ;IF SO: BR
4072  017444  005237  021004                    INC     FCER            ;SET FC ERROR FLAG
4073  017450  000416                            BR      ER3
4074  017452  032737  002000  000552  ER2D:     BIT     #2000,UDES      ;SEE IF PE
4075  017460  001346                            BNE     ER2A            ;IF SO: BR
4076  017462  032737  010000  000562            BIT     #10000,RDCMD    ;SEE IF READ REVERSE
4077  017470  001003                            BNE     ER2E            ;IF SO: BR
4078  017472  012703  000002                    MOV     #2,R3
4079  017476  000757                            BR      ER2C            ;LOOK FOR EXPT = 2
4080  017500  012703  000001          ER2E:     MOV     #1,R3
4081  017504  000754                            BR      ER2C            ;GO CHECK FC FOR TM
4082  017506  032777  160000  160774  ER3:      BIT     #160000,@C1     ;SEE IF COUNT ERROR
4083  017514  001437                            BEQ     ER4
4084  017516  017703  160776                    MOV     @CS,R3          ;GET CONT STATUS REG
4085  017522  042703  000307                    BIC     #307,R3         ;MASK OUT IR,OR,UNIT NO. & SEE IF OTHER ERRORS
4086  017526  001406                            BEQ     ER3A            ;IF NOT: BR
4087  017530  005737  000700                    TST     TMFLG           ;SEE IF TAPE MARK TIME
4088  017534  001425                            BEQ     ER3B            ;IF NOT: BR
4089  017536  042703  001000                    BIC     #1000,R3        ;MASK MISSED TRANS & BR IF OTHER ERRORS
4090  017542  001022                            BNE     ER3B
4091  017544  032777  060000  160736  ER3A:     BIT     #60000,@C1      ;SEE IF EITHER TRE OR MCPE
4092  017552  001420                            BEQ     ER4             ;IF NOT: BR
4093  017554  005737  000700                    TST     TMFLG           ;SEE IF TM TIME
4094  017560  001413                            BEQ     ER3B            ;IF NOT: BR
4095  017562  017703  160736                    MOV     @ER,R3          ;GET ERROR REGISTER
4096  017566  032737  000010  000552            BIT     #10,UDES        ;SEE IF EVEN PARITY
4097  017574  001402                            BEQ     ER3A1           ;IF NOT: BR
4098  017576  042703  000100                    BIC     #100,R3         ;MASK PAR
4099  017602  042703  001000          ER3A1:    BIC     #1000,R3        ;MASK FCE
4100  017606  001402                            BEQ     ER4             ;IF NO ERRORS EXCEPT FCE: BR
4101  017610  005237  021000          ER3B:     INC     CONER           ;SET CONT ERROR FLAG
4102  017614  032777  040000  160700  ER4:      BIT     #40000,@DS      ;SEE IF DRIVE ERROR
4103  017622  001420                            BEQ     ER6             ;IF NOT: BR
4104  017624  005737  000700                    TST     TMFLG           ;SEE IF TAPE MARK TIME
4105  017630  001413                            BEQ     ER4A            ;IF NOT: BR
4106  017632  017703  160666                    MOV     @ER,R3          ;GET ER
4107  017636  032737  000010  000552            BIT     #10,UDES        ;SEE IF EVEN PARITY
4108  017644  001402                            BEQ     ER4A1           ;IF NOT: BR
4109  017646  042703  000100                    BIC     #100,R3         ;MASK PAR
4110  017652  042703  001000          ER4A1:    BIC     #1000,R3        ;MASK OUT FCE & BRANCH IF
4111  017656  001402                            BEQ     ER6             ;NO OTHER ERRORS
4112  017660  005237  021002          ER4A:     INC     DRVER           ;SET DRIVER ERROR FLAG
4113  017664  032737  002000  000552  ER6:      BIT     #2000,UDES
4114  017672  001071                            BNE     ERPT            ;IF IN PE MODE: BR
4115  017674  032777  020000  160706            BIT     #20000,@SWR     ;SEE IF NO DATA CHECK
4116  017702  001065                            BNE     ERPT            ;IF NOT: BR (ALLOW READ OF UNKNOWN TAPES)
4117  017704  032737  000040  000674            BIT     #40,MTC1        ;SEE IF WRITE OR READ OP
```

```
4118  017712  001461                        BEQ     ERPT            ;IF NOT: BR
4119  017714  005737  000700                TST     TMFLG           ;SEE IF TAPE MARK TIME
4120  017720  001413                        BEQ     ER6A            ;IF NOT: BR
4121  017722  013737  015362  021016        MOV     EXCRC,CRCSV     ;SAVE CRC
4122  017730  013737  015364  021014        MOV     EXLRC,LRCSV     ;SAVE LRC
4123  017736  005037  015362                CLR     EXCRC
4124  017742  012737  000023  015364        MOV     #23,EXLRC       ;SET CRC/LRC FOR TM
4125  017750  032737  000060  000552  ER6A:  BIT     #60,UDES        ;SEE IF FORMAT 14
4126  017756  001037                        BNE     ERPT            ;IF NOT: BR
4127  017760  017703  160544                MOV     @CC,R3          ;GET CRC CHARACTER
4128  017764  042703  177000                BIC     #177000,R3
4129  017770  023703  015362                CMP     EXCRC,R3
4130  017774  001402                        BEQ     ER7             ;IF CRC GOOD: BR
4131  017776  005237  021010                INC     CRCER           ;SET ERROR FLAG
4132  020002  017703  160526        ER7:    MOV     @MR,R3          ;GET LRC
4133  020006  000303                        SWAB    R3
4134  020010  005703                        TST     R3
4135  020012  100002                        BPL     ER10
4136  020014  052703  000400                BIS     #400,R3
4137  020020  042703  177000        ER10:   BIC     #177000,R3
4138  020024  023703  015364                CMP     EXLRC,R3
4139  020030  001412                        BEQ     ERPT            ;IF LRC GOOD: BR
4140  020032  010337  021012                MOV     R3,ACTLRC       ;SAVE ACTUAL LRC
4141  020036  005237  021006                INC     LRCER           ;SET LRC ERROR FLAG
4142  020042  032737  010000  000562        BIT     #10000,RDCMD    ;SEE IF READ REVERSE
4143  020050  001402                        BEQ     ERPT            ;IF NOT: BR
4144  020052  005037  021006                CLR     LRCER           ;ELSE CLEAR LRC ERROR
4145  020056  012703  000006        ERPT:   MOV     #6,R3
4146  020062  005037  000710                CLR     SERFL           ;CLEAR ERROR FLAG
4147  020066  005037  000724                CLR     ERSAV
4148  020072  012704  020776                MOV     #BAER,R4
4149  020076  005724                ERPTT:  TST     (R4)+           ;SEE IF ANY ERROR
4150  020100  001004                        BNE     ERPTG           ;IF SO: BR
4151  020102  005303                        DEC     R3
4152  020104  001374                        BNE     ERPTT
4153  020106  000137  020740                JMP     ERPX1
4154  020112  005237  000710        ERPTG:  INC     SERFL           ;SET ERROR FLAG
4155  020116  017737  160402  000724        MOV     @ER,ERSAV       ;SAVE ERROR REGISTER
4156  020124  032777  002000  160456        BIT     #2000,@SHR      ;SEE IF PRINT
4157  020132  001420                        BEQ     ERPTO           ;IF SO: BR
4158  020134  022737  000002  000714        CMP     #2,RTYFL        ;SEE IF READ RETRY
4159  020142  001006                        BNE     ERPTG1          ;IF NOT: BR
4160  020144  013703  000704                MOV     RTCNT,R3
4161  020150  005203                        INC     R3              ;BUMP RETRY COUNT
4162  020152  020337  000604                CMP     R3,RETRY        ;SEE IF LAST RETRY
4163  020156  001406                        BEQ     ERPTO           ;IF SO: BR
4164  020160  022737  000002  021002  ERPTG1: CMP   #2,DRVER        ;SEE IF TM STATUS ERROR
4165  020166  001402                        BEQ     ERPTO           ;IF SO: BR
4166  020170  000137  020620                JMP     ERPXO
4167  020174  005237  000672        ERPTO:  INC     PFLG
4168  020200  004737  022570                JSR     PC,PAPRT        ;PRINT HEADER
4169  020204  013704  000654                MOV     EMADDR,R4
4170  020210  000004                        TYPE                    ;TYPE MSG
4171  020212  004737  021020                JSR     PC,FRPRT        ;PRINT F OR R
4172  020216  005737  000700                TST     TMFLG
4173  020222  001407                        BEQ     ERPT1
```

```
4174  020224  022737  026026  000654          CMP     #MSG54,EMADDR
4175  020232  001403                          BEQ     ERPT1
4176  020234  012704  026044                  MOV     #MSG56,R4       ;PRINT TM
4177  020240  000004                          TYPE                    ;TYPE MSG
4178  020242  005737  021000          ERPT1:  TST     CONER
4179  020246  001414                          BEQ     ERPT2           ;IF NO CONT ERROR: BR
4180  020250  012704  024725                  MOV     #MSG23,R4
4181  020254  000004                          TYPE                    ;TYPE MSG
4182  020256  017703  160226                  MOV     @C1,R3
4183  020262  104400                          TYPOCT                  ;PRINT CONTROL 1
4184  020264  012704  024752                  MOV     #MSG23D,R4      ;PRINT CS TAG
4185  020270  000004                          TYPE                    ;TYPE MSG
4186  020272  017703  160222                  MOV     @CS,R3
4187  020276  104400                          TYPOCT                  ;PRINT CONT STATUS
4188  020300  005737  021002          ERPT2:  TST     DRVER
4189  020304  001414                          BEQ     ERPT3           ;IF NO DRIVE ERROR: BR
4190  020306  012704  024760                  MOV     #MSG23E,R4
4191  020312  000004                          TYPE                    ;TYPE MSG
4192  020314  017703  160202                  MOV     @DS,R3
4193  020320  104400                          TYPOCT                  ;PRINT DRIVE STATUS
4194  020322  012704  024765                  MOV     #MSG23F,R4
4195  020326  000004                          TYPE                    ;TYPE MSG
4196  020330  017703  160170                  MOV     @ER,R3
4197  020334  104400                          TYPOCT                  ;PRINT DRIVE ERROR
4198  020336  005737  020776          ERPT3:  TST     BAER
4199  020342  001416                          BEQ     ERPT4           ;IF NO BA ERROR: BR
4200  020344  012704  024740                  MOV     #MSG238,R4
4201  020350  000004                          TYPE                    ;TYPE MSG
4202  020352  017703  160136                  MOV     @BA,R3
4203  020356  104400                          TYPOCT                  ;PRINT BUS ADDRESS
4204  020360  012737  000255  000640          MOV     #255,TOB
4205  020366  004737  023744                  JSR     PC,TOG          ;PRINT /
4206  020372  013703  020774                  MOV     CAOER,R3
4207  020376  104400                          TYPOCT                  ;PRINT EXPT BUS ADDRESS
4208  020400  005737  021004          ERPT4:  TST     FCER
4209  020404  001406                          BEQ     ERPT5           ;IF NO FC ERROR: BR
4210  020406  012704  024745                  MOV     #MSG23C,R4
4211  020412  000004                          TYPE                    ;TYPE MSG
4212  020414  017703  160076                  MOV     @FC,R3
4213  020420  104400                          TYPOCT                  ;PRINT FRAME COUNT
4214  020422  012704  024733          ERPT5:  MOV     #MSG239,R4
4215  020426  000004                          TYPE                    ;TYPE MSG
4216  020430  017703  160056                  MOV     @WC,R3
4217  020434  104400                          TYPOCT                  ;PRINT WORD COUNT
4218  020436  005737  021010                  TST     CRCER
4219  020442  001420                          BEQ     ERPT5A          ;IF NO CRC ERROR: BR
4220  020444  012704  026071                  MOV     #MSG58,R4
4221  020450  000004                          TYPE                    ;TYPE MSG
4222  020452  017703  160052                  MOV     @CC,R3
4223  020456  042703  177000                  BIC     #177000,R3
4224  020462  104400                          TYPOCT                  ;PRINT ACTUAL CRC
4225  020464  012737  000255  000640          MOV     #255,TOB
4226  020472  004737  023744                  JSR     PC,TOG
4227  020476  013703  015362                  MOV     EXCRC,R3
4228  020502  104400                          TYPOCT                  ;PRINT EXPECTED CRC
4229  020504  005737  021006          ERPT5A: TST     LRCER
```

```
4230  020510  001416                          BEQ     ERPT6           ;IF NO LRC ERROR: BR
4231  020512  012704  026077                  MOV     #MSG59,R4
4232  020516  000004                          TYPE                    ;TYPE MSG
4233  020520  013703  021012                  MOV     ACTLRC,R3
4234  020524  104400                          TYPOCT                  ;PRINT ACTUAL LRC
4235  020526  012737  000255  000640          MOV     #255,TOB
4236  020534  004737  023744                  JSR     PC,TOG
4237  020540  013703  015364                  MOV     EXLRC,R3
4238  020544  104400                          TYPOCT                  ;PRINT EXPECTED LRC
4239  020546  005737  021002          ERPT6:  TST     DRVER
4240  020552  001421                          BEQ     ERPT7           ;IF NO DRIVE ERROR: BR
4241  020554  032737  002000  000552          BIT     #2000,UDES
4242  020562  001415                          BEQ     ERPT7           ;IF NO FE: BR
4243  020564  017704  157734                  MOV     @ER,R4
4244  020570  042704  075477                  BIC     #75477,R4       ;MASK OUT ALL BUT BITS 15,10,7,6
4245  020574  001410                          BEQ     ERPT7           ;IF NO CONDITIONALS SET: BR
4246  020576  012704  024777                  MOV     #MSG23H,R4
4247  020602  000004                          TYPE                    ;TYPE MSG
4248  020604  017703  157720                  MOV     @CC,R3
4249  020610  042703  177000                  BIC     #177000,R3      ;MASK CC
4250  020614  104400                          TYPOCT                  ;PRINT CHECK CHARACTERS
4251  020616  000240          ERPT7:  NOP
4252  020620  005777  157764          ERPX0:  TST     @SWR            ;BRANCH IF NOT HALT ON ERROR
4253  020624  100012                          BPL     ERPX
4254  020626  000000                          HALT
4255  020630  005737  000672                  TST     PFLG            ;SEE IF HAVE PRINTED
4256  020634  001006                          BNE     ERPX            ;IF SO: BR
4257  020636  032777  002000  157744          BIT     #2000,@SWR      ;SEE IF SHOULD PRINT
4258  020644  001002                          BNE     ERPX            ;IF NOT: BR
4259  020646  000137  020174                  JMP     ERPT0           ;PRINT ERROR
4260  020652  005037  000672          ERPX:   CLR     PFLG
4261  020656  005737  000566                  TST     CRCC            ;BRANCH IF CRC ERROR
4262  020662  001007                          BNE     1$              ;CORRECTION DESIRED
4263  020664  012777  000040  157626          MOV     #40,@CS         ;ELSE INIT
4264  020672  013777  000550  157620          MOV     DVN,@CS         ;RESET DRIVE NUMBER
4265  020700  000414                          BR      2$
4266  020702  012777  000011  157600  1$:     MOV     #11,@C1         ;DRIVE CLEAR
4267  020710  017704  157612                  MOV     @AS,R4
4268  020714  010477  157606                  MOV     R4,@AS          ;CLEAR AS
4269  020720  013704  000510                  MOV     C1,R4
4270  020724  005204                          INC     R4
4271  020726  152714  000100                  BISB    #100,(R4)       ;RESET TRE
4272  020732  013777  000552  157602  2$:     MOV     UDES,@TC        ;RESET TC
4273  020740  032737  000040  000674  ERPX1:  BIT     #40,MTC1
4274  020746  001411                          BEQ     ERPX2           ;IF NOT READ/WRITE OP: BR
4275  020750  005737  000700                  TST     TMFLG
4276  020754  001406                          BEQ     ERPX2           ;IF NOT TM TIME: BR
4277  020756  013737  021016  015362          MOV     CRCSV,EXCRC     ;RESTORE CRC
4278  020764  013737  021014  015364          MOV     LRCSV,EXLRC     ;RESTORE LRC
4279  020772  000207          ERPX2:  RTS     PC                      ;EXIT
4280  020774  000000          CADER:  0                               ;EXPT ADDRESS SAVE
4281  020776  000000          BAER:   0
4282  021000  000000          CONER:  0
4283  021002  000000          DRVER:  0
4284  021004  000000          FCER:   0
4285  021006  000000          LRCER:  0
```

```
4286  021010  000000                    CRCER:  0
4287  021012  000000                    ACTLRC: 0
4288  021014  000000                    LRCSV:  0
4289  021016  000000                    CRCSV:  0
4290
4291                                    ;*********************************************************
4292                                    ;F FOR FORWARD/R FOR REVERSE PRINT SUBROUTINE:
4293                                    ;
4294                                    ;THIS SUBROUTINE IS USED TO PRINT OUT THE
4295                                    ;TAPE DIRECTION USED WHEN ANY ERROR IS
4296                                    ;DETECTED IN STATUS OF READ OR WRITE, DATA, OR
4297                                    ;SPACING OPERATIONS.
4298                                    ;*********************************************************
4299
4300  021020  032737  000010  000674    FRPRT:  BIT     #10,MTC1        ;SEE IF WRITE COMMAND
4301  021026  001411                            BEQ     2$              ;IF SO: BR
4302  021030  012704  024633                    MOV     #MSG17,R4       ;SET TO TYPE REVERSE MSG
4303  021034  032737  000002  000674            BIT     #2,MTC1         ;BRANCH IF REVERSE
4304  021042  001002                            BNE     1$
4305  021044  012704  024630                    MOV     #MSG16,R4       ;SET FORWARD MESSAGE
4306  021050  000004                    1$:     TYPE                    ;TYPE MSG
4307  021052  000207                    2$:     RTS     PC              ;EXIT
4308
```

```
4309
4310                            ;*************************************************************
4311                            ;TAPE COMMAND EXECUTE SUBROUTINE:
4312                            ;
4313                            ;THIS SUBROUTINE IS USED TO EXECUTE THE
4314                            ;MAG TAPE COMMAND DESCRIBED BY THE READ
4315                            ;OR WRITE ROUTINE.  THE FINAL COMMAND IS
4316                            ;SENT TO THE DEVICE REGISTER ALONG WITH THE
4317                            ;INTERRUPT ENABLE AND GO BITS.
4318                            ;ONCE THE COMMAND IS ISSUED, AN INTERRUPT
4319                            ;TIMER IS STARTED AND IF NO INTERRUPT IS RETURNED
4320                            ;BEFORE TIME OUT OCCURS, AN ERROR WILL BE
4321                            ;PRINTED AND THE PROGRAM STOPPED.  TESTING MAY
4322                            ;BE RESUMED BY PRESSING THE CONTINUE SWITCH.
4323                            ;TWO INTERRUPT HANDLERS ARE USED, ONE FOR MAG TAPE
4324                            ;AND ANOTHER FOR TELETYPE (TTY).
4325                            ;UPON RECEIPT OF A MAG TAPE INTERRUPT, HOUSEKEEPING
4326                            ;IS PERFORMED AND CONTROL RETURNED TO THE CALLING
4327                            ;ROUTINE (READ,WRITE,ETC).
4328                            ;RECEIPT OF A TTY INTERRUPT WILL CAUSE THE
4329                            ;PROGRAM TO CHECK FOR ENTRY OF A CNTRL C CHARACTER.
4330                            ;IF NOT CNTRL C, THEN CONTINUATION OF WAIT FOR MAG
4331                            ;TAPE INTERRUPT IS RETURNED.  IF, HOWEVER, THE TTY
4332                            ;INTERRUPT WAS CAUSED BY ENTRY OF A CNTRL C,
4333                            ;THEN AT THIS TIME REQUESTS FOR NEW STALL VALUES
4334                            ;ARE PRINTED AND THE RESPONSES ENTERED.  RESUMPTION
4335                            ;OF TAPE INTERRUPT WAIT IS THEN RESUMED.
4336                            ;*************************************************************
4337
4338  021054  005037  000644        TAPG:   CLR     TEMP1
4339  021060  013777  000550  157432        MOV     DVN,@CS         ;SET DRIVE NO.
4340  021066  032777  010000  157426 TAPG0: BIT     #10000,@DS      ;SEE IF HAVE MOL
4341  021074  001026                BNE     TAPG3           ;IF SO: BR
4342  021076  005237  000644        INC     TEMP1           ;SEE IF TIMED OUT
4343  021102  001371                BNE     TAPG0           ;WAIT FOR READY
4344  021104  004737  022570        JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
4345  021110  032737  000010  000674 BIT     #10,MTC1        ;SEE IF WRITE OP
4346  021116  001004                BNE     TAPG1           ;IF NOT: BR
4347  021120  012704  024470        MOV     #MSG5,R4
4348  021124  000004                TYPE                    ;TYPE MSG
4349  021126  000405                BR      TAPG2
4350  021130  012704  024475 TAPG1: MOV     #MSG6,R4
4351  021134  000004                TYPE                    ;TYPE MSG
4352  021136  004737  021020        JSR     PC,FRPRT        ;PRINT F OR R
4353  021142  012704  025055 TAPG2: MOV     #MSG25,R4
4354  021146  000004                TYPE                    ;TYPE MSG
4355  021150  000000                HALT
4356  021152  032777  020000  157342 TAPG3: BIT     #20000,@DS      ;SEE IF PIP RESET
4357  021160  001411                BEQ     TAPG3F          ;IF SO: BR
4358  021162  004737  022570        JSR     PC,PAPRT        ;PRINT HEADER
4359  021166  012704  027154        MOV     #MSG116,R4
4360  021172  000004                TYPE                    ;TYPE MSG
4361  021174  032777  020000  157320 1$:     BIT     #20000,@DS
4362  021202  001374                BNE     1$              ;AWAIT PIP RESET
4363  021204  022737  000026  000674 TAPG3F: CMP     #26,MTC1        ;SEE IF WRITE TM
4364  021212  001003                BNE     TAPG3A          ;IF NOT: BR
```

```
 4365  021214  012704  177777                    MOV     #-1,R4          ;ELSE SET FC FOR -1
 4366  021220  000406                            BR      TAPG3B
 4367  021222  013704  000556          TAPG3A:   MOV     FMCNT,R4
 4368  021226  032704  000001                    BIT     #1,R4
 4369  021232  001401                            BEQ     TAPG3B
 4370  021234  005304                            DEC     R4
 4371  021236  000261          TAPG3B:   SEC
 4372  021240  006004                            ROR     R4              ;SET WC = FC/2 FOR NORMAL FORMAT
 4373  021242  032737  000020  000552            BIT     #20,UDES        ;SEE IF CORE DUMP FORMAT
 4374  021250  001402                            BEQ     TAPG3C          ;IF NOT: BR
 4375  021252  000261                            SEC
 4376  021254  006004                            ROR     R4              ;SET WC = FC/4 FOR CORE DUMP
 4377  021256  010477  157230          TAPG3C:   MOV     R4,@WC          ;SET WORD COUNT
 4378  021262  012777  000011  157220            MOV     #11,@C1         ;DRIVE CLEAR
 4379  021270  017777  157222  157220            MOV     @FC,@FC         ;RESET FC LOADED
 4380  021276  005737  000570                    TST     INTRF           ;SEE IF INTERCHANGE READ
 4381  021302  001407                            BEQ     TAPG3D          ;IF NOT: BR
 4382  021304  032737  000040  000674            BIT     #40,MTC1        ;SEE IF READ OP
 4383  021312  001403                            BEQ     TAPG3D          ;IF NOT: BR
 4384  021314  012777  000003  157212            MOV     #3,@MR          ;SET INTERCHANGE READ MAINT. MODE
 4385  021322  013704  000674          TAPG3D:   MOV     MTC1,R4         ;GET COMMAND
 4386  021326  042704  177707                    BIC     #177707,R4      ;MASK OP CODE
 4387  021332  022704  000030                    CMP     #30,R4          ;SEE IF SPACE OP CODE
 4388  021336  001403                            BEQ     TAPG3E          ;IF SO: BR
 4389  021340  012737  177740  000670            MOV     #-40,STAL       ;SET INTERRUPT DELAY MULT TO 40
 4390  021346  052737  000101  000674  TAPG3E:   BIS     #101,MTC1       ;SET INTERRUPT ENABLE AND GO
 4391  021354  000240                            NOP
 4392  021356  013777  000674  157124            MOV     MTC1,@C1        ;EXECUTE COMMAND
 4393  021364  005077  157216                    CLR     @PSW            ;CLEAR PRIORITY
 4394  021370  005037  000644                    CLR     TEMP1
 4395  021374  005237  000644          TAPG4:    INC     TEMP1           ;SEE IF HAVE TIMED OUT
 4396  021400  001375                            BNE     TAPG4           ;IF NOT: BR
 4397  021402  005237  000670                    INC     STAL
 4398  021406  001372                            BNE     TAPG4           ;DO TIME DELAY MULTIPLIER
 4399  021410  012777  000340  157170  TAPG5:    MOV     #340,@PSW       ;RESET PRIORITY
 4400  021416  032777  002000  157164            BIT     #2000,@SWR      ;SEE IF SHOULD PRINT ERRORS
 4401  021424  001012                            BNE     TAPG6           ;IF NOT: BR
 4402  021426  004737  022570                    JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
 4403  021432  013704  000654                    MOV     EMADDR,R4
 4404  021436  000004                            TYPE                    ;TYPE MSG
 4405  021440  004737  021020                    JSR     PC,FRPRT        ;PRINT F OR R
 4406  021444  012704  025035                    MOV     #MSG24,R4
 4407  021450  000004                            TYPE                    ;TYPE MSG
 4408  021452  005777  157132          TAPG6:    TST     @SWR            ;BRANCH IF NOT HALT ON ERROR
 4409  021456  100001                            BPL     TAPG7
 4410  021460  000000                            HALT
 4411  021462  000137  021720          TAPG7:    JMP     MTINTA          ;RETURN TO CALLING ROUTINE
 4412
```

```
4413
4414                                                    ;TTY INTERRUPT HANDLER
4415  021466  017746  157122          TTINT:  MOV     @TKB,-(SP)      ;GET CHARACTER
4416  021472  042716  000200                  BIC     #200,(SP)       ;STRIP PARITY BIT
4417  021476  122716  000003                  CMPB    #3,(SP)         ;BRANCH IF NOT ^C
4418  021502  001005                          BNE     1$
4419  021504  000005                          RESET                   ;RESET ALL I/O
4420  021506  005077  157074                  CLR     @PSW            ;CLEAR PSW
4421  021512  000137  000200                  JMP     @#200           ;RESTART PROGRAM
4422  021516  122716  000001          1$:     CMPB    #1,(SP)         ;BRANCH IF NOT ^A
4423  021522  001016                          BNE     2$
4424  021524  022737  000176  000610        CMP     #SWREG,SWR      ;BRANCH IF HARDWARE SWR IS INVOKED
4425  021532  001015                          BNE     3$
4426  021534  012737  177570  000610        MOV     #177570,SWR     ;INVOKE HARDWARE SWR
4427  021542  004737  023274                  JSR     PC,.SAVE        ;SAVE REGISTERS ON THE STACK
4428  021546  012704  027300                  MOV     #MSG121,R4      ;TYPE 'HARDWARE SWR IN USE'
4429  021552  000004                          TYPE
4430  021554  004737  023316                  JSR     PC,.RESTORE     ;RESTORE REGISTERS
4431  021560  022716  000007          2$:     CMP     #7,(SP)         ;BRANCH IF NOT ^G
4432  021564  001005                          BNE     4$
4433  021566  012737  000176  000610  3$:     MOV     #SWREG,SWR      ;INVOKE SOFTWARE SWR
4434  021574  004737  023204                  JSR     PC,GTSWR        ;GET SWITCHES
4435  021600  022716  000002          4$:     CMP     #2,(SP)         ;BRANCH IF NOT ^B
4436  021604  001042                          BNE     6$
4437  021606  004737  023274                  JSR     PC,.SAVE        ;SAVE REGISTERS ON THE STACK
4438  021612  005237  014062                  INC     SCVFL           ;SET FLAG
4439  021616  004737  013634                  JSR     PC,TINP3A       ;GO CHECK CRC CORRECTION
4440  021622  032777  000040  156760        BIT     #40,@SWR        ;BRANCH IF NOT YOZZLING
4441  021630  001426                          BEQ     5$
4442  021632  012704  025623                  MOV     #MSG44,R4       ;REQUEST NEW YOZZLE STALL
4443  021636  000004                          TYPE                    ;TYPE MSG
4444  021640  013703  000602                  MOV     YSTAL,R3
4445  021644  104400                          TYPOCT                  ;PRINT PRESENT STALL
4446  021646  012705  000602                  MOV     #YSTAL,R5       ;SET ADDRESS OF YSTL
4447  021652  012701  000007                  MOV     #7,R1           ;SET NUMBER OF CHAR TO INPUT
4448  021656  012702  177777                  MOV     #-1,R2          ;SET MAXIMUM LIMIT
4449  021662  012703  002000                  MOV     #2000,R3        ;SET MINIMUM LIMIT
4450  021666  004737  023340                  JSR     PC,TTR          ;GO GET VALUE
4451  021672  004737  023316                  JSR     PC,.RESTORE     ;RESTORE REGISTERS
4452  021676  005726                          TST     (SP)+           ;POP CHARACTER OF THE STACK
4453  021700  012716  011252                  MOV     #YOZ,(SP)       ;RETURN TO 'YOZ'
4454  021704  000002                          RTI                     ;RETURN TO YOZ
4455  021706  004737  023316          5$:     JSR     PC,.RESTORE
4456  021712  005726                  6$:     TST     (SP)+           ;POP CHARACTER OFF THE STACK
4457  021714  000002                          RTI                     ;RETURN
4458
4459                                                    ;MAG TAPE INTERRUPT HANDLER
4460  021716  000240          MTINT:  NOP
4461  021720  042777  000037  156606  MTINTA: BIC     #37,@MR         ;CLEAR MAINT MODE
4462  021726  013716  000664                  MOV     RTRN,(SP)       ;SET RETURN TO (RTRN)
4463  021732  000002                          RTI                     ;RETURN
```

```
4464                                    ;*************************************
4465                                    ;AUTO SEQUENCE
4466                                    ;
4467                                    ;THIS ROUTINE ,ENTERED VIA STARTING ADDRESS 240
4468                                    ;WILL EXERCISE ALL AVAILABLE SLAVES ON ALL AVAILABLE
4469                                    ;DRIVES IN BOTH PE AND NRZ ACCORDING TO THE PRESELECTED
4470                                    ;TEST PLAN. IF NRZ ONLY, PE TESTING WILL NOT BE ATTEMPTED.
4471                                    ;*************************************
4472
4473  021734  012704  026576   ASEQ:   MOV     #MSG104,R4
4474  021740  000004                   TYPE                    ;TYPE MSG
4475  021742  012705  000744           MOV     #ASEQCF,R5      ;SET ADDRESS OF ENTRY
4476  021746  012701  000002           MOV     #2,R1           ;SET SIZE OF ENTRY
4477  021752  012702  000001           MOV     #1,R2           ;SET UPPER LIMIT
4478  021756  012703  000000           MOV     #0,R3           ;SET LOWER LIMIT
4479  021762  004737  023340           JSR     PC,TTR          ;GO GET INPUT
4480  021766  005037  000740   ASEQ0:  CLR     ADRVN           ;CLEAR DRV NUM
4481  021772  004737  022110   ASEQ1:  JSR     PC,HRDS         ;GO SELECT HARDWARE CONFIGURATION
4482  021776  012704  026527           MOV     #MSG101,R4
4483  022002  000004                   TYPE                    ;TYPE MSG
4484  022004  012704  026556           MOV     #MSG102,R4
4485  022010  000004                   TYPE                    ;TYPE MSG
4486  022012  013703  000740           MOV     ADRVN,R3
4487  022016  104400                   TYPOCT                  ;PRINT TM03
4488  022020  012704  026565           MOV     #MSG103,R4
4489  022024  000004                   TYPE                    ;TYPE MSG
4490  022026  012700  000746           MOV     #UN1,R0         ;POINT TO START OF SLAVE TABLE
4491  022032  005710           ASEQ2:  TST     (R0)            ;SEE IF END
4492  022034  100403                   BMI     ASEQ3           ;IF SO: BR
4493  022036  012003                   MOV     (R0)+,R3
4494  022040  104400                   TYPOCT                  ;PRINT SLAVE TABLE
4495  022042  000773                   BR      ASEQ2           ;DO ALL
4496  022044  004737  022274   ASEQ3:  JSR     PC,AMOD1        ;GO DO MODE 1(NRZ)
4497  022050  004737  022434           JSR     PC,AMOD2        ;GO DO MODE 2(PE)
4498  022054  022737  000007  000740 ASEQ4:  CMP     #7,ADRVN        ;SEE IF DONE ALL DRIVES
4499  022062  001403                   BEQ     ASEQX           ;IF SO: BR
4500  022064  005237  000740           INC     ADRVN           ;BUMP DRIVE NUMBER
4501  022070  000740                   BR      ASEQ1           ;CONTINUE
4502  022072  005737  000744   ASEQX:  TST     ASEQCF          ;SEE IF CONTINUOUS AUTO SEQ
4503  022076  001003                   BNE     ASEQXX          ;IF SO: BR
4504  022100  000137  005004           JMP     TEND
4505  022104  000000                   HALT
4506  022106  000727           ASEQXX: BR      ASEQ0
```

```
 4507
 4508                                        ;SUBROUTINE TO SELECT AUTO SEQUENCE HARDWARE********
 4509
 4510  022110  005037  005054        HRDS:   CLR     REOTC           ;CLEAR EOT UNIT CNTR
 4511  022114  005037  000644                CLR     TEMP1
 4512  022120  012777  000040  156372        MOV     #40,@CS         ;INIT
 4513  022126  013777  000740  156364        MOV     ADRVN,@CS       ;SET DRIVE
 4514  022134  032777  010000  156356        BIT     #10000,@CS      ;TEST FOR NON-EXISTANT DRIVE
 4515  022142  001403                        BEQ     2$              ;IF DRIVE AVAIL: BR
 4516  022144  005726                1$:     TST     (SP)+           ;RESET STACK POINTER
 4517  022146  000137  022054                JMP     ASEQ4           ;GO SEE IF TRIED ALL DRIVES
 4518  022152  005000                2$:     CLR     R0
 4519  022154  012701  000746                MOV     #UN1,R1         ;SET START OF SLAVE TABLE
 4520  022160  005737  003040                TST     CHNFLG          ;BRANCH IF NOT IN CHAIN MODE
 4521  022164  001410                        BEQ     3$
 4522  022166  122737  000006  000041        CMPB    #6,@#41         ;BRANCH IF NOT LOADED VIA TMDP
 4523  022174  001004                        BNE     3$
 4524  022176  005737  000740                TST     ADRVN           ;BRANCH IF NOT DRIVE 0
 4525  022202  001001                        BNE     3$
 4526  022204  005200                        INC     R0              ;DO NOT TEST SLAVE 0
 4527  022206  010077  156330        3$:     MOV     R0,@TC          ;SELECT SLAVE
 4528  022212  032777  010000  156302        BIT     #10000,@DS      ;SEE IF SLAVE AVAIL FOR TEST(MOL)
 4529  022220  001403                        BEQ     4$              ;IF NOT: BR
 4530  022222  005237  000644                INC     TEMP1           ;SET SLAVE FOUND FLAG
 4531  022226  010021                        MOV     R0,(R1)+        ;LOAD SLAVE TABLE
 4532  022230  005200                4$:     INC     R0              ;STEP TO NEXT SLAVE
 4533  022232  022700  000010                CMP     #10,R0          ;BRANCH IF ALL SLAVE NOT DONE
 4534  022236  001363                        BNE     3$
 4535  022240  005737  000644        5$:     TST     TEMP1           ;SEE IF FOUND ANY SLAVES
 4536  022244  001737                        BEQ     1$              ;IF NOT: BR
 4537  022246  013737  000644  005054        MOV     TEMP1,REOTC     ;SET NUMBER OF UNITS
 4538  022254  000337  000644                SWAB    TEMP1
 4539  022260  053737  000644  005054        BIS     TEMP1,REOTC     ;SET EOT CNTR
 4540  022266  012711  177777                MOV     #-1,(R1)        ;TERMINATE SLAVE TABLE
 4541  022272  000207                        RTS     PC              ;RETURN TO SEQ
```

```
4542
4543                                        ;SUBROUTINE TO SELECT NRZ AUTO TEST MODE**********
4544
4545  022274  005037  000656        AMOD1:  CLR     BLCNTR          ;ASSURE BLOCK COUNTER IS 0
4546  022300  012701  000746                MOV     #UN1,R1         ;GET START OF SLAVE TABLE
4547  022304  052721  001700        1$:     BIS     #1700,(R1)+     ;SET ALL SLAVE TO NRZ,NORM,ODD
4548  022310  022711  177777                CMP     #-1,(R1)        ;LOOP UNTIL REACED END OF TABLE
4549  022314  001373                        BNE     1$
4550  022316  004737  005070                JSR     PC,RWNDA        ;GO REWIND ALL AVAIL SLAVES
4551  022322  012737  000006  000742        MOV     #6,ABLCNT       ;SET NUMBER OF BLOCKS FOR MODE 1
4552  022330  012737  174000  000556        MOV     #-4000,FMCNT    ;SET FC = 4000
4553  022336  012737  000100  000554        MOV     #100,RCNT       ;SET REC CNTR = 100
4554  022344  013737  000740  000550        MOV     ADRVN,DVN       ;SELECT DRIVE
4555  022352  012737  000001  000560        MOV     #1,PATRN        ;SELECT PATTERN 1
4556  022360  005037  000564                CLR     TMEX            ;ASSURE NO TMK
4557  022364  005037  000570                CLR     INTRF           ;ASSURE NORMAL READ
4558  022370  004737  003346                JSR     PC,STAUTO       ;GO DO AUTO MODE 1
4559  022374  012737  000010  000560        MOV     #10,PATRN       ;SELECT PATTERN 10
4560  022402  004737  003346                JSR     PC,STAUTO       ;GO DO PATTERN 10
4561  022406  012737  000014  000560        MOV     #14,PATRN       ;SELECT PATTERN 14
4562  022414  004737  003346                JSR     PC,STAUTO
4563  022420  012737  177777  000560  3$:   MOV     #-1,PATRN       ;SELECT AUTO RANDOM DATA
4564  022426  004737  003346                JSR     PC,STAUTO
4565  022432  000207                        RTS     PC              ;RETURN TO SEQ
```

```
4566
4567                                        ;SUBROUTINE TO SELECT PE AUTO TEST MODE**********
4568
4569  022434  005037  000656        AMOD2:  CLR     BLCNTR          ;CLEAR BLOCK CNTR
4570  022440  012701  000746                MOV     #UN1,R1         ;SET START OF SLAVE TABLE
4571  022444  042711  001700        1$:     BIC     #1700,(R1)      ;CLEAR NRZ
4572  022450  052721  002300                BIS     #2300,(R1)+     ;SET TO PE NORM, 000
4573  022454  022711  177777                CMP     #-1,(R1)        ;LOOP UNTIL END OF TABLE
4574  022460  001371                        BNE     1$
4575  022462  004737  005070                JSR     PC,RWNDA        ;REWIND ALL SLAVES
4576  022466  012737  000006  000742        MOV     #6,ABLCNT       ;SET AUTO BLOCK COUNT
4577  022474  012737  174000  000556        MOV     #-4000,FMCNT    ;SET FC = 4000
4578  022502  012737  000100  000554        MOV     #100,RCNT       ;SET REC CNTR TO 100
4579  022510  012737  000010  000560        MOV     #10,PATRN       ;SELECT PATTERN 10
4580  022516  004737  003346                JSR     PC,STAUTO       ;GO DO AUTO SEQ
4581  022522  012737  000014  000560        MOV     #14,PATRN       ;SELECT PATTERN 14
4582  022530  004737  003346                JSR     PC,STAUTO
4583  022534  012737  000015  000560        MOV     #15,PATRN       ;SELECT PATTERN 15
4584  022542  004737  003346                JSR     PC,STAUTO
4585  022546  012737  177777  000742        MOV     #-1,ABLCNT      ;FORCE TO END OF TAPE
4586  022554  012737  177777  000560        MOV     #-1,PATRN       ;SELECT AUTO RANDOM DATA
4587  022562  004737  003346                JSR     PC,STAUTO
4588  022566  000207            3$:     RTS     PC                  ;RETURN TO SEQ
4589
4590
```

```
4591
4592                    ;*********************************************************
4593                    ;ERROR HEADER PRINT SUBROUTINE:
4594                    ;
4595                    ;THIS ROUTINE IS USED TO PRINT OUT A HEADER
4596                    ;WITH EACH ERROR MESSAGE.  THE PRINT IS IN TWO
4597                    ;LINES AND CONTAINS THE FOLLOWING INFORMATION.
4598                    ;LINE 1:  DRIVE NO. SLAVE NO. DENSITY PARITY FORMAT
4599                    ;LINE 2:  CURRENT BLOCK NUMBER, RECORD NUMBER IN
4600                    ;WHICH THE ERROR OCCURED PLUS THE TOTAL NUMBER
4601                    ;OF RECORDS IN THIS BLOCK, THE RECORD SIZE (NUMBER
4602                    ;OF CHARACTERS), AND THE ERROR TYPE (READ,WRITE, SPACE, ETC)
4603                    ;PLUS THE TAPE DIRECTION (FORWARD OR REVERSE).
4604                    ;ALL NUMBERS ARE IN OCTAL.
4605                    ;*********************************************************
4606
4607  022570 012704 024546      PAPRT:  MOV     #MSG12,R4
4608  022574 000004                     TYPE                    ;TYPE MSG
4609  022576 013703 000550              MOV     DVN,R3
4610  022602 104400                     TYPOCT                  ;PRINT DRIVE NUMBER
4611  022604 012704 024532              MOV     #MSG11,R4
4612  022610 000004                     TYPE                    ;TYPE MSG
4613  022612 013703 000552              MOV     UDES,R3
4614  022616 042703 177770              BIC     #177770,R3
4615  022622 104400                     TYPOCT                  ;PRINT UNIT NUMBER
4616  022624 012704 026105              MOV     #MSG60,R4
4617  022630 000004                     TYPE                    ;TYPE MSG
4618  022632 013703 000552              MOV     UDES,R3
4619  022636 000303                     SWAB    R3
4620  022640 042703 177770              BIC     #177770,R3
4621  022644 104400                     TYPOCT                  ;PRINT DENSITY
4622  022646 012704 026111              MOV     #MSG61,R4
4623  022652 000004                     TYPE                    ;TYPE MSG
4624  022654 005003                     CLR     R3
4625  022656 032737 000010 000552      BIT     #10,UDES
4626  022664 001402                     BEQ     PAPRTO
4627  022666 012703 000001              MOV     #1,R3
4628  022672 104400           PAPRTO: TYPOCT                  ;PRINT PARITY
4629  022674 012704 026115              MOV     #MSG62,R4
4630  022700 000004                     TYPE                    ;TYPE MSG
4631  022702 013703 000552              MOV     UDES,R3
4632  022706 000241                     CLC
4633  022710 006003                     ROR     R3
4634  022712 006003                     ROR     R3
4635  022714 006003                     ROR     R3              ;PONTION FORMAT
4636  022716 006003                     ROR     R3
4637  022720 042703 177760              BIC     #177760,R3
4638  022724 104400                     TYPOCT                  ;PRINT FORMAT
4639  022726 012704 024507              MOV     #MSG8,R4
4640  022732 000004                     TYPE                    ;TYPE MSG
4641  022734 032777 000400 155646       BIT     #400,@SWR       ;SEE IF RANDOM DATA
4642  022742 001406                     BEQ     PAPRTB          ;IF NOT: BR
4643  022744 012737 000122 000640 PAPRTA: MOV   #122,TOB
4644  022752 004737 023744              JSR     PC,TOG          ;PRINT R
4645  022756 000411                     BR      PAPRTD
4646  022760 005737 000736      PAPRTB: TST     ASEQF           ;SEE IF AUTO SEQ
```

```
4647  022764  001403                          BEQ     PAPRTC          ;IF NOT: BR
4648  022766  005737  000560                  TST     PATRN           ;SEE IF AUTO RANDOM
4649  022772  100764                          BMI     PAPRTA          ;IF SO: BR
4650  022774  013703  000560          PAPRTC: MOV     PATRN,R3
4651  023000  104400                          TYPOCT                  ;PRINT PATRN NUMBER
4652  023002  012704  024564          PAPRTD: MOV     #MSG13,R4
4653  023006  000004                          TYPE                    ;TYPE MSG
4654  023010  013703  000656                  MOV     BLCNTR,R3
4655  023014  104400                          TYPOCT                  ;PRINT NUMBER
4656  023016  012704  024572                  MOV     #MSG14,R4
4657  023022  000004                          TYPE                    ;TYPE MSG
4658  023024  010003                          MOV     R0,R3
4659  023026  032737  000010  000674          BIT     #10,MTC1        ;SEE IF WRITE OPERATION
4660  023034  001404                          BEQ     PAPRT1          ;IF SO: BR
4661  023036  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
4662  023044  001016                          BNE     PAPRT3          ;IF SO: BR
4663  023046  013703  000554          PAPRT1: MOV     RCNT,R3
4664  023052  005737  000700                  TST     TMFLG           ;SEE IF TAPE MARK TIME
4665  023056  001010                          BNE     PAPRT2          ;IF SO: BR
4666  023060  022737  012072  000664          CMP     #81,RTRN
4667  023066  001003                          BNE     PAPRTY          ;IF NOT BACK SPACE: BR
4668  023070  005737  000714                  TST     RTYFL
4669  023074  001402                          BEQ     PAPRT3          ;IF NOT RETRY: BR
4670  023076  160003          PAPRTY: SUB     R0,R3           ;GET RECORD NUMBER
4671  023100  005203          PAPRT2: INC     R3
4672  023102  104400          PAPRT3: TYPOCT                  ;PRINT RECORD NUMBER
4673  023104  012737  000055  000640          MOV     #55,T0B         ;LOAD DASH (-)
4674  023112  004737  023744                  JSR     PC,T0G          ;PRINT DASH (-)
4675  023116  013703  000554                  MOV     RCNT,R3
4676  023122  104400                          TYPOCT                  ;PRINT RECORD COUNT
4677  023124  012704  024502                  MOV     #MSG7,R4
4678  023130  000004                          TYPE                    ;TYPE MSG
4679  023132  013703  000556                  MOV     FMCNT,R3        ;GET CHARACTER COUNT
4680  023136  005403                          NEG     R3              ;FORM TWO'S COMPLEMENT
4681  023140  104400                          TYPOCT                  ;PRINT RECORD SIZE
4682  023142  012737  000001  000666          MOV     #1,HDRFL        ;SET HEADER FLAG
4683  023150  000207                          RTS     PC              ;RETURN
4684
```

```
                                        ;*********************************************************
                                        ;RANDOM NUMBER GENERATOR SUBROUTINE:
                                        ;
                                        ;THIS SUBROUTINE IS USED TO GENERATE THE RANDOM
                                        ;NUMBERS REQUIRED FOR USE AS RANDOM DATA,
                                        ;RECORD COUNT, AND CHARACTER COUNT.
                                        ;*********************************************************

023152  063737  000630  000626  RANG:  ADD     RANSAV,RANBAS
023160  063737  000626  000630         ADD     RANBAS,RANSAV   ;GET NEW NUMBER
023166  023701  000630                 CMP     RANSAV,R1       ;SEE IF NUMBER TOO BIG
023172  101367                         BHI     RANG            ;IF SO: BR
023174  020237  000630                 CMP     R2,RANSAV       ;SEE IF NUMBER TOO SMALL
023200  101364                         BHI     RANG            ;IF SO: BR
023202  000207                         RTS     PC              ;EXIT

                                ;SUBROUTINE TO GET NEW SOFTWARE SWR

023204  022737  000176  000610  GTSWR: CMP     @#SWREG,SWR     ;BRANCH IF SOFTWARE SWR
023212  001027                         BNE     1$              ;NOT INVOKED
023214  004737  023274                 JSR     PC,.SAVE        ;SAVE REGISTERS ON THE STACK
023220  012704  024424                 MOV     #SMSWR,R4       ;TYPE 'SWR = '
023224  000004                         TYPE                    ;TYPE MSG
023226  017703  155356                 MOV     @SWR,R3         ;GET CURRENT SWR
023232  104400                         TYPOCT
023234  012704  024434                 MOV     #SMNEW,R4       ;ASK FOR NEW SETTING
023240  000004                         TYPE                    ;TYPE MSG
023242  013705  000610                 MOV     SWR,R5          ;TTR ROUTINE RETURNS VALUE TO (R5)
023246  012701  000007                 MOV     #7,R1           ;LIMIT RESPONSE TO 7 CHARS
023252  012702  177777                 MOV     #177777,R2      ;BETWEEN 0 AND 177777
023256  012703  000000                 MOV     #0,R3
023262  004737  023340                 JSR     PC,TTR          ;GET RESPONSE
023266  004737  023316                 JSR     PC,.RESTORE     ;RESTORE REGISTERS
023272  000207                 1$:     RTS     PC              ;RETURN

                                ;;ROUTINE TO SAVE REGISTERS ON THE STACK
023274  010546                 .SAVE:  MOV     %5,-(SP)        ;;R5 IS SAVED AT 12(SP)
023276  010446                         MOV     %4,-(SP)        ;;R4 IS SAVED AT 10(SP)
023300  010346                         MOV     %3,-(SP)        ;;R3 IS SAVED AT 6(SP)
023302  010246                         MOV     %2,-(SP)        ;;R2 IS SAVED AT 4(SP)
023304  010146                         MOV     %1,-(SP)        ;;R1 IS SAVED AT 2(SP)
023306  010046                         MOV     %0,-(SP)        ;;R0 IS SAVED AT (SP)
023310  016646  000014                 MOV     14(SP),-(SP)    ;;PUSH RETURN PC ON THE STACK
023314  000207                         RTS     PC              ;;RETURN TO CALLER

                                ;;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
023316  012666  000014         .RESTORE:MOV    (SP)+,14(SP)    ;;STORE RETURN PC ON STACK
023322  012600                         MOV     (SP)+,%0
023324  012601                         MOV     (SP)+,%1
023326  012602                         MOV     (SP)+,%2
023330  012603                         MOV     (SP)+,%3
023332  012604                         MOV     (SP)+,%4
023334  012605                         MOV     (SP)+,%5
023336  000207                         RTS     PC              ;;RETURN
```

```
4741                                    ;*******************************************************
4742                                    ;TTY ENTRY SUBROUTINE:
4743                                    ;
4744                                    ;THIS SUBROUTINE IS USED BY THE TEST CONDITION
4745                                    ;ENTRY ROUTINE TO READ THE RESPONSE ENTERED
4746                                    ;AT THE TTY AND CHECK THEM FOR LEGALITY AND
4747                                    ;LIMITS.  ALL RESPONSE MUST BE TYPED IN OCTAL
4748                                    ;(0-7) AND MUST FALL WITHIN THE LIMITS SET BY
4749                                    ;THE CALLING ROUTINE.
4750                                    ;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
4751                                    ;A QUESTION MARK (?) IS TYPED (') AND THE RESPONSE
4752                                    ;MAY BE REENTERED.
4753                                    ;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
4754                                    ;MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
4755                                    ;CARRIAGE RETURN
4756                                    ;*******************************************************
4757
4758  023340  010146          TTR:    MOV     R1,-(SP)        ;SAVE CHAR COUNT
4759  023342  011601          10$:    MOV     (SP),R1         ;RESTORE CHAR COUNT (FOR ^U)
4760  023344  005037  000644          CLR     TEMP1           ;CLEAR FIRST CHARACTER FLAG
4761  023350  005000                  CLR     R0
4762  023352  004737  023572  1$:     JSR     PC,TTIN         ;GO READ CHARACTER
4763  023356  122737  000003  000642  CMPB    #3,TIB          ;BRANCH IF NOT ^C
4764  023364  001003                  BNE     11$
4765  023366  000005                  RESET
4766  023370  000137  000200          JMP     @#200   ;RESTART AT 200
4767  023374  122737  000015  000642  11$:    CMPB    #15,TIB         ;SEE IF CR
4768  023402  001004                  BNE     2$              ;IF NOT: BR
4769  023404  005737  000644          TST     TEMP1           ;SEE IF FIRST CHARACTER
4770  023410  001457                  BEQ     9$              ;IF SO: BR
4771  023412  000451                  BR      6$              ;ELSE GO LOAD VALUE
4772  023414  122737  000025  000642  2$:     CMPB    #25,TIB         ;BRANCH IF NOT CONTROL U
4773  023422  001004                  BNE     21$
4774  023424  012704  025117          MOV     #MSG28,R4       ;TYPE <CR><LF>
4775  023430  000004                  TYPE                    ;TYPE MSG
4776  023432  000743                  BR      10$
4777  023434  122737  000177  000642  21$:    CMPB    #177,TIB                ;BRANCH IF NOT 'RUBOUT'
4778  023442  001011                  BNE     3$
4779  023444  000241                  CLC                     ;REMOVE LAST CHARACTER
4780  023446  006000                  ROR     R0
4781  023450  006200                  ASR     R0
4782  023452  006200                  ASR     R0
4783  023454  012704  027230          MOV     #MSG118,R4      ;TYPE '\'
4784  023460  000004                  TYPE                    ;TYPE MSG
4785  023462  005201                  INC     R1              ;DEC CHAR RECEIVED COUNT
4786  023464  000732                  BR      1$              ;GET NEXT CHARACTER
4787  023466  122737  000060  000642  3$:     CMPB    #60,TIB         ;SEE IF CHAR IS LESS THAN 0
4788  023474  101027                  BHI     TINER
4789  023476  122737  000070  000642  4$:     CMPB    #70,TIB         ;SEE IF CHAR IS GREATER THAN 7
4790  023504  101423                  BLOS    TINER
4791  023506  005237  000644          5$:     INC     TEMP1           ;SET FIRST CHARACTER FLAG
4792  023512  006300                  ASL     R0
4793  023514  006300                  ASL     R0              ;SHIFT 3 LEFT
4794  023516  006300                  ASL     R0
4795  023520  042737  177770  000642  BIC     #177770,TIB     ;STRIP ASCII
4796  023526  053700  000642          BIS     TIB,R0          ;LOAD CHARACTER
```

```
 4797  023532  005301                    DEC     R1              ;SEE IF DONE
 4798  023534  001306                    BNE     1$              ;IF NOT: BR
 4799  023536  020002            6$:     CMP     R0,R2           ;SEE IF EXCEEDED MAXIMUM LIMIT
 4800  023540  101005                    BHI     TIMER
 4801  023542  020300            7$:     CMP     R3,R0           ;SEE IF BELOW MINIMUM LIMIT
 4802  023544  101003                    BHI     TIMER
 4803  023546  010015            8$:     MOV     R0,(R5)         ;LOAD VALUE
 4804  023550  005726            9$:     TST     (SP)+           ;POP CHAR COUNT OFF STACK
 4805  023552  000207                    RTS     PC              ;EXIT
 4806
 4807  023554  012704  025617    TIMER:  MOV     #MSG43,R4
 4808  023560  000004                    TYPE                    ;TYPE MSG
 4809  023562  005726                    TST     (SP)+           ;POP CHAR COUNT OFF STACK
 4810  023564  162716  000020            SUB     #20,(SP)        ;RESET SP TO START OF VALUE ROUTINE
 4811  023570  000207                    RTS     PC              ;REDO VALUE ENTRY
```

```
4812
4813                                        ;TTY READ SUBROUTINE**********
4814
4815  023572  005277  155014        TTIN:   INC     @TKS
4816  023576  105777  155010        1$:     TSTB    @TKS
4817  023602  100375                        BPL     1$
4818  023604  017737  155004  000642        MOV     @TKB,TIB
4819  023612  042737  000200  000642        BIC     #200,TIB        ;STRIP PARITY BIT
4820  023620  013737  000642  000640        MOV     TIB,TOB         ;MOVE CHAR TO TTY OUTPUT BFR
4821  023626  004737  0'  44                JSR     PC,TOG          ;ECHO CHARACTER
4822  023632  000207                        RTS     PC
4823
4824                                        ;TTY OUTPUT SUBROUTINE*************
4825
4826  023634  112437  000640        TTOUT:  MOVB    (R4)+,TOB
4827  023640  105737  000640                TSTB    TOB
4828  023644  001436                        BEQ     3$
4829  023646  122737  000045  000640        CMPB    #45,TOB
4830  023654  001407                        BEQ     1$
4831  023656  122737  000041  000640        CMPB    #41,TOB
4832  023664  001436                        BEQ     TBELL           ;DO BELL
4833  023666  004737  023744                JSR     PC,TOG
4834  023672  000760                        BR      TTOUT
4835  023674  112737  000015  000640  1$:   MOVB    #15,TOB
4836  023702  004737  023744                JSR     PC,TOG
4837  023706  012703  000006                MOV     #6,R3
4838  023712  005037  000640        2$:     CLR     TOB
4839  023716  004737  023744                JSR     PC,TOG
4840  023722  005303                        DEC     R3
4841  023724  001372                        BNE     2$              ;DO FILLERS
4842  023726  112737  000012  000640        MOVB    #12,TOB
4843  023734  004737  023744                JSR     PC,TOG
4844  023740  000735                        BR      TTOUT
4845  023742  000002                3$:     RTI                     ;RETURN
4846
4847  023744  105777  154646        TOG:    TSTB    @TPS
4848  023750  100375                        BPL     TOG
4849  023752  113777  000640  154640        MOVB    TOB,@TPB
4850  023760  000207                        RTS     PC              ;RETURN
4851
4852  023762  012703  000002        TBELL:  MOV     #2,R3
4853  023766  012737  000007  000640  1$:   MOV     #7,TOB
4854  023774  004737  023744                JSR     PC,TOG
4855  024000  005303                        DEC     R3
4856  024002  001371                        BNE     1$
4857  024004  000713                        BR      TTOUT
4858
4859
```

```
                                        ;OCTAL OUTPUT SUBROUTINE***********
024006 005037 024220    OCTP:   CLR     OFL             ;CLEAR FLAG FOR LEADING ZERO
024012 010304                   MOV     R3,R4           ;SEE IF NUMBER IS ZERO
024014 001003                   BNE     OCTP0           ;IF NOT ZERO: BR
024016 004737 024200            JSR     PC,OCTPG1       ;ELSE PRINT ZERO
024022 000447                   BR      OCTP3           ;SPACE AND EXIT
024024 005704           OCTP0:  TST     R4              ;BRANCH IF MSD = 0
024026 100006                   BPL     OCTP1
024030 012704 000001            MOV     #1,R4
024034 004737 024156            JSR     PC,OCTPG        ;PRINT 1
024040 000137 024052            JMP     OCTP2
024044 005004           OCTP1:  CLR     R4
024046 004737 024156            JSR     PC,OCTPG        ;PRINT 0
024052 010304           OCTP2:  MOV     R3,R4
024054 006004                   ROR     R4
024056 006004                   ROR     R4
024060 006004                   ROR     R4              ;POSITION DIGIT
024062 006004                   ROR     R4
024064 000304                   SWAB    R4
024066 004737 024156            JSR     PC,OCTPG        ;PRINT DIGIT 2
024072 010304                   MOV     R3,R4
024074 006004                   ROR     R4
024076 000304                   SWAB    R4
024100 004737 024156            JSR     PC,OCTPG        ;PRINT DIGIT 3
024104 010304                   MOV     R3,R4
024106 006104                   ROL     R4
024110 006104                   ROL     R4
024112 000304                   SWAB    R4
024114 004737 024156            JSR     PC,OCTPG        ;PRINT DIGIT 4
024120 010304                   MOV     R3,R4
024122 006004                   ROR     R4
024124 006004                   ROR     R4
024126 006004                   ROR     R4
024130 004737 024156            JSR     PC,OCTPG
024134 010304                   MOV     R3,R4
024136 004737 024156            JSR     PC,OCTPG        ;PRINT DIGIT 5
024142 012737 000240 000640 OCTP3:  MOV     #240,TOB
024150 004737 023744            JSR     PC,TOG          ;PRINT SPACE
024154 000002                   RTI                     ;EXIT
024156 042704 177770    OCTPG:  BIC     #177770,R4
024162 001004                   BNE     OCTPGO
024164 005737 024220            TST     OFL
024170 001001                   BNE     OCTPGO
024172 000207                   RTS     PC

024174 005237 024220    OCTPGO: INC     OFL
024200 052704 000260    OCTPG1: BIS     #260,R4
024204 010437 000640            MOV     R4,TOB
024210 004737 023744            JSR     PC,TOG
024214 010304                   MOV     R3,R4
024216 000207                   RTS     PC
024220 000000           OFL:    0                       ;FIRST CHAR FLAG
```

```
4914
4915                                        ;DATA CHARACTER OUTPUT SUBROUTINE************
4916
4917  024222  012704  000010        DOUT:   MOV     #10,R4          ;SET NUMBER TO PRINT
4918  024226  110337  000640                MOVB    R3,@TOB
4919  024232  105777  154360        1$:     TSTB    @TPS
4920  024236  100375                        BPL     1$
4921  024240  105737  000640                TSTB    TOB
4922  024244  100004                        BPL     2$
4923  024246  012777  000061  154344        MOV     #061,@TPB
4924  024254  000403                        BR      3$
4925  024256  012777  000060  154334  2$:   MOV     #060,@TPB
4926  024264  006337  000640        3$:     ASL     TOB
4927  024270  005304                        DEC     R4
4928  024272  001357                        BNE     1$
4929  024274  000207                        RTS     PC
4930
4931  024276  013703  000650        DOUTD:  MOV     TEMP3,R3
4932  024302  000303                        SWAB    R3
4933  024304  004737  024222                JSR     PC,DOUT
4934  024310  013703  000650                MOV     TEMP3,R3
4935  024314  004737  024222                JSR     PC,DOUT
4936  024320  000207                        RTS     PC
4937
4938                                        ;TU16 SERIAL NUMBER PRINT SUBROUTINE*******
4939
4940  024322  010304        SNPT:   MOV     R3,R4
4941  024324  000304                SWAB    R4
4942  024326  006004                ROR     R4
4943  024330  006004                ROR     R4
4944  024332  006004                ROR     R4
4945  024334  006004                ROR     R4
4946  024336  004737  024400        JSR     PC,SNPG         ;PRINT FIRST DIGIT
4947  024342  010304                MOV     R3,R4
4948  024344  000304                SWAB    R4
4949  024346  004737  024400        JSR     PC,SNPG         ;PRINT SECOND DIGIT
4950  024352  010304                MOV     R3,R4
4951  024354  006004                ROR     R4
4952  024356  006004                ROR     R4
4953  024360  006004                ROR     R4
4954  024362  006004                ROR     R4
4955  024364  004737  024400        JSR     PC,SNPG         ;PRINT THIRD DIGIT
4956  024370  010304                MOV     R3,R4
4957  024372  004737  024400        JSR     PC,SNPG         ;PRINT FOURTH DIGIT
4958  024376  000207                RTS     PC              ;EXIT
4959  024400  012737  000260  000640  SNPG:   MOV     #260,TOB        ;SET NUMBER BASE
4960  024406  042704  177760                BIC     #177760,R4      ;MASK NUMBER
4961  024412  050437  000640                BIS     R4,TOB          ;BUILD DIGIT
4962  024416  004737  023744                JSR     PC,TOG          ;GO TYPE
4963  024422  000207                        RTS     PC              ;RETURN
4964
```

```
                                                ;ERROR MESSAGES**********

024424 051445 051127 036440 SMSWR:  .ASCIZ  /%SWR = /
024432 000040
024434 047040 053505 036440 SMNEW:  .ASCIZ  / NEW = /
024442 000040
024444 042052 020105    000 MSG1:   .ASCIZ  /*DE /
024451    045 035507 000040 MSG2:   .ASCIZ  /%G; /
024456 041045 020073    000 MSG3:   .ASCIZ  /%B; /
024463    045 047103 000040 MSG4:   .ASCIZ  /%CN /
024470 053452 020105    000 MSG5:   .ASCIZ  /*ME /
024475    052 042522 000040 MSG6:   .ASCIZ  /*RE /
024502 051052 020123    000 MSG7:   .ASCIZ  /*RS /
024507    052 040520 051124 MSG8:   .ASCIZ  /*PATRN /
024514 020116    000
024517    045 047123 020072 MSG9:   .ASCIZ  /%SN: /
024524    000
024525    052 042523 000040 MSG10:  .ASCIZ  /*SE /
024532 051452 040514 042526 MSG11:  .ASCIZ  /*SLAVE NO. /
024540 047040 027117 000040
024546 022445 042045 044522 MSG12:  .ASCIZ  /%%%DRIVE NO. /
024554 042526 047040 027117
024562 000040
024564 025045 047102 000040 MSG13:  .ASCIZ  /%*BN /
024572 051052 020116    000 MSG14:  .ASCIZ  /*RN /
024577    045 020041 020040 MSG15:  .ASCIZ  /%!           BAD RECORD%%/
024604 020040 020040 020040
024612 041040 042101 051040
024620 041505 051117 022504
024626 000045
024630 043040    000        MSG16:  .ASCIZ  / F/
024633    040 000122        MSG17:  .ASCIZ  / R/
024636 020041 047505 020124 MSG20:  .ASCIZ  /! EOT NO: /
024644 047516 020072    000

024651    045 047111 042524 MSG21:  .ASCIZ  /%INTERCHANGE READ = /
024656 041522 040510 043516
024664 020105 042522 042101
024672 036440 000040
024676 020445 046111 042514 MSG22:  .ASCIZ  /%!ILLEGAL BOT: HALT%%%/
024704 040507 020114 047502
024712 035124 044040 046101
024720 022524 022445    000
024725    045 051503 020061 MSG23:  .ASCIZ  /%CS1 /
024732    000
024733    045 041527 000040 MSG23A: .ASCIZ  /%WC /
024740 041045 020101    000 MSG23B: .ASCIZ  /%BA /
024745    045 041506 000040 MSG23C: .ASCIZ  /%FC /
024752 041445 031123 000040 MSG23D: .ASCIZ  /%CS2 /
024760 042045 020123    000 MSG23E: .ASCIZ  /%DS /
024765    045 051105 000040 MSG23F: .ASCIZ  /%ER /
024772 040445 020123    000 MSG23G: .ASCIZ  /%AS /
024777    045 045503 000040 MSG23H: .ASCIZ  /%CK /
025004 042045 020102    000 MSG23I: .ASCIZ  /%DB /
025011    045 051115 000040 MSG23J: .ASCIZ  /%MR /
```

```
5021  025016 042045 020124    000  MSG23K: .ASCIZ  /%DT /
5022  025023    045 041524 000040  MSG23L: .ASCIZ  /%TC /
5023  025030 051445 020116    000  MSG23M: .ASCIZ  /%SN /
5024  025035    045 047041 020117  MSG24:  .ASCIZ  /%!NO INTERRUPT%/
5025  025042 047111 042524 051122
5026  025050 050125 022524    000
5027  025055    045 047041 020117  MSG25:  .ASCIZ  /%!NO MOL: HALT%/
5028  025062 047515 035114 044040
5029  025070 046101 022524    000
5030  025075    045 051104 050117  MSG26:  .ASCIZ  /%DROPS: /
5031  025102 035123 000040
5032  025106 050045 041511 051513  MSG27:  .ASCIZ  /%PICKS: /
5033  025114 020072    000         MSG28:  .ASCIZ  /%/
5034  025117    045    000         MSG30:  .ASCIZ  /%%TE16 AUTO SEQUENCE (DZTED-A)%/
5035  025121    045 052045 030505
5036  025126 020066 052501 047524
5037  025134 051440 050505 042525
5038  025142 041516 020105 042050
5039  025150 052132 042105 040455
5040  025156 022451    000
5041  025161    045 052045 030115  MSG31:  .ASCIZ  /%%TM03-TE16 DATA RELIABILITY TEST (DZTED-A)%/
5042  025166 026463 042524 033061
5043  025174 042040 052101 020101
5044  025202 042522 044514 041101
5045  025210 046111 052111 020131
5046  025216 042524 052123 024040
5047  025224 055104 042524 026504
5048  025232 024501 000045
5049  025236 054524 042520 036040  MSG31A: .ASCIZ  /TYPE <CR> TO TERMINATE ALL REQUESTS & ^C TO RESTART%/
5050  025244 051103 020076 047524
5051  025252 052040 051105 044515
5052  025260 040516 042524 040440
5053  025266 046114 051040 050505
5054  025274 042525 052123 020123
5055  025302 020046 041536 052040
5056  025310 020117 042522 052123
5057  025316 051101 022524    000
5058  025323    045 046123 053101  MSG32:  .ASCIZ  /%SLAVE NUMBER = /
5059  025330 020105 052516 041115
5060  025336 051105 036440 000040
5061  025344 042045 047105 044523  MSG33:  .ASCIZ  /%DENSITY = /
5062  025352 054524 036440 000040
5063  025360 050045 051101 052111  MSG34:  .ASCIZ  /%PARITY = /
5064  025366 020131 020075    000
5065  025373    045 042522 047503  MSG35:  .ASCIZ  /%RECORD COUNT = /
5066  025400 042122 041440 052517
5067  025406 052116 036440 000040
5068  025414 041445 040510 020122  MSG36:  .ASCIZ  /%CHAR COUNT = /
5069  025422 047503 047125 020124
5070  025430 020075    000
5071  025433    045 040520 052124  MSG37:  .ASCIZ  /%PATTERN NUMBER = /
5072  025440 051105 020116 052516
5073  025446 041115 051105 036440
5074  025454 000040
5075  025456 051445 047111 046107  MSG38:  .ASCIZ  /%SINGLE PASS = /
5076  025464 020105 040520 051523
```

```
5077  025472  036440  000040
5078  025476  041445  041522  041440  MSG39:  .ASCIZ  /%CRC CORRECTION (YES=1,NO=0) = /
5079  025504  051117  042522  052103
5080  025512  047511  020116  054450
5081  025520  051505  030475  047054
5082  025526  036517  024460  036440
5083  025534  000040
5084  025536  022445  047105  042524  MSG40:  .ASCIZ  /%%ENTER STALLS%READ = /
5085  025544  020122  052123  046101
5086  025552  051514  051045  040505
5087  025560  020104  020075     000
5088  025565     045  051127  052111  MSG41:  .ASCIZ  /%WRITE = /
5089  025572  020105  020075     000
5090
5091  025577     045  052524  047122  MSG42:  .ASCIZ  /%TURN AROUND = /
5092  025604  040440  047522  047125
5093  025612  020104  020075     000
5094  025617     045  022477     000  MSG43:  .ASCIZ  /%?%/
5095  025623     045  047105  042524  MSG44:  .ASCIZ  /%ENTER YOZZLE STALL = /
5096  025630  020122  047531  055132
5097  025636  042514  051440  040524
5098  025644  046114  036440  000040
5099  025652  042445  051122  040440  MSG45:  .ASCIZ  /%ERR AMT /
5100  025660  052115  000040
5101  025664  043045  020103     000  MSG46:  .ASCIZ  /%FC /
5102  025671     045  040503  000040  MSG47:  .ASCIZ  /%CA /
5103  025676  020445  047516  041040  MSG48:  .ASCIZ  /%!NO BOT ON REWIND: HALT%%/
5104  025704  052117  047440  020116
5105  025712  042522  044527  042116
5106  025720  020072  040510  052114
5107  025726  022445     000
5108  025731     045  047516  020124  MSG49:  .ASCIZ  /%NOT AVAIL /
5109  025736  053101  044501  020114
5110  025744     000
5111  025745     045  046111  042514  MSG50:  .ASCIZ  /%ILLEGAL DRIVE TYPE /
5112  025752  040507  020114  051104
5113  025760  053111  020105  054524
5114  025766  042520  000040
5115  025772  042045  044522  042526  MSG52:  .ASCIZ  /%DRIVE NUMBER = /
5116  026000  047040  046525  042502
5117  026006  020122  020075     000  MSG53:  .ASCIZ  /%FORMAT = /
5118  026013     045  047506  046522
5119  026020  052101  036440  000040
5120  026026  053452  020105  046524  MSG54:  .ASCIZ  /*WE TM/
5121  026034     000
5122  026035     052  042523  052040  MSG55:  .ASCIZ  /*SE TM/
5123  026042  000115
5124  026044  052040  000115          MSG56:  .ASCIZ  / TM/
5125  026050  047045  047117  042455  MSG57:  .ASCIZ  /%NON-EXIST SLAVE/
5126  026056  044530  052123  051440
5127  026064  040514  042526     000
5128  026071     045  051103  020103  MSG58:  .ASCIZ  /%CRC /
5129  026076     000
5130  026077     045  051114  020103  MSG59:  .ASCIZ  /%LRC /
5131  026104     000
5132  026105     052  020104     000  MSG60:  .ASCIZ  /*D /
```

```
5133  026111      052  020120      000  MSG61:  .ASCIZ  /*P /
5134  026115      053  020106      000  MSG62:  .ASCIZ  /*F /
5135  026121      045  047452  044522  MSG64:  .ASCIZ  /%*ORIGINAL ERROR*/
5136  026126  044507  040516  020114
5137  026134  051105  047522  025122
5138  026142      000
5139  026143      045  042522  051124  MSG65:  .ASCIZ  /%RETRY: /
5140  026150  035131  000040
5141  026154  020452  042523  051040  MSG66:  .ASCIZ  /*!SE RTRY /
5142  026162  051124  020131      000
5143  026167      052  042441  040522  MSG67:  .ASCIZ  /*!ERASE/
5144  026174  042523      000
5145  026177      045  042522  042522  MSG68:  .ASCIZ  /%REREV: /
5146  026204  035126  000040
5147  026210  052045  050101  020105  MSG69:  .ASCIZ  /%TAPE MARK = /
5148  026216  040515  045522  036440
5149  026224  000040
5150  026226  020445  047516  042040  MSG70:  .ASCIZ  /%!NO DRY FROM REWIND: HALT%/
5151  026234  054522  043040  047522
5152  026242  020115  042522  044527
5153  026250  042116  020072  040510
5154  026256  052114  000045
5155  026262  047045  047117  042455  MSG71:  .ASCIZ  /%NON-EXIST DRIVE/
5156  026270  044530  052123  042040
5157  026276  044522  042526      000
5158  026303      045  042522  053506  MSG72:  .ASCIZ  /%REFWD: /
5159  026310  035104  000040
5160  026314  053445  042524  051122  MSG73:  .ASCIZ  /%WTERR: /
5161  026322  020072      000
5162  026325      045  042522  044507  MSG74:  .ASCIZ  /%REGISTER START = /
5163  026332  052123  051105  051440
5164  026340  040524  052122  036440
5165  026346  000040
5166  026350  053045  041505  047524  MSG75:  .ASCIZ  /%VECTOR = /
5167  026356  020122  020075      000
5168  026363      045  042504  042522  MSG76:  .ASCIZ  /%DEREV: /
5169  026370  035126  000040
5170  026374  042045  043105  042127  MSG77:  .ASCIZ  /%DEFWD: /
5171  026402  020072      000
5172  026405      045  047041  047117  MSG78:  .ASCIZ  /%!NON-RETRYABLE WRITE ERROR: ER /
5173  026412  051055  052105  054522
5174  026420  041101  042514  053440
5175  026426  044522  042524  042440
5176  026434  051122  051117  020072
5177  026442  051105  000040
5178  026446  020445  047516  026516  MSG79:  .ASCIZ  /%!NON-RETRYABLE READ ERROR: ER /
5179  026454  042522  051124  040531
5180  026462  046102  020105  042522
5181  026470  042101  042440  051122
5182  026476  051117  020072  051105
5183  026504  000040
5184  026506  020445  042441  042116  MSG100: .ASCIZ  /%!!END OF PASS %/
5185  026514  047440  020106  040520
5186  026522  051523  022440      000
5187  026527      045  025045  025052  MSG101: .ASCIZ  /%%**********************/
5188  026534  025052  025052  025052
```

```
5189  026542  025052  025053  025052
5190  026550  025052  025052  000052
5191  026556  052052  030115  020063  MSG102: .ASCIZ  /*TM03 /
5192  026564      000
5193  026565      052  046123  053101  MSG103: .ASCIZ  /*SLAVES /
5194  026572  051505  000040
5195  026576  040445  052125  020117  MSG104: .ASCIZ  /%AUTO CONT: /
5196  026604  047503  052116  020072
5197  026612      000
5198  026613      045  042522  047503  MSG105: .ASCIZ  /%RECOVERED/
5199  026620  042526  042522  000104
5200  026626  020452  041041  042101  MSG106: .ASCIZ  /*!!BAD TAPE OVERFLOW/
5201  026634  052040  050101  020105
5202  026642  053117  051105  046106
5203  026650  053517      000
5204  026653      045  042522  044527  MSG16A: .ASCIZ  /%REWIND TAPE; RESTART AT BLOCK 1/
5205  026660  042116  052040  050101
5206  026666  035505  051040  051505
5207  026674  040524  052122  040440
5208  026702  020124  046102  041517
5209  026710  020113  000061
5210  026714  020445  052441  051116  MSG107: .ASCIZ  /%!!UNRECOVERABLE BAD SPOT/
5211  026722  041505  053117  051105
5212  026730  041101  042514  041040
5213  026736  042101  051440  047520
5214  026744  000124
5215  026746  041045  042101  051040          .ASCIZ  /%BAD RECORD LEFT ON TAPE%/
5216  026754  041505  051117  020104
5217  026762  042514  052106  047440
5218  026770  020116  040524  042520
5219  026776  000045
5220  027000  020452  050041  051517  MSG109: .ASCIZ  /*!!POSITION LOST IN RETRY/
5221  027006  052111  047511  020116
5222  027014  047514  052123  044440
5223  027022  020116  042522  051124
5224  027030  000131
5225  027032  051445  051525  042520  MSG110: .ASCIZ  /%SUSPECT BAD TAPE/
5226  027040  052103  041040  042101
5227  027046  052040  050101  000105
5228  027054  051045  050105  040505  MSG111: .ASCIZ  /%REPEAT: /
5229  027062  035124  000040
5230  027066  041040  042101  052040  MSG112: .ASCIZ  / BAD TAPE SPOTS%/
5231  027074  050101  020105  050123
5232  027102  052117  022523      000
5233
5234  027107      045  051440  043117  MSG113: .ASCIZ  /% SOFT: /
5235  027114  035124  000040
5236
5237  027120  020045  040510  042122  MSG114: .ASCIZ  /% HARD: /
5238  027126  020072      000
5239
5240  027131      045  020441  040510  MSG115: .ASCIZ  /%!!HARD READ ERROR/
5241  027136  042122  051040  040505
5242  027144  020104  051105  047522
5243  027152  000122
5244  027154  020445  047125  052111  MSG116: .ASCIZ  /%!UNIT IS REWINDING: TEST WILL START AT BOT/
```

```
5245  027162  044440  020123  042522
5246  027170  044527  042116  047111
5247  027176  035107  052040  051505
5248  027204  020124  044527  046114
5249  027212  051440  040524  052122
5250  027220  040440  020124  047502
5251  027226  000124
5252  027230  000134                          MSG118: .ASCIZ  /\/
5253  027232  051045  046505  053117          MSG120: .ASCIZ  /%REMOVE TMDP FROM SLAVE TO BE TESTED%/
5254  027240  020105  046524  050104
5255  027246  043040  047522  020115
5256  027254  046123  053101  020105
5257  027262  047524  041040  020105
5258  027270  042524  052123  042105
5259  027276  000045
5260  027300  044045  051101  053504          MSG121: .ASCIZ  /%HARDWARE SWR IN USE%/
5261  027306  051101  020105  053523
5262  027314  020122  047111  052440
5263  027322  042523  000045
5264
5265                                                  .EVEN
5266  027326  000000                          WDATA:  0                       ;WRITE BUFFER
5267
5268          033334                                  .=.+4004
5269  033334  000000                          RDATA:  0                       ;READ BUFFER
5270
5271          000001                                  .END
```

```
ABLCNT  000742
ACTLRC  021012
ADRVN   000740
AM001   022274
AM002   022434
APATS   014740
AS      000526
ASEQ    021734
ASEQCF  000744
ASEQF   000736
ASEQX   022072
ASEQXX  022106
ASEQ0   021766
ASEQ1   021772
ASEQ2   022032
ASEQ3   022044
ASEQ4   022054
BA      000514
BAER    020776
BBC     000660
BCNT    000712
BDPP    000720
BD00    001410
BD10    001430
BD20    001450
BD30    001470
BD40    001510
BD50    001530
BD60    001550
BD70    001570
BKRT    012046
BKSP    011672
BKTM    011754
BKTM0   012016
BLCNTR  000656
BPKP    000722
BP00    001210
BP10    001230
BP20    001250
BP30    001270
BP40    001310
BP50    001330
BP60    001350
BP70    001370
BTADDR  001030
BTFLG   000726
BTOV    007166
BTOVX   007324
BTOV0   007204
BTOV1   007214
BTOV2   007300
BTOV3   007316
BTPRT   007326
BTPRT1  007376
BTPT    000732
BTSTF   000730
BTUR    007402
BT00    001610
BT01    001714
BT02    002020
BT03    002124
BT04    002230
BT05    002334
BT06    002440
BT07    002544
B0      012026
B1      012072
B2      012102
CADER   020774
CC      000530
CCNTR   012126
CHNFLG  003040
CLLAST  015122
CLP     015232
CLPE    015256
CLP2    015314
CLP3    015326
CL0     015042
CL1     015070
CL2     015112
CL3     015174
COMER   021000
CRCC    000566
CRCER   021010
CRCLRC  015024
CRCSV   021016
CS      000520
C1      000510
DATA0   002772
DATA1   002774
DATA10  003012
DATA11  003014
DATA12  003016
DATA13  003020
DATA14  003022
DATA15  003024
DATA2   002776
DATA3   003000
DATA4   003002
DATA5   003004
DATA6   003006
DATA7   003010
DATBL   002770
DATER1  001130
DATR    014760
DAT0    014274
DAT0A   014320
DAT0B   014326
DAT0C   014372
DAT0D   014400
DAT0E   014410
DAT0F   014424
DAT1    014434
DAT1A   014440
DAT10   014556
DAT11   014606
DAT12   014626
DAT13   014650
DAT14   014660
DAT15   014710
DAT2    014454
DAT3    014460
DAT3A   014466
DAT4    014504
DAT5    014514
DAT6    014522
DAT7    014530
DB      000532
DCHK    015366
DCHK0   015414
DEREV1  001170
DEREX   016450
DEREX1  016502
DERFL   000706
DERR    016000
DERR0   016010
DERR0A  016040
DERR0B  016072
DERR0C  016116
DERR0D  016120
DERR1   016146
DERR2   016150
DERR3   016164
DERR4   016166
DERR4A  016326
DERR4B  016374
DERR5   016432
DERR6   016444
DEX     015776
DF0     015670
DF0A    015564
DF0A0   015606
DF0A1   015622
DF0A2   015636
DF0A3   015652
DF0A4   015656
DF0B    015524
DF0B0   015546
DF0C    015504
DF0C0   015514
DF0D    015470
DF0E    015462
DF0F    015454
DF1     015702
DF2     015712
DF3     015730
DF4     015772
DOUT    024222
DOUTD   024276
DPC     016624
DPCG    016632
DPC0    016640
DPC0A   016702
DPC1    016710
DPC1A   016736
DPC2    017000
DPC2A   016742
DPC2B   016764
DPC3    017032
DPPRT   017100
DPPRTX  017234
DPPRT0  017150
DPPRT1  017174
DPPRT2  017210
DROP    016614
DRPK    016602
DRPKF   016514
DRP1    001010
DRP2    001012
DRP3    001014
DRP4    001016
DRP5    001020
DRP6    001022
DRP7    001024
DRP8    001026
DRVER   021002
DS      000522
DSUP    014110
DS0     014116
DS0A    014200
DS0B    014176
DS0C    014140
DS2A    014226
DS3     014232
DS4     014244
DT      000536
DVN     000550
DOFL    014432
EMADDR  000654
ENDFLG  000736
ENDTBL  002770
EOTC0   002650
EOTREC  000662
ER      000524
ERCHK   017236
ERPT    020056
ERPTG   020112
ERPTG1  020160
ERPTT   020076
ERPT0   020174
ERPT1   020242
ERPT2   020300
ERPT3   020336
ERPT4   020400
ERPT5   020422
ERPT5A  020504
ERPT6   020546
ERPT7   020616
ERPX    020652
ERPX0   020620
ERPX1   020740
ERPX2   020772
ERSAV   000724
ERTFL   000734
ER0     017252
ER0A    017320
ER0B    017266
ER1     017326
ER10    020020
ER2     017332
ER2A    017376
ER2A0   017346
ER2A1   017366
ER2B    017412
ER2C    017436
ER2D    017452
ER2E    017500
ER3     017506
ER3A    017544
ER3A1   017602
ER3B    017610
ER4     017614
ER4A    017660
ER4A1   017652
ER6     017664
ER6A    017750
ER7     020002
EXCRC   015362
EXLRC   015364
FC      000516
FCER    021004
FCSAV   000634
FMCNT   000556
FRPRT   021020
GTSWR   023204
```

```
MDRFL   000466      MSG23L  025023      MSG72   026303      RANSAV  000630      R05     010472
MF E    0 ..2       MSG23M  025030      MSG73   026314      RANSET  004276      R06     010514
MRUS    0 110       MSG24   025036      MSG74   026 5       RCNT    000554      R07     010540
INIT    0 .6        MSG25   025 5       MSG75   026 0       RCNTR   012166      R07A    010570
INTRF   000570      MSG26   025075      MSG76   026363      RCSAV   000632      READ    007730
LRCER   021006      MSG27   025106      MSG77   026374      RDA     007770      REGS    000544
LRCSV   021014      MSG28   025117      MSG78   026405      RDATA   033334      REOT    004330
MR      000534      MSG29   024456      MSG79   026446      RDCMD   000562      REOTC   005054
MSG1    024444      MSG30   025121      MSG8    024507      RDERR1  001150      REOTX   004760
MSG10   024525      MSG31   025161      MSG9    024517      RDER1   001110      REOTXX  005050
MSG100  026506      MSG31A  025236      MT 1    000674      RDER2   001112      REOT1A  004376
MSG101  026 7       MSG32   025323      MT T    021716      RDER3   001114      REOT1B  004424
MSG102  026 56      MSG33   025344      MTINTA  021720      RDER4   001116      REOT1C  004444
MSG103  026 65      MSG34   025360      NOP   = 000240      RDER5   001120      REOT1E  004470
MSG104  02 576      MSG35   025373      NRTP    011236      RDER6   001122      REOT1F  004440
MSG105  02 613      MSG36   025414      NRZOF   000 52      RDER7   001124      REOT2   004512
MSG106  026626      MSG37   025433      OCTP    024006      RDER8   001126      REOT2A  004550
MSG107  026714      MSG38   025456      OCTPG   024156      RDEX    010656      REOT3   004576
MSG109  027000      MSG39   025476      OCTPGO  024174      RDFL    015022      REOT4   004626
MSG11   024532      MSG4    024463      OCTPG1  024200      RDRTG   010752      REOT5   004642
MSG110  027032      MSG40   025536      OCTP0   024024      RDRTX   011234      REOT6   004706
MSG111  027054      MSG41   025565      OCTP1   024044      RDRTY   010664      REOT7   004736
MSG112  027066      MSG42   025577      OCTP2   024052      RDRT0   010676      RETRY   000604
MSG113  027107      MSG43   025617      OCTP3   024142      RDRT1   010730      RFHARD  002730
MSG114  027120      MSG44   025623      OFL     024220      RDRT1A  010726      RFSOFT  002670
MSG115  027131      MSG45   025652      PAPRT   022570      RDRT1B  010746      RPCNT   000702
MSG116  027154      MSG46   025664      PAPRTA  022744      RDRT2   011030      RRHARD  002750
MSG118  027230      MSG47   025671      PAPRTB  022760      RDRT3   011052      RRSOFT  002710
MSG12   024546      MSG48   025676      PAPRTC  022774      RDRT4   011056      RSEQ    007416
MSG120  027232      MSG49   025731      PAPRTD  023002      RDRT5   011060      RSEX    007552
MSG121  027300      MSG5    024470      PAPRTY  023076      RDRT5A  011124      RSF     007470
MSG13   024564      MSG50   025745      PAPRT0  022672      RDRT5B  011134      RSFR    007566
MSG14   024572      MSG52   025772      PAPRT1  023046      RDRT6   011164      RSFRX   007726
MSG15   024577      MSG53   026013      PAPRT2  023100      RDRT7   011230      RSFR0   007620
MSG16   024630      MSG54   026026      PAPRT3  023102      RDX     010662      RSFR1   007632
MSG16A  026653      MSG55   026035      PARS    014272      RD0     010026      RSFR2   007664
MSG17   024633      MSG56   026044      PATRN   000560      RD1     010074      RSF0    007524
MSG2    024451      MSG57   026050      PATS    014270      RD1A    010110      RSF1    007540
MSG20   024636      MSG58   026071      PFLG    000672      RD1B    010116      RSR     00 446
MSG21   024651      MSG59   026077      PICK    017034      RD1D    010124      RSTAL   005574
MSG22   024676      MSG6    024475      PIK1    000770      RD10    010602      RTCNT   000704
MSG23   024725      MSG60   026105      PIK2    000772      RD11    010612      RTRN    000664
MSG23A  024733      MSG61   026111      PIK3    000774      RD2     010130      RTYFL   000714
MSG23B  024740      MSG62   026115      PIK4    000776      RD3     010170      RTY1    001050
MSG23C  024745      MSG64   026121      PIK5    001000      RD4     010220      RTY2    001052
MSG23D  024752      MSG65   026143      PIK6    001002      RD4A    010316      RTY3    001054
MSG23E  024760      MSG66   026154      PIK7    001004      RD4A0   010350      RTY4    001056
MSG23F  024765      MSG67   026167      PIK8    001006      RD4A1   010372      RTY5    001060
MSG23G  024772      MSG68   026177      PRB     000624      RD4A2   010414      RTY6    001062
MSG23H  024777      MSG69   026210      PRS     000622      RD4B    010422      RTY7    001064
MSG23I  025004      MSG7    024502      PSW     000606      RD4C    010426      RTY8    001066
MSG23J  025011      MSG70   026226      RANBAS  007626      RD4D    010456      RWND    005056
MSG23K  025016      MSG71   026262      RANG    023152      RD4E    010462      RWNDA   005070
```