TM02/TU16

DATA RELIABILITY

MD-11-DZTUA-F

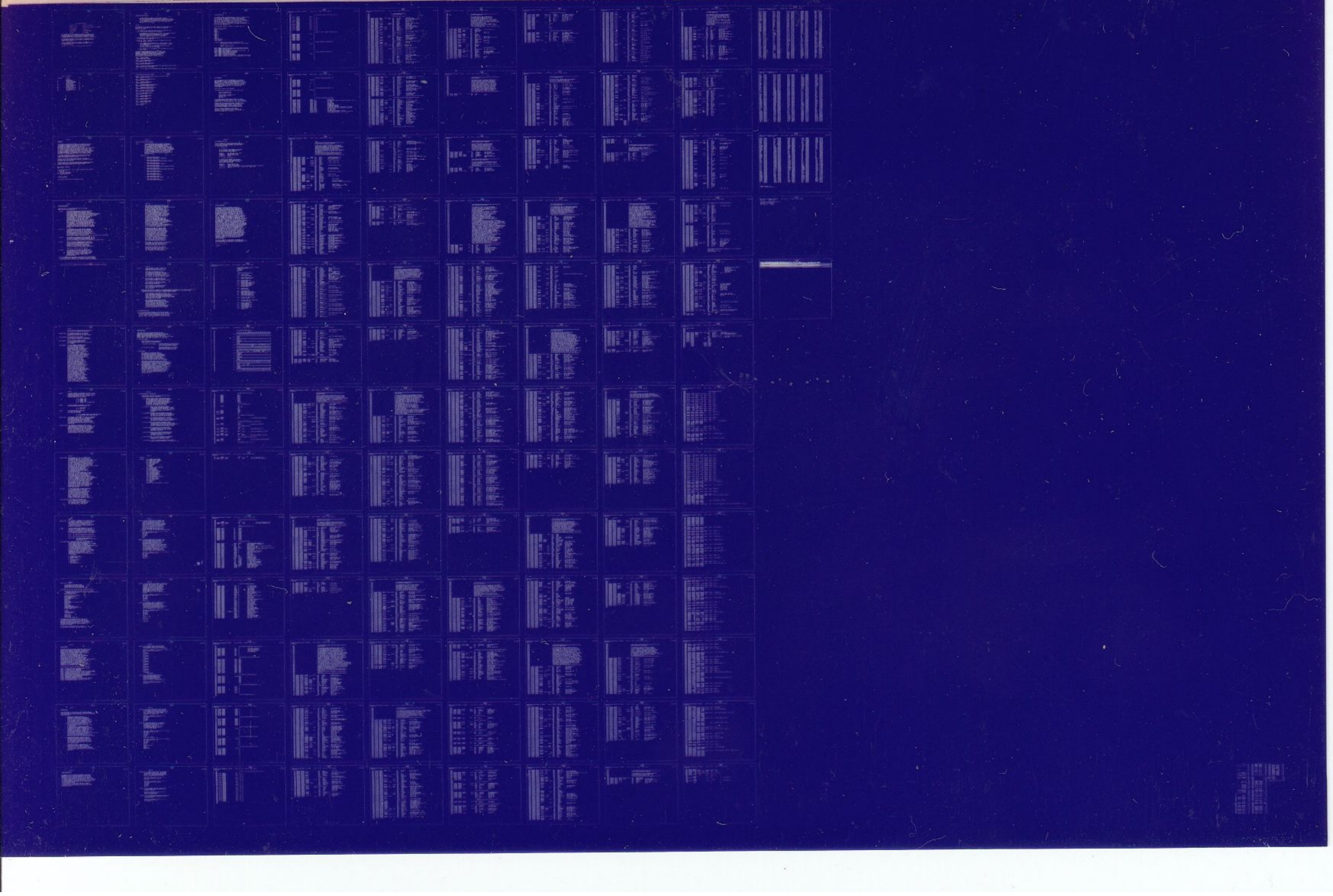
EP-DZTUA-F-DL-A

OCT 1976

digital

Made in U.S.A.

FICHE 1 OF 1



IDENTIFICATION

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

PRODUCT TITLE: TM02/TU16 DATA RELIBILITY PROGRAM

DATE CREATED: 21 APRIL 76

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: R.B. BARNES

TABLE OF CONTENTS

1. ABSTRACT

THIS PROGRAM IS DESIGNED TO BE USED BY AN EXPERIENCED ENGINEER /TECHNICIAN FOR EVALUATION AND DEBUGGING OF MAG TAPE DRIVES. THE PROGRAM IS CAPABLE OF EXERCISING ANY TAPE DRIVE THAT CAN BE OPERATED ON A MASSBUS THROUGH THE TM02 MAG TAPE CONTROLLER. ANY TYPE OF TAPE DRIVE; NRZI,PE,7 OR 9 TRACK MAY BE USED. ANY NUMBER OF DRIVES, SINGLE OR MULTIDRIVE SYSTEMS, UP TO EIGHT (8), MAY BE TESTED BY A SINGLE EXECUTION OF THE PROGRAM. THIS FLEXIBILITY IS POSSIBLE BECAUSE THE PROGRAM HAS NO FIXED PARAMETERS OR TESTING SEQUENCE. THE ENTIRE TEST PLAN, INCLUDING PARAMETERS AND OPERATING SEQUENCE, IS DETERMINED BY THE OPERATOR THROUGH RESPONSES TO TELETYPE REQUESTS AND SETTING OF CONSOLE SWITCHES.

THE PROGRAM PROVIDES FOR TESTING OF ALL TAPE DRIVE FUNCTIONS SUCH AS WRITING,READING,REWINDING,TAPE POSITIONING,EOT - BOT SENSING AND ASSUMES A GOOD RH AND TM02.

HOWEVER; THE RH AND TM02 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 TM02.

2. REQUIREMENTS (HARDWARE)

A. ANY PDP-11 PROCESSOR - WITH OR WITHOUT HARDWARE SWITCH REGISTER
B. 8K OF CORE
C. TELETYPE
D. TM02 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(TM02 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 THERFORE 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.

SEE ITEM 11. (PAGE 24) FOR FULL DETAILS

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.

***IF THE SOFTWARE SWITCH REGISTER IS USED THEN THE FOLLOWING MESSAGE WILL BE TYPED FIRST ; SWR=XXXXXX NEW= THIS WILL ALLOW THE LOADING OF THE SOFTWARE SWITCH REGISTER LOC. 176 BEFORE THE TESTING IS STARTED.

SEQ 0005

(REFER TO SECTION 5 FOR OPERATOR OPTION) THIS PRINTOUT WILL BE TYPED BEFORE THE HEADER.

FQ1

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 TM02) 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 PARITICULAR 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.

H01

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. 0 = 200BPI, NRZI
B. 1 = 556BPI, NRZI
C. 2 = 800BPI, NRZI
D. 3 = 800BPI, NRZI
E. 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)

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.

(PAGE 6)

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. AS EACH PARAMETER REQUEST (PATTERN NUMBER, RECORD 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

```
                (PAGE 7)
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:

        ***SWR=XXXXXX NEW= WILL BE TYPED FIRST IF THE SOFTWARE
           REGISTER IS SELECTED(REFER TO SECTION 8 FOR OPERATOR OPTIONS).
        TU16 TAPE DRIVE TEST
        ENTER CONDITIONS IN OCTAL

        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) 7 CHAN SN: 0022
        DENSITY=(2)
        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, F     PROXIMATELY .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 C CHARACTER AT THE TELETYPE. AS SOON AS THE BUS IS RELEASED BY THE MAG TAPE OPERATION IN PROGRESS, THE PROGRAM WILL RESPOND TO THE CONTROL C INPUT BY TYPING A REQUEST FOR NEW STALL PARAMETERS. THE LAST VALUES THAT WERE ENTERED WILL BE PRINTED AS THE STORED VALUES AND MAY BE CHANGED BY ENTERING NEW VALUES OR LEFT UNCHANGED BY TYPING A CARRIAGE RETURN.
THE YOZZLE STALL IS ALSO DYNAMIC AND CAN BE CHANGED BY TYPING A CNTRL C WHILE DOING A YOZZLE. A YOZZLE STALL REQUEST WILL BE PRINTED AND SHOULD BE RESPONDED TO WITH THE DESIRED VALUE.

8. CONSOLE SWITCH SETTINGS

IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER. IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH REGISTER (LOC. 176) IS USED.

CONTROL:

THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY DOING THE FOLLOWING:

1) TYPE CONTROL G (↑G); THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.

   A) THIS PROGRAM WILL PROCESS THE ‹↑G› EITHER IN FLAG MODE OR INTERRUPT DEPENDING ON WHERE IN THE PROGRAM THE ‹↑G› IS EXCEPTED. THE PROGRAM WILL SERVICE THE INTERRUPT ONLY WHEN THE PRIORTY IS LOWERED TO ALLOW THE TTY TO INTERRUPT.

2) THE MACHINE WILL THEN TYPE: SWR=XXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)

3. 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 LOC. 176 FOLLOWED BY A ‹CR›. (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS WILL BE ALLOWED)
   IF A ‹CR› IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.

   B) IF A CONTROL U (↑U) IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 2.

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
(100000)0=CONTINUE ON ERROR

SW14: 1=PRINT READ/WRITE STATISTICS
(040000)0=DO NOT PRINT STATS

SW13: 1=DO NOT CHECK DATA ERRORS
(020000)0=CHECK DATA ERRORS

SW12: 1=DO NOT CHECK WRITE STATUS ERRORS (NOR CLEAR THEM IF THEY DO OCCUR)
(010000)0=CHECK WRITE STATUS ERRORS

SEQ 0015

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

(004000)0=CHECK READ STATUS ERRORS

SW10: 1=DO NOT PRINT ANY ERRORS (EXCEPT CATASTROPHIC ERRORS)
(020000)0=PRINT ALL ERRORS

SW9: 1=REWIND ALL AVAILABLE TAPES
(010000)0=DO NOT REWIND

SW8: 1=GENERATE RANDOM DATA
(004000)0=USED FIXED DATA

SW7: 1=GENERATE RANDOM CHARACTER COUNT
(002000)0=USE FIXED CHARACTER COUNT

SW6: 1=GENERATE RANDOM RECORD COUNT
(001000)0=USED FIXED RECORD COUNT

SW5: 1=YOZZLE ON CURRENT RECORD
(000400)0=DO NOT YOZZLE ON RECORD

SW4: 1=DO WRITE/READ RETRIES
(000200)0=DO NOT RETRY

SW3: 1=DO NOT READ FORWARD
(000100)0=READ FORWARD

SW2: 1=DO NOT READ REVERSE
(000004)0=READ REVERSE

SW1: 1=READ FORWARD FIRST
(000002)0=READ REVERSE FIRST

SW0: 1=DO NOT WRITE
(000001)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.

(PAGE 14)

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.

*****PROGRAM HALTS*****

***IF THE SOFTWARE SWITCH REGISTER IS USED AND THE PROGRAM HALTS THEN THE OPERATOR CAN PRESS A <↑G> CONTROL G BEFORE HITTING CONTINUE. THIS WILL ALLOW THE OPERATOR TO ENTER DATA INTO LOC. 176 (SWREG). THE FOLLOWING MESSAGE WILL BE TYPED OUT ;
SWR=XXXXXX NEW= (REFER TO SECTION B FOR OPERATOR OPTIONS).

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 TM02 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.

(PAGE 16)

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 TM02 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.

(PAGE 17)

D. EXAMPLES:

GLOSSARY:

BN = CURRENT BLOCK NUMBER
RN = CURRENT RECORD NUMBER
RS = RECORD SIZE, IN FRAMES
WF = WRITE STATUS ERROR
RE = READ STATUS ERROR
SE = SPACE ERROR
TM = TAPE MARK
F = FORWARD
R = REVERSE
CS1 = RH/TU16 CONTROL REGISTER
WC = RH WORD COUNT
BA = RH BUS ADDRESS
FC = TU16 FRAME COUNT
CS2 = RH CONTROLLER STATUS
DS = TU16 DRIVE STATUS
ER = TU16 ERROR REGISTER
AS = ATTENTION SUMMARY
CK = TU16 CHECK CHARACTER
DB = RH DATA BUFFER
MR = TU16 MAINTENENCE REGISTER
DT = TU16 DRIVE TYPE
SN = TU16 SERIAL NUMBER
TC = TU16 TEST CONTROL
*F = DATA FORMAT
*P = PARITY
*D = DENSITY
*PATRN = DATA PATTERN NUMBER (R = RANDOM)

EXAMPLE 1: IN THIS EXAMPLE SLAVE 1 ON TM02 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 TM02 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
```

(PAGE 19)

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 TM02. 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:

```
        TU16 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 TM02 AND ITS SLAVES ARE FOUND, A DIVIDER LINE OF ASTERICKS WILL BE PRINTED FOLLOWED BY A PRINTOUT OF THE TM02 AND ITS SLAVES BEING TESTED. AS EACH TM02 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).

(PAGE 25)

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.

13. LISTING

```
                        .TITLE TM02/TU16 DATA RELIABILITY PROGRAM
                        ;MAINDEC-11-DZTUA-F-D
                        ;21 APRIL 76
                        ;R. BARNES
                        .ABS

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

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

                        ;SW14:  1=PRINT READ/WRITE STATS
                        ;       0=DO NOT PRINT STATS

                        ;SW13:  1=DO NOT CHECK DATA
                        ;       0=CHECK DATA
                        ;SW12:  1=DO NOT CHECk WRITE ERRORS
                        ;       0=CHECK WRITE ERRORS
                        ;SW11:  1=DO NOT CHECK READ ERRORS
                        ;       0=CHECK READ ERRORS
                        ;SW10:  1=DO NOT PRINT ERRORS
                        ;       0=PRINT ERRORS

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

                        ;SW8:   1=USE RANDOM DATA
                        ;       0=USE FIXED DATA PATTERN
                        ;SW7:   1=USE RANDOM CHARACTER COUNT
                        ;       0=USE FIXED CHAR COUNT
                        ;SW6:   1=USE RANDOM RECORD COUNT
                        ;       0=USE FIXED RECORD COUNT

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

                        ;SW4:   1=DO BOTH READ AND WRITE RETRIES
                        ;       0=INHIBIT RETRIES

                        ;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
```

```
                        ;TU16 REGISTER BITS**********

                        ;15 ;14 ;13 ;12 ;11 ;10 ;09 ;08 ;07 ;06 ;05 ;04 ;03 ;02 ;01 ;00 ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;CS1:                   ;SC ;TRE;MCP;SPR;DNA;PSL;A17;A16;RDY;IE ;FUN;FUN;FUN;FUN;GO ;
                        ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;WC:                    ; WORD                                                        ;
                        ; COUNT                                                       ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;BA:                    ; BUS                                                         ;
                        ; ADDRESS                                                     ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;FC.                    ; FRAME                                                       ;
                        ; COUNT                                                       ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;CS2:                   ;DLT;WCE;UPE;NED;NEM;PGE;MXF;MDP;OR ;IR ;CLR;PAT;BAI;U2 ;U1 ;U0 ;
                        ;   ;   ;   ;   ;   ;   ;   ;ERR;   ;   ;   ;   ;   ;   ;   ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;DS:                    ;ATA;ERR;PIP;MOL;WRL;EOT;SPR;DPR;DRY;SSC;PES;SDN;IDB;EOF;BOT;SLA;
                        ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;ER:                    ;CDE;UNS;OPI;DTE;NEF;CS ;FCE;NSG;PEF;INC;DAT;FMT;CNT;RMR;ILR;ILF;
                        ;CRC;   ;   ;   ;   ;ITM;   ;   ;LRC;VPE;BPE;   ;BPE;   ;   ;   ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;ATS:                   ; ATTENTION                                                   ;
                        ; SUMMARY                                                     ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;CC:                    ; CHECK                                                       ;
                        ; CHARACTER                                                   ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;DB:                    ; DATA                                                        ;
                        ; BUFFER                                                      ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;MR:                    ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;WRT;MM ;OP ;OP ;OP ;OP ;MM ;
                        ; 7 ; 6 ; 5 ; 4 ; 3 ; 2 ; 1 ; 0 ; P ;CLK;CLK; 4 ; 3 ; 2 ; 1 ;GO ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;DT:                    ; DRIVE     ; 7 ;   ;SLV;                                    ;
                        ; TYPE      ;CH ;   ;PR ;                                    ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;SN:                    ; SERIAL                                                      ;
                        ; NUMBER                                                      ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;TC:                    ;ACL;FCS;TCW;ENA;SPR;DEN;DEN;DEN;FMT;FMT;FMT;FMT;EVN;SSN;SSN;SSN;
                        ;   ;   ;   ;DT ;   ; 4 ; 2 ; 1 ; 8 ; 4 ; 2 ; 1 ;PAR; 4 ; 2 ; 1 ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

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

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

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

                    000000              .=0
                                        .REPT 200
                                        .+2
                                        HALT
                                        .ENDR
                    000030              .=30
000030  024656                          TRAP30
                    000032              .=32
000032  000340                          340

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

                    000060              .=60
000060  021610                          TTINT                   ;TTY INTERRUPT HANDLER ADDRESS
000062  000000                          0


                                        ;SOFTWARE SWITCH REGISTER LOC. 176**********

                    000176              .=176
000176  000000                          SWREG:  0               ;SOFTWARE SWITCH REGISTER

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

                    000200              .=200
000200  000167  002622                  JMP     START           ;ENTER PARAMETERS VIA TTY

                    000204              .=204
000204  000167  002640                  JMP     STARTC          ;USE FIXED PARAMETERS; HOLD DATA

                    000210              .=210
000210  005067  014660                  CLR     RDFL
000214  000167  002640                  JMP     STARTA          ;USE FIXED PARAMETERS; NEW DATA

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

                    000224              .=224
000224  021714                          MTINT                   ;MAG TAPE INTERRUPT HANDLER ADDRESS
000226  000340                          340


                                        ;AUTO SEQUENCE START**********
```

```
   150            000240                          .=240
   151  000240    005267  000470                  INC     ASEQF           ;SET AUTO SEQUENCE FLAG
   152  000244    000167  002562                  JMP     STAUT           ;GO TO START OF AUTO SEQUENCE
```

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

                000300                  .=300
000300  005267  013526                  INC     SCVFL                   ;SET SHORT CONVERSATION FLAG
000304  000167  002516                  JMP     START                   ;ENTER SHORT PARAMETER LIST

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

000510  172440                  C1:     172440
000512  172442                  WC:     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                  C2:     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  177600          FMCNT:  177600          ;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  000000          TMEX:   0               ;TAPE MARK FLAG: 1=TM 0=NO TM
000566  000000          INTRF:  0               ;INTERCHANGE READ 1=YES 0=NO
000570  000000          SPFLG:  0               ;SINGLE PASS 1=YES 0=NO
000572  000001          RSTAL.  1               ;READ STALL
000574  000001          WSTAL:  1               ;WRITE STALL
000576  000001          TSTAL:  1               ;TURN AROUND STAL
000600  002000          YSTAL:  2000            ;YOZZLE STAL
000602  000010          RETRY:  10              ;READ RETRY NUMBER
000604  177776          PSW:    177776          ;PROCESSOR STATUS
000606  177570          SWR:    177570          ;CONSOLE SWITCHES
000610  177560          TKS:    177560          ;TTY READ STATUS REGISTER
000612  177562          TKB:    177562          ;TTY READ BUFFER
000614  177564          TPS:    177564          ;TTY PUNCH STATUS REGISTER
000616  177566          TPB:    177566          ;TTY PUNCH OUTPUT REGISTER
000620  177550          PRS:    177550          ;H/S READER STATUS REGISTER
000622  177552          PRB:    177552          ;H/S READER BUFFER
000624  153624          RANBAS: 153624          ;RANDOM NUMBER GENERATOR BASE
000626  032561          RANSAV: 032561          ;RANDOM NUMBER BUFFER
000630  000000          RCSAV:  0               ;RECORD COUNT SAVE
000632  000000          FCSAV:  0               ;FRAME COUNT SAVE
```

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

000634  000000  TINF:   0               ;TTY ENTERY FLAG
000636  000000  TOB:    0               ;TTY OUTPUT BUFFER
000640  000000  TIB:    0               ;TTY INPUT BUFFER
000642  000000  TEMP1:  0               ;TEMP STORAGE
000644  000000  TEMP2:  0               ;TEMP STORAGE
000646  000000  TEMP3:  0               ;TEMP STORAGE
000650  000000  NRZOF:  0               ;NRZ ONLY FLAG
000652  000000  EMADDR: 0               ;ERROR MSG ADDRESS STORAGE
000654  000000  BLCNTR: 0               ;BLOCK COUNTER
000656  000000  BBC:    0               ;BAD RECORD COUNTER
000660  000000  EOTREC: 0               ;EOT FLAG
000662  000000  RTRN:   0               ;INTERRUUPT RETURN STORAGE
000664  000000  HDRFL:  0               ;HEADER FLAG
000666  000000  STAL:   0               ;DELAY STORAGE
000670  000000  PFLG:   0               ;PRINT FLAG
000672  000000  MTC1:   0               ;MAG TAPE CONT REGISTER BUFFER
000674  000000  UNP:    0               ;UNIT TABLE POINTER
000676  000000  TMFLG:  0               ;TAPE MARK FLAG
000700  000000  RPCNT:  0               ;REPEAT COUNTER
000702  000000  RTCNT:  0               ;RETRY COUNTER
000704  000000  DERFL:  0               ;DATA ERROR FLAG
000706  000000  SERFL:  0               ;STATUS ERROR FLAG
000710  000000  BCNT:   0               ;BIT COUNTER
000712  000000  RTYFL:  0               ;RETRY FLAG
000714  000000  UPS:    0               ;UNIT POINTER SAVE
000716  000000  BDPP:   0               ;BITS DROPPED POINTER
000720  000000  BPKP:   0               ;BITS PICKED POINTER
000722  000000  ERSAV:  0               ;ERROR SAVE LOC
000724  000000  BTFLG:  0               ;BAD TAPE FLAG
000726  000000  BTSTF:  0               ;STATISTIC PRINT FLAG
000730  000000  BTPT:   0               ;BAD TAPE POINTER
000732  000000  ERTFL:  0               ;ERASE FLAG
000734  000000  ASEQF:  0               ;AUTO SEQ FLAG
000736  000000  ADRVN:  0               ;UTO SEQ DRIVE NUMBER
000740  000000  ABLCNT: 0               ;AUTO BLOCK COUNTER
000742  000000  ASEQCF: 0               ;AUTO SEQ CONTINUOUS FLAG
000744  000000  EOPBI:  0               ;EOP FLAG
```

```
                                        ;UNIT ORDER AND DESCRIPTION TABLE **********
000746  000000          UN1:    0               ;THIS TABLE IS LOADED
000750  000000          UN2:    0               ;WITH UNIT NUMBERS AND
000752  000000          UN3:    0               ;THEIR DESCRIPTIONS IN
000754  000000          UN4:    0               ;THE ORDER THAT THEY
000756  000000          UN5:    0               ;WILL BE TESTED
000760  000000          UN6:    0
000762  000000          UN7:    0
000764  000000          UN8:    0
000766  177777          UNX:    -1

                                        ;UNIT DROPS AND PICKS POINTERS*********

000770  001210          PIK1:   BP00
000772  001230          PIK2:   BP10
000774  001250          PIK3:   BP20
000776  001270          PIK4:   BP30
001000  001310          PIK5:   BP40
001002  001330          PIK6:   BP50
001004  001350          PIK7:   BP60
001006  001370          PIK8:   BP70
001010  001410          DRP1:   BD00
001012  001430          DRP2:   BD10
001014  001450          DRP3:   BD20
001016  001470          DRP4:   BD30
001020  001510          DRP5:   BD40
001022  001530          DRP6:   BD50
001024  001550          DRP7:   BD60
001026  001570          DRP8:   BD70

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

001030  001610          BTADDR: BT00
001032  001714                  BT01
001034  002020                  BT02
001036  002124                  BT03
001040  002230                  BT04
001042  002334                  BT05
001044  002440                  BT06
001046  002544                  BT07

                                        ;UNIT WRITE RETRY COUNTER**********

001050  000000          RTY1:   0
001052  000000          RTY2:   0
001054  000000          RTY3:   0
001056  000000          RTY4:   0
001060  000000          RTY5:   0
001062  000000          RTY6:   0
001064  000000          RTY7:   0
001066  000000          RTY8:   0

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

```
  303  001070  000000                WTER1:  0
  304  001072  000000                WTER2:  0
  305  001074  000000                WTER3:  0
  306  001076  000000                WTER4:  0
  307  001100  000000                WTER5:  0
  308  001102  000000                WTER6:  0
  309  001104  000000                WTER7:  0
  310  001106  000000                WTER8:  0
  311
  312                                       ;UNIT READ FORWARD ERRORS**********
  313
  314  001110  000000                RDER1:  0
  315  001112  000000                RDER2:  0
   16  001114  000000                RDER3:  0
  317  001116  000000                RDER4:  0
  318  001120  000000                RDER5:  0
  319  001122  000000                RDER6:  0
  320  001124  000000                RDER7:  0
  321  001126  000000                RDER8:  0
  322
  323                                       ;UNIT DATA ERRORS FORWARD**********
  324
  325  001130  000000                DATER1: 0
  326  001132  000000                        0
  327  001134  000000                        0
  328  001136  000000                        0
  329  001140  000000                        0
  330  001142  000000                        0
  331  001144  000000                        0
  332  001146  000000                        0
  333
  334                                       ;UNIT READ REVERSE ERRORS**********
  335
  336  001150  000000                RDERR1: 0
  337  001152  000000                        0
  338  001154  000000                        0
  339  001156  000000                        0
  340  001160  000000                        0
  341  001162  000000                        0
  342  001164  000000                        0
  343  001166  000000                        0
  344
  345                                       ;UNIT DATA ERRORS REVERSE**********
  346
  347  001170  000000                DEREV1: 0
  348  001172  000000                        0
  349  001174  000000                        0
  350  001176  000000                        0
  351  001200  000000                        0
  352  001202  000000                        0
  353  001204  000000                        0
  354  001206  000000                        0
```

```
                                        ;UROPS + PICKS PER CHANNEL PER UNIT**********

001210  000000          BP00:   0
        001230                  .=.+16
001230  000000          BP10:   0
        001250                  .=.+16
001250  000000          BP20:   0
        001270                  .=.+16
001270  000000          BP30:   0
        001310                  .=.+16
001310  000000          BP40:   0
        001330                  .=.+16
001330  000000          BP50:   0
        001350                  .=.+16
001350  000000          BP60:   0
        001370                  .=.+16
001370  000000          BP70:   0
        001410                  .=.+16
001410  000000          BD00:   0
        001430                  .=.+16
001430  000000          BD10:   0
        001450                  .=.+16
001450  000000          BD20:   0
        001470                  .=.+16
001470  000000          BD30:   0
        001510                  .=.+16
001510  000000          BD40:   0
        001530                  .=.+16
001530  000000          BD50:   0
        001550                  .=.+16
001550  000000          BD60:   0
        001570                  .=.+16
001570  000000          BD70:   0
        001610                  .=.+16
```

```

                                                ;UNIT BAD TAPE COUNTER:16 PER SLAVE**********

 001610  000000                      BT00:      0
         001714                                 .=.+102
 001714  000000                      BT01:      0
         002020                                 .=.+102
 002020  000000                      BT02:      0
         002124                                 .=.+102
 002124  000000                      BT03:      0
         002230                                 .=.+102
 002230  000000                      BT04:      0
         002334                                 .=.+102
 002334  000000                      BT05:      0
         002440                                 .=.+102
 002440  000000                      BT06:      0
         002544                                 .=.+102
 002544  000000                      BT07:      0
         002650                                 .=.+102

                                                ;UNIT END OF TAPE COUNTERS 1 PER SLAVE*******

 002650  000000                      EOTC0:     0
 002652  000000                                 0
 002654  000000                                 0
 002656  000000                                 0
 002660  000000                                 0
 002662  000000                                 0
 002664  000000                                 0
 002666  000000                                 0

                                                ;UNIT READ FORWARD SOFT ERROR*******

 002670  000000                      RFSOFT:    0
 002672  000000                                 0
 002674  000000                                 0
 002676  000000                                 0
 002700  000000                                 0
 002702  000000                                 0
 002704  000000                                 0
 002706  000000                                 0

                                                ;UNIT READ REVERSE SOFT ERROR*******

 002710  000000                      RRSOFT:    0
 002712  000000                                 0
 002714  000000                                 0
 002716  000000                                 0
 002720  000000                                 0
 002722  000000                                 0
 002724  000000                                 0
 002726  000000                                 0

```

```
                                        ;UNIT READ FORWARD HARD ERROR*******

002730  000000          RFHARD: 0
002732  000000                  0
002734  000000                  0
002736  000000                  0
002740  000000                  0
002742  000000                  0
002744  000000                  0
002746  000000                  0

                                        ;UNIT READ REVERSE HARD ERROR*******

002750  000000          RRHARD: 0
002752  000000                  0
002754  000000                  0
002756  000000                  0
002760  000000                  0
002762  000000                  0
002764  000000                  0
002766  000000                  0

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

002770  002770          DATBL:  .                       ;ENTRY TABLE
002772  014300          DATA0:  DAT0                    ;EXTERNAL INPUT FROM H/S READER(SEE MAINDEC-11-DZTUF)
002774  014452          DATA1:  DAT1                    ;ALL ONES
002776  014474          DATA2:  DAT2                    ;ALL ZEROS
003000  014502          DATA3:  DAT3                    ;WALKING ONE
003002  014530          DATA4:  DAT4                    ;WALKING ZERO
003004  014542          DATA5:  DAT5                    ;ALTERNATING ONE/ZERO
003006  014552          DATA6:  DAT6                    ;ALTERNATING ZERO/ONE
003010  014562          DATA7:  DAT7                    ;ALTERNATING ONE/ZERO IN ALTERNATING CHARACTERS
003012  014612          DATA10: DAT10                   ;WALKING ONE/ALL ONE IN ALTERNATING CHARACTERS
003014  014644          DATA11: DAT11                   ;ALL BITS 0-377
003016  014666          DATA12: DAT12                   ;ALL BITS 377-0
003020  014712          DATA13: DAT13                   ;ALTERNATING CHARACTERS  0 AND 377
003022  014722          DATA14: DAT14                   ;WALKING ZERO/ALL ZERO IN ALTERNATING CHARACTERS
003024  014754          DATA15: DAT15                   ;AUTO SEQUENCE PATTERN 0,0,-1,-1,-1,0,0
```

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

003026  005067  175702          START:  CLR     ASEQF           ;CLEAR AUTO SEQUENCE FLAG
003032  012767  000001  175574  STAUT:  MOV     #1,TINF         ;SET TTY ENTRY FLAG
003040  005067  012030                  CLR     ROFL            ;CLEAR RANDOM DATA FLAG
003044  000167  000014                  JMP     STARTB
003050  005067  175560          STARTC: CLR     TINF            ;CLEAR TTY INPUT FLAG
003054  000167  000070                  JMP     STARTD
003060  005067  175550          STARTA: CLR     TINF            ;CLEAR TTY ENTRY FLAG
003064  012700  000636          STARTB: MOV     #TOB,R0
003070  012701  000037                  MOV     #37,R1
003074  005020                  STARTO: CLR     (R0)+           ;CLEAR FLAGS AND COUNTERS
003076  005301                          DEC     R1
003100  001375                          BNE     STARTO
003102  012706  000500                  MOV     #500,SP         ;SET STACK POINTER
003106  004767  001140                  JSR     PC,RANSET       ;GO RESET RANDOM BASE
003112  012700  001050                  MOV     #RTY1,R0
003116  012701  000750                  MOV     #750,R1
003122  005020                  STARTF: CLR     (R0)+           ;CLEAR STATISTIC COUNTERS
003124  005301                          DEC     R1
003126  001375                          BNE     STARTF
003130  012767  177777  011136          MOV     #-1,PATS        ;PRESET PATTERN
003136  005067  175602                  CLR     EOPB1
003142  012767  000001  175504  STARTE: MOV     #1,BLCNTR       ;PRESET BLOCK COUNTER
003150  012706  000500          STARTD: MOV     #500,SP
003154  012777  000340  175422          MOV     #340,@PSW
003162  013746  000006          SUSWR:  MOV     @#6,-(SP)               ;SAVE VECTORS
003166  013746  000004                  MOV     @#4,-(SP)
003172  012737  003212  000004          MOV     #1$,@#4                 ;SET UP FOR TIMEOUT
003200  022777  177777  175400          CMP     #-1,@SWR                ;REFERENCE HARDWARE SWITCH REGISTER
003206  001402                          BEQ     2$
003210  000404                          BR      3$
003212  022626                  1$:     CMP     (SP)+,(SP)+             ;ADJUST STACK
003214  012767  000176  175364  2$:     MOV     #SWREG,SWR              ;POINT TO SOFTWARE SWITCH REG
003222  012637  000004          3$:     MOV     (SP)+,@#4               ;RESTORE VECTORS
003226  012637  000006                  MOV     (SP)+,@#6
003232  022767  000176  175346          CMP     #SWREG,SWR              ;IS SWREG SELECTED
003240  001007                          BNE     4$
003242  005767  175476                  TST     EOPB1
003246  001004                          BNE     4$
```

```
003250  005067  175470                CLR     EOPB1
003254  004767  021150                JSR     PC,CNTLU                ;CHECK FOR CONTROL G
003260  004767  006772        4$:     JSR     PC,TINP         ;GO GET PARAMETERS FROM TTY
003264  012777  000040  175226        MOV     #40,@CS         ;INITIALIZE
003272  005000                STAUTO: CLR     R0              ;POINT TO FIRST ENTRY
003274  005160  000746        STAR0A: COM     UN1(R0)         ;SEE IF LAST ENTRY
003300  001411                        BEQ     STAR0B          ;IF SO: BR
003302  005160  000746                COM     UN1(R0)
003306  042760  100000  000746        BIC     #100000,UN1(R0) ;CLEAR EOT FLAG
003314  062700  000002                ADD     #2,R0           ;POINT TO NEXT UNIT ENTRY
003320  000167  177750                JMP     STAR0A          ;CONTINUE CLEARING
003324  005160  000746        STAR0B: COM     UN1(R0)
003330  016703  001456                MOV     REOTC,R3
003334  000303                        SWAB    R3
003336  110367  001450                MOVB    R3,REOTC        ;RESTORE EOT CNTR
003342  012777  000100  175240 START1: MOV     #100,@TKS       ;SET TTY INTERRUPT ENABLE
003350  016700  175320                MOV     UNP,R0          ;R0 = UNIT TABLE POINTER
003354  005160  000746        STAR1A: COM     UN1(R0)
003360  001407                        BEQ     STAR1B          ;IF LAST UNIT IN STRING: BR
003362  005160  000746                COM     UN1(R0)
003366  016067  000746  175156        MOV     UN1(R0),UDES    ;LOAD NEXT UNIT DESCRIPTION
003374  000167  000124                JMP     START4
003400  005267  175250        STAR1B: INC     BLCNTR          ;BUMP BLOCK COUNTER
003404  005767  175324                TST     ASEQF           ;SEE IF AUTO SEQ
003410  001413                        BEQ     STAR1C          ;IF NOT: BR
003412  026767  175236  175320        CMP     BLCNTR,ABLCNT   ;SEE IF DONE SEQ
003420  001007                        BNE     STAR1C          ;IF NOT: BR
003422  005160  000746                COM     UN1(R0)         ;RESET UNIT TABLE TERMINATOR
003426  005067  175222                CLR     BLCNTR          ;RESET BLOCK CNTR
003432  005067  175236                CLR     UNP             ;RESET UNIT POINTER
003436  000207                        RTS     PC              ;RETURN TO AUTO SEQ
003440  005067  175230        STAR1C: CLR     UNP
003444  005160  000746                COM     UN1(R0)
003450  005000                        CLR     R0
003452  016067  000746  175072        MOV     UN1(R0),UDES    ;LOAD FIRST UNIT DESCRIPTION
003460  032777  000200  175120        BIT     #200,@SWR       ;SEE IF RANDOM RECORD SIZE
003466  001402                        BEQ     START2          ;IF NOT: BR
003470  004767  006476                JSR     PC,CCNTR        ;GO GENERATE RANDOM RECORD SIZE
003474  032777  000400  175104 START2: BIT     #400,@SWR       ;SEE IF RANDOM DATA
003502  001402                        BEQ     START3          ;IF NOT: BR
003504  004767  011316                JSR     PC,DATR         ;GO GENERATE RANDOM DATA
003510  032777  000100  175070 START3: BIT     #100,@SWR       ;SEE IF RANDOM RECORD COUNT
003516  001402                        BEQ     START4          ;IF NOT: BR
003520  004767  006506                JSR     PC,RCNTR        ;GO GENERATE RANDOM RECORD COUNT
003524  032760  100000  000746 START4: BIT     #100000,UN1(R0) ;SEE IF REACHED EOT
003532  001402                        BEQ     STAR4O          ;IF NOT: BR
003534  000167  000500                JMP     START7          ;ELSE GO TO NEXT UNIT
003540  016777  175004  174752 STAR4O: MOV     DVN,@CS         ;SET DRIVE NUMBER
003546  016777  175000  174766        MOV     UDES,@C2        ;SET UNIT NUMBER
003554  032777  000200  174740        BIT     #200,@DS        ;SEE IF UNIT AVAIL
003562  001013                        BNE     STAR4A          ;IF SO: BR
003564  005367  175076                DEC     STAL
003570  001355                        BNE     START4          ;AWAIT TUR
003572  004767  017064                JSR     PC,PAPRT        ;PRINT HEADER
003576  012704  026124                MOV     #MSG49,R4
003602  104000                TTOUTT  ;PRINT NOT AVAIL
```

```
  597  003604  104006                        STOPP                   ;STOP
  598  003606  000167  177712                JMP     START4          ;RETRY
  599  003612  004767  010244        STAR4A: JSR     PC,DSUP         ;GO SET UP WRITE DATA
  600  003616  004767  001172                JSR     PC,RWND         ;REWIND
  601  003622  004767  001556                JSR     PC,WRITE        ;WRITE
  602  003626  016767  174744  175032        MOV     TSTAL,STAL      ;SET TURN AROUND DELAY
  603  003634  004767  006322                JSR     PC,STALL                ;DELAY
  604  003640  004767  003422                JSR     PC,RSEQ         ;GO TO READ SEQUENCER
  605  003644  016767  174726  175014        MOV     TSTAL,STAL      ;SET TURN AROUND DELAY
  606  003652  004767  006304                JSR     PC,STALL        ;DELAY
  607  003656  032777  040000  174722        BIT     #40000,@SWR     ;SEE IF SHOULD PRINT STATISTICS
  608  003664  001541                        BEQ     START5          ;IF NOT: BR
  609  003666  012700  000001                MOV     #1,R0           ;SET RECORD COUNTER TO 1
  610  003672  004767  016764                JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
  611  003676  004767  000004                JSR     PC,STP          ;GO PRINT STATS
  612  003702  000167  000246                JMP     STPX
  613  003706  004767  013304        STP:    JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
  614  003712  012704  026337                MOV     #MSG65,R4
  615  003716  104000                        TTOUTT  ;PRINT RETRY TOTAL
  616  003720  016704  174750                MOV     UNP,R4
  617  003724  016403  001050                MOV     RTY1(R4),R3
  618  003730  104002                        OCTPP                   ;PRINT RETRIES
  619  003732  012704  026510                MOV     #MSG73,R4
  620  003736  104000                        TTOUTT                  ;PRINT WRITE ERROR TAG
  621  003740  016704  174730                MOV     UNP,R4
  622  003744  016403  001070                MOV     WTER1(R4),R3
  623  003750  104002                        OCTPP                   ;PRINT WRITE ERRORS
  624  003752  012704  026477                MOV     #MSG72,R4
  625  003756  104000                        TTOUTT                  ;PRINT READ FORWARD ERROR TAG
  626  003760  016704  174710                MOV     UNP,R4
  627  003764  016403  001110                MOV     RDER1(R4),R3
  628  003770  104002                        OCTPP                   ;PRINT READ FORWARD ERRORS
  629  003772  012704  027305                MOV     #MSG113,R4
  630  003776  104000                        TTOUTT                  ;PRINT SOFT TAG
  631  004000  016704  174670                MOV     UNP,R4
  632  004004  016403  002670                MOV     RFSOFT(R4),R3
  633  004010  104002                        OCTPP                   ;PRINT FORWARD SOFT ERRORS
  634  004012  012704  027316                MOV     #MSG114,R4
  635  004016  104000                        TTOUTT                  ;PRINT HARD TAG
  636  004020  016704  174650                MOV     UNP,R4
  637  004024  016403  002730                MOV     RFHARD(R4),R3
  638  004030  104002                        OCTPP                   ;PRINT HARD FORWARE ERRORS
  639  004032  012704  026570                MOV     #MSG77,R4
  640  004036  104000                        TTOUTT                  ;PRINT DATA ERROR FORWARD TAG
  641  004040  016704  174630                MOV     UNP,R4
  642  004044  016403  001130                MOV     DATER1(R4),R3
  643  004050  104002                        OCTPP                   ;PRINT DATA ERROR FORWARD NUMBER
  644  004052  012704  026373                MOV     #MSG68,R4
  645  004056  104000                        TTOUTT                  ;PRINT READ ERROR REVERSE TAG
  646  004060  016704  174610                MOV     UNP,R4
  647  004064  016403  001150                MOV     RDERR1(R4),R3
  648  004070  104002                        OCTPP                   ;PRINT REVESE ERROR NUMBER
  649  004072  012704  027305                MOV     #MSG113,R4
  650  004076  104000                        TTOUTT                  ;PRINT SOFT TAG
  651  004100  016704  174570                MOV     UNP,R4
  652  004104  016403  002710                MOV     RRSOFT(R4),R3
```

```
                                                OCTPP                   ;PRINT REVERSE SOFT ERROR
004112  012704  027316                          MOV     #MSG114,R4
004116  104000                                  TTOUTT                  ;PRINT HARD TAG
004120  016704  174550                          MOV     UNP,R4
004124  016403  002750                          MOV     RRHARD(R4),R3
004130  104002                                  OCTPP
004132  012704  026557                          MOV     #MSG76,R4
004136  104000                                  TTOUTT                  ;PRINT DATA ERROR REVERSE TAG
004140  016704  174530                          MOV     UNP,R4
004144  016403  001170                          MOV     DEREV1(R4),R3
004150  104002                                  OCTPP                   ;PRINT DATA REVERSE ERROR NUMBER
004152  000207                                  RTS     PC              ;RETURN
004154  005267  174546          STPX:           INC     BTSTF           ;SET STAT ONLY PRINT
004160  004767  003012                          JSR     PC,BTPRT        ;PRINT BAD TAPE STATS
004164  005067  174536                          CLR     BTSTF           ;CLEAR FLAG
004170  017700  174412          START5:         MOV     @SWR,R0         ;LOAD SWR
004174  042700  177762                          BIC     #177762,R0      ;MASK READ/WRITE SWITCHES
004200  022700  000015                          CMP     #15,R0          ;SEE IF HAVE READ OR WRITE
004204  001420                                  BEQ     START8          ;IF NOT: BR
004206  032777  000200  174306  START6:         BIT     #200,@DS        ;SEE IF HAVE UNIT READY
004214  001011                                  BNE     START7          ;IF SO: BR
004216  005367  174444                          DEC     STAL
004222  001371                                  BNE     START6          ;DELAY FOR TUR
004224  004767  016432                          JSR     PC,PAPRT        ;PRINT HEADER
004230  012704  026124                          MOV     #MSG49,R4
004234  104000                                  TTOUTT                  ;PRINT NOT AVAIL
004236  104006                                  STOPP                           ;STOP
004240  062767  000002  174426  START7:         ADD     #2,UNP          ;POINT TO NEXT UNIT
004246  000167  177070          START8:         JMP     START1          ;CONTINUE

                                                ;RANDOM BASE RESET**********

004252  012767  153624  174344  RANSET:         MOV     #153624,RANBAS  ;RESET BASE
004260  012767  032561  174340                  MOV     #32561,RANSAV   ;RESET BUFFER
004266  016767  174336  174260                  MOV     RCSAV,RCNT      ;RESET RECORD COUNT
004274  016767  174332  174254                  MOV     FCSAV,FMCNT     ;RESET FRAME COUNT
004302  000207                                  RTS     PC
```

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

004304  016777  174242  174230  REOT:   MOV     UDES,@C2        ;LOAD COMMAND REGISTER
004312  012777  000011  174170          MOV     #11,@C1         ;DRIVE CLEAR
004320  032777  000200  174174  REOT1:  BIT     #200,@DS
004326  001774                        BEQ     REOT1           ;AWAIT DRY
004330  012777  000007  174152          MOV     #7,@C1          ;START REWIND
004336  005767  174362                  TST     BTFLG           ;SEE IF BAD TAPE OVERFLOW REWIND
004342  001004                        BNE     REOT1A          ;IF SO: BR
004344  016700  174310                  MOV     EOTREC,R0
004350  042700  100000                  BIC     #100000,R0      ;SET RECORD NUMBER OF EOT
004354  004767  016302          REOT1A: JSR     PC,PAPRT        ;PRINT HEADER
004360  022767  000002  174336          CMP     #2,BTFLG        ;SEE IF POSITION ERROR
004366  001003                        BNE     REOT1B          ;IF NOT: BR
004370  012704  027176                  MOV     #MSG109,R4      ;SET POSITION ERROR MSG
004374  000406                        BR      REOT1F
004376  022767  000001  174320  REOT1B: CMP     #1,BTFLG        ;SEE IF BAD TAPE OVERFLOW
004404  001004                        BNE     REOT1C          ;IF NOT: BR
004406  012704  027007                  MOV     #MSG106,R4      ;SET BAD TAPE OVERFLOW MSG
004412  104000          REOT1F: TTOUTT                          ;PRINT REWIND REASON
004414  000412                        BR      REOT1E
004416  012704  025106  REOT1C: MOV     #MSG20,R4       ;SET EOT MSG
004422  104000          REOT1D: TTOUTT                  ;PRINT MSG
004424  016704  174244          MOV     UNP,R4
004430  005264  002650          INC     EOTCO(R4)       ;BUMP CNTR
004434  016403  002650          MOV     EOTCO(R4),R3
004440  104002                  OCTPP                   ;PRINT EOT CNTR
004442  012704  027034  REOT1E: MOV     #MSG16A,R4
004446  104000                  TTOUTT                  ;PRINT RESTART MSG
004450  005067  174250          CLR     BTFLG           ;CLEAR BAD TAPE FLAG
004454  004767  177226          JSR     PC,STP          ;PRINT STATS
004460  004767  002512          JSR     PC,BTPRT        ;PRINT BAD TAPE STATS
004464  032777  000200  174030  REOT2:  BIT     #200,@DS
004472  001014                  BNE     REOT2A          ;IF DRY: BR
004474  005367  174166          DEC     STAL
004500  001371                  BNE     REOT2           ;WAIT DRY
004502  012767  024747  174142          MOV     #MSG6,EMADDR
004510  004767  016146          JSR     PC,PAPRT        ;PRINT HEADER
004514  012704  026301          MOV     #MSG60,R4
004520  104000                  TTOUTT                  ;PRINT NO DRIVE READY
004522  104006                  STOPP
004524  105367  000262  REOT2A: DECB    REOTC           ;SEE IF LAST UNIT TO REACH EOT
004530  001410                  BEQ     REOT3           ;IF SO: BR
004532  016700  174136          MOV     UNP,R0
004536  052760  100000  000746          BIS     #100000,UN1(R0) ;SET EOT FLAG
004544  005726                  TST     (SP)+           ;RESET STACK POINTER
```

```
747  004546  000167  177466                 JMP     START7          ;GO TO NEXT UNIT
748  004552  000367  000234         REOT3:  SWAB    REOTC
749  004556  016700  000230                 MOV     REOTC,R0
750  004562  000367  000224                 SWAB    REOTC
751  004566  110067  000220                 MOVB    R0,REOTC        ;RESTORE EOT UNIT COUNTER
752  004572  005067  174076                 CLR     UNP
753  004576  016700  174072                 MOV     UNP,R0          ;POINT TO FIRST UNIT
754  004602  016067  000746  173742 REOT4:  MOV     UN1(R0),UDES    ;LOAD UNIT DESCRIPTION
755  004610  016777  173736  173724         MOV     UDES,@C2        ;LOAD COMMAND REGISTER
756  004616  032777  020000  173676 REOT5:  BIT     #20000,@DS
757  004624  001374                 BNE     REOT5           ;AWAIT PIP RESET
758  004626  032777  000002  173666         BIT     #2,@DS          ;SEE IF HAVE BOT
759  004634  001012                 BNE     REOT6           ;IF SO: BR
760  004636  012700  000001                 MOV     #1,R0
761  004642  004767  016014                 JSR     PC,PAPRT        ;PRINT HEADER
762  004646  012704  026072                 MOV     #MSG48,R4
763  004652  104000                 TTOUTT                  ;PRINT BOT ERROR
764  004654  104006                 STOPP
765  004656  016700  174012                 MOV     UNP,R0
766  004662  042760  100000  000746 REOT6:  BIC     #100000,UN1(R0) ;CLEAR EOT FLAG
767  004670  062767  000002  173776         ADD     #2,UNP
768  004676  016700  173772                 MOV     UNP,R0          ;POINT TO NEXT UNIT
769  004702  005160  000746                 COM     UN1(R0)         ;SEE IF LAST UNIT
770  004706  001404                 BEQ     REOT7           ;IF SO: BR
771  004710  005160  000746                 COM     UN1(R0)
772  004714  000167  177662                 JMP     REOT4           ;DO NEXT UNIT
773  004720  005160  000746         REOT7:  COM     UN1(R0)
774  004724  005067  173744                 CLR     UNP             ;CLEAR UNIT POINTER
775  004730  005067  173700                 CLR     TINF            ;CLEAR TTY INPUT FLAG
776  004734  005767  173774                 TST     ASEQF           ;SEE IF AUTO SEQ
777  004740  001402                 BEQ     REOTX           ;IF NOT: BR
778  004742  005726                 TST     (SP)+           ;RESET STACK POINTER
779  004744  000207                 RTS     PC              ;RETURN TO AUTO SEQ
780  004746  004767  177300         REOTX:  JSR     PC,RANSET       ;GO RESET RANDOM BASE
781  004752  012FF7  177777  007314         MOV     #-1,PATS        ;PRESET PATTERN
782  004760  005067  010110                 CLR     RDFL            ;CLEAR RANDOM FLAG
783  004764  005767  173600                 TST     SPFLG           ;SEE IF SINGLE PASS
784  004770  001404                 BEQ     REOTXX          ;IF NOT: BR
785  004772  012704  026702                 MOV     #MSG100,R4
786  004776  104000                 TTOUTT                  ;PRINT END OF PASS
787  005000  104006                 STOPP
788  005002  005267  173736         REOTXX: INC     EOPB1
789  005006  000167  176130                 JMP     STARTE          ;RESTART AT BLOCK NUMBER ONE
790  005012  000000                 REOTC:  0                       ;EOT UNIT COUNTER
```

```
                                        ;*****************************************************
                                        ;REWIND ALL AVAIL TAPES:
                                        ;
                                        ;THIS ROUTINE; ENTERED VIA CONSOLE SWITCH NINE (9),
                                        ;WILL REWIND ALL AVAILABLE TAPES TO BOT NO MATTER
                                        ;WHERE THEY ARE CURRENTLY POSITIONED AND RESUME TESTING
                                        ;ON THE CURRENTLY SELECTED UNIT.
                                        ;*****************************************************

005014  032777  001000  173564  RWND:   BIT     #1000,@SWR      ;SEE IF SHOULD REWIND
005022  001001                          BNE     RWNDA           ;IF SO: BR
005024  000207                          RTS     PC              ;ELSE EXIT
005026  016767  173642  173660  RWNDA:  MOV     UNP,UPS         ;SAVE UNIT POINTER
005034  005067  173634                  CLR     UNP             ;CLEAR POINTER
005040  005067  173614                  CLR     EOTREC          ;CLEAR EOT FLAG
005044  000367  177742                  SWAB    REOTC
005050  016700  177736                  MOV     REOTC,R0
005054  000367  177732                  SWAB    REOTC
005060  110067  177726                  MOVB    R0,REOTC        ;RESTORE EOT UNIT COUNTER
005064  016700  173604          RWND0:  MOV     UNP,R0          ;POINT TO UNIT ENTRY
005070  005160  000746                  COM     UN1(R0)         ;SEE IF LAST ENTRY
005074  001452                          BEQ     RWND2           ;IF SO: BR
005076  005160  000746                  COM     UN1(R0)
005102  032760  100000  000746          BIT     #100000,UN1(R0) ;SEE IF ALREADY REWINDING
005110  001034                          BNE     RWND1A          ;IF SO: BR
005112  016067  000746  173432          MOV     UN1(R0),UDES    ;SET UNIT DESCRIPTION
005120  016777  173426  173414          MOV     UDES,@C2        ;LOAD COMMAND REGISTER
005126  012777  000011  173354          MOV     #11,@C1         ;DRIVE CLEAR
005134  012777  000007  173346          MOV     #7,@C1          ;START REWIND
005142  032777  000200  173352  RWND1:  BIT     #200,@DS
005150  001014                          BNE     RWND1A          ;IF DRY: BR
005152  005367  173510                  DEC     STAL
005156  001371                          BNE     RWND1           ;AWAIT DRY
005160  012767  024747  173464          MOV     #MSG6,EMADDR
005166  004767  015470                  JSR     PC,PAPRT        ;PRINT HEADER
005172  012704  026422                  MOV     #MSG70,R4
005176  104000                          TTOUTT                  ;PRINT NO DRIVE READY
005200  104006                          STOPP
005202  042760  100000  000746  RWND1A: BIC     #100000,UN1(R0) ;CLEAR EOT FLAG
005210  062767  000002  173456          ADD     #2,UNP          ;BUMP POINTER
005216  000167  177642                  JMP     RWND0           ;DO NEXT UNIT
005222  005160  000746          RWND2:  COM     UN1(R0)
005226  005067  173442                  CLR     UNP             ;CLEAR POINTER
005232  016700  173436          RWND3:  MOV     UNP,R0          ;POINT TO UNIT ENTRY
005236  005160  000746                  COM     UN1(R0)         ;SEE IF LAST ENTRY:
005242  001442                          BEQ     RWNDX           ;IF SO: BR
005244  005160  000746                  COM     UN1(R0)
005250  016067  000746  173274          MOV     UN1(R0),UDES    ;SET UNIT DESCRIPTION
005256  016777  173270  173256          MOV     UDES,@C2        ;LOAD COMMAND REGISTER
005264  032777  020000  173230  RWND4:  BIT     #20000,@DS
005272  001374                          BNE     RWND4           ;AWAIT PIP RESET
005274  032777  000002  173220          BIT     #2,@DS          ;SEE IF HAVE BOT
005302  001410                          BEQ     RWND6           ;IF NOT: BR
005304  062767  000002  173362  RWND5:  ADD     #2,UNP          ;BUMP POINTER
005312  012777  000011  173170          MOV     #11,@C1         ;DRIVE CLEAR
005320  000167  177706                  JMP     RWND3           ;DO NEXT UNIT
```

```
  847  005324  012700  000001          RWND6:  MOV     #1,R0
  848  005330  004767  015326                  JSR     PC,PAPRT        ;PRINT HEADER
  849  005334  012704  026072                  MOV     #MSG48,R4
  850  005340  104000                          TTOUTT                  ;PRINT NO BOT
  851  005342  104006                          STOPP
  852  005344  000167  177734                  JMP     RWND5           ;DO NEXT UNIT
  853  005350  005160  000746          RWNDX:  COM     UN1(R0)
  854  005354  016767  173334  173312          MOV     UPS,UNP         ;RESTORE UNIT POINTER
  855  005362  016700  173306                  MOV     UNP,R0
  856  005366  016067  000746  173156          MOV     UN1(R0),UDES    ;RESET UNIT DESCRIPTION
  857  005374  016777  173152  173140          MOV     UDES,@C2
  858  005402  000207                          RTS     PC              :RETURN TO TEST
  859
```

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

005404  005067  173250          WRITE:  CLR     EOTREC          ;CLEAR EOT FLAG
005410  032777  000001  173170          BIT     #1,@SWR         ;SEE IF SHOULD WRITE
005416  001402                          BEQ     WRTE
005420  000167  000552                  JMP     WEX             ;IF NOT: BR
005424  016700  173124          WRTE:   MOV     RCNT,R0         ;R0=RECORD COUNT
005430  012767  024742  173214  W0:     MOV     #MSGS,EMADDR    ;SET ERROR MSG ADDRESS
005436  016777  173114  173052          MOV     FMCNT,@FC       ;LOAD CHAR COUNT
005444  012777  027454  173042          MOV     #WDATA,@BA      ;SET DATA ADDR
005452  112767  000060  173212          MOVB    #60,MTC1        ;SET WRITE OP COMMAND
005460  012767  005472  173174          MOV     #W1,RTRN        ;SET RETURN ADDRESS
005466  000167  013502                  JMP     TAPG            ;GO EXECUTE COMMAND
005472  032777  002000  173022  W1:     BIT     #2000,@DS       ;SEE IF EOT
005500  001405                          BEQ     W2              ;IF NOT AT EOT: BR
005502  010067  173152                  MOV     R0,EOTREC       ;SAVE EOT RECORD NUMBER
005506  052767  100000  173144          BIS     #100000,EOTREC  ;SET EOT FLAG
005514  032777  010000  173064  W2:     BIT     #10000,@SWR     ;SEE IF SHOULD CHECK ERRORS
005522  001002                          BNE     W3              ;IF NOT: BR
005524  004767  011624                  JSR     PC,ERCHK        ;GO CHECK ERRORS
005530  016767  173040  173130  W3:     MOV     WSTAL,STAL      ;SET DELAY
005536  004767  004420                  JSR     PC,STALL        ;DELAY
005542  005767  173144                  TST     RTYFL           ;SEE IF RETRY TIME
005546  001401                          BEQ     W3A             ;IF NOT: BR
005550  000207                          RTS     PC              ;ELSE RETURN
```

```
916  005552  005767  173130          W3A:    TST     SERFL           ;SEE IF WRITE ERROR
917  005556  001450                          BEQ     W5              ;IF NOT: BR
918  005560  016704  173110                  MOV     UNP,R4
919  005564  005264  001070                  INC     WTER1(R4)       ;BUMP WRITE ERROR
920  005570  005067  173112                  CLR     SERFL           ;CLEAR STATUS ERROR FLAG
921  005574  032777  000020  173004          BIT     #20,@SWR        ;SEE IF RETRY
922  005602  001436                          BEQ     W5              ;IF NOT: BR
923  005604  016703  173112                  MOV     ERSAV,R3
924  005610  042703  102700                  BIC     #102700,R3      ;MASK UNRECOVERABLE ERROR
925  005614  001410                          BEQ     W4              ;IF SO: BR
926  005616  004767  015040                  JSR     PC,PAPRT        ;PRINT HEADER
927  005622  012704  026601                  MOV     #MSG78,R4
928  005626  104000                          TTOUTT                  ;PRINT NON-RETRYABLE ERROR TAG
929  005630  004767  003302                  JSR     PC,NRTP         ;PRINT ER FOR NON-RETRYABLE
930  005634  000421                          BR      W5
931  005636  016704  173032          W4:     MOV     UNP,R4
932  005642  005264  001050                  INC     RTY1(R4)        ;BUMP RETRY CNTR
933  005646  032777  002000  172732          BIT     #2000,@SWR      ;SEE IF PRINT ERRORS
934  005654  001003                          BNE     W4A             ;IF NOT: BR
935  005656  012704  026315                  MOV     #MSG64,R4
936  005662  104000                          TTOUTT                  ;PRINT ORIGINAL ERROR TAG
937  005664  005067  173012          W4A:    CLR     RTCNT           ;CLEAR RETRY NUMBER
938  005670  005067  173004                  CLR     RPCNT           ;CLEAR REPEAT COUNTER
939  005674  004767  000342                  JSR     PC,WRTY         ;GO RETRY WRITE ERROR
940  005700  005067  173006          W5:     CLR     RTYFL           ;CLEAR RETRY COUNTER
941  005704  005767  172750                  TST     EOTREC          ;SEE IF EOT FOUND
942  005710  100402                          BMI     W6              ;IF SO: BR
943  005712  005300                          DEC     R0              ;SEE IF DONE ALL
944  005714  001245                          BNE     W0              ;IF NOT: BR
945  005716  005767  172642          W6:     TST     TMEX            ;SEE IF TM
946  005722  001525                          BEQ     WEX             ;IF NOT: BR
947  005724  005267  172746                  INC     TMFLG           ;SET TM FLAG
948  005730  012767  026222  172714  WTM:    MOV     #MSG54,EMADDR   ;POINT TO TM ERROR MSG
949  005736  012767  000026  172726          MOV     #26,MTC1        ;SET TM OP CODE
950  005744  012777  000000  172544          MOV     #0,@FC          ;LOAD FRAME COUNTER
951  005752  012777  027454  172534          MOV     #WDATA,@BA      ;LOAD BUS ADDRESS
952  005760  012767  005772  172674          MOV     #WTM0,RTRN      ;SAVE RETURN ADDRESS
953  005766  000167  013202                  JMP     TAPG            ;WRITE TM
954  005772  032777  010000  172606  WTM0:   BIT     #10000,@SWR     ;SEE IF SHOULD CHECK ERRORS
955  006000  001076                          BNE     WEX
956  006002  032777  000004  172512          BIT     #4,@DS          ;SEE IF TM STATUS
957  006010  001011                          BNE     WTM1            ;IF SO: BR
958  006012  012767  027454  013070          MOV     #WDATA,CADER    ;SET EXPT BUS ADDRESS
959  006020  012767  000001  013070          MOV     #1,DRVER        ;INDICATE ERROR
960  006026  004767  012160                  JSR     PC,ERPT         ;PRINT TM ERROR
961  006032  000404                          BR      WTM2
962  006034  012703  027454          WTM1:   MOV     #WDATA,R3       ;SET EXPT ADDRESS
963  006040  004767  011406                  JSR     PC,ER2          ;GO CHECK FOR OTHER ERRORS
964  006044  005767  172642          WTM2:   TST     RTYFL           ;SEE IF RETRY
965  006050  001401                          BEQ     WTM3            ;IF NOT: BR
966  006052  000207                          RTS     PC              ;ELSE RETURN TO RETRY ROUTINE
967  006054  005767  172626          WTM3:   TST     SERFL           ;SEE IF WRITE ERROR
968  006060  001446                          BEQ     WEX             ;IF NOT: BR
969  006062  016704  172606                  MOV     UNP,R4
970  006066  005264  001070                  INC     WTER1(R4)       ;BUMP WRITE ERROR
971  006072  032777  000020  172506          BIT     #20,@SWR        ;SEE IF SHOULD RETRY
```

```
  972  006100  001436                    BEQ     WEX             ;IF NOT: BR
  973  006102  016703  172614            MOV     ERSAV,R3
  974  006106  042703  102700            BIC     #102700,R3      ;MASK UNRECOVERABLE ERROR
  975  006112  001410                    BEQ     WTM4            ;IF SO: BR
  976  006114  004767  014542            JSR     PC,PAPRT        ;PRINT HEADER
  977  006120  012704  026601            MOV     #MSG78,R4
  978  006124  104000                    TTOUTT                  ;PRINT UNRETRYABLE TAG
  979  006126  004767  003004            JSR     PC,NRTP         ;PRINT ER FOR NON-RETRYABLE
  980  006132  000421                    BR      WEX
  981  006134  005067  172540    WTM4:   CLR     RPCNT           ;CLEAR REPEAT CNTR
  982  006140  016704  172530            MOV     UNP,R4
  983  006144  005264  001050            INC     RTY1(R4)        ;BUMP RETRY CNTR
  984  006150  005067  172526            CLR     RTCNT           ;CLEAR RETRY CNTR
  985  006154  032777  002000  172424    BIT     #2000,@SWR      ;SEE IF PRINT ERRORS
  986  006162  001003                    BNE     WTM4A           ;IF NOT: BR
  987  006164  012704  026315            MOV     #MSG64,R4
  988  006170  104000                    TTOUTT                  ;PRINT ORIGINAL ERROR TAG
  989  006172  004767  000044    WTM4A:  JSR     PC,WRTY         ;GO DO RETRY
  990  006176  005067  172510    WEX:    CLR     RTYFL           ;CLEAR RETRY FLAG
  991  006202  005067  172470            CLR     TMFLG           ;CLEAR TAPE MARK FLAG
  992  006206  005767  172446            TST     EOTREC          ;SEE IF EOT
  993  006212  100401                    BMI     WRW             ;IF SO: BR
  994  006214  000207                    RTS     PC              ;ELSE EXIT
  995  006216  017703  172364    WRW:    MOV     @SWR,R3
  996  006222  042703  177763            BIC     #177763,R3
  997  006226  022703  000014            CMP     #14,R3          ;SEE IF WRITE ONLY
  998  006232  001002                    BNE     WRWX            ;IF NOT: BR
  999  006234  000167  176044            JMP     REOT            ;ELSE REWIND
 1000  006240  000207    WRWX:   RTS     PC              ;EXIT
```

```
 1001                                   ;*************************
 1002                                   ;WRITE ERROR RETRY
 1003                                   ;
 1004                                   ;*************************
 1005
 1006  006242  012767  000001  172442  WRTY:   MOV     #1,RTYFL        ;SET RETRY FLAG
 1007  006250  004767  000364          WRTY0:  JSR     PC,WRTSB        ;GO SPACE REVERSE FOR REPEAT
 1008  006254  005767  172416                  TST     TMFLG           ;SEE IF TAPE MARK TIME
 1009  006260  001003                          BNE     WRTYTM          ;IF SO: BR
 1010  006262  004767  177142                  JSR     PC,WD           ;REWRITE RECORD
 1011  006266  000402                          BR      WRTYR           ;GO ON
 1012  006270  004767  177434          WRTYTM: JSR     PC,WTM          ;GO WRITE TAPE MARK AGAIN
 1013  006274  005767  172406          WRTYR:  TST     SERFL           ;REWRITE GOOD
 1014  006300  001024                          BNE     WRTY2           ;IF NOT: BR
 1015  006302  005267  172372                  INC     RPCNT           ;BUMP REPEAT COUNTER
 1016  006306  022767  000004  172364          CMP     #4,RPCNT        ;SEE IF FOUR GOOD REPEATS
 1017  006314  001355                          BNE     WRTY0           ;IF NOT: REPEAT
 1018  006316  032777  002000  172262          BIT     #2000,@SWR      ;SEE IF PRINT
 1019  006324  001011                          BNE     WRTY1           ;IF NOT: BR
 1020  006326  012704  026774                  MOV     #MSG105,R4
 1021  006332  104000                          TTOUTT                  ;PRINT RECOVERED MESSAGE
 1022  006334  012704  026337                  MOV     #MSG65,R4
 1023  006340  104000                          TTOUTT                  ;PRINT RETRY TAG
 1024  006342  016703  172334                  MOV     RTCNT,R3
 1025  006346  104002                          OCTPP                   ;PRINT RETRY NUMBER
 1026  006350  000207          WRTY1:  RTS     PC                      ;RESUME TESTING
 1027  006352  016703  172344          WRTY2:  MOV     ERSAV,R3        ;GET ER
 1028  006356  042703  102700                  BIC     #102700,R3      ;MASK RECOVERABLE BITS
 1029  006362  001413                          BEQ     WRTY2A          ;IF RECOVERABLE: BR
 1030  006364  004767  014272                  JSR     PC,PAPRT        ;PRINT HEADER
 1031  006370  012704  026601                  MOV     #MSG78,R4
 1032  006374  104000                          TTOUTT                  ;PRINT NON-RECOVERABLE MSG
 1033  006376  004767  002534                  JSR     PC,NRTP         ;PRINT ER
 1034  006402  012767  000001  172236          MOV     #1,TEMP3        ;SET FLAG
 1035  006410  000407                          BR      WRTY2B
 1036  006412  032777  002000  172166  WRTY2A: BIT     #2000,@SWR      ;SEE IF PRINT
 1037  006420  001025                          BNE     WRTY3           ;IF NOT: BR
 1038  006422  012704  027230                  MOV     #MSG110,R4
 1039  006426  104000                          TTOUTT                  ;PRINT BAD TAPE SUSPECT
 1040  006430  012704  026337          WRTY2B: MOV     #MSG65,R4
 1041  006434  104000                          TTOUTT                  ;PRINT RETRY TAG
 1042  006436  016703  172240                  MOV     RTCNT,R3
 1043  006442  104002                          OCTPP                   ;PRINT RETRY NUMBER
 1044  006444  012704  027252                  MOV     #MSG111,R4
 1045  006450  104000                          TTOUTT                  ;PRINT REPEAT TAG
 1046  006452  016703  172222                  MOV     RPCNT,R3
 1047  006456  104002                          OCTPP                   ;PRINT REPEAT NUMBER
 1048  006460  005767  172162                  TST     TEMP3           ;SEE IF DID NON-RECOVERABLE
 1049  006464  001403                          BEQ     WRTY3           ;IF NOT: BR
 1050  006466  005067  172154                  CLR     TEMP3           ;CLEAR FLAG
 1051  006472  000207                          RTS     PC              ;EXIT
 1052  006474  005767  172202          WRTY3:  TST     RTCNT           ;SEE IF FIRST RETRY
 1053  006500  001004                          BNE     WRTY3A          ;IF NOT: BR
 1054  006502  016704  172166                  MOV     UNP,R4
 1055  006506  005364  001070                  DEC     WTER1(R4)       ;DECREMENT WRITE ERROR CNTR
 1056  006512  016704  172156          WRTY3A: MOV     UNP,R4          ;GET UNIT NUMBER
```

```
1057  006516  016467  001030  172204          MOV     BTADDR(R4),BTPT ;GET ADDRESS OF UNIT BAD TAPE CNTR
1058  006524  017704  172200                  MOV     @BTPT,R4        ;GET COUNTER
1059  006530  005724                          TST     (R4)+           ;SET POINTER OFFSET
1060  006532  010477  172172                  MOV     R4,@BTPT
1061  006536  016703  172166                  MOV     BTPT,R3
1062  006542  060304                          ADD     R3,R4           ;SET ABSOLUTE POINTER
1063  006544  016714  172104                  MOV     BLCNTR,(R4)     ;SET BLOCK NUMBER
1064  006550  062704  000040                  ADD     #40,R4          ;ADD RCNT OFFSET
1065  006554  016714  171774                  MOV     RCNT,(R4)
1066  006560  160014                          SUB     R0,(R4)         ;SET RECORD NUMBER
1067  006562  005214                          INC     (R4)            ;CORRECT RECORD NUMBER
1068  006564  022777  000040  172136          CMP     #40,@BTPT       ;SEE IF TOO MANY BAD SPOTS
1069  006572  001002                          BNE     WRTY4           ;IF NOT: BR
1070  006574  000167  000234                  JMP     BTOV            ;ELSE GO TO BAD TAPE OVERFLOW
1071  006600  005267  172076          WRTY4:  INC     RTCNT           ;BUMP RETRY COUNTER
1072  006604  022767  000004  172070          CMP     #4,RTCNT        ;SEE IF DONE 4 RETRIES
1073  006612  001410                          BEQ     WRTY5           ;IF SO: BR
1074  006614  016704  172054                  MOV     UNP,R4
1075  006620  005264  001050                  INC     RTY1(R4)        ;BUMP RETRY COUNTER
1076  006624  005267  172102                  INC     ERTFL           ;SET ERASE FLAG
1077  006630  000167  177414                  JMP     WRTY0           ;DO NEXT RETRY
1078  006634  000167  000412          WRTY5:  JMP     BTUR            ;ELSE GO TO BAD TAPE UNRECOVERABLE
1079
1080                                          ;WRITE RETRY BACKSPACE-ERASE SUBROUTINE**********
1081
1082  006640  005067  172042          WRTSB:  CLR     SERFL           ;CLEAR FLAG
1083  006644  016767  171726  172014          MOV     TSTAL,STAL
1084  006652  004767  003304                  JSR     PC,STALL        ;DO TURN AROUND DELAY
1085  006656  012767  026350  171766          MOV     #MSG66,EMADDR   ;SET ERROR CODE
1086  006664  012777  177777  171624          MOV     #-1,@FC         ;SET TO BACKSPACE 1 RECORD
1087  006672  012777  033462  171614          MOV     #RDATA,@BA      ;SET BA
1088  006700  004767  003206                  JSR     PC,BKRT         ;GO BACKSPACE
1089  006704  005767  171776                  TST     SERFL           ;SEE IF ERROR
1090  006710  001406                          BEQ     WRTSB1          ;IF NOT: BR
1091  006712  012767  000002  172004  WRTSB0: MOV     #2,BTFLG        ;SET FLAG
1092  006720  022626                          CMP     (SP)+,(SP)+     ;RESET STACK
1093  006722  000167  175356                  JMP     REOT            ;GO REWIND AND REMOVE FROM TESTING
1094  006726  005767  172000          WRTSB1: TST     ERTFL           ;SEE IF SHOULD ERASE
1095  006732  001001                          BNE     WRTSB2          ;IF SO: BR
1096  006734  000207                          RTS     PC              ;RETURN
1097  006736  005067  171770          WRTSB2: CLR     ERTFL           ;CLEAR ERASE FLAG
1098  006742  005067  171732                  CLR     RPCNT           ;CLEAR REPEAT CNTR
1099  006746  005067  171734                  CLR     SERFL           ;CLEAR FLAG
1100  006752  012767  026363  171672          MOV     #MSG67,EMADDR   ;SET ERROR CODE
1101  006760  005077  171532                  CLR     @FC             ;CLEAR FRAME COUNT
1102  006764  012767  000024  171700          MOV     #24,MTC1        ;SET ERASE OP-CODE
1103  006772  012777  027454  171514          MOV     #WDATA,@BA      ;SET BA
1104  007000  012767  007012  171654          MOV     #WRTSB3,RTRN    ;SET RETURN ADDRESS
1105  007006  000167  012162                  JMP     TAPG            ;GO ERASE
1106  007012  012703  027454          WRTSB3: MOV     #WDATA,R3       ;SET EXPT BA
1107  007016  004767  010430                  JSR     PC,ER2          ;GO CHECK ERRORS
1108  007022  005767  171660                  TST     SERFL           ;SEE IF ERROR
1109  007026  001737                          BEQ     WRTSB1          ;IF NOT: BR
1110  007030  0001F~  177656                  JMP     WRTSB0
1111
1112                                          ;BAD TAPE OVERFLOW SUBROUTINE**********
```

```
1113
1114  007034  005067  171652          BTOV:   CLR     RTYFL           ;CLEAR RETRY FLAG
1115  007040  012767  000001  171656          MOV     #1,BTFLG        ;SET BAD TAPE OVERFLOW FLAG
1116  007046  000167  175232                  JMP     REOT            ;GO REWIND AND REMOVE FROM TESTING
1117  007052  016701  171652          BTOV0:  MOV     BTPT,R1         ;SET TABLE POINTER
1118  007056  005721                          TST     (R1)+
1119  007060  005000                          CLR     R0
1120  007062  010003                  BTOV1:  MOV     R0,R3
1121  007064  000241                          CLC
1122  007066  006003                          ROR     R3              ;R3=R3/2 FOR CORRECT NUMBER
1123  007070  104002                          OCTPP                   ;PRINT ENTRY NUMBER
1124  007072  012704  025034                  MOV     #MSG13,R4
1125  007076  105724                          TSTB    (R4)+           ;SKIP CR/LF
1126  007100  104000                          TTOUTT                  ;PRINT BLOCK NUMBER TAG
1127  007102  011103                          MOV     (R1),R3
1128  007104  104002                          OCTPP                   ;PRINT BLOCK NUMBER
1129  007106  012704  025042                  MOV     #MSG14,R4
1130  007112  104000                          TTOUTT                  ;PRINT RECORD NUMBER TAG
1131  007114  062701  000040                  ADD     #40,R1          ;SET POINTER OFFSET FOR RECOED NUMBER
1132  007120  012103                          MOV     (R1)+,R3
1133  007122  104002                          OCTPP                   ;PRINT RECORD NUMBER
1134  007124  162701  000040                  SUB     #40,R1          ;RESET POINTER FOR BLOCK NUMBER
1135  007130  005720                          TST     (R0)+
1136  007132  020077  171572                  CMP     R0,@BTPT        ;SEE IF DONE
1137  007136  001404                          BEQ     BTOV2           ;IF SO: BR
1138  007140  012704  025365                  MOV     #MSG28,R4
1139  007144  104000                          TTOUTT                  ;DO CR/LF
1140  007146  000745                          BR      BTOV1           ;CONTINUE
1141  007150  005767  171552          BTOV2:  TST     BTSTF           ;SEE IF STAT ONLY PRINT
1142  007154  001007                          BNE     BTOVX           ;IF SO: BR
1143  007156  012703  000041                  MOV     #41,R3          ;SET SIZE OF TABLE
1144  007162  016704  171542                  MOV     BTPT,R4         ;SET POINTER
1145  007166  005024                  BTOV3:  CLR     (R4)+           ;CLEAR TABLE
1146  007170  005303                          DEC     R3              ;SEE IF DONE.
1147  007172  001375                          BNE     BTOV3           ;IF NOT: BR
1148  007174  000207                  BTOVX:  RTS     PC              ;RETURN
1149
```

```
 1150
 1151                                        ;BAD TAPE STATISTIC PRINT********
 1152
 1153  007176  012704  025365        BTPRT:  MOV     #MSG28,R4
 1154  007202  104000                        TTOUTT                  ;DO CR/LF
 1155  007204  016704  171464                MOV     UNP,R4
 1156  007210  016467  001030  171512        MOV     BTADDR(R4),BTPT ;SET TABLE POINTER
 1157  007216  017703  171506                MOV     @BTPT,R3
 1158  007222  000241                        CLC
 1159  007224  006003                        ROR     R3              ;CORRECT NUMBER
 1160  007226  104002                        OCTPP                   ;PRINT NUMBER OF BAD SPOTS
 1161  007230  012704  027264                MOV     #MSG112,R4
 1162  007234  104000                        TTOUTT                  ;PRINT BAD TAPE TAG
 1163  007236  005777  171466                TST     @BTPT           ;SEE IF ANY BAD SPOTS
 1164  007242  001001                        BNE     BTPRT1          ;IF SO: BR
 1165  007244  000207                        RTS     PC              ;ELSE RETURN
 1166  007246  000167  177600        BTPRT1: JMP     BTOVD           ;PRINT STATS
 1167
 1168                                        ;BAD TAPE UNRECOVERABLE SUBROUTINE**********
 1169
 1170  007252  004767  013404        BTUR:   JSR     PC,PAPRT        ;PRINT HEADER
 1171  007256  012704  027077                MOV     #MSG107,R4
 1172  007262  104000                        TTOUTT                  ;PRINT UNRECOVERABLE BAD SPOT MSG
 1173  007264  000207                        RTS     PC              ;RESUME TESTING
 1174
```

```
 1175                                   ;*****************************************************
 1176                                   ;READ SEQUENCER:
 1177                                   ;
 1178                                   ;THIS ROUTINE IS USED TO DETERMINE THE SEQUENCE
 1179                                   ;IN WHICH READ TAPE OPERATIONS ARE TO BE PERFORMED.
 1180                                   ;THIS IS NECESSARY WHEN THE UNIT BEING TESTED IS
 1181                                   ;CAPABLE OF READING DATA IN BOTH THE FORWARD AND
 1182                                   ;REVERSE DIRECTIONS.  CONSOLE SWITCHES ONE (1), TWO (2),
 1183                                   ;AND THREE (3) ARE USED TO DETERMINE THE READ SEQUENCE.
 1184                                   ;CONSOLE SWITCH ONE (1) DETERMINES WHETHER TO READ
 1185                                   ;THE BLOCK OF DATA FORWARD FIRST OR REVERSE FIRST.
 1186                                   ;SWITCH TWO (2) DISALLOWS READING IN THE REVERSE
 1187                                   ;DIRECTION AND SWITCH THREE (3) DISALLOWS READING IN
 1188                                   ;THE FORWARD DIRECTION.
 1189                                   ;*****************************************************
 1190
 1191  007266  012767  000002  171266  RSEQ:   MOV     #2,RDCMD
 1192  007274  017704  171306          MOV     @SWR,R4         ;READ SWITCHES
 1193  007300  042704  177763          BIC     #177763,R4      ;MASK READ BITS
 1194  007304  005704                  TST     R4              ;SEE IF BOTH READS
 1195  007306  001004                  BNE     RSR             ;IF NOT: BR
 1196  007310  032777  000002  171270  BIT     #2,@SWR         ;SEE IF READ REVERSE FIRST
 1197  007316  001053                  BNE     RSFR            ;IF NOT: BR
 1198  007320  032777  000004  171260  RSR:    BIT     #4,@SWR         ;SEE IF SHOULD READ REVERSE
 1199  007326  001005                  BNE     RSF             ;IF NOT: BR
 1200  007330  012767  010000  171224          MOV     #10000,RDCMD    ;LOAD READ REVERSE COMMAND
 1201  007336  004767  000252          JSR     PC,READ         ;GO READ REVERSE
 1202  007342  032777  000010  171236  RSF:    BIT     #10,@SWR        ;SEE IF SHOULD READ FORWARD
 1203  007350  001033                  BNE     RSEX            ;IF NOT: BR
 1204  007352  032767  010000  171202          BIT     #10000,RDCMD    ;SEE IF HAVE READ REVERSE
 1205  007360  001407                  BEQ     RSF0            ;IF NOT: BR
 1206  007362  016767  171210  171276          MOV     TSTAL,STAL
 1207  007370  004767  002566          JSR     PC,STALL        ;DO READ STALL
 1208  007374  000167  000014          JMP     RSF1
 1209  007400  032777  000001  171200  RSF0:   BIT     #1,@SWR         ;SEE IF WRITE
 1210  007406  001002                  BNE     RSF1            ;IF NOT: BR
 1211  007410  004767  002320          JSR     PC,BKSP         ;GO BACKSPACE
 1212  007414  012767  000002  171140  RSF1:   MOV     #2,RDCMD        ;LOAD READ FORWARD COMMAND
 1213  007422  004767  000166          JSR     PC,READ         ;GO READ
 1214  007426  005767  171226          TST     EOTREC          ;SEE IF AT END OF TAPE
 1215  007432  100002                  BPL     RSEX            ;IF NOT: BR
 1216  007434  000167  174644          JMP     REOT            ;ELSE GO TO REWIND
 1217  007440  005067  171214  RSEX:   CLR     EOTREC          ;CLEAR EOT FLAG
 1218  007444  000207                  RTS     PC              ;EXIT
 1219  007446  012767  010000  171106  RSFR:   MOV     #10000,RDCMD
 1220  007454  032777  000010  171124          BIT     #10,@SWR        ;SEE IF SHOULD READ FORWARD
 1221  007462  001013                  BNE     RSFR1           ;IF NOT: BR
 1222  007464  032777  000001  171114          BIT     #1,@SWR         ;SEE IF WRITE
 1223  007472  001002                  BNE     RSFR0           ;IF NOT: BR
 1224  007474  004767  002234          JSR     PC,BKSP         ;GO BACKSPACE TO START
 1225  007500  012767  000002  171054  RSFR0:  MOV     #2,RDCMD        ;LOAD READ FORWARD COMMAND
 1226  007506  004767  000102          JSR     PC,READ         ;GO READ FORWARD
 1227  007512  032777  000004  171066  RSFR1:  BIT     #4,@SWR         ;SEE IF SHOULD READ REVERSE
 1228  007520  001347                  BNE     RSEX            ;IF NOT: BR
 1229  007522  032767  010000  171032          BIT     #10000,RDCMD
 1230  007530  001005                  BNE     RSFR2           ;IF READ REVERSE: BR
```

```
1231  007532  016767  171040  171126          MOV     TSTAL,STAL      ;DO READ STALL
1232  007540  004767  002416                  JSR     PC,STALL
1233  007544  012767  010000  171010  RSFR2:  MOV     #10000,RDCMD    ;LOAD READ REVERSE
1234  007552  004767  000036                  JSR     PC,READ         ;GO READ REVERSE
1235  007556  005767  171076                  TST     EOTREC          ;SEE IF AT END OF TAPE
1236  007562  001411                          BEQ     RSFRX           ;IF NOT: BR
1237  007564  166767  170764  171066          SUB     RCNT,EOTREC
1238  007572  005467  171062                  NEG     EOTREC          ;SET TO PROPER RECORD NUMBER
1239  007576  005267  171056                  INC     EOTREC
1240  007602  000167  174476                  JMP     REOT            ;ELSE GO TO REWIND
1241  007606  005067  171046          RSFRX:  CLR     EOTREC          ;CLEAR EOT FLAG
1242  007612  000207                          RTS     PC              ;EXIT
1243
```

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

007614  016700  170734          READ:   MOV     RCNT,R0         ;LOAD REC CNTR
007620  005767  171034                  TST     EOTREC          ;SEE IF EOT
007624  100013                          BPL     RDA             ;IF NOT: BR
007626  032767  010000  170726          BIT     #10000,RDCMD    ;SEE IF READ FORWARD
007634  001407                          BEQ     RDA             ;IF SO: BR
007636  042767  100000  171014          BIC     #100000,EOTREC  ;CLEAR FLAG
007644  016703  171010                  MOV     EOTREC,R3       ;GET MODIFIED RECORD COUNT
007650  160300                          SUB     R3,R0           ;SET RECORD AT
007652  005200                          INC     R0              ;SET TO PROPER NUMBER OF RECORDS
007654  012767  024747  170770  RDA:    MOV     #MSG6,EMADDR    ;SET ERROR MSG ADDRESS
007662  005067  171010                  CLR     TMFLG
007666  032767  010000  170666          BIT     #10000,RDCMD
007674  001406                          BEQ     RD0             ;IF READ FORWARD: BR
007676  005767  170662                  TST     TMEX            ;SEE IF TM
007702  001403                          BEQ     RD0             ;IF NOT: BR
007704  005267  170766                  INC     TMFLG           ;SET TM FLAG
007710  005200                          INC     R0
007712  016777  170640  170576  RD0:    MOV     FMCNT,@FC       ;LOAD CHAR CNTR
007720  012777  033462  170566          MOV     #RDATA,@BA      ;LOAD DATA ADDR
007726  032767  010000  170626          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
007734  001417                          BEQ     RD1A            ;IF NOT: BR
007736  016703  170614                  MOV     FMCNT,R3
007742  005103                          COM     R3
007744  032767  000020  170600          BIT     #20,UDES        ;SEE IF CORE DUMP
007752  001402                          BEQ     RD1             ;IF NOT: BR
007754  000241                          CLC
007756  006003                          ROR     R3              ;R3 = FC/2
007760  060377  170530          RD1:    ADD     R3,@BA          ;SET REVERSE BUS ADDRESS
007764  012767  000076  170700          MOV     #76,MTC1        ;SET READ REVERSE
007772  000403                          BR      RD1B
```

```
 1300  007774  012767  000070  170670  RD1A:   MOV     #70,MTC1        ;SET READ FORWARD
 1301  010002  012767  010014  170652  RD1B:   MOV     #RD2,RTRN       ;SET INTERRUPT RETURN ADDRESS
 1302  010010  000167  011160          RD1D:   JMP     TAPG            ;GO EXECUTE TAPE COMMAND
 1303  010014  032767  010000  170540  RD2:    BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 1304  010022  001024                          BNE     RD3             ;IF SO: BR
 1305  010024  032777  000020  170470          BIT     #20,@DS
 1306  010032  001404                          BEQ     RD2B            ;AWAIT SWDN
 1307  010034  032777  000020  170460  RD2A:   BIT     #20,@DS
 1308  010042  001374                          BNE     RD2A            ;AWAIT TUR
 1309  010044  032777  002000  170450  RD2B:   BIT     #2000,@DS       ;SEE IF EOT
 1310  010052  001410                          BEQ     RD3             ;IF NOT: BR
 1311  010054  005767  170616                  TST     TMFLG           ;SEE IF TM
 1312  010060  001005                          BNE     RD3             ;IF SO: BR
 1313  010062  010067  170572                  MOV     R0,EOTREC
 1314  010066  052767  100000  170564          BIS     #100000,EOTREC  ;SET EOT FLAG
 1315  010074  032777  000002  170420  RD3:    BIT     #2,@DS          ;SEE IF AT LOAD POINT
 1316  010102  001410                          BEQ     RD4             ;IF NOT: BR
 1317  010104  004767  012552                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
 1318  010110  012704  025146                  MOV     #MSG22,R4
 1319  010114  104000                          TTOUTT                  ;PRINT BOT ERROR
 1320  010116  104006                          STOPP
 1321  010120  000167  172734                  JMP     STARTA          ;RESTART
 1322  010124  032777  004000  170454  RD4:    BIT     #4000,@SWR      ;SEE IF SHOULD CHECK ERRORS
 1323  010132  001121                          BNE     RD5             ;IF NOT: BR
 1324  010134  005767  170536                  TST     TMFLG
 1325  010140  001472                          BEQ     RD4B            ;IF NO TM EXPT: BR
 1326  010142  032777  000004  170352          BIT     #4,@DS
 1327  010150  001024                          BNE     RD4A            ;IF TM RECVD: BR
 1328  010152  012767  033462  010730          MOV     #RDATA,CADER    ;SAVE EXPT BUS ADDRESS
 1329  010160  012767  000002  010730          MOV     #2,DRVER        ;SET TM STATUS ERROR FLAG
 1330  010166  004767  010020                  JSR     PC,ERPT         ;GO PRINT TM ERROR
 1331  010172  016704  170476                  MOV     UNP,R4
 1332  010176  032767  010000  170356          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 1333  010204  001403                          BEQ     1$              ;IF NOT: BR
 1334  010206  005264  001150                  INC     RDERR1(R4)      ;BUMP READ REVERSE ERROR
 1335  010212  000502                          BR      RD6
 1336  010214  005264  001110          1$:     INC     RDER1(R4)       ;BUMP READ FORWARD ERROR
 1337  010220  000477                          BR      RD6
 1338  010222  012703  033462          RD4A:   MOV     #RDATA,R3
 1339  010226  032767  010000  170326          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 1340  010234  001007                          BNE     RD4A0           ;IF SO: BR
 1341  010236  032767  002000  170306          BIT     #2000,UDES      ;SEE IF IN PE
 1342  010244  001025                          BNE     RD4A2           ;IF SO: BR
 1343  010246  062703  000002                  ADD     #2,R3
 1344  010252  000422                          BR      RD4A2
 1345  010254  016704  170276          RD4A0:  MOV     FMCNT,R4
 1346  010260  005104                          COM     R4
 1347  010262  032767  000020  170262          BIT     #20,UDES        ;SEE IF CORE DUMP
 1348  010270  001402                          BEQ     RD4A1           ;IF NOT: BR
 1349  010272  000241                          CLC
 1350  010274  006004                          ROR     R4              ;SET TO FC/2
 1351  010276  060403          RD4A1:  ADD     R4,R3           ;SET EXPT BUS ADDRESS
 1352  010300  042703  000001                  BIC     #1,R3           ;MAKE EXPT ADDRESS EVEN
 1353  010304  032767  002000  170240          BIT     #2000,UDES      ;SEE IF IN PE
 1354  010312  001002                          BNE     RD4A2           ;IF SO: BR
 1355  010314  162703  000002                  SUB     #2,R3
```

```
1356  010320  004767  007126          RD4A2:  JSR     PC,ER2
1357  010324  000402                  RD4C    BR      RD4C
1358  010326  004767  007022          RD4B:   JSR     PC,ERCHK        ;GO CHECK ERRORS
1359  010332  005767  170350          RD4C:   TST     SERFL
1360  010336  001417                          BEQ     RD5             ;IF NO ERROR: BR
1361  010340  016704  170330                  MOV     UNP,R4
1362  010344  032767  010000  170210          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
1363  010352  001003                          BNE     RD4D            ;IF SO: BR
1364  010354  005264  001110                  INC     RDER1(R4)       ;BUMP READ FORWARD ERROR
1365  010360  000402                          BR      RD4E
1366  010362  005264  001150          RD4D:   INC     RDERR1(R4)      ;BUMP READ REVERSE ERROR
1367  010366  004767  000176          RD4E:   JSR     PC,RDRTY        ;GO RETRY
1368  010372  005067  170314                  CLR     RTYFL           ;CLEAR RETRY FLAG
1369  010376  032777  020000  170202  RD5:    BIT     #20000,@SWR     ;SEE IF SHOULD DO DATA CHECK
1370  010404  001005                          BNE     RD6             ;IF NOT; BR
1371  010406  005767  170264                  TST     TMFLG
1372  010412  001002                          BNE     RD6
1373  010414  004767  005052                  JSR     PC,DCHK         ;GO CHECK DATA
1374  010420  005067  170262          RD6:    CLR     SERFL           ;CLEAR STATUS ERROR FLAG
1375  010424  004767  003610                  JSR     PC,DS3          ;CLEAR BUFFER
1376  010430  032777  000040  170150          BIT     #40,@SWR        ;SEE IF SHOULD YOZZLE
1377  010436  001402                          BEQ     RD7             ;IF NOT: BR
1378  010440  004767  000506                  JSR     PC,YOZ          ;ELSE GO YOZZLE
1379  010444  016767  170122  170214  RD7:    MOV     RSTAL,STAL      ;SET DELAY
1380  010452  004767  001504                  JSR     PC,STALL        ;STALL
1381  010456  032767  010000  170076          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
1382  010464  001403                          BEQ     RD7A            ;IF NOT: BR
1383  010466  005067  170204                  CLR     TMFLG           ;CLEAR TAPE MARK FLAG
1384  010472  000405                          BR      RD10
1385  010474  005767  170160          RD7A:   TST     EOTREC          ;SEE IF EOT FOUND
1386  010500  100002                          BPL     RD10            ;IF NOT: BR
1387  010502  012700  000001                  MOV     #1,R0           ;SET TO EOT
1388  010506  005300                  RD10:   DEC     R0
1389  010510  001402                          BEQ     RD11            ;IF DONE ALL: BR
1390  010512  000167  177174                  JMP     RD0
1391  010516  032767  010000  170036  RD11:   BIT     #10000,RDCMD    ;SEE IF READ REVERSE
1392  010524  001016                          BNE     RDEX            ;IF SO: BR
1393  010526  005767  170126                  TST     EOTREC          ;SEE IF FOUND EOT
1394  010532  100413                          BMI     RDEX            ;IF SO: BR
1395  010534  005767  170024                  TST     TMEX            ;SEE IF TM EXPECTED
1396  010540  001410                          BEQ     RDEX            ;IF NOT: BR
1397  010542  005767  170130                  TST     TMFLG           ;SEE IF TM FOUND
1398  010546  001005                          BNE     RDEX            ;IF SO: BR
1399  010550  005267  170122                  INC     TMFLG           ;ELSE SET FLAG
1400  010554  005200                          INC     R0              ;SET RECORD COUNT TO ONE
1401  010556  000167  177130                  JMP     RD0             ;GO READ TM
1402  010562  005067  170110          RDEX:   CLR     TMFLG
1403  010566  000207                  RDX:    RTS     PC              ;EXIT
```

```
1404                                    ;*******************************************************
1405                                    ;READ ERROR RETRY SUBROUTINE:
1406                                    ;
1407                                    ;THIS SUBROUTINE WILL RETRY ALL DATA RELATED
1408                                    ;READ ERRORS UP TO EIGHT (8) TIMES.  IF ALL
1409                                    ;FOUR RETRIES ARE BAD, IT IS CONSIDERED
1410                                    ;A HARD ERROR.  IF ANY ARE GOOD, IT IS A
1411                                    ;SOFT ERROR.  RETRIES MAY BE INHIBITED
1412                                    ;VIA SWITCH FOUR (SW4=0: INHIBIT RETRIES)
1413                                    ;*******************************************************
1414
1415  010570  032777  000020  170010  RDRTY:  BIT     #20,@SWR        ;SEE IF RETRY INHIBITED
1416  010576  001001                   BNE     RDRT0           ;IF NOT: BR
1417  010600  000207                   RTS     PC              ;ELSE RETURN
1418  010602  016703  170114          RDRT0:  MOV     ERSAV,R3
1419  010606  042703  102700                  BIC     #102700,R3      ;MARK NON-RECOVERABLE ERROR BITS
1420  010612  001410                   BEQ     RDRT1           ;IF NOT: BR
1421  010614  004767  012042                  JSR     PC,PAPRT        ;PRINT HEADER
1422  010620  012704  026642                  MOV     #MSG79,R4
1423  010624  104000                           TTOUTT                  ;PRINT NON-RECOVERABLE MESSAGE
1424  010626  004767  000304                  JSR     PC,NRTP         ;PRINT ER FOR NON-RETRYABLE ERROR
1425  010632  000207          RDRT1A: RTS     PC              ;RETURN
1426  010634  032777  002000  167744  RDRT1:  BIT     #2000,@SWR      ;SEE IF PRINT INHIBITED
1427  010642  001003                   BNE     RDRT1B          ;IF SO: BR
1428  010644  012704  026315                  MOV     #MSG64,R4
1429  010650  104000                           TTOUTT                  ;PRINT ORIGINAL ERROR TAG
1430  010652  005067  170024  RDRT1B: CLR     RTCNT           ;CLEAR RETRY COUNTER
1431  010656  005067  170024  RDRTG:  CLR     SERFL           ;CLEAR STATUS ERROR FLAG
1432  010662  012767  000002  170022          MOV     #2,RTYFL        ;SET READ RETRY FLAG
1433  010670  004767  000256                  JSR     PC,YOZ          ;GO TO YOZZLE TO RETRY READ
1434  010674  005767  170006                  TST     SERFL           ;SEE IF RETRY ERROR
1435  010700  001031                   BNE     RDRT5           ;IF SO: BR
1436  010702  032777  002000  167676          BIT     #2000,@SWR
1437  010710  001011                   BNE     RDRT2
1438  010712  012704  026774                  MOV     #MSG105,R4
1439  010716  104000                           TTOUTT                  ;PRINT RECOVERED MESSAGE
1440  010720  012704  026337                  MOV     #MSG65,R4
1441  010724  104000                           TTOUTT                  ;PRINT RETRY TAG
1442  010726  016703  167750                  MOV     RTCNT,R3
1443  010732  104002                           OCTPP                   ;PRINT RETRY NUMBER
1444  010734  016704  167734  RDRT2:  MOV     UNP,R4
1445  010740  032767  010000  167614          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
1446  010746  001003                   BNE     RDRT3           ;IF SO: BR
1447  010750  005264  002670                  INC     RFSOFT(R4)      ;ELSO BUMP FORWARD SOFT ERROR COUNTER
1448  010754  000402                   BR      RDRT4
1449  010756  005264  002710  RDRT3:  INC     RRSOFT(R4)      ;BUMP ERRORS SOFT CNTR
1450  010762  000207          RDRT4:  RTS     PC              ;RETURN
1451  010764  016703  167732  RDRT5:  MOV     ERSAV,R3        ;GET ER
1452  010770  042703  102700                  BIC     #102700,R3      ;MASK RECOVERABLE BITS
1453  010774  001413                   BEQ     RDRT5A          ;IF RECOVERABLE: BR
1454  010776  004767  011660                  JSR     PC,PAPRT        ;PRINT HEADER
1455  011002  012704  026642                  MOV     #MSG79,R4
1456  011006  104000                           TTOUTT                  ;PRINT NON-RECOVERABLE MSG
1457  011010  004767  000122                  JSR     PC,NRTP         ;PRINT ER
1458  011014  012767  000001  167624          MOV     #1,TEMP3        ;SET FLAG
1459  011022  000404                   BR      RDRT5B
```

```
1460  011024  032777  002000  167554  RDRT5A:  BIT     #2000,@SWR      ;SEE IF PRINT INHIBITED
1461  011032  001014                   BNE     RDRT6           ;IF SO: BR
1462  011034  012704  026337          RDRT5B:  MOV     #MSG65,R4
1463  011040  104000                           TTOUTT                  ;PRINT RETRY TAG
1464  011042  016703  167634                   MOV     RTCNT,R3
1465  011046  104002                           OCTPP                   ;PRINT RETRY NUMBER
1466  011050  005767  167572                   TST     TEMP3           ;SEE IF DID NON-RECOVERABLE
1467  011054  001403                           BEQ     RDRT6           ;IF NOT: BR
1468  011056  005067  167564                   CLR     TEMP3           ;CLEAR FLAG
1469  011062  000207                           RTS     PC              ;EXIT
1470  011064  005267  167612          RDRT6:   INC     RTCNT
1471  011070  026767  167606  167504         CMP     RTCNT,RETRY     ;SEE IF DONE 8 RETRIES
1472  011076  001267                         BNE     RDRTG           ;IF NOT: BR
1473  011100  012704  027327                MOV     #MSG115,R4
1474  011104  104000                         TTOUTT                  ;PRINT HARD ERROR MESSAGE
1475  011106  016704  167562                MOV     UNP,R4
1476  011112  032767  010000  167442       BIT     #10000,RDCMD    ;SEE IF READ REVERSE
1477  011120  001003                       BNE     RDRT7           ;IF SO: BR
1478  011122  005264  002730              INC     RFHARD(R4)      ;BUMP FORWARD HARD ERROR CNTR
1479  011126  000402                      BR      RDRTX
1480  011130  005264  002750      RDRT7:  INC     RRHARD(R4)      ;BUMP REVERSE HARD ERROR CNTR
1481  011134  000207               RDRTX:  RTS     PC              ;RETURN
1482
1483  011136  016703  167560      NRTP:   MOV     ERSAV,R3        ;GET ER REGISTER
1484  011142  104002                       OCTPP                   ;PRINT ER
1485  011144  004767  007764               JSR     PC,FRPRT        ;PRINT F OR R
1486  011150  000207                       RTS     PC              ;RETURN
```

```
 1487                                   ;*******************************************************
 1488                                   ;YOZZLE SUBROUTINE:
 1489                                   ;
 1490                                   ;THIS SUBROUTINE, ENTERED VIA SWITCH FIVE (5), IS USED TO PERFORM
 1491                                   ;A CONTINUOUS READ AND SPACE OVER OF THE CURRENT RECORD ON TAPE.
 1492                                   ;FULL STATUS AND DATA CHECKING MAY BE PERFORMED
 1493                                   ;OR NOT VIA CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13).
 1494                                   ;A SOFTWARE DELAY IS PERFORMED BETWEEN EACH READ
 1495                                   ;AND SPACE OPERATION AND MAY BE VARIED BY TYPING
 1496                                   ;CNTRL C ON THE TTY AND ENTERING A VALUE IN RESPONSE
 1497                                   ;TO THE PRINTED REQUEST.
 1498                                   ;*******************************************************
 1499
 1500  011152  012777  000001  167430  YOZ:    MOV     #1,@TKS         ;SET TTY ENABLE
 1501  011160  016767  167414  167500          MOV     YSTAL,STAL
 1502  011166  004767  000770                  JSR     PC,STALL        ;DO YOZZLE STALL
 1503  011172  012777  177777  167316  YOZO:   MOV     #-1,@FC         ;SET TO 1 RECORD SPACING
 1504  011200  032767  010000  167354          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 1505  011206  001404                          BEQ     YOZA            ;IF NOT: BR
 1506  011210  112767  000030  167454          MOVB    #30,MTC1        ;SET TO SPACE FORWARD
 1507  011216  000403                          BR      YOZB
 1508  011220  112767  000032  167444  YOZA:   MOVB    #32,MTC1        ;SET TO SPACE REVERSE
 1509  011226  012767  011246  167426  YOZB:   MOV     #YOZC,RTRN      ;SET RETURN ADDRESS
 1510  011234  012767  177775  167424          MOV     #177775,STAL    ;SET TIME MULTIPLIER
 1511  011242  000167  007726                  JMP     TAPG            ;GO YOZZLE
 1512  011246  005767  167424          YOZC:   TST     TMFLG           ;SEE IF TM
 1513  011252  001404                          BEQ     1$              ;IF NOT: BR
 1514  011254  012767  040000  167404          MOV     #40000,STAL     ;SET TM STALL
 1515  011262  000403                          BR      2$
 1516  011264  016767  167310  167374  1$:     MOV     YSTAL,STAL
 1517  011272  004767  000664          2$:     JSR     PC,STALL        ;DO YOZZLE STALL
 1518  011276  012777  033462  167210          MOV     #RDATA,@BA      ;SET BUS ADDRESS
 1519  011304  032767  010000  167250          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 1520  011312  001417                          BEQ     YOZC1           ;IF NOT: BR
 1521  011314  016703  167236                  MOV     FMCNT,R3
 1522  011320  005103                          COM     R3
 1523  011322  032767  000020  167222          BIT     #20,UDES        ;SEE IF CORE DUMP
 1524  011330  001402                          BEQ     YOZCO           ;IF NOT: BR
 1525  011332  000241                          CLC
 1526  011334  006003                          ROR     R3              ;R3 = FC/2
 1527  011336  060377  167152          YOZCO:  ADD     R3,@BA          ;SET REVERSE BUS ADDRESS
 1528  011342  012767  000076  167322          MOV     #76,MTC1        ;SET READ REVERSE
 1529  011350  000403                          BR      YOZC2
 1530  011352  012767  000070  167312  YOZC1:  MOV     #70,MTC1        ;SET READ FORWARD
 1531  011360  016777  167172  167130  YOZC2:  MOV     FMCNT,@FC       ;SET CHARACTER COUNT
 1532  011366  012767  011400  167266          MOV     #YOZD,RTRN      ;SET RETURN ADDRESS
 1533  011374  000167  007574                  JMP     TAPG            ;GO READ
 1534  011400  032777  004000  167200  YOZD:   BIT     #4000,@SWR      ;SEE IF SHOULD CHECK ERRORS
 1535  011406  001051                          BNE     YOZE            ;IF NOT: BR
 1536  011410  005767  167262                  TST     TMFLG           ;SEE IF TAPE MARK TIME
 1537  011414  001444                          BEQ     YOZD1           ;IF NOT: BR
 1538  011416  032767  010000  167136          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 1539  011424  001426                          BEQ     YOZDO           ;IF NOT: BR
 1540  011426  012703  033462                  MOV     #RDATA,R3
 1541  011432  016704  167120                  MOV     FMCNT,R4
 1542  011436  005104                          COM     R4
```

```
1543  011440  032767  000020  167104          BIT     #20,UDES        ;SEE IF CORE DUMP
1544  011446  001402                          BEQ     YOZD4           ;IF NOT: BR
1545  011450  000241                          CLC
1546  011452  006004                          ROR     R4              ;SET TO FC/2
1547  011454  060403                  YOZD4:  ADD     R4,R3           ;SET EXPT BUS ADDRESS
1548  011456  042703  000001                BIC     #1,R3           ;MAKE EXPT ADDRESS EVEN
1549  011462  032767  002000  167062        BIT     #2000,UDES      ;SEE IF PE
1550  011470  001001                          BNE     YOZD2           ;IF SO: BR
1551  011472  005743                          TST     -(R3)           ;SET EXPT BA
1552  011474  004767  005752          YOZD2:  JSR     PC,ER2          ;GO CHECK ERRORS
1553  011500  000430                          BR      YOZF
1554  011502  012703  033462          YOZD0:  MOV     #RDATA,R3
1555  011506  032767  002000  167036        BIT     #2000,UDES      ;SEE IF PE
1556  011514  001001                          BNE     YOZD3           ;IF SO: BR
1557  011516  005723                          TST     (R3)+           ;SET EXPT BA
1558  011520  004767  005726          YOZD3:  JSR     PC,ER2          ;GO CHECK ERRORS
1559  011524  000416                          BR      YOZF
1560  011526  004767  005622          YOZD1:  JSR     PC,ERCHK        ;ELSE GO CHECK ERRORS
1561  011532  032777  020000  167046  YOZE:   BIT     #20000,@SWR     ;SEE IF SHOULD CHECK DATA
1562  011540  001010                          BNE     YOZF            ;IF NOT: BR
1563  011542  005767  167130                  TST     TMFLG           ;SEE IF TAPE MARK
1564  011546  001005                          BNE     YOZF            ;IF SO: BR
1565  011550  005767  167136                  TST     RTYFL           ;SEE IF RETRY
1566  011554  001004                          BNE     YOZF0           ;IF SO: BR
1567  011556  004767  003710                  JSR     PC,DCHK         ;ELSE GO CHECK DATA
1568  011562  004767  002452          YOZF:   JSR     PC,DS3          ;GO CLEAR DATA AREA
1569  011566  105777  167016          YOZF0:  TSTB    @TKS            ;SEE IF HAVE NEW STALL VALUE
1570  011572  100032                          BPL     YOZG            ;IF NOT: BR
1571  011574  122777  000203  167010        CMPB    #203,@TKB       ;SEE IF CONT C
1572  011602  001026                          BNE     YOZG            ;IF NOT: BR
1573  011604  012704  026017                  MOV     #MSG44,R4
1574  011610  104000                          TTOUTT                  ;PRINT YSTALL REQUEST
1575  011612  016703  166762                  MOV     YSTAL,R3
1576  011616  104002                          OCTPP                   ;PRINT PRESENT STALL
1577  011620  010067  167022                  MOV     R0,TEMP3        ;SAVE R0(REC CNT)
1578  011624  012705  000600                  MOV     #YSTAL,R5       ;SET ADDRESS OF YSTL
1579  011630  012701  000006                  MOV     #6,R1           ;SET NUMBER OF CHAR TO INPUT
1580  011634  012702  177777                  MOV     #-1,R2          ;SET MAXIMUM LIMIT
1581  011640  012703  002000                  MOV     #2000,R3        ;SET MINIMUM LIMIT
1582  011644  004767  011430                  JSR     PC,TTR          ;GO GET VALUE
1583  011650  016700  166772                  MOV     TEMP3,R0        ;RESTORE R0(REC CNTR)
1584  011654  000167  177272                  JMP     YOZ             ;RESTART YOZZLE
1585  011660  122777  000207  166724  YOZG:   CMPB    #207,@TKB       ;CHECK FOR CNTL G
1586  011666  001010                          BNE     YOZI
1587  011670  022767  000176  166710        CMP     #SWREG,SWR      ;IS SWREG SELECTED
1588  011676  001004                          BNE     YOZI
1589  011700  005077  166706                  CLR     @TKB            ;CLEAR CNTL G OUT OF BUFFER
1590  011704  004767  012512                  JSR     PC,CNTG         ;GO CHANGE SWREG
1591  011710  032777  000040  166670  YOZI:   BIT     #40,@SWR        ;SEE IF SHOULD CONTINUE YOZZLE
1592  011716  001402                          BEQ     YOZH            ;IF NOT: BR
1593  011720  000167  177246                  JMP     YOZO
1594  011724  012777  000100  166656  YOZH:   MOV     #100,@TKS       ;SET TTY INTERRUPT ENABLE
1595  011732  000207                          RTS     PC              ;EXIT
1596
```

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

011734  016767  166636  166724  BKSP:   MOV     TSTAL,STAL
011742  004767  000214                  JSR     PC,STALL        ;DO TURN AROUND STALL
011746  012767  024777  166676          MOV     #MSG10,EMADDR
011754  012777  033462  166532          MOV     #RDATA,@BA
011762  005767  166576                  TST     TMEX            ;SEE IF TM
011766  001440                          BEQ     B0              ;IF NOT: BR
011770  012777  177777  166520          MOV     #-1,@FC
011776  012767  000032  166666          MOV     #32,MTC1
012004  012767  012016  166650          MOV     #BKTM,RTRN
012012  000167  007156                  JMP     TAPG            ;SPACE TO TM
012016  032777  010000  166562  BKTM:   BIT     #10000,@SWR     ;SEE IF SHOULD CHECK ERROR
012024  001021                          BNE     B0              ;IF NOT: BR
012026  012767  026231  166616          MOV     #MSG55,EMADDR
012034  032777  000004  166460          BIT     #4,@DS          ;SEE IF TM
012042  001006                          BNE     BKTM0           ;IF SO: BR
012044  012767  033462  007036          MOV     #RDATA,CADER
012052  004767  006134                  JSR     PC,ERPT         ;PRINT ERROR
012056  000404                          BR      B0
012060  012703  033462          BKTM0:  MOV     #RDATA,R3
012064  004767  005362                  JSR     PC,ER2
012070  016700  166460          B0:     MOV     RCNT,R0
012074  005100                          COM     R0              ;BUILD SPACE AMOUNT
012076  005200                          INC     R0
012100  012767  024777  166544          MOV     #MSG10,EMADDR   ;SET ERROR MESG ADDRESS
012106  010077  166404                  MOV     R0,@FC
012112  012767  000032  166552  BKRT:   MOV     #32,MTC1        ;SET SPACE REVERSE
012120  012767  012136  166534          MOV     #B1,RTRN        ;SET RETURN ADDRESS
012126  010067  166534                  MOV     R0,STAL         ;SET INTERRUPT TIME MULTIPLIER
012132  000167  007036                  JMP     TAPG            ;GO DO SPACE
012136  012703  033462          B1:     MOV     #RDATA,R3
012142  004767  005304                  JSR     PC,ER2
012146  016767  166424  166512  B2:     MOV     TSTAL,STAL      ;DO STALL
012154  004767  000002                  JSR     PC,STALL        ;STALL
012160  000207                          RTS     PC              ;EXIT
```

```
 1649
 1650                                   ;*******************************************************
 1651                                   ;STALL ROUTINE:
 1652                                   ;
 1653                                   ;THIS ROUTINE IS USED TO PROVIDE SOFTWARE DELAYS
 1654                                   ;DURING READ, WRITE, TURN AROUND, AND YOZZLE.
 1655                                   ;THE DELAY TIMES MAY BE SET BY THE OPERATOR AT
 1656                                   ;INITIAL START FROM 200(8) OR MAY BE MODIFIED
 1657                                   ;AT ANY TIME BY ENTERING CNTRL C ON THE TTY AND
 1658                                   ;INSERTING NEW VALUES IN RESPONSE TO THE REQUEST.
 1659                                   ;THE READ STALL AND THE WRITE STALL ARE DELAYS
 1660                                   ;EXECUTED BETWEEN EACH RECORD OF THE DATA BLOCK.
 1661                                   ;THE TURN AROUND STALL IS EXECUTED EACH TIME
 1662                                   ;THE DIRECTION OF TAPE MOVEMENT IS CHANGED AND
 1663                                   ;ALSO EACH TIME THE TAPE OPERATION CHANGES FROM
 1664                                   ;WRITE TO READ OR READ TO WRITE. THE YOZZLE
 1665                                   ;STALL IS EXECUTED ONLY DURING THE YOZZLE ROUTINE.
 1666                                   ;*******************************************************
 1667
 1668  012162  005367  166500  STALL:  DEC     STAL
 1669  012166  001375                 BNE     STALL           ;DELAY
 1670  012170  000207                 RTS     PC              ;EXIT
```

```
                                        ;*****************************************************
                                        ;RANDOM CHARACTER COUNT GENERATOR:
                                        ;
                                        ;THIS ROUTINE ENTERED VIA CONSOLE SWITCH
                                        ;SEVEN (7) IS USED TO GENERATE A RANDOM
                                        ;CHARACTER COUNT FOR EACH DATA BLOCK.
                                        ;ALL RECORDS WITHIN A GIVEN BLOCK WILL BE
                                        ;THE SAME, BUT EACH BLOCK WILL VARY.
                                        ;THE LIMITS ARE TWENTY (20) TO FOUR THOUSAND
                                        ;(4000) OCTAL CHARACTERS PER RECORD.
                                        ;*****************************************************

012172  012701  177760          CONTR:  MOV     #-20,R1         ;SET HIGH LIMIT
012176  012702  174000                  MOV     #-4000,R2       ;SET LOW LIMIT
012202  004767  011040                  JSR     PC,RANG         ;GO GENERATE NUMBER
012206  042767  000001  166412          BIC     #1,RANSAV
012214  016767  166406  166334          MOV     RANSAV,FMCNT    ;SET CHAR COUNT
012222  012767  177777  002044          MOV     #-1,PATS        ;PRESET DATA PATTERN
012230  000207                          RTS     PC              ;EXIT

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

012232  012702  000001          RCNTR:  MOV     #1,R2           ;SET LOW LIMIT
012236  012701  000500                  MOV     #500,R1         ;SET HIGH LIMIT
012242  004767  011000                  JSR     PC,RANG         ;GO GENERATE NUMBER
012246  016767  166354  166300          MOV     RANSAV,RCNT     ;SET RECORD COUNT
012254  000207                          RTS     PC              ;EXIT
```

```
1709
1710                    ;*********************************************************
1711                    ;TEST CONDITION ENTRY ROUTINE:
1712                    ;
1713                    ;THIS ROUTINE IS USED TO ALLOW THE OPERATOR
1714                    ;TO ENTER, AT THE TTY, THE NECESSARY PARAMETERS
1715                    ;TO RUN THE PROGRAM AS HE WISHES.  THE
1716                    ;ROUTINE IS ONLY ENTERED UPON INITIAL STARTING
1717                    ;FROM LOCATION 200(8).
1718                    ;THE MAIN PURPOSE OF THIS ROUTINE IS TO ESTABLISH
1719                    ;A TABLE OF DEVICES TO BE TESTED.  THIS TABLE
1720                    ;CONSISTS OF AN ENTRY FOR EACH OF ONE (1) TO
1721                    ;EIGHT (8) DEVICES.  EACH ENTRY CONTAINS THE
1722                    ;SLAVE NUMBER, DENSITY, PARITY, AND
1723                    ;FORMAT.  THE INFORMATION IS ENTERED
1724                    ;IN RESPONSE TO PRINTED REQUESTS AT THE TTY.
1725                    ;SLAVES MAY BE ENTERED IN ANY ORDER.  EACH
1726                    ;PARAMETER IS CHECKED FOR LEGALITY BEFORE BEING
1727                    ;SET INTO THE TABLE.
1728                    ;THE DRIVE NUMBER REQUEST WILL ALSO CHECK THE MASSBUS
1729                    ;FOR THE PRESENCE OF THE REQUESTED DRIVE.  IF IT IS NOT FOUND,
1730                    ;A NON-EXIST DRIVE MESSAGE WILL BE PRINTED AND ANOTHER DRIVE
1731                    ;REQUEST MADE.  WHEN THE DRIVE IS FOUND, THE RESPONSE IS STORED
1732                    ;AND CONTROL PASSED TO THE SLAVE SELECT ROUTINE.
1733                    ;THE SLAVE SELECT ROUTINE ALSO CHECKS FOR THE PRESENCE OF THE
1734                    ;SLAVE.  IF IT IS NOT PRESENT, A MESSAGE IS PRINTED AND ANOTHER
1735                    ;REQUEST IS ISSUED.  WHEN THE SELECTED SLAVE IS FOUND TO BE
1736                    ;PRESENT, A MESSAGE IS PRINTED IF IT IS A 7 CHANNEL DRIVE
1737                    ;TO ASSIST IN SELECTING DENSITY, PARITY, AND FORMAT.
1738                    ;UPON COMPLETION OF THE DEVICE TABLE, REQUESTS
1739                    ;ARE PRINTED FOR ENTRY OF THE NUMBER OF CHARACTERS
1740                    ;PER RECORD AND THE NUMBER OF RECORDS PER BLOCK.  THE
1741                    ;NEXT REQUEST IS FOR A PATTERN NUMBER TO BE USED
1742                    ;FOR WRITING AND CHECKING OF READ DATA.
1743                    ;FOLLOWING THE PATTERN REQUEST IS THE TAPE MARK OPTION.
1744                    ;RESPONDING TO THE REQUEST (TM=) WITH A ONE (1)
1745                    ;WILL CAUSE THE PROGRAM TO WRITE A TM AT THE
1746                    ;END OF EACH DATA BLOCK AND TO EXPECT THE
1747                    ;TM TO BE DETECTED IN EITHER READ FORWARD AND REVERSE
1748                    ;OR DURING SPACE OPERATION.  A RESPONSE OF ZERO (TM=0)
1749                    ;DISALLOWS WRITTING OF THE TM AND CAUSES THE READ
1750                    ;AND SPACE ROUTINES TO EXPECT NO TM TO BE PRESENT.
1751                    ;THE LAST REQUESTS ARE FOR ENTRY OF THE DESIRED
1752                    ;WRITE, READ, AND TURN AROUND STALLS.
1753                    ;*********************************************************
1754
1755  012256  005767  166352    TINP:   TST     TINF            ;SEE IF SHOULD INPUT FROM TTY
1756  012262  001001                    BNE     TINPA           ;IF SO: BR
1757  012264  000207                    RTS     PC              ;EXIT
1758  012266  005067  166402    TINPA:  CLR     UNP             ;CLEAR TABLE POINTER
1759  012272  005067  172514            CLR     RECTC           ;CLEAR EOT UNIT COUNTER
1760  012276  012700  000010            MOV     #10,R0          ;SET SIZE OF TABLE
1761  012302  012701  000746            MOV     #UNI,R1         ;SET START OF TABLE
1762  012306  005021            TINPB:  CLR     (R1)+           ;CLEAR TABLE
1763  012310  005300                    DEC     R0              ;SEE IF DONE
1764  012312  001375                    BNE     TINPB           ;IF NOT: BR
```

```
1765  012314  012704  025455                MOV     #MSG31,R4
1766  012320  005767  166410                TST     ASEQF           ;SEE IF AUTO SEQ
1767  012324  001402                        BEQ     TINPB1          ;IF NOT: BR
1768  012326  012704  025367                MOV     #MSG30,R4       ;SET AUTO SEQ HDR
1769  012332  104000          TINPB1: TTOUTT                          ;PRINT PROGRAM NAME
1770  012334  005767  001472                TST     SCVFL           ;SEE IF SHORT CONVERSATION
1771  012340  001067                        BNE     TINPC           ;IF SO: BR
1772  012342  012704  026521                MOV     #MSG74,R4
1773  012346  104000                        TTOUTT                  ;REQUEST STARTING REGISTER ADDRESS
1774  012350  016703  166170                MOV     REGS,R3
1775  012354  104002                        OCTPP                   ;PRINT CURRENT REG START
1776  012356  012705  000544                MOV     #REGS,R5        ;SAVE ADDRESS LOCATION
1777  012362  012701  000006                MOV     #6,R1           ;SET SIZE OF ENTRY
1778  012366  012702  176400                MOV     #176400,R2      ;SET UPPER LIMIT
1779  012372  012703  172300                MOV     #172300,R3      ;SET LOWER LIMIT
1780  012376  004767  010676                JSR     PC,TTR          ;GO GET RESPONSE
1781  012402  012704  026544                MOV     #MSG75,R4
1782  012406  104000                        TTOUTT                  ;GO REQUEST VECTOR ADDRESS
1783  012410  016703  166132                MOV     VECT,R3
1784  012414  104002                        OCTPP                   ;PRINT CURRENT VECTOR
1785  012416  012705  000546                MOV     #VECT,R5        ;SET SAVE LOCATION
1786  012422  012701  000003                MOV     #3,R1           ;SET SIZE OF ENTRY
1787  012426  012702  000224                MOV     #224,R2         ;SET UPPER LIMIT
1788  012432  012703  000150                MOV     #150,R3         ;SET LOWER LIMIT
1789  012436  004767  010636                JSR     PC,TTR          ;GO GET RESPONSE
1790  012442  016700  166100                MOV     VECT,R0         ;GET VECTOR ADDRESS
1791  012446  012720  021714                MOV     #MTINT,(R0)+    ;LOAD VECTOR WITH HANDLER ADDRESS
1792  012452  012710  000340                MOV     #340,(R0)       ;LOAD PRIORITY LEVEL
1793  012456  016700  166062                MOV     REGS,R0         ;GET STARTING REGISTER ADDRESS
1794  012462  012701  000016                MOV     #16,R1          ;SET NUMBER OF REGISTERS
1795  012466  012702  000510                MOV     #C1,R2          ;GET FIRST ADDRESS LOCATION
1796  012472  010022          TINPB0: MOV     R0,(R2)+        ;BUILD TABLE OF ADDRESSES
1797  012474  062700  000002                ADD     #2,R0           ;BUMP ADDRESS
1798  012500  005301                        DEC     R1              ;SEE IF DONE
1799  012502  001373                        BNE     TINPB0          ;IF NOT: BR
1800  012504  005767  166224                TST     ASEQF           ;SEE IF AUTO SEQ
1801  012510  001403                        BEQ     TINPC           ;IF NOT: BR
1802  012512  005726                        TST     (SP)+           ;RESET STACK POINTER
1803  012514  000167  007212                JMP     ASEQ            ;GO TO AUTO SEQUENCE
1804  012520  012777  000040  165772 TINPC: MOV     #40,@CS         ;INITIALIZE
1805  012526  012704  026165                MOV     #MSG52,R4
1806  012532  104060                        TTOUTT                  ;REQUEST DRIVE NUMBER
1807  012534  012705  000550                MOV     #DVN,R5         ;GET ADDRESS
1808  012540  012701  000001                MOV     #1,R1           ;SET SIZE OF RESPONSE
1809  012544  012702  000007                MOV     #7,R2           ;SET UPPER LIMIT
1810  012550  012703  000000                MOV     #0,R3           ;SET LOWER LIMIT
1811  012554  004767  010520                JSR     PC,TTR          ;GO GET DRIVE NUMBER
1812  012560  016777  165764  165732        MOV     DVN,@CS
1813  012566  005777  165716                TST     @C1             ;ACCESS DRIVE
1814  012572  032777  010000  165720        BIT     #10000,@CS      ;SEE IF NED
1815  012600  001412                        BEQ     TINPD           ;IF NOT: BR
1816  012602  012704  026456                MOV     #MSG71,R4
1817  012606  104000                        TTOUTT                  ;PRINT NED
1818  012610  016704  165674                MOV     C1,R4
1819  012614  005204                        INC     R4
1820  012616  152714  000100                BISB    #100,(R4)       ;CLEAR TRE
```

```
1821  012622  000167  177672                 JMP     TINPC           ;RETRY DVN
1822  012626  012704  025553         TINPO:  MOV     #MSG32,R4
1823  012632  104000                         TTOUTT                  ;PRINT UNIT NUMBER REQUEST
1824  012634  005067  166004                 CLR     TEMP2           ;CLEAR BUFFER
1825  012640  012705  000644                 MOV     #TEMP2,R5       ;SET UNIT DESCRIPTION BUFFER ADDRESS
1826  012644  012701  000001                 MOV     #1,R1           ;SET NUMBER OF CHARACTERS TO INPUT
1827  012650  012702  000007                 MOV     #7,R2           ;SET MAXIMUM LIMIT
1828  012654  012703  000000                 MOV     #0,R3           ;SET MINIMUM LIMIT
1829  012660  004767  010414                 JSR     PC,TTR          ;GO GET UNIT NUMBER
1830  012664  005767  165752                 TST     TEMP1           ;SEE IF HAVE NEW PARAMETER
1831  012670  001014                         BNE     TINPOB          ;IF SO: BR
1832  012672  005767  165776                 TST     UNP             ;SEE IF FIRST ENTRY
1833  012676  001002                         BNE     TINPOA          ;IF NOT: BR
1834  012700  000167  177722                 JMP     TINPO           ;ELSE RETRY
1835  012704  016700  165764         TINPOA: MOV     UNP,R0
1836  012710  012760  177777  000746         MOV     #-1,UN1(R0)     ;SET END UNIT TABLE
1837  012716  000167  000364                 JMP     TINP2C          ;GO GET RECORD COUNT
1838  012722  016700  165746         TINPOB: MOV     UNP,R0
1839  012726  042760  000007  000746         BIC     #7,UN1(R0)      ;CLEAR UNIT NUMBER
1840  012734  004767  001106                 JSR     PC,TPOS1        ;GO LOAD UNIT NUMBER TO PROPER POSITION
1841  012740  012777  000040  165552         MOV     #40,@CS
1842  012746  016777  165576  165544         MOV     DVN,@CS
1843  012754  016077  000746  165560         MOV     UN1(R0),@C2     ;LOAD UNIT NUMBER
1844  012762  032777  002000  165546 TINPOC: BIT     #2000,@DT       ;SEE  IF SLAVE PRESENT
1845  012770  001005                         BNE     TINPOD          ;IF SO: BR
1846  012772  012704  026244                 MOV     #MSG57,R4
1847  012776  104000                         TTOUTT                  ;PRINT NON-EXIST SLAVE
1848  013000  000167  177622                 JMP     TINPO           ;REDO
1849  013004  022777  142011  165524 TINPOD: CMP     #142011,@DT     ;SEE IF 9TRK TM02,TU16
1850  013012  001406                         BEQ     TINPOE          ;IF SO: BR
1851  013014  012704  026140                 MOV     #MSG50,R4       ;ILLEGAL DRIVE TYPE
1852  013020  104000                         TTOUTT                  ;GO PRINT
1853  013022  017703  165510                 MOV     @DT,R3
1854  013026  104002                         OCTPP                   ;PRINT DRIVE TYPE REGISTER
1855  013030  012704  024771         TINPOE: MOV     #MSG9,R4
1856  013034  104000                         TTOUTT                  ;PRINT SERIAL NUMBER TAG
1857  013036  017703  165476                 MOV     @SN,R3
1858  013042  004767  011202                 JSR     PC,$NPT         ;PRINT SERIAL NUMBER
1859  013046  012704  025574         TINP1:  MOV     #MSG33,R4
1860  013052  104000                         TTOUTT                  ;PRINT DENSITY REQUEST
1861  013054  005067  165564                 CLR     TEMP2           ;CLEAR BUFFER
1862  013060  012701  000001                 MOV     #1,R1           ;SET NUMBER OF CHARACTERS TO INPUT
1863  013064  012702  000007                 MOV     #7,R2           ;SET MAXIMUM LIMIT
1864  013070  012703  000000                 MOV     #0,R3           ;SET MINIMUM LIMIT
1865  013074  004767  010200                 JSR     PC,TTR          ;GO GET DENSITY
1866  013100  005767  165536                 TST     TEMP1           ;SEE IF HAVE NEW PARAMETER
1867  013104  001407                         BEQ     TINP2           ;IF NOT: BR
1868  013106  042767  003400  165436         BIC     #3400,UDES      ;ELSE CLEAR OLD PARAMETER
1869  013114  012703  000010                 MOV     #10,R3          ;SET POSITION FACTOR
1870  013120  004767  000710                 JSR     PC,TPOS         ;GO LOAD DENSITY INTO PROPER POSITION
1871  013124  012704  025610         TINP2:  MOV     #MSG34,R4
1872  013130  104000                         TTOUTT                  ;PRINT PARITY REQUEST
1873  013132  005067  165506                 CLR     TEMP2           ;CLR BUFFER
1874  013136  012701  000001                 MOV     #1,R1           ;SET NUMBER OF CHARACTERS TO INPUT
1875  013142  012702  000001                 MOV     #1,R2           ;SET MAXIMUM LIMIT
1876  013146  012703  000000                 MOV     #0,R3           ;SET MINIMUM LIMIT
```

```
1877  013152  004767  010122            JSR     PC,TTR          ;GO INPUT PARITY
1878  013156  005767  165460            TST     TEMP1           ;SEE IF HAVE NEW PARAMETER
1879  013162  001407                    BEQ     TINP2A          ;IF NOT: BR
1880  013164  042767  000010  165360    BIC     #10,UDES        ;ELSE CLEAR OLD PARAMETER
1881  013172  012703  000003            MOV     #3,R3           ;SET POSITION FACTOR
1882  013176  004767  000632            JSR     PC,TPOS         ;GO LOAD PARITY TO PROPER POSITION
1883  013202  012704  026207  TINP2A:  MOV     #MSG53,R4
1884  013206  104000                    TTOUTT                  ;REQUEST FORMAT
1885  013210  005067  165430            CLR     TEMP2
1886  013214  012701  000002            MOV     #2,R1
1887  013220  012702  000016            MOV     #16,R2
1888  013224  012703  000014            MOV     #14,R3
1889  013230  004767  010044            JSR     PC,TTR          ;GO GET FORMAT
1890  013234  005767  165402            TST     TEMP1           ;SEE IF NEW PARAMETER
1891  013240  001407                    BEQ     TINP2B          ;IF NOT: BR
1892  013242  042767  000170  165332    BIC     #170,UDES
1893  013250  012703  000004            MOV     #4,R3
1894  013254  004767  000554            JSR     PC,TPOS
1895  013260  005267  171526  TINP2B:  INC     REOTC           ;BUMP EOT UNIT COUNTER
1896  013264  022767  000016  165402    CMP     #16,UNP         ;SEE IF DONE UNITS
1897  013272  001405                    BEQ     TINP2C          ;IF SO: BR
1898  013274  062767  000002  165372    ADD     #2,UNP          ;POINT TO NEXT UNIT
1899  013302  000167  177320            JMP     TINPO           ;ELSE LOOK FOR NEXT UNIT
1900  013306  005067  165362  TINP2C:  CLR     UNP             ;CLEAR UNIT POINTER
1901  013312  016700  171474            MOV     REOTC,R0
1902  013316  000367  171470            SWAB    REOTC
1903  013322  110067  171464            MOVB    R0,REOTC        ;SET UNIT EOT COUNTER
1904  013326  012704  025623  TINP3:  MOV     #MSG35,R4
1905  013332  104000                    TTOUTT                  ;PRINT RECORD COUNT REQUEST
1906  013334  016703  165214            MOV     RCNT,R3
1907  013340  104002                    OCTPP                   ;PRINT RECORD COUNT
1908  013342  012705  000554            MOV     #RCNT,R5        ;SET RECORD COUNT ADDRESS
1909  013346  012701  000006            MOV     #6,R1           ;SET NUMBER OF CHARACTERS TO INPUT
1910  013352  012702  177777            MOV     #-1,R2          ;SET MAXIMUM LIMIT
1911  013356  012703  000001            MOV     #1,R3           ;SET MINIMUM LIMIT
1912  013362  004767  007712            JSR     PC,TTR          ;GO GET RECORD COUNT
1913  013366  016767  165162  165234    MOV     RCNT,RCSAV      ;SAVE RECORD COUNT
1914  013374  012704  025644            MOV     #MSG36,R4
1915  013400  104000                    TTOUTT                  ;PRINT CHARACTER COUNT REQUEST
1916  013402  005467  165150            NEG     FMCNT
1917  013406  016703  165144            MOV     FMCNT,R3
1918  013412  104002                    OCTPP                   ;PRINT CHAR COUNT
1919  013414  012705  000556            MOV     #FMCNT,R5       ;SET CHARACTER COUNT ADDRESS
1920  013420  012701  000006            MOV     #6,R1           ;SET NUMBER OF CHARACTERS TO INPUT
1921  013424  012702  004000            MOV     #4000,R2        ;SET MAXIMUM LIMIT
1922  013430  012703  000004            MOV     #4,R3           ;SET MINIMUM LIMIT
1923  013434  004767  007640            JSR     PC,TTR          ;GO GET CHARACTER COUNT
1924  013440  005467  165112            NEG     FMCNT           ;SET TO TWO'S COMPLIMENT
1925  013444  016767  165106  165160    MOV     FMCNT,FCSAV     ;SAVE FRAME COUNT
1926  013452  012704  025670            MOV     #MSG37,R4       ;PRINT PATTERN NUMBER REQUEST
1927  013456  104000                    TTOUTT
1928  013460  016703  165074            MOV     PATRN,R3
1929  013464  104002                    OCTPP                   ;PRINT PATTERN
1930  013466  005067  000756            CLR     DOFL            ;CLEAR EXTERNAL DATA FLAG
1931  013472  012705  000560            MOV     #PATRN,R5       ;SET PATTERN NUMBER ADDRESS
1932  013476  012701  000002            MOV     #2,R1           ;SET NUMBER OF CHARACTERS TO INPUT
```

```
 1933  013502  012702  000015                MOV     #15,R2          ;SET MAXIMUM LIMIT
 1934  013506  012703  000000                MOV     #0,R3           ;SET MINIMUM LIMIT
 1935  013512  004767  007562                JSR     PC,TTR          ;GO GET PATTERN NUMBER
 1936  013516  012704  026404                MOV     #MSG69,R4
 1937  013522  104000                        TTOUTT                  ;REQUEST TM
 1938  013524  016703  165034                MOV     TMEX,R3
 1939  013530  104002                        OCTPP                   ;PRINT CURRENT TM FLAG SETTING
 1940  013532  012705  000564                MOV     #TMEX,R5        ;GET TM FLAG ADDRESS
 1941  013536  012701  000001                MOV     #1,R1           ;SET SIZE OF RESPONSE
 1942  013542  012702  000001                MOV     #1,R2           ;SET UPPER LIMIT
 1943  013546  012703  000000                MOV     #0,R3           ;SET LOWER LIMIT
 1944  013552  004767  007522                JSR     PC,TTR          ;TM 1=YES
 1945  013556  012704  025121                MOV     #MSG21,R4
 1946  013562  104000                        TTOUTT                  ;REQUEST INTERCHANGE READ
 1947  013564  016703  164776                MOV     INTRF,R3
 1948  013570  104002                        OCTPP                   ;PRINT CURRENT SETTING
 1949  013572  012705  000566                MOV     #INTRF,R5       ;GET FLAG ADDRESS
 1950  013576  012701  000001                MOV     #1,R1           ;SET SIZE OF RESPONSE
 1951  013602  012702  000001                MOV     #1,R2           ;SET UPPER LIMIT
 1952  013606  012703  000000                MOV     #0,R3           ;SET LOWER LIMIT
 1953  013612  004767  007462                JSR     PC,TTR          ;GO GET RESPONSE
 1954  013616  012704  025713                MOV     #MSG38,R4
 1955  013622  104000                        TTOUTT                  ;REQUEST SINGLE PASS
 1956  013624  016703  164740                MOV     SPFLG,R3
 1957  013630  104002                        OCTPP                   ;PRINT CURRENT SETTING
 1958  013632  012705  000570                MOV     #SPFLG,R5       ;SET ADDRESS OF FLAG
 1959  013636  012701  000001                MOV     #1,R1           ;SET SIZE OF RESPONSE
 1960  013642  012702  000001                MOV     #1,R2           ;SET UPPER LIMIT
 1961  013646  012703  000000                MOV     #0,R3           ;SET LOWER LIMIT
 1962  013652  004767  007422                JSR     PC,TTR          ;GO GET RESPONSE
 1963  013656  005767  000150        TINP4:  TST     SCVFL           ;SEE IF SHORT CONVERSATION
 1964  013662  001060                        BNE     TINPX           ;IF SO: BR
 1965  013664  012704  025733                MOV     #MSG40,R4
 1966  013670  104000                        TTOUTT                  ;PRINT READ STALL REQUEST
 1967  013672  016703  164674                MOV     RSTAL,R3
 1968  013676  104002                        OCTPP                   ;PRINT READ STALL
 1969  013700  012705  000572                MOV     #RSTAL,R5       ;SET READ STALL ADDRESS
 1970  013704  012701  000006                MOV     #6,R1           ;SET NUMBER OF CHARACTERS TO INPUT
 1971  013710  012702  177777                MOV     #-1,R2          ;SET MAXIMUM LIMIT
 1972  013714  012703  000001                MOV     #1,R3           ;SET MINIMUM LIMIT
 1973  013720  004767  007354                JSR     PC,TTR          ;GO GET READ STALL
 1974  013724  012704  025761                MOV     #MSG41,R4
 1975  013730  104000                        TTOUTT                  ;PRINT WRITE STALL REQUEST
 1976  013732  016703  164636                MOV     WSTAL,R3
 1977  013736  104002                        OCTPP                   ;PRINT READ STALL
 1978  013740  012705  000574                MOV     #WSTAL,R5       ;SET WRITE STALL ADDRESS
 1979  013744  012701  000006                MOV     #6,R1           ;SET NUMBER OF CHARACTERS TO INPUT
 1980  013750  012702  177777                MOV     #-1,R2          ;SET MAXIMUM LIMIT
 1981  013754  012703  000001                MOV     #1,R3           ;SET MINIMUM LIMIT
 1982  013760  004767  007314                JSR     PC,TTR          ;GO GET WRITE STALL
 1983  013764  012704  025773                MOV     #MSG42,R4
 1984  013770  104000                        TTOUTT                  ;PRINT TURN AROUND STALL REQUEST
 1985  013772  016703  164600                MOV     TSTAL,R3
 1986  013776  104002                        OCTPP                   ;PRINT TA STALL
 1987  014000  012705  000576                MOV     #TSTAL,R5       ;SET TURN AROUND STALL ADDRESS
 1988  014004  012701  000006                MOV     #6,R1           ;SET NUMBER OF CHARACTERS TO INPUT
```

```
 1989  014010  012702  177777                MOV     #-1,R2          ;SET MAXIMUM LIMIT
 1990  014014  012703  000001                MOV     #1,R3           ;SET MINIMUM LIMIT
 1991  014020  004767  007254                JSR     PC,TTR          ;GO GET TURN AROUND STALL
 1992  014024  005067  000002        TINPX:  CLR     SCVFL           ;CLEAR SHORT CONVERSATION FLAG
 1993  014030  000207                        RTS     PC              ;EXIT
 1994  014032  000000                SCVFL:  0                       ;SHORT CONVERSATION FLAG
 1995
 1996                                        ;UNIT DESCRIPTION POSITIONING SUBROUTINE**********
 1997
 1998  014034  000241                TPOS:   CLC
 1999  014036  006167  164602                ROL     TEMP2           ;POSITION CHARACTER
 2000  014042  005303                        DEC     R3              ;SEE IF DONE
 2001  014044  001373                        BNE     TPOS            ;IF NOT: BR
 2002  014046  016700  164622        TPOS1:  MOV     UNP,R0          ;LOAD UNIT POINTER
 2003  014052  056760  164566  000746        BIS     TEMP2,UN1(R0)   ;LOAD CHARACTER INTO UN1(R0)
 2004  014060  000207                        RTS     PC              ;EXIT
 2005
```

```
2006
2007                                    ;*******************************************************
2008                                    ;DATA SETUP ROUTINE:
2009                                    ;
2010                                    ;THIS ROUTINE IS USED TO GENERATE INTO THE ENTIRE
2011                                    ;WRITE BUFFER (4000 OCTAL CHARACTERS) THE DATA PATTERN
2012                                    ;SELECTED BY THE OPERATOR.  THERE ARE 15 (8) FIXED
2013                                    ;DATA PATTERNS AVAILABLE AND ONE SELECTION (DATA PATTERN 0)
2014                                    ;WHICH WILL READ ANY PATTERN PRESENTED AT THE
2015                                    ;HIGH SPEED PAPER TAPE READER.  THIS TAPE MUST BE PREPARED
2016                                    ;BY USING THE PROGRAM CALLED DTC. (MAINDEC-11-DZTUF-A-D)
2017                                    ;RANDOM DATA MAY ALSO BE USED VIA CONSOLE
2018                                    ;SWITCH EIGHT (8).
2019                                    ;THIS ROUTINE IS ALSO USED TO CLEAR OUT THE
2020                                    ;READ BUFFER (4000 OCTAL CHARACTERS) BEFORE EACH
2021                                    ;RECORD IS READ.
2022                                    ;*******************************************************
2023
2024  014062  005767  001006    DSUP:   TST     RDFL            ;SEE IF DID RANDOM DATA
2025  014066  001401                    BEQ     DSO             ;IF NOT: BR
2026  014070  000445                    BR      DS1             ;ELSE EXIT
2027  014072  005767  164636    DSO:    TST     ASEQF           ;SEE IF AUTO SEQ
2028  014076  001406                    BEQ     DSOC            ;IF NOT: BR
2029  014100  005767  164454            TST     PATRN           ;SEE IF AUTO RANDOM
2030  014104  100003                    BPL     DSOC            ;IF NOT: BR
2031  014106  004767  000714            JSR     PC,DATR         ;ELSE GO GENERATE RANDOM DATA
2032  014112  000207                    RTS     PC              ;RETURN
2033  014114  026767  164440  000152  DSOC:   CMP     PATRN,PATS      ;SEE IF NEW PATTERN
2034  014122  001014                    BNE     DSOA            ;IF SO: BR
2035  014124  016703  164422            MOV     UDES,R3         ;GET UNIT DESCRIPTION
2036  014130  042703  177767            BIC     #177767,R3      ;MASK EVEN PARITY
2037  014134  026703  000136            CMP     PARS,R3         ;SEE IF SAME AS LAST TIME
2038  014140  001404                    BEQ     DSOB            ;IF SO: BR
2039  014142  010367  000130            MOV     R3,PARS         ;SAVE PARITY
2040  014146  004767  000724            JSR     PC,CRCLRC       ;GO GENERATE EXPT CRC/LRC
2041  014152  000207            DSOB:   RTS     PC
2042  014154  012703  027454    DSOA:   MOV     #WDATA,R3       ;R3 = ADDRS OF WRITE BUFFER
2043  014160  016701  164374            MOV     PATRN,R1        ;R1 = PATTERN SELECTOR
2044  014164  010167  000104            MOV     R1,PATS
2045  014170  062701  000001            ADD     #1,R1           ;BUMP POINTER
2046  014174  000241                    CLC
2047  014176  006101                    ROL     R1              ;MAKE PATTERN SELECTOR EVEN
2048  014200  000171  002770            JMP     @DATBL(R1)      ;GO GENERATE PATTERN
2049  014204  032777  010000  164324  DS1:    BIT     #10000,@DT      ;SEE IF 7 CH
2050  014212  001410                    BEQ     DS2A            ;IF NOT: BR
2051  014214  012702  002002            MOV     #2002,R2        ;SET BUFFER SIZE
2052  014220  012701  027454            MOV     #WDATA,R1       ;SET START OF BUFFER
2053  014224  042721  140300    DS2:    BIC     #140300,(R1)+   ;MASK FOR 7 CH
2054  014230  005302                    DEC     R2              ;SEE IF DONE
2055  014232  001374                    BNE     DS2             ;IF NOT: BR
2056  014234  004767  000636    DS2A:   JSR     PC,CRCLRC       ;GO GENERATE EXPT CRC/LRC
2057  014240  012702  002000    DS3:    MOV     #2000,R2        ;R2=BUFFER SIZE
2058  014244  012701  033462            MOV     #RDATA,R1       ;R1=READ DATA START
2059  014250  005021            DS4:    CLR     (R1)+           ;CLEAR BUFFER
2060  014252  005302                    DEC     R2              ;SEE IF DONE ALL
2061  014254  001375                    BNE     DS4             ;IF NOT: BR
```

```
2062  014256  016767  164270  000012          MOV     UDES,PARS       ;GET UNIT DESCRIPTION
2063  014264  042767  177767  000004          BIC     #177767,PARS    ;MASK PARITY
2064  014272  000207                          RTS     PC              ;EXIT
2065  014274  177777                  PATS:   -1                      ;PATTERN NUMBER SAVE
2066  014276  000000                  PARS:   0
2067
2068                                          ;EXTERNAL DATA INPUT FROM H/S READER (256 CHARACTER MAXIMUM)
2069
2070  014300  005767  000144          DATO:   TST     DOFL            ;SEE IF SHOULD DO EXTERNAL INPUT
2071  014304  001337                          BNE     DS1             ;IF NOT: BR
2072  014306  012767  000001  000134          MOV     #1,DOFL         ;SET EXTERNAL FLAG
2073  014314  005077  164302                  CLR     @PRB            ;CLEAR READER BUFFER
2074  014320  005077  164274                  CLR     @PRS            ;CLEAR READER STATUS
2075  014324  005067  164312                  CLR     TEMP1           ;CLEAR FOR USE AS CHARACTER FLAG
2076  014330  052777  000001  164262  DATOA:  BIS     #1,@PRS         ;START READER
2077  014336  032777  000200  164254  DATOB:  BIT     #200,@PRS       ;SEE IF DONE
2078  014344  001774                          BEQ     DATOB           ;IF NOT : BR
2079  014346  005001                          CLR     R1              ;CLEAR SAVE LOCATION
2080  014350  117701  164246                  MOVB    @PRB,R1         ;SAVE CHARACTER
2081  014354  005767  164262                  TST     TEMP1           ;SEE IF HAVE FOUND START CHARACTER
2082  014360  001012                          BNE     DATOC           ;IF SO : BR
2083  014362  105701                          TSTB    R1              ;SEE IF CHARACTER IS 0
2084  014364  001761                          BEQ     DATOA           ;IF SO : BR
2085  014366  012767  000001  164246          MOV     #1,TEMP1        ;ELSE SET CHARACTER FOUND FLAG
2086  014374  010167  164244                  MOV     R1,TEMP2        ;SAVE DATA SIZE
2087  014400  010102                          MOV     R1,R2           ;SAVE DATA SIZE
2088  014402  000167  177722                  JMP     DATOA           ;GO GET FIRST DATA CHAR
2089  014406  110123                  DATOC:  MOVB    R1,(R3)+        ;LOAD BUFFER
2090  014410  005302                          DEC     R2              ;SEE IF READ ALL
2091  014412  001346                          BNE     DATOA           ;IF NOT : BR
2092  014414  012701  027454          DATOD:  MOV     #WDATA,R1       ;R1 = START OF WRITE BUFFER
2093  014420  016702  164220                  MOV     TEMP2,R2        ;R2 = SIZE OF DATA FIELD
2094  014424  112123                  DATOE:  MOVB    (R1)+,(R3)+     ;REPEAT LOAD OF DATA FIELD
2095  014426  022703  033462                  CMP     #RDATA,R3       ;SEE IF DONE
2096  014432  003002                          BGT     DATOF           ;IF NOT: BR
2097  014434  000167  177544                  JMP     DS1             ;EXIT
2098  014440  005302                  DATOF:  DEC     R2              ;SEE IF AT END OF DATA FIELD
2099  014442  001370                          BNE     DATOE           ;IF NOT : BR
2100  014444  000167  177744                  JMP     DATOD           ;ELSE RESTART FILL
2101  014450  000000                  DOFL:   0                       ;EXTERNAL DATA FLAG=1 IF ALREADY DONE
2102
```

```
2103                                        ;ALL ONES**********
2104
2105  014452  012701  177777    DAT1:     MOV     #-1,R1          ;R1=DATA
2106  014456  012702  002002    DAT1A:    MOV     #2002,R2        ;R2=WORD COUNT +2
2107  014462  010123             DAT1B:    MOV     R1,(R3)+        ;LOAD BUFFER
2108  014464  005302                       DEC     R2              ;SEE IF DONE
2109  014466  001375                       BNE     DAT1B           ;IF NOT: BR
2110  014470  000167  177510               JMP     DS1             ;RETURN
2111
2112                                        ;ALL ZEROS**********
2113
2114  014474  005001             DAT2:     CLR     R1              ;R1=DATA
2115  014476  000167  177754               JMP     DAT1A           ;LOAD BUFFER
2116
2117                                        ;WALKING ONE**********
2118
2119  014502  012701  000001     DAT3:     MOV     #1,R1           ;R1=DATA
2120  014506  000241                       CLC
2121  014510  012702  004004     DAT3A:    MOV     #4004,R2        ;R2=CHARACTER COUNT+4
2122  014514  110123             DAT3B:    MOVB    R1,(R3)+        ;LOAD BUFFER
2123  014516  106101                       ROLB    R1              ;SET NEXT CHARACTER
2124  014520  005302                       DEC     R2              ;SEE IF DONE
2125  014522  001374                       BNE     DAT3B           ;IF NOT: BR
2126  014524  000167  177454               JMP     DS1             ;RETURN
2127
2128                                        ;WALKING ZERO**********
2129
2130  014530  012701  000376     DAT4:     MOV     #376,R1         ;R1=START OF DATA
2131  014534  000261                       SEC
2132  014536  000167  177746               JMP     DAT3A           ;LOAD BUFFER
2133
2134                                        ;ALTERNATING ONE/ZERO**********
2135
2136
2137  014542  012701  052525     DAT5:     MOV     #52525,R1       ;R1=DATA
2138  014546  000167  177704               JMP     DAT1A           ;LOAD BUFFER
2139
2140                                        ;ALTERNATING ZERO/ONE**********
2141
2142  014552  012701  125252     DAT6:     MOV     #125252,R1      ;R1=DATA
2143  014556  000167  177674               JMP     DAT1A           ;LOAD BUFFER
2144
2145                                        ;ONE/ZERO IN ALTERNATING WORDS**********
2146
2147  014562  012701  125252     DAT7:     MOV     #125252,R1      ;SET WORD 1
2148  014566  012702  052525               MOV     #52525.R2       ;SET WORD 2
2149  014572  012704  001002               MOV     #1002,R4        ;SET NUMBER OF ENTRIES
2150  014576  010123             DAT7A:    MOV     R1,(R3)+        ;LOAD WORD 1
2151  014600  010223                       MOV     R2,(R3)+        ;LOAD WORD 2
2152  014602  005304                       DEC     R4              ;SEE IF DONE
2153  014604  001374                       BNE     DAT7A           ;IF NOT: BR
2154  014606  000167  177372               JMP     DS1             ;RETURN
2155
```

```
2156                                        ;WALKING ONE/ALL ONE IN ALTERNATING CHARS****
2157
2158  014612  012702  002002    DAT10:  MOV     #2002,R2        ;SET BUFFER SIZE
2159  014616  012701  000001            MOV     #1,R1           ;SET WALK BASE
2160  014622  000241                    CLC
2161  014624  012713  177400    DAT10A: MOV     #177400,(R3)    ;LOAD ALL ONE BYTE
2162  014630  050123                    BIS     R1,(R3)+        ;LOAD WALK BYTE
2163  014632  106101                    ROLB    R1              ;WALK ONE
2164  014634  005302                    DEC     R2
2165  014636  001372                    BNE     DAT10A          ;DO FULL BUFFER
2166  014640  000167  177340            JMP     DS1             ;RETURN
2167
2168                                        ;ALL BITS 0-377**********
2169
2170  014644  005001    DAT11:  CLR     R1              ;R1=STARTING DATA
2171  014646  012702  004004            MOV     #4004,R2        ;R2=CHARACTER COUNT+4
2172  014652  110123    DAT11A: MOVB    R1,(R3)+        ;LOAD BUFFER
2173  014654  105201                    INCB    R1              ;BUMP DATA
2174  014656  005302                    DEC     R2              ;SEE IF DONE
2175  014660  001374                    BNE     DAT11A          ;IF NOT: BR
2176  014662  000167  177316            JMP     DS1             ;RETURN
2177
2178                                        ;ALL BITS 377-0**********
2179
2180  014666  012701  000377    DAT12:  MOV     #377,R1         ;R1=STARTING DATA
2181  014672  012702  004004            MOV     #4004,R2        ;R2=CHARACTER COUNT+4
2182  014676  110123    DAT12A: MOVB    R1,(R3)+        ;LOAD BUFFER
2183  014700  105301                    DECB    R1              ;BUMP DATA
2184  014702  005302                    DEC     R2              ;SEE IF DONE
2185  014704  001374                    BNE     DAT12A          ;IF NOT: BR
2186  014706  000167  177272            JMP     DS1             ;RETURN
2187
2188                                        ;ALTERNATING CHARACTERS 0 AND 377**********
2189
2190  014712  012701  000377    DAT13:  MOV     #377,R1         ;R1 = DATA
2191  014716  000167  177534            JMP     DAT1A           ;LOAD BUFFER
2192
2193                                        ;WALKING ZERO/ALL ZERO IN ALTERNATING CHARS******
2194
2195  014722  012702  002002    DAT14:  MOV     #2002,R2        ;SET BUFFER SIZE
2196  014726  012701  000376            MOV     #376,R1         ;SET WALK BASE
2197  014732  000261                    SEC
2198  014734  010113    DAT14A: MOV     R1,(R3)         ;LOAD WALK BYTE
2199  014736  042723  177400            BIC     #177400,(R3)+   ;CLEAR HIGH BYTE
2200  014742  106101                    ROLB    R1              ;WALK ZERO BIT
2201  014744  005302                    DEC     R2
2202  014746  001372                    BNE     DAT14A          ;FILL BUFFER
2203  014750  000167  177230            JMP     DS1             ;RETURN
2204
```

```
2205                                        ;AUTO SEQUENCE PATTERN*********
2206
2207  014754  012702  000200    DAT15:  MOV     #200,R2         ;SET NUMBER OF ENTRIES
2208  014760  012701  015006    DAT15A: MOV     #APATS,R1       ;SET START OF PATTERN
2209  014764  012704  000010            MOV     #10,R4          ;SET SIZE OF PATTERN
2210  014770  012123            DAT15B: MOV     (R1)+,(R3)+     ;FILL BUFFER
2211  014772  005304                    DEC     R4              ;SEE IF DONE PATTERN
2212  014774  001375                    BNE     DAT15B          ;IF NOT: BR
2213  014776  005302                    DEC     R2              ;SEE IF DONE BUFER
2214  015000  001367                    BNE     DAT15A          ;IF NOT: BR
2215  015002  000167  177176            JMP     DS1             ;RETURN
2216  015006  000000            APATS:  0
2217  015010  177400                    177400
2218  015012  000377                    377
2219  015014  000000                    0
2220  015016  177777                    -1
2221  015020  000377                    377
2222  015022  177400                    177400
2223  015024  177777                    -1
2224
2225                                        ;RANDOM DATA GENERATOR SUBROUTINE**********
2226
2227  015026  016704  163524    DATR:   MOV     FMCNT,R4        ;SET NUMBER OF FRAMES
2228  015032  012703  027454            MOV     #WDATA,R3       ;SET ADDRESS OF START OF BUFFER
2229  015036  012701  177777            MOV     #-1,R1          ;SET HIGH LIMIT
2230  015042  005002                    CLR     R2              ;SET LOW LIMIT
2231  015044  004767  006176    DATRG:  JSR     PC,RANG         ;GO GENERATE NUMBER
2232  015050  016723  163552            MOV     RANSAV,(R3)+    ;LOAD BUFFER
2233  015054  005204                    INC     R4              ;SEE IF DONE WHOLE BUFFER
2234  015056  001372                    BNE     DATRG           ;IF NOT: BR
2235  015060  004767  177120            JSR     PC,DS1          ;GO CHECK FOR 7 CH
2236  015064  012767  000001  000002    MOV     #1,RDFL         ;SET RANDOM DATA FLAG
2237  015072  000207                    RTS     PC              ;EXIT
2238  015074  000000            RDFL:   0                       ;RANDOM DATA SELECT FLAG
```

```
2239
2240                                    ;*************************************************************
2241                                    ;CRC/LRC CHARACTER BUILD:
2242                                    ;
2243                                    ;THIS ROUTINE WILL CONSTRUCT AND SAVE THE EXPECTED
2244                                    ;CRC AND LRC CHARACTERS ACCORDING TO DATA AND
2245                                    ;RECORD SIZE IF OPERATING IN NRZ MODE
2246                                    ;*************************************************************
2247
2248  015076  016700  163454   CRCLRC: MOV     FMCNT,R0        ;SET RECORD SIZE
2249  015102  005400                   NEG     R0
2250  015104  012701  027454           MOV     #WDATA,R1       ;SET START OF BUFFER
2251  015110  005067  000346           CLR     XORS
2252  015114  111104           CL0:    MOVB    (R1),R4         ;GET CHARACTER
2253  015116  004767  000166           JSR     PC,CLP          ;GO GET PARITY OF CHARACTER
2254  015122  004767  000310           JSR     PC,XOR          ;XOR CHARACTER
2255  015126  000241                   CLC
2256  015130  006004                   ROR     R4              ;ROTATE 1 RIGHT
2257  015132  103014                   BCC     CL2             ;IF NO CARRY: BR
2258  015134  052704  000400           BIS     #400,R4         ;SET BIT NINE
2259  015140  000241                   CLC
2260  015142  010405           CL1:    MOV     R4,R5           ;SAVE CHARACTER
2261  015144  042705  177703           BIC     #177703,R5
2262  015150  005105                   COM     R5
2263  015152  042705  177703           BIC     #177703,R5
2264  015156  042704  000074           BIC     #74,R4
2265  015162  050504                   BIS     R5,R4           ;COMPLIMENT BITS 2,3,4,5
2266  015164  010467  000272   CL2:    MOV     R4,XORS
2267  015170  005300                   DEC     R0
2268  015172  001402                   BEQ     CLLAST          ;IF LAST CHARACTER: BR
2269  015174  000167  177714           JMP     CL0             ;GET NEXT
2270  015200  016704  000256   CLLAST: MOV     XORS,R4
2271  015204  005167  000252           COM     XORS
2272  015210  042767  177050  000244   BIC     #177050,XORS
2273  015216  042704  177727           BIC     #177727,R4      ;COMPLIMENT ALL BUT BITS 3&5
2274  015222  050467  000234           BIS     R4,XORS
2275  015226  016767  000230  000230   MOV     XORS,EXCRC      ;SAVE EXPECTED CRC
2276  015234  016700  163316           MOV     FMCNT,R0
2277  015240  005400                   NEG     R0
2278  015242  012701  027454           MOV     #WDATA,R1       ;DO EXPT LRC
2279  015246  005067  000210           CLR     XORS
2280  015252  111104           CL3:    MOVB    (R1),R4
2281  015254  004767  000030           JSR     PC,CLP          ;GET PARITY
2282  015260  004767  000152           JSR     PC,XOR          ;XOR CHARACTER
2283  015264  005300                   DEC     R0
2284  015266  001371                   BNE     CL3             ;DO ALL FOR LRC
2285  015270  016704  000170           MOV     EXCRC,R4
2286  015274  004767  000136           JSR     PC,XOR          ;XOR CRC TO DATA
2287  015300  016767  000156  000160   MOV     XORS,EXLRC      ;SAVE EXPT LRC
2288  015306  000207                   RTS     PC              ;RETURN
2289  015310  005704           CLP:    TST     R4              ;SEE IF 0 CHAR
2290  015312  001010                   BNE     CLPE            ;IF NOT: BR
2291  015314  032767  000010  163230   BIT     #10,UDES        ;SEE IF EVEN PARITY
2292  015322  001404                   BEQ     CLPE            ;IF NOT: BR
2293  015324  012704  000420           MOV     #420,R4         ;SET 0 CHAR EVEN PARITY
2294  015330  005201                   INC     R1              ;BUMP POINTER
```

```
2295  015332  000207                          RTS     PC              ;RETURN
2296  015334  005067  000130          CLPE:   CLR     PARCNT          ;CLEAR BIT COUNTER
2297  015340  012703  000010                  MOV     #10,R3          ;SET NUMBER OF BITS
2298  015344  032704  000001          CLPO:   BIT     #1,R4           ;SEE IF ONE BIT
2299  015350  001402                          BEQ     CLP1            ;IF NOT: BR
2300  015352  005267  000112                  INC     PARCNT          ;BUMP COUNTER
2301  015356  000241                  CLP1:   CLC
2302  015360  006004                          ROR     R4              ;ROTATE TO NEXT BIT
2303  015362  005303                          DEC     R3
2304  015364  001367                          BNE     CLPO            ;CONTINUE FOR ALL BITS
2305  015366  112104                          MOVB    (R1)+,R4
2306  015370  042704  177400                  BIC     #177400,R4
2307  015374  032767  000001  000066          BIT     #1,PARCNT       ;SEE IF ODD NUMBER OF ONE BITS
2308  015402  001005                          BNE     CLP2            ;IF SO: BR
2309  015404  032767  000010  163140          BIT     #10,UDES        ;SEE IF SHOULD BE EVEN PARITY
2310  015412  001406                          BEQ     CLP3            ;IF NOT: BR
2311  015414  000207                          RTS     PC              ;ELSE EXIT
2312  015416  032767  000010  163126  CLP2:   BIT     #10,UDES        ;SEE IF SHOULD BE ODD PARITY
2313  015424  001001                          BNE     CLP3            ;IF NOT: BR
2314  015426  000207                          RTS     PC              ;ELSE EXIT
2315  015430  052704  000400          CLP3:   BIS     #400,R4         ;SET PARITY BIT
2316  015434  000207                          RTS     PC
2317  015436  010446                  XOR:    MOV     R4,-(SP)
2318  015440  046716  000016                  BIC     XORS,(SP)
2319  015444  040467  000012                  BIC     R4,XORS         ;XOR SUBROUTINE: R4 WITH XORS
2320  015450  052667  000006                  BIS     (SP)+,XORS
2321  015454  016704  000002                  MOV     XORS,R4
2322  015460  000207                          RTS     PC
2323
2324  015462  000000          XORS:   0               ;XOR SAVE
2325  015464  000000          EXCRC:  0               ;EXPECTED CRC
2326  015466  000000          EXLRC:  0               ;EXPECTED LRC
2327  015470  000000          PARCNT: 0               ;PARITY COUNTER
2328
```

```
2329
2330                                    ;*********************************************************
2331                                    ;DATA CHECK SUBROUTINE:
2332                                    ;
2333                                    ;THIS SUBROUTINE IS USED TO COMPARE EACH CHARACTER
2334                                    ;OF DATA READ FROM TAPE WITH THE EXPECTED CHARACTER.
2335                                    ;ANY ERROR DETECTED WILL CAUSE CONTROL TO BE
2336                                    ;PASSED TO AN ERROR PRINT SUBROUTINE AND A
2337                                    ;SUBROUTINE TO ACCUMULATE THE NUMBER OF BITS
2338                                    ;DROPPED AND PICKED UP FROM EACH CHARACTER.
2339                                    ;THE NUMBER OF READ ERRORS IS ALSO ACCUMULATED.
2340                                    ;DATA CHECKING MAY BE TERMINATED BY USE OF
2341                                    ;CONSOLE SWITCH THIRTEEN (13).
2342                                    ;*********************************************************
2343
2344  015472  005067  163160        DCHK:   CLR     BBC             ;CLEAR BAD RECORD CNTR
2345  015476  005067  163202                CLR     DERFL           ;CLEAR DATA ERROR FLAG
2346  015502  016705  163050                MOV     FMCNT,R5        ;LOAD CHAR COUNT
2347  015506  032767  000020  163036        BIT     #20,UDES        ;SEE IF CORE DUMP
2348  015514  001402                        BEQ     DCHKO           ;IF NOT: BR
2349  015516  000261                        SEC
2350  015520  006005                        ROR     R5              ;R5 = FC/2
2351  015522  012701  027454        DCHKO:  MOV     #WDATA,R1       ;SET WRITE DATA ADDR
2352  015526  012702  033462                MOV     #RDATA,R2       ;SET READ DATA ADDR
2353  015532  032767  000010  163012        BIT     #10,UDES        ;SEE IF EVEN PARITY
2354  015540  001430                        BEQ     DFOCO           ;IF NOT: BR
2355  015542  032767  000020  163002        BIT     #20,UDES        ;SEE IF CORE DUMP PARITY
2356  015550  001024                        BNE     DFOCO           ;IF SO: BR
2357  015552  032767  002000  162772        BIT     #2000,UDES      ;SEE IF PE MODE
2358  015560  001020                        BNE     DFOCO           ;IF SO: BR
2359  015562  105711                DFOF:   TSTB    (R1)            ;SEE IF 0 CHAR
2360  015564  001404                        BEQ     DFOD            ;IF SO: BR
2361  015566  005201                        INC     R1              ;BUMP POINTER
2362  015570  005205                DFOE:   INC     R5              ;SEE IF DONE
2363  015572  001373                        BNE     DFOF            ;IF NOT: BR
2364  015574  000406                        BR      DFOC            ;ELSE CONTINUE
2365  015576  112721  000020        DFOD:   MOVB    #20,(R1)+       ;SET 20 IN PLACE OF 0
2366  015602  012767  177777  176464        MOV     #-1,PATS        ;SET PATTERN GENERATE FLAG
2367  015610  000767                        BR      DFOE
2368  015612  016705  162740        DFOC:   MOV     FMCNT,R5        ;RESET CHAR CNT
2369  015616  012701  027454                MOV     #WDATA,R1       ;RESET DATA ADDRESS
2370  015622  032767  010000  162732  DFOCO: BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2371  015630  001462                        BEQ     DFO             ;IF NOT: BR
2372  015632  016704  162720        DFOB:   MOV     FMCNT,R4        ;GET FRAME COUNT
2373  015636  005404                        NEG     R4              ;SET TO WHOLE NUMBER
2374  015640  032767  000020  162704        BIT     #20,UDES        ;SEE IF CORE DUMP
2375  015646  001402                        BEQ     DFOBO           ;IF NOT: BR
2376  015650  000241                        CLC
2377  015652  006004                        ROR     R4              ;SET TO FC/2
2378  015654  060401                DFOBO:  ADD     R4,R1           ;POINT TO START OF WRITE DATA
2379  015656  060402                        ADD     R4,R2           ;POINT TO START OF READ DATA
2380  015660  032767  000001  162670        BIT     #1,FMCNT        ;SEE IF ODD FRAME COUNT
2381  015666  001401                        BEQ     DFOA            ;IF NOT: BR
2382  015670  105722                        TSTB    (R2)+           ;BUMP POINTER
2383  015672  032767  000020  162652  DFOA:  BIT     #20,UDES        ;SEE IF CORE DUMP
2384  015700  001431                        BEQ     DFOA4           ;IF NOT: BR
```

```
2385  015702  000241                          CLC
2386  015704  132742  000001                  BITB    #1,-(R2)        ;SEE IF BIT 0 = 1
2387  015710  001401                          BEQ     DFOA0           ;IF NOT: BR
2388  015712  000261                          SEC
2389  015714  106012                  DFOA0:  RORB    (R2)
2390  015716  000241                          CLC
2391  015720  132712  000001                  BITB    #1,(R2)
2392  015724  001401                          BEQ     DFOA1
2393  015726  000261                          SEC
2394  015730  106012                  DFOA1:  RORB    (R2)            ;POSITION BITS FOR REVERSE CORE DUMP
2395  015732  000241                          CLC
2396  015734  132712  000001                  BITB    #1,(R2)
2397  015740  001401                          BEQ     DFOA2
2398  015742  000261                          SEC
2399  015744  106012                  DFOA2:  RORB    (R2)
2400  015746  000241                          CLC
2401  015750  132712  000001                  BITB    #1,(R2)
2402  015754  001401                          BEQ     DFOA3
2403  015756  000261                          SEC
2404  015760  106012                  DFOA3:  RORB    (R2)
2405  015762  005202                          INC     R2              ;RESET POINTER
2406  015764  124142                  DFOA4:  CMPB    -(R1),-(R2)     ;TEST DATA CHARACTER
2407  015766  001010                          BNE     DF1             ;IF NOT GOOD: BR
2408  015770  105067  162662                  CLRB    BBC             ;CLEAR BAD RECORD COUNTER
2409  015774  000411                          BR      DF2
2410  015776  122122                  DF0:    CMPB    (R1)+,(R2)+     ;CHECK DATA
2411  016000  001003                          BNE     DF1             ;IF BAD: BR
2412  016002  105067  162650                  CLRB    BBC             ;CLEAR BAD RECORD CNTR
2413  016006  000404                          BR      DF2
2414  016010  004767  000616          DF1:    JSR     PC,DRPKF        ;GO GET DROPS AND PICKS
2415  016014  004767  000066                  JSR     PC,DERR         ;GO DO PRINT
2416  016020  005205                  DF2:    INC     R5              ;BUMP CHAR CNTR
2417  016022  001405                          BEQ     DF3             ;IF DONE ALL: BR
2418  016024  032767  010000  162530          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2419  016032  001761                          BEQ     DF0             ;IF NOT: BR
2420  016034  000716                          BR      DFOA            ;ELSE CONTINUE READ REV
2421  016036  005067  162622          DF3:    CLR     HDRFL           ;CLEAR HEADER FLAG
2422  016042  005767  162636                  TST     DERFL           ;SEE IF HAD DATA ERROR
2423  016046  001416                          BEQ     DFX             ;IF NOT: BR
2424  016050  005767  162632                  TST     SERFL
2425  016054  001013                          BNE     DFX             ;IF NOT DATA ERROR ONLY: BR
2426  016056  016704  162612                  MOV     UNP,R4
2427  016062  032767  010000  162472          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2428  016070  001003                          BNE     DF4             ;IF SO: BR
2429  016072  005264  001130                  INC     DATER1(R4)      ;BUMP DATA ERROR FORWARD COUNTER
2430  016076  000402                          BR      DFX
2431  016100  005264  001170          DF4:    INC     DEREV1(R4)      ;BUMP REVERSE DATA ERROR
2432  016104  000207                  DFX:    RTS     PC              ;EXIT
2433
```

```
2434
2435                    ;*********************************************************
2436                    ;DATA ERROR SUBROUTINE:
2437                    ;
2438                    ;THIS SUBROUTINE IS USED TO PRINT OUT ANY
2439                    ;ERRORS FOUND DURING THE DATA CHECK.
2440                    ;EACH CHARACTER FOUND BAD WILL BE PRINTED
2441                    ;IN BIT FORMAT ALONG WITH ITS EXPECTED CHARACTER.
2442                    ;AN ERROR HEADER CONSISTING OF THE UNIT NUMBER,
2443                    ;BLOCK NUMBER, RECORD NUMBER, SIZE OF RECORD, AND
2444                    ;ERROR TYPE (READ FORWARD, READ REVERSE, WRITE, ETC)
2445                    ;IS PRINTED ONLY ONCE FOR EACH RECORD FOUND BAD.
2446                    ;A COUNT IS MADE OF THE NUMBER OF SUCCESSIVE BAD
2447                    ;CHARACTERS, AND IF TEN (10) SUCCESSIVE BAD CHARACTERS
2448                    ;ARE FOUND IN A SINGLE RECORD, A MESSAGE INDICATING
2449                    ;A BAD RECORD CONDITION IS PRINTED AND THE NEXT
2450                    ;TWENTY (20) CHARACTERS ARE SKIPPED BEFORE CHECKING
2451                    ;IS RESUMED.  IF THE BAD RECORD CONDITION IS FOUND
2452                    ;THREE TIMES IN A RECORD, ALL REMAINING DATA IS
2453                    ;SKIPPED EXCEPT THE FINAL TEN (10) CHARACTERS.
2454                    ;THIS SKIPPING IS OF COURSE ONLY POSSIBLE IN
2455                    ;RECORDS WHICH CONTAIN A SUFFICIENT NUMBER OF CHARACTERS.
2456                    ;PRINTING OF ERRORS MAY BE DISALLOWED AT ANY TIME
2457                    ;BY SETTING CONSOLE SWITCH TEN (10) TO A ONE.
2458                    ;THE OPERATOR MAY CAUSE THE PROGRAM TO HALT ON ANY ERROR
2459                    ;BY SETTING CONSOLE SWITCH FIFTEEN (15) TO A ONE.
2460                    ;*********************************************************
2461
2462  016106  032777  002000  162472  DERR:   BIT     #2000,@SWR      ;SEE IF SHOULD PRINT ERRORS
2463  016114  001402                    BEQ     DERRO           ;IF SO: BR
2464  016116  000167  000160            JMP     DERR4           ;ELSE SKIP PRINT
2465  016122  005267  162542    DERRO:  INC     PFLG            ;SET PRINT FLAG
2466  016126  005767  162532            TST     HDRFL           ;SEE IF HAVE PRINTED HEADER
2467  016132  001007                    BNE     DERROA          ;IF SO: BR
2468  016134  004767  004522            JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
2469  016140  012704  024716            MOV     #MSG1,R4        ;LOAD ERROR MSG ADDR
2470  016144  104000                    TTOUTT                  ;PRINT ERROR
2471  016146  004767  002762            JSR     PC,FRPRT        ;PRINT F OR R
2472  016152  012704  024735    DERROA: MOV     #MSG4,R4
2473  016156  104000                    TTOUTT                  ;PRINT CHAR NO. HEADER
2474  016160  010203                    MOV     R2,R3
2475  016162  162703  033462            SUB     #RDATA,R3       ;POINT TO CHAR
2476  016166  005303                    DEC     R3
2477  016170  032767  010000  162364    BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2478  016176  001402                    BEQ     DERROB          ;IF NOT: BR
2479  016200  010503                    MOV     R5,R3           ;GET CHAR NUMBER
2480  016202  005103                    COM     R3
2481  016204  104002    DERROB: OCTPP                           ;PRINT CHAR NUMBER
2482  016206  012704  024723            MOV     #MSG2,R4
2483  016212  104000                    TTOUTT                  ;PRINT EXPECTED DATA
2484  016214  032767  010000  162340    BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2485  016222  001402                    BEQ     DERROC          ;IF NOT: BR
2486  016224  111103                    MOVB    (R1),R3         ;GET CHAR
2487  016226  000401                    BR      DERROD
2488  016230  114103    DERROC: MOVB    -(R1),R3        ;LOAD EXPECTED DATA
2489  016232  004767  005704    DERROD: JSR     PC,DOUT         ;GO PRINT CHAR
```

```
2490  016236  012704  024730                  MOV     #MSG3,R4
2491  016242  104000                          TTOUTT                  ;PRINT RECIEVED DATA
2492  016244  032767  010000  162310          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2493  016252  001402                          BEQ     DERR1           ;IF NOT: BR
2494  016254  111203                          MOVB    (R2),R3         ;GET CHAR
2495  016256  000401                          BR      DERR2
2496  016260  114203                  DERR1:  MOVB    -(R2),R3
2497  016262  004767  005654          DERR2:  JSR     PC,DOUT         ;PRINT BAD CHAR
2498  016266  032767  010000  162266          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2499  016274  001401                          BEQ     DERR3           ;IF SO: BR
2500  016276  000401                          BR      DERR4
2501  016300  122122                  DERR3:  CMPB    (R1)+,(R2)+     ;RESET POINTERS
2502  016302  105267  162350          DERR4:  INCB    BBC             ;BUMP BAD RECORD CNTR
2503  016306  122767  000010  162342          CMPB    #10,BBC         ;SEE IF BLD BTH
2504  016314  001123                          BNE     DEREX           ;IF NOT: BR
2505  016316  032777  002000  162262          BIT     #2000,@SWR      ;SEE IF PRINT INHIBIT
2506  016324  001003                          BNE     1$              ;IF SO: BR
2507  016326  012704  025047                  MOV     #MSG15,R4
2508  016332  104000                          TTOUTT                  ;PRINT BLD BTH
2509  016334  105067  162316          1$:     CLRB    BBC             ;RESET BAD RECORD CNTR
2510  016340  000367  162312                  SWAB    BBC             ;POSITION BLD BTH AMOUNT
2511  016344  105267  162306                  INCB    BBC             ;BUMP AMOUNT
2512  016350  122767  000003  162300          CMPB    #3,BBC          ;SEE IF HAD 3 BLD BTHS
2513  016356  101054                          BHI     DERR4B          ;IF NOT: BR
2514  016360  000367  162272                  SWAB    BBC             ;REPOSITION BBC
2515  016364  022705  177767                  CMP     #177767,R5      ;SEE IF ON LAST EIGHT CHARS
2516  016370  101473                          BLOS    DERR6           ;IF SO: BR
2517  016372  012705  177767                  MOV     #177767,R5      ;SET CHAR CNTR TO 8
2518  016376  032767  010000  162156          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2519  016404  001416                          BEQ     DERR4A          ;IF NOT: BR
2520  016406  012701  027454                  MOV     #WDATA,R1       ;GET START OF BUFFER
2521  016412  012702  033462                  MOV     #RDATA,R2       ;GET START OF BUFFER
2522  016416  062701  000010                  ADD     #10,R1
2523  016422  062702  000010                  ADD     #10,R2          ;POINT TO START +10
2524  016426  032767  000001  162122          BIT     #1,FMCNT        ;SEE IF ODD FRAME COUNT
2525  016434  001+53                          BEQ     DEREX           ;IF NOT: BR
2526  016436  105722                          TSTB    (R2)+           ;BUMP POINTER
2527  016440  000451                          BR      DEREX
2528  016442  016767  162110  162172  DERR4A: MOV     FMCNT,TEMP1     ;LOAD CHAR COUNT
2529  016450  005167  162166                  COM     TEMP1
2530  016454  005267  162162                  INC     TEMP1
2531  016460  162767  000010  162154          SUB     #10,TEMP1       ;POINT TO BUFFER -8
2532  016466  016701  162150                  MOV     TEMP1,R1        ;POINT TO NEXT CHAR
2533  016472  062701  027454                  ADD     #WDATA,R1       ;POINT TO NEXT WRITE CHAR
2534  016476  016702  162140                  MOV     TEMP1,R2        ;POINT TO END OF READ DATA -8 FORWARD
2535  016502  062702  033462                  ADD     #RDATA,R2       ;POINT TO NEXT CHAR
2536  016506  000426                          BR      DEREX           ;EXIT
2537  016510  000367  162142          DERR4B: SWAB    BBC             ;REPOSITION BBC
2538  016514  000241                          CLC
2539  016516  062705  000024                  ADD     #24,R5          ;SKIP 20 CHARS
2540  016522  103416                          BCS     DERR6           ;IF EXCEED RECORD SIZE: BR
2541  016524  032767  010000  162030          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2542  016532  001405                          BEQ     DERR5           ;IF NOT: BR
2543  016534  162701  000024                  SUB     #24,R1
2544  016540  162702  000024                  SUB     #24,R2          ;RESET POINTERS
2545  016544  000407                          BR      DEREX
```

```
2546  016546  062701  000024          DERR5:  ADD     #24,R1          ;SKIP 20 CHARS
2547  016552  062702  000024                  ADD     #24,R2          ;SKIP FORWARD 20 CHARS
2548  016556  000402                  BR      DEREX
2549  016560  012705  177777          DERR6:  MOV     #-1,R5          ;SET TO EOR
2550  016564  032777  100000  162014  DEREX:  BIT     #100000,@SWR    ;SEE IF SHOULD HALT ON ERROR
2551  016572  001412                  BEQ     DEREX1          ;IF NOT: BR
2552  016574  104006                  STOPP
2553  016576  005767  162066                  TST     PFLG            ;SEE IF PRINTED
2554  016602  001006                  BNE     DEREX1          ;IF SO: BR
2555  016604  032777  002000  161774          BIT     #2000,@SWR      ;SEE IF SHOULD PRINT
2556  016612  001002                  BNE     DEREX1          ;IF NOT: BR
2557  016614  000167  177302                  JMP     DERR0           ;ELSE PRINT
2558  016620  005067  162044          DEREX1: CLR     PFLG            ;CLEAR FLAG
2559  016624  005267  162054                  INC     DERFL           ;BUMP DATA ERROR FLAG
2560  016630  000207                  RTS     PC              ;RETURN
2561
```

```
2562
2563                                    ;*********************************************************
2564                                    ;DROPS AND PICKS SUBROUTINE:
2565                                    ;
2566                                    ;THIS SUBROUTINE IS USED TO ACCUMULATE FROM
2567                                    ;EACH BAD DATA CHARACTER FOUND THE NUMBER
2568                                    ;OF BITS WHICH WERE EITHER DROPPED OR PICKED UP.
2569                                    ;TWO COUNTERS PER SLAVE ARE USED TO ACCUMULATE THIS
2570                                    ;INFORMATION AND CAN STORE UP TO 32K DROPS
2571                                    ;OR PICKS BEFORE OVERFLOWING.  IF OVERFLOW IS
2572                                    ;ABOUT TO OCCUR, THESE ACCUMULATORS ARE
2573                                    ;PRINTED IN OCTAL AND RESET TO ZERO.
2574                                    ;THE CONTENTS OF THE ACCUMULATORS MAY BE
2575                                    ;DISPLAYED AT ANY TIME BY SETTING CONSOLE
2576                                    ;SWITCH FOURTEEN TO A ONE (1). THE PRINTOUT WILL OCCUR
2577                                    ;AT THE END OF THE CURRENT BLOCK CYCLE.
2578                                    ;*********************************************************
2579
2580  016632  005067  162004           DRPKF:  CLR     TEMP1
2581  016636  005067  162002                   CLR     TEMP2
2582  016642  005067  162000                   CLR     TEMP3
2583  016646  111167  161770                   MOVB    (R1),TEMP1      ;LOAD GOOD CHAR
2584  016652  111267  161766                   MOVB    (R2),TEMP2      ;LOAD BAD CHAR
2585  016656  016704  162012                   MOV     UNP,R4
2586  016662  016467  000770  162030           MOV     PIK1(R4),BPKP
2587  016670  016467  001010  162020           MOV     DRP1(R4),BDPP
2588  016676  032767  010000  161656           BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2589  016704  001005                           BNE     DRPK            ;IF SO: BR
2590  016706  124142                           CMPB    -(R1),-(R2)     ;POINT TO CHAR
2591  016710  112167  161726                   MOVB    (R1)+,TEMP1     ;LOAD GOOD CHAR
2592  016714  112267  161724                   MOVB    (R2)+,TEMP2     ;LOAD BAD CHAR
2593  016720  004767  000006           DRPK:   JSR     PC,DROP         ;GET DROPS
2594  016724  004767  000222                   JSR     PC,PICK         ;GET PICKS
2595  016730  000207                           RTS     PC              ;EXIT
2596  016732  116703  161704           DROP:   MOVB    TEMP1,R3        ;R3 = GOOD CHAR
2597  016736  116704  161702                   MOVB    TEMP2,R4        ;R4 = BAD CHAR
2598  016742  140403                   DPC:    BICB    R4,R3           ;GET DROPS/PICKS
2599  016744  001001                           BNE     DPCG            ;IF SOME: BR
2600  016746  000207                           RTS     PC              ;RETURN
2601  016750  012767  000010  161732   DPCG:   MOV     #10,BCNT        ;SET NUMBER TO CHECK
2602  016756  132703  000001           DPC0:   BITB    #1,R3           ;SEE IF DROPPED OR PICKED THIS BIT
2603  016762  001455                           BEQ     DPC2            ;IF NOT: BR
2604  016764  105767  161656                   TSTB    TEMP3           ;SEE IF ON PICKS
2605  016770  001016                           BNE     DPC1            ;IF SO: BR
2606  016772  005277  161720                   INC     @BDPP           ;BUMP DROP CNTR
2607  016776  005777  161714                   TST     @BDPP
2608  017002  100045                           BPL     DPC2            ;IF NO OVERFLOW: BR
2609  017004  032777  002000  161574           BIT     #2000,@SWR      ;SEE IF HAVE PRINTED DATA
2610  017012  001402                           BEQ     DPC0A           ;IF SO: BR
2611  017014  004767  003642                   JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
2612  017020  004767  000172           DPC0A:  JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
2613  017024  000415                           BR      DPC2A
2614  017026  005277  161666           DPC1:   INC     @BPKP           ;BUMP PICK CNTR
2615  017032  005777  161662                   TST     @BPKP           ;SEE IF OVERFLOW
2616  017036  100027                           BPL     DPC2            ;IF NOT: BR
2617  017040  032777  002000  161540           BIT     #2000,@SWR      ;SEE IF HAVE PRINTED DATA
```

```
 2618  017046  001402                          BEQ     DPC1A           ;IF SO: BR
 2619  017050  004767  003606                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
 2620  017054  004767  000136          DPC1A:  JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
 2621  017060  016704  161610          DPC2A:  MOV     UNP,R4
 2622  017064  016403  001010                  MOV     DRP1(R4),R3     ;SET DROP POINTER
 2623  017070  016404  000770                  MOV     PIK1(R4),R4     ;SET PICK POINTER
 2624  017074  012767  000010  161606          MOV     #10,BCNT        ;SET NUMBER OF BITS
 2625  017102  005023                  DPC2B:  CLR     (R3)+           ;CLEAR DROPS
 2626  017104  005024                          CLR     (R4)+           ;CLEAR PICK
 2627  017106  005367  161576                  DEC     BCNT            ;SEE IF DONE
 2628  017112  001373                          BNE     DPC2B           ;IF NOT: BR
 2629  017114  000207                          RTS     PC              ;EXIT
 2630  017116  000241                  DPC2:   CLC
 2631  017120  106003                          RORB    R3              ;GET NEXT BIT
 2632  017122  005367  161562                  DEC     BCNT            ;SEE IF DONE
 2633  017126  001410                          BEQ     DPC3
 2634  017130  062767  000002  161562          ADD     #2,BPKP
 2635  017136  062767  000002  161552          ADD     #2,BDPP
 2636  017144  000167  177606                  JMP     DPC0            ;CONTINUE
 2637  017150  000207                  DPC3:   RTS     PC              ;RETURN
 2638  017152  016704  161516          PICK:   MOV     UNP,R4          ;GET UNIT POINTER
 2639  017156  016467  000770  161534          MOV     PIK1(R4),BPKP   ;SET PICK POINTER
 2640  017164  016467  001010  161524          MOV     DRP1(R4),BDPP   ;SET DROP POINTER
 2641  017172  116704  161444                  MOVB    TEMP1,R4        ;R4 = GOOD CHAR
 2642  017176  116703  161442                  MOVB    TEMP2,R3        ;R3 = BAD CHAR
 2643  017202  112767  000001  161436          MOVB    #1,TEMP3        ;SET PICK FLAG
 2644  017210  004767  177526                  JSR     PC,DPC          ;GO CHECK PICKS
 2645  017214  000207                          RTS     PC              ;EXIT
 2646  017216  012704  025343          DPPRT:  MOV     #MSG26,R4
 2647  017222  104000                          TTOUTT                  ;PRINT DROP HEADER
 2648  017224  016704  161444                  MOV     UNP,R4
 2649  017230  016467  001010  161460          MOV     DRP1(R4),BDPP   ;SET DROP POINTER
 2650  017236  016467  000770  161454          MOV     PIK1(R4),BPKP   ;SET PICK POINTER
 2651  017244  062767  000016  161444          ADD     #16,BDPP
 2652  017252  062767  000016  161440          ADD     #16,BPKP
 2653  017260  012767  000010  161422          MOV     #10,BCNT        ;SET NUMBER TO PRINT
 2654  017266  017703  161424          DPPRT0: MOV     @BDPP,R3
 2655  017272  104002                          OCTPP                   ;PRINT DROPS
 2656  017274  005367  161410                  DEC     BCNT            ;SEE IF DONE
 2657  017300  001404                          BEQ     DPPRT1          ;IF NOT: BR
 2658  017302  162767  000002  161406          SUB     #2,BDPP         ;BUMP POINTER
 2659  017310  000766                          BR      DPPRT0          ;CONTINUE FOR ALL 8 BITS
 2660  017312  012767  000010  161370  DPPRT1: MOV     #10,BCNT        ;SET NUMBER TO PRINT
 2661  017320  012704  025354                  MOV     #MSG27,R4
 2662  017324  104000                          TTOUTT                  ;PRINT PICK HEADER
 2663  017326  017703  161366          DPPRT2: MOV     @BPKP,R3
 2664  017332  104002                          OCTPP                   ;PRINT PICKS
 2665  017334  005367  161350                  DEC     BCNT            ;SEE IF DONE
 2666  017340  001404                          BEQ     DPPRTX          ;IF SO: BR
 2667  017342  162767  000002  161350          SUB     #2,BPKP         ;BUMP POINTER
 2668  017350  000766                          BR      DPPRT2          ;CONTINUE FOR ALL 8 BITS
 2669  017352  000207                  DPPRTX: RTS     PC              ;RETURN
```

```
                                ;*******************************************************
                                ;STATUS CHECK SUBROUTINE:
                                ;
                                ;THIS SUBROUTINE IS USED TO PERFORM A CHECK OF
                                ;BOTH THE MASSBUS CONTROLLER (RH11) AND THE TAPE
                                ;CONTROLLER (TM02).  THE RH11 IS CHECKED FOR ERRORS
                                ;AS REFLECTED IN REGISTERS CS1 AND CS2 AND ALSO THAT
                                ;THE BUS ADDRESS (BA) AND WORD COUNT (WC) ARE
                                ;CORRECT.  THE TM02 IS CHECKED FOR DRIVE STATIS (DS),
                                ;DRIVE ERRORS (ER), AND PROPER FRAME COUNT.  THE SPECIAL
                                ;CHECK CHARACTERS (CRC+LRC) ARE ALSO CHECKED WHEN
                                ;APPROPIATE (IE: NRZ READ OR WRITE).  CERTAIN TYPES
                                ;OF DRIVE ERRORS IN PE OPERATION WILL BE ACCOMPANIED
                                ;BY THE DISPLAY OF THE DEAD TRACK REGISTER (CC).  THESE
                                ;TYPES ARE ER BITS 15,10,7,6.  THE PRINTOUTS OF BAD
                                ;CRC,LRC,FC, AND BA WILL SHOW BOTH THE EXPECTED AND
                                ;RECEIVED VALUES (IE: EXPT-RCVD).  ONLY THOSE REGISTERS
                                ;WHICH ARE IN ERROR WILL BE PRINTED AND ALL PRINTOUTS
                                ;ARE IN OCTAL FORMAT WITH NO LEADING ZEROS.  AS IN
                                ;DATA ERRORS, STATUS ERRORS ARE PRECEEDED BY HEADER
                                ;DESCRIBING THE HARDWARE UNDER TEST, THE BLOCKING
                                ;INFORMATION, AND THE ERROR TYPE.
                                ;*******************************************************

017354  016703  161176        ERCHK:  MOV     FMCNT,R3        ;GET FRAME COUNT
017360  032703  000001                BIT     #1,R3           ;SEE IF ODD
017364  001401                       BEQ     ERO             ;IF NOT: BR
017366  005303                       DEC     R3              ;BUMP COUNT
017370  005403                ERO:   NEG     R3
017372  032767  000020  161152       BIT     #20,UDES        ;SEE IF CORE DUMP
017400  001402                       BEQ     EROB            ;IF NOT: BR
017402  000241                       CLC
017404  006003                       ROR     R3              ;SET TO FC/2
017406  032767  000010  161256 EROB:  BIT     #10,MTC1        ;SEE IF WRITE OP
017414  001414                       BEQ     ER1             ;IF SO: BR
017416  032767  010000  161136       BIT     #10000,RDCMD
017424  001405                       BEQ     EROA
017426  012703  033462               MOV     #RDATA,R3
017432  162703  000002               SUB     #2,R3           ;SET POINTER
017436  000405                       BR      ER2
017440  062703  033462        EROA:  ADD     #RDATA,R3       ;BUILD EXPT READ ADDRESS
017444  000402                       BR      ER2
017446  062703  027454        ER1:   ADD     #WDATA,R3       ;BUILD EXPT WRITE ADDRESS
017452  010367  001432        ER2:   MOV     R3,CADER        ;SAVE ADDRESS
017456  012704  000007               MOV     #7,R4
017462  012701  021112               MOV     #BAER,R1
017466  005021                ER2A0: CLR     (R1)+           ;CLEAR FLAGS
017470  005304                       DEC     R4
017472  001375                       BNE     ER2A0
017474  020377  161014               CMP     R3,@BA          ;SEE IF ADDRESS OK
017500  001402                       BEQ     ER2A1           ;IF SO: BR
017502  005267  001404               INC     BAER            ;SET BUS ADDRESS ERROR
017506  032767  000010  161156 ER2A1: BIT     #10,MTC1        ;SEE IF WRITE OPER
017514  001007                       BNE     ER2B            ;IF NOT: BR
017516  005777  160774        ER2A:  TST     @FC             ;SEE IF FC=0
```

```
2726  017522  001443                        BEQ     ER3             ;IF SO: BR
2727  017524  005267  001370                INC     FCER            ;SET FC ERROR
2728  017530  000167  000076                JMP     ER3
2729  017534  032767  000040  161130  ER2B: BIT     #40,MTC1        ;SEE IF SPACE OPER
2730  017542  001765                        BEQ     ER2A            ;IF SO: BR
2731  017544  005767  161126                TST     TMFLG           ;SEE IF TM TIME
2732  017550  001012                        BNE     ER2D            ;IF SO: BR
2733  017552  016703  161000                MOV     FMCNT,R3
2734  017556  005403                        NEG     R3              ;R3 = EXPT RECORD SIZE
2735  017560  020377  160732          ER2C: CMP     R3,@FC          ;SEE IF FC = EXPT
2736  017564  001422                        BEQ     ER3             ;IF SO: BR
2737  017566  005267  001326                INC     FCER            ;SET FC ERROR FLAG
2738  017572  000167  000034                JMP     ER3
2739  017576  032767  002000  160746  ER2D: BIT     #2000,UDES      ;SEE IF PE
2740  017604  001344                        BNE     ER2A            ;IF SO: BR
2741  017606  032767  010000  160746        BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2742  017614  001003                        BNE     ER2E            ;IF SO: BR
2743  017616  012703  000002                MOV     #2,R3
2744  017622  000756                        BR      ER2C            ;LOOK FOR EXPT = 2
2745  017624  012703  000001          ER2E: MOV     #1,R3
2746  017630  000753                        BR      ER2C            ;GO CHECK FC FOR TM
2747  017632  032777  160000  160650  ER3:  BIT     #160000,@C1     ;SEE IF COUNT ERROR
2748  017640  001442                        BEQ     ER4
2749  017642  017703  160652                MOV     @CS,R3          ;GET CONT STATUS REG
2750  017646  042703  000307                BIC     #307,R3         ;MASK OUT IR,OR,UNIT NO.
2751  017652  005703                        TST     R3              ;SEE IF ANY OTHER ERRORS
2752  017654  001407                        BEQ     ER3A            ;IF NOT: BR
2753  017656  005767  161014                TST     TMFLG           ;SEE IF TAPE MARK TIME
2754  017662  001427                        BEQ     ER3B            ;IF NOT: BR
2755  017664  042703  001000                BIC     #1000,R3        ;MASK MISSED TRANS
2756  017670  005703                        TST     R3              ;SEE IF ANY OTHER ERRORS
2757  017672  001023                        BNE     ER3B            ;IF SO: BR
2758  017674  032777  060000  160606  ER3A: BIT     #60000,@C1      ;SEE IF EITHER TRE OR MCPE
2759  017702  001421                        BEQ     ER4             ;IF NOT: BR
2760  017704  005767  160766                TST     TMFLG           ;SEE IF TM TIME
2761  017710  001414                        BEQ     ER3B            ;IF NOT: BR
2762  017712  017703  160606                MOV     @ER,R3          ;GET ERROR REGISTER
2763  017716  032767  000010  160626        BIT     #10,UDES        ;SEE IF EVEN PARITY
2764  017724  001402                        BEQ     ER3A1           ;IF NOT: BR
2765  017726  042703  000100                BIC     #100,R3         ;MASK PAR
2766  017732  042703  001000          ER3A1: BIC    #1000,R3        ;MASK FCE
2767  017736  005703                        TST     R3
2768  017740  001402                        BEQ     ER4             ;IF NO ERRORS EXCEPT FCE: BR
2769  017742  005267  001146          ER3B: INC     CONER           ;SET CONT ERROR FLAG
2770  017746  032777  040000  160546  ER4:  BIT     #40000,@DS      ;SEE IF DRIVE ERROR
2771  017754  001421                        BEQ     ER6             ;IF NOT: BR
2772  017756  005767  160714                TST     TMFLG           ;SEE IF TAPE MARK TIME
2773  017762  001414                        BEQ     ER4A            ;IF NOT: BR
2774  017764  017703  160534                MOV     @ER,R3          ;GET ER
2775  017770  032767  000010  160554        BIT     #10,UDES        ;SEE IF EVEN PARITY
2776  017776  001402                        BEQ     ER4A1           ;IF NOT: BR
2777  020000  042703  000100                BIC     #100,R3         ;MASK PAR
2778  020004  042703  001000          ER4A1: BIC    #1000,R3        ;MASK OUT FCE
2779  020010  005703                        TST     R3              ;SEE IF ANY OTHER ERRORS
2780  020012  001402                        BEQ     ER6             ;IF NOT: BR
2781  020014  005267  001076          ER4A: INC     DRVER           ;SET DRIVER ERROR FLAG
```

```
2782  020020  032767  002000  160524  ER6:    BIT     #2000,UDES
2783  020026  001071                  BNE     ERPT            ;IF IN PE MODE: BR
2784  020030  032777  020000  160550          BIT     #20000,@SWR     ;SEE IF NO DATA CHECK
2785  020036  001065                  BNE     ERPT            ;IF NOT: BR (ALLOW READ OF UNKNOWN TAPES)
2786  020040  032767  000040  160624          BIT     #40,MTC1        ;SEE IF WRITE OR READ OP
2787  020046  001461                  BEQ     ERPT            ;IF NOT: BR
2788  020050  005767  160622                  TST     TMFLG           ;SEE IF TAPE MARK TIME
2789  020054  001413                  BEQ     ER6A            ;IF NOT: BR
2790  020056  016767  175402  001046          MOV     EXCRC,CRCSV     ;SAVE CRC
2791  020064  016767  175376  001036          MOV     EXLRC,LRCSV     ;SAVE LRC
2792  020072  005067  175366                  CLR     EXCRC
2793  020076  012767  000023  175362          MOV     #23,EXLRC       ;SET CRC/LRC FOR TM
2794  020104  032767  000060  160440  ER6A:   BIT     #60,UDES        ;SEE IF FORMAT 14
2795  020112  001037                  BNE     ERPT            ;IF NOT: BR
2796  020114  017703  160410                  MOV     @CC,R3          ;GET CRC CHARACTER
2797  020120  042703  177000                  BIC     #177000,R3
2798  020124  026703  175334                  CMP     EXCRC,R3
2799  020130  001402                  BEQ     ER7             ;IF CRC GOOD: BR
2800  020132  005267  000766                  INC     CRCER           ;SET ERROR FLAG
2801  020136  017703  160372          ER7:    MOV     @MR,R3          ;GET LRC
2802  020142  000303                  SWAB    R3
2803  020144  005703                  TST     R3
2804  020146  100002                  BPL     ER10
2805  020150  052703  000400                  BIS     #400,R3
2806  020154  042703  177000          ER10:   BIC     #177000,R3
2807  020160  026703  175302                  CMP     EXLRC,R3
2808  020164  001412                  BEQ     ERPT            ;IF LRC GOOD: BR
2809  020166  010367  000734                  MOV     R3,ACTLRC       ;SAVE ACTUAL LRC
2810  020172  005267  000724                  INC     LRCER           ;SET LRC ERROR FLAG
2811  020176  032767  010000  160356          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2812  020204  001402                  BEQ     ERPT            ;IF NOT: BR
2813  020206  005067  000710                  CLR     LRCER           ;ELSE CLEAR LRC ERROR
2814  020212  012703  000006          ERPT:   MOV     #6,R3
2815  020216  005067  160464                  CLR     SERFL           ;CLEAR ERROR FLAG
2816  020222  005067  160474                  CLR     ERSAV
2817  020226  012704  021112                  MOV     #BAER,R4
2818  020232  005724                  ERPTT:  TST     (R4)+           ;SEE IF ANY ERROR
2819  020234  001004                  BNE     ERPTG           ;IF SO: BR
2820  020236  005303                  DEC     R3
2821  020240  001374                  BNE     ERPTT
2822  020242  000167  000606                  JMP     ERPX1
2823  020246  005267  160434          ERPTG:  INC     SERFL           ;SET ERROR FLAG
2824  020252  017767  160246  160442          MOV     @ER,ERSAV       ;SAVE ERROR REGISTER
2825  020260  032777  002000  160320          BIT     #2000,@SWR      ;SEE IF PRINT
2826  020266  001420                  BEQ     ERPTO           ;IF SO: BR
2827  020270  022767  000002  160414          CMP     #2,RTYFL        ;SEE IF READ RETRY
2828  020276  001006                  BNE     ERPTG1          ;IF NOT: BR
2829  020300  016703  160376                  MOV     RTCNT,R3
2830  020304  005203                  INC     R3              ;BUMP RETRY COUNT
2831  020306  020367  160270                  CMP     R3,RETRY        ;SEE IF LAST RETRY
2832  020312  001406                  BEQ     ERPTO           ;IF SO: BR
2833  020314  022767  000002  000574  ERPTG1: CMP     #2,DRVER        ;SEE IF TM STATUS ERROR
2834  020322  001402                  BEQ     ERPTO           ;IF SO: BR
2835  020324  000167  000426                  JMP     ERPXO
2836  020330  005267  160334          ERPTO:  INC     PFLG
2837  020334  004767  002322                  JSR     PC,PAPRT        ;PRINT HEADER
```

```
2838  020340  016704  160306                MOV     EMADDR,R4
2839  020344  104000                        TTOUTT                  ;PRINT ERROR TYPE
2840  020346  004767  000562                JSR     PC,FRPRT        ;PRINT F OR R
2841  020352  005767  160320                TST     TMFLG
2842  020356  001407                        BEQ     ERPT1
2843  020360  022767  026222  160264        CMP     #MSG54,EMADDR
2844  020366  001403                        BEQ     ERPT1
2845  020370  012704  026240                MOV     #MSG56,R4       ;PRINT TM
2846  020374  104000                        TTOUTT
2847  020376  005767  000512        ERPT1:  TST     CONER
2848  020402  001414                        BEQ     ERPT2           ;IF NO CONT ERROR: BR
2849  020404  012704  025173                MOV     #MSG23,R4
2850  020410  104000                        TTOUTT                  ;PRINT C1 TAG
2851  020412  017703  160072                MOV     @C1,R3
2852  020416  104002                        OCTPP                   ;PRINT CONTROL 1
2853  020420  012704  025220                MOV     #MSG23D,R4      ;PRINT CS TAG
2854  020424  104000                        TTOUTT
2855  020426  017703  160066                MOV     @CS,R3
2856  020432  104002                        OCTPP                   ;PRINT CONT STATUS
2857  020434  005767  000456        ERPT2:  TST     DRVER
2858  020440  001414                        BEQ     ERPT3           ;IF SO DRIVE ERROR: BR
2859  020442  012704  025226                MOV     #MSG23E,R4
2860  020446  104000                        TTOUTT                  ;PRINT DS TAG
2861  020450  017703  160046                MOV     @DS,R3
2862  020454  104002                        OCTPP                   ;PRINT DRIVE STATUS
2863  020456  012704  025233                MOV     #MSG23F,R4
2864  020462  104000                        TTOUTT                  ;PRINT ER TAG
2865  020464  017703  160034                MOV     @ER,R3
2866  020470  104002                        OCTPP                   ;PRINT DRIVE ERROR
2867  020472  005767  000414        ERPT3:  TST     BAER
2868  020476  001416                        BEQ     ERPT4           ;IF NO BA ERROR: BR
2869  020500  012704  025206                MOV     #MSG23B,R4
2870  020504  104000                        TTOUTT                  ;PRINT BA TAG
2871  020506  017703  160002                MOV     @BA,R3
2872  020512  104002                        OCTPP                   ;PRINT BUS ADDRESS
2873  020514  012767  000255  160114        MOV     #255,TOB
2874  020522  004767  003114                JSR     PC,TOG          ;PRINT /
2875  020526  016703  000356                MOV     CADER,R3
2876  020532  104002                        OCTPP                   ;PRINT EXPT BUS ADDRESS
2877  020534  005767  000360        ERPT4:  TST     FCER
2878  020540  001405                        BEQ     ERPT5           ;IF NO FC ERROR: BR
2879  020542  012704  025213                MOV     #MSG23C,R4
2880  020546  104000                        TTOUTT                  ;PRINT FC TAG
2881  020550  017703  157742                MOV     @FC,R3
2882  020554  104002                        OCTPP                   ;PRINT FRAME COUNT
2883  020556  012704  025201        ERPT5:  MOV     #MSG23A,R4
2884  020562  104000                        TTOUTT                  ;PRINT WC TAG
2885  020564  017703  157722                MOV     @WC,R3
2886  020570  104002                        OCTPP                   ;PRINT WORD COUNT
2887  020572  005767  000326                TST     CRCER
2888  020576  001420                        BEQ     ERPT5A          ;IF NO CRC ERROR: BR
2889  020600  012704  026265                MOV     #MSG58,R4
2890  020604  104000                        TTOUTT                  ;PRINT CRC TAG
2891  020606  017703  157716                MOV     @CC,R3
2892  020612  042703  177000                BIC     #177000,R3
2893  020616  104002                        OCTPP                   ;PRINT ACTUAL CRC
```

```
2894  020620  012767  000255  160010          MOV     #255,TOB
2895  020626  004767  003010                  JSR     PC,TOG
2896  020632  016703  174626                  MOV     EXCRC,R3
2897  020636  104002                          OCTPP                   ;PRINT EXPECTED CRC
2898  020640  005767  000256          ERPT5A: TST     LRCER
2899  020644  001416                          BEQ     ERPT6           ;IF NO LRC ERROR: BR
2900  020646  012704  026273                  MOV     #MSG59,R4
2901  020652  104000                          TTOUTT                  ;PRINT LRC TAG
2902  020654  016703  000246                  MOV     ACTLRC,R3
2903  020660  104002                          OCTPP                   ;PRINT ACTUAL LRC
2904  020662  012767  000255  157746          MOV     #255,TOB
2905  020670  004767  002746                  JSR     PC,TOG
2906  020674  016703  174566                  MOV     EXLRC,R3
2907  020700  104002                          OCTPP                   ;PRINT EXPECTED LRC
2908  020702  005767  000210          ERPT6:  TST     DRVER
2909  020706  001422                          BEQ     ERPT7           ;IF NO DRIVE ERROR: BR
2910  020710  032767  002000  157634          BIT     #2000,UDES
2911  020716  001416                          BEQ     ERPT7           ;IF NO PE: BR
2912  020720  017704  157600                  MOV     @ER,R4
2913  020724  042704  075477                  BIC     #75477,R4       ;MASK OUT ALL BUT BITS 15,10,7,6
2914  020730  005704                          TST     R4
2915  020732  001410                          BEQ     ERPT7           ;IF NO CONDITIONALS SET: BR
2916  020734  012704  025245                  MOV     #MSG23H,R4
2917  020740  104000                          TTOUTT                  ;PRINT CC TAG
2918  020742  017703  157562                  MOV     @CC,R3
2919  020746  042703  177000                  BIC     #177000,R3      ;MASK CC
2920  020752  104002                          OCTPP                   ;PRINT CHECK CHARACTERS
2921  020754  000240          ERPT7:  NOP
2922  020756  032777  100000  157622  ERPX0:  BIT     #100000,@SWR    ;SEE IF STOP ON ERROR
2923  020764  001412                          BEQ     ERPX            ;IF NOT: BR
2924  020766  104006                          STOPP
2925  020770  005767  157674                  TST     PFLG            ;SEE IF HAVE PRINTED
2926  020774  001006                          BNE     ERPX            ;IF SO: BR
2927  020776  032777  002000  157602          BIT     #2000,@SWR      ;SEE IF SHOULD PRINT
2928  021004  001002                          BNE     ERPX            ;IF NOT: BR
2929  021006  000167  177316                  JMP     ERPT0           ;PRINT ERROR
2930  021012  005367  157652          ERPX:   CLR     PFLG
2931  021016  012777  000011  157464          MOV     #11.@C1         ;DRIVE CLEAR
2932  021024  017704  157476                  MOV     @AS,R4
2933  021030  010477  157472                  MOV     R4,@AS          ;CLEAR AS
2934  021034  016704  157450                  MOV     C1,R4
2935  021040  005204                          INC     R4
2936  021042  152714  000100                  BISB    #100,(R4)       ;RESET TRE
2937  021046  016777  157500  157466          MOV     UDES,@C2        ;RESET TC
2938  021054  032767  000040  157610  ERPX1:  BIT     #40,MTC1
2939  021062  001411                          BEQ     ERPX2           ;IF NOT READ/WRITE OP: BR
2940  021064  005767  157606                  TST     TMFLG
2941  021070  001406                          BEQ     ERPX2           ;IF NOT TM TIME: BR
2942  021072  016767  000034  174364          MOV     CRCSV,EXCRC     ;RESTORE CRC
2943  021100  016767  000024  174360          MOV     LRCSV,EXLRC     ;RESTORE LRC
2944  021106  000207          ERPX2:  RTS     PC                      ;EXIT
2945  021110  000000          CADER:  0                               ;EXPT ADDRESS SAVE
2946  021112  000000          BAER:   0
2947  021114  000000          CONER:  0
2948  021116  000000          DRVER:  0
2949  021120  000000          FCER:   0
```

```
2950  021122  000000                    LRCER:  0
2951  021124  000000                    CRCER:  0
2952  021126  000000                    ACTLRC: 0
2953  021130  000000                    LRCSV:  0
2954  021132  000000                    CRCSV:  0
2955
2956                                    ;*********************************************************
2957                                    ;F FOR FORWARD/R FOR REVERSE PRINT SUBROUTINE:
2958                                    ;
2959                                    ;THIS SUBROUTINE IS USED TO PRINT OUT THE
2960                                    ;TAPE DIRECTION USED WHEN ANY ERROR IS
2961                                    ;DETECTED IN STATUS OF READ OR WRITE, DATA, OR
2962                                    ;SPACING OPERATIONS.
2963                                    ;*********************************************************
2964
2965  021134  032767  000010  157530  FRPRT:  BIT     #10,MTC1        ;SEE IF WRITE COMMAND
2966  021142  001413                           BEQ     FREX            ;IF SO: BR
2967  021144  032767  000002  157520           BIT     #2,MTC1         ;SEE IF REVERSE
2968  021152  001404                           BEQ     FRO             ;IF NOT: BR
2969  021154  012704  025103                   MOV     #MSG17,R4
2970  021160  104000                           TTOUTT                  ;PRINT R
2971  021162  000403                           BR      FREX
2972  021164  012704  025100           FRO:    MOV     #MSG16,R4
2973  021170  104000                           TTOUTT                  ;PRINT F
2974  021172  000207                   FREX:   RTS     PC              ;EXIT
2975
```

```
                                        ;*********************************************************
                                        ;TAPE COMMAND EXECUTE SUBROUTINE:
                                        ;
                                        ;THIS SUBROUTINE IS USED TO EXECUTE THE
                                        ;MAG TAPE COMMAND DESCRIBED BY THE READ
                                        ;OR WRITE ROUTINE.  THE FINAL COMMAND IS
                                        ;SENT TO THE DEVICE REGISTER ALONG WITH THE
                                        ;INTERRUPT ENABLE AND GO BITS.
                                        ;ONCE THE COMMAND IS ISSUED, AN INTERRUPT
                                        ;TIMER IS STARTED AND IF NO INTERRUPT IS RETURNED
                                        ;BEFORE TIME OUT OCCURS, AN ERROR WILL BE
                                        ;PRINTED AND THE PROGRAM STOPPED.  TESTING MAY
                                        ;BE RESUMED BY PRESSING THE CONTINUE SWITCH.
                                        ;TWO INTERRUPT HANDLERS ARE USED, ONE FOR MAG TAPE
                                        ;AND ANOTHER FOR TELETYPE (TTY).
                                        ;UPON RECEIPT OF A MAG TAPE INTERRUPT, HOUSEKEEPING
                                        ;IS PERFORMED AND CONTROL RETURNED TO THE CALLING
                                        ;ROUTINE (READ,WRITE,ETC).
                                        ;RECEIPT OF A TTY INTERRUPT WILL CAUSE THE
                                        ;PROGRAM TO CHECK FOR ENTRY OF A CNTRL C CHARACTER.
                                        ;IF NOT CNTRL C, THEN CONTINUATION OF WAIT FOR MAG
                                        ;TAPE INTERRUPT IS RETURNED.  IF, HOWEVER, THE TTY
                                        ;INTERRUPT WAS CAUSED BY ENTRY OF A CNTRL C,
                                        ;THEN AT THIS TIME REQUESTS FOR NEW STALL VALUES
                                        ;ARE PRINTED AND THE RESPONSES ENTERED.  RESUMPTION
                                        ;OF TAPE INTERRUPT WAIT IS THEN RESUMED.
                                        ;*********************************************************

021174  005067  157442          TAPG:   CLR     TEMP1
021200  016777  157344  157312        MOV     DVN,@CS         ;SET DRIVE NO.
021206  032777  010000  157306  TAPG0: BIT     #10000,@DS      ;SEE IF HAVE MOL
021214  001026                        BNE     TAPG3           ;IF SO: BR
021216  005267  157420                INC     TEMP1           ;SEE IF TIMED OUT
021222  001371                        BNE     TAPG0           ;WAIT FOR READY
021224  004767  001432                JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
021230  032767  000010  157434        BIT     #10,MTC1        ;SEE IF WRITE OP
021236  001004                        BNE     TAPG1           ;IF NOT: BR
021240  012704  024742                MOV     #MSG5,R4
021244  104000                        TTOUTT                  ;PRINT WRITE ERR
021246  000405                        BR      TAPG2
021250  012704  024747          TAPG1: MOV     #MSG6,R4
021254  104000                        TTOUTT                  ;PRINT READ ERR
021256  004767  177652                JSR     PC,FRPRT        ;PRINT F OR R
021262  012704  025323          TAPG2: MOV     #MSG25,R4
021266  104000                        TTOUTT                  ;PRINT NO MOL ERR
021270  104006                        STOPP
021272  032777  020000  157222  TAPG3: BIT     #20000,@DS      ;SEE IF PIP RESET
021300  001411                        BEQ     TAPG3F          ;IF SO: BR
021302  004767  001354                JSR     PC,PAPRT        ;PRINT HEADER
021306  012704  027352                MOV     #MSG116,R4
021312  104000                        TTOUTT                  ;PRINT REWINDING MESSAGE
021314  032777  020000  157200  1$:    BIT     #20000,@DS
021322  001374                        BNE     1$              ;AWAIT PIP RESET
021324  022767  000026  157340  TAPG3F: CMP     #26,MTC1        ;SEE IF WRITE TM
021332  001003                        BNE     TAPG3A          ;IF NOT: BR
```

```
3032  021334  012704  177777                    MOV     #-1,R4          ;ELSE SET FC FOR -1
3033  021340  000406                            BR      TAPG3B
3034  021342  016704  157210          TAPG3A:   MOV     FMCNT,R4
3035  021346  032704  000001                    BIT     #1,R4
3036  021352  001401                            BEQ     TAPG3B
3037  021354  005304                            DEC     R4
3038  021356  000261                  TAPG3B:   SEC
3039  021360  006004                            ROR     R4              ;SET WC = FC/2 FOR NORMAL FORMAT
3040  021362  032767  000020  157162            BIT     #20,UDES        ;SEE IF CORE DUMP FORMAT
3041  021370  001402                            BEQ     TAPG3C          ;IF NOT: BR
3042  021372  000261                            SEC
3043  021374  006004                            ROR     R4              ;SET WC = FC/4 FOR CORE DUMP
3044  021376  010477  157110          TAPG3C:   MOV     R4,@WC          ;SET WORD COUNT
3045  021402  012777  000011  157100            MOV     #11,@C1         ;DRIVE CLEAR
3046  021410  017777  157102  157100            MOV     @FC,@FC         ;RESET FC LOADED
3047  021416  005767  157144                    TST     INTRF           ;SEE IF INTERCHANGE READ
3048  021422  001407                            BEQ     TAPG3D          ;IF NOT: BR
3049  021424  032767  000040  157240            BIT     #40,MTC1        ;SEE IF READ OP
3050  021432  001403                            BEQ     TAPG3D          ;IF NOT: BR
3051  021434  012777  000003  157072            MOV     #3,@MR          ;SET INTERCHANGE READ MAINT. MODE
3052  021442  016704  157224          TAPG3D:   MOV     MTC1,R4         ;GET COMMAND
3053  021446  042704  177707                    BIC     #177707,R4      ;MASK OP CODE
3054  021452  022704  000030                    CMP     #30,R4          ;SEE IF SPACE OP CODE
3055  021456  001403                            BEQ     TAPG3E          ;IF SO: BR
3056  021460  012767  177740  157200            MOV     #-40,STAL       ;SET INTERRUPT DELAY MULT TO -3
3057  021466  052767  000101  157176  TAPG3E:   BIS     #101,MTC1       ;SET INTERRUPT ENABLE AND GO
3058  021474  000240                            NOP
3059  021476  016777  157170  157004            MOV     MTC1,@C1        ;EXECUTE COMMAND
3060  021504  005077  157074                    CLR     @PSW            ;CLEAR PRIORITY
3061  021510  005067  157126                    CLR     TEMP1
3062  021514  005267  157122          TAPG4:    INC     TEMP1           ;SEE IF HAVE TIMED OUT
3063  021520  001375                            BNE     TAPG4           ;IF NOT: BR
3064  021522  005267  157140                    INC     STAL
3065  021526  001372                            BNE     TAPG4           ;DO TIME DELAY MULTIPLIER
3066  021530  012777  000340  157046  TAPG5:    MOV     #340,@PSW       ;RESET PRIORITY
3067  021536  032777  002000  157042            BIT     #2000,@SWR      ;SEE IF SHOULD PRINT ERRORS
3068  021544  001012                            BNE     TAPG6           ;IF NOT: BR
3069  021546  004767  001110                    JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
3070  021552  016704  157074                    MOV     EMADDR,R4
3071  021556  104000                            TTOUTT                  ;PRINT ERROR OP
3072  021560  004767  177350                    JSR     PC,FRPRT        ;PRINT F OR R
3073  021564  012704  025303                    MOV     #MSG24,R4
3074  021570  104000                            TTOUTT                  ;PRINT NO INTERRUPT
3075  021572  032777  100000  157006  TAPG6:    BIT     #100000,@SWR    ;SEE IF SHOULD HALT ON ERROR
3076  021600  001401                            BEQ     TAPG7           ;IF NOT: BR
3077  021602  104006                            STOPP
3078  021604  000167  000110          TAPG7:    JMP     MTINTA          ;RETURN TO CALLING ROUTINE
3079
```

```
3080
3081                                                    ;TTY INTERRUPT HANDLER**********
3082
3083  021610  012777  000340  156766  TTINT:  MOV     #340,@PSW       ;RESET PSW
3084  021616  005077  156766          CLR     @TKS            ;CLEAR TTY STATUS
3085  021622  122777  000203  156762  CMPB    #203,@TKB       ;SEE IF CONT C
3086  021630  001415                  BEQ     TTINTO          ;IF SO: BR
3087  021632  122777  000207  156752  CMPB    #207,@TKB       ;CHECK FOR CNTL G
3088  021640  001010                  BNE     RETURN
3089  021642  022767  000176  156736  CMP     #SWREG,SWR      ;IS SOFTWARE SWITCH REGISTER USED
3090  021650  001004                  BNE     RETURN          ;NO, GET OUT
3091  021652  005077  156734          CLR     @TKB            ;CLEAR CNTL G OUT OF BUFFER
3092  021656  004767  002540          JSR     PC,CNTG         ;GO CHANGE SWREG
3093  021662  000002          RETURN: RTI                     ;ELSE RETURN
3094  021664  010067  156756  TTINTO: MOV     R0,TEMP3        ;SAVE R0(REC CNTR)
3095  021670  004767  171762          JSR     PC,TINP4        ;GO GET STALL VALUES
3096  021674  016700  156746          MOV     TEMP3,R0        ;RESTORE R0(REC CNTR)
3097  021700  005077  156706          CLR     @TKB            ;CLEAR TTY BUFFER
3098  021704  012777  000100  156676  MOV     #100,@TKS       ;RESET INTERRUPT ENABLE
3099  021712  000002                  RTI                     ;RETURN
3100
3101                                                    ;MAG TAPE INTERRUPT HANDLER**********
3102
3103  021714  000240          MTINT:  NOP
3104  021716  022626                  CMP     (SP)+,(SP)+     ;RESET STACK POINTER
3105  021720  042777  000037  156606  MTINTA: BIC     #37,@MR         ;CLEAR MAINT MODE
3106  021726  000177  156730          JMP     @RTRN           ;RETURN
```

```
3107                                    ;**********************************
3108                                    ;AUTO SEQUENCE
3109                                    ;
3110                                    ;THIS ROUTINE ,ENTERED VIA STARTING ADDRESS 240
3111                                    ;WILL EXERCISE ALL AVAILABLE SLAVES ON ALL AVAILABLE
3112                                    ;DRIVES IN BOTH PE AND NRZ ACCORDING TO THE PRESELECTED
3113                                    ;TEST PLAN. IF NRZ ONLY, PE TESTING WILL NOT BE ATTEMPTED.
3114                                    ;**********************************
3115
3116  021732  012704  027162   ASEQ:   MOV     #MSG108,R4
3117  021736  104000                   TTOUTT                  ;PRINT NRZ ONLY REQUEST
3118  021740  012705  000650           MOV     #NRZOF,R5       ;SET ADDRESS OF FLAG
3119  021744  012701  000001           MOV     #1,R1           ;SET SIZE OF ENTRY
3120  021750  012702  000001           MOV     #1,R2           ;SET UPPER LIMIT
3121  021754  012703  000000           MOV     #0,R3           ;SET LOWER LIMIT
3122  021760  004767  001314           JSR     PC,TTR          ;GO GET RESPONSE
3123  021764  012704  026757           MOV     #MSG104,R4
3124  021770  104000                   TTOUTT                  ;REQUEST CONT OR NOT
3125  021772  012705  000742           MOV     #ASEQCF,R5      ;SET ADDRESS OF ENTRY
3126  021776  012701  000001           MOV     #1,R1           ;SET SIZE OF ENTRY
3127  022002  012702  000001           MOV     #1,R2           ;SET UPPER LIMIT
3128  022006  012703  000000           MOV     #0,R3           ;SET LOWER LIMIT
3129  022012  004767  001262           JSR     PC,TTR          ;GO GET INPUT
3130  022016  005067  156714   ASEQ0:  CLR     ADRVN           ;CLEAR DRV NUM
3131  022022  004767  000114   ASEQ1:  JSR     PC,HRDS         ;GO SELECT HARDWARE CONFIGURATION
3132  022026  012704  026723           MOV     #MSG101,R4
3133  022032  104000                   TTOUTT                  ;PRINT DIVIDER
3134  022034  012704  026737           MOV     #MSG102,R4
3135  022040  104000                   TTOUTT                  ;PRINT TM02 NUMBER
3136  022042  016703  156670           MOV     ADRVN,R3
3137  022046  104002                   OCTPP                   ;PRINT TM02
3138  022050  012704  026746           MOV     #MSG103,R4
3139  022054  104000                   TTOUTT                  ;PRINT SLAVE HDR
3140  022056  012700  000746           MOV     #UN1,R0         ;POINT TO START OF SLAVE TABLE
3141  022062  005710           ASEQ2:  TST     (R0)            ;SEE IF END
3142  022064  100403                   BMI     ASEQ3           ;IF SO: BR
3143  022066  012003                   MOV     (R0)+,R3
3144  022070  104002                   OCTPP                   ;PRINT SLAVE TABLE
3145  022072  000773                   BR      ASEQ2           ;DO ALL
3146  022074  004767  000220   ASEQ3:  JSR     PC,AMOD1        ;GO DO MODE 1(NRZ)
3147  022100  004767  000410           JSR     PC,AMOD2        ;GO DO MODE 2(PE)
3148  022104  022767  000007  156624 ASEQ4:  CMP     #7,ADRVN        ;SEE IF DONE ALL DRIVES
3149  022112  001403                   BEQ     ASEQX           ;IF SO: BR
3150  022114  005267  156616           INC     ADRVN           ;BUMP DRIVE NUMBER
3151  022120  000740                   BR      ASEQ1           ;CONTINUE
3152  022122  005767  156614   ASEQX:  TST     ASEQCF          ;SEE IF CONTINUOUS AUTO SEQ
3153  022126  001004                   BNE     ASEQXX          ;IF SO: BR
3154  022130  012704  026702           MOV     #MSG100,R4
3155  022134  104000                   TTOUTT                  ;PRINT END OF PASS
3156  022136  104006                   STOPP
3157  022140  000726           ASEQXX: BR      ASEQ0
```

```
3158
3159                                        ;SUBROUTINE TO SELECT AUTO SEQUENCE HARDWARE********
3160
3161  022142  005067  162644        HRDS:   CLR     REOTC           ;CLEAR EOT UNIT CNTR
3162  022146  005067  156470                CLR     TEMP1
3163  022152  012777  000040  156340        MOV     #40,@CS         ;INIT
3164  022160  016777  156552  156332        MOV     ADRVN,@CS       ;SET DRIVE
3165  022166  017701  156344                MOV     @DT,R1          ;READ DRIVE TYPE
3166  022172  032777  010000  156320        BIT     #10000,@CS      ;TEST FOR NON-EXISTANT DRIVE
3167  022200  001403                        BEQ     HRDS1           ;IF DRIVE AVAIL: BR
3168  022202  005726                HRDS0:  TST     (SP)+           ;RESET STACK POINTER
3169  022204  000167  177674                JMP     ASEQ4           ;GO SEE IF TRIED ALL DRIVES
3170  022210  042701  002007        HRDS1:  BIC     #2007,R1        ;MASK SLAVE TYPE
3171  022214  022701  140010                CMP     #140010,R1      ;SEE IF TU16 TAPE
3172  022220  001370                        BNE     HRDS0           ;IF NOT: BR
3173  022222  005000                        CLR     R0
3174  022224  012701  000746                MOV     #UN1,R1         ;SET START OF SLAVE TABLE
3175  022230  010077  156306        HRDS2:  MOV     R0,@C2          ;SELECT SLAVE
3176  022234  032777  010000  156260        BIT     #10000,@DS      ;SEE IF SLAVE AVAIL FOR TEST(MOL)
3177  022242  001403                        BEQ     HRDS3           ;IF NOT: BR
3178  022244  005267  156372                INC     TEMP1           ;SET SLAVE FOUND FLAG
3179  022250  010021                        MOV     R0,(R1)+        ;LOAD SLAVE TABLE
3180  022252  022700  000007        HRDS3:  CMP     #7,R0           ;SEE IF DONE ALL SLAVES
3181  022256  001402                        BEQ     HRDS4           ;IF SO: BR
3182  022260  005200                        INC     R0              ;ELSE BUMP SLAVE NUMBER
3183  022262  000762                        BR      HRDS2           ;CONTINUE SELECTION
3184  022264  005767  156352        HRDS4:  TST     TEMP1           ;SEE IF FOUND ANY SLAVES
3185  022270  001744                        BEQ     HRDS0           ;IF NOT: BR
3186  022272  016767  156344  162512        MOV     TEMP1,REOTC     ;SET NUMBER OF UNITS
3187  022300  000367  156336                SWAB    TEMP1
3188  022304  056767  156332  162500        BIS     TEMP1,REOTC     ;SET EOT CNTR
3189  022312  012711  177777                MOV     #-1,(R1)        ;TERMINATE SLAVE TABLE
3190  022316  000207                        RTS     PC              ;RETURN TO SEQ
```

```
3191
3192                                        ;SUBROUTINE TO SELECT NRZ AUTO TEST MODE**********
3193
3194  022320  005067  156330        AMOD1:  CLR     BLCNTR          ;ASSURE BLOCK COUNTER IS 0
3195  022324  012701  000746              MOV     #UN1,R1         ;GET START OF SLAVE TABLE
3196  022330  052721  001700        AMOD1A: BIS     #1700,(R1)+     ;SET ALL SLAVE TO NRZ,NORM,ODD
3197  022334  005111                      COM     (R1)
3198  022336  001402                      BEQ     AMOD1B          ;IF FILLED ALL SLAVES: BR
3199  022340  005111                      COM     (R1)
3200  022342  000772                      BR      AMOD1A          ;ELSE DO ALL
3201  022344  005111                AMOD1B: COM     (R1)
3202  022346  004767  162454              JSR     PC,RWNDA        ;GO REWIND ALL AVAIL SLAVES
3203  022352  012767  000006  156360      MOV     #6,ABLCNT       ;SET NUMBER OF BLOCKS FOR MODE 1
3204  022360  012767  174000  156170      MOV     #-4000,FMCNT    ;SET FC = 4000
3205  022366  012767  000100  156160      MOV     #100,RCNT       ;SET REC CNTR = 100
3206  022374  016767  156336  156146      MOV     ADRVN,DVN       ;SELECT DRIVE
3207  022402  012767  000001  156150      MOV     #1,PATRN        ;SELECT PATTERN 1
3208  022410  005067  156150              CLR     TMEX            ;ASSURE NO TMK
3209  022414  005067  156146              CLR     INTRF           ;ASSURE NORMAL READ
3210  022420  004767  160646              JSR     PC,STAUTO       ;GO DO AUTO MODE 1
3211  022424  012767  000010  156126      MOV     #10,PATRN       ;SELECT PATTERN 10
3212  022432  004767  160634              JSR     PC,STAUTO       ;GO DO PATTERN 10
3213  022436  012767  000014  156114      MOV     #14,PATRN       ;SELECT PATTERN 14
3214  022444  004767  160622              JSR     PC,STAUTO
3215  022450  005767  156174              TST     NRZOF           ;SEE IF NRZ ONLY
3216  022454  001411                      BEQ     AMOD1C          ;IF NOT: BR
3217  022456  012767  177777  156254      MOV     #-1,ABLCNT      ;FORCE TO EOT
3218  022464  012767  153624  156132      MOV     #153624,RANBAS
3219  022472  012767  032561  156126      MOV     #32561,RANSAV   ;RESET RANDOM DATA BASE
3220  022500  012767  177777  156052  AMOD1C: MOV     #-1,PATRN       ;SELECT AUTO RANDOM DATA
3221  022506  004767  160560              JSR     PC,STAUTO
3222  022512  000207                      RTS     PC              ;RETURN TO SEQ
```

```
3223
3224                                        ;SUBROUTINE TO SELECT PE AUTO TEST MODE**********
3225
3226  022514  005767  156130        AMOD2:  TST     NRZOF           ;SEE IF NRZ ONLY
3227  022520  001157                        BNE     AMOD2X          ;IF SO: BR
3228  022522  005067  156126                CLR     BLCNTR          ;CLEAR BLOCK CNTR
3229  022526  012701  000746                MOV     #UN1,R1         ;SET START OF SLAVE TABLE
3230  022532  042711  001700        AMOD2A: BIC     #1700,(R1)      ;CLEAR NRZ
3231  022536  052721  002300                BIS     #2300,(R1)+     ;SET TO PE NORM, ODD
3232  022542  005111                        COM     (R1)            ;SEE IF END OF TABLE
3233  022544  001402                        BEQ     AMOD2B          ;IF SO: BR
3234  022546  005111                        COM     (R1)
3235  022550  000770                        BR      AMOD2A          ;CONTINUE
3236  022552  005111                AMOD2B: COM     (R1)
3237  022554  004767  162246                JSR     PC,RWNDA        ;REWIND ALL SLAVES
3238  022560  012767  000006  156152        MOV     #6,ABLCNT       ;SET AUTO BLOCK COUNT
3239  022566  012767  174000  155762        MOV     #-4000,FMCNT    ;SET FC = 4000
3240  022574  012767  000100  155752        MOV     #100,RCNT       ;SET REC CNTR TO 100
3241  022602  012767  000010  155750        MOV     #10,PATRN       ;SELECT PATTERN 10
3242  022610  004767  160456                JSR     PC,STAUTO       ;GO DO AUTO SEQ
3243  022614  012767  000014  155736        MOV     #14,PATRN       ;SELECT PATTERN 14
3244  022622  004767  160444                JSR     PC,STAUTO
3245  022626  012767  000015  155724        MOV     #15,PATRN       ;SELECT PATTERN 15
3246  022634  004767  160432                JSR     PC,STAUTO
3247  022640  012767  177777  156072        MOV     #-1,ABLCNT      ;FORCE TO END OF TAPE
3248  022646  012767  177777  155704        MOV     #-1,PATRN       ;SELECT AUTO RANDOM DATA
3249  022654  004767  160412                JSR     PC,STAUTO
3250  022660  000207                AMOD2X: RTS     PC              ;RETURN TO SEQ
3251
3252
```

```
3253
3254                                    ;*******************************************************
3255                                    ;ERROR HEADER PRINT SUBROUTINE:
3256                                    ;
3257                                    ;THIS ROUTINE IS USED TO PRINT OUT A HEADER
3258                                    ;WITH EACH ERROR MESSAGE.  THE PRINT IS IN TWO
3259                                    ;LINES AND CONTAINS THE FOLLOWING INFORMATION.
3260                                    ;LINE 1:  DRIVE NO. SLAVE NO. DENSITY PARITY FORMAT
3261                                    ;LINE 2:  CURRENT BLOCK NUMBER, RECORD NUMBER IN
3262                                    ;WHICH THE ERROR OCCURED PLUS THE TOTAL NUMBER
3263                                    ;OF RECORDS IN THIS BLOCK, THE RECORD SIZE (NUMBER
3264                                    ;OF CHARACTERS), AND THE ERROR TYPE (READ,WRITE, SPACE, ETC)
3265                                    ;PLUS THE TAPE DIRECTION (FORWARD OR REVERSE).
3266                                    ;ALL NUMBERS ARE IN OCTAL.
3267                                    ;*******************************************************
3268
3269  022662  012704  025020       PAPRT:  MOV     #MSG12,R4
3270  022666  104000                       TTOUTT                  ;PRINT DRIVE HEADER
3271  022670  016703  155654               MOV     DVN,R3
3272  022674  104002                       OCTPP                   ;PRINT DRIVE NUMBER
3273  022676  012704  025004               MOV     #MSG11,R4
3274  022702  104000                       TTOUTT                  ;PRINT UNIT HEADER
3275  022704  016703  155642               MOV     UDES,R3
3276  022710  042703  177770               BIC     #177770,R3
3277  022714  104002                       OCTPP                   ;PRINT UNIT NUMBER
3278  022716  012704  026301               MOV     #MSG60,R4
3279  022722  104000                       TTOUTT                  ;PRINT DENSITY TAG
3280  022724  016703  155622               MOV     UDES,R3
3281  022730  000303                       SWAB    R3
3282  022732  042703  177770               BIC     #177770,R3
3283  022736  104002                       OCTPP                   ;PRINT DENSITY
3284  022740  012704  026305               MOV     #MSG61,R4
3285  022744  104000                       TTOUTT                  ;PRINT PARITY TAG
3286  022746  005003                       CLR     R3
3287  022750  032767  000010  155574       BIT     #10,UDES
3288  022756  001402                       BEQ     PAPRTO
3289  022760  012703  000001               MOV     #1,R3
3290  022764  104002             PAPRTO:   OCTPP                   ;PRINT PARITY
3291  022766  012704  026311               MOV     #MSG62,R4
3292  022772  104000                       TTOUTT                  ;PRINT FORMAT TAG
3293  022774  016703  155552               MOV     UDES,R3
3294  023000  000241                       CLC
3295  023002  006003                       ROR     R3
3296  023004  006003                       ROR     R3
3297  023006  006003                       ROR     R3              ;PONTION FORMAT
3298  023010  006003                       ROR     R3
3299  023012  042703  177760               BIC     #177760,R3
3300  023016  104002                       OCTPP                   ;PRINT FORMAT
3301  023020  012704  024761               MOV     #MSG8,R4
3302  023024  104000                       TTOUTT                  ;PRINT PATRN TAG
3303  023026  032777  000400  155552       BIT     #400,@SWR       ;SEE IF RANDOM DATA
3304  023034  001406                       BEQ     PAPRTB          ;IF NOT: BR
3305  023036  012767  000122  155572 PAPRTA:  MOV     #122,TOB
3306  023044  004767  000572               JSR     PC,TUG          ;PRINT R
3307  023050  000411                       BR      PAPRTD
3308  023052  005767  155656       PAPRTB:  TST     ASEQF           ;SEE IF AUTO SEQ
```

```
3309  023056  001403                        BEQ     PAPRTC          ;IF NOT: BR
3310  023060  005767  155474                TST     PATRN           ;SEE IF AUTO RANDOM
3311  023064  100764                        BMI     PAPRTA          ;IF SO: BR
3312  023066  016703  155466        PAPRTC: MOV     PATRN,R3
3313  023072  104002                        OCTPP                   ;PRINT PATRN NUMBER
3314  023074  012704  025034        PAPRTD: MOV     #MSG13,R4
3315  023100  104000                        TTOUTT                  ;PRINT BLOCK NO. HEADER
3316  023102  016703  155546                MOV     BLCNTR,R3
3317  023106  104002                        OCTPP                   ;PRINT NUMBER
3318  023110  012704  025042                MOV     #MSG14,R4
3319  023114  104000                        TTOUTT                  ;PRINT REC NO. HEADER
3320  023116  010003                        MOV     R0,R3
3321  023120  032767  000010  155544        BIT     #10,MTC1        ;SEE IF WRITE OPERATION
3322  023126  001404                        BEQ     PAPRT1          ;IF SO: BR
3323  023130  032767  010000  155424        BIT     #10000,RDCMD    ;SEE IF READ REVERSE
3324  023136  001016                        BNE     PAPRT3          ;IF SO: BR
3325  023140  016703  155410        PAPRT1: MOV     RCNT,R3
3326  023144  005767  155526                TST     TMFLG           ;SEE IF TAPE MARK TIME
3327  023150  001010                        BNE     PAPRT2          ;IF SO: BR
3328  023152  022767  012136  155502        CMP     #B1,RTRN
3329  023160  001003                        BNE     PAPRTY          ;IF NOT BACK SPACE: BR
3330  023162  005767  155524                TST     RTYFL
3331  023166  001402                        BEQ     PAPRT3          ;IF NOT RETRY: BR
3332  023170  160003        PAPRTY: SUB     R0,R3           ;GET RECORD NUMBER
3333  023172  005203        PAPRT2: INC     R3
3334  023174  104002        PAPRT3: OCTPP                   ;PRINT RECORD NUMBER
3335  023176  012767  000055  155432        MOV     #55,TOB         ;LOAD DASH (-)
3336  023204  004767  000432                JSR     PC,TOG          ;PRINT DASH (-)
3337  023210  016703  155340                MOV     RCNT,R3
3338  023214  104002                        OCTPP                   ;PRINT RECORD COUNT
3339  023216  012704  024754                MOV     #MSG7,R4
3340  023222  104000                        TTOUTT                  ;PRINT RECORD SIZE HEADER
3341  023224  016703  155326                MOV     FMCNT,R3        ;GET CHARACTER COUNT
3342  023230  005303                        DEC     R3
3343  023232  005103                        COM     R3              ;REMOVE TWOS COMPLEMENT
3344  023234  104002                        OCTPP                   ;PRINT RECORD SIZE
3345  023236  012767  000001  155420        MOV     #1,HDRFL        ;SET HEADER FLAG
3346  023244  000207                        RTS     PC              ;RETURN
3347
```

```
 3348
 3349                                   ;*********************************************************
 3350                                   ;RANDOM NUMBER GENERATOR SUBROUTINE:
 3351                                   ;
 3352                                   ;THIS SUBROUTINE IS USED TO GENERATE THE RANDOM
 3353                                   ;NUMBERS REQUIRED FOR USE AS RANDOM DATA,
 3354                                   ;RECORD COUNT, AND CHARACTER COUNT.
 3355                                   ;*********************************************************
 3356
 3357  023246  066767  155354  155350  RANG:  ADD     RANSAV,RANBAS
 3358  023254  066767  155344  155344         ADD     RANBAS,RANSAV   ;GET NEW NUMBER
 3359  023262  026701  155340                 CMP     RANSAV,R1       ;SEE IF NUMBER TOO BIG
 3360  023266  101367                         BHI     RANG            ;IF SO: BR
 3361  023270  020267  155332                 CMP     R2,RANSAV       ;SEE IF NUMBER TOO SMALL
 3362  023274  101364                         BHI     RANG            ;IF SO: BR
 3363  023276  000207                         RTS     PC              ;EXIT
 3364
```

```
                                        ;*********************************************************
                                        ;TTY ENTRY SUBROUTINE:
                                        ;
                                        ;THIS SUBROUTINE IS USED BY THE TEST CONDITION
                                        ;ENTRY ROUTINE TO READ THE RESPONSE ENTERED
                                        ;AT THE TTY AND CHECK THEM FOR LEGALITY AND
                                        ;LIMITS.  ALL RESPONSE MUST BE TYPED IN OCTAL
                                        ;(0-7) AND MUST FALL WITHIN THE LIMITS SET BY
                                        ;THE CALLING ROUTINE.
                                        ;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
                                        ;A QUESTION MARK IS TYPED (?) AND THE RESPONSE
                                        ;MAY BE REENTERED.
                                        ;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
                                        ;MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
                                        ;CARRIAGE RETURN
                                        ;*********************************************************

023300  005067  155336        TTR:    CLR     TEMP1           ;CLEAR FIRST CHARACTER FLAG
023304  005000                TTR0:   CLR     R0
023306  104010                        TTINN                   ;GO READ CHARACTER
023310  122767  000215  155322        CMPB    #215,TIB        ;SEE IF CR
023316  001005                        BNE     TTR1            ;IF NOT: BR
023320  005767  155316                TST     TEMP1           ;SEE IF FIRST CHARACTER
023324  001446                        BEQ     TTR5            ;IF SO: BR
023326  000167  000066                JMP     TTR2            ;ELSE GO LOAD VALUE
023332  122767  000260  155300  TTR1: CMPB    #260,TIB        ;SEE IF CHAR IS LESS THAN 0
023340  101402                        BLOS    TTR1A           ;IF NOT: BR
023342  000167  000076                JMP     TINER           ;ELSE GO TO ERROR
023346  122767  000270  155264  TTR1A:CMPB    #270,TIB        ;SEE IF CHAR IS GREATER THAN 7
023354  101002                        BHI     TTR1B           ;IF NOT: BR
023356  000167  000062                JMP     TINER           ;ELSE GO TO ERROR
023362  005267  155254        TTR1B:  INC     TEMP1           ;SET FIRST CHARACTER FLAG
023366  000241                        CLC
023370  006100                        ROL     R0
023372  000241                        CLC
023374  006100                        ROL     R0              ;SHIFT 3 LEFT
023376  000241                        CLC
023400  006100                        ROL     R0
023402  042767  177770  155230        BIC     #177770,TIB     ;STRIP ASCII
023410  056700  155224                BIS     TIB,R0          ;LOAD CHARACTER
023414  005301                        DEC     R1              ;SEE IF DONE
023416  001333                        BNE     TTR0            ;IF NOT: BR
023420  020002                TTR2:   CMP     R0,R2           ;SEE IF EXCEEDED MAXIMUM LIMIT
023422  101402                        BLOS    TTR3            ;IF NOT: BR
023424  000167  000014                JMP     TINER           ;ELSE GO TO ERROR
023430  020300                TTR3:   CMP     R3,R0           ;SEE IF BELOW MINIMUM LIMIT
023432  101402                        BLOS    TTR4            ;IF NOT: BR
023434  000167  000004                JMP     TINER           ;ELSE GO TO ERROR
023440  010015                TTR4:   MOV     R0,(R5)         ;LOAD VALUE
023442  000207                TTR5:   RTS     PC              ;EXIT
023444  012704  026013        TINER:  MOV     #MSG43,R4
023450  104000                        TTOUTT                  ;PRINT?
023452  162716  000020                SUB     #20,(SP)        ;RESET SP TO START OF VALUE ROUTINE
023456  000207                        RTS     PC              ;REDO VALUE ENTRY
```

```
3419
3420                                            ;TTY READ SUBROUTINE**********
3421
3422  023460  005077  155124          TTIN:     CLR     @TKS
3423  023464  005077  155122                    CLR     @TKB
3424  023470  005067  155144                    CLR     TIB
3425  023474  005277  155110                    INC     @TKS
3426  023500  105777  155104          TTIN1:    TSTB    @TKS
3427  023504  100375                            BPL     TTIN1
3428  023506  017767  155100  155124            MOV     @TKB,TIB
3429  023514  105777  155074          TTIN2:    TSTB    @TPS
3430  023520  100375                            BPL     TTIN2
3431  023522  116777  155112  155066            MOVB    TIB,@TPB
3432  023530  000207                            RTS     PC
3433
3434                                            ;TTY OUTPUT SUBROUTINE************
3435
3436  023532  112467  155100          TTOUT:    MOVB    (R4)+,TOB
3437  023536  122767  000043  155072            CMPB    #43,TOB
3438  023544  001444                            BEQ     TEX
3439  023546  122767  000045  155062            CMPB    #45,TOB
3440  023554  001407                            BEQ     TCRLF
3441  023556  122767  000041  155052            CMPB    #41,TOB
3442  023564  001435                            BEQ     TBELL           :DO BELL
3443  023566  004767  000050                    JSR     PC,TOG
3444  023572  000757                            BR      TTOUT
3445  023574  112767  000015  155034  TCRLF:    MOVB    #15,TOB
3446  023602  004767  000034                    JSR     PC,TOG
3447  023606  012703  000006                    MOV     #6,R3
3448  023612  005067  155020          TCRLFA:   CLR     TOB
3449  023616  004767  000020                    JSR     PC,TOG
3450  023622  005303                            DEC     R3
3451  023624  001372                            BNE     TCRLFA          :DO FILLERS
3452  023626  112767  000012  155002            MOVB    #12,TOB
3453  023634  004767  000002                    JSR     PC,TOG
3454  023640  000734                            BR      TTOUT
3455  023642  105777  154746          TOG:      TSTB    @TPS
3456  023646  100375                            BPL     TOG
3457  023650  116777  154762  154740            MOVB    TOB,@TPB
3458  023656  000207                  TEX:      RTS     PC
3459  023660  012703  000002          TBELL:    MOV     #2,R3
3460  023664  012767  000007  154744  TBELA:    MOV     #7,TOB
3461  023672  004767  177744                    JSR     PC,TOG
3462  023676  005303                            DEC     R3
3463  023700  001371                            BNE     TBELA
3464  023702  000713                            BR      TTOUT
3465
3466
```

```
3467                                                ;OCTAL OUTPUT SUBROUTINE************
3468
3469  023704  005067  000230          OCTP:   CLR     OFL             ;CLEAR FLAG FOR LEADING ZERO
3470  023710  000403                          BR      OCTPE1
3471  023712  012767  000001  000220  OCTPE:  MOV     #1,OFL
3472  023720  010304                  OCTPE1: MOV     R3,R4           ;SEE IF NUMBER IS ZERO
3473  023722  001007                          BNE     OCTPO           ;IF NOT ZERO: BR
3474  023724  005767  000210                  TST     OFL
3475  023730  001004                          BNE     OCTPO
3476  023732  004767  000162                  JSR     PC,OCTPG1       ;ELSE PRINT ZERO
3477  023736  000167  000120                  JMP     OCTP3           ;SPACE AND EXIT
3478  023742  032704  100000          OCTPO:  BIT     #100000,R4      ;SEE IF MSD = 1
3479  023746  001406                          BEQ     OCTP1           ;IF NOT: BR
3480  023750  012704  000001                  MOV     #1,R4
3481  023754  004767  000116                  JSR     PC,OCTPG        ;PRINT 1
3482  023760  000167  000006                  JMP     OCTP2
3483  023764  005004                  OCTP1:  CLR     R4
3484  023766  004767  000104                  JSR     PC,OCTPG        ;PRINT 0
3485  023772  010304                  OCTP2:  MOV     R3,R4
3486  023774  006004                          ROR     R4
3487  023776  006004                          ROR     R4
3488  024000  006004                          ROR     R4              ;POSITION DIGIT
3489  024002  006004                          ROR     R4
3490  024004  000304                          SWAB    R4
3491  024006  004767  000064                  JSR     PC,OCTPG        ;PRINT DIGIT 2
3492  024012  010304                          MOV     R3,R4
3493  024014  006004                          ROR     R4
3494  024016  000304                          SWAB    R4
3495  024020  004767  000052                  JSR     PC,OCTPG        ;PRINT DIGIT 3
3496  024024  010304                          MOV     R3,R4
3497  024026  006104                          ROL     R4
3498  024030  006104                          ROL     R4
3499  024032  000304                          SWAB    R4
3500  024034  004767  000036                  JSR     PC,OCTPG        ;PRINT DIGIT 4
3501  024040  010304                          MOV     R3,R4
3502  024042  006004                          ROR     R4
3503  024044  006004                          ROR     R4
3504  024046  006004                          ROR     R4
3505  024050  004767  000022                  JSR     PC,OCTPG
3506  024054  010304                          MOV     R3,R4
3507  024056  004767  000014                  JSR     PC,OCTPG        ;PRINT DIGIT 5
3508  024062  012767  000240  154546  OCTP3:  MOV     #240,TOB
3509  024070  004767  177546                  JSR     PC,TOG          ;PRINT SPACE
3510  024074  000207                          RTS     PC              ;EXIT
3511  024076  042704  177770          OCTPG:  BIC     #177770,R4
3512  024102  001004                          BNE     OCTPGO
3513  024104  005767  000030                  TST     OFL
3514  024110  001001                          BNE     OCTPGO
3515  024112  000207                          RTS     PC
3516  024114  005267  000020          OCTPGO: INC     OFL
3517  024120  052704  000260          OCTPG1: BIS     #260,R4
3518  024124  010467  154506                  MOV     R4,TOB
3519  024130  004767  177506                  JSR     PC,TOG
3520  024134  010304                          MOV     R3,R4
3521  024136  000207                          RTS     PC
3522  024140  000000                  OFL:    0                       ;FIRST CHAR FLAG
```

```
3523
3524                                        ;DATA CHARACTER OUTPUT SUBROUTINE************
3525
3526  024142  005067  154470          DOUT: CLR     TOB
3527  024146  012704  000010                MOV     #10,R4          ;SET NUMBER TO PRINT
3528  024152  110367  154460                MOVB    R3,TOB
3529  024156  105777  154432          DOUT1: TSTB   @TPS
3530  024162  100375                       BPL     DOUT1
3531  024164  132767  000200  154444        BITB    #200,TOB
3532  024172  001404                       BEQ     DOUT2
3533  024174  012777  000061  154414        MOV     #061,@TPB
3534  024202  000403                       BR      DOUT3
3535  024204  012777  000060  154404  DOUT2: MOV     #060,@TPB
3536  024212  006167  154420        DOUT3: ROL     TOB
3537  024216  005304                       DEC     R4
3538  024220  001356                       BNE     DOUT1
3539  024222  000207                       RTS     PC
3540  024224  016703  154416        DOUTD: MOV     TEMP3,R3
3541  024230  000303                       SWAB    R3
3542  024232  004767  177704               JSR     PC,DOUT
3543  024236  016703  154404               MOV     TEMP3,R3
3544  024242  004767  177674               JSR     PC,DOUT
3545  024246  000207                       RTS     PC
3546
3547                                        ;TU16 SERIAL NUMBER PRINT SUBROUTINE*******
3548
3549  024250  010304                 SNPT: MOV     R3,R4
3550  024252  000304                       SWAB    R4
3551  024254  006004                       ROR     R4
3552  024256  006004                       ROR     R4
3553  024260  006004                       ROR     R4
3554  024262  006004                       ROR     R4
3555  024264  004767  000036               JSR     PC,SNPG         ;PRINT FIRST DIGIT
3556  024270  010304                       MOV     R3,R4
3557  024272  000304                       SWAB    R4
3558  024274  004767  000026               JSR     PC,SNPG         ;PRINT SECOND DIGIT
3559  024300  010304                       MOV     R3,R4
3560  024302  006004                       ROR     R4
3561  024304  006004                       ROR     R4
3562  024306  006004                       ROR     R4
3563  024310  006004                       ROR     R4
3564  024312  004767  000010               JSR     PC,SNPG         ;PRINT THIRD DIGIT
3565  024316  010304                       MOV     R3,R4
3566  024320  004767  000002               JSR     PC,SNPG         ;PRINT FOURTH DIGIT
3567  024324  000207                       RTS     PC              ;EXIT
3568  024326  012767  000260  154302 SNPG: MOV     #260,TOB        ;SET NUMBER BASE
3569  024334  042704  177760               BIC     #177760,R4      ;MASK NUMBER
3570  024340  050467  154272               BIS     R4,TOB          ;BUILD DIGIT
3571  024344  004767  177272               JSR     PC,TOG          ;GO TYPE
3572  024350  000207                       RTS     PC              ;RETURN
3573
3574
3575                                        ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ^G TO ALLOW CHANGING
3576                                        ;OF LOC.176.
3577                                        ;CALL IS BY WAY OF CKSWRR
3578                                        ;LOCATIONS USED:
```

```
3579  024352  000000                          TEMPST: .WORD   0
3580  024354  000000                          COUNT:  .WORD   0
3581  024356  000000                          RDSW:   .WORD   0
3582  024360  022767  000176  154220  CkSWR:  CMP     #SWREG,SWR              ;SOFTWARE SWITCH REG PRESENT
3583  024366  001127                          BNE     OUT                     ;NO, GET OUT
3584  024370  105777  154214                  TSTB    @TKS                    ;YES,WAIT FOR
3585  024374  100124                          BPL     OUT                     ;READY, GET CHARACTER
3586  024376  017767  154210  154234          MOV     @TKB,TIB                ;AND STRIP OFF
3587  024404  042767  177600  154226          BIC     #177600,TIB             ;THE GARBAGE
3588  024412  022767  000007  154220          CMP     #7,TIB          ;IS IT A ^G
3589  024420  001112                          BNE     OUT
3590  024422  012704  027426          CNTG:   MOV     #$CNTG,R4
3591  024426  104000                          TTOUTT
3592  024430  012704  027432          CNTLU:  MOV     #$MSWR,R4
3593  024434  104000                          TTOUTT
3594  024436  017703  154144                  MOV     @SWR,R3
3595  024442  004767  177244                  JSR     PC,OCTPE
3596  024446  012704  027441                  MOV     #$MNEW,R4
3597  024452  104000                          TTOUTT
3598  024454  005067  177672          $READ:  CLR     TEMPST
3599  024460  012767  000007  177666          MOV     #7,COUNT
3600  024466  104010                  1$:     TTINN                           ;GO  READ A CHARACTER
3601  024470  042767  177600  154142          BIC     #177600,TIB             ;STRIP OFF GARBAGE
3602  024476  122767  000025  154134          CMPB    #25,TIB                 ;IS IT A ^U?
3603  024504  001002                          BNE     2$                      ;BRANCH IF NOT
3604  024506  005726                  3$:     TST     (SP)+                   ;POP THE STACK
3605  024510  000747                          BR CNTLU                        ;START OVER
3606  024512  122767  000015  154120  2$:     CMPB    #15,TIB                 ;IS IT A <CR>?
3607  024520  001013                          BNE     4$                      ;BRANCH IF NOT
3608  024522  012767  000200  177626          MOV     #200,RDSW
3609  024530  012704  027451                  MOV     #MCRLF,R4
3610  024534  104000                          TTOUTT
3611  024536  022767  000007  177610          CMP     #7,COUNT                ;WAS IT FIRST CHARACTER
3612  024544  001035                          BNE     7$                      ;CHANGE SWR IF NOT FIRST ONE
3613  024546  000437                          BR      OUT                     ;GET OUT
3614  024550  122767  000060  154062  4$:     CMPB    #60,TIB
3615  024556  003004                          BGT     5$
3616  024560  122767  000067  154052          CMPB    #67,TIB
3617  024566  002004                          BGE     6$
3618  024570  012704  026013          5$:     MOV     #MSG43,R4
3619  024574  104000                          TTOUTT
3620  024576  000743                          BR      3$                      ;START OVER IF NOT LEGAL CHARACTER
3621  024600  006367  177546          6$:     ASL     TEMPST
3622  024604  006367  177542                  ASL     TEMPST
3623  024610  006367  177536                  ASL     TEMPST
3624  024614  142767  000060  154016          BICB    #60,TIB                 ;GET NITTY-GRITTY
3625  024622  156767  154012  177522          BISB    TIB,TEMPST
3626  024630  005367  177520                  DEC     COUNT                   ;ONLY WANT 6   DIGITS
3627  024634  001755                          BEQ     5$
3628  024636  000713                          BR      1$
3629  024640  016777  177506  153740  7$:     MOV     TEMPST,@SWR             ;CHANGE SWITCH REGISTER CONTENTS
3630  024646  000207                  OUT:    RTS     PC                      ;RETURN TO BODY OF PROGRAM
3631                                          ;HALT HANDLER**********
3632
3633  024650  000000                  STOP:   HALT
3634  024652  104004                          CKSWRR                          ;CHECK FOR CONTROL G
```

```
3635  024654  000207                          R5      PC
3636
3637
3638                                          ;TRAP HANDLER*************
3639
3640  024656  016677  000002  153720  TRAP30: MOV     2(6),@PSW       ;ADJUST PSW
3641  024664  011666  000002                  MOV     @SP,2(6)        ;PLACE RETURN ADDRESS OVER PSW
3642  024670  162716  000002                  SUB     #2,@SP          ;SUB. 2 FROM RETURN ADDRESS
3643  024674  013646                          MOV     @(6)+,-(6)
3644  024676  062716  120704                  ADD     #TABLE-104000,@SP       ;GET SUBROUTINE STARTING ADDRESS
3645  024702  013607                          MOV     @(SP)+,PC       ;GO TO SUBROUTINE
3646  024704  023532                  TABLE:  TTOUT
3647  024706  023704                          OCTP
3648  024710  024360                          CKSWR
3649  024712  024650                          STOP
3650  024714  023460                          TTIN
3651          104000                  TTOUTT= 104000
3652          104002                  OCTPP=  104002
3653          104004                  CKSWRR= 104004
3654          104006                  STOPP=  104006
3655          104010                  TTINN=  104010
3656
```

```
                                        ;ERROR MESSAGES***********
024716 042052 020105     043  MSG1:   .ASCII  /*DE #/
024723     045 035507  021440  MSG2:   .ASCII  /%G; #/
024730 041045 020073     043  MSG3:   .ASCII  /%B; #/
024735     045 047103  021440  MSG4:   .ASCII  /%CN #/
024742 053452 020105     043  MSG5:   .ASCII  /*WE #/
024747     052 042522  021440  MSG6:   .ASCII  /*RE #/
024754 051052 020123     043  MSG7:   .ASCII  /*RS #/
024761     052 040520  051124  MSG8:   .ASCII  /*PATRN #/
024766 020116     043
024771     040 047123  020072  MSG9:   .ASCII  / SN: #/
024776     043
024777     052 042523  021440  MSG10:  .ASCII  /*SE #/
025004 051452 040514  042526  MSG11:  .ASCII  /*SLAVE NO. #/
025012 047040 027117  021440
025020 042045 044522  042526  MSG12:  .ASCII  /%DRIVE NO. #/
025026 047040 027117  021440
025034 025045 047102  021440  MSG13:  .ASCII  /%*BN #/
025042 051052 020116     043  MSG14:  .ASCII  /*RN #/
025047     045 020041  020040  MSG15:  .ASCII  /%!          BAD RECORD%%#/
025054 020040 020040  020040
025062 041040 042101  051040
025070 041505 051117  022504
025076 021445
025100 043040     043        MSG16:  .ASCII  / F#/
025103     040 021522         MSG17:  .ASCII  / R#/
025106 020041 047505  020124  MSG20:  .ASCII  /! EOT NO: #/
025114 047516 020072     043
025121     045 047111  042524  MSG21:  .ASCII  /%INTERCHANGE READ = #/
025126 041522 040510  043516
025134 020105 042522  042101
025142 036440 021440
025146 020445 046111  042514  MSG22:  .ASCII  /%!ILLEGAL BOT: HALT%#/
025154 040507 020114  047502
025162 035124 044040  046101
025170 022524     043
```

```
3713
3714  025173    045  051503  020061  MSG23:  .ASCII  /%CS1 #/
3715  025200    043
3716
3717  025201    045  041527  02144C  MSG23A: .ASCII  /%WC #/
3718
3719  025206  041045  020101    043  MSG23B: .ASCII  /%BA #/
3720
3721  025213    045  041506  021440  MSG23C: .ASCII  /%FC #/
3722
3723  025220  041445  031123  021440  MSG23D: .ASCII  /%CS2 #/
3724
3725  025226  042045  020123    043  MSG23E: .ASCII  /%DS #/
3726
3727  025233    045  051105  021440  MSG23F: .ASCII  /%ER #/
3728
3729  025240  040445  020123    043  MSG23G: .ASCII  /%AS #/
3730
3731  025245    045  045503  021440  MSG23H: .ASCII  /%CK #/
3732
3733  025252  042045  020102    043  MSG23I: .ASCII  /%DB #/
3734
3735  025257    045  051115  021440  MSG23J: .ASCII  /%MR #/
3736
3737  025264  042045  020124    043  MSG23K: .ASCII  /%DT #/
3738
3739  025271    045  041524  021440  MSG23L: .ASCII  /%TC #/
3740
3741  025276  051445  020116    043  MSG23M: .ASCII  /%SN #/
3742
3743  025303    045  047041  020117  MSG24:  .ASCII  /%!NO INTERRUPT%#/
3744  025310  047111  042524  051122
3745  025316  050125  022524    043
3746
3747  025323    045  047041  020117  MSG25:  .ASCII  /%!NO MOL: HALT%#/
3748  025330  047515  035114  044040
3749  025336  046101  022524    043
3750
3751  025343    045  051104  050117  MSG26:  .ASCII  /%DROPS: #/
3752  025350  035123  021440
3753
3754  025354  050045  041511  051513  MSG27:  .ASCII  /%PICKS: #/
3755  025362  020072    043
3756
3757  025365    045    043          MSG28:  .ASCII  /%#/
3758  025367    045  052045  030525  MSG30:  .ASCII  /%%TU16 AUTO SEQUENCE (DZTUA-F)%/
3759  025374  020066  052501  047524
3760  025402  051440  050505  042525
3761  025410  041516  020105  042050
3762  025416  052132  040525  043055
3763  025424  022451
3764  025426  040515  042513  042440          .ASCII  /MAKE ENTRIES IN OCTAL%#/
3765  025434  052116  044522  051505
3766  025442  044440  020116  041517
3767  025450  040524  022514    043
3768  025455    045  052045  030525  MSG31:  .ASCII  /%%TU16 DATA RELIABILITY TEST (DZTUA-F)%MAKE ENTRIES IN OCTAL%:#/
```

```
 3769  025462  020066  040504  040524
 3770  025470  051040  046105  040511
 3771  025476  044502  044514  054524
 3772  025504  052040  051505  020124
 3773  025512  042050  052132  040525
 3774  025520  043055  022451  040515
 3775  025526  042513  042440  052116
 3776  025534  044522  051505  044440
 3777  025542  020116  041517  040524
 3778  025550  022514     043
 3779
 3780  025553     045  046123  053101  MSG32:  .ASCII  /%SLAVE NUMBER = #/
 3781  025560  020105  052516  041115
 3782  025566  051105  036440  021440
 3783
 3784  025574  042045  047105  044523  MSG33:  .ASCII  /%DENSITY = #/
 3785  025602  054524  036440  021440
 3786
 3787  025610  050045  051101  052111  MSG34:  .ASCII  /%PARITY = #/
 3788  025616  020131  020075     043
 3789
 3790  025623     045  042522  047503  MSG35:  .ASCII  /%RECORD COUNT = #/
 3791  025630  042122  041440  052517
 3792  025636  052116  036440  021440
 3793
 3794  025644  041445  040510  040522  MSG36:  .ASCII  /%CHARACTER COUNT = #/
 3795  025652  052103  051105  041440
 3796  025660  052517  052116  036440
 3797  025666  021440
 3798
 3799  025670  050045  052101  042524  MSG37:  .ASCII  /%PATTERN NUMBER = #/
 3800  025676  047122  047040  046525
 3801  025704  042502  020122  020075
 3802  025712     043
 3803  025713     045  044523  043516  MSG38:  .ASCII  /%SINGLE PASS = #/
 3804  025720  042514  050040  051501
 3805  025726  020123  020075     043
 3806  025733     045  047105  042524  MSG40:  .ASCII  /%ENTER STALLS%READ = #/
 3807  025740  020122  052123  046101
 3808  025746  051514  051045  040505
 3809  025754  020104  020075     043
 3810
 3811  025761     045  051127  052111  MSG41:  .ASCII  /%WRITE = #/
 3812  025766  020105  020075     043
 3813
 3814  025773     045  052524  047122  MSG42:  .ASCII  /%TURN AROUND = #/
 3815  026000  040440  047522  047125
 3816  026006  020104  020075     043
 3817
 3818  026013     045  022477     043  MSG43:  .ASCII  /%?%#/
 3819
 3820  026017     045  047105  042524  MSG44:  .ASCII  /%ENTER YOZZLE STALL = #/
 3821  026024  020122  047531  055132
 3822  026032  042514  051440  040524
 3823  026040  046114  036440  021440
 3824
```

```
3825  026046  042445  051122  040440  MSG45:  .ASCII  /%ERR AMT #/
3826  026054  052115  021440
3827
3828  026060  043045  020103     043  MSG46:  .ASCII  /%FC #/
3829
3830  026065     045  040503  021440  MSG47:  .ASCII  /%CA #/
3831
3832  026072  020445  047516  041040  MSG48:  .ASCII  /%!NO BOT ON REWIND: HALT%#/
3833  026100  052117  047440  020116
3834  026106  042522  044527  042116
3835  026114  020072  040510  052114
3836  026122  021445
3837
3838  026124  047040  052117  040440  MSG49:  .ASCII  / NOT AVAIL #/
3839  026132  040526  046111  021440
3840  026140  044440  046114  043505  MSG50:  .ASCII  / ILLEGAL DRIVE TYPE #/
3841  026146  046101  042040  044522
3842  026154  042526  052040  050131
3843  026162  020105     043
3844  026165     045  042045  044522  MSG52:  .ASCII  /%%DRIVE NUMBER = #/
3845  026172  042526  047040  046525
3846  026200  042502  020122  020075
3847  026206     043
3848
3849  026207     045  047506  046522  MSG53:  .ASCII  /%FORMAT = #/
3850  026214  052101  036440  021440
3851
3852  026222  053452  020105  046524  MSG54:  .ASCII  /*WE TM#/
3853  026230     043
3854
3855  026231     052  042523  052040  MSG55:  .ASCII  /*SE TM#/
3856  026236  021515
3857
3858  026240  052040  021515          MSG56:  .ASCII  / TM#/
3859
3860  026244  047040  047117  042455  MSG57:  .ASCII  / NON-EXIST SLAVE#/
3861  026252  044530  052123  051440
3862  026260  040514  042526     043
3863  026265     045  051103  020103  MSG58:  .ASCII  /%CRC #/
3864  026272     043
3865  026273     045  051114  020103  MSG59:  .ASCII  /%LRC #/
3866  026300     043
3867  026301     052  020104     043  MSG60:  .ASCII  /*D #/
3868  026305     052  020120     043  MSG61:  .ASCII  /*P #/
3869  026311     052  020106     043  MSG62:  .ASCII  /*F #/
3870
3871  026315     045  047452  044522  MSG64:  .ASCII  /%*ORIGINAL ERROR*#/
3872  026322  044507  040516  020114
3873  026330  051105  047522  025122
3874  026336     043
3875
3876  026337     045  042522  051124  MSG65:  .ASCII  /%RETRY: #/
3877  026344  035131  021440
3878
3879  026350  020452  042523  051040  MSG66:  .ASCII  /*!SE RTRY #/
3880  026356  051124  020131     043
```

```
 3881
 3882  026363     052  042441  040522  MSG67:  .ASCII  /*!ERASE*/
 3883  026370  042523     043
 3884
 3885  026373     045  042522  042522  MSG68:  .ASCII  /%REREV: */
 3886  026400  035126  021440
 3887  026404  052045  050101  020105  MSG69:  .ASCII  /%TAPE MARK = */
 3888  026412  040515  045522  036440
 3889  026420  021440
 3890
 3891  026422  020445  047516  042040  MSG70:  .ASCII  /%!NO DRY FROM REWIND: HALT%*/
 3892  026430  054522  043040  047522
 3893  026436  020115  042522  044527
 3894  026444  042116  020072  040510
 3895  026452  052114  021445
 3896  026456  047040  047117  042455  MSG71:  .ASCII  / NON-EXIST DRIVE*/
 3897  026464  044530  052123  042040
 3898  026472  044522  042526     043  MSG72:  .ASCII  /%REFWD: */
 3899  026477     045  042522  053506  MSG73:  .ASCII  /%WTERR: */
 3900  026504  035104  021440
 3901  026510  053445  042524  051122  MSG74:  .ASCII  /%REGISTER START = */
 3902  026516  020072     043
 3903  026521     045  042522  044507  MSG75:  .ASCII  /%VECTOR = */
 3904  026526  052123  051105  051440
 3905  026534  040524  052122  036440
 3906  026542  021440
 3907  026544  053045  041505  047524  MSG76:  .ASCII  /%DEREV: */
 3908  026552  020122  020075     043
 3909  026557     045  042504  042522  MSG77:  .ASCII  /%DEFWD: */
 3910  026564  035126  021440
 3911  026570  042045  043105  042127  MSG78:  .ASCII  /%!NON-RETRYABLE WRITE ERROR: ER */
 3912  026576  020072     043
 3913  026601     045  047041  047117  MSG79:  .ASCII  /%!NON-RETRYABLE READ ERROR: ER */
 3914  026606  051055  052105  054522
 3915  026614  041101  042514  053440
 3916  026622  044522  042524  042440
 3917  026630  051122  051117  020072
 3918  026636  051105  021440
 3919  026642  020445  047516  026516  MSG100: .ASCII  /%!!END OF PASS %*/
 3920  026650  042522  051124  040531
 3921  026656  046102  020105  042522
 3922  026664  042101  042440  051122
 3923  026672  051117  020072  051105
 3924  026700  021440
 3925  026702  020445  042441  042116  MSG101: .ASCII  /%**********%*/
 3926  026710  047440  020106  040520
 3927  026716  051523  022440     043
 3928  026723     045  025052  025052  MSG102: .ASCII  /*TM02 */
 3929  026730  025052  025052  022452
 3930  026736     043
 3931  026737     052  046524  031060  MSG103: .ASCII  /*SLAVES */
 3932  026744  021440
 3933  026746  051452  040514  042526  MSG104: .ASCII  /%AUTO CONT: */
 3934  026754  020123     043
 3935  026757     045  052501  047524
 3936  026764  041440  047117  035124
```

```
 3937  026772  021440
 3938  026774  051045  041505  053117  MSG105: .ASCII  /%RECOVERED#/
 3939  027002  051105  042105     043
 3940  027007     052  020441  040502  MSG106: .ASCII  /*!!BAD TAPE OVERFLOW#/
 3941  027014  020104  040524  042520
 3942  027022  047440  042526  043122
 3943  027030  047514  021527
 3944  027034  051045  053505  047111  MSG16A: .ASCII  /%REWIND TAPE; RESTART AT BLOCK ONE#/
 3945  027042  020104  040524  042520
 3946  027050  020073  042522  052123
 3947  027056  051101  020124  052101
 3948  027064  041040  047514  045503
 3949  027072  047440  042516     043
 3950  027077     045  020441  047125  MSG107: .ASCII  /%!!UNRECOVERABLE BAD SPOT/
 3951  027104  042522  047503  042526
 3952  027112  040522  046102  020105
 3953  027120  040502  020104  050123
 3954  027126  052117
 3955  027130  041045  042101  051040          .ASCII  /%BAD RECORD LEFT ON TAPE%#/
 3956  027136  041505  051117  020104
 3957  027144  042514  052106  047440
 3958  027152  020116  040524  042520
 3959  027160  021445
 3960  027162  047045  055122  047440  MSG108: .ASCII  /%NRZ ONLY: #/
 3961  027170  046116  035131  021440
 3962  027176  020452  050041  051517  MSG109: .ASCII  /*!!POSITION LOST IN RETRY#/
 3963  027204  052111  047511  020116
 3964  027212  047514  052123  044440
 3965  027220  020116  042522  051124
 3966  027226  021531
 3967  027230  051445  051525  042520  MSG110: .ASCII  /%SUSPECT BAD TAPE#/
 3968  027236  052103  041040  042101
 3969  027244  052040  050101  021505
 3970  027252  051045  050105  040505  MSG111: .ASCII  /%REPEAT: #/
 3971  027260  035124  021440
 3972  027264  041040  042101  052040  MSG112: .ASCII  / BAD TAPE SPOTS%#/
 3973  027272  050101  020105  050123
 3974  027300  052117  022523     043
 3975
 3976  027305     045  051440  043117  MSG113: .ASCII  /% SOFT: #/
 3977  027312  035124  021440
 3978
 3979  027316  020045  040510  042122  MSG114: .ASCII  /% HARD: #/
 3980  027324  020072     043
 3981
 3982  027327     045  020441  040510  MSG115: .ASCII  /%!!HARD READ ERROR#/
 3983  027334  042122  051040  040505
 3984  027342  020104  051105  047522
 3985  027350  021522
 3986  027352  020445  047125  052111  MSG116: .ASCII  /%!UNIT IS REWINDING: TEST WILL START AT BOT#/
 3987  027360  044440  020123  042522
 3988  027366  044527  042116  047111
 3989  027374  035107  052040  051505
 3990  027402  020124  044527  046114
 3991  027410  051440  040524  052122
 3992  027416  040440  020124  047502
```

```
 3993  027424  021524
 3994  027426  057045  021507          $CNTG:  .ASCII  /%↑G#/
 3995  027432  051445  051127  020075  $MSWR:  .ASCII  /%SWR= #/
 3996  027440     043
 3997  027441     040  047040  053505  $MNEW:  .ASCII  /  NEW= #/
 3998  027446  020075     043
 3999  027451     045     043          MCRLF:  .ASCII  /%#/
 4000
 4001          027454                          .EVEN
 4002  027454  000000                  WDATA:  0                       ;WRITE BUFFER
 4003
 4004          033462                          .=.+4004
 4005  033462  000000                  RDATA:  0                       ;READ BUFFER
 4006
 4007          000001                          .END
```

```
ABLCNT  000740
ACT_RC  021126
ACRVN   000736
AMOD1   022320
AMOD1A  022330
AMOD1B  022344
AMOD1C  022500
AMOD2   022514
AMOD2A  022532
AMOD2B  022552
AMOD2X  022660
APDTS   015006
AS      000526
ASEQ    021732
ASEQCF  000742
ASEQF   000734
ASEQX   022122
ASEQXX  022140
ASEQ0   022016
ASEQ1   022022
ASEQ2   022062
ASEQ3   022074
ASEQ4   022104
BA      000514
BAER    021112
BBC     000656
BCNT    000710
BDPP    000716
BD00    001410
BD10    001430
BD20    001450
BD30    001470
BD40    001510
BD50    001530
BD60    001550
BD70    001570
BKRT    012112
BKSP    011734
BKTM    012016
BKTMO   012060
BLCNTR  000654
BPKP    000720
BP00    001210
BP10    001230
BP20    001250
BP30    001270
BP40    001310
BP50    001330
BP60    001350
BP70    001370
BTACOR  001030
BTFLG   000724
BTOV    007034
BTOVX   007174
BTOV0   007052
BTOV1   007062
BTOV2   007150
BTOV3   007166
BTPRT   007176
BTPRT1  007246
BTPT    000730
BTSTF   000726
BTUR    007252
BT00    001610
BT01    001714
BT02    002020
BT03    002124
BT04    002230
BT05    002334
BT06    002440
BT07    002544
B0      012070
B1      012136
B2      012146
CADER   021110
CC      000530
CCNTR   012172
CKSWR   024360
CKSWRR= 104004
CLLAST  015200
CLP     015310
CLPE    015334
CLPO    015344
CLP1    015356
CLP2    015416
CLP3    015430
CLO     015114
CL1     015142
CL2     015164
CL3     015252
CNTG    024422
CNTLU   024430
CONER   021114
COUNT   024354
CRCER   021124
CRCLRC  015076
CRCSV   021132
CS      000520
C1      000510
C2      000542
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    015026
DATRO   015044
DATO    014300
DATOA   014330
DATOB   014336
DATOC   014406
DATOD   014414
DATOE   014424
DATOF   014440
DAT1    014452
DAT1A   014456
DAT1B   014462
DAT1O   014612
DAT1OA  014624
DAT11   014644
DAT11A  014652
DAT12   014666
DAT12A  014676
DAT13   014712
DAT14   014722
DAT14A  014734
DAT15   014754
DAT15A  014760
DAT15B  014770
DAT2    014474
DAT3    014502
DAT3A   014510
DAT3B   014514
DAT4    014530
DAT5    014542
DAT6    014552
DAT7    014562
DAT7A   014576
DB      000532
DCHK    015472
DCHKO   015522
DEREV1  001170
DEREX   016564
DEREX1  016620
DERFL   000704
DERR    016106
DERRO   016122
DERROA  016152
DERROB  016204
DERROC  016230
DERROD  016232
DERR1   016260
DERR2   016262
DERR3   016300
DERR4   016302
DERR4A  016442
DERR4B  016510
DERR5   016546
DERR6   016560
DFX     016104
DF0     015776
DF0A    015672
DF0A0   015714
DF0A1   015730
DF0A2   015744
DF0A3   015760
DF0A4   015764
DF0B    015632
DF0B0   015654
DF0C    015612
DF0C0   015622
DF0D    015576
DF0E    015570
DF0F    015562
DF1     016010
DF2     016020
DF3     016036
DF4     016100
DOUT    024142
DOUTD   024224
DOUT1   024156
DOUT2   024204
DOUT3   024212
DPC     016742
DPCG    016750
DPC0    016756
DPC0A   017020
DPC1    017026
DPC1A   017054
DPC2    017116
DPC2A   017060
DPC2B   017102
DPC3    017150
DPPRT   017216
DPPRTX  017352
DPPRT0  017266
DPPRT1  017312
DPPRT2  017326
DROP    016732
DRPK    016720
DRPKF   016632
DRP1    001010
DRP2    001012
DRP3    001014
DRP4    001016
DRP5    001020
DRP6    001022
DRP7    001024
DRP8    001026
DRVER   021116
DS      000522
DSJP    014062
DSO     014072
DSOA    014154
DSOB    014152
DSOC    014114
DS1     014204
DS2     014224
DS2A    014234
DS3     014240
DS4     014250
DT      000536
DVN     000550
DOFL    014450
EMADDR  000652
EOPBI   000744
EOTCO   002650
EOTREC  000660
ER      000524
ERCHK   017364
ERPT    020212
ERPTG   020246
ERPTG1  020314
ERPTT   020232
ERPT0   020330
ERPT1   020376
ERPT2   020434
ERPT3   020472
ERPT4   020534
ERPT5   020556
ERPT5A  020640
ERPT6   020702
ERPT7   020754
ERPX    021012
ERPX0   020756
ERPX1   021054
ERPX2   021106
ERSAV   000722
ERTFL   000732
ER0     017370
ER0A    017440
ER0B    017406
ER1     017446
ER1O    020154
```

```
ER2     017452
ER2A    017516
ER2A0   017466
ER2A1   017506
ER2B    017534
ER2C    017560
ER2D    017576
ER2E    017624
ER3     017632
ER3A    017674
ER3A1   017732
ER3B    017742
ER4     017746
ER4A    020014
ER4A1   020004
ER6     020020
ER6A    020104
ER7     020136
EXCRC   015464
EXLRC   015466
FC      000516
FCER    021120
FCSAV   000632
FMCNT   000556
FREX    021172
FRPRT   021134
FRQ     021164
HDRFL   000664
HRDS    022142
HRDS0   022202
HRDS1   022210
HRDS2   022230
HRDS3   022252
HRDS4   022264
INTRF   000566
LRCER   021122
LRCSV   021130
MCRLF   027451
MR      000534
MSG1    024716
MSG10   024777
MSG100  026702
MSG101  026723
MSG102  026737
MSG103  026746
MSG104  026757
MSG105  026774
MSG106  027007
MSG107  027077
MSG108  027162
MSG109  027176
MSG11   025004
MSG110  027230
MSG111  027252
MSG112  027264
MSG113  027305
MSG114  027316
MSG115  027327
MSG116  027352
MSG12   025020
MSG13   025034
MSG14   025042
MSG15   025047
MSG16   025100
MSG16A  027034
MSG17   025103
MSG2    024723
MSG20   025106
MSG21   025121
MSG22   025146
MSG23   025173
MSG23A  025201
MSG23B  025206
MSG23C  025213
MSG23D  025220
MSG23E  025226
MSG23F  025233
MSG23G  025240
MSG23H  025245
MSG23I  025252
MSG23J  025257
MSG23K  025264
MSG23L  025271
MSG23M  025276
MSG24   025303
MSG25   025323
MSG26   025343
MSG27   025354
MSG28   025365
MSG3    024730
MSG30   025367
MSG31   025455
MSG32   025553
MSG33   025574
MSG34   025610
MSG35   025623
MSG36   025644
MSG37   025670
MSG38   025713
MSG4    024735
MSG40   025733
MSG41   025761
MSG42   025773
MSG43   026013
MSG44   026017
MSG45   026046
MSG46   026060
MSG47   026065
MSG48   026072
MSG49   026124
MSG5    024742
MSG50   026140
MSG52   026165
MSG53   026207
MSG54   026222
MSG55   026231
MSG56   026240
MSG57   026244
MSG58   026265
MSG59   026273
MSG6    024747
MSG60   026301
MSG61   026305
MSG62   026311
MSG64   026315
MSG65   026337
MSG66   026350
MSG67   026363
MSG68   026373
MSG69   026404
MSG7    024754
MSG70   026422
MSG71   026456
MSG72   026477
MSG73   026510
MSG74   026521
MSG75   026544
MSG76   026557
MSG77   026570
MSG78   026601
MSG79   026642
MSG8    024761
MSG9    024771
MTC1    000672
MTINT   021714
MTINTA  021720
NOP   = 000240
NRTP    011136
NRZOF   000650
OCTP    023704
OCTPE   023712
OCTPE1  023720
OCTPG   024076
OCTPG0  024114
OCTPG1  024120
OCTPP = 104002
OCTP0   023742
OCTP1   023764
OCTP2   023772
OCTP3   024062
OFL     024140
OUT     024646
PAPRT   022662
PAPRTA  023036
PAPRTB  023052
PAPRTC  023066
PAPRTD  023074
PAPRTY  023170
PAPRT0  022764
PAPRT1  023140
PAPRT2  023172
PAPRT3  023174
PARCNT  015470
PARS    014276
PATRN   000560
PATS    014274
PC    =%000007
PFLG    000670
PICK    017152
PIK1    000770
PIK2    000772
PIK3    000774
PIK4    000776
PIK5    001000
PIK6    001002
PIK7    001004
PIK8    001006
PRB     000622
PRS     000620
PSW     000604
RANBAS  000624
RANG    023246
RANSAV  000626
RANSET  004252
RCNT    000554
RCNTR   012232
RCSAV   000630
RDA     007654
RDATA   033462
RDCMD   000562
RDERR1  001150
RDER1   001110
RDER2   001112
RDER3   001114
RDER4   001116
RDER5   001120
RDER6   001122
RDER7   001124
RDER8   001126
RDEX    010562
RDFL    015074
RDRTG   010656
RDRTX   011134
RDRTY   010570
RDRT0   010602
RDRT1   010634
RDRT1A  010632
RDRT1B  010652
RDRT2   010734
RDRT3   010756
RDRT4   010762
RDRT5   010764
RDRT5A  011024
RDRT5B  011034
RDRT6   011064
RDRT7   011130
RDSW    024356
RDX     010566
RD0     007712
RD1     007760
RD1A    007774
RD1B    010002
RD1D    010010
RD10    010500
RD11    010516
RD2     010014
RD2A    010034
RD2B    010044
RD3     010074
RD4     010124
RD4A    010222
RD4A0   010254
RD4A1   010276
RD4A2   010320
RD4B    010326
RD4C    010332
RD4D    010362
RD4E    010366
RD5     010376
RD6     010420
RD7     010444
RD7A    010474
READ    007614
REGS    000544
REOT    004304
REOTC   005012
REOTX   004746
REOTXX  005002
REOT1   004320
REOT1A  004354
REOT1B  004376
REOT1C  004416
REOT1D  004422
REOT1E  004442
REOT1F  004412
```

REOT2	004464	R5	=%000005	TAPG3F	021324	TTIN1	023500	WTER3	001074
REOT2A	004524	SCVFL	014032	TAPG4	021514	TTIN2	023514	WTER4	001076
REOT3	004552	SERFL	000706	TAPG5	021530	TTOUT	023532	WTER5	001100
REOT4	004602	SN	000540	TAPG6	021572	TTOUTT=	104000	WTER6	001102
REOT5	004616	SNPG	024326	TAPG7	021604	TTR	023300	WTER7	001104
REOT6	004662	SNPT	024250	TBELA	023664	TTRO	023306	WTER8	001106
REOT7	004720	SP	=%000006	TBELL	023660	TTR1	023332	WTM	005730
RETRY	000602	SPFLG	000570	TCRLF	023574	TTR1A	023346	WTMO	005772
RETURN	021662	STAL	000666	TCRLFA	023612	TTR1B	023362	WTM1	006034
RFHARD	002730	STALL	012162	TEMPST	024352	TTR2	023420	WTM2	006044
RFSOFT	002670	START	003026	TEMP1	000642	TTR3	023430	WTM3	006054
RPCNT	000700	STARTA	003060	TEMP2	000644	TTR4	023440	WTM4	006134
RRHARD	002750	STARTB	003064	TEMP3	000646	TTR5	023442	WTM4A	006172
RRSOFT	002710	STARTC	003050	TEX	023656	UDES	000552	W0	005430
RSEQ	007266	STARTD	003150	TIB	000640	UNP	000674	W1	005472
RSEX	007440	STARTE	003142	TINER	023444	UNX	000766	W2	005514
RSF	007342	STARTF	003122	TINF	000634	UN1	000746	W3	005530
RSFR	007446	STARTO	003074	TINP	012256	UN2	000750	W3A	005552
RSFRX	007606	START1	003342	TINPA	012266	UN3	000752	W4	005636
RSFRO	007500	START2	003474	TINPB	012306	UN4	000754	W4A	005664
RSFR1	007512	START3	003510	TINPBO	012472	UN5	000756	W5	005700
RSFR2	007544	START4	003524	TINPB1	012332	UN6	000760	W6	005716
RSFO	007400	START5	004170	TINPC	012520	UN7	000762	XOR	015436
RSF1	007414	START6	004206	TINPX	014024	UN8	000764	XORS	015462
RSR	007320	START7	004240	TINPO	012626	UPS	000714	YOZ	011152
RSTAL	000572	START8	004246	TINPOA	012704	VECT	000546	YOZA	011220
RTCNT	000702	STAROA	003274	TINPOB	012722	WC	000512	YOZB	011226
RTRN	000662	STAROB	003324	TINPOC	012762	WDATA	027454	YOZC	011246
RTYFL	000712	STAR1A	003354	TINPOD	013004	WEX	006176	YOZCO	011336
RTY1	001050	STAR1B	003400	TINPOE	013030	WRITE	005404	YOZC1	011252
RTY2	001052	STAR1C	003440	TINP1	013046	WRTE	005424	YOZC2	011360
RTY3	001054	STAR4A	003612	TINP2	013124	WRTSB	006640	YOZD	011400
RTY4	001056	STAR4O	003540	TINP2A	013202	WRTSBO	006712	YOZDO	011502
RTY5	001060	STAUT	003032	TINP2B	013260	WRTSB1	006726	YOZD1	011526
RTY6	001062	STAUTO	003272	TINP2C	013306	WRTSB2	006736	YOZD2	011474
RTY7	001064	STOP	024650	TINP3	013326	WRTSB3	007012	YOZD3	011520
RTY8	001066	STOPP =	104006	TINP4	013656	WRTY	006242	YOZD4	011454
RWND	005014	STP	003706	TKB	000612	WRTYR	006274	YOZE	011532
RWNDA	005026	STPX	004154	TKS	000610	WRTYTM	006270	YOZF	011562
RWNDX	005350	SUSWR	003162	TMEX	000564	WRTYO	006250	YOZFO	011566
RWNDO	005064	SWR	000606	TMFLG	000676	WRTY1	006350	YOZG	011660
RWND1	005142	SWREG	000176	TOB	000636	WRTY2	006352	YOZH	011724
RWND1A	005202	TABLE	024704	TOG	023642	WRTY2A	006412	YOZI	011710
RWND2	005222	TAPG	021174	TPB	000616	WRTY2B	006430	YOZO	011172
RWND3	005232	TAPGO	021206	TPOS	014034	WRTY3	006474	YSTAL	000600
RWND4	005264	TAPG1	021250	TPOS1	014046	WRTY3A	006512	$CNTG	027426
RWND5	005304	TAPG2	021262	TPS	000614	WRTY4	006600	$MNEW	027441
RWND6	005324	TAPG3	021272	TRAP30	024656	WRTY5	006634	$MSWR	027432
R0	=%000000	TAPG3A	021342	TSTAL	000576	WRW	006216	$READ	024454
R1	=%000001	TAPG3B	021356	TTIN	023460	WRWX	006240	.	= 033464

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

```
*DSKM:DZTUAF,DZTUAF/SOL←DSKM:DZTUAF.P11
RUN-TIME:  11 25 2 SECONDS
RUN-TIME RATIO: 109/40=2.7
CORE USED:  7K  (13 PAGES)
```

F10

Spooler runtime 15 Seconds, 66 KCS, 450 disk reads, 5 disk writes, 121 pages