TM02/TU45

DATA RELIABILITY PROGRAM

CZTUIA0

AH-E470A-MC

FICHE 1 OF 1

JUL 1978

digital

MADE IN USA

```
        .NLIST  SEQ,LOC,BIN
        .REM_
```

IDENTIFICATION

PRODUCT CODE: AC-E469A-MC

PRODUCT TITLE: CZTUIAO TM02/TU45 DATA RELIABILITY PROGRAM

DATE CREATED: 25 MAY 1978

MAINTAINER: COMPUTER SPECIAL SYSTEMS

AUTHOR: R.B. BARNES/R. J. COLLINS

TABLE OF CONTENTS

(PAGE 1)

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 PROCESSER
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

(PAGE 2)

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, AND TAPE MARK.

(PAGE 3)

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.

(PAGE 4)

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, NRZ1
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.

(PAGE 5)

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. (ZZ-CZTUNAO) 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.

(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. SINGLE PASS = 0
F. READ STALL = 1
G. WRITE STALL = 1
H. 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:

TU45 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)
SINGLE PASS=(0)

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

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

NO TAPE MARKS WILL BE WRITTEN.

(PAGE 8)

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 ZZ-CZTUNA0)
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

(PAGE 9)

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.

(PAGE 10)

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.

(PAGE 11)

8. CONSOLE SWITCH SETTINGS

THE CONSOLE SWITCHES ARE USED TO SET UP THE TEST CYCLE DESIRED, TO GENERATE RANDOM VALUES, AND TO CONTROL ERROR RESPONSES. THE SWITCHES SHOULD BE SET IN THE DESIRED MANNER BEFORE PRESSING THE START SWITCH BECAUSE THEY ARE ALL DYNAMIC AND WILL RUN THE PROGRAM IN ANY CONFIGURATION. ALL SWITCHES SET TO ZERO(0) IS NORMAL. IF A CONSOLE SWITCH REGISTER IS NOT PRESENT, THE SWITCHES ARE NOT DYNAMIC AND MUST BE SET BEFORE STARTING, USING THE FOLLOWING PROCEDURE:

A) LOAD ADDRESS 606(8) LABELLED "SWR"
B) DEPOSIT THE VALUE 176(8)
C) LOAD ADDRESS 176(8)
D) DEPOSIT THE DESIRED SWITCH VALUE.

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

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

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

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

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

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

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

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

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

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

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

SW4: 1=DO WRITE/READ RETRIES

```
                        0=DO NOT RETRY

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

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

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

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

(PAGE 12)

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

(PAGE 13)

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.

(PAGE 15)

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
WE = WRITE STATUS ERROR
RE = READ STATUS ERROR
SE = SPACE ERROR
TM = TAPE MARK
F = FORWARD
R = REVERSE
CS1 = RH/TU45 CONTROL REGISTER
WC = RH WORD COUNT
BA = RH BUS ADDRESS
FC = TU45 FRAME COUNT
CS2 = RH CONTROLLER STATUS
DS = TU45 DRIVE STATUS
ER = TU45 ERROR REGISTER
AS = ATTENTION SUMMARY
CK = TU45 CHECK CHARACTER
DB = RH DATA BUFFER
MR = TU45 MAINTENENCE REGISTER
DT = TU45 DRIVE TYPE
SN = TU45 SERIAL NUMBER
TC = TU45 TEST CONTROL
*F = DATA FORMAT
*P = PARITY
*D = DENSITY
*PATRN = DATA PATTERN NUMBER (R = RANDOM)

(PAGE 18)

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
REVcRSE 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
```

(PAGE 20)

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
```

(PAGE 21)

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
```

(PAGE 22)

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
```

(PAGE 23)

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

(PAGE 24)

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:

```
        TU45 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

(PAGE 26)

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

-

```
1312                                    .LIST   SEQ,LOC,BIN
1313                                            .TITLE TM02/TU45 DATA RELIABILITY PROGRAM
1314                                            ;ZZ-CZTUIA0
1315                                            ;15 MAY 1978
1316                                            ;R. BARNES/R. J. COLLINS
1317
1318                                            .ENABLE ABS
1319
1320                                            ;CONSOLE SWITCHES**********
1321
1322                                            ;SW15: 1=STOP ON ERROR
1323                                            ;       0=CONTINUE ON ERROR
1324
1325                                            ;SW14: 1=PRINT READ/WRITE STATS
1326                                            ;       0=DO NOT PRINT STATS
1327
1328                                            ;SW13: 1=DO NOT CHECK DATA
1329                                            ;       0=CHECK DATA
1330                                            ;SW12: 1=DO NOT CHECK WRITE ERRORS
1331                                            ;       0=CHECK WRITE ERRORS
1332                                            ;SW11: 1=DO NOT CHECK READ ERRORS
1333                                            ;       0=CHECK READ ERRORS
1334                                            ;SW10: 1=DO NOT PRINT ERRORS
1335                                            ;       0=PRINT ERRORS
1336
1337                                            ;SW9:  1=REWIND TAPE
1338                                            ;       0=DO NOT REWIND
1339
1340                                            ;SW8:  1=USE RANDOM DATA
1341                                            ;       0=USE FIXED DATA PATTERN
1342                                            ;SW7:  1=USE RANDOM CHARACTER COUNT
1343                                            ;       0=USE FIXED CHAR COUNT
1344                                            ;SW6:  1=USE RANDOM RECORD COUNT
1345                                            ;       0=USE FIXED RECORD COUNT
1346
1347                                            ;SW5:  1=YOZZLE ON CURRENT RECORD
1348                                            ;       0=DO NOT YOZZLE
1349
1350                                            ;SW4:  1=DO BOTH READ AND WRITE RETRIES
1351                                            ;       0=INHIBIT RETRIES
1352
1353                                            ;SW3:  1=DO NOT READ FORWARD
1354                                            ;       0=READ FORWARD
1355                                            ;SW2:  1=DO NOT READ REVERSE
1356                                            ;       0=READ REVERSE
1357                                            ;SW1:  1=READ FORWARD FIRST
1358                                            ;       0=READ REVERSE FIRST
1359                                            ;SW0:  1=DO NOT WRITE
1360                                            ;       0=WRITE
```

```
                        ;TU45 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;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 ;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
                    005726              POPSP=5726
                    022626              POPPOP=22626

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

                    000000              .=0
                    000200              .REPT 200
                                        .+2
                                        HALT
                                        .ENDR

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

                    000060              .=60
000060  022142                          TTINT                   ;TTY INTERRUPT HANDLER ADDRESS
000062  000000                          0

                                        ;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  014774                  CLR     RDFL
000214  000167  002640                  JMP     STARTA          ;USE FIXED PARAMETERS; NEW DATA

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

                    000224              .=224
000224  022216                          MTINT                   ;MAG TAPE INTERRUPT HANDLER ADDRESS
000226  000340                          340

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

                    000240              .=240
000240  005267  000472                  INC     ASEQF           ;SET AUTO SEQUENCE FLAG
000244  000167  002562                  JMP     STAUT           ;GO TO START OF AUTO SEQUENCE
```

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

                    000300              .=300
000300  005267  013642                  INC     SCVFL                   ;SET SHORT CONVERSATION FLAG
000304  000167  002516                  JMP     START                   ;ENTER SHORT PARAMETER LIST

                    000510              .=510
                                        ;TU45 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 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
```

```
1512
1513                                    ;FLAGS AND COUNTERS**********
1514
1515  000634  000000          MOLSW:  0         ;MOL ERROR PRINT SWITCH
1516  000636  000000          TINF:   0         ;TTY ENTERY FLAG
1517  000640  000000          TOB:    0         ;TTY OUTPUT BUFFER
1518  000642  000000          TIB:    0         ;TTY INPUT BUFFER
1519  000644  000000          TEMP1:  0         ;TEMP STORAGE
1520  000646  000000          TEMP2:  0         ;TEMP STORAGE
1521  000650  000000          TEMP3:  0         ;TEMP STORAGE
1522  000652  000000          NRZOF:  0         ;NRZ ONLY FLAG
1523  000654  000000          EMADDR: 0         ;ERROR MSG ADDRESS STORAGE
1524  000656  000000          BLCNTR: 0         ;BLOCK COUNTER
1525  000660  000000          BBC:    0         ;BAD RECORD COUNTER
1526  000662  000000          EOTREC: 0         ;EOT FLAG
1527  000664  000000          RTRN:   0         ;INTERRUUPT RETURN STORAGE
1528  000666  000000          HDRFL:  0         ;HEADER FLAG
1529  000670  000000          STAL:   0         ;DELAY STORAGE
1530  000672  000000          PFLG:   0         ;PRINT FLAG
1531  000674  000000          MTC1:   0         ;MAG TAPE CONT REGISTER BUFFER
1532  000676  000000          UNP:    0         ;UNIT TABLE POINTER
1533  000700  000000          TMFLG:  0         ;TAPE MARK FLAG
1534  000702  000000          RPCNT:  0         ;REPEAT COUNTER
1535  000704  000000          RTCNT:  0         ;RETRY COUNTER
1536  000706  000000          DERFL:  0         ;DATA ERROR FLAG
1537  000710  000000          SERFL:  0         ;STATUS ERROR FLAG
1538  000712  000000          BCNT:   0         ;BIT COUNTER
1539  000714  000000          RTYFL:  0         ;RETRY FLAG
1540  000716  000000          UPS:    0         ;UNIT POINTER SAVE
1541  000720  000000          BDPP:   0         ;BITS DROPPED POINTER
1542  000722  000000          BPKP:   0         ;BITS PICKED POINTER
1543  000724  000000          ERSAV:  0         ;ERROR SAVE LOC
1544  000726  000000          BTFLG:  0         ;BAD TAPE FLAG
1545  000730  000000          BTSTF:  0         ;STATISTIC PRINT FLAG
1546  000732  000000          BTPT:   0         ;BAD TAPE POINTER
1547  000734  000000          ERTFL:  0         ;ERASE FLAG
1548  000736  000000          ASEQF:  0         ;AUTO SEQ FLAG
1549  000740  000000          ADRVN:  0         ;UTO SEQ DRIVE NUMBER
1550  000742  000000          ABLCNT: 0         ;AUTO BLOCK COUNTER
1551  000744  000000          ASEQCF: 0         ;AUTO SEQ CONTINUOUS FLAG
```

```
1552
1553                                    ;UNIT ORDER AND DESCRIPTION TABLE **********
1554
1555  000746  000000            UN1:    0               ;THIS TABLE IS LOADED
1556  000750  000000            UN2:    0               ;WITH UNIT NUMBERS AND
1557  000752  000000            UN3:    0               ;THEIR DESCRIPTIONS IN
1558  000754  000000            UN4:    0               ;THE ORDER THAT THEY
1559  000756  000000            UN5:    0               ;WILL BE TESTED
1560  000760  000000            UN6:    0
1561  000762  000000            UN7:    0
1562  000764  000000            UN8:    0
1563  000766  177777            UNX:    -1
1564
1565                                    ;UNIT DROPS AND PICKS POINTERS*********
1566
1567  000770  001210            PIK1:   BP00
1568  000772  001230            PIK2:   BP10
1569  000774  001250            PIK3:   BP20
1570  000776  001270            PIK4:   BP30
1571  001000  001310            PIK5:   BP40
1572  001002  001330            PIK6:   BP50
1573  001004  001350            PIK7:   BP60
1574  001006  001370            PIK8:   BP70
1575  001010  001410            DRP1:   BD00
1576  001012  001430            DRP2:   BD10
1577  001014  001450            DRP3:   BD20
1578  001016  001470            DRP4:   BD30
1579  001020  001510            DRP5:   BD40
1580  001022  001530            DRP6:   BD50
1581  001024  001550            DRP7:   BD60
1582  001026  001570            DRP8:   BD70
1583
1584                                    ;UNIT BAD TAPE POINTERS**********
1585
1586  001030  001610            BTADDR: BT00
1587  001032  001714                    BT01
1588  001034  002020                    BT02
1589  001036  002124                    BT03
1590  001040  002230                    BT04
1591  001042  002334                    BT05
1592  001044  002440                    BT06
1593  001046  002544                    BT07
1594
1595                                    ;UNIT WRITE RETRY COUNTER**********
1596
1597  001050  000000            RTY1:   0
1598  001052  000000            RTY2:   0
1599  001054  000000            RTY3:   0
1600  001056  000000            RTY4:   0
1601  001060  000000            RTY5:   0
1602  001062  000000            RTY6:   0
1603  001064  000000            RTY7:   0
1604  001066  000000            RTY8:   0
1605
1606                                    ;UNIT WRITE ERRORS**********
1607
```

```
 1608  001070  000000                  WTER1:  0
 1609  001072  000000                  WTER2:  0
 1610  001074  000000                  WTER3:  0
 1611  001076  000000                  WTER4:  0
 1612  001100  000000                  WTER5:  0
 1613  001102  000000                  WTER6:  0
 1614  001104  000000                  WTER7:  0
 1615  001106  000000                  WTER8:  0
 1616
 1617                                  ;UNIT READ FORWARD ERRORS**********
 1618
 1619  001110  000000                  RDER1:  0
 1620  001112  000000                  RDER2:  0
 1621  001114  000000                  RDER3:  0
 1622  001116  000000                  RDER4:  0
 1623  001120  000000                  RDER5:  0
 1624  001122  000000                  RDER6:  0
 1625  001124  000000                  RDER7:  0
 1626  001126  000000                  RDER8:  0
 1627
 1628                                  ;UNIT DATA ERRORS FORWARD**********
 1629
 1630  001130  000000                  DATER1: 0
 1631  001132  000000                          0
 1632  001134  000000                          0
 1633  001136  000000                          0
 1634  001140  000000                          0
 1635  001142  000000                          0
 1636  001144  000000                          0
 1637  001146  000000                          0
 1638
 1639                                  ;UNIT READ REVERSE ERRORS**********
 1640
 1641  001150  000000                  RDERR1: 0
 1642  001152  000000                          0
 1643  001154  000000                          0
 1644  001156  000000                          0
 1645  001160  000000                          0
 1646  001162  000000                          0
 1647  001164  000000                          0
 1648  001166  000000                          0
 1649
 1650                                  ;UNIT DATA ERRORS REVERSE**********
 1651
 1652  001170  000000                  DEREV1: 0
 1653  001172  000000                          0
 1654  001174  000000                          0
 1655  001176  000000                          0
 1656  001200  000000                          0
 1657  001202  000000                          0
 1658  001204  000000                          0
 1659  001206  000000                          0
```

```
                                        ;DROPS + 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
```

```
 1696
 1697                                   ;UNIT BAD TAPE COUNTER:16 PER SLAVE**********
 1698
 1699  001610  000000          BT00:    0
 1700          001714                   .=.+102
 1701  001714  000000          BT01:    0
 1702          002020                   .=.+102
 1703  002020  000000          BT02:    0
 1704          002124                   .=.+102
 1705  002124  000000          BT03:    0
 1706          002230                   .=.+102
 1707  002230  000000          BT04:    0
 1708          002334                   .=.+102
 1709  002334  000000          BT05:    0
 1710          002440                   .=.+102
 1711  002440  000000          BT06:    0
 1712          002544                   .=.+102
 1713  002544  000000          BT07:    0
 1714          002650                   .=.+102
 1715
 1716                                   ;UNIT END OF TAPE COUNTERS 1 PER SLAVE*******
 1717
 1718  002650  000000          EOTC0:   0
 1719  002652  000000                   0
 1720  002654  000000                   0
 1721  002656  000000                   0
 1722  002660  000000                   0
 1723  002662  000000                   0
 1724  002664  000000                   0
 1725  002666  000000                   0
 1726
 1727                                   ;UNIT READ FORWARD SOFT ERROR*******
 1728
 1729  002670  000000          RFSOFT:  0
 1730  002672  000000                   0
 1731  002674  000000                   0
 1732  002676  000000                   0
 1733  002700  000000                   0
 1734  002702  000000                   0
 1735  002704  000000                   0
 1736  002706  000000                   0
 1737
 1738                                   ;UNIT READ REVERSE SOFT ERROR*******
 1739
 1740  002710  000000          RRSOFT:  0
 1741  002712  000000                   0
 1742  002714  000000                   0
 1743  002716  000000                   0
 1744  002720  000000                   0
 1745  002722  000000                   0
 1746  002724  000000                   0
 1747  002726  000000                   0
 1748
```

```
1749
1750                                    ;UNIT READ FORWARD HARD ERROR*******
1751
1752  002730  000000            RFHARD: 0
1753  002732  000000                    0
1754  002734  000000                    0
1755  002736  000000                    0
1756  002740  000000                    0
1757  002742  000000                    0
1758  002744  000000                    0
1759  002746  000000                    0
1760
1761                                    ;UNIT READ REVERSE HARD ERROR*******
1762
1763  002750  000000            RRHARD: 0
1764  002752  000000                    0
1765  002754  000000                    0
1766  002756  000000                    0
1767  002760  000000                    0
1768  002762  000000                    0
1769  002764  000000                    0
1770  002766  000000                    0
1771
1772                                    ;DATA PATTERN GENERATORS************
1773
1774  002770  002770            DATBL:  .                       ;ENTRY TABLE
1775  002772  014414            DATA0:  DAT0                    ;EXTERNAL INPUT FROM H/S READER(SEE ZZ-CZTUNA0)
1776  002774  014566            DATA1:  DAT1                    ;ALL ONES
1777  002776  014610            DATA2:  DAT2                    ;ALL ZEROS
1778  003000  014616            DATA3:  DAT3                    ;WALKING ONE
1779  003002  014644            DATA4:  DAT4                    ;WALKING ZERO
1780  003004  014656            DATA5:  DAT5                    ;ALTERNATING ONE/ZERO
1781  003006  014666            DATA6:  DAT6                    ;ALTERNATING ZERO/ONE
1782  003010  014676            DATA7:  DAT7                    ;ALTERNATING ONE/ZERO IN ALTERNATING CHARACTERS
1783  003012  014726            DATA10: DAT10                   ;WALKING ONE/ALL ONE IN ALTERNATING CHARACTERS
1784  003014  014760            DATA11: DAT11                   ;ALL BITS 0-377
1785  003016  015002            DATA12: DAT12                   ;ALL BITS 377-0
1786  003020  015026            DATA13: DAT13                   ;ALTERNATING CHARACTERS  0 AND 377
1787  003022  015036            DATA14: DAT14                   ;WALKING ZERO/ALL ZERO IN ALTERNATING CHARACTERS
1788  003024  015070            DATA15: DAT15                   ;AUTO SEQUENCE PATTERN 0,0,-1,-1,-1,0,0
1789
```

```
1790
1791                                    .EVEN
1792                                    ;*********************************************************
1793                                    ;PROGRAM START AND SEQUENCE FORMATTER:
1794                                    ;
1795                                    ;THIS ROUTINE IS USED TO PERFORM ALL HOUSEKEEPING,
1796                                    ;DECIDE WHICH TRANSPORT TO TEST AND ITS AVAILABILITY,
1797                                    ;LOAD THE WRITE BUFFER WITH THE SELECTED DATA PATTERN,
1798                                    ;GENERATE ANY RANDOM NUMBER AND THEN EXECUTE
1799                                    ;THE TEST CYCLE REQUESTED BY THE SWITCH SETTING.
1800                                    ;AT THE END OF THE TEST CYCLE THE NEXT UNIT IS SELECTED
1801                                    ;AND CHECKED FOR AVAILABILITY AND THE TEST CYCLE IS
1802                                    ;EXECUTED ON IT.
1803                                    ;THE READ WRITE STATS MAY BE PRINTED AT THE END OF
1804                                    ;EACH TEST CYCLE VIA CONSOLE SWITCH FOURTEEN (14).
1805                                    ;*********************************************************
1806
1807
1808  003026  005067  175704          START:  CLR     ASEQF           ;CLEAR AUTO SEQUENCE FLAG
1809  003032  012767  000001  175576  STAUT:  MOV     #1,TINF         ;SET TTY ENTRY FLAG
1810  003040  005067  012144                  CLR     RDFL            ;CLEAR RANDOM DATA FLAG
1811  003044  000167  000014                  JMP     STARTB
1812  003050  005067  175562          STARTC: CLR     TINF            ;CLEAR TTY INPUT FLAG
1813  003054  000167  000064                  JMP     STARTD
1814  003060  005067  175552          STARTA: CLR     TINF            ;CLEAR TTY ENTRY FLAG
1815  003064  012700  000640          STARTB: MOV     #TOB,R0
1816  003070  012701  000037                  MOV     #37,R1
1817  003074  005020                  STARTO: CLR     (R0)+           ;CLEAR FLAGS AND COUNTERS
1818  003076  005301                          DEC     R1
1819  003100  001375                          BNE     STARTO
1820  003102  012706  000500                  MOV     #500,SP         ;SET STACK POINTER
1821  003106  004767  001116                  JSR     PC,RANSET       ;GO RESET RANDOM BASE
1822  003112  012700  001050                  MOV     #RTY1,R0
1823  003116  012701  000750                  MOV     #750,R1
1824  003122  005020                  STARTF: CLR     (R0)+           ;CLEAR STATISTIC COUNTERS
1825  003124  005301                          DEC     R1
1826  003126  001375                          BNE     STARTF
1827  003130  012767  177777  011252          MOV     #-1,PATS        ;PRESET PATTERN
1828  003136  012767  000001  175512  STARTE: MOV     #1,BLCNTR       ;PRESET BLOCK COUNTER
1829  003144  012706  000500          STARTD: MOV     #500,SP
1830  003150  012777  000340  175426          MOV     #340,@PSW
1831  003156  004767  007152                  JSR     PC,TINP         ;GO GET PARAMETERS FROM TTY
1832  003162  012777  000040  175330          MOV     #40,@CS         ;INITIALIZE
1833  003170  005000                  STAUTO: CLR     R0              ;POINT TO FIRST ENTRY
1834  003172  005160  000746          STAROA: COM     UN1(R0)         ;SEE IF LAST ENTRY
1835  003176  001411                          BEQ     STAROB          ;IF SO: BR
1836  003200  005160  000746                  COM     UN1(R0)
1837  003204  042760  100000  000746          BIC     #100000,UN1(R0) ;CLEAR EOT FLAG
1838  003212  062700  000002                  ADD     #2,R0           ;POINT TO NEXT UNIT ENTRY
1839  003216  000167  177750                  JMP     STAROA          ;CONTINUE CLEARING
1840  003222  005160  000746          STAROB: COM     UN1(R0)
1841  003226  016703  001550                  MOV     REOTC,R3
1842  003232  000303                          SWAB    R3
1843  003234  110367  001542                  MOVB    R3,REOTC        ;RESTORE EOT CNTR
1844  003240  005067  175370                  CLR     MOLSW           ;;RESET OFFLINE SW
1845  003244  012777  000100  175336  START1: MOV     #100,@TKS       ;SET TTY INTERRUPT ENABLE
```

```
1846  003252  016700  175420            MOV     UNP,R0          ;R0 = UNIT TABLE POINTER
1847  003256  005160  000746  STAR1A:  COM     UN1(R0)
1848  003262  001407                    BEQ     STAR1B          ;IF LAST UNIT IN STRING: BR
1849  003264  005160  000746            COM     UN1(R0)
1850  003270  016067  000746  175254    MOV     UN1(R0),UDES    ;LOAD NEXT UNIT DESCRIPTION
1851  003276  000167  000124            JMP     START4
1852  003302  005267  175350  STAR1B:  INC     BLCNTR          ;BUMP BLOCK COUNTER
1853  003306  005767  175424            TST     ASEQF           ;SEE IF AUTO SEQ
1854  003312  001413                    BEQ     STAR1C          ;IF NOT: BR
1855  003314  026767  175336  175420    CMP     BLCNTR,ABLCNT   ;SEE IF DONE SEQ
1856  003322  001007                    BNE     STAR1C          ;IF NOT: BR
1857  003324  005160  000746            COM     UN1(R0)         ;RESET UNIT TABLE TERMINATOR
1858  003330  005067  175322            CLR     BLCNTR          ;RESET BLOCK CNTR
1859  003334  005067  175336            CLR     UNP             ;RESET UNIT POINTER
1860  003340  000207                    RTS     PC              ;RETURN TO AUTO SEQ
1861  003342  005067  175330  STAR1C:  CLR     UNP
1862  003346  005160  000746            COM     UN1(R0)
1863  003352  005000                    CLR     R0
1864  003354  016067  000746  175170    MOV     UN1(R0),UDES    ;LOAD FIRST UNIT DESCRIPTION
1865  003362  032777  000200  175216    BIT     #200,@SWR       ;SEE IF RANDOM RECORD SIZE
1866  003370  001402                    BEQ     START2          ;IF NOT: BR
1867  003372  004767  006652            JSR     PC,CCNTR        ;GO GENERATE RANDOM RECORD SIZE
1868  003376  032777  000400  175202  START2:  BIT     #400,@SWR       ;SEE IF RANDOM DATA
1869  003404  001402                    BEQ     START3          ;IF NOT: BR
1870  003406  004767  011530            JSR     PC,DATR         ;GO GENERATE RANDOM DATA
1871  003412  032777  000100  175166  START3:  BIT     #100,@SWR       ;SEE IF RANDOM RECORD COUNT
1872  003420  001402                    BEQ     START4          ;IF NOT: BR
1873  003422  004767  006662            JSR     PC,RCNTR        ;GO GENERATE RANDOM RECORD COUNT
1874  003426  032760  100000  000746  START4:  BIT     #100000,UN1(R0) ;SEE IF REACHED EOT
1875  003434  001402                    BEQ     STAR40          ;IF NOT: BR
1876  003436  000167  000554            JMP     START7          ;ELSE GO TO NEXT UNIT
1877  003442  016777  175102  175050  STAR40:  MOV     DVN,@CS         ;SET DRIVE NUMBER
1878  003450  016777  175076  175064    MOV     UDES,@C2        ;SET UNIT NUMBER
1879  003456  032777  000200  175036    BIT     #200,@DS        ;SEE IF UNIT AVAIL
1880  003464  001014                    BNE     STAR4A          ;IF SO: BR
1881  003466  005367  175176            DEC     STAL
1882  003472  001355                    BNE     START4          ;AWAIT TUR
1883  003474  004767  017500            JSR     PC,PAPRT        ;PRINT HEADER
1884  003500  012704  026113            MOV     #MSG49,R4
1885  003504  004767  020412            JSR     PC,TTOUT        ;PRINT NOT AVAIL
1886  003510  000000                    HALT                    ;STOP
1887  003512  000167  177710            JMP     START4          ;RETRY
1888  003516  004767  010454  STAR4A:  JSR     PC,DSUP         ;GO SET UP WRITE DATA
1889  003522  004767  001256            JSR     PC,RWND         ;REWIND
1890  003526  004767  001646            JSR     PC,WRITE        ;WRITE
1891  003532  016767  175040  175130    MOV     TSTAL,STAL      ;SET TURN AROUND DELAY
1892  003540  004767  006474            JSR     PC,STALL                ;DELAY
1893  003544  004767  003572            JSR     PC,RSEQ         ;GO TO READ SEQUENCER
1894  003550  016767  175022  175112    MOV     TSTAL,STAL      ;SET TURN AROUND DELAY
1895  003556  004767  006456            JSR     PC,STALL        ;DELAY
1896  003562  032777  040000  175016    BIT     #40000,@SWR     ;SEE IF SHOULD PRINT STATISTICS
1897  003570  001565                    BEQ     START5          ;IF NOT: BR
1898  003572  012700  000001            MOV     #1,R0           ;SET RECORD COUNTER TO 1
1899  003576  004767  017376            JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
1900  003602  004767  000004            JSR     PC,STP          ;GO PRINT STATS
1901  003606  000167  000316            JMP     STPX
```

```
 1902  003612  004767  013530          STP:    JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
 1903  003616  012704  026325                  MOV     #MSG65,R4
 1904  003622  004767  020274                  JSR     PC,TTOUT        ;PRINT RETRY TOTAL
 1905  003626  016704  175044                  MOV     UNP,R4
 1906  003632  016403  001050                  MOV     RTY1(R4),R3
 1907  003636  004767  020432                  JSR     PC,OCTP         ;PRINT RETRIES
 1908  003642  012704  026476                  MOV     #MSG73,R4
 1909  003646  004767  020250                  JSR     PC,TTOUT        ;PRINT WRITE ERROR TAG
 1910  003652  016704  175020                  MOV     UNP,R4
 1911  003656  016403  001070                  MOV     WTER1(R4),R3
 1912  003662  004767  020406                  JSR     PC,OCTP         ;PRINT WRITE ERRORS
 1913  003666  012704  026465                  MOV     #MSG72,R4
 1914  003672  004767  020224                  JSR     PC,TTOUT        ;PRINT READ FORWARD ERROR TAG
 1915  003676  016704  174774                  MOV     UNP,R4
 1916  003702  016403  001110                  MOV     RDER1(R4),R3
 1917  003706  004767  020362                  JSR     PC,OCTP         ;PRINT READ FORWARD ERRORS
 1918  003712  012704  027305                  MOV     #MSG113,R4
 1919  003716  004767  020200                  JSR     PC,TTOUT        ;PRINT SOFT TAG
 1920  003722  016704  174750                  MOV     UNP,R4
 1921  003726  016403  002670                  MOV     RFSOFT(R4),R3
 1922  003732  004767  020336                  JSR     PC,OCTP         ;PRINT FORWARD SOFT ERRORS
 1923  003736  012704  027316                  MOV     #MSG114,R4
 1924  003742  004767  020154                  JSR     PC,TTOUT        ;PRINT HARD TAG
 1925  003746  016704  174724                  MOV     UNP,R4
 1926  003752  016403  002730                  MOV     RFHARD(R4),R3
 1927  003756  004767  020312                  JSR     PC,OCTP         ;PRINT HARD FORWARE ERRORS
 1928  003762  012704  026556                  MOV     #MSG77,R4
 1929  003766  004767  020130                  JSR     PC,TTOUT        ;PRINT DATA ERROR FORWARD TAG
 1930  003772  016704  174700                  MOV     UNP,R4
 1931  003776  016403  001130                  MOV     DATER1(R4),R3
 1932  004002  004767  020266                  JSR     PC,OCTP         ;PRINT DATA ERROR FORWARD NUMBER
 1933  004006  012704  026361                  MOV     #MSG68,R4
 1934  004012  004767  020104                  JSR     PC,TTOUT        ;PRINT READ ERROR REVERSE TAG
 1935  004016  016704  174654                  MOV     UNP,R4
 1936  004022  016403  001150                  MOV     RDERR1(R4),R3
 1937  004026  004767  020242                  JSR     PC,OCTP         ;PRINT REVESE ERROR NUMBER
 1938  004032  012704  027305                  MOV     #MSG113,R4
 1939  004036  004767  020060                  JSR     PC,TTOUT        ;PRINT SOFT TAG
 1940  004042  016704  174630                  MOV     UNP,R4
 1941  004046  016403  002710                  MOV     RRSOFT(R4),R3
 1942  004052  004767  020216                  JSR     PC,OCTP         ;PRINT REVERSE SOFT ERROR
 1943  004056  012704  027316                  MOV     #MSG114,R4
 1944  004062  004767  020034                  JSR     PC,TTOUT        ;PRINT HARD TAG
 1945  004066  016704  174604                  MOV     UNP,R4
 1946  004072  016403  002750                  MOV     RRHARD(R4),R3
 1947  004076  004767  020172                  JSR     PC,OCTP
 1948  004102  012704  026545                  MOV     #MSG76,R4
 1949  004106  004767  020010                  JSR     PC,TTOUT        ;PRINT DATA ERROR REVERSE TAG
 1950  004112  016704  174560                  MOV     UNP,R4
 1951  004116  016403  001170                  MOV     DEREV1(R4),R3
 1952  004122  004767  020146                  JSR     PC,OCTP         ;PRINT DATA REVERSE ERROR NUMBER
 1953  004126  000207                          RTS     PC              ;RETURN
 1954  004130  005267  174574          STPX:   INC     BTSTF           ;SET STAT ONLY PRINT
 1955  004134  004767  003102                  JSR     PC,BTPRT        ;PRINT BAD TAPE STATS
 1956  004140  005067  174564                  CLR     BTSTF           ;CLEAR FLAG
 1957  004144  017700  174436          START5: MOV     @SWR,R0         ;LOAD SWR
```

```
 1958  004150  042700  177762              BIC     #177762,R0      ;MASK READ/WRITE SWITCHES
 1959  004154  022700  000015              CMP     #15,R0          ;SEE IF HAVE READ OR WRITE
 1960  004160  001421                      BEQ     START8          ;IF NOT: BR
 1961  004162  032777  000200  174332 START6: BIT     #200,@DS        ;SEE IF HAVE UNIT READY
 1962  004170  001012                      BNE     START7          ;IF SO: BR
 1963  004172  005367  174472              DEC     STAL
 1964  004176  001371                      BNE     START6          ;DELAY FOR TUR
 1965  004200  004767  016774              JSR     PC,PAPRT        ;PRINT HEADER
 1966  004204  012704  026113              MOV     #MSG49,R4
 1967  004210  004767  017706              JSR     PC,TTOUT        ;PRINT NOT AVAIL
 1968  004214  000000                      HALT                    ;STOP
 1969  004216  062767  000002  174452 START7: ADD     #2,UNP          ;POINT TO NEXT UNIT
 1970  004224  000167  177014        START8: JMP     START1          ;CONTINUE
 1971
 1972                                      ;RANDOM BASE RESET**********
 1973
 1974  004230  012767  153624  174366 RANSET: MOV     #153624,RANBAS  ;RESET BASE
 1975  004236  012767  032561  174362        MOV     #32561,RANSAV   ;RESET BUFFER
 1976  004244  016767  174360  174302        MOV     RCSAV,RCNT      ;RESET RECORD COUNT
 1977  004252  016767  174354  174276        MOV     FCSAV,FMCNT     ;RESET FRAME COUNT
 1978  004260  000207                      RTS     PC
 1979
```

```
 1980                                   ;*****************************************************
 1981                                   ;REWIND FROM EOT:
 1982                                   ;
 1983                                   ;WHEN ANY TRANSPORT BEING TESTED REACHES END OF TAPE
 1984                                   ;DURING A READ OR WRITE OPERATION, IT WILL BE REWOUND
 1985                                   ;AND FLAGGED AS UNAVAILABLE UNTIL ALL AVAILABLE UNITS
 1986                                   ;HAVE REACHED EOT AT WHICH TIME ALL TESTING WILL BE RESUMED
 1987                                   ;AT A BLOCK COUNT OF ONE (1).  A MESSAGE WILL BE
 1988                                   ;PRINTED ON THE SUPERVISORS CONSOLE AS EACH UNIT REACHES
 1989                                   ;EOT AND IS REWOUND.
 1990                                   ;*****************************************************
 1991
 1992  004262  016777  174264  174252  REOT:   MOV     UDES,@C2        ;LOAD COMMAND REGISTER
 1993  004270  012777  000011  174212          MOV     #11,@C1         ;DRIVE CLEAR
 1994  004276  032777  000200  174216  REOT1:  BIT     #200,@DS
 1995  004304  001774                          BEQ     REOT1           ;AWAIT DRY
 1996  004306  012777  000007  174174          MOV     #7,@C1          ;START REWIND
 1997  004314  005767  174406                  TST     BTFLG           ;SEE IF BAD TAPE OVERFLOW REWIND
 1998  004320  001004                          BNE     REOT1A          ;IF SO: BR
 1999  004322  016700  174334                  MOV     EOTREC,R0
 2000  004326  042700  100000                  BIC     #100000,R0      ;SET RECORD NUMBER OF EOT
 2001  004332  004767  016642          REOT1A: JSR     PC,PAPRT        ;PRINT HEADER
 2002  004336  022767  000002  174362          CMP     #2,BTFLG        ;SEE IF POSITION ERROR
 2003  004344  001003                          BNE     REOT1B          ;IF NOT: BR
 2004  004346  012704  027176                  MOV     #MSG109,R4      ;SET POSITION ERROR MSG
 2005  004352  000406                          BR      REOT1F
 2006  004354  022767  000001  174344  REOT1B: CMP     #1,BTFLG        ;SEE IF BAD TAPE OVERFLOW
 2007  004362  001005                          BNE     REOT1C          ;IF NOT: BR
 2008  004364  012704  027007                  MOV     #MSG106,R4      ;SET BAD TAPE OVERFLOW MSG
 2009  004370  004767  017526          REOT1F: JSR     PC,TTOUT        ;PRINT REWIND REASON
 2010  004374  000414                          BR      REOT1E
 2011  004376  012704  025116          REOT1C: MOV     #MSG20,R4       ;SET EOT MSG
 2012  004402  004767  017514          REOT1D: JSR     PC,TTOUT        ;PRINT MSG
 2013  004406  016704  174264                  MOV     UNP,R4
 2014  004412  005264  002650                  INC     EOTC0(R4)       ;BUMP CNTR
 2015  004416  016403  002650                  MOV     EOTC0(R4),R3
 2016  004422  004767  017646                  JSR     PC,OCTP         ;PRINT EOT CNTR
 2017  004426  012704  027034          REOT1E: MOV     #MSG16A,R4
 2018  004432  004767  017464                  JSR     PC,TTOUT        ;PRINT RESTART MSG
 2019  004436  005067  174264                  CLR     BTFLG           ;CLEAR BAD TAPE FLAG
 2020  004442  004767  177144                  JSR     PC,STP          ;PRINT STATS
 2021  004446  004767  002570                  JSR     PC,BTPRT        ;PRINT BAD TAPE STATS
 2022  004452  032777  000200  174042  REOT2:  BIT     #200,@DS
 2023  004460  001015                          BNE     REOT2A          ;IF DRY: BR
 2024  004462  005367  174202                  DEC     STAL
 2025  004466  001371                          BNE     REOT2           ;WAIT DRY
 2026  004470  012767  024755  174156          MOV     #MSG6,EMADDR
 2027  004476  004767  016476                  JSR     PC,PAPRT        ;PRINT HEADER
 2028  004502  012704  026267                  MOV     #MSG60,R4
 2029  004506  004767  017410                  JSR     PC,TTOUT        ;PRINT NO DRIVE READY
 2030  004512  000000                          HALT
 2031  004514  105367  000262          REOT2A: DECB    REOTC           ;SEE IF LAST UNIT TO REACH EOT
 2032  004520  001410                          BEQ     REOT3           ;IF SO: BR
 2033  004522  016700  174150                  MOV     UNP,R0
 2034  004526  052760  100000  000746          BIS     #100000,UN1(R0) ;SET EOT FLAG
 2035  004534  005726                          TST     (SP)+           ;RESET STACK POINTER
```

```
004536  000167  177454                    JMP     START7            ;GO TO NEXT UNIT
004542  000367  000234          REOT3:    SWAB    REOTC
004546  016700  000230                    MOV     REOTC,R0
004552  000367  000224                    SWAB    REOTC
004556  110067  000220                    MOVB    R0,REOTC          ;RESTORE EOT UNIT COUNTER
004562  005067  174110                    CLR     UNP
004566  016700  174104                    MOV     UNP,R0            ;POINT TO FIRST UNIT
004572  016067  000746  173752  REOT4:    MOV     UN1(R0),UDES      ;LOAD UNIT DESCRIPTION
004600  016777  173746  173734            MOV     UDES,@C2          ;LOAD COMMAND REGISTER
004606  032777  020000  173706  REOT5:    BIT     #20000,@DS
004614  001374                            BNE     REOT5             ;AWAIT PIP RESET
004616  032777  000002  173676            BIT     #2,@DS            ;SEE IF HAVE BOT
004624  001013                            BNE     REOT6             ;IF SO: BR
004626  012700  000001                    MOV     #1,R0
004632  004767  016342                    JSR     PC,PAPRT          ;PRINT HEADER
004636  012704  026060                    MOV     #MSG48,R4
004642  004767  017254                    JSR     PC,TTOUT          ;PRINT BOT ERROR
004646  000000                            HALT
004650  016700  174022                    MOV     UNP,R0
004654  042760  100000  000746  REOT6:    BIC     #100000,UN1(R0) ;CLEAR EOT FLAG
004662  062767  000002  174006            ADD     #2,UNP
004670  016700  174002                    MOV     UNP,R0            ;POINT TO NEXT UNIT
004674  005160  000746                    COM     UN1(R0)           ;SEE IF LAST UNIT
004700  001404                            BEQ     REOT7             ;IF SO: BR
004702  005160  000746                    COM     UN1(R0)
004706  000167  177660                    JMP     REOT4             ;DO NEXT UNIT
004712  005160  000746          REOT7:    COM     UN1(R0)
004716  005067  173754                    CLR     UNP               ;CLEAR UNIT POINTER
004722  005067  173710                    CLR     TINF              ;CLEAR TTY INPUT FLAG
004726  005767  174004                    TST     ASEQF             ;SEE IF AUTO SEQ
004732  001402                            BEQ     REOTX             ;IF NOT: BR
004734  005726                            TST     (SP)+             ;RESET STACK POINTER
004736  000207                            RTS     PC                ;RETURN TO AUTO SEQ
004740  004767  177264          REOTX:    JSR     PC,RANSET         ;GO RESET RANDOM BASE
004744  012767  177777  007436            MOV     #-1,PATS          ;PRESET PATTERN
004752  005067  010232                    CLR     RDFL              ;CLEAR RANDOM FLAG
004756  005767  173606                    TST     SPFLG             ;SEE IF SINGLE PASS
004762  001405                            BEQ     REOTXX            ;IF NOT: BR
004764  012704  026670                    MOV     #MSG100,R4
004770  004767  017126                    JSR     PC,TTOUT          ;PRINT END OF PASS
004774  000000                            HALT
004776  000167  176134          REOTXX:   JMP     STARTE            ;RESTART AT BLOCK NUMBER ONE
005002  000000                  REOTC:    0                         ;EOT UNIT COUNTER
```

```
2079                                    ;*********************************************************
2080                                    ;REWIND ALL AVAIL TAPES:
2081                                    ;
2082                                    ;THIS ROUTINE; ENTERED VIA CONSOLE SWITCH NINE (9),
2083                                    ;WILL REWIND ALL AVAILABLE TAPES TO BOT NO MATTER
2084                                    ;WHERE THEY ARE CURRENTLY POSITIONED AND RESUME TESTING
2085                                    ;ON THE CURRENTLY SELECTED UNIT.
2086                                    ;*********************************************************
2087
2088  005004  032777  001000  173574  RWND:   BIT     #1000,@SWR      ;SEE IF SHOULD REWIND
2089  005012  001001                          BNE     RWNDA           ;IF SO: BR
2090  005014  000207                          RTS     PC              ;ELSE EXIT
2091  005016  016767  173654  173672  RWNDA:  MOV     UNP,UPS         ;SAVE UNIT POINTER
2092  005024  005067  173646                  CLR     UNP             ;CLEAR POINTER
2093  005030  005067  173626                  CLR     EOTREC          ;CLEAR EOT FLAG
2094  005034  000367  177742                  SWAB    REOTC
2095  005040  016700  177736                  MOV     REOTC,R0
2096  005044  000367  177732                  SWAB    REOTC
2097  005050  110067  177726                  MOVB    R0,REOTC        ;RESTORE EOT UNIT COUNTER
2098  005054  016700  173616          RWND0:  MOV     UNP,R0          ;POINT TO UNIT ENTRY
2099  005060  005160  000746                  COM     UN1(R0)         ;SEE IF LAST ENTRY
2100  005064  001453                          BEQ     RWND2           ;IF SO: BR
2101  005066  005160  000746                  COM     UN1(R0)
2102  005072  032760  100000  000746          BIT     #100000,UN1(R0) ;SEE IF ALREADY REWINDING
2103  005100  001035                          BNE     RWND1A          ;IF SO: BR
2104  005102  016067  000746  173442          MOV     UN1(R0),UDES    ;SET UNIT DESCRIPTION
2105  005110  016777  173436  173424          MOV     UDES,@C2        ;LOAD COMMAND REGISTER
2106  005116  012777  000011  173364          MOV     #11,@C1         ;DRIVE CLEAR
2107  005124  012777  000007  173356          MOV     #7,@C1          ;START REWIND
2108  005132  032777  000200  173362  RWND1:  BIT     #200,@DS
2109  005140  001015                          BNE     RWND1A          ;IF DRY: BR
2110  005142  005367  173522                  DEC     STAL
2111  005146  001371                          BNE     RWND1           ;AWAIT DRY
2112  005150  012767  024755  173476          MOV     #MSG6,EMADDR
2113  005156  004767  016016                  JSR     PC,PAPRT        ;PRINT HEADER
2114  005162  012704  026410                  MOV     #MSG70,R4
2115  005166  004767  016730                  JSR     PC,TTOUT        ;PRINT NO DRIVE READY
2116  005172  000000                          HALT
2117  005174  042760  100000  000746  RWND1A: BIC     #100000,UN1(R0) ;CLEAR EOT FLAG
2118  005202  062767  000002  173466          ADD     #2,UNP          ;BUMP POINTER
2119  005210  000167  177640                  JMP     RWND0           ;DO NEXT UNIT
2120  005214  005160  000746          RWND2:  COM     UN1(R0)
2121  005220  005067  173452                  CLR     UNP             ;CLEAR POINTER
2122  005224  016700  173446          RWND3:  MOV     UNP,R0          ;POINT TO UNIT ENTRY
2123  005230  005160  000746                  COM     UN1(R0)         ;SEE IF LAST ENTRY:
2124  005234  001443                          BEQ     RWNDX           ;IF SO: BR
2125  005236  005160  000746                  COM     UN1(R0)
2126  005242  016067  000746  173302          MOV     UN1(R0),UDES    ;SET UNIT DESCRIPTION
2127  005250  016777  173276  173264          MOV     UDES,@C2        ;LOAD COMMAND REGISTER
2128  005256  032777  020000  173236  RWND4:  BIT     #20000,@DS
2129  005264  001374                          BNE     RWND4           ;AWAIT PIP RESET
2130  005266  032777  000002  173226          BIT     #2,@DS          ;SEE IF HAVE BOT
2131  005274  001410                          BEQ     RWND6           ;IF NOT: BR
2132  005276  062767  000002  173372  RWND5:  ADD     #2,UNP          ;BUMP POINTER
2133  005304  012777  000011  173176          MOV     #11,@C1         ;DRIVE CLEAR
2134  005312  000167  177706                  JMP     RWND3           ;DO NEXT UNIT
```

```
 2135  005316  012700  000001          RWND6:  MOV     #1,R0
 2136  005322  004767  015652                  JSR     PC,PAPRT        ;PRINT HEADER
 2137  005326  012704  026060                  MOV     #MSG48,R4
 2138  005332  004767  016564                  JSR     PC,TTOUT        ;PRINT NO BOT
 2139  005336  000000                          HALT
 2140  005340  000167  177732                  JMP     RWND5           ;DO NEXT UNIT
 2141  005344  005160  000746          RWNDX:  COM     UN1(R0)
 2142  005350  016767  173342  173320          MOV     UPS,UNP         ;RESTORE UNIT POINTER
 2143  005356  016700  173314                  MOV     UNP,R0
 2144  005362  016067  000746  173162          MOV     UN1(R0),UDES    ;RESET UNIT DESCRIPTION
 2145  005370  016777  173156  173144          MOV     UDES,@C2
 2146  005376  000207                          RTS     PC              ;RETURN TO TEST
 2147
```

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

005400  005067  173256        WRITE:  CLR     EOTREC          ;CLEAR EOT FLAG
005404  032777  000001  173174        BIT     #1,@SWR         ;SEE IF SHOULD WRITE
005412  001402                       BEQ     WRTE
005414  000167  000562               JMP     WEX             ;IF NOT: BR
005420  016700  173130        WRTE:   MOV     RCNT,R0         ;R0=RECORD COUNT
005424  012767  024750  173222  W0:   MOV     #MSG5,EMADDR    ;SET ERROR MSG ADDRESS
005432  016777  173120  173056        MOV     FMCNT,@FC       ;LOAD CHAR COUNT
005440  012777  027426  173046        MOV     #WDATA,@BA      ;SET DATA ADDR
005446  112767  000060  173220        MOVB    #60,MTC1        ;SET WRITE OP COMMAND
005454  012767  005466  173202        MOV     #W1,RTRN        ;SET RETURN ADDRESS
005462  000167  013734               JMP     TAPG            ;GO EXECUTE COMMAND
005466  032777  002000  173026  W1:   BIT     #2000,@DS       ;SEE IF EOT
005474  001405                       BEQ     W2              ;IF NOT AT EOT: BR
005476  010067  173160               MOV     R0,EOTREC       ;SAVE EOT RECORD NUMBER
005502  052767  100000  173152        BIS     #100000,EOTREC  ;SET EOT FLAG
005510  032777  010000  173070  W2:   BIT     #10000,@SWR     ;SEE IF SHOULD CHECK ERRORS
005516  001002                       BNE     W3              ;IF NOT: BR
005520  004767  011770               JSR     PC,ERCHK        ;GO CHECK ERRORS
005524  016767  173044  173136  W3:   MOV     WSTAL,STAL      ;SET DELAY
005532  004767  004502               JSR     PC,STALL        ;DELAY
005536  005767  173152               TST     RTYFL           ;SEE IF RETRY TIME
005542  001401                       BEQ     W3A             ;IF NOT: BR
005544  000207                       RTS     PC              ;ELSE RETURN
```

```
 2204  005546  005767  173136          W3A:    TST     SERFL           ;SEE IF WRITE ERROR
 2205  005552  001452                          BEQ     W5              ;IF NOT: BR
 2206  005554  016704  173116                  MOV     UNP,R4
 2207  005560  005264  001070                  INC     WTER1(R4)       ;BUMP WRITE ERROR
 2208  005564  005067  173120                  CLR     SERFL           ;CLEAR STATUS ERROR FLAG
 2209  005570  032777  000020  173010          BIT     #20,@SWR        ;SEE IF RETRY
 2210  005576  001440                          BEQ     W5              ;IF NOT: BR
 2211  005600  016703  173120                  MOV     ERSAV,R3
 2212  005604  042703  102700                  BIC     #102700,R3      ;MASK UNRECOVERABLE ERROR
 2213  005610  001411                          BEQ     W4              ;IF SO: BR
 2214  005612  004767  015362                  JSR     PC,PAPRT        ;PRINT HEADER
 2215  005616  012704  026567                  MOV     #MSG78,R4
 2216  005622  004767  016274                  JSR     PC,TTOUT        ;PRINT NON-RETRYABLE ERROR TAG
 2217  005626  004767  003404                  JSR     PC,NRTP         ;PRINT ER FOR NON-RETRYABLE
 2218  005632  000422                          BR      W5
 2219  005634  016704  173036          W4:     MOV     UNP,R4
 2220  005640  005264  001050                  INC     RTY1(R4)        ;BUMP RETRY CNTR
 2221  005644  032777  002000  172734          BIT     #2000,@SWR      ;SEE IF PRINT ERRORS
 2222  005652  001004                          BNE     W4A             ;IF NOT: BR
 2223  005654  012704  026303                  MOV     #MSG64,R4
 2224  005660  004767  016236                  JSR     PC,TTOUT        ;PRINT ORIGINAL ERROR TAG
 2225  005664  005067  173014          W4A:    CLR     RTCNT           ;CLEAR RETRY NUMBER
 2226  005670  005067  173006                  CLR     RPCNT           ;CLEAR REPEAT COUNTER
 2227  005674  004767  000346                  JSR     PC,WRTY         ;GO RETRY WRITE ERROR
 2228  005700  005067  173010          W5:     CLR     RTYFL           ;CLEAR RETRY COUNTER
 2229  005704  005767  172752                  TST     EOTREC          ;SEE IF EOT FOUND
 2230  005710  100402                          BMI     W6              ;IF SO: BR
 2231  005712  005300                          DEC     R0              ;SEE IF DONE ALL
 2232  005714  001243                          BNE     W0              ;IF NOT: BR
 2233  005716  005767  172642          W6:     TST     TMEX            ;SEE IF TM
 2234  005722  001527                          BEQ     WEX             ;IF NOT: BR
 2235  005724  005267  172750                  INC     TMFLG           ;SET TM FLAG
 2236  005730  012767  026210  172716  WTM:    MOV     #MSG54,EMADDR   ;POINT TO TM ERROR MSG
 2237  005736  012767  000026  172730          MOV     #26,MTC1        ;SET TM OP CODE
 2238  005744  012777  000000  172544          MOV     #0,@FC          ;LOAD FRAME COUNTER
 2239  005752  012777  027426  172534          MOV     #WDATA,@BA      ;LOAD BUS ADDRESS
 2240  005760  012767  005772  172676          MOV     #WTM0,RTRN      ;SAVE RETURN ADDRESS
 2241  005766  000167  013430                  JMP     TAPG            ;WRITE TM
 2242  005772  032777  010000  172606  WTM0:   BIT     #10000,@SWR     ;SEE IF SHOULD CHECK ERRORS
 2243  006000  001100                          BNE     WEX
 2244  006002  032777  000004  172512  ..      BIT     #4,@DS          ;SEE IF TM STATUS
 2245  006010  001011                          BNE     WTM1            ;IF SO: BR
 2246  006012  012767  027426  013312          MOV     #WDATA,CADER    ;SET EXPT BUS ADDRESS
 2247  006020  012767  000001  013312          MOV     #1,DRVER        ;INDICATE ERROR
 2248  006026  004767  012320                  JSR     PC,ERPT         ;PRINT TM ERROR
 2249  006032  000404                          BR      WTM2
 2250  006034  012703  027426          WTM1:   MOV     #WDATA,R3       ;SET EXPT ADDRESS
 2251  006040  004767  011546                  JSR     PC,ER2          ;GO CHECK FOR OTHER ERRORS
 2252  006044  005767  172644          WTM2:   TST     RTYFL           ;SEE IF RETRY
 2253  006050  001401                          BEQ     WTM3            ;IF NOT: BR
 2254  006052  000207                          RTS     PC              ;ELSE RETURN TO RETRY ROUTINE
 2255  006054  005767  172630          WTM3:   TST     SERFL           ;SEE IF WRITE ERROR
 2256  006060  001450                          BEQ     WEX             ;IF NOT: BR
 2257  006062  016704  172610                  MOV     UNP,R4
 2258  006066  005264  001070                  INC     WTER1(R4)       ;BUMP WRITE ERROR
 2259  006072  032777  000020  172506          BIT     #20,@SWR        ;SEE IF SHOULD RETRY
```

```
2260  006100  001440                    BEQ     WEX             ;IF NOT: BR
2261  006102  016703  172616            MOV     ERSAV,R3
2262  006106  042703  102700            BIC     #102700,R3      ;MASK UNRECOVERABLE ERROR
2263  006112  001411                    BEQ     WTM4            ;IF SO: BR
2264  006114  004767  015060            JSR     PC,PAPRT        ;PRINT HEADER
2265  006120  012704  026567            MOV     #MSG78,R4
2266  006124  004767  015772            JSR     PC,TTOUT        ;PRINT UNRETRYABLE TAG
2267  006130  004767  003102            JSR     PC,NRTP         ;PRINT ER FOR NON-RETRYABLE
2268  006134  000422                    BR      WEX
2269  006136  005067  172540    WTM4:   CLR     RPCNT           ;CLEAR REPEAT CNTR
2270  006142  016704  172530            MOV     UNP,R4
2271  006146  005264  001050            INC     RTY1(R4)        ;BUMP RETRY CNTR
2272  006152  005067  172526            CLR     RTCNT           ;CLEAR RETRY CNTR
2273  006156  032777  002000  172422    BIT     #2000,@SWR      ;SEE IF PRINT ERRORS
2274  006164  001004                    BNE     WTM4A           ;IF NOT: BR
2275  006166  012704  026303            MOV     #MSG64,R4
2276  006172  004767  015724            JSR     PC,TTOUT        ;PRINT ORIGINAL ERROR TAG
2277  006176  004767  000044    WTM4A:  JSR     PC,WRTY         ;GO DO RETRY
2278  006202  005067  172506    WEX:    CLR     RTYFL           ;CLEAR RETRY FLAG
2279  006206  005067  172466            CLR     TMFLG           ;CLEAR TAPE MARK FLAG
2280  006212  005767  172444            TST     EOTREC          ;SEE IF EOT
2281  006216  100401                    BMI     WRW             ;IF SO: BR
2282  006220  000207                    RTS     PC              ;ELSE EXIT
2283  006222  017703  172360    WRW:    MOV     @SWR,R3
2284  006226  042703  177763            BIC     #177763,R3
2285  006232  022703  000014            CMP     #14,R3          ;SEE IF WRITE ONLY
2286  006236  001002                    BNE     WRWX            ;IF NOT: BR
2287  006240  000167  176016            JMP     REOT            ;ELSE REWIND
2288  006244  000207            WRWX:   RTS     PC              ;EXIT
```

```
2289                                    ;*************************
2290                                    ;WRITE ERROR RETRY
2291                                    ;
2292                                    ;*************************
2293
2294  006246  012767  000001  172440  WRTY:   MOV     #1,RTYFL        ;SET RETRY FLAG
2295  006254  004767  000406          WRTY0:  JSR     PC,WRTSB        ;GO SPACE REVERSE FOR REPEAT
2296  006260  005767  172414                  TST     TMFLG           ;SEE IF TAPE MARK TIME
2297  006264  001003                          BNE     WRTYTM          ;IF SO: BR
2298  006266  004767  177132                  JSR     PC,WO           ;REWRITE RECORD
2299  006272  000402                          BR      WRTYR           ;GO ON
2300  006274  004767  177430          WRTYTM: JSR     PC,WTM          ;GO WRITE TAPE MARK AGAIN
2301  006300  005767  172404          WRTYR:  TST     SERFL           ;REWRITE GOOD
2302  006304  001027                          BNE     WRTY2           ;IF NOT: BR
2303  006306  005267  172370                  INC     RPCNT           ;BUMP REPEAT COUNTER
2304  006312  022767  000003  172362          CMP     #3,RPCNT        ;SEE IF THREE GOOD REPEATS
2305  006320  001355                          BNE     WRTY0           ;IF NOT, REPEAT
2306  006322  032777  002000  172256          BIT     #2000,@SWR      ;SEE IF PRINT
2307  006330  001014                          BNE     WRTY1           ;IF NOT: BR
2308  006332  012704  026774                  MOV     #MSG105,R4
2309  006336  004767  015560                  JSR     PC,TTOUT        ;PRINT RECOVERED MESSAGE
2310  006342  012704  026325                  MOV     #MSG65,R4
2311  006346  004767  015550                  JSR     PC,TTOUT        ;PRINT RETRY TAG
2312  006352  016703  172326                  MOV     RTCNT,R3
2313  006356  004767  015712                  JSR     PC,OCTP         ;PRINT RETRY NUMBER
2314  006362  000207          WRTY1:  RTS     PC                      ;RESUME TESTING
2315  006364  016703  172334          WRTY2:  MOV     ERSAV,R3        ;GET ER
2316  006370  042703  102700                  BIC     #102700,R3      ;MASK RECOVERABLE BITS
2317  006374  001414                          BEQ     WRTY2A          ;IF RECOVERABLE: BR
2318  006376  004767  014576                  JSR     PC,PAPRT        ;PRINT HEADER
2319  006402  012704  026567                  MOV     #MSG78,R4
2320  006406  004767  015510                  JSR     PC,TTOUT        ;PRINT NON-RECOVERABLE MSG
2321  006412  004767  002620                  JSR     PC,NRTP         ;PRINT ER
2322  006416  012767  000001  172224          MOV     #1,TEMP3        ;SET FLAG
2323  006424  000410                          BR      WRTY2B
2324  006426  032777  002000  172152  WRTY2A: BIT     #2000,@SWR      ;SEE IF PRINT
2325  006434  001032                          BNE     WRTY3           ;IF NOT: BR
2326  006436  012704  027230                  MOV     #MSG110,R4
2327  006442  004767  015454                  JSR     PC,TTOUT        ;PRINT BAD TAPE SUSPECT
2328  006446  012704  026325          WRTY2B: MOV     #MSG65,R4
2329  006452  004767  015444                  JSR     PC,TTOUT        ;PRINT RETRY TAG
2330  006456  016703  172222                  MOV     RTCNT,R3
2331  006462  004767  015606                  JSR     PC,OCTP         ;PRINT RETRY NUMBER
2332  006466  012704  027252                  MOV     #MSG111,R4
2333  006472  004767  015424                  JSR     PC,TTOUT        ;PRINT REPEAT TAG
2334  006476  016703  172200                  MOV     RPCNT,R3
2335  006502  004767  015566                  JSR     PC,OCTP         ;PRINT REPEAT NUMBER
2336  006506  005767  172136                  TST     TEMP3           ;SEE IF DID NON-RECOVERABLE
2337  006512  001403                          BEQ     WRTY3           ;IF NOT: BR
2338  006514  005067  172130                  CLR     TEMP3           ;CLEAR FLAG
2339  006520  000207                          RTS     PC              ;EXIT
2340  006522  005767  172156          WRTY3:  TST     RTCNT           ;SEE IF FIRST RETRY
2341  006526  001004                          BNE     WRTY3A          ;IF NOT: BR
2342  006530  016704  172142                  MOV     UNP,R4
2343  006534  005364  001070                  DEC     WTER1(R4)       ;DECREMENT WRITE ERROR CNTR
2344  006540  016704  172132          WRTY3A: MOV     UNP,R4          ;GET UNIT NUMBER
```

```
2345  006544  016467  001030  172160          MOV     BTADDR(R4),BTPT ;GET ADDRESS OF UNIT BAD TAPE CNTR
2346  006552  017704  172154                  MOV     @BTPT,R4        ;GET COUNTER
2347  006556  005724                          TST     (R4)+           ;SET POINTER OFFSET
2348  006560  010477  172146                  MOV     R4,@BTPT
2349  006564  016703  172142                  MOV     BTPT,R3
2350  006570  060304                          ADD     R3,R4           ;SET ABSOLUTE POINTER
2351  006572  016714  172060                  MOV     BLCNTR,(R4)     ;SET BLOCK NUMBER
2352  006576  062704  000040                  ADD     #40,R4          ;ADD RCNT OFFSET
2353  006602  016714  171746                  MOV     RCNT,(R4)
2354  006606  160014                          SUB     R0,(R4)         ;SET RECORD NUMBER
2355  006610  005214                          INC     (R4)            ;CORRECT RECORD NUMBER
2356  006612  022777  000040  172112          CMP     #40,@BTPT       ;SEE IF TOO MANY BAD SPOTS
2357  006620  001002                          BNE     WRTY4           ;IF NOT: BR
2358  006622  000167  000234                  JMP     BTOV            ;ELSE GO TO BAD TAPE OVERFLOW
2359  006626  005267  172052          WRTY4:  INC     RTCNT           ;BUMP RETRY COUNTER
2360  006632  022767  000004  172044          CMP     #4,RTCNT        ;SEE IF DONE 4 RETRIES
2361  006640  001410                          BEQ     WRTY5           ;IF SO: BR
2362  006642  016704  172030                  MOV     UNP,R4
2363  006646  005264  001050                  INC     RTY1(R4)        ;BUMP RETRY COUNTER
2364  006652  005267  172056                  INC     ERTFL           ;SET ERASE FLAG
2365  006656  000167  177372                  JMP     WRTY0           ;DO NEXT RETRY
2366  006662  000167  000436          WRTY5:  JMP     BTUR            ;ELSE GO TO BAD TAPE UNRECOVERABLE
2367
2368                                          ;WRITE RETRY BACKSPACE-ERASE SUBROUTINE**********
2369
2370  006666  005067  172016          WRTSB:  CLR     SERFL           ;CLEAR FLAG
2371  006672  016767  171700  171770          MOV     TSTAL,STAL
2372  006700  004767  003334                  JSR     PC,STALL        ;DO TURN AROUND DELAY
2373  006704  012767  026336  171742          MOV     #MSG66,EMADDR   ;SET ERROR CODE
2374  006712  012777  177777  171576          MOV     #-1,@FC         ;SET TO BACKSPACE 1 RECORD
2375  006720  012777  033434  171566          MOV     #RDATA,@BA      ;SET BA
2376  006726  004767  003236                  JSR     PC,BKRT         ;GO BACKSPACE
2377  006732  005767  171752                  TST     SERFL           ;SEE IF ERROR
2378  006736  001406                          BEQ     WRTSB1          ;IF NOT: BR
2379  006740  012767  000002  171760  WRTSB0: MOV     #2,BTFLG        ;SET FLAG
2380  006746  022626                          CMP     (SP)+,(SP)+     ;RESET STACK
2381  006750  000167  175306                  JMP     REOT            ;GO REWIND AND REMOVE FROM TESTING
2382  006754  005767  171754          WRTSB1: TST     ERTFL           ;SEE IF SHOULD ERASE
2383  006760  001001                          BNE     WRTSB2          ;IF SO: BR
2384  006762  000207                          RTS     PC              ;RETURN
2385  006764  005067  171744          WRTSB2: CLR     ERTFL           ;CLEAR ERASE FLAG
2386  006770  005067  171706                  CLR     RPCNT           ;CLEAR REPEAT CNTR
2387  006774  005067  171710                  CLR     SERFL           ;CLEAR FLAG
2388  007000  012767  026351  171646          MOV     #MSG67,EMADDR   ;SET ERROR CODE
2389  007006  005077  171504                  CLR     @FC             ;CLEAR FRAME COUNT
2390  007012  012767  000024  171654          MOV     #24,MTC1        ;SET ERASE OP-CODE
2391  007020  012777  027426  171466          MOV     #WDATA,@BA      ;SET BA
2392  007026  012767  007040  171630          MOV     #WRTSB3,RTRN    ;SET RETURN ADDRESS
2393  007034  000167  012362                  JMP     TAPG            ;GO ERASE
2394  007040  012703  027426          WRTSB3: MOV     #WDATA,R3       ;SET EXPT BA
2395  007044  004767  010542                  JSR     PC,ER2          ;GO CHECK ERRORS
2396  007050  005767  171634                  TST     SERFL           ;SEE IF ERROR
2397  007054  001737                          BEQ     WRTSB1          ;IF NOT: BR
2398  007056  000167  177656                  JMP     WRTSB0
2399
2400                                          ;BAD TAPE OVERFLOW SUBROUTINE**********
```

```
2401
2402  007062  005067  171626          BTOV:   CLR     RTYFL           ;CLEAR RETRY FLAG
2403  007066  012767  000001  171632          MOV     #1,BTFLG        ;SET BAD TAPE OVERFLOW FLAG
2404  007074  005726                          POPSP
2405  007076  000167  175160                  JMP     REOT            ;GO REWIND AND REMOVE FROM TESTING
2406  007102  016701  171624          BTOV0:  MOV     BTPT,R1         ;SET TABLE POINTER
2407  007106  005721                          TST     (R1)+
2408  007110  005000                          CLR     R0
2409  007112  010003                  BTOV1:  MOV     R0,R3
2410  007114  000241                          CLC
2411  007116  006003                          ROR     R3              ;R3=R3/2 FOR CORRECT NUMBER
2412  007120  004767  015150                  JSR     PC,OCTP         ;PRINT ENTRY NUMBER
2413  007124  012704  025044                  MOV     #MSG13,R4
2414  007130  105724                          TSTB    (R4)+           ;SKIP CR/LF
2415  007132  004767  014764                  JSR     PC,TTOUT        ;PRINT BLOCK NUMBER TAG
2416  007136  011103                          MOV     (R1),R3
2417  007140  004767  015130                  JSR     PC,OCTP         ;PRINT BLOCK NUMBER
2418  007144  012704  025052                  MOV     #MSG14,R4
2419  007150  004767  014746                  JSR     PC,TTOUT        ;PRINT RECORD NUMBER TAG
2420  007154  062701  000040                  ADD     #40,R1          ;SET POINTER OFFSET FOR RECOED NUMBER
2421  007160  012103                          MOV     (R1)+,R3
2422  007162  004767  015106                  JSR     PC,OCTP         ;PRINT RECORD NUMBER
2423  007166  162701  000040                  SUB     #40,R1          ;RESET POINTER FOR BLOCK NUMBER
2424  007172  005720                          TST     (R0)+
2425  007174  020077  171532                  CMP     R0,@BTPT        ;SEE IF DONE
2426  007200  001405                          BEQ     BTOV2           ;IF SO: BR
2427  007202  012704  025352                  MOV     #MSG28,R4
2428  007206  004767  014710                  JSR     PC,TTOUT        ;DO CR/LF
2429  007212  000737                          BR      BTOV1           ;CONTINUE
2430  007214  005767  171510          BTOV2:  TST     BTSTF           ;SEE IF STAT ONLY PRINT
2431  007220  001007                          BNE     BTOVX           ;IF SO: BR
2432  007222  012703  000041                  MOV     #41,R3          ;SET SIZE OF TABLE
2433  007226  016704  171500                  MOV     BTPT,R4         ;SET POINTER
2434  007232  005024                  BTOV3:  CLR     (R4)+           ;CLEAR TABLE
2435  007234  005303                          DEC     R3              ;SEE IF DONE
2436  007236  001375                          BNE     BTOV3           ;IF NOT: BR
2437  007240  000207                  BTOVX:  RTS     PC              ;RETURN
2438
```

```
 2439
 2440                                        ;BAD TAPE STATISTIC PRINT********
 2441
 2442  007242  012704  025352        BTPRT:  MOV     #MSG28,R4
 2443  007246  004767  014650                JSR     PC,TTOUT        ;DO CR/LF
 2444  007252  016704  171420                MOV     UNP,R4
 2445  007256  016467  001030  17144o        MOV     BTADDR(R4),BTPT ;SET TABLE POINTER
 2446  007264  017703  171442                MOV     @BTPT,R3
 2447  007270  000241                        CLC
 2448  007272  006003                        ROR     R3              ;CORRECT NUMBER
 2449  007274  004767  014774                JSR     PC,OCTP         ;PRINT NUMBER OF BAD SPOTS
 2450  007300  012704  027264                MOV     #MSG112,R4
 2451  007304  004767  014612                JSR     PC,TTOUT        ;PRINT BAD TAPE TAG
 2452  007310  005777  171416                TST     @BTPT           ;SEE IF ANY BAD SPOTS
 2453  007314  001001                        BNE     BTPRT1          ;IF SO: BR
 2454  007316  000207                        RTS     PC              ;ELSE RETURN
 2455  007320  000167  177556        BTPRT1: JMP     BTOV0           ;PRINT STATS
 2456
 2457                                        ;BAD TAPE UNRECOVERABLE SUBROUTINE**********
 2458
 2459  007324  004767  013650        BTUR:   JSR     PC,PAPRT        ;PRINT HEADER
 2460  007330  012704  027077                MOV     #MSG107,R4
 2461  007334  004767  014562                JSR     PC,TTOUT        ;PRINT UNRECOVERABLE BAD SPOT MSG
 2462  007340  000207                        RTS     PC              ;RESUME TESTING
 2463
```

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

007342  012767  000002  171212  RSEQ:  MOV     #2,RDCMD
007350  017704  171232                 MOV     @SWR,R4         ;READ SWITCHES
007354  042704  177763                 BIC     #177763,R4      ;MASK READ BITS
007360  005704                         TST     R4              ;SEE IF BOTH READS
007362  001004                         BNE     RSR             ;IF NOT: BR
007364  032777  000002  171214         BIT     #2,@SWR         ;SEE IF READ REVERSE FIRST
007372  001053                         BNE     RSFR            ;IF NOT: BR
007374  032777  000004  171204  RSR:   BIT     #4,@SWR         ;SEE IF SHOULD READ REVERSE
007402  001005                         BNE     RSF             ;IF NOT: BR
007404  012767  010000  171150         MOV     #10000,RDCMD    ;LOAD READ REVERSE COMMAND
007412  004767  000252                 JSR     PC,READ         ;GO READ REVERSE
007416  032777  000010  171162  RSF:   BIT     #10,@SWR        ;SEE IF SHOULD READ FORWARD
007424  001033                         BNE     RSEX            ;IF NOT: BR
007426  032767  010000  171126         BIT     #10000,RDCMD    ;SEE IF HAVE READ REVERSE
007434  001407                         BEQ     RSF0            ;IF NOT: BR
007436  016767  171134  171224         MOV     TSTAL,STAL
007444  004767  002570                 JSR     PC,STALL        ;DO READ STALL
007450  000167  000014                 JMP     RSF1
007454  032777  000001  171124  RSF0:  BIT     #1,@SWR         ;SEE IF WRITE
007462  001002                         BNE     RSF1            ;IF NOT: BR
007464  004767  002322                 JSR     PC,BKSP         ;GO BACKSPACE
007470  012767  000002  171064  PSF1:  MOV     #2,RDCMD        ;LOAD READ FORWARD COMMAND
007476  004767  000166                 JSR     PC,READ         ;GO READ
007502  005767  171154                 TST     EOTREC          ;SEE IF AT END OF TAPE
007506  100002                         BPL     RSEX            ;IF NOT: BR
007510  000167  174546                 JMP     REOT            ;ELSE GO TO REWIND
007514  005067  171142         RSEX:   CLR     EOTREC          ;CLEAR EOT FLAG
007520  000207                         RTS     PC              ;EXIT
007522  012767  010000  171032  RSFR:  MOV     #10000,RDCMD
007530  032777  000010  171050         BIT     #10,@SWR        ;SEE IF SHOULD READ FORWARD
007536  001013                         BNE     RSFR1           ;IF NOT: BR
007540  032777  000001  171040         BIT     #1,@SWR         ;SEE IF WRITE
007546  001002                         BNE     RSFR0           ;IF NOT: BR
007550  004767  002236                 JSR     PC,BKSP         ;GO BACKSPACE TO START
007554  012767  000002  171000  RSFR0: MOV     #2,RDCMD        ;LOAD READ FORWARD COMMAND
007562  004767  000102                 JSR     PC,READ         ;GO READ FORWARD
007566  032777  000004  171012  RSFR1: BIT     #4,@SWR         ;SEE IF SHOULD READ REVERSE
007574  001347                         BNE     RSEX            ;IF NOT: BR
007576  032767  010000  170756         BIT     #10000,RDCMD
007604  001005                         BNE     RSFR2           ;IF READ REVERSE: BR
```

```
 2520  007606  016767  170764  171054          MOV     TSTAL,STAL      ;DO READ STALL
 2521  007614  004767  002420                  JSR     PC,STALL
 2522  007620  012767  010000  170734  RSFR2:  MOV     #10000,RDCMD    ;LOAD READ REVERSE
 2523  007626  004767  000036                  JSR     PC,READ         ;GO READ REVERSE
 2524  007632  005767  171024                  TST     EOTREC          ;SEE IF AT END OF TAPE
 2525  007636  001411                          BEQ     RSFRX           ;IF NOT: BR
 2526  007640  166767  170710  171014          SUB     RCNT,EOTREC
 2527  007646  005467  171010                  NEG     EOTREC          ;SET TO PROPER RECORD NUMBER
 2528  007652  005267  171004                  INC     EOTREC
 2529  007656  000167  174400                  JMP     REOT            ;ELSE GO TO REWIND
 2530  007662  005067  170774          RSFRX:  CLR     EOTREC          ;CLEAR EOT FLAG
 2531  007666  000207                          RTS     PC              ;EXIT
 2532
```

```
2533                                    ;*********************************************************
2534                                    ;READ ROUTINE:
2535                                    ;
2536                                    ;THIS ROUTINE PERFORMS THE READ OPERATION DETERMINED
2537                                    ;BY THE READ SEQUENCE ROUTINE ONE RECORD AT A TIME.
2538                                    ;AT THE END OF EACH READ OPERATION THE STATUS REGISTER
2539                                    ;IS SCANNED FOR EITHER END OF TAPE OR BEGINNING OF TAPE.
2540                                    ;IF EOT WAS REACHED, CONTROL WILL BE PASSED TO
2541                                    ;THE EOT SUBROUTINE TO REWIND THE UNIT AND FLAG IT
2542                                    ;UNAVAILABLE UNTIL ALL UNITS HAVE REACHED EOT.
2543                                    ;IF BOT WAS REACHED AN ERROR IS PRINTED AND THE
2544                                    ;PROGRAM WILL HALT.  TESTING MAY BE RESUMED BY PRESSING
2545                                    ;THE CONTINUE SWITCH.
2546                                    ;IF A TAPE MARK IS EXPECTED (TM=1) THEN THE
2547                                    ;READ ROUTINE EXPECTS THE FIRST RECORD OF A
2548                                    ;READ REVERSE TO BE A TM, AND THE LAST RECORD
2549                                    ;OF A READ FORWARD TO BE A TM.  REMEMBER
2550                                    ;THAT THE TM ADDS ONE (1) TO THE TOTAL NUMBER
2551                                    ;OF RECORDS IN A BLOCK.
2552                                    ;CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13) DETERMINE WHETHER
2553                                    ;OR NOT TO CHECK FOR STATUS ERRORS (11) OR DATA ERRORS (13),
2554                                    ;CONSOLE SWITCH FIVE (5) IS USED TO CAUSE A CONTINUOUS
2555                                    ;READ AND SPACE (FORWARD OR REVERSE) OF THE CURRENT
2556                                    ;RECORD ON TAPE (YOZZLE).
2557                                    ;*********************************************************
2558
2559  007670  016700  170660       READ:  MOV     RCNT,R0          ;LOAD REC CNTR
2560  007674  005767  170762              TST     EOTREC           ;SEE IF EOT
2561  007700  100013                      BPL     RDA              ;IF NOT: BR
2562  007702  032767  010000  170652      BIT     #10000,RDCMD     ;SEE IF READ FORWARD
2563  007710  001407                      BEQ     RDA              ;IF SO: BR
2564  007712  042767  100000  170742      BIC     #100000,EOTREC   ;CLEAR FLAG
2565  007720  016703  170736              MOV     EOTREC,R3        ;GET MODIFIED RECORD COUNT
2566  007724  160300                      SUB     R3,R0            ;SET RECORD AT
2567  007726  005200                      INC     R0               ;SET TO PROPER NUMBER OF RECORDS
2568  007730  012767  024755  170716  RDA:  MOV     #MSG6,EMADDR     ;SET ERROR MSG ADDRESS
2569  007736  005067  170736              CLR     TMFLG
2570  007742  032767  010000  170612      BIT     #10000,RDCMD
2571  007750  001406                      BEQ     RD0              ;IF READ FORWARD: BR
2572  007752  005767  170606              TST     TMEX             ;SEE IF TM
2573  007756  001403                      BEQ     RD0              ;IF NOT: BR
2574  007760  005267  170714              INC     TMFLG            ;SET TM FLAG
2575  007764  005200                      INC     R0
2576  007766  016777  170564  170522  RD0:  MOV     FMCNT,@FC        ;LOAD CHAR CNTR
2577  007774  012777  033434  170512      MOV     #RDATA,@BA       ;LOAD DATA ADDR
2578  010002  032767  010000  170552      BIT     #10000,RDCMD     ;SEE IF READ REVERSE
2579  010010  001417                      BEQ     RD1A             ;IF NOT: BR
2580  010012  016703  170540              MOV     FMCNT,R3
2581  010016  005103                      COM     R3
2582  010020  032767  000020  170524      BIT     #20,UDES         ;SEE IF CORE DUMP
2583  010026  001402                      BEQ     RD1              ;IF NOT: BR
2584  010030  000241                      CLC
2585  010032  006003                      ROR     R3               ;R3 = FC/2
2586  010034  060377  170454       RD1:  ADD     R3,@BA           ;SET REVERSE BUS ADDRESS
2587  010040  012767  000076  170626      MOV     #76,MTC1         ;SET READ REVERSE
2588  010046  000403                      BR      RD1B
```

```
 2589  010050  012767  000070  170616  RD1A:   MOV     #70,MTC1          ;SET READ FORWARD
 2590  010056  012767  010070  170600  RD1B:   MOV     #RD2,RTRN         ;SET INTERRUPT RETURN ADDRESS
 2591  010064  000167  011332          RD1D:   JMP     TAPG              ;GO EXECUTE TAPE COMMAND
 2592  010070  032767  010000  170464  RD2:    BIT     #10000,RDCMD      ;SEE IF READ REVERSE
 2593  010076  001024                          BNE     RD3               ;IF SO: BR
 2594  010100  032777  000020  170414          BIT     #20,@DS
 2595  010106  001404                          BEQ     RD2B              ;AWAIT SWDN
 2596  010110  032777  000020  170404  RD2A:   BIT     #20,@DS
 2597  010116  001374                          BNE     RD2A              ;AWAIT TUR
 2598  010120  032777  002000  170374  RD2B:   BIT     #2000,@DS         ;SEE IF EOT
 2599  010126  001410                          BEQ     RD3               ;IF NOT: BR
 2600  010130  005767  170544                  TST     TMFLG             ;SEE IF TM
 2601  010134  001005                          BNE     RD3               ;IF SO: BR
 2602  010136  010067  170520                  MOV     R0,EOTREC
 2603  010142  052767  100000  170512          BIS     #100000,EOTREC    ;SET EOT FLAG
 2604  010150  032777  000002  170344  RD3:    BIT     #2,@DS            ;SEE IF AT LOAD POINT
 2605  010156  001411                          BEQ     RD4               ;IF NOT: BR
 2606  010160  004767  013014                  JSR     PC,PAPRT          ,PRINT CYCLE NUMBER
 2607  010164  012704  025131                  MOV     #MSG22,R4
 2608  010170  004767  013726                  JSR     PC,TTOUT          ;PRINT BOT ERROR
 2609  010174  000000                          HALT
 2610  010176  000167  172656                  JMP     STARTA            ;RESTART
 2611  010202  032777  004000  170376  RD4:    BIT     #4000,@SWR        ;SEE IF SHOULD CHECK ERRORS
 2612  010210  001121                          BNE     RD5               ;IF NOT: BR
 2613  010212  005767  170462                  TST     TMFLG
 2614  010216  001472                          BEQ     RD4B              ;IF NO TM EXPT: BR
 2615  010220  032777  000004  170274          BIT     #4,@DS
 2616  010226  001024                          BNE     RD4A              ;IF TM RECVD: BR
 2617  010230  012767  033434  011074          MOV     #RDATA,CADER      ;SAVE EXPT BUS ADDRESS
 2618  010236  012767  000002  011074          MOV     #2,DRVER          ;SET TM STATUS ERROR FLAG
 2619  010244  004767  010102                  JSR     PC,ERPT           ;GO PRINT TM ERROR
 2620  010250  016704  170422                  MOV     UNP,R4
 2621  010254  032767  010000  170300          BIT     #10000,RDCMD      ;SEE IF READ REVERSE
 2622  010262  001403                          BEQ     1$                ;IF NOT: BR
 2623  010264  005264  001150                  INC     RDERR1(R4)        ;BUMP READ REVERSE ERROR
 2624  010270  000502                          BR      RD6
 2625  010272  005264  001110          1$:     INC     RDER1(R4)         ;BUMP READ FORWARD ERROR
 2626  010276  000477                          BR      RD6
 2627  010300  012703  033434          RD4A:   MOV     #RDATA,R3
 2628  010304  032767  010000  170250          BIT     #10000,RDCMD      ;SEE IF READ REVERSE
 2629  010312  001007                          BNE     RD4A0             ;IF SO: BR
 2630  010314  032767  002000  170230          BIT     #2000,UDES        ;SEE IF IN PE
 2631  010322  001025                          BNE     RD4A2             ;IF SO: BR
 2632  010324  062703  000002                  ADD     #2,R3
 2633  010330  000422                          BR      RD4A2
 2634  010332  016704  170220          RD4A0:  MOV     FMCNT,R4
 2635  010336  005104                          COM     R4
 2636  010340  032767  000020  170204          BIT     #20,UDES          ;SEE IF CORE DUMP
 2637  010346  001402                          BEQ     RD4A1             ;IF NOT: BR
 2638  010350  000241                          CLC
 2639  010352  006004                          ROR     R4                ;SET TO FC/2
 2640  010354  060403          RD4A1:  ADD     R4,R3             ;SET EXPT BUS ADDRESS
 2641  010356  042703  000001                  BIC     #1,R3             ;MAKE EXPT ADDRESS EVEN
 2642  010362  032767  002000  170162          BIT     #2000,UDES        ;SEE IF IN PE
 2643  010370  001002                          BNE     RD4A2             ;IF SO: BR
 2644  010372  162703  000002                  SUB     #2,R3
```

```
2645  010376  004767  007210          RD4A2:  JSR     PC,ER2
2646  010402  000402                  RD4C:   BR      RD4C
2647  010404  004767  007104          RD4B:   JSR     PC,ERCHK        ;GO CHECK ERRORS
2648  010410  005767  170274          RD4C:   TST     SERFL
2649  010414  001417                          BEQ     RD5             ;IF NO ERROR: BR
2650  010416  016704  170254                  MOV     UNP,R4
2651  010422  032767  010000  170132          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2652  010430  001003                          BNE     RD4D            ;IF SO: BR
2653  010432  005264  001110                  INC     RDER1(R4)       ;BUMP READ FORWARD ERROR
2654  010436  000402                          BR      RD4E
2655  010440  005264  001150          RD4D:   INC     RDERR1(R4)      ;BUMP READ REVERSE ERROR
2656  010444  004767  000176          RD4E:   JSR     PC,RDRTY        ;GO RETRY
2657  010450  005067  170240                  CLR     RTYFL           ;CLEAR RETRY FLAG
2658  010454  032777  020000  170124  RD5:    BIT     #20000,@SWR     ;SEE IF SHOULD DO DATA CHECK
2659  010462  001005                          BNE     RD6             ;IF NOT; BR
2660  010464  005767  170210                  TST     TMFLG
2661  010470  001002                          BNE     RD6
2662  010472  004767  005110                  JSR     PC,DCHK         ;GO CHECK DATA
2663  010476  005067  170206          RD6:    CLR     SERFL           ;CLEAR STATUS ERROR FLAG
2664  010502  004767  003646                  JSR     PC,DS3          ;CLEAR BUFFER
2665  010506  032777  000040  170072          BIT     #40,@SWR        ;SEE IF SHOULD YOZZLE
2666  010514  001402                          BEQ     RD7             ;IF NOT: BR
2667  010516  004767  000532                  JSR     PC,YOZ          ;ELSE GO YOZZLE
2668  010522  016767  170044  170140  RD7:    MOV     RSTAL,STAL      ;SET DELAY
2669  010530  004767  001504                  JSR     PC,STALL        ;STALL
2670  010534  032767  010000  170020          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2671  010542  001403                          BEQ     RD7A            ;IF NOT: BR
2672  010544  005067  170130                  CLR     TMFLG           ;CLEAR TAPE MARK FLAG
2673  010550  000405                          BR      RD10
2674  010552  005767  170104          RD7A:   TST     EOTREC          ;SEE IF EOT FOUND
2675  010556  100002                          BPL     RD10            ;IF NOT: BR
2676  010560  012700  000001                  MOV     #1,R0           ;SET TO EOT
2677  010564  005300                  RD10:   DEC     R0
2678  010566  001402                          BEQ     RD11            ;IF DONE ALL: BR
2679  010570  000167  177172                  JMP     RD0
2680  010574  032767  010000  167760  RD11:   BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2681  010602  001016                          BNE     RDEX            ;IF SO: BR
2682  010604  005767  170052                  TST     EOTREC          ;SEE IF FOUND EOT
2683  010610  100413                          BMI     RDEX            ;IF SO: BR
2684  010612  005767  167746                  TST     TMEX            ;SEE IF TM EXPECTED
2685  010616  001410                          BEQ     RDEX            ;IF NOT: BR
2686  010620  005767  170054                  TST     TMFLG           ;SEE IF TM FOUND
2687  010624  001005                          BNE     RDEX            ;IF SO: BR
2688  010626  005267  170046                  INC     TMFLG           ;ELSE SET FLAG
2689  010632  005200                          INC     R0              ;SET RECORD COUNT TO ONE
2690  010634  000167  177126                  JMP     RD0             ;GO READ TM
2691  010640  005067  170034          RDEX:   CLR     TMFLG
2692  010644  000207                  RDX:    RTS     PC              ;EXIT
```

```
2693                                    ;*********************************************************
2694                                    ;READ ERROR RETRY SUBROUTINE:
2695                                    ;
2696                                    ;THIS SUBROUTINE WILL RETRY ALL DATA RELATED
2697                                    ;READ ERRORS UP TO EIGHT (8) TIMES.  IF ALL
2698                                    ;FOUR RETRIES ARE BAD, IT IS CONSIDERED
2699                                    ;A HARD ERROR.  IF ANY ARE GOOD, IT IS A
2700                                    ;SOFT ERROR.  RETRIES MAY BE INHIBITED
2701                                    ;VIA SWITCH FOUR (SW4=0: INHIBIT RETRIES)
2702                                    ;*********************************************************
2703
2704  010646  032777  000020  167732  RDRTY:  BIT     #20,@SWR        ;SEE IF RETRY INHIBITED
2705  010654  001001                          BNE     RDRT0           ;IF NOT: BR
2706  010656  000207                          RTS     PC              ;ELSE RETURN
2707  010660  016703  170040          RDRT0:  MOV     ERSAV,R3
2708  010664  042703  102700                  BIC     #102700,R3      ;MARK NON-RECOVERABLE ERROR BITS
2709  010670  001411                          BEQ     RDRT1           ;IF NOT: BR
2710  010672  004767  012302                  JSR     PC,PAPRT        ;PRINT HEADER
2711  010676  012704  026630                  MOV     #MSG79,R4
2712  010702  004767  013214                  JSR     PC,TTOUT        ;PRINT NON-RECOVERABLE MESSAGE
2713  010706  004767  000324                  JSR     PC,NRTP         ;PRINT ER FOR NON-RETRYABLE ERROR
2714  010712  000207          RDRT1A: RTS     PC              ;RETURN
2715  010714  032777  002000  167664  RDRT1:  BIT     #2000,@SWR      ;SEE IF PRINT INHIBITED
2716  010722  001004                          BNE     RDRT1B          ;IF SO: BR
2717  010724  012704  026303                  MOV     #MSG64,R4
2718  010730  004767  013166                  JSR     PC,TTOUT        ;PRINT ORIGINAL ERROR TAG
2719  010734  005067  167744          RDRT1B: CLR     RTCNT           ;CLEAR RETRY COUNTER
2720  010740  005067  167744          RDRTG:  CLR     SERFL           ;CLEAR STATUS ERROR FLAG
2721  010744  012767  000002  167742          MOV     #2,RTYFL        ;SET READ RETRY FLAG
2722  010752  004767  000276                  JSR     PC,YOZ          ;GO TO YOZZLE TO RETRY READ
2723  010756  005767  167726                  TST     SERFL           ;SEE IF RETRY ERROR
2724  010762  001034                          BNE     RDRT5           ;IF SO: BR
2725  010764  032777  002000  167614          BIT     #2000,@SWR
2726  010772  001014                          BNE     RDRT2
2727  010774  012704  026774                  MOV     #MSG105,R4
2728  011000  004767  013116                  JSR     PC,TTOUT        ;PRINT RECOVERED MESSAGE
2729  011004  012704  026325                  MOV     #MSG65,R4
2730  011010  004767  013106                  JSR     PC,TTOUT        ;PRINT RETRY TAG
2731  011014  016703  167664                  MOV     RTCNT,R3
2732  011020  004767  013250                  JSR     PC,OCTP         ;PRINT RETRY NUMBER
2733  011024  016704  167646          RDRT2:  MOV     UNP,R4
2734  011030  032767  010000  167524          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2735  011036  001003                          BNE     RDRT3           ;IF SO: BR
2736  011040  005264  002670                  INC     RFSOFT(R4)      ;ELSO BUMP FORWARD SOFT ERROR COUNTER
2737  011044  000402                          BR      RDRT4
2738  011046  005264  002710          RDRT3:  INC     RRSOFT(R4)      ;BUMP ERRORS SOFT CNTR
2739  011052  000207          RDRT4:  RTS     PC              ;RETURN
2740  011054  016703  167644          RDRT5:  MOV     ERSAV,R3        ;GET ER
2741  011060  042703  102700                  BIC     #102700,R3      ;MASK RECOVERABLE BITS
2742  011064  001414                          BEQ     RDRT5A          ;IF RECOVERABLE: BR
2743  011066  004767  012106                  JSR     PC,PAPRT        ;PRINT HEADER
2744  011072  012704  026630                  MOV     #MSG79,R4
2745  011076  004767  013020                  JSR     PC,TTOUT        ;PRINT NON-RECOVERABLE MSG
2746  011102  004767  000130                  JSR     PC,NRTP         ;PRINT ER
2747  011106  012767  000001  167534          MOV     #1,TEMP3        ;SET FLAG
2748  011114  000404                          BR      RDRT5B
```

```
 2749  011116  032777  002000  167462  RDRT5A: BIT     #2000,@SWR      ;SEE IF PRINT INHIBITED
 2750  011124  001016                          BNE     RDRT6           ;IF SO: BR
 2751  011126  012704  026325          RDRT5B: MOV     #MSG65,R4
 2752  011132  004767  012764                  JSR     PC,TTOUT        ;PRINT RETRY TAG
 2753  011136  016703  167542                  MOV     RTCNT,R3
 2754  011142  004767  013126                  JSR     PC,OCTP         ;PRINT RETRY NUMBER
 2755  011146  005767  167476                  TST     TEMP3           ;SEE IF DID NON-RECOVERABLE
 2756  011152  001403                          BEQ     RDRT6           ;IF NOT: BR
 2757  011154  005067  167470                  CLR     TEMP3           ;CLEAR FLAG
 2758  011160  000207                          RTS     PC              ;EXIT
 2759  011162  005267  167516          RDRT6:  INC     RTCNT
 2760  011166  026767  167512  167406        CMP     RTCNT,RETRY     ;SEE IF DONE 8 RETRIES
 2761  011174  001261                          BNE     RDRTG           ;IF NOT: BR
 2762  011176  012704  027327                  MOV     #MSG115,R4
 2763  011202  004767  012714                  JSR     PC,TTOUT        ;PRINT HARD ERROR MESSAGE
 2764  011206  016704  167464                  MOV     UNP,R4
 2765  011212  032767  010000  167342        BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 2766  011220  001003                          BNE     RDRT7           ;IF SO: BR
 2767  011222  005264  002730                  INC     RFHARD(R4)      ;BUMP FORWARD HARD ERROR CNTR
 2768  011226  000402                          BR      RDRTX
 2769  011230  005264  002750          RDRT7:  INC     RRHARD(R4)      ;BUMP REVERSE HARD ERROR CNTR
 2770  011234  000207                  RDRTX:  RTS     PC              ;RETURN
 2771
 2772  011236  016703  167462          NRTP:   MOV     ERSAV,R3        ;GET ER REGISTER
 2773  011242  004767  013026                  JSR     PC,OCTP         ;PRINT ER
 2774  011246  004767  010104                  JSR     PC,FRPRT        ;PRINT F OR R
 2775  011252  000207                          RTS     PC              ;RETURN
```

```
2776                                    ;*****************************************************
2777                                    ;YOZZLE SUBROUTINE:
2778                                    ;
2779                                    ;THIS SUBROUTINE, ENTERED VIA SWITCH FIVE (5), IS USED TO PERFORM
2780                                    ;A CONTINUOUS READ AND SPACE OVER OF THE CURRENT RECORD ON TAPE.
2781                                    ;FULL STATUS AND DATA CHECKING MAY BE PERFORMED
2782                                    ;OR NOT VIA CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13).
2783                                    ;A SOFTWARE DELAY IS PERFORMED BETWEEN EACH READ
2784                                    ;AND SPACE OPERATION AND MAY BE VARIED BY TYPING
2785                                    ;CNTRL C ON THE TTY AND ENTERING A VALUE IN RESPONSE
2786                                    ;TO THE PRINTED REQUEST.
2787                                    ;*****************************************************
2788
2789  011254  012777  000001  167326  YOZ:    MOV     #1,@TKS         ;SET TTY ENABLE
2790  011262  016767  167312  167400          MOV     YSTAL,STAL
2791  011270  004767  000744                  JSR     PC,STALL        ;DO YOZZLE STALL
2792  011274  012777  177777  167214  YOZ0:   MOV     #-1,@FC         ;SET TO 1 RECORD SPACING
2793  011302  032767  010000  167252          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2794  011310  001404                          BEQ     YOZA            ;IF NOT: BR
2795  011312  112767  000030  167354          MOVB    #30,MTC1        ;SET TO SPACE FORWARD
2796  011320  000403                          BR      YOZB
2797  011322  112767  000032  167344  YOZA:   MOVB    #32,MTC1        ;SET TO SPACE REVERSE
2798  011330  012767  011350  167326  YOZB:   MOV     #YOZC,RTRN      ;SET RETURN ADDRESS
2799  011336  012767  177775  167324          MOV     #177775,STAL    ;SET TIME MULTIPLIER
2800  011344  000167  010052                  JMP     TAPG            ;GO YOZZLE
2801  011350  005767  167324          YOZC:   TST     TMFLG           ;SEE IF TM
2802  011354  001404                          BEQ     1$              ;IF NOT: BR
2803  011356  012767  040000  167304          MOV     #40000,STAL     ;SET TM STALL
2804  011364  000403                          BR      2$
2805  011366  016767  167206  167274  1$:     MOV     YSTAL,STAL
2806  011374  004767  000640          2$:     JSR     PC,STALL        ;DO YOZZLE STALL
2807  011400  012777  033434  167106          MOV     #RDATA,@BA      ;SET BUS ADDRESS
2808  011406  032767  010000  167146          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2809  011414  001417                          BEQ     YOZC1           ;IF NOT: BR
2810  011416  016703  167134                  MOV     FMCNT,R3
2811  011422  005103                          COM     R3
2812  011424  032767  000020  167120          BIT     #20,UDES        ;SEE IF CORE DUMP
2813  011432  001402                          BEQ     YOZC0           ;IF NOT: BR
2814  011434  000241                          CLC
2815  011436  006003                          ROR     R3              ;R3 = FC/2
2816  011440  060377  167050          YOZC0:  ADD     R3,@BA          ;SET REVERSE BUS ADDRESS
2817  011444  012767  000076  167222          MOV     #76,MTC1        ;SET READ REVERSE
2818  011452  000403                          BR      YOZC2
2819  011454  012767  000070  167212  YOZC1:  MOV     #70,MTC1        ;SET READ FORWARD
2820  011462  016777  167070  167026  YOZC2:  MOV     FMCNT,@FC       ;SET CHARACTER COUNT
2821  011470  012767  011502  167166          MOV     #YOZD,RTRN      ;SET RETURN ADDRESS
2822  011476  000167  007720                  JMP     TAPG            ;GO READ
2823  011502  032777  004000  167076  YOZD:   BIT     #4000,@SWR      ;SEE IF SHOULD CHECK ERRORS
2824  011510  001051                          BNE     YOZE            ;IF NOT: BR
2825  011512  005767  167162                  TST     TMFLG           ;SEE IF TAPE MARK TIME
2826  011516  001444                          BEQ     YOZD1           ;IF NOT: BR
2827  011520  032767  010000  167034          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2828  011526  001426                          BEQ     YOZD0           ;IF NOT: BR
2829  011530  012703  033434                  MOV     #RDATA,R3
2830  011534  016704  167016                  MOV     FMCNT,R4
2831  011540  005104                          COM     R4
```

```
 2832  011542  032767  000020  167002          BIT     #20,UDES        ;SEE IF CORE DUMP
 2833  011550  001402                          BEQ     YOZD4           ;IF NOT: BR
 2834  011552  000241                          CLC
 2835  011554  006004                          ROR     R4              ;SET TO FC/2
 2836  011556  060403                  YOZD4:  ADD     R4,R3           ;SET EXPT BUS ADDRESS
 2837  011560  042703  000001                  BIC     #1,R3           ;MAKE EXPT ADDRESS EVEN
 2838  011564  032767  002000  166760          BIT     #2000,UDES      ;SEE IF PE
 2839  011572  001001                          BNE     YOZD2           ;IF SO: BR
 2840  011574  005743                          TST     -(R3)           ;SET EXPT BA
 2841  011576  004767  006J1`                  JSR     PC,ER2          ;GO CHECK ERRORS
 2842  011602  000430                          BR      YOZF
 2843  011604  012703  ..3434                  MOV     #RDATA,R3
 2844  011610  032767  002000  166734          BIT     #2000,UDES      ;SEE IF PE
 2845  011616  001001                          BNE     YOZD3           ;IF SO: BR
 2846  011620  005723                          TST     (R3)+           ;SET EXPT BA
 2847  011622  004767  005764                  JSR     PC,ER2          ;GO CHECK ERRORS
 2848  011626  000416                          BR      YOZF
 2849  011630  004767  005660          YOZD1:  JSR     PC,ERCHK        ;ELSE GO CHECK ERRORS
 2850  011634  032777  020000  166744  YOZE:   BIT     #20000,@SWR     ;SEE IF SHOULD CHECK DATA
 2851  011642  001010                          BNE     YOZF            ;IF NOT: BR
 2852  011644  005767  167030                  TST     TMFLG           ;SEE IF TAPE MARK
 2853  011650  001005                          BNE     YOZF            ;IF SO: BR
 2854  011652  005767  167036                  TST     RTYFL           ;SEE IF RETRY
 2855  011656  001004                          BNE     YOZF0           ;IF SO: BR
 2856  011660  004767  003722                  JSR     PC,DCHK         ;ELSE GO CHECK DATA
 2857  011664  004767  002464          YOZF:   JSR     PC,DS3          ;GO CLEAR DATA AREA
 2858  011670  105777  166714          YOZF0:  TSTB    @TKS            ;SEE IF HAVE NEW STALL VALUE
 2859  011674  100034                          BPL     YOZG            ;IF NOT: BR
 2860  011676  122777  000203  166706          CMPB    #203,@TKB       ;SEE IF CONT C
 2861  011704  001030                          BNE     YOZG            ;IF NOT: BR
 2862  011706  012704  026005                  MOV     #MSG44,R4
 2863  011712  004767  012204                  JSR     PC,TTOUT        ;PRINT YSTALL REQUEST
 2864  011716  016703  166656                  MOV     YSTAL,R3
 2865  011722  004767  012346                  JSR     PC,OCTP         ;PRINT PRESENT STALL
 2866  011726  010067  166716                  MOV     R0,TEMP3        ;SAVE R0(REC CNT)
 2867  011732  012705  000600                  MOV     #YSTAL,R5       ;SET ADDRESS OF YSTL
 2868  011736  012701  000006                  MOV     #6,R1           ;SET NUMBER OF CHAR TO INPUT
 2869  011742  012702  177777                  MOV     #-1,R2          ;SET MAXIMUM LIMIT
 2870  011746  012703  002000                  MOV     #2000,R3        ;SET MINIMUM LIMIT
 2871  011752  004767  011706                  JSR     PC,TTR          ;GO GET VALUE
 2872  011756  016700  166666                  MOV     TEMP3,R0        ;RESTORE R0(REC CNTR)
 2873  011762  000167  177266                  JMP     YOZ             ;RESTART YOZZLE
 2874  011766  032777  000040  166612  YOZG:   BIT     #40,@SWR        ;SEE IF SHOULD CONTINUE YOZZLE
 2875  011774  001402                          BEQ     YOZH            ;IF NOT: BR
 2876  011776  000167  177272                  JMP     YOZ0
 2877  012002  012777  000100  166600  YOZH:   MOV     #100,@TKS       ;SET TTY INTERRUPT ENABLE
 2878  012010  000207                          RTS     PC              ;EXIT
 2879
```

```
2880
2881                                    ;*********************************************************
2882                                    ;BACKSPACE SUBROUTINE:
2883                                    ;
2884                                    ;THIS SUBROUTINE IS USED TO PERFORM THE
2885                                    ;BACKSPACE OPERATION REQUIRED BY THE READ
2886                                    ;ROUTINE FOR READ FORWARD AFTER WRITING.
2887                                    ;IF A TAPE MARK IS EXPECTED (TM=1) THEN THE SPACE
2888                                    ;ROUTINE ASSUMES THAT THE TM WILL BE FIRST WHEN
2889                                    ;BACKSPACING.  THEREFORE TWO OPERATIONS ARE REQUIRED
2890                                    ;TO SPACE OVER A BLOCK.  FIRST SPACE OVER THE TM, THEN
2891                                    ;SPACE OVER THE DATA RECORDS.
2892                                    ;A CHECK FOR RECORD COUNT ZERO IS MADE AT THE
2893                                    ;END OF THE SPACE OPERATION TO ASSURE THAT PROPER
2894                                    ;TAPE POSITIONING WAS DONE.
2895                                    ;*********************************************************
2896
2897  012012  016767  166560  166650  BKSP:   MOV     TSTAL,STAL
2898  012020  004767  000214                  JSR     PC,STALL        ;DO TURN AROUND STALL
2899  012024  012767  025005  166622          MOV     #MSG10,EMADDR
2900  012032  012777  033434  166454          MOV     #RDATA,@BA
2901  012040  005767  166520                  TST     TMEX            ;SEE IF TM
2902  012044  001440                          BEQ     B0              ;IF NOT: BR
2903  012046  012777  177777  166442          MOV     #-1,@FC
2904  012054  012767  000032  166612          MOV     #32,MTC1
2905  012062  012767  012074  166574          MOV     #BKTM,RTRN
2906  012070  000167  007326                  JMP     TAPG            ;SPACE TO TM
2907  012074  032777  010000  166504  BKTM:   BIT     #10000,@SWR     ;SEE IF SHOULD CHECK ERROR
2908  012102  001021                          BNE     B0              ;IF NOT: BR
2909  012104  012767  026217  166542          MOV     #MSG55,EMADDR
2910  012112  032777  000004  166402          BIT     #4,@DS          ;SEE IF TM
2911  012120  001006                          BNE     BKTM0           ;IF SO: BR
2912  012122  012767  033434  007202          MOV     #RDATA,CADER
2913  012130  004767  006216                  JSR     PC,ERPT         ;PRINT ERROR
2914  012134  000404                          BR      B0
2915  012136  012703  033434          BKTM0:  MOV     #RDATA,R3
2916  012142  004767  005444                  JSR     PC,ER2
2917  012146  016700  166402          B0:     MOV     RCNT,R0
2918  012152  005100                          COM     R0              ;BUILD SPACE AMOUNT
2919  012154  005200                          INC     R0
2920  012156  012767  025005  166470          MOV     #MSG10,EMADDR   ;SET ERROR MESG ADDRESS
2921  012164  010077  166326                  MOV     R0,@FC
2922  012170  012767  000032  166476  BKRT:   MOV     #32,MTC1        ;SET SPACE REVERSE
2923  012176  012767  012214  166460          MOV     #B1,RTRN        ;SET RETURN ADDRESS
2924  012204  010067  166460                  MOV     R0,STAL         ;SET INTERRUPT TIME MULTIPLIER
2925  012210  000167  007206                  JMP     TAPG            ;GO DO SPACE
2926  012214  012703  033434          B1:     MOV     #RDATA,R3
2927  012220  004767  005366                  JSR     PC,ER2
2928  012224  016767  166346  166436  B2:     MOV     TSTAL,STAL      ;DO STALL
2929  012232  004767  000002                  JSR     PC,STALL        ;STALL
2930  012236  000207                          RTS     PC              ;EXIT
2931
```

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

012240  005367  166424          STALL:  DEC     STAL
012244  001375                          BNE     STALL           ;DELAY
012246  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.
                                        ;*********************************************************

012250  012701  177760          CCNTR:  MOV     #-20,R1         ;SET HIGH LIMIT
012254  012702  174000                  MOV     #-4000,R2       ;SET LOW LIMIT
012260  004767  011346                  JSR     PC,RANG         ;GO GENERATE NUMBER
012264  042767  000001  166334          BIC     #1,RANSAV
012272  016767  166330  166256          MOV     RANSAV,FMCNT    ;SET CHAR COUNT
012300  012767  177777  002102          MOV     #-1,PATS        ;PRESET DATA PATTERN
012306  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.
                                        ;*********************************************************

012310  012702  000001          RCNTR:  MOV     #1,R2           ;SET LOW LIMIT
012314  012701  000500                  MOV     #500,R1         ;SET HIGH LIMIT
012320  004767  011306                  JSR     PC,RANG         ;GO GENERATE NUMBER
012324  016767  166276  166222          MOV     RANSAV,RCNT     ;SET RECORD COUNT
012332  000207                          RTS     PC              ;EXIT
```

```
                                ;*********************************************************
                                ;TEST CONDITION ENTRY ROUTINE:
                                ;
                                ;THIS ROUTINE IS USED TO ALLOW THE OPERATOR
                                ;TO ENTER, AT THE TTY, THE NECESSARY PARAMETERS
                                ;TO RUN THE PROGRAM AS HE WISHES.  THE
                                ;ROUTINE IS ONLY ENTERED UPON INITIAL STARTING
                                ;FROM LOCATION 200(8).
                                ;THE MAIN PURPOSE OF THIS ROUTINE IS TO ESTABLISH
                                ;A TABLE OF DEVICES TO BE TESTED.  THIS TABLE
                                ;CONSISTS OF AN ENTRY FOR EACH OF ONE (1) TO
                                ;EIGHT (8) DEVICES.  EACH ENTRY CONTAINS THE
                                ;SLAVE NUMBER, DENSITY, PARITY, AND
                                ;FORMAT.  THE INFORMATION IS ENTERED
                                ;IN RESPONSE TO PRINTED REQUESTS AT THE TTY.
                                ;SLAVES MAY BE ENTERED IN ANY ORDER.  EACH
                                ;PARAMETER IS CHECKED FOR LEGALITY BEFORE BEING
                                ;SET INTO THE TABLE.
                                ;THE DRIVE NUMBER REQUEST WILL ALSO CHECK THE MASSBUS
                                ;FOR THE PRESENCE OF THE REQUESTED DRIVE.  IF IT IS NOT FOUND,
                                ;A NON-EXIST DRIVE MESSAGE WILL BE PRINTED AND ANOTHER DRIVE
                                ;REQUEST MADE.  WHEN THE DRIVE IS FOUND, THE RESPONSE IS STORED
                                ;AND CONTROL PASSED TO THE SLAVE SELECT ROUTINE.
                                ;THE SLAVE SELECT ROUTINE ALSO CHECKS FOR THE PRESENCE OF THE
                                ;SLAVE.  IF IT IS NOT PRESENT, A MESSAGE IS PRINTED AND ANOTHER
                                ;REQUEST IS ISSUED.  WHEN THE SELECTED SLAVE IS FOUND TO BE
                                ;PRESENT, A MESSAGE IS PRINTED IF IT IS A 7 CHANNEL DRIVE
                                ;TO ASSIST IN SELECTING DENSITY, PARITY, AND FORMAT.
                                ;UPON COMPLETION OF THE DEVICE TABLE, REQUESTS
                                ;ARE PRINTED FOR ENTRY OF THE NUMBER OF CHARACTERS
                                ;PER RECORD AND THE NUMBER OF RECORDS PER BLOCK.  THE
                                ;NEXT REQUEST IS FOR A PATTERN NUMBER TO BE USED
                                ;FOR WRITING AND CHECKING OF READ DATA.
                                ;FOLLOWING THE PATTERN REQUEST IS THE TAPE MARK OPTION.
                                ;RESPONDING TO THE REQUEST (TM=) WITH A ONE (1)
                                ;WILL CAUSE THE PROGRAM TO WRITE A TM AT THE
                                ;END OF EACH DATA BLOCK AND TO EXPECT THE
                                ;TM TO BE DETECTED IN EITHER READ FORWARD AND REVERSE
                                ;OR DURING SPACE OPERATION.  A RESPONSE OF ZERO (TM=0)
                                ;DISALLOWS WRITTING OF THE TM AND CAUSES THE READ
                                ;AND SPACE ROUTINES TO EXPECT NO TM TO BE PRESENT.
                                ;THE LAST REQUESTS ARE FOR ENTRY OF THE DESIRED
                                ;WRITE, READ, AND TURN AROUND STALLS.
                                ;*********************************************************

012334  005767  166276  TINP:   TST     TINF            ;SEE IF SHOULD INPUT FROM TTY
012340  001001            BNE     TINPA           ;IF SO: BR
012342  000207            RTS     PC              ;EXIT
012344  005067  166326  TINPA:  CLR     UNP             ;CLEAR TABLE POINTER
012350  005067  172426          CLR     REOTC           ;CLEAR EOT UNIT COUNTER
012354  012700  000010          MOV     #10,R0          ;SET SIZE OF TABLE
012360  012701  000746          MOV     #UN1,R1         ;SET START OF TABLE
012364  005021          TINPB:  CLR     (R1)+           ;CLEAR TABLE
012366  005300            DEC     R0              ;SEE IF DONE
012370  001375            BNE     TINPB           ;IF NOT: BR
```

```
 3048  012372  012704  025442                 MOV     #MSG31,R4
 3049  012376  005767  166334                 TST     ASEQF           ;SEE IF AUTO SEQ
 3050  012402  001402                         BEQ     TINPB1          ;IF NOT: BR
 3051  012404  012704  025354                 MOV     #MSG30,R4       ;SET AUTO SEQ HDR
 3052  012410  004767  011506       TINPB1:   JSR     PC,TTOUT        ;PRINT PROGRAM NAME
 3053  012414  005767  001526                 TST     SCVFL           ;SEE IF SHORT CONVERSATION
 3054  012420  001073                         BNE     TINPC           ;IF SO: BR
 3055  012422  012704  026507                 MOV     #MSG74,R4
 3056  012426  004767  011470                 JSR     PC,TTOUT        ;REQUEST STARTING REGISTER ADDRESS
 3057  012432  016703  166106                 MOV     REGS,R3
 3058  012436  004767  011632                 JSR     PC,OCTP         ;PRINT CURRENT REG START
 3059  012442  012705  000544                 MOV     #REGS,R5        ;SAVE ADDRESS LOCATION
 3060  012446  012701  000006                 MOV     #6,R1           ;SET SIZE OF ENTRY
 3061  012452  012702  177400                 MOV     #177400,R2      ;SET UPPER LIMIT
 3062  012456  012703  160000                 MOV     #160000,R3      ;SET LOWER LIMIT
 3063  012462  004767  011176                 JSR     PC,TTR          ;GO GET RESPONSE
 3064  012466  012704  026532                 MOV     #MSG75,R4
 3065  012472  004767  011424                 JSR     PC,TTOUT        ;GO REQUEST VECTOR ADDRESS
 3066  012476  016703  166044                 MOV     VECT,R3
 3067  012502  004767  011566                 JSR     PC,OCTP         ;PRINT CURRENT VECTOR
 3068  012506  012705  000546                 MOV     #VECT,R5        ;SET SAVE LOCATION
 3069  012512  012701  000003                 MOV     #3,R1           ;SET SIZE OF ENTRY
 3070  012516  012702  000474                 MOV     #474,R2         ;SET UPPER LIMIT
 3071  012522  012703  000100                 MOV     #100,R3         ;SET LOWER LIMIT
 3072  012526  004767  011132                 JSR     PC,TTR          ;GO GET RESPONSE
 3073  012532  016700  166010                 MOV     VECT,R0         ;GET VECTOR ADDRESS
 3074  012536  012720  022216                 MOV     #MTINT,(R0)+    ;LOAD VECTOR WITH HANDLER ADDRESS
 3075  012542  012710  000340                 MOV     #340,(R0)       ;LOAD PRIORITY LEVEL
 3076  012546  016700  165772                 MOV     REGS,R0         ;GET STARTING REGISTER ADDRESS
 3077  012552  012701  000016                 MOV     #16,R1          ;SET NUMBER OF REGISTERS
 3078  012556  012702  000510                 MOV     #C1,R2          ;GET FIRST ADDRESS LOCATION
 3079  012562  010022             TINPB0:     MOV     R0,(R2)+        ;BUILD TABLE OF ADDRESSES
 3080  012564  062700  000002                 ADD     #2,R0           ;BUMP ADDRESS
 3081  012570  005301                         DEC     R1              ;SEE IF DONE
 3082  012572  001373                         BNE     TINPB0          ;IF NOT: BR
 3083  012574  005767  166136                 TST     ASEQF           ;SEE IF AUTO SEQ
 3084  012600  001403                         BEQ     TINPC           ;IF NOT: BR
 3085  012602  005726                         TST     (SP)+           ;RESET STACK POINTER
 3086  012604  000167  007424                 JMP     ASEQ            ;GO TO AUTO SEQUENCE
 3087  012610  012777  000040  165702 TINPC:  MOV     #40,@CS         ;INITIALIZE
 3088  012616  012704  026154                 MOV     #MSG52,R4
 3089  012622  004767  011274                 JSR     PC,TTOUT        ;REQUEST DRIVE NUMBER
 3090  012626  012705  000550                 MOV     #DVN,R5         ;GET ADDRESS
 3091  012632  012701  000001                 MOV     #1,R1           ;SET SIZE OF RESPONSE
 3092  012636  012702  000007                 MOV     #7,R2           ;SET UPPER LIMIT
 3093  012642  012703  000000                 MOV     #0,R3           ;SET LOWER LIMIT
 3094  012646  004767  011012                 JSR     PC,TTR          ;GO GET DRIVE NUMBER
 3095  012652  016777  165672  165640         MOV     DVN,@CS
 3096  012660  005777  165624                 TST     @C1             ;ACCESS DRIVE
 3097  012664  032777  010000  165626         BIT     #10000,@CS      ;SEE IF NED
 3098  012672  001413                         BEQ     TINP0           ;IF NOT: BR
 3099  012674  012704  026444                 MOV     #MSG71,R4
 3100  012700  004767  011216                 JSR     PC,TTOUT        ;PRINT NED
 3101  012704  016704  165600                 MOV     C1,R4
 3102  012710  005204                         INC     R4
 3103  012712  152714  000100                 BISB    #100,(R4)       ;CLEAR TRE
```

```
3104  012716  000167  177666                JMP     TINPC           ;RETRY DVN
3105  012722  012704  025540          TINP0:  MOV     #MSG32,R4
3106  012726  004767  011170                JSR     PC,TTOUT        ;PRINT UNIT NUMBER REQUEST
3107  012732  005067  165710                CLR     TEMP2           ;CLEAR BUFFER
3108  012736  012705  000646                MOV     #TEMP2,R5       ;SET UNIT DESCRIPTION BUFFER ADDRESS
3109  012742  012701  000001                MOV     #1,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3110  012746  012702  000007                MOV     #7,R2           ;SET MAXIMUM LIMIT
3111  012752  012703  000000                MOV     #0,R3           ;SET MINIMUM LIMIT
3112  012756  004767  010702                JSR     PC,TTR          ;GO GET UNIT NUMBER
3113  012762  005767  165656                TST     TEMP1           ;SEE IF HAVE NEW PARAMETER
3114  012766  001014                        BNE     TINP0B          ;IF SO: BR
3115  012770  005767  165702                TST     UNP             ;SEE IF FIRST ENTRY
3116  012774  001002                        BNE     TINP0A          ;IF NOT: BR
3117  012776  000167  177720                JMP     TINP0           ;ELSE RETRY
3118  013002  016700  165670          TINP0A: MOV     UNP,R0
3119  013006  012760  177777  000746        MOV     #-1,UN1(R0)     ;SET END UNIT TABLE
3120  013014  000167  000402                JMP     TINP2C          ;GO GET RECORD COUNT
3121  013020  016700  165652          TINP0B: MOV     UNP,R0
3122  013024  042760  000007  000746        BIC     #7,UN1(R0)      ;CLEAR UNIT NUMBER
3123  013032  004767  001124                JSR     PC,TPOS1        ;GO LOAD UNIT NUMBER TO PROPER POSITION
3124  013036  012777  000040  165454        MOV     #40,@CS
3125  013044  016777  165500  165446        MOV     DVN,@CS
3126  013052  016077  000746  165462        MOV     UN1(R0),@C2     ;LOAD UNIT NUMBER
3127  013060  032777  002000  165450  TINP0C: BIT     #2000,@DT       ;SEE  IF SLAVE PRESENT
3128  013066  001006                        BNE     TINP0D          ;IF SO: BR
3129  013070  012704  026232                MOV     #MSG57,R4
3130  013074  004767  011022                JSR     PC,TTOUT        ;PRINT NON-EXIST SLAVE
3131  013100  000167  177616                JMP     TINP0           ;REDO
3132  013104  022777  142012  165424  TINP0D: CMP     #142012,@DT     ;SEE IF 9TRK TM02,TU45
3133  013112  001410                        BEQ     TINP0E          ;IF SO: BR
3134  013114  012704  026127                MOV     #MSG50,R4       ;ILLEGAL DRIVE TYPE
3135  013120  004767  010776                JSR     PC,TTOUT        ;GO PRINT
3136  013124  017703  165406                MOV     @DT,R3
3137  013130  004767  011140                JSR     PC,OCTP         ;PRINT DRIVE TYPE REGISTER
3138  013134  012704  024777          TINP0E: MOV     #MSG9,R4
3139  013140  004767  010756                JSR     PC,TTOUT        ;PRINT SERIAL NUMBER TAG
3140  013144  017703  165370                MOV     @SN,R3
3141  013150  004767  011446                JSR     PC,SNPT         ;PRINT SERIAL NUMBER
3142  013154  012704  025561          TINP1:  MOV     #MSG33,R4
3143  013160  004767  010736                JSR     PC,TTOUT        ;PRINT DENSITY REQUEST
3144  013164  005067  165456                CLR     TEMP2           ;CLEAR BUFFER
3145  013170  012701  000001                MOV     #1,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3146  013174  012702  000007                MOV     #7,R2           ;SET MAXIMUM LIMIT
3147  013200  012703  000000                MOV     #0,R3           ;SET MINIMUM LIMIT
3148  013204  004767  010454                JSR     PC,TTR          ;GO GET DENSITY
3149  013210  005767  165430                TST     TEMP1           ;SEE IF HAVE NEW PARAMETER
3150  013214  001407                        BEQ     TINP2           ;IF NOT: BR
3151  013216  042767  003400  165326        BIC     #3400,UDES      ;ELSE CLEAR OLD PARAMETER
3152  013224  012703  000010                MOV     #10,R3          ;SET POSITION FACTOR
3153  013230  004767  000714                JSR     PC,TPOS         ;GO LOAD DENSITY INTO PROPER POSITION
3154  013234  012704  025575          TINP2:  MOV     #MSG34,R4
3155  013240  004767  010656                JSR     PC,TTOUT        ;PRINT PARITY REQUEST
3156  013244  005067  165376                CLR     TEMP2           ;CLR BUFFER
3157  013250  012701  000001                MOV     #1,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3158  013254  012702  000001                MOV     #1,R2           ;SET MAXIMUM LIMIT
3159  013260  012703  000000                MOV     #0,R3           ;SET MINIMUM LIMIT
```

```
3160  013264  004767  010374          JSR     PC,TTR          ;GO INPUT PARITY
3161  013270  005767  165350          TST     TEMP1           ;SEE IF HAVE NEW PARAMETER
3162  013274  001407                  BEQ     TINP2A          ;IF NOT: BR
3163  013276  042767  000010  165246  BIC     #10,UDES        ;ELSE CLEAR OLD PARAMETER
3164  013304  012703  000003          MOV     #3,R3           ;SET POSITION FACTOR
3165  013310  004767  000634          JSR     PC,TPOS         ;GO LOAD PARITY TO PROPER POSITION
3166  013314  012704  026175  TINP2A: MOV     #MSG53,R4
3167  013320  004767  010576          JSR     PC,TTOUT        ;REQUEST FORMAT
3168  013324  005067  165316          CLR     TEMP2
3169  013330  012701  000002          MOV     #2,R1
3170  013334  012702  000016          MOV     #16,R2
3171  013340  012703  000014          MOV     #14,R3
3172  013344  004767  010314          JSR     PC,TTR          ;GO GET FORMAT
3173  013350  005767  165270          TST     TEMP1           ;SEE IF NEW PARAMETER
3174  013354  001407                  BEQ     TINP2B          ;IF NOT: BR
3175  013356  042767  000170  165166  BIC     #170,UDES
3176  013364  012703  000004          MOV     #4,R3
3177  013370  004767  000554          JSR     PC,TPOS
3178  013374  005267  171402  TINP2B: INC     REOTC           ;BUMP EOT UNIT COUNTER
3179  013400  022767  000016  165270  CMP     #16,UNP         ;SEE IF DONE UNITS
3180  013406  001405                  BEQ     TINP2C          ;IF SO: BR
3181  013410  062767  000002  165260  ADD     #2,UNP          ;POINT TO NEXT UNIT
3182  013416  000167  177300          JMP     TINP0           ;ELSE LOOK FOR NEXT UNIT
3183  013422  005067  165250  TINP2C: CLR     UNP             ;CLEAR UNIT POINTER
3184  013426  016700  171350          MOV     REOTC,R0
3185  013432  000367  171344          SWAB    REOTC
3186  013436  110067  171340          MOVB    R0,REOTC        ;SET UNIT EOT COUNTER
3187  013442  012704  025610  TINP3:  MOV     #MSG35,R4
3188  013446  004767  010450          JSR     PC,TTOUT        ;PRINT RECORD COUNT REQUEST
3189  013452  016703  165076          MOV     RCNT,R3
3190  013456  004767  010612          JSR     PC,OCTP         ;PRINT RECORD COUNT
3191  013462  012705  000554          MOV     #RCNT,R5        ;SET RECORD COUNT ADDRESS
3192  013466  012701  000006          MOV     #6,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3193  013472  012702  177777          MOV     #-1,R2          ;SET MAXIMUM LIMIT
3194  013476  012703  000001          MOV     #1,R3           ;SET MINIMUM LIMIT
3195  013502  004767  010156          JSR     PC,TTR          ;GO GET RECORD COUNT
3196  013506  016767  165042  165114  MOV     RCNT,RCSAV      ;SAVE RECORD COUNT
3197  013514  012704  025631          MOV     #MSG36,R4
3198  013520  004767  010376          JSR     PC,TTOUT        ;PRINT CHARACTER COUNT REQUEST
3199  013524  005467  165026          NEG     FMCNT
3200  013530  016703  165022          MOV     FMCNT,R3
3201  013534  004767  010534          JSR     PC,OCTP         ;PRINT CHAR COUNT
3202  013540  012705  000556          MOV     #FMCNT,R5       ;SET CHARACTER COUNT ADDRESS
3203  013544  012701  000006          MOV     #6,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3204  013550  012702  004000          MOV     #4000,R2        ;SET MAXIMUM LIMIT
3205  013554  012703  000004          MOV     #4,R3           ;SET MINIMUM LIMIT
3206  013560  004767  010100          JSR     PC,TTR          ;GO GET CHARACTER COUNT
3207  013564  005467  164766          NEG     FMCNT           ;SET TO TWO'S COMPLIMENT
3208  013570  016767  164762  165034  MOV     FMCNT,FCSAV     ;SAVE FRAME COUNT
3209  013576  012704  025655          MOV     #MSG37,R4       ;PRINT PATTERN NUMBER REQUEST
3210  013602  004767  010314          JSR     PC,TTOUT
3211  013606  016703  164746          MOV     PATRN,R3
3212  013612  004767  010456          JSR     PC,OCTP         ;PRINT PATTERN
3213  013616  005067  000742          CLR     DOFL            ;CLEAR EXTERNAL DATA FLAG
3214  013622  012705  000560          MOV     #PATRN,R5       ;SET PATTERN NUMBER ADDRESS
3215  013626  012701  000002          MOV     #2,R1           ;SET NUMBER OF CHARACTERS TO INPUT
```

```
3216  013632  012702  000015              MOV     #15,R2          ;SET MAXIMUM LIMIT
3217  013636  012703  000000              MOV     #0,R3           ;SET MINIMUM LIMIT
3218  013642  004767  010016              JSR     PC,TTR          ;GO GET PATTERN NUMBER
3219  013646  012704  026372              MOV     #MSG69,R4
3220  013652  004767  010244              JSR     PC,TTOUT        ;REQUEST TM
3221  013656  016703  164702              MOV     TMEX,R3
3222  013662  004767  010406              JSR     PC,OCTP         ;PRINT CURRENT TM FLAG SETTING
3223  013666  012705  000564              MOV     #TMEX,R5        ;GET TM FLAG ADDRESS
3224  013672  012701  000001              MOV     #1,R1           ;SET SIZE OF RESPONSE
3225  013676  012702  000001              MOV     #1,R2           ;SET UPPER LIMIT
3226  013702  012703  000000              MOV     #0,R3           ;SET LOWER LIMIT
3227  013706  004767  007752              JSR     PC,TTR          ;TM 1=YES
3228  013712  012704  025700              MOV     #MSG38,R4
3229  013716  004767  010200              JSR     PC,TTOUT        ;REQUEST SINGLE PASS
3230  013722  016703  164642              MOV     SPFLG,R3
3231  013726  004767  010342              JSR     PC,OCTP         ;PRINT CURRENT SETTING
3232  013732  012705  000570              MOV     #SPFLG,R5       ;SET ADDRESS OF FLAG
3233  013736  012701  000001              MOV     #1,R1           ;SET SIZE OF RESPONSE
3234  013742  012702  000001              MOV     #1,R2           ;SET UPPER LIMIT
3235  013746  012703  000000              MOV     #0,R3           ;SET LOWER LIMIT
3236  013752  004767  007706              JSR     PC,TTR          ;GO GET RESPONSE
3237  013756  005767  000164      TINP4:  TST     SCVFL           ;SEE IF SHORT CONVERSATION
3238  013762  001066                      BNE     TINPX           ;IF SO: BR
3239  013764  012704  025720              MOV     #MSG40,R4
3240  013770  004767  010126              JSR     PC,TTOUT        ;PRINT READ STALL REQUEST
3241  013774  016703  164572              MOV     RSTAL,R3
3242  014000  004767  010270              JSR     PC,OCTP         ;PRINT READ STALL
3243  014004  012705  000572              MOV     #RSTAL,R5       ;SET READ STALL ADDRESS
3244  014010  012701  000006              MOV     #6,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3245  014014  012702  177777              MOV     #-1,R2          ;SET MAXIMUM LIMIT
3246  014020  012703  000001              MOV     #1,R3           ;SET MINIMUM LIMIT
3247  014024  004767  007634              JSR     PC,TTR          ,GO GET READ STALL
3248  014030  012704  025747              MOV     #MSG41,R4
3249  014034  004767  010062              JSR     PC,TTOUT        ;PRINT WRITE STALL REQUEST
3250  014040  016703  164530              MOV     WSTAL,R3
3251  014044  004767  010224              JSR     PC,OCTP         ;PRINT READ STALL
3252  014050  012705  000574              MOV     #WSTAL,R5       ;SET WRITE STALL ADDRESS
3253  014054  012701  000006              MOV     #6,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3254  014060  012702  177777              MOV     #-1,R2          ;SET MAXIMUM LIMIT
3255  014064  012703  000001              MOV     #1,R3           ;SET MINIMUM LIMIT
3256  014070  004767  007570              JSR     PC,TTR          ;GO GET WRITE STALL
3257  014074  012704  025761              MOV     #MSG42,R4
3258  014100  004767  010016              JSR     PC,TTOUT        ;PRINT TURN AROUND STALL REQUEST
3259  014104  016703  164466              MOV     TSTAL,R3
3260  014110  004767  010160              JSR     PC,OCTP         ;PRINT TA STALL
3261  014114  012705  000576              MOV     #TSTAL,R5       ;SET TURN AROUND STALL ADDRESS
3262  014120  012701  000006              MOV     #6,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3263  014124  012702  177777              MOV     #-1,R2          ;SET MAXIMUM LIMIT
3264  014130  012703  000001              MOV     #1,R3           ;SET MINIMUM LIMIT
3265  014134  004767  007524              JSR     PC,TTR          ;GO GET TURN AROUND STALL
3266  014140  005067  000002      TINPX:  CLR     SCVFL           ;CLEAR SHORT CONVERSATION FLAG
3267  014144  000207                      RTS     PC              ;EXIT
3268  014146  000000              SCVFL:  0                       ;SHORT CONVERSATION FLAG
3269
3270                                      ;UNIT DESCRIPTION POSITIONING SUBROUTINE**********
3271
```

```
3272 014150 000241                  TPOS:   CLC
3273 014152 006167 164470                   ROL     TEMP2           ;POSITION CHARACTER
3274 014156 005303                          DEC     R3              ;SEE IF DONE
3275 014160 001373                          BNE     TPOS            ;IF NOT: BR
3276 014162 016700 164510           TPOS1:  MOV     UNP,R0          ;LOAD UNIT POINTER
3277 014166 056760 164454 000746            BIS     TEMP2,UN1(R0)   ;LOAD CHARACTER INTO UN1(R0)
3278 014174 000207                          RTS     PC              ;EXIT
3279
```

```
                                        ;*******************************************************
                                        ;DATA SETUP ROUTINE:
                                        ;
                                        ;THIS ROUTINE IS USED TO GENERATE INTO THE ENTIRE
                                        ;WRITE BUFFER (4000 OCTAL CHARACTERS) THE DATA PATTERN
                                        ;SELECTED BY THE OPERATOR.  THERE ARE 15 (8) FIXED
                                        ;DATA PATTERNS AVAILABLE AND ONE SELECTION (DATA PATTERN 0)
                                        ;WHICH WILL READ ANY PATTERN PRESENTED AT THE
                                        ;HIGH SPEED PAPER TAPE READER.  THIS TAPE MUST BE PREPARED
                                        ;BY USING THE PROGRAM CALLED DTC. (ZZ-CZTUNA0)
                                        ;RANDOM DATA MAY ALSO BE USED VIA CONSOLE
                                        ;SWITCH EIGHT (8).
                                        ;THIS ROUTINE IS ALSO USED TO CLEAR OUT THE
                                        ;READ BUFFER (4000 OCTAL CHARACTERS) BEFORE EACH
                                        ;RECORD IS READ.
                                        ;*******************************************************

014176  005767  001006        DSUP:   TST     RDFL            ;SEE IF DID RANDOM DATA
014202  001401                        BEQ     DS0             ;IF NOT: BR
014204  000445                        BR      DS1             ;ELSE EXIT
014206  005767  164524        DS0:    TST     ASEQF           ;SEE IF AUTO SEQ
014212  001406                        BEQ     DS0C            ;IF NOT: BR
014214  005767  164340                TST     PATRN           ;SEE IF AUTO RANDOM
014220  100003                        BPL     DS0C            ;IF NOT: BR
014222  004767  000714                JSR     PC,DATR         ;ELSE GO GENERATE RANDOM DATA
014226  000207                        RTS     PC              ;RETURN
014230  026767  164324  000152  DS0C:  CMP     PATRN,PATS      ;SEE IF NEW PATTERN
014236  001014                        BNE     DS0A            ;IF SO: BR
014240  016703  164306                MOV     UDES,R3         ;GET UNIT DESCRIPTION
014244  042703  177767                BIC     #177767,R3      ;MASK EVEN PARITY
014250  026703  000136                CMP     PARS,R3         ;SEE IF SAME AS LAST TIME
014254  001404                        BEQ     DS0B            ;IF SO: BR
014256  010367  000130                MOV     R3,PARS         ;SAVE PARITY
014262  004767  000724                JSR     PC,CRCLRC       ;GO GENERATE EXPT CRC/LRC
014266  000207                DS0B:   RTS     PC
014270  012703  027426        DS0A:   MOV     #WDATA,R3       ;R3 = ADDRS OF WRITE BUFFER
014274  016701  164260                MOV     PATRN,R1        ;R1 = PATTERN SELECTOR
014300  010167  000104                MOV     R1,PATS
014304  062701  000001.                ADD     #1,R1           ;BUMP POINTER
014310  000241                        CLC
014312  006101                        ROL     R1              ;MAKE PATTERN SELECTOR EVEN
014314  000171  002770                JMP     @DATBL(R1)      ;GO GENERATE PATTERN
014320  032777  010000  164210  DS1:  BIT     #10000,@DT      ;SEE IF 7 CH
014326  001410                        BEQ     DS2A            ;IF NOT: BR
014330  012702  002002                MOV     #2002,R2        ;SET BUFFER SIZE
014334  012701  027426                MOV     #WDATA,R1       ;SET START OF BUFFER
014340  042721  140300        DS2:    BIC     #140300,(R1)+   ;MASK FOR 7 CH
014344  005302                        DEC     R2              ;SEE IF DONE
014346  001374                        BNE     DS2             ;IF NOT: BR
014350  004767  000636        DS2A:   JSR     PC,CRCLRC       ;GO GENERATE EXPT CRC/LRC
014354  012702  002000        DS3:    MOV     #2000,R2        ;R2=BUFFER SIZE
014360  012701  033434                MOV     #RDATA,R1       ;R1=READ DATA START
014364  005021                DS4:    CLR     (R1)+           ;CLEAR BUFFER
014366  005302                        DEC     R2              ;SEE IF DONE ALL
014370  001375                        BNE     DS4             ;IF NOT: BR
```

```
 3336  014372  016767  164154  000012          MOV     UDES,PARS       ;GET UNIT DESCRIPTION
 3337  014400  042767  177767  000004          BIC     #177767,PARS    ;MASK PARITY
 3338  014406  000207                          RTS     PC              ;EXIT
 3339  014410  177777                  PATS:   -1                      ;PATTERN NUMBER SAVE
 3340  014412  000000                  PARS:   0
 3341
 3342                                          ;EXTERNAL DATA INPUT FROM H/S READER (256 CHARACTER MAXIMUM)
 3343
 3344  014414  005767  000144          DAT0:   TST     DOFL            ;SEE IF SHOULD DO EXTERNAL INPUT
 3345  014420  001337                          BNE     DS1             ;IF NOT: BR
 3346  014422  012767  000001  000134          MOV     #1,DOFL         ;SET EXTERNAL FLAG
 3347  014430  005077  164166                  CLR     @PRB            ;CLEAR READER BUFFER
 3348  014434  005077  164160                  CLR     @PRS            ;CLEAR READER STATUS
 3349  014440  005067  164200                  CLR     TEMP1           ;CLEAR FOR USE AS CHARACTER FLAG
 3350  014444  052777  000001  164146  DAT0A:  BIS     #1,@PRS         ;START READER
 3351  014452  032777  000200  164140  DAT0B:  BIT     #200,@PRS       ;SEE IF DONE
 3352  014460  001774                          BEQ     DAT0B           ;IF NOT : BR
 3353  014462  005001                          CLR     R1              ;CLEAR SAVE LOCATION
 3354  014464  117701  164132                  MOVB    @PRB,R1         ;SAVE CHARACTER
 3355  014470  005767  164150                  TST     TEMP1           ;SEE IF HAVE FOUND START CHARACTER
 3356  014474  001012                          BNE     DAT0C           ;IF SO : BR
 3357  014476  105701                          TSTB    R1              ;SEE IF CHARACTER IS 0
 3358  014500  001761                          BEQ     DAT0A           ;IF SO : BR
 3359  014502  012767  000001  164134          MOV     #1,TEMP1        ;ELSE SET CHARACTER FOUND FLAG
 3360  014510  010167  164132                  MOV     R1,TEMP2        ;SAVE DATA SIZE
 3361  014514  010102                          MOV     R1,R2           ;SAVE DATA SIZE
 3362  014516  000167  177722                  JMP     DAT0A           ;GO GET FIRST DATA CHAR
 3363  014522  110123                  DAT0C:  MOVB    R1,(R3)+        ;LOAD BUFFER
 3364  014524  005302                          DEC     R2              ;SEE IF READ ALL
 3365  014526  001346                          BNE     DAT0A           ;IF NOT : BR
 3366  014530  012701  027426          DAT0D:  MOV     #WDATA,R1       ;R1 = START OF WRITE BUFFER
 3367  014534  016702  164106                  MOV     TEMP2,R2        ;R2 = SIZE OF DATA FIELD
 3368  014540  112123                  DAT0E:  MOVB    (R1)+,(R3)+     ;REPEAT LOAD OF DATA FIELD
 3369  014542  022703  033434                  CMP     #RDATA,R3       ;SEE IF DONE
 3370  014546  003002                          BGT     DAT0F           ;IF NOT: BR
 3371  014550  000167  177544                  JMP     DS1             ;EXIT
 3372  014554  005302                  DAT0F:  DEC     R2              ;SEE IF AT END OF DATA FIELD
 3373  014556  001370                          BNE     DAT0E           ;IF NOT : BR
 3374  014560  000167  177744                  JMP     DAT0D           ;ELSE RESTART FILL
 3375  014564  000000                  DOFL:   0                       ;EXTERNAL DATA FLAG=1 IF ALREADY DONE
 3376
```

```
3377                                        ;ALL ONES**********
3378
3379  014566  012701  177777    DAT1:   MOV     #-1,R1          ;R1=DATA
3380  014572  012702  002002    DAT1A:  MOV     #2002,R2        ;R2=WORD COUNT +2
3381  014576  010123            DAT1B:  MOV     R1,(R3)+        ;LOAD BUFFER
3382  014600  005302                    DEC     R2              ;SEE IF DONE
3383  014602  001375                    BNE     DAT1B           ;IF NOT: BR
3384  014604  000167  177510            JMP     DS1             ;RETURN
3385
3386                                        ;ALL ZEROS**********
3387
3388  014610  005001    DAT2:   CLR     R1              ;R1=DATA
3389  014612  000167  177754            JMP     DAT1A           ;LOAD BUFFER
3390
3391                                        ;WALKING ONE**********
3392
3393  014616  012701  000001    DAT3:   MOV     #1,R1           ;R1=DATA
3394  014622  000241            CLC
3395  014624  012702  004004    DAT3A:  MOV     #4004,R2        ;R2=CHARACTER COUNT+4
3396  014630  110123            DAT3B:  MOVB    R1,(R3)+        ;LOAD BUFFER
3397  014632  106101                    ROLB    R1              ;SET NEXT CHARACTER
3398  014634  005302                    DEC     R2              ;SEE IF DONE
3399  014636  001374                    BNE     DAT3B           ;IF NOT: BR
3400  014640  000167  177454            JMP     DS1             ;RETURN
3401
3402                                        ;WALKING ZERO**********
3403
3404  014644  012701  000376    DAT4:   MOV     #376,R1         ;R1=START OF DATA
3405  014650  000261            SEC
3406  014652  000167  177746            JMP     DAT3A           ;LOAD BUFFER
3407
3408                                        ,ALTERNATING ONE/ZERO**********
3409
3410
3411  014656  012701  052525    DAT5:   MOV     #52525,R1       ;R1=DATA
3412  014662  000167  177704            JMP     DAT1A           ;LOAD BUFFER
3413
3414                                        ;ALTERNATING ZERO/ONE**********
3415
3416  014666  012701  125252    DAT6:   MOV     #125252,R1      ;R1=DATA
3417  014672  000167  177674            JMP     DAT1A           ;LOAD BUFFER
3418
3419                                        ;ONE/ZERO IN ALTERNATING WORDS**********
3420
3421  014676  012701  125252    DAT7:   MOV     #125252,R1      ;SET WORD 1
3422  014702  012702  052525            MOV     #52525,R2       ;SET WORD 2
3423  014706  012704  001002            MOV     #1002,R4        ;SET NUMBER OF ENTRIES
3424  014712  010123            DAT7A:  MOV     R1,(R3)+        ;LOAD WORD 1
3425  014714  010223                    MOV     R2,(R3)+        ;LOAD WORD 2
3426  014716  005304                    DEC     R4              ;SEE IF DONE
3427  014720  001374                    BNE     DAT7A           ;IF NOT: BR
3428  014722  000167  177372            JMP     DS1             ;RETURN
3429
```

```
3430                                    ;WALKING ONE/ALL ONE IN ALTERNATING CHARS****
3431
3432  014726  012702  004002    DAT10:  MOV     #4002,R2        ;SET BUFFER SIZE
3433  014732  012701  000001            MOV     #1,R1           ;SET WALK BASE
3434  014736  000241                    CLC
3435  014740  012713  177400    DAT10A: MOV     #177400,(R3)    ;LOAD ALL ONE BYTE
3436  014744  050123                    BIS     R1,(R3)+        ;LOAD WALK BYTE
3437  014746  106101                    ROLB    R1              ;WALK ONE
3438  014750  005302                    DEC     R2
3439  014752  001372                    BNE     DAT10A          ;DO FULL BUFFER
3440  014754  000167  177340            JMP     DS1             ;RETURN
3441
3442                                    ;ALL BITS 0-377**********
3443
3444  014760  005001    DAT11:  CLR     R1              ;R1=STARTING DATA
3445  014762  012702  004004            MOV     #4004,R2        ;R2=CHARACTER COUNT+4
3446  014766  110123    DAT11A: MOVB    R1,(R3)+        ;LOAD BUFFER
3447  014770  105201            INCB    R1              ;BUMP DATA
3448  014772  005302            DEC     R2              ;SEE IF DONE
3449  014774  001374            BNE     DAT11A          ;IF NOT: BR
3450  014776  000167  177316            JMP     DS1             ;RETURN
3451
3452                                    ;ALL BITS 377-0**********
3453
3454  015002  012701  000377    DAT12:  MOV     #377,R1         ;R1=STARTING DATA
3455  015006  012702  004004            MOV     #4004,R2        ;R2=CHARACTER COUNT+4
3456  015012  110123    DAT12A: MOVB    R1,(R3)+        ;LOAD BUFFER
3457  015014  105301            DECB    R1              ;BUMP DATA
3458  015016  005302            DEC     R2              ;SEE IF DONE
3459  015020  001374            BNE     DAT12A          ;IF NOT: BR
3460  015022  000167  177272            JMP     DS1             ;RETURN
3461
3462                                    ;ALTERNATING CHARACTERS 0 AND 377**********
3463
3464  015026  012701  000377    DAT13:  MOV     #377,R1         ;R1 = DATA
3465  015032  000167  177534            JMP     DAT1A           ;LOAD BUFFER
3466
3467                                    ;WALKING ZERO/ALL ZERO IN ALTERNATING CHARS******
3468
3469  015036  012702  004002    DAT14:  MOV     #4002,R2        ;SET BUFFER SIZE
3470  015042  012701  000376            MOV     #376,R1         ;SET WALK BASE
3471  015046  000261                    SEC
3472  015050  010113    DAT14A: MOV     R1,(R3)         ;LOAD WALK BYTE
3473  015052  042723  177400            BIC     #177400,(R3)+   ;CLEAR HIGH BYTE
3474  015056  106101                    ROLB    R1              ;WALK ZERO BIT
3475  015060  005302                    DEC     R2
3476  015062  001372                    BNE     DAT14A          ;FILL BUFFER
3477  015064  000167  177230            JMP     DS1             ;RETURN
3478
```

```
 3479                                           ;AUTO SEQUENCE PATTERN*********
 3480
 3481  015070  012702  000400          DAT15:  MOV     #400,R2         ;SET NUMBER OF ENTRIES
 3482  015074  012701  015122          DAT15A: MOV     #APATS,R1       ;SET START OF PATTERN
 3483  015100  012704  000010                  MOV     #10,R4          ;SET SIZE OF PATTERN
 3484  015104  012123                  DAT15B: MOV     (R1)+,(R3)+     ;FILL BUFFER
 3485  015106  005304                          DEC     R4              ;SEE IF DONE PATTERN
 3486  015110  001375                          BNE     DAT15B          ;IF NOT: BR
 3487  015112  005302                          DEC     R2              ;SEE IF DONE BUFER
 3488  015114  001367                          BNE     DAT15A          ;IF NOT: BR
 3489  015116  000167  177176                  JMP     DS1             ;RETURN
 3490  015122  000000                  APATS:  0
 3491  015124  177400                          177400
 3492  015126  000377                          377
 3493  015130  000000                          0
 3494  015132  177777                          -1
 3495  015134  000377                          377
 3496  015136  177400                          177400
 3497  015140  177777                          -1
 3498
 3499                                           ;RANDOM DATA GENERATOR SUBROUTINE**********
 3500
 3501  015142  016704  163410          DATR:   MOV     FMCNT,R4        ;SET NUMBER OF FRAMES
 3502  015146  012703  027426                  MOV     #WDATA,R3       ;SET ADDRESS OF START OF BUFFER
 3503  015152  012701  177777                  MOV     #-1,R1          ;SET HIGH LIMIT
 3504  015156  005002                          CLR     R2              ;SET LOW LIMIT
 3505  015160  004767  006446          DATR0:  JSR     PC,RANG         ;GO GENERATE NUMBER
 3506  015164  016723  163436                  MOV     RANSAV,(R3)+    ;LOAD BUFFER
 3507  015170  005204                          INC     R4              ;SEE IF DONE WHOLE BUFFER
 3508  015172  001372                          BNE     DATR0           ;IF NOT: BR
 3509  015174  004767  177120                  JSR     PC,DS1          ;GO CHECK FOR 7 CH
 3510  015200  012767  000001  000002          MOV     #1,RDFL         ;SET RANDOM DATA FLAG
 3511  015206  000207                          RTS     PC              ;EXIT
 3512  015210  000000                  RDFL:   0                       ;RANDOM DATA SELECT FLAG
```

```
3513
3514                                    ;*************************************************************
3515                                    ;CRC/LRC CHARACTER BUILD;
3516                                    ;
3517                                    ;THIS ROUTINE WILL CONSTRUCT AND SAVE THE EXPECTED
3518                                    ;CRC AND LRC CHARACTERS ACCORDING TO DATA AND
3519                                    ;RECORD SIZE IF OPERATING IN NRZ MODE
3520                                    ;*************************************************************
3521
3522  015212  016700  163340    CRCLRC: MOV     FMCNT,R0        ;SET RECORD SIZE
3523  015216  005400                    NEG     R0
3524  015220  012701  027426            MOV     #WDATA,R1       ;SET START OF BUFFER
3525  015224  005067  000346            CLR     XORS
3526  015230  111104            CL0:    MOVB    (R1),R4         ;GET CHARACTER
3527  015232  004767  000166            JSR     PC,CLP          ;GO GET PARITY OF CHARACTER
3528  015236  004767  000310            JSR     PC,XOR          ;XOR CHARACTER
3529  015242  000241                    CLC
3530  015244  006004                    ROR     R4              ;ROTATE 1 RIGHT
3531  015246  103014                    BCC     CL2             ;IF NO CARRY: BR
3532  015250  052704  000400            BIS     #400,R4         ;SET BIT NINE
3533  015254  000241                    CLC
3534  015256  010405            CL1:    MOV     R4,R5           ;SAVE CHARACTER
3535  015260  042705  177703            BIC     #177703,R5
3536  015264  005105                    COM     R5
3537  015266  042705  177703            BIC     #177703,R5
3538  015272  042704  000074            BIC     #74,R4
3539  015276  050504                    BIS     R5,R4           ;COMPLIMENT BITS 2,3,4,5
3540  015300  010467  000272    CL2:    MOV     R4,XORS
3541  015304  005300                    DEC     R0
3542  015306  001402                    BEQ     CLLAST          ;IF LAST CHARACTER: BR
3543  015310  000167  177714            JMP     CL0             ;GET NEXT
3544  015314  016704  000256    CLLAST: MOV     XORS,R4
3545  015320  005167  000252            COM     XORS
3546  015324  042767  177050  000244    BIC     #177050,XORS
3547  015332  042704  177727            BIC     #177727,R4      ;COMPLIMENT ALL BUT BITS 3&5
3548  015336  050467  000234            BIS     R4,XORS
3549  015342  016767  000230  000230    MOV     XORS,EXCRC      ;SAVE EXPECTED CRC
3550  015350  016700  163202            MOV     FMCNT,R0
3551  015354  005400                    NEG     R0
3552  015356  012701  027426            MOV     #WDATA,R1       ;DO EXPT LRC
3553  015362  005067  000210            CLR     XORS
3554  015366  111104            CL3:    MOVB    (R1),R4
3555  015370  004767  000030            JSR     PC,CLP          ;GET PARITY
3556  015374  004767  000152            JSR     PC,XOR          ;XOR CHARACTER
3557  015400  005300                    DEC     R0
3558  015402  001371                    BNE     CL3             ;DO ALL FOR LRC
3559  015404  016704  000170            MOV     EXCRC,R4
3560  015410  004767  000136            JSR     PC,XOR          ;XOR CRC TO DATA
3561  015414  016767  000156  000160    MOV     XORS,EXLRC      ;SAVE EXPT LRC
3562  015422  000207                    RTS     PC              ;RETURN
3563  015424  005704            CLP:    TST     R4              ;SEE IF 0 CHAR
3564  015426  001010                    BNE     CLPE            ;IF NOT: BR
3565  015430  032767  000010  163114    BIT     #10,UDES        ;SEE IF EVEN PARITY
3566  015436  001404                    BEQ     CLPE            ;IF NOT: BR
3567  015440  012704  000420            MOV     #420,R4         ;SET 0 CHAR EVEN PARITY
3568  015444  005201                    INC     R1              ;BUMP POINTER
```

```
3569  015446  000207                          RTS     PC              ;RETURN
3570  015450  005067  000130          CLPE:   CLR     PARCNT          ;CLEAR BIT COUNTER
3571  015454  012703  000010                  MOV     #10,R3          ;SET NUMBER OF BITS
3572  015460  032704  000001          CLP0:   BIT     #1,R4           ;SEE IF ONE BIT
3573  015464  001402                          BEQ     CLP1            ;IF NOT: BR
3574  015466  005267  000112                  INC     PARCNT          ;BUMP COUNTER
3575  015472  000241                  CLP1:   CLC
3576  015474  006004                          ROR     R4              ;ROTATE TO NEXT BIT
3577  015476  005303                          DEC     R3
3578  015500  001367                          BNE     CLP0            ;CONTINUE FOR ALL BITS
3579  015502  112104                          MOVB    (R1)+,R4
3580  015504  042704  177400                  BIC     #177400,R4
3581  015510  032767  000001  000066          BIT     #1,PARCNT       ;SEE IF ODD NUMBER OF ONE BITS
3582  015516  001005                          BNE     CLP2            ;IF SO: BR
3583  015520  032767  000010  163024          BIT     #10,UDES        ;SEE IF SHOULD BE EVEN PARITY
3584  015526  001406                          BEQ     CLP3            ;IF NOT: BR
3585  015530  000207                          RTS     PC              ;ELSE EXIT
3586  015532  032767  000010  163012  CLP2:   BIT     #10,UDES        ;SEE IF SHOULD BE ODD PARITY
3587  015540  001001                          BNE     CLP3            ;IF NOT: BR
3588  015542  000207                          RTS     PC              ;ELSE EXIT
3589  015544  052704  000400          CLP3:   BIS     #400,R4         ;SET PARITY BIT
3590  015550  000207                          RTS     PC
3591  015552  010446                  XOR:    MOV     R4,-(SP)
3592  015554  046716  000016                  BIC     XORS,(SP)
3593  015560  040467  000012                  BIC     R4,XORS         ;XOR SUBROUTINE: R4 WITH XORS
3594  015564  052667  000006                  BIS     (SP)+,XORS
3595  015570  016704  000002                  MOV     XORS,R4
3596  015574  000207                          RTS     PC
3597
3598  015576  000000                  XORS:   0                       ;XOR SAVE
3599  015600  000000                  EXCRC:  0                       ;EXPECTED CRC
3600  015602  000000                  EXLRC:  0                       ;EXPECTED LRC
3601  015604  000000                  PARCNT: 0                       ;PARITY COUNTER
3602
```

```
 3603
 3604                                   ;*********************************************************
 3605                                   ;DATA CHECK SUBROUTINE:
 3606                                   ;
 3607                                   ;THIS SUBROUTINE IS USED TO COMPARE EACH CHARACTER
 3608                                   ;OF DATA READ FROM TAPE WITH THE EXPECTED CHARACTER.
 3609                                   ;ANY ERROR DETECTED WILL CAUSE CONTROL TO BE
 3610                                   ;PASSED TO AN ERROR PRINT SUBROUTINE AND A
 3611                                   ;SUBROUTINE TO ACCUMULATE THE NUMBER OF BITS
 3612                                   ;DROPPED AND PICKED UP FROM EACH CHARACTER.
 3613                                   ;THE NUMBER OF READ ERRORS IS ALSO ACCUMULATED.
 3614                                   ;DATA CHECKING MAY BE TERMINATED BY USE OF
 3615                                   ;CONSOLE SWITCH THIRTEEN (13).
 3616                                   ;*********************************************************
 3617
 3618  015606  005067  163046      DCHK:  CLR     BBC             ;CLEAR BAD RECORD CNTR
 3619  015612  005067  163070             CLR     DERFL           ;CLEAR DATA ERROR FLAG
 3620  015616  016705  162734             MOV     FMCNT,R5        ;LOAD CHAR COUNT
 3621  015622  032767  000020  162722     BIT     #20,UDES        ;SEE IF CORE DUMP
 3622  015630  001402                     BEQ     DCHK0           ;IF NOT: BR
 3623  015632  000261                     SEC
 3624  015634  006005                     ROR     R5              ;R5 = FC/2
 3625  015636  012701  027426      DCHK0: MOV     #WDATA,R1       ;SET WRITE DATA ADDR
 3626  015642  012702  033434             MOV     #RDATA,R2       ;SET READ DATA ADDR
 3627  015646  032767  000010  162676     BIT     #10,UDES        ;SEE IF EVEN PARITY
 3628  015654  001430                     BEQ     DF0C0           ;IF NOT: BR
 3629  015656  032767  000020  162666     BIT     #20,UDES        ;SEE IF CORE DUMP PARITY
 3630  015664  001024                     BNE     DF0C0           ;IF SO: BR
 3631  015666  032767  002000  162656     BIT     #2000,UDES      ;SEE IF PE MODE
 3632  015674  001020                     BNE     DF0C0           ;IF SO: BR
 3633  015676  105711              DF0F:  TSTB    (R1)            ;SEE IF 0 CHAR
 3634  015700  001404                     BEQ     DF0D            ;IF SO: BR
 3635  015702  005201                     INC     R1              ;BUMP POINTER
 3636  015704  005205              DF0E:  INC     R5              ;SEE IF DONE
 3637  015706  001373                     BNE     DF0F            ;IF NOT: BR
 3638  015710  000406                     BR      DF0C            ;ELSE CONTINUE
 3639  015712  112721  000020      DF0D:  MOVB    #20,(R1)+       ;SET 20 IN PLACE OF 0
 3640  015716  012767  177777  176464     MOV     #-1,PATS        ;SET PATTERN GENERATE FLAG
 3641  015724  000767                     BR      DF0E
 3642  015726  016705  162624      DF0C:  MOV     FMCNT,R5        ;RESET CHAR CNT
 3643  015732  012701  027426             MOV     #WDATA,R1       ;RESET DATA ADDRESS
 3644  015736  032767  010000  162616 DF0C0: BIT   #10000,RDCMD   ;SEE IF READ REVERSE
 3645  015744  001462                     BEQ     DF0             ;IF NOT: BR
 3646  015746  016704  162604      DF0B:  MOV     FMCNT,R4        ;GET FRAME COUNT
 3647  015752  005404                     NEG     R4              ;SET TO WHOLE NUMBER
 3648  015754  032767  000020  162570     BIT     #20,UDES        ;SEE IF CORE DUMP
 3649  015762  001402                     BEQ     DF0B0           ;IF NOT: BR
 3650  015764  000241                     CLC
 3651  015766  006004                     ROR     R4              ;SET TO FC/2
 3652  015770  060401              DF0B0: ADD     R4,R1           ;POINT TO START OF WRITE DATA
 3653  015772  060402                     ADD     R4,R2           ;POINT TO START OF READ DATA
 3654  015774  032767  000001  162554     BIT     #1,FMCNT        ;SEE IF ODD FRAME COUNT
 3655  016002  001401                     BEQ     DF0A            ;IF NOT: BR
 3656  016004  105722                     TSTB    (R2)+           ;BUMP POINTER
 3657  016006  032767  000020  162536 DF0A: BIT     #20,UDES        ;SEE IF CORE DUMP
 3658  016014  001431                     BEQ     DF0A4           ;IF NOT: BR
```

```
 3659  016016  000241                        CLC
 3660  016020  132742  000001                BITB    #1,-(R2)        ;SEE IF BIT 0 = 1
 3661  016024  001401                        BEQ     DF0A0           ;IF NOT: BR
 3662  016026  000261                        SEC
 3663  016030  106012                DF0A0:  RORB    (R2)
 3664  016032  000241                        CLC
 3665  016034  132712  000001                BITB    #1,(R2)
 3666  016040  001401                        BEQ     DF0A1
 3667  016042  000261                        SEC
 3668  016044  106012                DF0A1:  RORB    (R2)            ;POSITION BITS FOR REVERSE CORE DUMP
 3669  016046  000241                        CLC
 3670  016050  132712  000001                BITB    #1,(R2)
 3671  016054  001401                        BEQ     DF0A2
 3672  016056  000261                        SEC
 3673  016060  106012                DF0A2:  RORB    (R2)
 3674  016062  000241                        CLC
 3675  016064  132712  000001                BITB    #1,(R2)
 3676  016070  001401                        BEQ     DF0A3
 3677  016072  000261                        SEC
 3678  016074  106012                DF0A3:  RORB    (R2)
 3679  016076  005202                        INC     R2              ;RESET POINTER
 3680  016100  124142                DF0A4:  CMPB    -(R1),-(R2)     ;TEST DATA CHARACTER
 3681  016102  001010                        BNE     DF1             ;IF NOT GOOD: BR
 3682  016104  105067  162550                CLRB    BBC             ;CLEAR BAD RECORD COUNTER
 3683  016110  000411                        BR      DF2
 3684  016112  122122                DF0:    CMPB    (R1)+,(R2)+     ;CHECK DATA
 3685  016114  001003                        BNE     DF1             ;IF BAD: BR
 3686  016116  105067  162536                CLRB    BBC             ;CLEAR BAD RECORD CNTR
 3687  016122  000404                        BR      DF2
 3688  016124  004767  000632        DF1:    JSR     PC,DRPKF        ;GO GET DROPS AND PICKS
 3689  016130  004767  000066                JSR     PC,DERR         ;GO DO PRINT
 3690  016134  005205                DF2:    INC     R5              ;BUMP CHAR CNTR
 3691  016136  001405                        BEQ     DF3             ;IF DONE ALL: BR
 3692  016140  032767  010000  162414        BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 3693  016146  001761                        BEQ     DF0             ;IF NOT: BR
 3694  016150  000716                        BR      DF0A            ;ELSE CONTINUE READ REV
 3695  016152  005067  162510        DF3:    CLR     HDRFL           ;CLEAR HEADER FLAG
 3696  016156  005767  162524                TST     DERFL           ;SEE IF HAD DATA ERROR
 3697  016162  001416                        BEQ     DFX             ;IF NOT: BR
 3698  016164  005767  162520                TST     SERFL
 3699  016170  001013                        BNE     DFX             ;IF NOT DATA ERROR ONLY: BR
 3700  016172  016704  162500                MOV     UNP,R4
 3701  016176  032767  010000  162356        BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 3702  016204  001003                        BNE     DF4             ;IF SO: BR
 3703  016206  005264  001130                INC     DATER1(R4)      ;BUMP DATA ERROR FORWARD COUNTER
 3704  016212  000402                        BR      DFX
 3705  016214  005264  001170        DF4:    INC     DEREV1(R4)      ;BUMP REVERSE DATA ERROR
 3706  016220  000207                DFX:    RTS     PC              ;EXIT
 3707
```

```
3708
3709                                    ;*********************************************************
3710                                    ;DATA ERROR SUBROUTINE:
3711                                    ;
3712                                    ;THIS SUBROUTINE IS USED TO PRINT OUT ANY
3713                                    ;ERRORS FOUND DURING THE DATA CHECK.
3714                                    ;EACH CHARACTER FOUND BAD WILL BE PRINTED
3715                                    ;IN BIT FORMAT ALONG WITH ITS EXPECTED CHARACTER.
3716                                    ;AN ERROR HEADER CONSISTING OF THE UNIT NUMBER,
3717                                    ;BLOCK NUMBER, RECORD NUMBER, SIZE OF RECORD, AND
3718                                    ;ERROR TYPE (READ FORWARD, READ REVERSE, WRITE, ETC)
3719                                    ;IS PRINTED ONLY ONCE FOR EACH RECORD FOUND BAD.
3720                                    ;A COUNT IS MADE OF THE NUMBER OF SUCCESSIVE BAD
3721                                    ;CHARACTERS, AND IF TEN (10) SUCCESSIVE BAD CHARACTERS
3722                                    ;ARE FOUND IN A SINGLE RECORD, A MESSAGE INDICATING
3723                                    ;A BAD RECORD CONDITION IS PRINTED AND THE NEXT
3724                                    ;TWENTY (20) CHARACTERS ARE SKIPPED BEFORE CHECKING
3725                                    ;IS RESUMED.  IF THE BAD RECORD CONDITION IS FOUND
3726                                    ;THREE TIMES IN A RECORD, ALL REMAINING DATA IS
3727                                    ;SKIPPED EXCEPT THE FINAL TEN (10) CHARACTERS.
3728                                    ;THIS SKIPPING IS OF COURSE ONLY POSSIBLE IN
3729                                    ;RECORDS WHICH CONTAIN A SUFFICIENT NUMBER OF CHARACTERS.
3730                                    ;PRINTING OF ERRORS MAY BE DISALLOWED AT ANY TIME
3731                                    ;BY SETTING CONSOLE SWITCH TEN (10) TO A ONE.
3732                                    ;THE OPERATOR MAY CAUSE THE PROGRAM TO HALT ON ANY ERROR
3733                                    ;BY SETTING CONSOLE SWITCH FIFTEEN (15) TO A ONE.
3734                                    ;*********************************************************
3735
3736  016222  032777  002000  162356  DERR:   BIT     #2000,@SWR      ;SEE IF SHOULD PRINT ERRORS
3737  016230  001402                          BEQ     DERR0           ;IF SO: BR
3738  016232  000167  000172                  JMP     DERR4           ;ELSE SKIP PRINT
3739  016236  005267  162430          DERR0:  INC     PFLG            ;SET PRINT FLAG
3740  016242  005767  162420                  TST     HDRFL           ;SEE IF HAVE PRINTED HEADER
3741  016246  001010                          BNE     DERR0A          ;IF SO: BR
3742  016250  004767  004724                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
3743  016254  012704  024724                  MOV     #MSG1,R4        ;LOAD ERROR MSG ADDR
3744  016260  004767  005636                  JSR     PC,TTOUT        ;PRINT ERROR
3745  016264  004767  003066                  JSR     PC,FRPRT        ;PRINT F OR R
3746  016270  012704  024743          DERR0A: MOV     #MSG4,R4
3747  016274  004767  005622                  JSR     PC,TTOUT        ;PRINT CHAR NO. HEADER
3748  016300  010203                          MOV     R2,R3
3749  016302  162703  033434                  SUB     #RDATA,R3       ;POINT TO CHAR
3750  016306  005303                          DEC     R3
3751  016310  032767  010000  162244          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
3752  016316  001402                          BEQ     DERR0B          ;IF NOT: BR
3753  016320  010503                          MOV     R5,R3           ;GET CHAR NUMBER
3754  016322  005103                          COM     R3
3755  016324  004767  005744          DERR0B: JSR     PC,OCTP         ;PRINT CHAR NUMBER
3756  016330  012704  024731                  MOV     #MSG2,R4
3757  016334  004767  005562                  JSR     PC,TTOUT        ;PRINT EXPECTED DATA
3758  016340  032767  010000  162214          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
3759  016346  001402                          BEQ     DERR0C          ;IF NOT: BR
3760  016350  111103                          MOVB    (R1),R3         ;GET CHAR
3761  016352  000401                          BR      DERR0D
3762  016354  114103          DERR0C: MOVB    -(R1),R3        ;LOAD EXPECTED DATA
3763  016356  004767  006132          DERR0D: JSR     PC,DOUT         ;GO PRINT CHAR
```

```
 3764  016362  012704  024736                MOV     #MSG3,R4
 3765  016366  004767  005530                JSR     PC,TTOUT        ;PRINT RECIEVED DATA
 3766  016372  032767  010000  162162        BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 3767  016400  001402                        BEQ     DERR1           ;IF NOT: BR
 3768  016402  111203                        MOVB    (R2),R3         ;GET CHAR
 3769  016404  000401                        BR      DERR2
 3770  016406  114203                DERR1:  MOVB    -(R2),R3
 3771  016410  004767  006100        DERR2:  JSR     PC,DOUT         ;PRINT BAD CHAR
 3772  016414  032767  010000  162140        BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 3773  016422  001401                        BEQ     DERR3           ;IF SO: BR
 3774  016424  000401                        BR      DERR4
 3775  016426  122122                DERR3:  CMPB    (R1)+,(R2)+     ;RESET POINTERS
 3776  016430  105267  162224        DERR4:  INCB    BBC             ;BUMP BAD RECORD CNTR
 3777  016434  122767  000010  162216        CMPB    #10,BBC         ;SEE IF BLD BTH
 3778  016442  001124                        BNE     DEREX           ;IF NOT: BR
 3779  016444  032777  002000  162134        BIT     #2000,@SWR      ;SEE IF PRINT INHIBIT
 3780  016452  001004                        BNE     1$              ;IF SO: BR
 3781  016454  012704  025057                MOV     #MSG15,R4
 3782  016460  004767  005436                JSR     PC,TTOUT        ;PRINT BLD BTH
 3783  016464  105067  162170        1$:     CLRB    BBC             ;RESET BAD RECORD CNTR
 3784  016470  000367  162164                SWAB    BBC             ;POSITION BLD BTH AMOUNT
 3785  016474  105267  162160                INCB    BBC             ;BUMP AMOUNT
 3786  016500  122767  000003  162152        CMPB    #3,BBC          ;SEE IF HAD 3 BLD BTHS
 3787  016506  101054                        BHI     DERR4B          ;IF NOT: BR
 3788  016510  000367  162144                SWAB    BBC             ;REPOSITION BBC
 3789  016514  022705  177767                CMP     #177767,R5      ;SEE IF ON LAST EIGHT CHARS
 3790  016520  101473                        BLOS    DERR6           ;IF SO: BR
 3791  016522  012705  177767                MOV     #177767,R5      ;SET CHAR CNTR TO 8
 3792  016526  032767  010000  162026        BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 3793  016534  001416                        BEQ     DERR4A          ;IF NOT: BR
 3794  016536  012701  027426                MOV     #WDATA,R1       ;GET START OF BUFFER
 3795  016542  012702  033434                MOV     #RDATA,R2       ;GET START OF BUFFER
 3796  016546  062701  000010                ADD     #10,R1
 3797  016552  062702  000010                ADD     #10,R2          ;POINT TO START +10
 3798  016556  032767  000001  161772        BIT     #1,FMCNT        ;SEE IF ODD FRAME COUNT
 3799  016564  001453                        BEQ     DEREX           ;IF NOT: BR
 3800  016566  105722                        TSTB    (R2)+           ;BUMP POINTER
 3801  016570  000451                        BR      DEREX
 3802  016572  016767  161760  162044  DERR4A: MOV    FMCNT,TEMP1     ;LOAD CHAR COUNT
 3803  016600  005167  162040                COM     TEMP1
 3804  016604  005267  162034                INC     TEMP1
 3805  016610  162767  000010  162026        SUB     #10,TEMP1       ;POINT TO BUFFER -8
 3306  016616  016701  162022                MOV     TEMP1,R1        ;POINT TO NEXT CHAR
 3807  016622  062701  027426                ADD     #WDATA,R1       ;POINT TO NEXT WRITE CHAR
 3808  016626  016702  162012                MOV     TEMP1,R2        ;POINT TO END OF READ DATA -8 FORWARD
 3809  016632  062702  033434                ADD     #RDATA,R2       ;POINT TO NEXT CHAR
 3810  016636  000426                        BR      DEREX           ;EXIT
 3811  016640  000367  162014        DERR4B: SWAB    BBC             ;REPOSITION BBC
 3812  016644  000241                        CLC
 3813  016646  062705  000024                ADD     #24,R5          ;SKIP 20 CHARS
 3814  016652  103416                        BCS     DERR6           ;IF EXCEED RECORD SIZE: BR
 3815  016654  032767  010000  161700        BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 3816  016662  001405                        BEQ     DERR5           ;IF NOT: BR
 3817  016664  162701  000024                SUB     #24,R1
 3818  016670  162702  000024                SUB     #24,R2          ;RESET POINTERS
 3819  016674  000407                        BR      DEREX
```

```
 3820  016676  062701  000024          DERR5:  ADD     #24,R1          ;SKIP 20 CHARS
 3821  016702  062702  000024                  ADD     #24,R2          ;SKIP FORWARD 20 CHARS
 3822  016706  000402                          BR      DEREX
 3823  016710  012705  177777          DERR6:  MOV     #-1,R5          ;SET TO EOR
 3824  016714  032777  100000  161664  DEREX:  BIT     #100000,@SWR    ;SEE IF SHOULD HALT ON ERROR
 3825  016722  001412                          BEQ     DEREX1          ;IF NOT: BR
 3826  016724  000000                          HALT
 3827  016726  005767  161740                  TST     PFLG            ;SEE IF PRINTED
 3828  016732  001006                          BNE     DEREX1          ;IF SO: BR
 3829  016734  032777  002000  161644          BIT     #2000,@SWR      ;SEE IF SHOULD PRINT
 3830  016742  001002                          BNE     DEREX1          ;IF NOT: BR
 3831  016744  000167  177266                  JMP     DERR0           ;ELSE PRINT
 3832  016750  005067  161716          DEREX1: CLR     PFLG            ;CLEAR FLAG
 3833  016754  005267  161726                  INC     DERFL           ;BUMP DATA ERROR FLAG
 3834  016760  000207                          RTS     PC              ;RETURN
 3835
```

```
3836
3837                                    ;*********************************************************
3838                                    ;DROPS AND PICKS SUBROUTINE:
3839                                    ;
3840                                    ;THIS SUBROUTINE IS USED TO ACCUMULATE FROM
3841                                    ;EACH BAD DATA CHARACTER FOUND THE NUMBER
3842                                    ;OF BITS WHICH WERE EITHER DROPPED OR PICKED UP.
3843                                    ;TWO COUNTERS PER SLAVE ARE USED TO ACCUMULATE THIS
3844                                    ;INFORMATION AND CAN STORE UP TO 32K DROPS
3845                                    ;OR PICKS BEFORE OVERFLOWING.  IF OVERFLOW IS
3846                                    ;ABOUT TO OCCUR, THESE ACCUMULATORS ARE
3847                                    ;PRINTED IN OCTAL AND RESET TO ZERO.
3848                                    ;THE CONTENTS OF THE ACCUMULATORS MAY BE
3849                                    ;DISPLAYED AT ANY TIME BY SETTING CONSOLE
3850                                    ;SWITCH FOURTEEN TO A ONE (1). THE PRINTOUT WILL OCCUR
3851                                    ;AT THE END OF THE CURRENT BLOCK CYCLE.
3852                                    ;*********************************************************
3853
3854  016762  005067  161656           DRPKF:  CLR     TEMP1
3855  016766  005067  161654                   CLR     TEMP2
3856  016772  005067  161652                   CLR     TEMP3
3857  016776  111167  161642                   MOVB    (R1),TEMP1      ;LOAD GOOD CHAR
3858  017002  111267  161640                   MOVB    (R2),TEMP2      ;LOAD BAD CHAR
3859  017006  016704  161664                   MOV     UNP,R4
3860  017012  016467  000770  161702           MOV     PIK1(R4),BPKP
3861  017020  016467  001010  161672           MOV     DRP1(R4),BDPP
3862  017026  032767  010000  161526           BIT     #10000,RDCMD    ;SEE IF READ REVERSE
3863  017034  001005                           BNE     DRPK            ;IF SO: BR
3864  017036  124142                           CMPB    -(R1),-(R2)     ;POINT TO CHAR
3865  017040  112167  161600                   MOVB    (R1)+,TEMP1     ;LOAD GOOD CHAR
3866  017044  112267  161576                   MOVB    (R2)+,TEMP2     ;LOAD BAD CHAR
3867  017050  004767  000006           DRPK:   JSR     PC,DROP         ;GET DROPS
3868  017054  004767  000222                   JSR     PC,PICK         ;GET PICKS
3869  017060  000207                           RTS     PC              ;EXIT
3870  017062  116703  161556           DROP:   MOVB    TEMP1,R3        ;R3 = GOOD CHAR
3871  017066  116704  161554                   MOVB    TEMP2,R4        ;R4 = BAD CHAR
3872  017072  140403                    DPC:    BICB    R4,R3           ;GET DROPS/PICKS
3873  017074  001001                           BNE     DPCG            ;IF SOME: BR
3874  017076  000207                           RTS     PC              ;RETURN
3875  017100  012767  000010  161604   DPCG:   MOV     #10,BCNT        ;SET NUMBER TO CHECK
3876  017106  132703  000001           DPC0:   BITB    #1,R3           ;SEE IF DROPPED OR PICKED THIS BIT
3877  017112  001455                           BEQ     DPC2            ;IF NOT: BR
3878  017114  105767  161530                   TSTB    TEMP3           ;SEE IF ON PICKS
3879  017120  001016                           BNE     DPC1            ;IF SO: BR
3880  017122  005277  161572                   INC     @BDPP           ;BUMP DROP CNTR
3881  017126  005777  161566                   TST     @BDPP
3882  017132  100045                           BPL     DPC2            ;IF NO OVERFLOW: BR
3883  017134  032777  002000  161444           BIT     #2000,@SWR      ;SEE IF HAVE PRINTED DATA
3884  017142  001402                           BEQ     DPC0A           ;IF SO: BR
3885  017144  004767  004030                   JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
3886  017150  004767  000172           DPC0A:  JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
3887  017154  000415                           BR      DPC2A
3888  017156  005277  161540           DPC1:   INC     @BPKP           ;BUMP PICK CNTR
3889  017162  005777  161534                   TST     @BPKP           ;SEE IF OVERFLOW
3890  017166  100027                           BPL     DPC2            ;IF NOT: BR
3891  017170  032777  002000  161410           BIT     #2000,@SWR      ;SEE IF HAVE PRINTED DATA
```

```
3892  017176  001402                          BEQ     DPC1A           ;IF SO: BR
3893  017200  004767  003774                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
3894  017204  004767  000136          DPC1A:  JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
3895  017210  016704  161462          DPC2A:  MOV     UNP,R4
3896  017214  016403  001010                  MOV     DRP1(R4),R3     ;SET DROP POINTER
3897  017220  016404  000770                  MOV     PIK1(R4),R4     ;SET PICK POINTER
3898  017224  012767  000010  161460         MOV     #10,BCNT        ;SET NUMBER OF BITS
3899  017232  005023                  DPC2B:  CLR     (R3)+           ;CLEAR DROPS
3900  017234  005024                          CLR     (R4)+           ;CLEAR PICK
3901  017236  005367  161450                  DEC     BCNT            ;SEE IF DONE
3902  017242  001373                          BNE     DPC2B           ;IF NOT: BR
3903  017244  000207                          RTS     PC              ;EXIT
3904  017246  000241                  DPC2:   CLC
3905  017250  106003                          RORB    R3              ;GET NEXT BIT
3906  017252  005367  161434                  DEC     BCNT            ;SEE IF DONE
3907  017256  001410                          BEQ     DPC3
3908  017260  062767  000002  161434         ADD     #2,BPKP
3909  017266  062767  000002  161424         ADD     #2,BDPP
3910  017274  000167  177606                  JMP     DPC0            ;CONTINUE
3911  017300  000207                  DPC3:   RTS     PC              ;RETURN
3912  017302  016704  161370          PICK:   MOV     UNP,R4          ;GET UNIT POINTER
3913  017306  016467  000770  161406         MOV     PIK1(R4),BPKP   ;SET PICK POINTER
3914  017314  016467  001010  161376         MOV     DRP1(R4),BDPP   ;SET DROP POINTER
3915  017322  116704  161316                  MOVB    TEMP1,R4        ;R4 = GOOD CHAR
3916  017326  116703  161314                  MOVB    TEMP2,R3        ;R3 = BAD CHAR
3917  017332  112767  000001  161310         MOVB    #1,TEMP3        ;SET PICK FLAG
3918  017340  004767  177526                  JSR     PC,DPC          ;GO CHECK PICKS
3919  017344  000207                          RTS     PC              ;EXIT
3920  017346  012704  025330          DPPRT:  MOV     #MSG26,R4
3921  017352  004767  004544                  JSR     PC,TTOUT        ;PRINT DROP HEADER
3922  017356  016704  161314                  MOV     UNP,R4
3923  017362  016467  001010  161330         MOV     DRP1(R4),BDPP   ;SET DROP POINTER
3924  017370  016467  000770  161324         MOV     PIK1(R4),BPKP   ;SET PICK POINTER
3925  017376  062767  000016  161314         ADD     #16,BDPP
3926  017404  062767  000016  161310         ADD     #16,BPKP
3927  017412  012767  000010  161272         MOV     #10,BCNT        ;SET NUMBER TO PRINT
3928  017420  017703  161274          DPPRT0: MOV     @BDPP,R3
3929  017424  004767  004644                  JSR     PC,OCTP         ;PRINT DROPS
3930  017430  005367  161256                  DEC     BCNT            ;SEE IF DONE
3931  017434  001404                          BEQ     DPPRT1          ;IF NOT: BR
3932  017436  162767  000002  161254         SUB     #2,BDPP         ;BUMP POINTER
3933  017444  000765                          BR      DPPRT0          ;CONTINUE FOR ALL 8 BITS
3934  017446  012767  000010  161236  DPPRT1: MOV     #10,BCNT        ;SET NUMBER TO PRINT
3935  017454  012704  025341                  MOV     #MSG27,R4
3936  017460  004767  004436                  JSR     PC,TTOUT        ;PRINT PICK HEADER
3937  017464  017703  161232          DPPRT2: MOV     @BPKP,R3
3938  017470  004767  004600                  JSR     PC,OCTP         ;PRINT PICKS
3939  017474  005367  161212                  DEC     BCNT            ;SEE IF DONE
3940  017500  001404                          BEQ     DPPRTX          ;IF SO: BR
3941  017502  162767  000002  161212         SUB     #2,BPKP         ;BUMP POINTER
3942  017510  000765                          BR      DPPRT2          ;CONTINUE FOR ALL 8 BITS
3943  017512  000207                  DPPRTX: RTS     PC              ;RETURN
```

```
3944
3945                                    ;*********************************************************
3946                                    ;STATUS CHECK SUBROUTINE:
3947                                    ;
3948                                    ;THIS SUBROUTINE IS USED TO PERFORM A CHECK OF
3949                                    ;BOTH THE MASSBUS CONTROLLER (RH11) AND THE TAPE
3950                                    ;CONTROLLER (TM02).  THE RH11 IS CHECKED FOR ERRORS
3951                                    ;AS REFLECTED IN REGISTERS CS1 AND CS2 AND ALSO THAT
3952                                    ;THE BUS ADDRESS (BA) AND WORD COUNT (WC) ARE
3953                                    ;CORRECT.  THE TM02 IS CHECKED FOR DRIVE STATIS (DS),
3954                                    ;DRIVE ERRORS (ER), AND PROPER FRAME COUNT.  THE SPECIAL
3955                                    ;CHECK CHARACTERS (CRC+LRC) ARE ALSO CHECKED WHEN
3956                                    ;APPROPIATE (IE: NRZ READ OR WRITE).  CERTAIN TYPES
3957                                    ;OF DRIVE ERRORS IN PE OPERATION WILL BE ACCOMPANIED
3958                                    ;BY THE DISPLAY OF THE DEAD TRACK REGISTER (CC).  THESE
3959                                    ;TYPES ARE ER BITS 15,10,7,6.  THE PRINTOUTS OF BAD
3960                                    ;CRC,LRC,FC, AND BA WILL SHOW BOTH THE EXPECTED AND
3961                                    ;RECEIVED VALUES (IE: EXPT-RCVD).  ONLY THOSE REGISTERS
3962                                    ;WHICH ARE IN ERROR WILL BE PRINTED AND ALL PRINTOUTS
3963                                    ;ARE IN OCTAL FORMAT WITH NO LEADING ZEROS.  AS IN
3964                                    ;DATA ERRORS, STATUS ERRORS ARE PRECEEDED BY HEADER
3965                                    ;DESCRIBING THE HARDWARE UNDER TEST, THE BLOCKING
3966                                    ;INFORMATION, AND THE ERROR TYPE.
3967                                    ;*********************************************************
3968
3969  017514  016703  161036       ERCHK:  MOV     FMCNT,R3        ;GET FRAME COUNT
3970  017520  032703  000001               BIT     #1,R3           ;SEE IF ODD
3971  017524  001401                       BEQ     ER0             ;IF NOT: BR
3972  017526  005303                       DEC     R3              ;BUMP COUNT
3973  017530  005403               ER0:    NEG     R3
3974  017532  032767  000020  161012       BIT     #20,UDES        ;SEE IF CORE DUMP
3975  017540  001402                       BEQ     ER0B            ;IF NOT: BR
3976  017542  000241                       CLC
3977  017544  006003                       ROR     R3              ;SET TO FC/2
3978  017546  032767  000010  161120 ER0B:  BIT     #10,MTC1        ;SEE IF WRITE OP
3979  017554  001414                       BEQ     ER1             ;IF SO: BR
3980  017556  032767  010000  160776       BIT     #10000,RDCMD
3981  017564  001405                       BEQ     ER0A
3982  017566  012703  033434               MOV     #RDATA,R3
3983  017572  162703  000002               SUB     #2,R3           ;SET POINTER
3984  017576  000405                       BR      ER2
3985  017600  062703  033434       ER0A:   ADD     #RDATA,R3       ;BUILD EXPT READ ADDRESS
3986  017604  000402                       BR      ER2
3987  017606  062703  027426       ER1:    ADD     #WDATA,R3       ;BUILD EXPT WRITE ADDRESS
3988  017612  010367  001514       ER2:    MOV     R3,CADER        ;SAVE ADDRESS
3989  017616  012704  000007               MOV     #7,R4
3990  017622  012701  021334               MOV     #BAER,R1
3991  017626  005021               ER2A0:  CLR     (R1)+           ;CLEAR FLAGS
3992  017630  005304                       DEC     R4
3993  017632  001375                       BNE     ER2A0
3994  017634  020377  160654               CMP     R3,@BA          ;SEE IF ADDRESS OK
3995  017640  001402                       BEQ     ER2A1           ;IF SO: BR
3996  017642  005267  001466               INC     BAER            ;SET BUS ADDRESS ERROR
3997  017646  032767  000010  161020 ER2A1:  BIT     #10,MTC1        ;SEE IF WRITE OPER
3998  017654  001007                       BNE     ER2B            ;IF NOT: BR
3999  017656  005777  160634       ER2A:   TST     @FC             ;SEE IF FC=0
```

```
 40C0  017662  001443                        BEQ     ER_             ;IF SO: BR
 4001  017664  005267  001452                INC     FCER            ;SET FC ERROR
 4002  017670  000167  000076                JMP     ER3
 4003  017674  032767  000040  160772 ER2B:  BIT     #40,MTC1        ;SEE IF SPACE OPER
 4004  017702  001765                        BEQ     ER2A            ;IF SO: BR
 4005  017704  005767  160770                TST     TMFLG           ;SEE IF TM TIME
 4006  017710  001012                        BNE     ER2D            ;IF SO: BR
 4007  017712  016703  160640                MOV     FMCNT,R3
 4008  017716  005403                        NEG     R3              ;R3 = EXPT RECORD SIZE
 4009  017720  020377  160572        ER2C:   CMP     R3,@FC          ;SEE IF FC = EXPT
 4010  017724  001422                        BEQ     ER3             ;IF SO: BR
 4011  017726  005267  001410                INC     FCER            ;SET FC ERROR FLAG
 4012  017732  000167  000034                JMP     ER3
 4013  017736  032767  002000  160606 ER2D:  BIT     #2000,UDES      ;SEE IF PE
 4014  017744  001344                        BNE     ER2A            ;IF SO: BR
 4015  017746  032767  010000  160606        BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 4016  017754  001003                        BNE     ER2E            ;IF SO: BR
 4017  017756  012703  000002                MOV     #2,R3
 4018  017762  000756                        BR      ER2C            ;LOOK FOR EXPT = 2
 4019  017764  012703  000001        ER2E:   MOV     #1,R3
 4020  017770  000753                        BR      ER2C            ;GO CHECK FC FOR TM
 4021  017772  032777  160000  160510 ER3:   BIT     #160000,@C1     ;SEE IF COUNT ERROR
 4022  020000  001442                        BEQ     ER4
 4023  020002  017703  160512                MOV     @CS,R3          ;GET CONT STATUS REG
 4024  020006  042703  000307                BIC     #307,R3         ;MASK OUT IR,OR,UNIT NO.
 4025  020012  005703                        TST     R3              ;SEE IF ANY OTHER ERRORS
 4026  020014  001407                        BEQ     ER3A            ;IF NOT: BR
 4027  020016  005767  160656                TST     TMFLG           ;SEE IF TAPE MARK TIME
 4028  020022  001427                        BEQ     ER3B            ;IF NOT: BR
 4029  020024  042703  001000                BIC     #1000,R3        ;MASK MISSED TRANS
 4030  020030  005703                        TST     R3              ;SEE IF ANY OTHER ERRORS
 4031  020032  001023                        BNE     ER3B            ;IF SO: BR
 4032  020034  032777  060000  160446 ER3A:  BIT     #60000,@C1      ;SEE IF EITHER TRE OR MCPE
 4033  020042  001421                        BEQ     ER4             ;IF NOT: BR
 4034  020044  005767  160630                TST     TMFLG           ;SEE IF TM TIME
 4035  020050  001414                        BEQ     ER3B            ;IF NOT: BR
 4036  020052  017703  160446                MOV     @ER,R3          ;GET ERROR REGISTER
 4037  020056  032767  000010  160466        BIT     #10,UDES        ;SEE IF EVEN PARITY
 4038  020064  001402                        BEQ     ER3A1           ;IF NOT: BR
 4039  020066  042703  000100                BIC     #100,R3         ;MASK PAR
 4040  020072  042703  001000        ER3A1:  BIC     #1000,R3        ;MASK FCE
 4041  020076  005703                        TST     R3
 4042  020100  001402                        BEQ     ER4             ;IF NO ERRORS EXCEPT FCE: BR
 4043  020102  005267  001230        ER3B:   INC     CONER           ;SET CONT ERROR FLAG
 4044  020106  032777  040000  160406 ER4:   BIT     #40000,@DS      ;SEE IF DRIVE ERROR
 4045  020114  001421                        BEQ     ER6             ;IF NOT: BR
 4046  020116  005767  160556                TST     TMFLG           ;SEE IF TAPE MARK TIME
 4047  020122  001414                        BEQ     ER4A            ;IF NOT: BR
 4048  020124  017703  160374                MOV     @ER,R3          ;GET ER
 4049  020130  032767  000010  160414        BIT     #10,UDES        ;SEE IF EVEN PARITY
 4050  020136  001402                        BEQ     ER4A1           ;IF NOT: BR
 4051  020140  042703  000100                BIC     #100,R3         ;MASK PAR
 4052  020144  042703  001000        ER4A1:  BIC     #1000,R3        ;MASK OUT FCE
 4053  020150  005703                        TST     R3              ;SEE IF ANY OTHER ERRORS
 4054  020152  001402                        BEQ     ER6             ;IF NOT: BR
 4055  020154  005267  001160        ER4A:   INC     DRVER           ;SET DRIVER ERROR FLAG
```

```
 4056  020160  032767  002000  160364  ER6:    BIT     #2000,UDES
 4057  020166  001071                  BNE     ERPT            ;IF IN PE MODE: BR
 4058  020170  032777  020000  160410          BIT     #20000,@SWR     ;SEE IF NO DATA CHECK
 4059  020176  001065                  BNE     ERPT            ;IF NOT: BR (ALLOW READ OF UNKNOWN TAPES)
 4060  020200  032767  000040  160466          BIT     #40,MTC1        ;SEE IF WRITE OR READ OP
 4061  020206  001461                  BEQ     ERPT            ;IF NOT: BR
 4062  020210  005767  160464                  TST     TMFLG           ;SEE IF TAPE MARK TIME
 4063  020214  001413                  BEQ     ER6A            ;IF NOT: BR
 4064  020216  016767  175356  001130          MOV     EXCRC,CRCSV     ;SAVE CRC
 4065  020224  016767  175352  001120          MOV     EXLRC,LRCSV     ;SAVE LRC
 4066  020232  005067  175342                  CLR     EXCRC
 4067  020236  012767  000023  175336          MOV     #23,EXLRC       ;SET CRC/LRC FOR TM
 4068  020244  032767  000060  160300  ER6A:   BIT     #60,UDES
 4069  020252  001037                  BNE     ERPT
 4070  020254  017703  160250                  MOV     @CC,R3          ;GET CRC CHARACTER
 4071  020260  042703  177000                  BIC     #177000,R3
 4072  020264  026703  175310                  CMP     EXCRC,R3
 4073  020270  001402                  BEQ     ER7             ;IF CRC GOOD: BR
 4074  020272  005267  001050                  INC     CRCER           ;SET ERROR FLAG
 4075  020276  017703  160232          ER7:    MOV     @MR,R3          ;GET LRC
 4076  020302  000303                  SWAB    R3
 4077  020304  005703                  TST     R3
 4078  020306  100002                  BPL     ER10
 4079  020310  052703  000400                  BIS     #400,R3
 4080  020314  042703  177000          ER10:   BIC     #177000,R3
 4081  020320  026703  175256                  CMP     EXLRC,R3
 4082  020324  001412                  BEQ     ERPT            ;IF LRC GOOD: BR
 4083  020326  010367  001016                  MOV     R3,ACTLRC       ;SAVE ACTUAL LRC
 4084  020332  005267  001006                  INC     LRCER           ;SET LRC ERROR FLAG
 4085  020336  032767  010000  160216          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 4086  020344  001402                  BEQ     ERPT            ;IF NOT: BR
 4087  020346  005067  000772                  CLR     LRCER           ;ELSE CLEAR LRC ERROR
 4088  020352  012703  000006          ERPT:   MOV     #6,R3
 4089  020356  005067  160326                  CLR     SERFL           ;CLEAR ERROR FLAG
 4090  020362  005067  160336                  CLR     ERSAV
 4091  020366  012704  021334                  MOV     #BAER,R4
 4092  020372  005724                  ERPTT:  TST     (R4)+           ;SEE IF ANY ERROR
 4093  020374  001004                  BNE     ERPTG           ;IF SO: BR
 4094  020376  005303                  DEC     R3
 4095  020400  001374                  BNE     ERPTT
 4096  020402  000167  000670                  JMP     ERPX1
 4097  020406  005267  160276          ERPTG:  INC     SERFL           ;SET ERROR FLAG
 4098  020412  017767  160106  160304          MOV     @ER,ERSAV       ;SAVE ERROR REGISTER
 4099  020420  032777  002000  160160          BIT     #2000,@SWR      ;SEE IF PRINT
 4100  020426  001420                  BEQ     ERPTO           ;IF SO: BR
 4101  020430  022767  000002  160256          CMP     #2,RTYFL        ;SEE IF READ RETRY
 4102  020436  001006                  BNE     ERPTG1          ;IF NOT: BR
 4103  020440  016703  160240                  MOV     RTCNT,R3
 4104  020444  005203                  INC     R3              ;BUMP RETRY COUNT
 4105  020446  020367  160130                  CMP     R3,RETRY        ;SEE IF LAST RETRY
 4106  020452  001406                  BEQ     ERPTO           ;IF SO: BR
 4107  020454  022767  000002  000656  ERPTG1: CMP     #2,DRVER        ;SEE IF TM STATUS ERROR
 4108  020462  001402                  BEQ     ERPTO           ;IF SO: BR
 4109  020464  000167  000510                  JMP     ERPXO
 4110  020470  005267  160176          ERPTO:  INC     PFLG
 4111  020474  004767  002500                  JSR     PC,PAPRT        ;PRINT HEADER
```

```
 4112  020500  016704  160150                MOV     EMADDR,R4
 4113  020504  004767  003412                JSR     PC,TTOUT        ;PRINT ERROR TYPE
 4114  020510  004767  000642                JSR     PC,FRPRT        ;PRINT F OR R
 4115  020514  005767  160160                TST     TMFLG
 4116  020520  001410                        BEQ     ERPT1
 4117  020522  022767  026210  160124        CMP     #MSG54,EMADDR
 4118  020530  001404                        BEQ     ERPT1
 4119  020532  012704  026226                MOV     #MSG56,R4       ;PRINT TM
 4120  020536  004767  003360                JSR     PC,TTOUT
 4121  020542  005767  000570        ERPT1:  TST     CONER
 4122  020546  001420                        BEQ     ERPT2           ;IF NO CONT ERROR: BR
 4123  020550  012704  025160                MOV     #MSG23,R4
 4124  020554  004767  003342                JSR     PC,TTOUT        ;PRINT C1 TAG
 4125  020560  017703  157724                MOV     @C1,R3
 4126  020564  004767  003504                JSR     PC,OCTP         ;PRINT CONTROL 1
 4127  020570  012704  025205                MOV     #MSG23D,R4      ;PRINT CS TAG
 4128  020574  004767  003322                JSR     PC,TTOUT
 4129  020600  017703  157714                MOV     @CS,R3
 4130  020604  004767  003464                JSR     PC,OCTP         ;PRINT CONT STATUS
 4131  020610  005767  000524        ERPT2:  TST     DRVER
 4132  020614  001420                        BEQ     ERPT3           ;IF SO DRIVE ERROR: BR
 4133  020616  012704  025213                MOV     #MSG23E,R4
 4134  020622  004767  003274                JSR     PC,TTOUT        ;PRINT DS TAG
 4135  020626  017703  157670                MOV     @DS,R3
 4136  020632  004767  003436                JSR     PC,OCTP         ;PRINT DRIVE STATUS
 4137  020636  012704  025220                MOV     #MSG23F,R4
 4138  020642  004767  003254                JSR     PC,TTOUT        ;PRINT ER TAG
 4139  020646  017703  157652                MOV     @ER,R3
 4140  020652  004767  003416                JSR     PC,OCTP         ;PRINT DRIVE ERROR
 4141  020656  005767  000452        ERPT3:  TST     BAER
 4142  020662  001421                        BEQ     ERPT4           ;IF NO BA ERROR: BR
 4143  020664  012704  025173                MOV     #MSG23B,R4
 4144  020670  004767  003226                JSR     PC,TTOUT        ;PRINT BA TAG
 4145  020674  017703  157614                MOV     @BA,R3
 4146  020700  004767  003370                JSR     PC,OCTP         ;PRINT BUS ADDRESS
 4147  020704  012767  000255  157726        MOV     #255,TOB
 4148  020712  004767  003314                JSR     PC,TOG          ;PRINT /
 4149  020716  016703  000410                MOV     CADER,R3
 4150  020722  004767  003346                JSR     PC,OCTP         ;PRINT EXPT BUS ADDRESS
 4151  020726  005767  000410        ERPT4:  TST     FCER
 4152  020732  001410                        BEQ     ERPT5           ;IF NO FC ERROR: BR
 4153  020734  012704  025200                MOV     #MSG23C,R4
 4154  020740  004767  003156                JSR     PC,TTOUT        ;PRINT FC TAG
 4155  020744  017703  157546                MOV     @FC,R3
 4156  020750  004767  003320                JSR     PC,OCTP         ;PRINT FRAME COUNT
 4157  020754  012704  025166        ERPT5:  MOV     #MSG23A,R4
 4158  020760  004767  003136                JSR     PC,TTOUT        ;PRINT WC TAG
 4159  020764  017703  157522                MOV     @WC,R3
 4160  020770  004767  003300                JSR     PC,OCTP         ;PRINT WORD COUNT
 4161  020774  005767  000346                TST     CRCER
 4162  021000  001423                        BEQ     ERPT5A          ;IF NO CRC ERROR: BR
 4163  021002  012704  026253                MOV     #MSG58,R4
 4164  021006  004767  003110                JSR     PC,TTOUT        ;PRINT CRC TAG
 4165  021012  017703  157512                MOV     @CC,R3
 4166  021016  042703  177000                BIC     #177000,R3
 4167  021022  004767  003246                JSR     PC,OCTP         ;PRINT ACTUAL CRC
```

```
 4168  021026  012767  000255  157604          MOV     #255,TOB
 4169  021034  004767  003172                  JSR     PC,TOG
 4170  021040  016703  174534                  MOV     EXCRC,R3
 4171  021044  004767  003224                  JSR     PC,OCTP         ;PRINT EXPECTED CRC
 4172  021050  005767  000270          ERPT5A: TST     LRCER
 4173  021054  001421                          BEQ     ERPT6           ;IF NO LRC ERROR: BR
 4174  021056  012704  026261                  MOV     #MSG59,R4
 4175  021062  004767  003034                  JSR     PC,TTOUT        ;PRINT LRC TAG
 4176  021066  016703  000256                  MOV     ACTLRC,R3
 4177  021072  004767  003176                  JSR     PC,OCTP         ;PRINT ACTUAL LRC
 4178  021076  012767  000255  157534          MOV     #255,TOB
 4179  021104  004767  003122                  JSR     PC,TOG
 4180  021110  016703  174466                  MOV     EXLRC,R3
 4181  021114  004767  003154                  JSR     PC,OCTP         ;PRINT EXPECTED LRC
 4182  021120  005767  000214          ERPT6:  TST     DRVER
 4183  021124  001424                          BEQ     ERPT7           ;IF NO DRIVE ERROR: BR
 4184  021126  032767  002000  157416          BIT     #2000,UDES
 4185  021134  001420                          BEQ     ERPT7           ;IF NO PE: BR
 4186  021136  017704  157362                  MOV     @ER,R4
 4187  021142  042704  075477                  BIC     #75477,R4       ;MASK OUT ALL BUT BITS 15,10,7,6
 4188  021146  005704                          TST     R4
 4189  021150  001412                          BEQ     ERPT7           ;IF NO CONDITIONALS SET: BR
 4190  021152  012704  025232                  MOV     #MSG23H,R4
 4191  021156  004767  002740                  JSR     PC,TTOUT        ;PRINT CC TAG
 4192  021162  017703  157342                  MOV     @CC,R3
 4193  021166  042703  177000                  BIC     #177000,R3      ;MASK CC
 4194  021172  004767  003076                  JSR     PC,OCTP         ;PRINT CHECK CHARACTERS
 4195  021176  000240                  ERPT7:  NOP
 4196  021200  032777  100000  157400  ERPX0:  BIT     #100000,@SWR    ;SEE IF STOP ON ERROR
 4197  021206  001412                          BEQ     ERPX            ;IF NOT: BR
 4198  021210  000000                          HALT
 4199  021212  005767  157454                  TST     PFLG            ;SEE IF HAVE PRINTED
 4200  021216  001006                          BNE     ERPX            ;IF SO: BR
 4201  021220  032777  002000  157360          BIT     #2000,@SWR      ;SEE IF SHOULD PRINT
 4202  021226  001002                          BNE     ERPX            ;IF NOT: BR
 4203  021230  000167  177234                  JMP     ERPTO           ;PRINT ERROR
 4204  021234  005067  157432          ERPX:   CLR     PFLG
 4205  021240  012777  000011  157242          MOV     #11,@C1         ;DRIVE CLEAR
 4206  021246  017704  157254                  MOV     @AS,R4
 4207  021252  010477  157250                  MOV     R4,@AS          ;CLEAR AS
 4208  021256  016704  157226                  MOV     C1,R4
 4209  021262  005204                          INC     R4
 4210  021264  152714  000100                  BISB    #100,(R4)       ;RESET TRE
 4211  021270  016777  157256  157244          MOV     UDES,@C2        ;RESET TC
 4212  021276  032767  000040  157370  ERPX1:  BIT     #40,MTC1
 4213  021304  001411                          BEQ     ERPX2           ;IF NOT READ/WRITE OP: BR
 4214  021306  005767  157366                  TST     TMFLG
 4215  021312  001406                          BEQ     ERPX2           ;IF NOT TM TIME: BR
 4216  021314  016767  000034  174256          MOV     CRCSV,EXCRC     ;RESTORE CRC
 4217  021322  016767  000024  174252          MOV     LRCSV,EXLRC     ;RESTORE LRC
 4218  021330  000207                  ERPX2:  RTS     PC              ;EXIT
 4219  021332  000000                  CADER:  0                       ;EXPT ADDRESS SAVE
 4220  021334  000000                  BAER:   0
 4221  021336  000000                  CONER:  0
 4222  021340  000000                  DRVER:  0
 4223  021342  000000                  FCER:   0
```

```
 4224  021344  000000                    LRCER:  0
 4225  021346  000000                    CRCER:  0
 4226  021350  000000                    ACTLRC: 0
 4227  021352  000000                    LRCSV:  0
 4228  021354  000000                    CRCSV:  0
 4229
 4230                                    ;*******************************************************
 4231                                    ;F FOR FORWARD/R FOR REVERSE PRINT SUBROUTINE:
 4232                                    ;
 4233                                    ;THIS SUBROUTINE IS USED TO PRINT OUT THE
 4234                                    ;TAPE DIRECTION USED WHEN ANY ERROR IS
 4235                                    ;DETECTED IN STATUS OF READ OR WRITE, DATA, OR
 4236                                    ;SPACING OPERATIONS.
 4237                                    ;*******************************************************
 4238
 4239  021356  032767  000010  157310    FRPRT:  BIT     #10,MTC1        ;SEE IF WRITE COMMAND
 4240  021364  001415                            BEQ     FREX            ;IF SO: BR
 4241  021366  032767  000002  157300            BIT     #2,MTC1         ;SEE IF REVERSE
 4242  021374  001405                            BEQ     FRO             ;IF NOT: BR
 4243  021376  012704  025115                    MOV     #MSG17,R4
 4244  021402  004767  002514                    JSR     PC,TTOUT        ;PRINT R
 4245  021406  000404                            BR      FREX
 4246  021410  012704  025110            FRO:    MOV     #MSG16,R4
 4247  021414  004767  002502                    JSR     PC,TTOUT        ;PRINT F
 4248  021420  000207                    FREX:   RTS     PC              ;EXIT
.4249
```

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

021422  005067  157216          TAPG:   CLR     TEMP1
021426  016777  157116  157064          MOV     DVN,@CS         ;SET DRIVE NO.
021434  016704  157236          TAPG0:  MOV     UNP,R4          ;GET UNIT POINTER
021440  032777  010000  157054          BIT     #10000,@DS      ;SEE IF HAVE MOL
021446  001046                        BNE     TAPG3           ;IF SO: BR
021450  005267  157170                INC     TEMP1           ;SEE IF TIMED OUT
021454  001367                        BNE     TAPG0           ;WAIT FOR READY
021456  032777  002000  157122        BIT     #2000,@SWR      ;SKIP PRINT?
021464  001034                        BNE     MOLEX
021466  036467  022122  157140        BIT     MOLTAB(4),MOLSW ;FIRST TIME?
021474  001030                        BNE     MOLEX           ;NO
021476  056467  022122  157130        BIS     MOLTAB(4),MOLSW ;SET FLAG
021504  004767  001470                JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
021510  032767  000010  157156        BIT     #10,MTC1        ;SEE IF WRITE OP
021516  001005                        BNE     TAPG1           ;IF NOT: BR
021520  012704  024750                MOV     #MSG5,R4
021524  004767  002372                JSR     PC,TTOUT        ;PRINT WRITE ERR
021530  000406                        BR      TAPG2
021532  012704  024755          TAPG1:  MOV     #MSG6,R4
021536  004767  002360                JSR     PC,TTOUT        ;PRINT READ ERR
021542  004767  177610                JSR     PC,FRPRT        ;PRINT F OR R
021546  012704  025310          TAPG2:  MOV     #MSG25,R4
021552  004767  002344                JSR     PC,TTOUT        ;PRINT NO MOL ERR
021556  005726                  MOLEX:  POPSP
021560  000167  162432                JMP     START7
021564  016704  157106          TAPG3:  MOV     UNP,R4
021570  046467  022122  157036        BIC     MOLTAB(4),MOLSW ;CLEAR FLAG
```

```
 4306  021576  032777  020000  156716          BIT     #20000,@DS      ;SEE IF PIP RESET
 4307  021604  001412                          BEQ     TAPG3F          ;IF SO: BR
 4308  021606  004767  001366                  JSR     PC,PAPRT        ;PRINT HEADER
 4309  021612  012704  027352                  MOV     #MSG116,R4
 4310  021616  004767  002300                  JSR     PC,TTOUT        ;PRINT REWINDING MESSAGE
 4311  021622  032777  020000  156672  1$:     BIT     #20000,@DS
 4312  021630  001374                          BNE     1$              ;AWAIT PIP RESET
 4313  021632  022767  000026  157034  TAPG3F: CMP     #26,MTC1        ;SEE IF WRITE TM
 4314  021640  001003                          BNE     TAPG3A          ;IF NOT: BR
 4315  021642  012704  177777                  MOV     #-1,R4          ;ELSE SET FC FOR -1
 4316  021646  000406                          BR      TAPG3B
 4317  021650  016704  156702          TAPG3A: MOV     FMCNT,R4
 4318  021654  032704  000001                  BIT     #1,R4
 4319  021660  001401                          BEQ     TAPG3B
 4320  021662  005304                          DEC     R4
 4321  021664  000261                  TAPG3B: SEC
 4322  021666  006004                          ROR     R4              ;SET WC = FC/2 FOR NORMAL FORMAT
 4323  021670  032767  000020  156654          BIT     #20,UDES        ;SEE IF CORE DUMP FORMAT
 4324  021676  001402                          BEQ     TAPG3C          ;IF NOT: BR
 4325  021700  000261                          SEC
 4326  021702  006004                          ROR     R4              ;SET WC = FC/4 FOR CORE DUMP
 4327  021704  010477  156602          TAPG3C: MOV     R4,@WC          ;SET WORD COUNT
 4328  021710  012777  000011  156572          MOV     #11,@C1         ;DRIVE CLEAR
 4329  021716  017777  156574  156572          MOV     @FC,@FC         ;RESET FC LOADED
 4330  021724  005767  156636                  TST     INTRF           ;SEE IF INTERCHANGE READ
 4331  021730  001407                          BEQ     TAPG3D          ;IF NOT: BR
 4332  021732  032767  000040  156734          BIT     #40,MTC1        ;SEE IF READ OP
 4333  021740  001403                          BEQ     TAPG3D          ;IF NOT: BR
 4334  021742  012777  000003  156564          MOV     #3,@MR          ;SET INTERCHANGE READ MAINT. MODE
 4335  021750  016704  156720          TAPG3D: MOV     MTC1,R4         ;GET COMMAND
 4336  021754  042704  177707                  BIC     #177707,R4      ;MASK OP CODE
 4337  021760  022704  000030                  CMP     #30,R4          ;SEE IF SPACE OP CODE
 4338  021764  001403                          BEQ     TAPG3E          ;IF SO: BR
 4339  021766  012767  177740  156674          MOV     #-40,STAL       ;SET INTERRUPT DELAY MULT TO 40
 4340  021774  052767  000101  156672  TAPG3E: BIS     #101,MTC1       ;SET INTERRUPT ENABLE AND GO
 4341  022002  000240                          NOP
 4342  022004  016777  156664  156476          MOV     MTC1,@C1        ;EXECUTE COMMAND
 4343  022012  005077  156566                  CLR     @PSW            ;CLEAR PRIORITY
 4344  022016  005067  156622                  CLR     TEMP1
 4345  022022  005267  156616          TAPG4:  INC     TEMP1           ;SEE IF HAVE TIMED OUT
 4346  022026  001375                          BNE     TAPG4           ;IF NOT: BR
 4347  022030  005267  156634                  INC     STAL
 4348  022034  001372                          BNE     TAPG4           ;DO TIME DELAY MULTIPLIER
 4349  022036  012777  000340  156540  TAPG5:  MOV     #340,@PSW       ;RESET PRIORITY
 4350  022044  032777  002000  156534          BIT     #2000,@SWR      ;SEE IF SHOULD PRINT ERRORS
 4351  022052  001014                          BNE     TAPG6           ;IF NOT: BR
 4352  022054  004767  001120                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
 4353  022060  016704  156570                  MOV     EMADDR,R4
 4354  022064  004767  002032                  JSR     PC,TTOUT        ;PRINT ERROR OP
 4355  022070  004767  177262                  JSR     PC,FRPRT        ;PRINT F OR R
 4356  022074  012704  025270                  MOV     #MSG24,R4
 4357  022100  004767  002016                  JSR     PC,TTOUT        ;PRINT NO INTERRUPT
 4358  022104  032777  100000  156474  TAPG6:  BIT     #100000,@SWR    ;SEE IF SHOULD HALT ON ERROR
 4359  022112  001401                          BEQ     TAPG7           ;IF NOT: BR
 4360  022114  000000                          HALT
 4361  022116  000167  000100          TAPG7:  JMP     @TINTA          ;RETURN TO CALLING ROUTINE
```

4362

```
 4363  022122  000001          MOLTAB: 1
 4364  022124  000002                  2
 4365  022126  000004                  4
 4366  022130  000010                  10
 4367  022132  000020                  20
 4368  022134  000040                  40
 4369  022136  000100                  100
 4370  022140  000200                  200
```

```
 4371
 4372                                        ;TTY INTERRUPT HANDLER**********
 4373
 4374  022142  012777  000340  156434  TTINT:  MOV     #340,@PSW       ;RESET PSW
 4375  022150  005077  156434          CLR     @TKS            ;CLEAR TTY STATUS
 4376  022154  122777  000203  156430  CMPB    #203,@TKB       ;SEE IF CONT C
 4377  022162  001401                  BEQ     TTINT0          ;IF SO: BR
 4378  022164  000002                  RTI                     ;ELSE RETURN
 4379  022166  010067  156456  TTINT0: MOV     R0,TEMP3        ;SAVE R0(REC CNTR)
 4380  022172  004767  171560          JSR     PC,TINP4        ;GO GET STALL VALUES
 4381  022176  016700  156446          MOV     TEMP3,R0        ;RESTORE R0(REC CNTR)
 4382  022202  005077  156404          CLR     @TKB            ;CLEAR TTY BUFFER
 4383  022206  012777  000100  156374  MOV     #100,@TKS       ;RESET INTERRUPT ENABLE
 4384  022214  000002                  RTI                     ;RETURN
 4385
 4386                                        ;MAG TAPE INTERRUPT HANDLER**********
 4387
 4388  022216  000240          MTINT:  NOP
 4389  022220  022626                  CMP     (SP)+,(SP)+     ;RESET STACK POINTER
 4390  022222  042777  000037  156304  MTINTA: BIC     #37,@MR         ;CLEAR MAINT MODE
 4391  022230  000177  156430          JMP     @RTRN           ;RETURN
```

```
                                        ;************************************
                                        ;AUTO SEQUENCE
                                        ;
                                        ;THIS ROUTINE ,ENTERED VIA STARTING ADDRESS 240
                                        ;WILL EXERCISE ALL AVAILABLE SLAVES ON ALL AVAILABLE
                                        ;DRIVES IN BOTH PE AND NRZ ACCORDING TO THE PRESELECTED
                                        ;TEST PLAN. IF NRZ ONLY, PE TESTING WILL NOT BE ATTEMPTED.
                                        ;************************************

022234  012704  027162        ASEQ:   MOV     #MSG108,R4
022240  004767  001656                JSR     PC,TTOUT        ;PRINT NRZ ONLY REQUEST
022244  012705  000652                MOV     #NRZOF,R5       ;SET ADDRESS OF FLAG
022250  012701  000001                MOV     #1,R1           ;SET SIZE OF ENTRY
022254  012702  000001                MOV     #1,R2           ;SET UPPER LIMIT
022260  012703  000000                MOV     #0,R3           ;SET LOWER LIMIT
022264  004767  001374                JSR     PC,TTR          ;GO GET RESPONSE
022270  012704  026757                MOV     #MSG104,R4
022274  004767  001622                JSR     PC,TTOUT        ;REQUEST CONT OR NOT
022300  012705  000744                MOV     #ASEQCF,R5      ;SET ADDRESS OF ENTRY
022304  012701  000001                MOV     #1,R1           ;SET SIZE OF ENTRY
022310  012702  000001                MOV     #1,R2           ;SET UPPER LIMIT
022314  012703  000000                MOV     #0,R3           ;SET LOWER LIMIT
022320  004767  001340                JSR     PC,TTR          ;GO GET INPUT
022324  005067  156410        ASEQ0:  CLR     ADRVN           ;CLEAR DRV NUM
022330  004767  000130        ASEQ1:  JSR     PC,HRDS         ;GO SELECT HARDWARE CONFIGURATION
022334  012704  026711                MOV     #MSG101,R4
022340  004767  001556                JSR     PC,TTOUT        ;PRINT DIVIDER
022344  012704  026737                MOV     #MSG102,R4
022350  004767  001546                JSR     PC,TTOUT        ;PRINT TM02 NUMBER
022354  016703  156360                MOV     ADRVN,R3
022360  004767  001710                JSR     PC,OCTP         ;PRINT TM02
022364  012704  026746                MOV     #MSG103,R4
022370  004767  001526                JSR     PC,TTOUT        ;PRINT SLAVE HDR
022374  012700  000746                MOV     #UN1,R0         ;POINT TO START OF SLAVE TABLE
022400  005710                ASEQ2:  TST     (R0)            ;SEE IF END
022402  100404                        BMI     ASEQ3           ;IF SO: BR
022404  012003                        MOV     (R0)+,R3
022406  004767  001662                JSR     PC,OCTP         ;PRINT SLAVE TABLE
022412  000772                        BR      ASEQ2           ;DO ALL
022414  004767  000216        ASEQ3:  JSR     PC,AMOD1        ;GO DO MODE 1(NRZ)
022420  004767  000406                JSR     PC,AMOD2        ;GO DO MODE 2(PE)
022424  022767  000007  156306ASEQ4:  CMP     #7,ADRVN        ;SEE IF DONE ALL DRIVES
022432  001403                        BEQ     ASEQX           ;IF SO: BR
022434  005267  156300                INC     ADRVN           ;BUMP DRIVE NUMBER
022440  000733                        BR      ASEQ1           ;CONTINUE
022442  005767  156276        ASEQX:  TST     ASEQCF          ;SEE IF CONTINUOUS AUTO SEQ
022446  001005                        BNE     ASEQXX          ;IF SO: BR
022450  012704  026670                MOV     #MSG100,R4
022454  004767  001442                JSR     PC,TTOUT        ;PRINT END OF PASS
022460  000000                        HALT
022462  000720                ASEQXX: BR      ASEQ0
```

```
 4443
 4444                                   ;SUBROUTINE TO SELECT AUTO SEQUENCE HARDWARE********
 4445
 4446  022464  005067  162312          HRDS:   CLR     REOTC           ;CLEAR EOT UNIT CNTR
 4447  022470  005067  156150                  CLR     TEMP1
 4448  022474  012777  000040  156016          MOV     #40,@CS         ;INIT
 4449  022502  016777  156232  156010          MOV     ADRVN,@CS       ;SET DRIVE
 4450  022510  017701  156022                  MOV     @DT,R1          ;READ DRIVE TYPE
 4451  022514  032777  010000  155776          BIT     #10000,@CS      ;TEST FOR NON-EXISTANT DRIVE
 4452  022522  001403                          BEQ     HRDS1           ;IF DRIVE AVAIL: BR
 4453  022524  005726                  HRDS0:  TST     (SP)+           ;RESET STACK POINTER
 4454  022526  000167  177672                  JMP     ASEQ4           ;IF NOT: BR
 4455  022532  032701  040000          HRDS1:  BIT     #40000,R1       ;SEE IF DRIVE IS TAPE
 4456  022536  001772                          BEQ     HRDS0           ;IF NOT: BR
 4457  022540  005000                          CLR     R0
 4458  022542  012701  000746                  MOV     #UN1,R1         ;SET START OF SLAVE TABLE
 4459  022546  010077  155770          HRDS2:  MOV     R0,@C2          ;SELECT SLAVE
 4460  022552  032777  010000  155742          BIT     #10000,@DS      ;SEE IF SLAVE AVAIL FOR TEST(MOL)
 4461  022560  001403                          BEQ     HRDS3           ;IF NOT: BR
 4462  022562  005267  156056                  INC     TEMP1           ;SET SLAVE FOUND FLAG
 4463  022566  010021                          MOV     R0,(R1)+        ;LOAD SLAVE TABLE
 4464  022570  022700  000007          HRDS3:  CMP     #7,R0           ;SEE IF DONE ALL SLAVES
 4465  022574  001402                          BEQ     HRDS4           ;IF SO: BR
 4466  022576  005200                          INC     R0              ;ELSE BUMP SLAVE NUMBER
 4467  022600  000762                          BR      HRDS2           ;CONTINUE SELECTION
 4468  022602  005767  156036          HRDS4:  TST     TEMP1           ;SEE IF FOUND ANY SLAVES
 4469  022606  001746                          BEQ     HRDS0           ;IF NOT: BR
 4470  022610  016767  156030  162164          MOV     TEMP1,REOTC     ;SET NUMBER OF UNITS
 4471  022616  000367  156022                  SWAB    TEMP1
 4472  022622  056767  156016  162152          BIS     TEMP1,REOTC     ;SET EOT CNTR
 4473  022630  012711  177777                  MOV     #-1,(R1)        ;TERMINATE SLAVE TABLE
 4474  022634  000207                          RTS     PC              ;RETURN TO SEQ
```

```
4475
4476                                    ;SUBROUTINE TO SELECT NRZ AUTO TEST MODE**********
4477
4478  022636  005067  156014    AMOD1:  CLR     BLCNTR          ;ASSURE BLOCK COUNTER IS 0
4479  022642  012701  000746            MOV     #UN1,R1         ;GET START OF SLAVE TABLE
4480  022646  052721  001700    AMOD1A: BIS     #1700,(R1)+     ;SET ALL SLAVE TO NRZ,NORM,ODD
4481  022652  005111                    COM     (R1)
4482  022654  001402                    BEQ     AMOD1B          ;IF FILLED ALL SLAVES: BR
4483  022656  005111                    COM     (R1)
4484  022660  000772                    BR      AMOD1A          ;ELSE DO ALL
4485  022662  005111            AMOD1B: COM     (R1)
4486  022664  004767  162126            JSR     PC,RWNDA        ;GO REWIND ALL AVAIL SLAVES
4487  022670  012767  000062  156044    MOV     #50.,ABLCNT     ;SET NUMBER OF BLOCKS FOR MODE 1
4488  022676  012767  174000  155652    MOV     #-4000,FMCNT    ;SET FC = 4000
4489  022704  012767  000100  155642    MOV     #100,RCNT       ;SET REC CNTR = 100
4490  022712  016767  156022  155630    MOV     ADRVN,DVN       ;SELECT DRIVE
4491  022720  012767  000001  155632    MOV     #1,PATRN        ;SELECT PATTERN 1
4492  022726  005067  155632            CLR     TMEX            ;ASSURE NO TMK
4493  022732  005067  155630            CLR     INTRF           ;ASSURE NORMAL READ
4494  022736  004767  160226            JSR     PC,STAUTO       ;GO DO AUTO MODE 1
4495  022742  012767  000010  155610    MOV     #10,PATRN       ;SELECT PATTERN 10
4496  022750  004767  160214            JSR     PC,STAUTO       ;GO DO PATTERN 10
4497  022754  012767  000014  155576    MOV     #14,PATRN       ;SELECT PATTERN 14
4498  022762  004767  160202            JSR     PC,STAUTO
4499  022766  005767  155660            TST     NRZOF           ;SEE IF NRZ ONLY
4500  022772  001411                    BEQ     AMOD1C          ;IF NOT: BR
4501  022774  012767  177777  155740    MOV     #-1,ABLCNT      ;FORCE TO EOT
4502  023002  012767  153624  155614    MOV     #153624,RANBAS
4503  023010  012767  032561  155610    MOV     #32561,RANSAV   ;RESET RANDOM DATA BASE
4504  023016  012767  177777  155534    MOV     #-1,PATRN       ;SELECT AUTO RANDOM DATA
4505  023024  004767  160140            JSR     PC,STAUTO
4506  023030  000207                    RTS     PC              ;RETURN TO SEQ
```

```
4507
4508                                        ;SUBROUTINE TO SELECT PE AUTO TEST MODE**********
4509
4510  023032  005767  155614        AMOD2:  TST     NRZOF           ;SEE IF NRZ ONLY
4511  023036  001057                        BNE     AMOD2X          ;IF SO: BR
4512  023040  005067  155612                CLR     BLCNTR          ;CLEAR BLOCK CNTR
4513  023044  012701  000746                MOV     #UN1,R1         ;SET START OF SLAVE TABLE
4514  023050  042711  001700        AMOD2A: BIC     #1700,(R1)      ;CLEAR NRZ
4515  023054  052721  002300                BIS     #2300,(R1)+     ;SET TO PE NORM, ODD
4516  023060  005111                        COM     (R1)            ;SEE IF END OF TABLE
4517  023062  001402                        BEQ     AMOD2B          ;IF SO: BR
4518  023064  005111                        COM     (R1)
4519  023066  000770                        BR      AMOD2A          ;CONTINUE
4520  023070  005111                AMOD2B: COM     (R1)
4521  023072  004767  161720                JSR     PC,RWNDA        ;REWIND ALL SLAVES
4522  023076  012767  000006  155636        MOV     #6,ABLCNT       ;SET AUTO BLOCK COUNT
4523  023104  012767  174000  155444        MOV     #-4000,FMCNT    ;SET FC = 4000
4524  023112  012767  000100  155434        MOV     #100,RCNT       ;SET REC CNTR TO 100
4525  023120  012767  000010  155432        MOV     #10,PATRN       ;SELECT PATTERN 10
4526  023126  004767  160036                JSR     PC,STAUTO       ;GO DO AUTO SEQ
4527  023132  012767  000014  155420        MOV     #14,PATRN       ;SELECT PATTERN 14
4528  023140  004767  160024                JSR     PC,STAUTO
4529  023144  012767  000015  155406        MOV     #15,PATRN       ;SELECT PATTERN 15
4530  023152  004767  160012                JSR     PC,STAUTO
4531  023156  012767  177777  155556        MOV     #-1,ABLCNT      ;FORCE TO END OF TAPE
4532  023164  012767  177777  155366        MOV     #-1,PATRN       ;SELECT AUTO RANDOM DATA
4533  023172  004767  157772                JSR     PC,STAUTO
4534  023176  000207                AMOD2X: RTS     PC              ;RETURN TO SEQ
4535
4536
```

```
                                ;*********************************************************
                                ;ERROR HEADER PRINT SUBROUTINE:
                                ;
                                ;THIS ROUTINE IS USED TO PRINT OUT A HEADER
                                ;WITH EACH ERROR MESSAGE.  THE PRINT IS IN TWO
                                ;LINES AND CONTAINS THE FOLLOWING INFORMATION.
                                ;LINE 1:  DRIVE NO. SLAVE NO. DENSITY PARITY FORMAT
                                ;LINE 2:  CURRENT BLOCK NUMBER, RECORD NUMBER IN
                                ;WHICH THE ERROR OCCURED PLUS THE TOTAL NUMBER
                                ;OF RECORDS IN THIS BLOCK, THE RECORD SIZE (NUMBER
                                ;OF CHARACTERS), AND THE ERROR TYPE (READ,WRITE, SPACE, ETC)
                                ;PLUS THE TAPE DIRECTION (FORWARD OR REVERSE).
                                ;ALL NUMBERS ARE IN OCTAL.
                                ;*********************************************************

023200  012704  025026    PAPRT:  MOV     #MSG12,R4
023204  004767  000712            JSR     PC,TTOUT        ;PRINT DRIVE HEADER
023210  016703  155334            MOV     DVN,R3
023214  004767  001054            JSR     PC,OCTP         ;PRINT DRIVE NUMBER
023220  012704  025012            MOV     #MSG11,R4
023224  004767  000672            JSR     PC,TTOUT        ;PRINT UNIT HEADER
023230  016703  155316            MOV     UDES,R3
023234  042703  177770            BIC     #177770,R3
023240  004767  001030            JSR     PC,OCTP         ;PRINT UNIT NUMBER
023244  012704  026267            MOV     #MSG60,R4
023250  004767  000646            JSR     PC,TTOUT        ;PRINT DENSITY TAG
023254  016703  155272            MOV     UDES,R3
023260  000303                    SWAB    R3
023262  042703  177770            BIC     #177770,R3
023266  004767  001002            JSR     PC,OCTP         ;PRINT DENSITY
023272  012704  026273            MOV     #MSG61,R4
023276  004767  000620            JSR     PC,TTOUT        ;PRINT PARITY TAG
023302  005003                    CLR     R3
023304  032767  000010  155240    BIT     #10,UDES
023312  001402                    BEQ     PAPRTO
023314  012703  000001            MOV     #1,R3
023320  004767  000750  PAPRTO:   JSR     PC,OCTP         ;PRINT PARITY
023324  012704  026277            MOV     #MSG62,R4
023330  004767  000566            JSR     PC,TTOUT        ;PRINT FORMAT TAG
023334  016703  155212            MOV     UDES,R3
023340  000241                    CLC
023342  006003                    ROR     R3
023344  006003                    ROR     R3
023346  006003                    ROR     R3              ;PONTION FORMAT
023350  006003                    ROR     R3
023352  042703  177760            BIC     #177760,R3
023356  004767  000712            JSR     PC,OCTP         ;PRINT FORMAT
023362  012704  024767            MOV     #MSG8,R4
023366  004767  000530            JSR     PC,TTOUT        ;PRINT PATRN TAG
023372  032777  000400  155206    BIT     #400,@SWR       ;SEE IF RANDOM DATA
023400  001406                    BEQ     PAPRTB          ;IF NOT: BR
023402  012767  000122  155230  PAPRTA:  MOV     #122,TOB
023410  004767  000616            JSR     PC,TOG          ;PRINT R
023414  000412                    BR      PAPRTD
023416  005767  155314  PAPRTB:  TST     ASEQF           ;SEE IF AUTO SEQ
```

```
 4593  023422  001403                        BEQ     PAPRTC          ;IF NOT: BR
 4594  023424  005767  155130                TST     PATRN           ;SEE IF AUTO RANDOM
 4595  023430  100764                        BMI     PAPRTA          ;IF SO: BR
 4596  023432  016703  155122        PAPRTC: MOV     PATRN,R3
 4597  023436  004767  000632                JSR     PC,OCTP         ;PRINT PATRN NUMBER
 4598  023442  012704  025044        PAPRTD: MOV     #MSG13,R4
 4599  023446  004767  000450                JSR     PC,TTOUT        ;PRINT BLOCK NO. HEADER
 4600  023452  016703  155200                MOV     BLCNTR,R3
 4601  023456  004767  000612                JSR     PC,OCTP         ;PRINT NUMBER
 4602  023462  012704  025052                MOV     #MSG14,R4
 4603  023466  004767  000430                JSR     PC,TTOUT        ;PRINT REC NO. HEADER
 4604  023472  010003                        MOV     R0,R3
 4605  023474  032767  000010  155172        BIT     #10,MTC1        ;SEE IF WRITE OPERATION
 4606  023502  001404                        BEQ     PAPRT1          ;IF SO: BR
 4607  023504  032767  010000  155050        BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 4608  023512  001016                        BNE     PAPRT3          ;IF SO: BR
 4609  023514  016703  155034        PAPRT1: MOV     RCNT,R3
 4610  023520  005767  155154                TST     TMFLG           ;SEE IF TAPE MARK TIME
 4611  023524  001010                        BNE     PAPRT2          ;IF SO: BR
 4612  023526  022767  012214  155130        CMP     #B1,RTRN
 4613  023534  001003                        BNE     PAPRTY          ;IF NOT BACK SPACE: BR
 4614  023536  005767  155152                TST     RTYFL
 4615  023542  001402                        BEQ     PAPRT3          ;IF NOT RETRY: BR
 4616  023544  160003        PAPRTY: SUB     R0,R3           ;GET RECORD NUMBER
 4617  023546  005203        PAPRT2: INC     R3
 4618  023550  004767  000520        PAPRT3: JSR     PC,OCTP         ;PRINT RECORD NUMBER
 4619  023554  012767  000055  155056        MOV     #55,TOB         ;LOAD DASH (-)
 4620  023562  004767  000444                JSR     PC,TOG          ;PRINT DASH (-)
 4621  023566  016703  154762                MOV     RCNT,R3
 4622  023572  004767  000476                JSR     PC,OCTP         ;PRINT RECORD COUNT
 4623  023576  012704  024762                MOV     #MSG7,R4
 4624  023602  004767  000314                JSR     PC,TTOUT        ;PRINT RECORD SIZE HEADER
 4625  023606  016703  154744                MOV     FMCNT,R3        ;GET CHARACTER COUNT
 4626  023612  005303                        DEC     R3
 4627  023614  005103                        COM     R3              ;REMOVE TWOS COMPLEMENT
 4628  023616  004767  000452                JSR     PC,OCTP         ;PRINT RECORD SIZE
 4629  023622  012767  000001  155036        MOV     #1,HDRFL        ;SET HEADER FLAG
 4630  023630  000207                        RTS     PC              ;RETURN
 4631
```

```
 4632
 4633                                   ;*********************************************************
 4634                                   ;RANDOM NUMBER GENERATOR SUBROUTINE:
 4635                                   ;
 4636                                   ;THIS SUBROUTINE IS USED TO GENERATE THE RANDOM
 4637                                   ;NUMBERS REQUIRED FOR USE AS RANDOM DATA,
 4638                                   ;RECORD COUNT, AND CHARACTER COUNT.
 4639                                   ;*********************************************************
 4640
 4641  023632  066767  154770  154764  RANG:   ADD     RANSAV,RANBAS
 4642  023640  066767  154760  154760          ADD     RANBAS,RANSAV   ;GET NEW NUMBER
 4643  023646  026701  154754                  CMP     RANSAV,R1       ;SEE IF NUMBER TOO BIG
 4644  023652  101367                          BHI     RANG            ;IF SO: BR
 4645  023654  020267  154746                  CMP     R2,RANSAV       ;SEE IF NUMBER TOO SMALL
 4646  023660  101364                          BHI     RANG            ;IF SO: BR
 4647  023662  000207                          RTS     PC              ;EXIT
 4648
```

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

023664  005067  154754        TTR:    CLR     TEMP1           ;CLEAR FIRST CHARACTER FLAG
023670  005000                        CLR     R0
023672  004767  000152        TTR0:   JSR     PC,TTIN         ;GO READ CHARACTER
023676  122767  000215  154736        CMPB    #215,TIB        ;SEE IF CR
023704  001005                        BNE     TTR1            ;IF NOT: BR
023706  005767  154732                TST     TEMP1           ;SEE IF FIRST CHARACTER
023712  001446                        BEQ     TTR5            ;IF SO: BR
023714  000167  000066                JMP     TTR2            ;ELSE GO LOAD VALUE
023720  122767  000260  154714  TTR1: CMPB    #260,TIB        ;SEE IF CHAR IS LESS THAN 0
023726  101402                        BLOS    TTR1A           ;IF NOT: BR
023730  000167  000076                JMP     TINER           ;ELSE GO TO ERROR
023734  122767  000270  154700  TTR1A:CMPB    #270,TIB        ;SEE IF CHAR IS GREATER THAN 7
023742  101002                        BHI     TTR1B           ;IF NOT: BR
023744  000167  000062                JMP     TINER           ;ELSE GO TO ERROR
023750  005267  154670        TTR1B:  INC     TEMP1           ;SET FIRST CHARACTER FLAG
023754  000241                        CLC
023756  006100                        ROL     R0
023760  000241                        CLC
023762  006100                        ROL     R0              ;SHIFT 3 LEFT
023764  000241                        CLC
023766  006100                        ROL     R0
023770  042767  177770  154644        BIC     #177770,TIB     ;STRIP ASCII
023776  056700  154640                BIS     TIB,R0          ;LOAD CHARACTER
024002  005301                        DEC     R1              ;SEE IF DONE
024004  001332                        BNE     TTR0            ;IF NOT: BR
024006  020002                TTR2:   CMP     R0,R2           ;SEE IF EXCEEDED MAXIMUM LIMIT
024010  101402                        BLOS    TTR3            ;IF NOT: BR
024012  000167  000014                JMP     TINER           ;ELSE GO TO ERROR
024016  020300                TTR3:   CMP     R3,R0           ;SEE IF BELOW MINIMUM LIMIT
024020  101402                        BLOS    TTR4            ;IF NOT: BR
024022  000167  000004                JMP     TINER           ;ELSE GO TO ERROR
024026  010015                TTR4:   MOV     R0,(R5)         ;LOAD VALUE
024030  000207                TTR5:   RTS     PC              ;EXIT
024032  012704  026001        TINER:  MOV     #MSG43,R4
024036  004767  000060                JSR     PC,TTOUT        ;PRINT?
024042  162716  000020                SUB     #20,(SP)        ;RESET SP TO START OF VALUE ROUTINE
024046  000207                        RTS     PC              ;REDO VALUE ENTRY
```

```
 4703
 4704                                        ;TTY READ SUBROUTINE**********
 4705
 4706  024050  005077  154534          TTIN:  CLR     @TKS
 4707  024054  005077  154532                 CLR     @TKB
 4708  024060  005067  154556                 CLR     TIB
 4709  024064  005277  154520                 INC     @TKS
 4710  024070  105777  154514          TTIN1: TSTB    @TKS
 4711  024074  100375                        BPL     TTIN1
 4712  024076  017767  154510  154536       MOV     @TKB,TIB
 4713  024104  105777  154504          TTIN2: TSTB    @TPS
 4714  024110  100375                        BPL     TTIN2
 4715  024112  116777  154524  154476       MOVB    TIB,@TPB
 4716  024120  000207                        RTS     PC
 4717
 4718                                        ;TTY OUTPUT SUBROUTINE************
 4719
 4720  024122  112467  154512          TTOUT: MOVB    (R4)+,TOB
 4721  024126  122767  000043  154504       CMPB    #43,TOB
 4722  024134  001444                        BEQ     TEX
 4723  024136  122767  000045  154474       CMPB    #45,TOB
 4724  024144  001407                        BEQ     TCRLF
 4725  024146  122767  000041  154464       CMPB    #41,TOB
 4726  024154  001435                        BEQ     TBELL           ;DO BELL
 4727  024156  004767  000050                JSR     PC,TOG
 4728  024162  000757                        BR      TTOUT
 4729  024164  112767  000015  154446  TCRLF: MOVB    #15,TOB
 4730  024172  004767  000034                JSR     PC,TOG
 4731  024176  012703  000006                MOV     #6,R3
 4732  024202  005067  154432          TCRLFA: CLR    TOB
 4733  024206  004767  000020                JSR     PC,TOG
 4734  024212  005303                        DEC     R3
 4735  024214  001372                        BNE     TCRLFA          ;DO FILLERS
 4736  024216  112767  000012  154414       MOVB    #12,TOB
 4737  024224  004767  000002                JSR     PC,TOG
 4738  024230  000734                        BR      TTOUT
 4739  024232  105777  154356          TOG:   TSTB    @TPS
 4740  024236  100375                        BPL     TOG
 4741  024240  116777  154374  154350       MOVB    TOB,@TPB
 4742  024246  000207                TEX:    RTS     PC
 4743  024250  012703  000002          TBELL: MOV     #2,R3
 4744  024254  012767  000007  154356  TBELA: MOV     #7,TOB
 4745  024262  004767  177744                JSR     PC,TOG
 4746  024266  005303                        DEC     R3
 4747  024270  001371                        BNE     TBELA
 4748  024272  000713                        BR      TTOUT
 4749
 4750
```

```
 4751                                          ;OCTAL OUTPUT SUBROUTINE***********
 4752
 4753  024274  005067  000212          OCTP:   CLR     OFL             ;CLEAR FLAG FOR LEADING ZERO
 4754  024300  010304                          MOV     R3,R4           ;SEE IF NUMBER IS ZERO
 4755  024302  001004                          BNE     OCTP0           ;IF NOT ZERO: BR
 4756  024304  004767  000162                  JSR     PC,OCTPG1       ;ELSE PRINT ZERO
 4757  024310  000167  000120                  JMP     OCTP3           ;SPACE AND EXIT
 4758  024314  032704  100000          OCTP0:  BIT     #100000,R4      ;SEE IF MSD = 1
 4759  024320  001406                          BEQ     OCTP1           ;IF NOT: BR
 4760  024322  012704  000001                  MOV     #1,R4
 4761  024326  004767  000116                  JSR     PC,OCTPG        ;PRINT 1
 4762  024332  000167  000006                  JMP     OCTP2
 4763  024336  005004                  OCTP1:  CLR     R4
 4764  024340  004767  000104                  JSR     PC,OCTPG        ;PRINT 0
 4765  024344  010304                  OCTP2:  MOV     R3,R4
 4766  024346  006004                          ROR     R4
 4767  024350  006004                          ROR     R4
 4768  024352  006004                          ROR     R4              ;POSITION DIGIT
 4769  024354  006004                          ROR     R4
 4770  024356  000304                          SWAB    R4
 4771  024360  004767  000064                  JSR     PC,OCTPG        ;PRINT DIGIT 2
 4772  024364  010304                          MOV     R3,R4
 4773  024366  006004                          ROR     R4
 4774  024370  000304                          SWAB    R4
 4775  024372  004767  000052                  JSR     PC,OCTPG        ;PRINT DIGIT 3
 4776  024376  010304                          MOV     R3,R4
 4777  024400  006104                          ROL     R4
 4778  024402  006104                          ROL     R4
 4779  024404  000304                          SWAB    R4
 4780  024406  004767  000036                  JSR     PC,OCTPG        ;PRINT DIGIT 4
 4781  024412  010304                          MOV     R3,R4
 4782  024414  006004                          ROR     R4
 4783  024416  006004                          ROR     R4
 4784  024420  006004                          ROR     R4
 4785  024422  004767  000022                  JSR     PC,OCTPG
 4786  024426  010304                          MOV     R3,R4
 4787  024430  004767  000014                  JSR     PC,OCTPG        ;PRINT DIGIT 5
 4788  024434  012767  000240  154176  OCTP3:  MOV     #240,TOB
 4789  024442  004767  177564                  JSR     PC,TOG          ;PRINT SPACE
 4790  024446  000207                          RTS     PC              ;EXIT
 4791  024450  042704  177770          OCTPG:  BIC     #177770,R4
 4792  024454  001004                          BNE     OCTPG0
 4793  024456  005767  000030                  TST     OFL
 4794  024462  001001                          BNE     OCTPG0
 4795  024464  000207                          RTS     PC
 4796  024466  005267  000020          OCTPG0: INC     OFL
 4797  024472  052704  000260          OCTPG1: BIS     #260,R4
 4798  024476  010467  154136                  MOV     R4,TOB
 4799  024502  004767  177524                  JSR     PC,TOG
 4800  024506  010304                          MOV     R3,R4
 4801  024510  000207                          RTS     PC
 4802  024512  000000                  OFL:    0                       ;FIRST CHAR FLAG
 4803
```

```
4804
4805                                        ;DATA CHARACTER OUTPUT SUBROUTINE***********
4806
4807  024514  005067  154120        DOUT:   CLR     TOB
4808  024520  012704  000010                MOV     #10,R4          ;SET NUMBER TO PRINT
4809  024524  110367  154110                MOVB    R3,TOB
4810  024530  105777  154060        DOUT1:  TSTB    @TPS
4811  024534  100375                        BPL     DOUT1
4812  024536  132767  000200  154074        BITB    #200,TOB
4813  024544  001404                        BEQ     DOUT2
4814  024546  012777  000061  154042        MOV     #061,@TPB
4815  024554  000403                        BR      DOUT3
4816  024556  012777  000060  154032  DOUT2:  MOV     #060,@TPB
4817  024564  006167  154050        DOUT3:  ROL     TOB
4818  024570  005304                        DEC     R4
4819  024572  001356                        BNE     DOUT1
4820  024574  000207                        RTS     PC
4821  024576  016703  154046        DOUTD:  MOV     TEMP3,R3
4822  024602  000303                        SWAB    R3
4823  024604  004767  177704                JSR     PC,DOUT
4824  024610  016703  154034                MOV     TEMP3,R3
4825  024614  004767  177674                JSR     PC,DOUT
4826  024620  000207                        RTS     PC
4827
4828                                        ;TU45 SERIAL NUMBER PRINT SUBROUTINE*******
4829
4830  024622  010304                SNPT:   MOV     R3,R4
4831  024624  000304                        SWAB    R4
4832  024626  006004                        ROR     R4
4833  024630  006004                        ROR     R4
4834  024632  006004                        ROR     R4
4835  024634  006004                        ROR     R4
4836  024636  004767  000036                JSR     PC,SNPG         ;PRINT FIRST DIGIT
4837  024642  010304                        MOV     R3,R4
4838  024644  000304                        SWAB    R4
4839  024646  004767  000026                JSR     PC,SNPG         ;PRINT SECOND DIGIT
4840  024652  010304                        MOV     R3,R4
4841  024654  006004                        ROR     R4
4842  024656  006004                        ROR     R4
4843  024660  006004                        ROR     R4
4844  024662  006004                        ROR     R4
4845  024664  004767  000010                JSR     PC,SNPG         ;PRINT THIRD DIGIT
4846  024670  010304                        MOV     R3,R4
4847  024672  004767  000002                JSR     PC,SNPG         ;PRINT FOURTH DIGIT
4848  024676  000207                        RTS     PC              ;EXIT
4849  024700  012767  000260  153732  SNPG:   MOV     #260,TOB        ;SET NUMBER BASE
4850  024706  042704  177760                BIC     #177760,R4      ;MASK NUMBER
4851  024712  050467  153722                BIS     R4,TOB          ;BUILD DIGIT
4852  024716  004767  177310                JSR     PC,TOG          ;GO TYPE
4853  024722  000207                        RTS     PC              ;RETURN
4854
```

```
                                        ;ERROR MESSAGES**********
024724  042052  020105     043  MSG1:   .ASCII  /*DE #/
024731     045  035507  021440  MSG2:   .ASCII  /%G; #/
024736  041045  020073     043  MSG3:   .ASCII  /%B; #/
024743     045  047103  021440  MSG4:   .ASCII  /%CN #/
024750  053452  020105     043  MSG5:   .ASCII  /*WE #/
024755     052  042522  021440  MSG6:   .ASCII  /*RE #/
024762  051052  020123     043  MSG7:   .ASCII  /*RS #/
024767     052  040520  051124  MSG8:   .ASCII  /*PATRN #/
024774  020116     043
024777     040  047123  020072  MSG9:   .ASCII  / SN: #/
025004     043
025005     052  042523  021440  MSG10:  .ASCII  /*SE #/
025012  051452  040514  042526  MSG11:  .ASCII  /*SLAVE NO. #/
025020  047040  027117  021440
025026  022445  042045  044522  MSG12:  .ASCII  /%%%DRIVE NO. #/
025034  042526  047040  027117
025042  021440
025044  025045  047102  021440  MSG13:  .ASCII  /%*BN #/
025052  051052  020116     043  MSG14:  .ASCII  /*RN #/
025057     045  020041  020040  MSG15:  .ASCII  /%!          BAD RECORD%%#/
025064  020040  020040  020040
025072  041040  042101  051040
025100  041505  051117  022504
025106  021445
025110  043040     043          MSG16:  .ASCII  / F#/
025113     040  021522          MSG17:  .ASCII  / R#/
025116  020041  047505  020124  MSG20:  .ASCII  /! EOT NO: #/
025124  047516  020072     043
025131     045  044441  046114  MSG22:  .ASCII  /%!ILLEGAL BOT: HALT%%%#/
025136  043505  046101  041040
025144  052117  020072  040510
025152  052114  022445  021445
025160  041445  030523  021440  MSG23:  .ASCII  /%CS1 #/
025166  053445  020103     043  MSG23A: .ASCII  /%WC #/
```

```
                                                        J  9
 4911
 4912  025173     045  040502  021440  MSG23B: .ASCII  /%BA #/
 4913
 4914  025200  043045  020103     043  MSG23C: .ASCII  /%FC #/
 4915
 4916  025205     045  051503  020062  MSG23D: .ASCII  /%CS2 #/
 4917  025212     043
 4918
 4919  025213     045  051504  021440  MSG23E: .ASCII  /%DS #/
 4920
 4921  025220  042445  020122     043  MSG23F: .ASCII  /%ER #/
 4922
 4923  025225     045  051501  021440  MSG23G: .ASCII  /%AS #/
 4924
 4925  025232  041445  020113     043  MSG23H: .ASCII  /%CK #/
 4926
 4927  025237     045  041104  021440  MSG23I: .ASCII  /%DB #/
 4928
 4929  025244  046445  020122     043  MSG23J: .ASCII  /%MR #/
 4930
 4931  025251     045  052104  021440  MSG23K: .ASCII  /%DT #/
 4932
 4933  025256  052045  020103     043  MSG23L: .ASCII  /%TC #/
 4934
 4935  025263     045  047123  021440  MSG23M: .ASCII  /%SN #/
 4936
 4937  025270  020445  047516  044440  MSG24:  .ASCII  /%!NO INTERRUPT%#/
 4938  025276  052116  051105  052522
 4939  025304  052120  021445
 4940
 4941  025310  020445  047516  046440  MSG25:  .ASCII  /%!NO MOL: HALT%#/
 4942  025316  046117  020072  040510
 4943  025324  052114  021445
 4944
 4945  025330  042045  047522  051520  MSG26:  .ASCII  /%DROPS: #/
 4946  025336  020072     043
 4947
 4948  025341     045  044520  045503  MSG27:  .ASCII  /%PICKS: #/
 4949  025346  035123  021440
 4950
 4951  025352  021445                  MSG28:  .ASCII  /%#/
 4952  025354  022445  052524  032464  MSG30:  .ASCII  /%%TU45 AUTO SEQUENCE (CZTUIA0)%/
 4953  025362  040440  052125  020117
 4954  025370  042523  052521  047105
 4955  025376  042503  024040  055103
 4956  025404  052524  040511  024460
 4957  025412     045
 4958  025413     115  045501  020105          .ASCII  /MAKE ENTRIES IN OCTAL%#/
 4959  025420  047105  051124  042511
 4960  025426  020123  047111  047440
 4961  025434  052103  046101  021445
 4962  025442  022445  052524  032464  MSG31:  .ASCII  /%%TU45 DATA RELIABILITY TEST (CZTUIA0)%MAKE ENTRIES IN OCTAL%#/
 4963  025450  042040  052101  020101
 4964  025456  042522  044514  041101
 4965  025464  046111  052111  020131
 4966  025472  042524  052123  024040
```

```
 4967  025500  055103  052524  040511
 4968  025506  024460  046445  045501
 4969  025514  020105  047105  051124
 4970  025522  042511  020123  047111
 4971  025530  047440  052103  046101
 4972  025536  021445
 4973
 4974  025540  051445  040514  042526  MSG32:  .ASCII  /%SLAVE NUMBER = #/
 4975  025546  047040  046525  042502
 4976  025554  020122  020075     043
 4977
 4978  025561     045  042504  051516  MSG33:  .ASCII  /%DENSITY = #/
 4979  025566  052111  020131  020075
 4980  025574     043
 4981
 4982  025575     045  040520  044522  MSG34:  .ASCII  /%PARITY = #/
 4983  025602  054524  036440  021440
 4984
 4985  025610  051045  041505  051117  MSG35:  .ASCII  /%RECORD COUNT = #/
 4986  025616  020104  047503  047125
 4987  025624  020124  020075     043
 4988
 4989  025631     045  044103  051101  MSG36:  .ASCII  /%CHARACTER COUNT = #/
 4990  025636  041501  042524  020122
 4991  025644  047503  047125  020124
 4992  025652  020075     043
 4993
 4994  025655     045  040520  052124  MSG37:  .ASCII  /%PATTERN NUMBER = #/
 4995  025662  051105  020116  052516
 4996  025670  041115  051105  036440
 4997  025676  021440
 4998  025700  051445  047111  046107  MSG38:  .ASCII  /%SINGLE PASS = #/
 4999  025706  020105  040520  051523
 5000  025714  036440  021440
 5001  025720  022445  047105  042524  MSG40:  .ASCII  /%%ENTER STALLS%READ = #/
 5002  025726  020122  052123  046101
 5003  025734  051514  051045  040505
 5004  025742  020104  020075     043
 5005
 5006  025747     045  051127  052111  MSG41:  .ASCII  /%WRITE = #/
 5007  025754  020105  020075     043
 5008
 5009  025761     045  052524  047122  MSG42:  .ASCII  /%TURN AROUND = #/
 5010  025766  040440  047522  047125
 5011  025774  020104  020075     043
 5012
 5013  026001     045  022477     043  MSG43:  .ASCII  /%?%#/
 5014
 5015  026005     045  047105  042524  MSG44:  .ASCII  /%ENTER YOZZLE STALL = #/
 5016  026012  020122  047531  055132
 5017  026020  042514  051440  040524
 5018  026026  046114  036440  021440
 5019
 5020  026034  042445  051122  040440  MSG45:  .ASCII  /%ERR AMT #/
 5021  026042  052115  021440
 5022
```

```
 5023  026046  043045  020103     043  MSG46:  .ASCII  /%FC #/
 5024
 5025  026053     045  040503  021440  MSG47:  .ASCII  /%CA #/
 5026
 5027  026060  020445  047516  041040  MSG48:  .ASCII  /%!NO BOT ON REWIND: HALT%%#/
 5028  026066  052117  047440  020116
 5029  026074  042522  044527  042116
 5030  026102  020072  040510  052114
 5031  026110  022445     043
 5032
 5033  026113     040  047516  020124  MSG49:  .ASCII  / NOT AVAIL #/
 5034  026120  053101  044501  020114
 5035  026126     043
 5036  026127     040  046111  042514  MSG50:  .ASCII  / ILLEGAL DRIVE TYPE #/
 5037  026134  040507  020114  051104
 5038  026142  053111  020105  054524
 5039  026150  042520  021440
 5040  026154  042045  044522  042526  MSG52:  .ASCII  /%DRIVE NUMBER = #/
 5041  026162  047040  046525  042502
 5042  026170  020122  020075     043
 5043
 5044  026175     045  047506  046522  MSG53:  .ASCII  /%FORMAT = #/
 5045  026202  052101  036440  021440
 5046
 5047  026210  053452  020105  046524  MSG54:  .ASCII  /*WE TM#/
 5048  026216     043
 5049
 5050  026217     052  042523  052040  MSG55:  .ASCII  /*SE TM#/
 5051  026224  021515
 5052
 5053  026226  052040  021515          MSG56:  .ASCII  / TM#/
 5054
 5055  026232  047040  047117  042455  MSG57:  .ASCII  / NON-EXIST SLAVE#/
 5056  026240  044530  052123  051440
 5057  026246  040514  042526     043
 5058  026253     045  051103  020103  MSG58:  .ASCII  /%CRC #/
 5059  026260     043
 5060  026261     045  051114  020103  MSG59:  .ASCII  /%LRC #/
 5061  026266     043
 5062  026267     052  020104     043  MSG60:  .ASCII  /*D #/
 5063  026273     052  020120     043  MSG61:  .ASCII  /*P #/
 5064  026277     052  020106     043  MSG62:  .ASCII  /*F #/
 5065
 5066  026303     045  047452  044522  MSG64:  .ASCII  /%*ORIGINAL ERROR*#/
 5067  026310  044507  040516  020114
 5068  026316  051105  047522  025122
 5069  026324     043
 5070
 5071  026325     045  042522  051124  MSG65:  .ASCII  /%RETRY: #/
 5072  026332  035131  021440
 5073
 5074  026336  020452  042523  051040  MSG66:  .ASCII  /*!SE RTRY #/
 5075  026344  051124  020131     043
 5076
 5077  026351     052  042441  040522  MSG67:  .ASCII  /*!ERASE#/
 5078  026356  042523     043
```

```
 5079
 5080  026361    045  042522  042522  MSG68:  .ASCII  /%REREV: #/
 5081  026366  035126  021440
 5082  026372  052045  050101  020105  MSG69:  .ASCII  /%TAPE MARK = #/
 5083  026400  040515  045522  036440
 5084  026406  021440
 5085
 5086  026410  020445  047516  042040  MSG70:  .ASCII  /%!NO DRY FROM REWIND: HALT%#/
 5087  026416  054522  043040  047522
 5088  026424  020115  042522  044527
 5089  026432  042116  020072  040510
 5090  026440  052114  021445
 5091  026444  047040  047117  042455  MSG71:  .ASCII  / NON-EXIST DRIVE#/
 5092  026452  044530  052123  042040
 5093  026460  044522  042526    043
 5094  026465    045  042522  053506  MSG72:  .ASCII  /%REFWD: #/
 5095  026472  035104  021440
 5096  026476  053445  042524  051122  MSG73:  .ASCII  /%WTERR: #/
 5097  026504  020072    043
 5098  026507    045  042522  044507  MSG74:  .ASCII  /%REGISTER START = #/
 5099  026514  052123  051105  051440
 5100  026522  040524  052122  036440
 5101  026530  021440
 5102  026532  053045  041505  047524  MSG75:  .ASCII  /%VECTOR = #/
 5103  026540  020122  020075    043
 5104  026545    045  042504  042522  MSG76:  .ASCII  /%DEREV: #/
 5105  026552  035126  021440
 5106  026556  042045  043105  042127  MSG77:  .ASCII  /%DEFWD: #/
 5107  026564  020072    043
 5108  026567    045  047041  047117  MSG78:  .ASCII  /%!NON-RETRYABLE WRITE ERROR: ER #/
 5109  026574  051055  052105  054522
 5110  026602  041101  042514  053440
 5111  026610  044522  042524  042440
 5112  026616  051122  051117  020072
 5113  026624  051105  021440
 5114  026630  020445  047516  026516  MSG79:  .ASCII  /%!NON-RETRYABLE READ ERROR: ER #/
 5115  026636  042522  051124  040531
 5116  026644  046102  020105  042522
 5117  026652  042101  042440  051122
 5118  026660  051117  020072  051105
 5119  026666  021440
 5120  026670  020445  042441  042116  MSG100: .ASCII  /%!!END OF PASS %#/
 5121  026676  047440  020106  040520
 5122  026704  051523  022440    043
 5123  026711    045  025045  025052  MSG101: .ASCII  /%%**********************/
 5124  026716  025052  025052  025052
 5125  026724  025052  025052  025052
 5126  026732  025052  025052    052
 5127  026737    052  046524  031060  MSG102: .ASCII  /*TM02 #/
 5128  026744  021440
 5129  026746  051452  040514  042526  MSG103: .ASCII  /*SLAVES #/
 5130  026754  020123    043
 5131  026757    045  052501  047524  MSG104: .ASCII  /%AUTO CONT: #/
 5132  026764  041440  047117  035124
 5133  026772  021440
 5134  026774  051045  041505  053117  MSG105: .ASCII  /%RECOVERED#/
```

```
                                                    N  9
 5135  027002  051105  042105     043
 5136  027007     052  020441  040502  MSG106: .ASCII  /*!!BAD TAPE OVERFLOW#/
 5137  027014  020104  040524  042520
 5138  027022  047440  042526  043122
 5139  027030  047514  021527
 5140  027034  051045  053505  047111  MSG16A: .ASCII  /%REWIND TAPE; RESTART AT BLOCK ONE#/
 5141  027042  020104  040524  042520
 5142  027050  020073  042522  052123
 5143  027056  051101  020124  052101
 5144  027064  041040  047514  045503
 5145  027072  047440  042516     043
 5146  027077     045  020441  047125  MSG107: .ASCII  /%!!UNRECOVERABLE BAD SPOT/
 5147  027104  042522  047503  042526
 5148  027112  040522  046102  020105
 5149  027120  040502  020104  050123
 5150  027126  052117
 5151  027130  041045  042101  051040          .ASCII  /%BAD RECORD LEFT ON TAPE%#/
 5152  027136  041505  051117  020104
 5153  027144  042514  052106  047440
 5154  027152  020116  040524  042520
 5155  027160  021445
 5156  027162  047045  055122  047440  MSG108: .ASCII  /%NRZ ONLY: #/
 5157  027170  046116  035131  021440
 5158  027176  020452  050041  051517  MSG109: .ASCII  /*!!POSITION LOST IN RETRY#/
 5159  027204  052111  047511  020116
 5160  027212  047514  052123  044440
 5161  027220  020116  042522  051124
 5162  027226  021531
 5163  027230  051445  051525  042520  MSG110: .ASCII  /%SUSPECT BAD TAPE#/
 5164  027236  052103  041040  042101
 5165  027244  052040  050101  021505
 5166  027252  051045  050105  040505  MSG111: .ASCII  /%REPEAT: #/
 5167  027260  035124  021440
 5168  027264  041040  042101  052040  MSG112: .ASCII  / BAD TAPE SPOTS%#/
 5169  027272  050101  020105  050123
 5170  027300  052117  022523     043
 5171
 5172  027305     045  051440  043117  MSG113: .ASCII  /% SOFT: #/
 5173  027312  035124  021440
 5174
 5175  027316  020045  040510  042122  MSG114: .ASCII  /% HARD: #/
 5176  027324  020072     043
 5177
 5178  027327     045  020441  040510  MSG115: .ASCII  /%!!HARD READ ERROR#/
 5179  027334  042122  051040  040505
 5180  027342  020104  051105  047522
 5181  027350  021522
 5182  027352  020445  047125  052111  MSG116: .ASCII  /%!UNIT IS REWINDING: TEST WILL START AT BOT#/
 5183  027360  044440  020123  042522
 5184  027366  044527  042116  047111
 5185  027374  035107  052040  051505
 5186  027402  020124  044527  046114
 5187  027410  051440  040524  052122
 5188  027416  040440  020124  047502
 5189  027424  021524
 5190
```

```
 5191                                    .EVEN
 5192  027426  000000            WDATA:  0                       ;WRITE BUFFER
 5193
 5194          033434                    .=.+4004
 5195  033434  000000            RDATA:  0                       ;READ BUFFER
 5196
 5197          000001                    .END
```

ABLCNT	000742	BTOVX	007240	DATA5	003004	DERR3	016426	DRP6	001022
ACTLRC	021350	BTOV0	007102	DATA6	003006	DERR4	016430	DRP7	001024
ADRVN	000740	BTOV1	007112	DATA7	003010	DERR4A	016572	DRP8	001026
AMOD1	022636	BTOV2	007214	DATBL	002770	DERR4B	016640	DRVER	021340
AMOD1A	022646	BTOV3	007232	DATER1	001130	DERR5	016676	DS	000522
AMOD1B	022662	BTPRT	007242	DATR	015142	DERR6	016710	DSUP	014176
AMOD1C	023016	BTPRT1	007320	DATRO	015160	DFX	016220	DS0	014206
AMOD2	023032	BTPT	000732	DATO	014414	DF0	016112	DS0A	014270
AMOD2A	023050	BTSTF	000730	DATOA	014444	DF0A	016006	DS0B	014266
AMOD2B	023070	BTUR	007324	DATOB	014452	DF0A0	016030	DS0C	014230
AMOD2X	023176	BT00	001610	DATOC	014522	DF0A1	016044	DS1	014320
APATS	015122	BT01	001714	DATOD	014530	DF0A2	016060	DS2	014340
AS	000526	BT02	002020	DATOE	014540	DF0A3	016074	DS2A	014350
ASEQ	022234	BT03	002124	DATOF	014554	DF0A4	016100	DS3	014354
ASEQCF	000744	BT04	002230	DAT1	014566	DF0B	015746	DS4	014364
ASEQF	000736	BT05	002334	DAT1A	014572	DF0B0	015770	DT	000536
ASEQX	022442	BT06	002440	DAT1B	014576	DF0C	015726	DVN	000550
ASEQXX	022462	BT07	002544	DAT10	014726	DF0C0	015736	DOFL	014564
ASEQ0	022324	B0	012146	DAT10A	014740	DF0D	015712	EMADDR	000654
ASEQ1	022330	B1	012214	DAT11	014760	DF0E	015704	EOTC0	002650
ASEQ2	022400	B2	012224	DAT11A	014766	DF0F	015676	EOTREC	000662
ASEQ3	022414	CADER	021332	DAT12	015002	DF1	016124	ER	000524
ASEQ4	022424	CC	000530	DAT12A	015012	DF2	016134	ERCHK	017514
BA	000514	CCNTR	012250	DAT13	015026	DF3	016152	ERPT	020352
BAER	021334	CLLAST	015314	DAT14	015036	DF4	016214	ERPTG	020406
BBC	000660	CLP	015424	DAT14A	015050	DOUT	024514	ERPTG1	020454
BCNT	000712	CLPE	015450	DAT15	015070	DOUTD	024576	ERPTT	020372
BDPP	000720	CLP0	015460	DAT15A	015074	DOUT1	024530	ERPT0	020470
BD00	001410	CLP1	015472	DAT15B	015104	DOUT2	024556	ERPT1	020542
BD10	001430	CLP2	015532	DAT2	014610	DOUT3	024564	ERPT2	020610
BD20	001450	CLP3	015544	DAT3	014616	DPC	017072	ERPT3	020656
BD30	001470	CL0	015230	DAT3A	014624	DPCG	017100	ERPT4	020726
BD40	001510	CL1	015256	DAT3B	014630	DPC0	017106	ERPT5	020754
BD50	001530	CL2	015300	DAT4	014644	DPC0A	017150	ERPT5A	021050
BD60	001550	CL3	015366	DAT5	014656	DPC1	017156	ERPT6	021120
BD70	001570	CONER	021336	DAT6	014666	DPC1A	017204	ERPT7	021176
BKRT	012170	CRCER	021346	DAT7	014676	DPC2	017246	ERPX	021234
BKSP	012012	CRCLRC	015212	DAT7A	014712	DPC2A	017210	ERPX0	021200
BKTM	012074	CRCSV	021354	DB	000532	DPC2B	017232	ERPX1	021276
BKTM0	012136	CS	000520	DCHK	015606	DPC3	017300	ERPX2	021330
BLCNTR	000656	C1	000510	DCHK0	015636	DPPRT	017346	ERSAV	000724
BPKP	000722	C2	000542	DEREV1	001170	DPPRTX	017512	ERTFL	000734
BP00	001210	DATA0	002772	DEREX	016714	DPPRT0	017420	ER0	017530
BP10	001230	DATA1	002774	DEREX1	016750	DPPRT1	017446	ER0A	017600
BP20	001250	DATA10	003012	DERFL	000706	DPPRT2	017464	ER0B	017546
BP30	001270	DATA11	003014	DERR	016222	DROP	017062	ER1	017606
BP40	001310	DATA12	003016	DERR0	016236	DRPK	017050	ER10	020314
BP50	001330	DATA13	003020	DERR0A	016270	DRPKF	016762	ER2	017612
BP60	001350	DATA14	003022	DERR0B	016324	DRP1	001010	ER2A	017656
BP70	001370	DATA15	003024	DERR0C	016354	DRP2	001012	ER2A0	017626
BTADDR	001030	DATA2	002776	DERR0D	016356	DRP3	001014	ER2A1	017646
BTFLG	000726	DATA3	003000	DERR1	016406	DRP4	001016	ER2B	017674
BTOV	007062	DATA4	003002	DERR2	016410	DRP5	001020	ER2C	017720

```
ER2D    017736      MSG115  027327      MSG50   026127      PAPRT0  023320      RDRT4   011052
ER2E    017764      MSG116  027352      MSG52   026154      PAPRT1  023514      RDRT5   011054
ER3     017772      MSG12   025026      MSG53   026175      PAPRT2  023546      RDRT5A  011116
ER3A    020034      MSG13   025044      MSG54   026210      PAPRT3  023550      RDRT5B  011126
ER3A1   020072      MSG14   025052      MSG55   026217      PARCNT  015604      RDRT6   011162
ER3B    020102      MSG15   025057      MSG56   026226      PARS    014412      RDRT7   011230
ER4     020106      MSG16   025110      MSG57   026232      PATRN   000560      RDX     010644
ER4A    020154      MSG16A  027034      MSG58   026253      PATS    014410      RD0     007766
ER4A1   020144      MSG17   025113      MSG59   026261      PFLG    000672      RD1     010034
ER6     020160      MSG2    024731      MSG6    024755      PICK    017302      RD1A    010050
ER6A    020244      MSG20   025116      MSG60   026267      PIK1    000770      RD1B    010056
ER7     020276      MSG22   025131      MSG61   026273      PIK2    000772      RD1D    010064
EXCRC   015600      MSG23   025160      MSG62   026277      PIK3    000774      RD10    010564
EXLRC   015602      MSG23A  025166      MSG64   026303      PIK4    000776      RD11    010574
FC      000516      MSG23B  025173      MSG65   026325      PIK5    001000      RD2     010070
FCER    021342      MSG23C  025200      MSG66   026336      PIK6    001002      RD2A    010110
FCSAV   000632      MSG23D  025205      MSG67   026351      PIK7    001004      RD2B    010120
FMCNT   000556      MSG23E  025213      MSG68   026361      PIK8    001006      RD3     010150
FREX    021420      MSG23F  025220      MSG69   026372      POPPOP= 022626      RD4     010202
FRPRT   021356      MSG23G  025225      MSG7    024762      POPSP = 005726      RD4A    010300
FRO     021410      MSG23H  025232      MSG70   026410      PRB     000622      RD4A0   010332
HDRFL   000666      MSG23I  025237      MSG71   026444      PRS     000620      RD4A1   010354
HRDS    022464      MSG23J  025244      MSG72   026465      PSW     000604      RD4A2   010376
HRDS0   022524      MSG23K  025251      MSG73   026476      RANBAS  000624      RD4B    010404
HRDS1   022532      MSG23L  025256      MSG74   026507      RANG    023632      RD4C    010410
HRDS2   022546      MSG23M  025263      MSG75   026532      RANSAV  000626      RD4D    010440
HRDS3   022570      MSG24   025270      MSG76   026545      RANSET  004230      RD4E    010444
HRDS4   022602      MSG25   025310      MSG77   026556      RCNT    000554      RD5     010454
INTRF   000566      MSG26   025330      MSG78   026567      RCNTR   012310      RD6     010476
LRCER   021344      MSG27   025341      MSG79   026630      RCSAV   000630      RD7     010522
LRCSV   021352      MSG28   025352      MSG8    024767      RDA     007730      RD7A    010552
MOLEX   021556      MSG3    024736      MSG9    024777      RDATA   033434      READ    007670
MOLSW   000634      MSG30   025354      MTC1    000674      RDCMD   000562      REGS    000544
MOLTAB  022122      MSG31   025442      MTINT   022216      RDERR1  001150      REOT    004262
MR      000534      MSG32   025540      MTINTA  022222      RDER1   001110      REOTC   005002
MSG1    024724      MSG33   025561      NOP   = 000240      RDER2   001112      REOTX   004740
MSG10   025005      MSG34   025575      NRTP    011236      RDER3   001114      REOTXX  004776
MSG100  026670      MSG35   025610      NRZOF   000652      RDER4   001116      REOT1   004276
MSG101  026711      MSG36   025631      OCTP    024274      RDER5   001120      REOT1A  004332
MSG102  026737      MSG37   025655      OCTPG   024450      RDER6   001122      REOT1B  004354
MSG103  026746      MSG38   025700      OCTPG0  024466      RDER7   001124      REOT1C  004376
MSG104  026757      MSG4    024743      OCTPG1  024472      RDER8   001126      REOT1D  004402
MSG105  026774      MSG40   025720      OCTP0   024314      RDEX    010640      REOT1E  004426
MSG106  027007      MSG41   025747      OCTP1   024336      RDFL    015210      REOT1F  004370
MSG107  027077      MSG42   025761      OCTP2   024344      RDRTG   010740      REOT2   004452
MSG108  027162      MSG43   026001      OCTP3   024434      RDRTX   011234      REOT2A  004514
MSG109  027176      MSG44   026005      OFL     024512      RDRTY   010646      REOT3   004542
MSG11   025012      MSG45   026034      PAPRT   023200      RDRT0   010660      REOT4   004572
MSG110  027230      MSG46   026046      PAPRTA  023402      RDRT1   010714      REOT5   004606
MSG111  027252      MSG47   026053      PAPRTB  023416      RDRT1A  010712      REOT6   004654
MSG112  027264      MSG48   026060      PAPRTC  023432      RDRT1B  010734      REOT7   004712
MSG113  027305      MSG49   026113      PAPRTD  023442      RDRT2   011024      RETRY   000602
MSG114  027316      MSG5    024750      PAPRTY  023544      RDRT3   011046      RFHARD  002730
```

```
RFSOFT  002670        STARTA  003060        TEMP2   000646        TTR1B   023750        WTER5   001100
RPCNT   000702        STARTB  003064        TEMP3   000650        TTR2    024006        WTER6   001102
RRHARD  002750        STARTC  003050        TEX     024246        TTR3    024016        WTER7   001104
RRSOFT  002710        STARTD  003144        TIB     000642        TTR4    024026        WTER8   001106
RSEQ    007342        STARTE  003136        TINER   024032        TTR5    024030        WTM     005730
RSEX    007514        STARTF  003122        TINF    000636        UDES    000552        WTM0    005772
RSF     007416        START0  003074        TINP    012334        UNP     000676        WTM1    006034
RSFR    007522        START1  003244        TINPA   012344        UNX     000766        WTM2    006044
RSFRX   007662        START2  003376        TINPB   012364        UN1     000746        WTM3    006054
RSFR0   007554        START3  003412        TINPB0  012562        UN2     000750        WTM4    006136
RSFR1   007566        START4  003426        TINPB1  012410        UN3     000752        WTM4A   006176
RSFR2   007620        START5  004144        TINPC   012610        UN4     000754        W0      005424
RSF0    007454        START6  004162        TINPX   014140        UN5     000756        W1      005466
RSF1    007470        START7  004216        TINP0   012722        UN6     000760        W2      005510
RSR     007374        START8  004224        TINP0A  013002        UN7     000762        W3      005524
RSTAL   000572        STAR0A  003172        TINP0B  013020        UN8     000764        W3A     005546
RTCNT   000704        STAR0B  003222        TINP0C  013060        UPS     000716        W4      005634
RTRN    000664        STAR1A  003256        TINP0D  013104        VECT    000546        W4A     005664
RTYFL   000714        STAR1B  003302        TINP0E  013134        WC      000512        W5      005700
RTY1    001050        STAR1C  003342        TINP1   013154        WDATA   027426        W6      005716
RTY2    001052        STAR4A  003516        TINP2   013234        WEX     006202        XOR     015552
RTY3    001054        STAR40  003442        TINP2A  013314        WRITE   005400        XORS    015576
RTY4    001056        STAUT   003032        TINP2B  013374        WRTE    005420        YOZ     011254
RTY5    001060        STAUT0  003170        TINP2C  013422        WRTSB   006666        YOZA    011322
RTY6    001062        STP     003612        TINP3   013442        WRTSB0  006740        YOZB    011330
RTY7    001064        STPX    004130        TINP4   013756        WRTSB1  006754        YOZC    011350
RTY8    001066        SWR     000606        TKB     000612        WRTSB2  006764        YOZC0   011440
RWND    005004        TAPG    021422        TKS     000610        WRTSB3  007040        YOZC1   011454
RWNDA   005016        TAPG0   021434        TMEX    000564        WRTY    006246        YOZC2   011462
RWNDX   005344        TAPG1   021532        TMFLG   000700        WRTYR   006300        YOZD    011502
RWND0   005054        TAPG2   021546        TOB     000640        WRTYTM  006274        YOZD0   011604
RWND1   005132        TAPG3   021564        TOG     024232        WRTY0   006254        YOZD1   011630
RWND1A  005174        TAPG3A  021650        TPB     000616        WRTY1   006362        YOZD2   011576
RWND2   005214        TAPG3B  021664        TPOS    014150        WRTY2   006364        YOZD3   011622
RWND3   005224        TAPG3C  021704        TPOS1   014162        WRTY2A  006426        YOZD4   011556
RWND4   005256        TAPG3D  021750        TPS     000614        WRTY2B  006446        YOZE    011634
RWND5   005276        TAPG3E  021774        TSTAL   000576        WRTY3   006522        YOZF    011664
RWND6   005316        TAPG3F  021632        TTIN    024050        WRTY3A  006540        YOZF0   011670
 CVFL   014146        TAPG4   022022        TTINT   022142        WRTY4   006626        YOZG    011766
 ERFL   000710        TAPG5   022036        TTINT0  022166        WRTY5   006662        YOZH    012002
SN      000540        TAPG6   022104        TTIN1   024070        WRW     006222        YOZ0    011274
SNPG    024700        TAPG7   022116        TTIN2   024104        WRWX    006244        YSTAL   000600
SNPT    024622        TBELA   024254        TTOUT   024122        WSTAL   000574        .       = 033436
SPFLG   000570        TBELL   024250        TTR     023664        WTER1   001070
STAL    000670        TCRLF   024164        TTR0    023672        WTER2   001072
STALL   012240        TCRLFA  024202        TTR1    023720        WTER3   001074
START   003026        TEMP1   000644        TTR1A   023734        WTER4   001076

. ABS.  033436      000


 ERRORS DETECTED:  0

 .CZTUIA.SEQ/SOL_CZTUIA.P11
 RUN-TIME: 31 65 5 SECONDS
 RUN-TIME RATIO: 526/102=5.1
```

CORE USED: 5K (10 PAGES)