TM02,TE16,
TU16

RELIAB
CZTUAIO

AH-9448I-MC
FICHE 1 OF 1

.REM %

IDENTIFICATION

PRODUCT CODE: AC-9447I-MC

PRODUCT TITLE: CZTUAI0 TM02-TU16/TE16 RELIAB

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: R.B. BARNES

PRODUCT DATE: 29-MAR-82 B.T. LEBLANC

TABLE OF CONTENTS

1. ABSTRACT

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

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

HOWEVER; THE RH AND TM02 ARE TESTED SOMEWHAT INTRINSICALLY DURING THE TEST CYCLE IN ORDER TO PROVIDE FULL INFORMATION ABOUT ANY ERROR CONDITIONS DETECTED.

DURING A TEST CYCLE, CHECKS ARE MADE FOR STATUS ERRORS,DATA ERRORS, POSITION ERRORS,WORD COUNT AND CURRENT MEMORY ADDRESS ERRORS WHEREVER APPLICABLE AS DETECTED BY THE RH OR TM02.

2. REQUIREMENTS (HARDWARE)

A. ANY PDP-11 PROCESSOR - WITH OR WITHOUT HARDWARE SWITCH REGISTER
B. 8K OF CORE
C. TELETYPE
D. TM02 TAPE CONTROLLER
E. 1 TO 8 MAG TAPE DRIVES
F. MASSBUS CONTROLLER

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING BINARY TAPES

4. STARTING PROCEDURE

THERE ARE FOUR (4) STARTING ADDRESSES THAT MAY BE USED:

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 ALSO SECTION 11 FOR DETAILS.

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

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 PARTICULAR SLAVE'S NEEDS. AS MANY AS EIGHT (8) SLAVE NUMBER REQUESTS MAY BE USED, HOWEVER, AT LEAST ONE MUST BE USED. THE SLAVE NUMBERS AND THEIR RESPECTIVE DENSITY, PARITY AND FORMAT MAY BE ENTERED IN ANY ORDER. THE INFORMATION FOR EACH SLAVE ENTERED IS LOADED INTO A TABLE FOR REFERENCE IN TESTING. IF LESS THAN EIGHT(8) SLAVES ARE REQUIRED, THEN RESPONDING TO THE SLAVE NUMBER REQUEST WITH A CARRIAGE RETURN WILL TERMINATE THE SLAVE ENTRIES AND CONTINUE TO THE NEXT PARAMETER. IT SHOULD BE REMEMBERED THAT AT LEAST ONE SLAVE NUMBER REQUEST

MUST BE ENTERED. IF THE FIRST
REQUEST IS RESPONDED TO BY A CARRIAGE
RETURN, THEN THE REQUEST WILL BE REPEATED.

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.

PATTERN NUMBER: THIS RESPONSE IS A TWO (2) CHARACTER OCTAL
NUMBER WITHIN THE LIMITS OF 0 THRU 15(8).
THE NUMBER ENTERED WILL CAUSE A SPECIFIC DATA
PATTERN TO BE USED FOR ALL READING AND WRITING. THIS
DATA PATTERN IS NOT CHANGED UNLESS RANDOM DATA
IS REQUESTED BY SETTING CONSOLE SWITCH EIGHT (8)
TO A ONE. RESETTING OF THE RANDOM DATA SWITCH
DOES NOT CAUSE REVERSION TO THE FIXED PATTERN,
BUT WILL HOLD THE LAST GENERATED PATTERN UNTIL
A RESTART IS DONE FROM LOCATION 200(8), 210(8), OR 300(8).
WHEN OPERATING IN NRZ MODE (DENSITY 0-3)
THE PROGRAM CONSTRUCTS AND SAVES
BOTH AN EXPECTED CRC CHARACTER AND AN
LRC CHARACTER FOR COMPARISONS WITH
THE HARDWARE GENERATED CHECK CHARACTER
IN BOTH READ AND WRITE.
THE SELECTION OF DATA PATTERN ZERO (0) HAS A
SPECIAL USE. PATTERN NUMBER ZERO (0) WILL
CAUSE TO BE READ IN AT THE HIGH SPEED PAPER
TAPE READER ANY DATA PATTERN DESIRED.
THE EXTERNAL INPUT DATA THOUGH THE READER
IS DONE BY PREPARING A PAPER TAPE WITH A PROGRAM
CALLED DTC. (MAINDEC-11-DZTUF-A-D) ANY CONFIGURATION
OF BITS AND CHARACTERS MAY BE USED AND A LIMIT
OF 377(8) CHARATERS IS IMPOSED. WHEN EXTERNAL
DATA IS INPUT, THE ENTIRE WRITE BUFFER IN
CORE IS FILLED WITH THE PATTERN SO THAT ANY
SIZE RECORD MAY BE USED. DATA PATTERN
PATTERN ZERO (0) EXTERNAL PAPER TAPE NEED ONLY
BE READ ONCE AT INITIAL START OF 200(8), AND
NEED NOT BE READ AGAIN UNLESS OVERWRITTEN BY
RANDOM DATA. BE SURE TO LOAD THE READER
BEFORE PRESSING START.

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

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

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

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

READ: THE TIME DELAY BETWEEN EACH RECORD READ

WRITE: THE TIME DELAY BETWEEN EACH RECORD WRITTEN

TURN AROUND: TIME DELAY BETWEEN CHANGES OF TAPE DIRECTION (FORWARD,TO REVERSE,ETC.) AND BETWEEN BLOCKS.

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

A. RECORD COUNT = 100
B. CHARACTER COUNT = 200
C. PATTERN NUMBER = 1
D. TM=0
E. INTERCHANGE READ = 0
F. SINGLE PASS = 0
G. READ STALL = 1
H. WRITE STALL = 1
I. TURN AROUND STALL = 1

```
SAMPLE START AT 200(8):

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

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

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

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

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

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

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

4.1 AUTOMATIC MODE OPERATION

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

** EXCEPTION: IF LOADED VIA TMDP CHAIN MODE THE PROGRAM WILL NOT TEST SLAVE 0 ON THE FIRST AVAILABLE DRIVE.

5. DATA PATTERNS

THERE ARE FIFTEEN DATA PATTERN GENERATORS STORED IN CORE AND ANY ONE OF THESE MAY BE SELECTED. THE ONE UNIQUE CASE IS PATTERN ZERO(0); SELECTION OF PATTERN ZERO(0) REQUIRES THAT A PREVIOUSLY PREPARED PAPER TAPE BE ENTERED AT THE HIGH SPEED READER. THIS TAPE CONTAINS A DATA PATTERN OF NO MORE THAN 377 OCTAL CHARACTERS. THE FIRST CHARACTER READ IN IS THE NUMBER OF ACTUAL DATA CHARACTERS THAT ARE CONTAINED ON THE TAPE. EACH DATA CHARACTER MAY BE ANY COMBINATION OF BITS AND WILL BE LOADED INTO CORE AS THEY APPEAR ON THE TAPE. NO MATTER HOW MANY CHARACTERS ARE ON TAPE, THE ENTIRE WRITE BUFFER (4000 CHARACTERS) WILL BE FILLED WITH THE PATTERN ENTERED SO THAT ANY SIZE RECORD CAN BE USED. (SEE DTC MAINDEC-11-DZTUF-A-D) THE PROGRAM GENERATES A CYLIC REDUNDENCY CHECK CHARACTER (CRC) AND A LONGITUDINAL REDUNDENCY CHECK CHARACTER (LRC) FOR COMPARISONS AGAINST THE CRC AND LRC GENERATED BY THE HARDWARE IN NRZI READS OR WRITES.

THE FOLLOWING IS A LIST OF THE DATA PATTERNS AVAILABLE:

DATA0: EXTERNAL INPUT THRU HIGH SPEED READER (SEE DTC)
DATA1: ALL ONE BITS IN ALL CHARACTERS
DATA2: ALL ZERO BITS IN ALL CHARACTERS
DATA3: A ONE BIT WALKING FROM RIGHT TO LEFT IN A FIELD OF ZEROS
DATA4: A ZERO BIT WALKING FROM RIGHT TO LEFT IN A FIELD OF ONES.
DATA5: ALTERNATING ONE AND ZERO BITS IN EACH CHARACTER
DATA6: ALTERNATING ZERO AND ONE BITS IN EACH CHARACTER
DATA7: SAME AS DATA5 BUT WITH EVERY OTHER CHARACTER COMPLEMENTED
DATA10: WALKING ONE/ALL ONE IN ALTERNATING CHARACTERS
DATA11: INCREMENTING CHARACTERS (000-377)
DATA12: DECREMENTING CHARACTERS (377-000)
DATA13: ALTERNATING CHARACTERS OF ALL ZERO AND ALL ONE BITS
DATA14: WALKING ZERO/ALL ZERO IN ALTERNATING CHARACTERS
DATA15: AUTO SEQUENCE PATTERN 0,0,-1,-1,-1,0,0

6. RANDOMIZATION

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

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

B. RANDOM CHARACTER COUNT: (CONSOLE SWITCH 7)
GENERATES A DIFFERENT NUMBER OF CHARACTERS PER RECORD TO BE WRITTEN ON EACH BLOCK CYCLE. THE SAME NUMBER OF CHARACTERS PER RECORD IS WRITTEN OR READ ON EACH AVAILABLE UNIT BEFORE BEING CHANGED. RESETTING SWITCH 7 HOLDS THE LAST VALUE GENERATED.

C. RANDOM RECORD COUNT: (CONSOLE SWITCH 6)
GENERATES A DIFFERENT NUMBER OF RECORDS FOR EACH BLOCK OF DATA WRITTEN OR READ ON EACH BLOCK CYCLE. THE SAME NUMBER OF RECORDS IS WRITTEN OR READ ON EACH AVAILABLE UNIT BEFORE BEING CHANGED. RESETTING SWITCH 6 HOLDS LAST VALUE GENERATED.

7. DYNAMIC PARAMETERS:

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

8. CONSOLE SWITCH SETTINGS

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

CONTROL:

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

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

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

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

3) AFTER THE ''NEW='' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE OF THE FOLLOWING AT THE TTY:

   A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>. (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS WILL BE ALLOWED)
   IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.

   B) IF A CONTROL U <^U> IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 2.

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

SW15: 1=STOP ON ERROR
(100000)0=CONTINUE ON ERROR

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

SW13: 1=DO NOT CHECK DATA ERRORS

```
(020000)0=CHECK DATA ERRORS

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

SW11:   1=DO NOT CHECK READ STATUS ERRORS (NOR CLEAR THEM IF THEY DO OCCUR)
(004000)0=CHECK READ STATUS ERRORS

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

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

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

SW7:    1=GENERATE RANDOM CHARACTER COUNT
000200)0=USE FIXED CHARACTER COUNT

SW6:    1=GENERATE RANDOM RECORD COUNT
(000100)0=USED FIXED RECORD COUNT

SW5:    1=YOZZLE ON CURRENT RECORD
(000040)0=DO NOT YOZZLE ON RECORD

SW4:    1=DO WRITE/READ RETRIES
(000020)0=DO NOT RETRY

SW3:    1=DO NOT READ FORWARD
(000010)0=READ FORWARD

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

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

SW0:    1=DO NOT WRITE
(000001)0=WRITE
```

SWITCH EXPLANATION AND EXAMPLES:

SW0-3: THESE SWITCHES ARE USED TO CONTROL THE SEQUENCE OF MAG TAPE OPERATIONS PREFORMED ON EACH AVAILABLE UNIT. THE BLOCK OF DATA DESCRIBED THROUGH THE RESPONSES TO TELETYPE REQUESTS AT INITIAL START WILL BE EITHER WRITTEN OR READ FROM EACH AVAILABLE UNIT IN THE ORDER THAT THEY WERE ENTERED. THE SEQUENCE OF OPERATIONS IS CALLED A CYCLE, AND WILL BE PERFORMED CONTINUOUSLY UNTIL STOPPED BY THE OPERATOR. WHEN END OF TAPE IS REACHED, THE UNIT WILL BE REWOUND AND FLAGGED AS UNAVAILABLE FOR TEST UNTIL ALL UNITS HAVE REACH EOT, AT WHICH TIME TESTING IS RESUMED ON ALL AVAILABLE UNITS.

EXAMPLES: 0-3

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

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

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

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

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

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

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

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

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

SW4:

SWITCH FOUR (4), WHEN SET TO A ONE (1), WILL
CAUSE ANY DATA RELATED ERROR TO BE RETRIED.
THE WRITE RETRY SCHEME CONSISTS OF REWRITING THE RECORD
IN THE SAME SPOT ON TAPE FOUR (4) TIMES. IF
ALL FOUR (4) REPEATS ARE SUCCESSFUL, THE
RECORD IS CONSIDERED AS RECOVERED, AND A TAPE
WRITE ERROR IS LOGGED. IF ANY OF THE FOUR
(4) REPEATS IS UNSUCESSFUL, A SKIP ERASE IS
DONE, A SUPECTED BAD TAPE SPOT IS LOGGED
AT THIS BLOCK AND RECORD NUMBER, AND A SECOND
RETRY OF FOUR REPEATS IS DONE. IF AFTER FOUR
(4) RETRIES, THE RECORD CANNOT BE RECOVERED
A NOTIFICATION IS PRINTED, AND TESTING IS
RESUMED ON THE NEXT RECORD.
IF 20(8) BAD TAPE SPOTS ARE FOUND, THE
SLAVE WILL BE REWOUND AND REMOVED FROM
TESTING WITH AN APPROPRIATE MESSAGE PRINTED.
THE READ RETRY SCHEME CONSISTS OF REREADING THE
RECORD UP TO EIGHT TIMES. IF ALL EIGHT REREADS ARE
BAD, IT IS A HARD ERROR. IF ANY REREAD IS
SUCCESSFUL, THIS IS A SOFT ERROR.
IF THE ORIGINAL ERROR IS OF THE NON-RETRYABLE
TYPE (IE: ILF,RMR,ILR,NEF,CBUSPE), THE RETRY SCHEME
IS NOT ENTERED AND A MESSAGE IS PRINTED.

SW5:

SWITCH FIVE (5) WHEN SET DURING A READ
FORWARD OR REVERSE WILL CAUSE THE TAPE
TO CONTINUOUSLY READ THE CURRENT RECORD
BY SPACING EITHER FORWARD OR REVERSE AND
REREADING THAT RECORD. THIS TAPE MOVEMENT
IS CALLED YOZZLING. THERE IS A SOFTWARE
DELAY EXECUTED BETWEEN EACH SPACE/READ
OF THE RECORD AND IT MAY BE VARIED BY
TYPING CONTROL C ON THE TELETYPE DURING
THE EXECUTION OF THE YOZZLE AND RESPONDING
TO THE PRINTED REQUEST WITH A SIX (6)
DIGIT VALUE. THE YOZZLE STALL IS PRESET
TO A VALUE OF 3000 IN THE PROGRAM TO
PREVENT EXCESSIVE TAPE WEAR, BUT MAY BE
SET TO ANY VALUE THROUGH THE TELETYPE.

SW6-8:

THESE THREE (3) SWITCHES CONTROL THE RANDOMIZATION
OF DATA AND BLOCK SIZE AND MAY BE SET
AND RESET AT ANY TIME. THE ACTUAL CHANGE
WILL TAKE PLACE BETWEEN BLOCK CYCLES.

SW9:

SWITCH NINE (9) WHEN SET WILL CAUSE ALL
AVAILABLE TAPE UNITS TO BE REWOUND AT
THE END OF THE CURRENT BLOCK CYCLE. TESTING
WILL BE RESUMED AT A BLOCK COUNT OF
ONE (1) WHEN ALL UNITS HAVE REACHED BOT.

SW10-13: THESE SWITCHES ARE USED TO CONTROL THE ERROR HANDLING TO BE DONE ON THE TAPE OPERATION DESCRIBED BY SWITCHES 0-3.

A. SWITCH TEN (10; WHEN SET TO A ONE WILL DISALLOW ANY ERROR PRINTOUTS MADE ON THE OPERATION IN PROGRESS. CATASTROPHIC FAILURES AND INFORMATION PRINTOUTS WILL STILL OCCUR. IE: UNIT NOT AVAILABLE, ILLEGAL BOT, DROP OR PICK OVERFLOW, AND EOT REWIND.

B. SWITCH ELEVEN (11) WHEN SET TO A ONE WILL DISALLOW THE CHECKING FOR STATUS ERRORS ON READ (FORWARD OR REVERSE) OPERATIONS.

C. SWITCH TWELVE (12) WHEN SET TO A ONE WILL DISALLOW THE CHECKING FOR STATUS ERRORS ON WRITE OPERATIONS.

D. SWITCH THIRTEEN (13) WHEN SET TO A ONE WILL DISALLOW THE CHECKING OF READ DATA. THIS SWITCH HAS NO EFFECT ON STATUS CHECKING.

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

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

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

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

***IF THE SOFTWARE SWITCH REGISTER IS USED AND THE PROGRAM HALTS THEN THE OPERATOR CAN PRESS A <^G> CONTROL G BEFORE HITTING CONTINUE. THIS WILL ALLOW THE OPERATOR TO ENTER DATA INTO THE SOFTWARE SWITCH REGISTER.

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.

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.

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/TU16 CONTROL REGISTER
WC = RH WORD COUNT
BA = RH BUS ADDRESS
FC = TU16 FRAME COUNT
CS2 = RH CONTROLLER STATUS
DS = TU16 DRIVE STATUS
ER = TU16 ERROR REGISTER
AS = ATTENTION SUMMARY
CK = TU16 CHECK CHARACTER
DB = RH DATA BUFFER
MR = TU16 MAINTENENCE REGISTER
DT = TU16 DRIVE TYPE
SN = TU16 SERIAL NUMBER
TC = TU16 TEST CONTROL
*F = DATA FORMAT
*P = PARITY
*D = DENSITY
*PATRN = DATA PATTERN NUMBER (R = RANDOM)

```
EXAMPLE 1: IN THIS EXAMPLE SLAVE 1 ON TM02 0 WAS
        OPERATING AT 1600 BPI IN ODD PARITY USING
        THE NINE CHANNEL NORMAL DATA FORMAT.  A
        WRITE STATUS ERROR WAS DETECTED.  THE BAD
        STATUS INDICATES THAT AN UNCORRECTABLE DATA
        ERROR (BIT 6 OF ER) AND A PE FORMAT ERROR
        (BIT 7 OF ER) OCCURED DURING THE WRITE
        OPERATION OF THE SIXTH (6) RECORD OF THE
        FIFTY (50) RECORDS IN BLOCK (2).  THE
        SIZE OF THE RECORD WAS TWO HUNDRED (200)
        FRAMES.  THE CHECK CHARACTER REFLECTS THE BAD TRACK.

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

EXAMPLE 2: IN THIS EXAMPLE SLAVE 3 ON TM02 1 WAS
        OPERATING AT 800 BPI IN EVEN PARITY USING
        THE NINE CHANNEL NORMAL DATA FORMAT.  A
        READ STATUS ERROR WAS DETECTED DURING
        THE REVERSE READ OF THE TENTH (10)
        RECORD OF THE 25 RECORDS IN THIS BLOCK (12).
        THE SIZE OF THE RECORD IS TWENTY (20) FRAMES.
        THE PRINTOUT INDICATES THE DETECTION OF A
        VERTICAL PARITY ERROR (VPE: BIT 6 OF ER) AND A
        CYCLIC REDUNDENCY ERROR (CRC: BIT 15 OF ER).
        THE CRC CHARACTER, AS RECEIVED, IS NOT
        AS EXPECTED AND IS PRINTED SHOWING BOTH
        THE ACTUAL (FIRST) AND THE EXPECTED (LAST).

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

SEQ 0026

10. STATISTICS PRINTOUT

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

STATISTICS PRINT EXAMPLE (A HEADER WILL PRECEED THE STATS)

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

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

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

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

11. AUTO SEQUENCE

THE AUTO SEQUENCE (START AT ADDRESS 240) WILL EXECUTE A PREDETERMINED TEST PLAN ON ALL AVAILABLE SLAVES ON EACH AVAILABLE TM02. THE ONLY OPERATOR RESPONSE IS TO THE TYPED REQUESTS FOR THE RH ADDRESS, VECTOR, CONTINUOUS OR SINGLE CYCLE, AND NRZ ONLY. ALL SWITCHES REMAIN ACTIVE AND MAY BE USED NORMALLY; HOWEVER THE IDEA IS TO LEAVE ALL SWITCHES DOWN AND ALLOW FULL EXECUTION OF THE TEST PLAN FOR SYSTEM CHECKOUT.

SAMPLE START AT 240(8): AUTO SEQUENCE.

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

```
        TU16 AUTO SEQUENCE TEST
        ENTER CONDITIONS IN OCTAL

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

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

AS EACH TM02 AND ITS SLAVES ARE FOUND, A DIVIDER LINE OF ASTERICKS WILL BE PRINTED FOLLOWED BY A PRINTOUT OF THE TM02 AND ITS SLAVES BEING TESTED. AS EACH TM02 AND ITS SLAVES ARE FINISHED, ANOTHER DIVIDER IS PRINTED BEFORE TESTING IS RESUMED ON THE NEXT AVAILABLE DRIVE.

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

AUTO SEQUENCE TEST PLAN:

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

1. NRZ CYCLE:

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

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

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

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

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

12. TESTING PROCEDURES

AS PREVIOUSLY STATED THIS PROGRAM CONTAINS NO FIXED TESTS. THE ENTIRE TEST CYCLE TO BE EXECUTED IS DESCRIBED BY THE OPERATOR THOUGH PESPONSES 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

%

```
1343
1344                    .LIST   BIN,LOC,SEQ
1345                    .TITLE CZTUAIO TM02-TU16/TE16 RELIAB
1346                    ;ZZ - CZTUAIO
1347                    ;21 APRIL 76
1348                    ;R. BARNES
1349
1350                    ;REVISED (++G) J.G.ADAMS MAY 1977        1)INTERMITTENT PGM FAILURE
1351                    ;++G                                     ON BAD TAPE OVERFLOW
1352                    ;++G                                     2)TAPE RUNAWAY AT EOT
1353                    ;++G                                     3)ERRONEOUS ERROR TYPEOUT
1354                    ;++G                                     4)CHANGED MISC INST'S TO
1355                    ;++G                                     CONSERVE MEMORY USAGE.
1356                    ;++G                                     5)ADDED ACT11 HOOKS
1357                    ;++G                                     6)FIXED TTY INPUT
1358
1359                    .MCALL  .$ACT11,.$EOP,$CHAIN    ;++G ACT11 HOOKS
1360                    .NLIST  MC              ;++G DO NOT LIST MACRO CALLS
1361                    .LIST   ME              ;++G LIST MACRO EXPANSIONS
1362                    .ENABLE ABS,AMA         ;++G ENABLE ABS AND MODE '37'
1363
1364                    ;CONSOLE SWITCHES************
1365
1366                    ;SW15: 1=STOP ON ERROR
1367                    ;       0=CONTINUE ON ERROR
1368
1369                    ;SW14: 1=PRINT READ/WRITE STATS
1370                    ;       0=DO NOT PRINT STATS
1371
1372                    ;SW13: 1=DO NOT CHECK DATA
1373                    ;       0=CHECK DATA
1374                    ;SW12: 1=DO NOT CHECK WRITE ERRORS
1375                    ;       0=CHECK WRITE ERRORS
1376                    ;SW11: 1=DO NOT CHECK READ ERRORS
1377                    ;       0=CHECK READ ERRORS
1378                    ;SW10: 1=DO NOT PRINT ERRORS
1379                    ;       0=PRINT ERRORS
1380
1381                    ;SW9:  1=REWIND TAPE
1382                    ;       0=DO NOT REWIND
1383
1384                    ;SW8:  1=USE RANDOM DATA
1385                    ;       0=USE FIXED DATA PATTERN
1386                    ;SW7:  1=USE RANDOM CHARACTER COUNT
1387                    ;       0=USE FIXED CHAR COUNT
1388                    ;SW6:  1=USE RANDOM RECORD COUNT
1389                    ;       0=USE FIXED RECORD COUNT
1390
1391                    ;SW5:  1=YOZZLE ON CURRENT RECORD
1392                    ;       0=DO NOT YOZZLE
1393
1394                    ;SW4:  1=DO BOTH READ AND WRITE RETRIES
1395                    ;       0=INHIBIT RETRIES
1396
1397                    ;SW3:  1=DO NOT READ FORWARD
```

```
1398                                    ;          0=READ FORWARD
1399                                    ;SW2:  1=DO NOT READ REVERSE
1400                                    ;          0=READ REVERSE
1401                                    ;SW1:  1=READ FORWARD FIRST
1402                                    ;          0=READ REVERSE FIRST
1403                                    ;SW0:  1=DO NOT WRITE
1404                                    ;          0=WRITE
```

```
                        ;TU16/TE16 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

                                        ;TRAP CATCHERS**********
000030                                  .=30
000030  024676                          TRAP30
000032                                  .=32
000032  000340                          340

                                ;ACT11 HOOK **********
000034                                  $SVPC=.                 ;SAVE CURRENT LOCATION CTR
000046                                  .=46
000046  005116                          .WORD   $ENDAD          ;SET LOCATION 46
000052                                  .=52
000052  000000                          .WORD   0               ;SET LOCATION 52 = 0
000034                                  .=$SVPC                 ;RESTORE LOCATION CTR

                                        ;TTY INTERRUPT VECTOR**********
000060                                  .=60
000060  021630                          TTINT                   ;TTY INTERRUPT HANDLER ADDRESS
000062  000000                          0

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

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

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

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

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

000210                                  .=210
000210  005037  015150                  CLR     RDFL
000214  000137  003160                  JMP     STARTA          ;USE FIXED PARAMETERS; NEW DATA

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

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

SEQ 0034

```
1507
1508                                    ;AUTO SEQUENCE START**********
1509
1510            000240                  .=240
1511  000240    005237  000734          INC     ASEQF           ;SET AUTO SEQUENCE FLAG
1512  000244    000137  003136          JMP     STAUT           ;GO TO START OF AUTO SEQUENCE
```

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

                000300                  .=300
000300  005237  014150                  INC     SCVFL                   ;SET SHORT CONVERSATION FLAG
0003C4  000137  003026                  JMP     START                   ;ENTER SHORT PARAMETER LIST

                000510                  .=510
                                        ;TU16/TE16 REGISTER EQUIVS**********

000510  172440                  C1:     172440
000512  172442                  WC:     172442
000514  172444                  BA:     172444
000516  172446                  FC:     172446
000520  172450                  CS:     172450
000522  172452                  DS:     172452
000524  172454                  ER:     172454
000526  172456                  AS:     172456
000530  172460                  CC:     172460
000532  172462                  DB:     172462
000534  172464                  MR:     172464
000536  172466                  DT:     172466
000540  172470                  SN:     172470
000542  172472                  C2:     172472

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

000544  172440          REGS:   172440          ;STARTING REGISTER ADDRESS (CS1)
000546  000224          VECT:   224             ;VECTOR ADDRESS (RH INTERRUPT)
000550  000000          DVN:    0               ;DRIVE NUMBER
000552  000000          UDES:   0               ;UNIT DESCRIPTION (PARITY,DENSITY,UNIT,FORMAT)
000554  000100          RCNT:   100             ;RECORD COUNTER
000556  177600          FMCNT:  177600          ;NUMBER OF CHAR (4 - 4000) OCTAL IN TWOS COMPLEMENT
000560  000001          PATRN:  1               ;DATA PATTERN SELECTOR (0 - 15) OCTAL
000562  000002          RDCMD:  2               ;READ COMMAND
000564  000000          TMEX:   0               ;TAPE MARK FLAG: 1=TM 0=NO TM
000566  000000          INTRF:  0               ;INTERCHANGE READ 1=YES 0=NO
000570  000000          SPFLG:  0               ;SINGLE PASS 1=YES 0=NO
000572  000001          RSTAL:  1               ;READ STALL
000574  000001          WSTAL:  1               ;WRITE STALL
000576  000001          TSTAL:  1               ;TURN AROUND STAL
000600  002000          YSTAL:  2000            ;YOZZLE STAL
000602  000010          RETRY:  10              ;READ RETRY NUMBER
000604  177776          PSW:    177776          ;PROCESSOR STATUS
000606  177570          SWR:    177570          ;CONSOLE SWITCHES
000610  177560          TKS:    177560          ;TTY READ STATUS REGISTER
000612  177562          TKB:    177562          ;TTY READ BUFFER
000614  177564          TPS:    177564          ;TTY PUNCH STATUS REGISTER
000616  177566          TPB:    177566          ;TTY PUNCH OUTPUT REGISTER
000620  177550          PRS:    177550          ;H/S READER STATUS REGISTER
000622  177552          PRB:    177552          ;H/S READER BUFFER
000624  153624          RANBAS: 153624          ;RANDOM NUMBER GENERATOR BASE
000626  032561          RANSAV: 032561          ;RANDOM NUMBER BUFFER
000630  000000          RCSAV:  0               ;RECORD COUNT SAVE
000632  000000          FCSAV:  0               ;FRAME COUNT SAVE
```

SEQ 0036

```
1567
1568                                    ;FLAGS AND COUNTERS**********
1569
1570  000634  000000  TINF:    0       ;TTY ENTERY FLAG
1571  000636  000000  TOB:     0       ;TTY OUTPUT BUFFER
1572  000640  000000  TIB:     0       ;TTY INPUT BUFFER
1573  000642  000000  TEMP1:   0       ;TEMP STORAGE
1574  000644  000000  TEMP2:   0       ;TEMP STORAGE
1575  000646  000000  TEMP3:   0       ;TEMP STORAGE
1576  000650  000000  NRZOF:   0       ;NRZ ONLY FLAG
1577  000652  000000  EMADDR:  0       ;ERROR MSG ADDRESS STORAGE
1578  000654  000000  BLCNTR:  0       ;BLOCK COUNTER
1579  000656  000000  BBC:     0       ;BAD RECORD COUNTER
1580  000660  000000  EOTREC:  0       ;EOT FLAG
1581  000662  000000  RTRN:    0       ;INTERRUUPT RETURN STORAGE
1582  000664  000000  HDRFL:   0       ;HEADER FLAG
1583  000666  000000  STAL:    0       ;DELAY STORAGE
1584  000670  000000  PFLG:    0       ;PRINT FLAG
1585  000672  000000  MTC1:    0       ;MAG TAPE CONT REGISTER BUFFER
1586  000674  000000  UNP:     0       ;UNIT TABLE POINTER
1587  000676  000000  TMFLG:   0       ;TAPE MARK FLAG
1588  000700  000000  RPCNT:   0       ;REPEAT COUNTER
1589  000702  000000  RTCNT:   0       ;RETRY COUNTER
1590  000704  000000  DERFL:   0       ;DATA ERROR FLAG
1591  000706  000000  SERFL:   0       ;STATUS ERROR FLAG
1592  000710  000000  BCNT:    0       ;BIT COUNTER
1593  000712  000000  RTYFL:   0       ;RETRY FLAG
1594  000714  000000  UPS:     0       ;UNIT POINTER SAVE
1595  000716  000000  BDPP:    0       ;BITS DROPPED POINTER
1596  000720  000000  BPKP:    0       ;BITS PICKED POINTER
1597  000722  000000  ERSAV:   0       ;ERROR SAVE LOC
1598  000724  000000  BT[illegible]G:   0       ;BAD TAPE FLAG
1599  000726  000000  BTSTF:   0       ;STATISTIC PRINT FLAG
1600  000730  000000  BTPT:    0       ;BAD TAPE POINTER
1601  000732  000000  ERTFL:   0       ;ERASE FLAG
1602  000734  000000  ASEQF:   0       ;AUTO SEQ FLAG
1603  000736  000000  ADRVN:   0       ;UTO SEQ DRIVE NUMBER
1604  000740  000000  ABLCNT:  0       ;AUTO BLOCK COUNTER
1605  000742  000000  ASEQCF:  0       ;AUTO SEQ CONTINUOUS FLAG
1606  000744  000000  EOPB1:   0       ;EOP FLAG
```

```
1607
1608                                        ;UNIT ORDER AND DESCRIPTION TABLE **********
1609
1610  000746  000000          UN1:    0             ;THIS TABLE IS LOADED
1611  000750  000000          UN2:    0             ;WITH UNIT NUMBERS AND
1612  000752  000000          UN3:    0             ;THEIR DESCRIPTIONS IN
1613  000754  000000          UN4:    0             ;THE ORDER THAT THEY
1614  000756  000000          UN5:    0             ;WILL BE TESTED
1615  000760  000000          UN6:    0
1616  000762  000000          UN7:    0
1617  000764  000000          UN8:    0
1618  000766  177777          UNX:    -1
1619
1620                                        ;UNIT DROPS AND PICKS POINTERS*********
1621
1622  000770  001210          PIK1:   BP00
1623  000772  001230          PIK2:   BP10
1624  000774  001250          PIK3:   BP20
1625  000776  001270          PIK4:   BP30
1626  001000  001310          PIK5:   BP40
1627  001002  001330          PIK6:   BP50
1628  001004  001350          PIK7:   BP60
1629  001006  001370          PIK8:   BP70
1630  001010  001410          DRP1:   BD00
1631  001012  001430          DRP2:   BD10
1632  001014  001450          DRP3:   BD20
1633  001016  001470          DRP4:   BD30
1634  001020  001510          DRP5:   BD40
1635  001022  001530          DRP6:   BD50
1636  001024  001550          DRP7:   BD60
1637  001026  001570          DRP8:   BD70
1638
1639                                        ;UNIT BAD TAPE POINTERS**********
1640
1641  001030  001610          BTADDR: BT00
1642  001032  001714                  BT01
1643  001034  002020                  BT02
1644  001036  002124                  BT03
1645  001040  002230                  BT04
1646  001042  002334                  BT05
1647  001044  002440                  BT06
1648  001046  002544                  BT07
1649
1650                                        ;UNIT WRITE RETRY COUNTER**********
1651
1652  001050  000000          RTY1:   0
1653  001052  000000          RTY2:   0
1654  001054  000000          RTY3:   0
1655  001056  000000          RTY4:   0
1656  001060  000000          RTY5:   0
1657  001062  000000          RTY6:   0
1658  001064  000000          RTY7:   0
1659  001066  000000          RTY8:   0
1660
1661                                        ;UNIT WRITE ERRORS**********
1662
```

```
1663  001070  000000                WTER1:  0
1664  001072  000000                WTER2:  0
1665  001074  000000                WTER3:  0
1666  001076  000000                WTER4:  0
1667  001100  000000                WTER5:  0
1668  001102  000000                WTER6:  0
1669  001104  000000                WTER7:  0
1670  001106  000000                WTER8:  0
1671
1672                                      ;UNIT READ FORWARD ERRORS**********
1673
1674  001110  000000                RDER1:  0
1675  001112  000000                RDER2:  0
1676  001114  000000                RDER3:  0
1677  001116  000000                RDER4:  0
1678  001120  000000                RDER5:  0
1679  001122  000000                RDER6:  0
1680  001124  000000                RDER7:  0
1681  001126  000000                RDER8:  0
1682
1683                                      ;UNIT DATA ERRORS FORWARD**********
1684
1685  001130  000000                DATER1: 0
1686  001132  000000                        0
1687  001134  000000                        0
1688  001136  000000                        0
1689  001140  000000                        0
1690  001142  000000                        0
1691  001144  000000                        0
1692  001146  000000                        0
1693
1694                                      ;UNIT READ REVERSE ERRORS**********
1695
1696  001150  000000                RDERR1: 0
1697  001152  000000                        0
1698  001154  000000                        0
1699  001156  000000                        0
1700  001160  000000                        0
1701  001162  000000                        0
1702  001164  000000                        0
1703  001166  000000                        0
1704
1705                                      ;UNIT DATA ERRORS REVERSE**********
1706
1707  001170  000000                DEREV1: 0
1708  001172  000000                        0
1709  001174  000000                        0
1710  001176  000000                        0
1711  001200  000000                        0
1712  001202  000000                        0
1713  001204  000000                        0
1714  001206  000000                        0
```

```
1715
1716                                        ;DROPS + PICKS PER CHANNEL PER UNIT**********
1717  001210  000000                BP00:   0
1718          001230
1719  001230  000000                BP10:   .=.+16
1720          001250
1721  001250  000000                BP20:   .=.+16
1722          001270
1723  001270  000000                BP30:   .=.+16
1724          001310
1725  001310  000000                BP40:   .=.+16
1726          001330
1727  001330  000000                BP50:   .=.+16
1728          001350
1729  001350  000000                BP60:   .=.+16
1730          001370
1731  001370  000000                BP70:   .=.+16
1732          001410
1733  001410  000000                BD00:   .=.+16
1734          001430
1735  001430  000000                BD10:   .=.+16
1736          001450
1737  001450  000000                BD20:   .=.+16
1738          001470
1739  001470  000000                BD30:   .=.+16
1740          001510
1741  001510  000000                BD40:   .=.+16
1742          001530
1743  001530  000000                BD50:   .=.+16
1744          001550
1745  001550  000000                BD60:   .=.+16
1746          001570
1747  001570  000000                BD70:   .=.+16
1748          001610
1749
1750
```

```
1751
1752
1753
1754  001610  000000                BT00:   0
1755          001714                        .=.+102
1756  001714  000000                BT01:   0
1757          002020                        .=.+102
1758  002020  000000                BT02:   0
1759          002124                        .=.+102
1760  002124  000000                BT03:   0
1761          002230                        .=.+102
1762  002230  000000                BT04:   0
1763          002334                        .=.+102
1764  002334  000000                BT05:   0
1765          002440                        .=.+102
1766  002440  000000                BT06:   0
1767          002544                        .=.+102
1768  002544  000000                BT07:   0
1769          002650                        .=.+102
1770
1771                                        ;UNIT END OF TAPE COUNTERS 1 PER SLAVE*******
1772
1773  002650  000000                EOTC0:  0
1774  002652  000000                        0
1775  002654  000000                        0
1776  002656  000000                        0
1777  002660  000000                        0
1778  002662  000000                        0
1779  002664  000000                        0
1780  002666  000000                        0
1781
1782                                        ;UNIT READ FORWARD SOFT ERROR*******
1783
1784  002670  000000                RFSOFT: 0
1785  002672  000000                        0
1786  002674  000000                        0
1787  002676  000000                        0
1788  002700  000000                        0
1789  002702  000000                        0
1790  002704  000000                        0
1791  002706  000000                        0
1792
1793                                        ;UNIT READ REVERSE SOFT ERROR*******
1794
1795  002710  000000                RRSOFT: 0
1796  002712  000000                        0
1797  002714  000000                        0
1798  002716  000000                        0
1799  002720  000000                        0
1800  002722  000000                        0
1801  002724  000000                        0
1802  002726  000000                        0
1803
```

```
1804
1805                                    ;UNIT READ FORWARD HARD ERROR*******
1806
1807  002730  000000            RFHARD: 0
1808  002732  000000                    0
1809  002734  000000                    0
1810  002736  000000                    0
1811  002740  000000                    0
1812  002742  000000                    0
1813  002744  000000                    0
1814  002746  000000                    0
1815
1816                                    ;UNIT READ REVERSE HARD ERROR*******
1817
1818  002750  000000            RRHARD: 0
1819  002752  000000                    0
1820  002754  000000                    0
1821  002756  000000                    0
1822  002760  000000                    0
1823  002762  000000                    0
1824  002764  000000                    0
1825  002766  000000                    0
1826
1827                                    ;DATA PATTERN GENERATORS*************
1828
1829  002770  002770            DATBL:  .                       ;ENTRY TABLE
1830  002772  014412            DATA0:  DAT0                    ;EXTERNAL INPUT FROM H/S READER(SEE MAINDEC-11-DZTUF)
1831  002774  014556            DATA1:  DAT1                    ;ALL ONES
1832  002776  014576            DATA2:  DAT2                    ;ALL ZEROS
1833  003000  014602            DATA3:  DAT3                    ;WALKING ONE
1834  003002  014626            DATA4:  DAT4                    ;WALKING ZERO
1835  003004  014636            DATA5:  DAT5                    ;ALTERNATING ONE/ZERO
1836  003006  014644            DATA6:  DAT6                    ;ALTERNATING ZERO/ONE
1837  003010  014652            DATA7:  DAT7                    ;ALTERNATING ONE/ZERO IN ALTERNATING CHARACTERS
1838  003012  014700            DATA10: DAT10                   ;WALKING ONE/ALL ONE IN ALTERNATING CHARACTERS
1839  003014  014730            DATA11: DAT11                   ;ALL BITS 0-377
1840  003016  014750            DATA12: DAT12                   ;ALL BITS 377-0
1841  003020  014772            DATA13: DAT13                   ;ALTERNATING CHARACTERS  0 AND 377
1842  003022  015002            DATA14: DAT14                   ;WALKING ZERO/ALL ZERO IN ALTERNATING CHARACTERS
1843  003024  015032            DATA15: DAT15                   ;AUTO SEQUENCE PATTERN 0,0,-1,-1,-1,0,0
1844
```

```
                                        .EVEN
                                        ;*************************************************************
                                        ;PROGRAM START AND SEQUENCE FORMATTER:
                                        ;
                                        ;THIS ROUTINE IS USED TO PERFORM ALL HOUSEKEEPING,
                                        ;DECIDE WHICH TRANSPORT TO TEST AND ITS AVAILABILITY,
                                        ;LOAD THE WRITE BUFFER WITH THE SELECTED DATA PATTERN,
                                        ;GENERATE ANY RANDOM NUMBER AND THEN EXECUTE
                                        ;THE TEST CYCLE REQUESTED BY THE SWITCH SETTING.
                                        ;AT THE END OF THE TEST CYCLE THE NEXT UNIT IS SELECTED
                                        ;AND CHECKED FOR AVAILABILITY AND THE TEST CYCLE IS
                                        ;EXECUTED ON IT.
                                        ;THE READ WRITE STATS MAY BE PRINTED AT THE END OF
                                        ;EACH TEST CYCLE VIA CONSOLE SWITCH FOURTEEN (14).
                                        ;*************************************************************


                                ;START 200 & 300 **********
003026  012706  000500          START:  MOV     #500,SP         ;++G SET STACK PTR
003032  005037  000734                  CLR     ASEQF           ;CLEAR AUTO SEQUENCE FLAG
003036  005027                          CLR     (PC)+           ;;CLEAR CHAIN INDICATOR
003040  000000                  CHNFLG: .WORD   0               ;;CHAIN MODE INDICATOR
                                                                ;;1/0 = CHAIN/NOT CHAIN MODE
003042  022737  005116  000042          CMP     #$ENDAD,@#42    ;;BRANCH IF LOADED VIA ACT11 CHAIN MODE
003050  001404                          BEQ     50$
003052  005737  000042                  TST     @#42            ;;BRANCH IF IN DUMP MODE
003056  001413                          BEQ     52$
003060  000406                          BR      51$
003062  012737  000176  000606  50$:    MOV     #SWREG,SWR      ;;INVOKE SOFTWARE SWR
003070  012777  100000  175510          MOV     #100000,@SWR    ;;HALT ON ERROR
003076  005237  003040          51$:    INC     CHNFLG          ;;SET CHNFLG = CHAIN MODE
003102  000137  003126                  JMP     3$              ;;GO TO CHAIN ADDRESS
003106                          52$:
003106  122737  000006  000041          CMPB    #6,@#41         ;++G BRANCH IF NOT LOADED VIA TMDP
003114  001010                          BNE     STAUT
003116  012704  027373                  MOV     #MSG120,R4
003122  104000                          TTOUTT                  ;++G ADVISE USER TO REMOVE TMDP
003124  000404                          BR      STAUT           ;++G
003126  005237  000734          3$:     INC     ASEQF           ;++G SET AUTO SEQUENCE FLAG
003132  000137  003262                  JMP     SUSWR           ;CHECK AND SET UP HRD/SOFT SWITCH REG ++ C.W

                                ;START 240 **********
003136  012737  000001  000634  STAUT:  MOV     #1,TINF         ;SET TTY ENTRY FLAG
003144  005037  015150                  CLR     RDFL            ;CLEAR RANDOM DATA FLAG
003150  000405                          BR      STARTB          ;++G

                                ;START 204 **********
003152  005037  000634          STARTC: CLR     TINF            ;CLEAR TTY INPUT FLAG
003156  000434                          BR      STARTD          ;++G

                                ;START 210 **********
003160  005037  000634          STARTA: CLR     TINF            ;CLEAR TTY ENTRY FLAG
003164  012700  000636          STARTB: MOV     #TOB,R0
003170  012701  000037                  MOV     #37,R1
003174  005020                  STARTO: CLR     (R0)+           ;CLEAR FLAGS AND COUNTERS
```

```
1901  003176  005301                        DEC     R1
1902  003200  001375                        BNE     STARTO
1903  003202  012706  000500                MOV     #500,SP         ;SET STACK POINTER
1904  003206  004737  004372                JSR     PC,RANSET       ;GO RESET RANDOM BASE
1905  003212  012700  001050                MOV     #RIY1,R0
1906  003216  012701  000750                MOV     #750,R1
1907  003222  005020              STARTF:   CLR     (R0)+           ;CLEAR STATISTIC COUNTERS
1908  003224  005301                        DEC     R1
1909  003226  001375                        BNE     STARTF
1910  003230  012737  177777  014406        MOV     #-1,PATS        ;PRESET PATTERN
1911  003236  005037  000744                CLR     EOPB1
1912  003242  012737  000001  000654  STARTE: MOV    #1,BLCNTR       ;PRESET BLOCK COUNTER
1913  003250  012706  000500  STARTD:  MOV     #500,SP
1914  003254  012777  000340  175322        MOV     #340,@PSW
1915  003262  013746  000006  SUSWR:   MOV     @#6,-(SP)               ;SAVE VECTORS
1916  003266  013746  000004            MOV     @#4,-(SP)
1917  003272  012737  003312  000004        MOV     #1$,@#4             ;SET UP FOR TIMEOUT
1918  003300  022777  177777  175300        CMP     #-1,@SWR            ;REFERENCE HARDWARE SWITCH REGISTER
1919  003306  001402                        BEQ     2$
1920  003310  000404                        BR      3$
1921  003312  022626              1$:       CMP     (SP)+,(SP)+         ;ADJUST STACK
1922  003314  012737  000176  000606  2$:  MOV     #SWREG,SWR          ;POINT TO SOFTWARE SWITCH REG
1923  003322  012637  000004        3$:   MOV     (SP)+,@#4           ;RESTORE VECTORS
1924  003326  012637  000006            MOV     (SP)+,@#6
1925  003332  022737  000176  000606        CMP     #SWREG,SWR          ;IS SWREG SELECTED
1926  003340  001020                        BNE     4$
1927  003342  005737  000744                TST     EOPB1
1928  003346  001015                        BNE     4$
1929  003350  005037  000744                CLR     EOPB1
1930  003354  022737  005116  000042        CMP     #$ENDAD,@#42    ;ACT MODE? ++ C.W
1931  003362  001402                        BEQ     6$              ;BRANCH - IF YES ++ C.W
1932  003364  004737  024460                JSR     PC,CNTLU        ;CHECK FOR CONTROL G
1933  003370  005737  000734        6$:     TST     ASEQF           ;AUTO SEQ MODE? ++ C.W
1934  003374  001402                        BEQ     4$              ;BRANCH - IF NO ++ C.W
1935  003376  000137  022016                JMP     ASEQO           ;GO DO AUTO SEQ ++ C.W
1936  003402  004737  012400        4$:     JSR     PC,TINP         ;GO GET PARAMETERS FROM TTY
1937  003406  012777  000040  175104        MOV     #40,@CS         ;INITIALIZE
1938  003414  005000              STAUTO:   CLR     R0              ;POINT TO FIRST ENTRY
1939  003416  022760  177777  000746  1$:   CMP     #-1,UN1(R0)     ;++G BRANCH IF LAST ENTRY
1940  003424  001406                        BEQ     2$
1941  003426  042760  100000  000746        BIC     #100000,UN1(R0) ;CLEAR EOT FLAG
1942  003434  062700  000002                ADD     #2,R0           ;POINT TO NEXT UNIT ENTRY
1943  003440  000766                        BR      1$              ;++G CONTINUE CLEARING
1944  003442  013703  005156        2$:     MOV     REOTC,R3
1945  003446  000303                        SWAB    R3
1946  003450  110337  005156                MOVB    R3,REOTC        ;RESTORE EOT CNTR
1947  003454  012777  000100  175126  START1: MOV    #100,@TKS       ;SET TTY INTERRUPT ENABLE
1948  003462  013700  000674                MOV     UNP,R0          ;R0 = UNIT TABLE POINTER
1949  003466  022760  177777  000746  STAR1A: CMP    #-1,UN1(R0)     ;++G BRANCH IF LAST ENTRY
1950  003474  001404                        BEQ     STAR1B          ;IF LAST UNIT IN STRING: BR
1951  003476  016037  000746  000552        MOV     UN1(R0),UDES    ;LOAD NEXT UNIT DESCRIPTION
1952  003504  000446                        BR      START4          ;++G
1953  003506  005237  000654  STAR1B:  INC     BLCNTR          ;BUMP BLOCK COUNTER
1954  003512  005737  000734                TST     ASEQF           ;SEE IF AUTO SEQ
1955  003516  001411                        BEQ     STAR1C          ;IF NOT: BR
1956  003520  023737  000654  000740        CMP     BLCNTR,ABLCNT   ;SEE IF DONE SEQ
```

```
1957  003526  001005                          BNE     STAR1C          ;IF NOT: BR
1958  003530  005037  000654                  CLR     BLCNTR          ;RESET BLOCK CNTR
1959  003534  005037  000674                  CLR     UNP             ;RESET UNIT POINTER
1960  003540  000207                          RTS     PC              ;RETURN TO AUTO SEQ
1961
1962  003542  005037  000674          STAR1C: CLR     UNP
1963  003546  005000                          CLR     R0
1964  003550  016037  000746  000552          MOV     UN1(R0),UDES    ;LOAD FIRST UNIT DESCRIPTION
1965  003556  105777  175024                  TSTB    @SWR            ;++G BRANCH IF NOT RANDOM RECORD
1966  003562  100003                          BPL     START2          ;++G SIZE REQUESTED.
1967  003564  001402                          BEQ     START2          ;IF NOT: BR
1968  003566  004737  012314                  JSR     PC,CCNTR        ;GO GENERATE RANDOM RECORD SIZE
1969  003572  032777  000400  175006  START2: BIT     #400,@SWR       ;SEE IF RANDOM DATA
1970  003600  001402                          BEQ     START3          ;IF NOT: BR
1971  003602  004737  015102                  JSR     PC,DATR         ;GO GENERATE RANDOM DATA
1972  003606  032777  000100  174772  START3: BIT     #100,@SWR       ;SEE IF RANDOM RECORD COUNT
1973  003614  001402                          BEQ     START4          ;IF NOT: BR
1974  003616  004737  012354                  JSR     PC,RCNTR        ;GO GENERATE RANDOM RECORD COUNT
1975  003622  005760  000746          START4: TST     UN1(R0)         ;++G BRANCH IF NOT AT EOT
1976  003626  100002                          BPL     STAR40          ;IF NOT: BR
1977  003630  000137  004360                  JMP     START7          ;ELSE GO TO NEXT UNIT
1978  003634  013777  000550  174656  STAR40: MOV     DVN,@CS         ;SET DRIVE NUMBER
1979  003642  013777  000552  174672          MOV     UDES,@C2        ;SET UNIT NUMBER
1980  003650  105777  174646                  TSTB    @DS             ;++G BRANCH IF UNIT AVAIL
1981  003654  100412                          BMI     STAR4A
1982  003656  005337  000666                  DEC     STAL
1983  003662  001357                          BNE     START4          ;AWAIT TUR
1984  003664  004737  022736                  JSR     PC,PAPRT        ;PRINT HEADER
1985  003670  012704  026074                  MOV     #MSG49,R4
1986  003674  104000                          TTOUTT  ;PRINT NOT AVAIL
1987  003676  104006                          STOPP                   ;STOP
1988  003700  000750                          BR      START4          ;++G RETRY
1989  003702  013746  000552          STAR4A: MOV     UDES,-(SP)      ;GET UNIT DESCRIPTION
1990  003706  042716  175400                  BIC     #175400,(SP)    ;++G CLEAR ALL BUT FORMAT BITS
1991                                          ;CMP    #1700,(SP)+     ;++G BRANCH IF NRZ
1992  003712  032726  002000                  BIT     #2000,(SP)+     ;++H BRANCH IF NZR
1993  003716  001406                          BEQ     1$              ;++G
1994  003720  032777  000040  174574          BIT     #40,@DS         ;++G BRANCH IF SLAVE IN PE FORMAT
1995  003726  001002                          BNE     1$              ;++G
1996  003730  000137  004360                  JMP     START7          ;++G GO TO NEXT UNIT
1997  003734  004737  014200          1$:     JSR     PC,DSUP         ;GO SET UP WRITE DATA
1998  003740  004737  005160                  JSR     PC,RWND         ;REWIND
1999  003744  004737  005522                  JSR     PC,WRITE        ;WRITE
2000  003750  013737  000576  000666          MOV     TSTAL,STAL      ;SET TURN AROUND DELAY
2001  003756  004737  012304                  JSR     PC,STALL        ;DELAY
2002  003762  004737  007414                  JSR     PC,RSEQ         ;GO TO READ SEQUENCER
2003  003766  013737  000576  000666          MOV     TSTAL,STAL      ;SET TURN AROUND DELAY
2004  003774  004737  012304                  JSR     PC,STALL        ;DELAY
2005  004000  032777  040000  174600          BIT     #40000,@SWR     ;SEE IF SHOULD PRINT STATISTICS
2006  004006  001541                          BEQ     START5          ;IF NOT: BR
2007  004010  012700  000001                  MOV     #1,R0           ;SET RECORD COUNTER TO 1
2008  004014  004737  022736                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
2009  004020  004737  004030                  JSR     PC,STP          ;GO PRINT STATS
2010  004024  000137  004276                  JMP     STPX
2011  004030  004737  017250          STP:    JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
2012  004034  012704  026307                  MOV     #MSG65,R4
```

```
2013  004040  104000
2014  004042  013704  000674
2015  004046  016403  001050
2016  004052  104002
2017  004054  012704  026460
2018  004060  104000
2019  004062  013704  000674
2020  004066  016403  001070
2021  004072  104002
2022  004074  012704  026447
2023  004100  104000
2024  004102  013704  000674
2025  004106  016403  001110
2026  004112  104002
2027  004114  012704  027255
2028  004120  104000
2029  004122  013704  000674
2030  004126  016403  002670
2031  004132  104002
2032  004134  012704  027266
2033  004140  104000
2034  004142  013704  000674
2035  004146  016403  002730
2036  004152  104002
2037  004154  012704  026540
2038  004160  104000
2039  004162  013704  000674
2040  004166  016403  001130
2041  004172  104002
2042  004174  012704  026343
2043  004200  104000
2044  004202  013704  000674
2045  004206  016403  001150
2046  004212  104002
2047  004214  012704  027255
2048  004220  104000
2049  004222  013704  000674
2050  004226  016403  002710
2051  004232  104002
2052  004234  012704  027266
2053  004240  104000
2054  004242  013704  000674
2055  004246  016403  002750
2056  004252  104002
2057  004254  012704  026527
2058  004260  104000
2059  004262  013704  000674
2060  004266  016403  001170
2061  004272  104002
2062  004274  000207
2063  004276  005237  000726
2064  004302  004737  007324
2065  004306  005037  000726
2066  004312  017700  174270
2067  004316  042700  177762
2068  004322  022700  000015

TTOUTT  ;PRINT RETRY TOTAL
        MOV     UNP,R4
        MOV     RTY1(R4),R3
        OCTPP                   ;PRINT RETRIES
        MOV     #MSG73,R4
        TTOUTT                  ;PRINT WRITE ERROR TAG
        MOV     UNP,R4
        MOV     WTER1(R4),R3
        OCTPP                   ;PRINT WRITE ERRORS
        MOV     #MSG72,R4
        TTOUTT                  ;PRINT READ FORWARD ERROR TAG
        MOV     UNP,R4
        MOV     RDER1(R4),R3
        OCTPP                   ;PRINT READ FORWARD ERRORS
        MOV     #MSG113,R4
        TTOUTT                  ;PRINT SOFT TAG
        MOV     UNP,R4
        MOV     RFSOFT(R4),R3
        OCTPP                   ;PRINT FORWARD SOFT ERRORS
        MOV     #MSG114,R4
        TTOUTT                  ;PRINT HARD TAG
        MOV     UNP,R4
        MOV     RFHARD(R4),R3
        OCTPP                   ;PRINT HARD FORWARE ERRORS
        MOV     #MSG77,R4
        TTOUTT                  ;PRINT DATA ERROR FORWARD TAG
        MOV     UNP,R4
        MOV     DATER1(R4),R3
        OCTPP                   ;PRINT DATA ERROR FORWARD NUMBER
        MOV     #MSG68,R4
        TTOUTT                  ;PRINT READ ERROR REVERSE TAG
        MOV     UNP,R4
        MOV     RDERR1(R4),R3
        OCTPP                   ;PRINT REVESE ERROR NUMBER
        MOV     #MSG113,R4
        TTOUTT                  ;PRINT SOFT TAG
        MOV     UNP,R4
        MOV     RRSOFT(R4),R3
        OCTPP                   ;PRINT REVERSE SOFT ERROR
        MOV     #MSG114,R4
        TTOUTT                  ;PRINT HARD TAG
        MOV     UNP,R4
        MOV     RRHARD(R4),R3
        OCTPP
        MOV     #MSG76,R4
        TTOUTT                  ;PRINT DATA ERROR REVERSE TAG
        MOV     UNP,R4
        MOV     DEREV1(R4),R3
        OCTPP                   ;PRINT DATA REVERSE ERROR NUMBER
        RTS     PC              ;RETURN
STPX:   INC     BTSTF           ;SET STAT ONLY PRINT
        JSR     PC,BTPRT        ;PRINT BAD TAPE STATS
        CLR     BTSTF           ;CLEAR FLAG
START5: MOV     @SWR,R0         ;LOAD SWR
        BIC     #177762,R0      ;MASK READ/WRITE SWITCHES
        CMP     #15,R0          ;SEE IF HAVE READ OR WRITE
```

```
 2069  004326  001417                    BEQ     START8          ;IF NOT: BR
 2070  004330  105777  174166  START6:   TSTB    @DS             ;++G BRANCH IF HAVE UNIT READY
 2071  004334  100411                    BMI     START7          ;++G
 2072  004336  005337  000666            DEC     STAL
 2073  004342  001372                    BNE     START6          ;DELAY FOR TUR
 2074  004344  004737  022736            JSR     PC,PAPRT        ;PRINT HEADER
 2075  004350  012704  026074            MOV     #MSG49,R4
 2076  004354  104000                    TTOUTT                  ;PRINT NOT AVAIL
 2077  004356  104006                    STOPP                           ;STOP
 2078  004360  062737  000002  000674  START7:  ADD     #2,UNP          ;POINT TO NEXT UNIT
 2079  004366  000137  003454          START8:  JMP     START1          ;CONTINUE
 2080
 2081                                    ;RANDOM BASE RESET**********
 2082
 2083  004372  012737  153624  000624  RANSET:  MOV     #153624,RANBAS  ;RESET BASE
 2084  004400  012737  032561  000626           MOV     #32561,RANSAV   ;RESET BUFFER
 2085  004406  013737  000630  000554           MOV     RCSAV,RCNT      ;RESET RECORD COUNT
 2086  004414  013737  000632  000556           MOV     FCSAV,FMCNT     ;RESET FRAME COUNT
 2087  004422  000207                    RTS     PC
 2088
```

```
2089                                    ;*************************************************************
2090                                    ;REWIND FROM EOT:
2091                                    ;
2092                                    ;WHEN ANY TRANSPORT BEING TESTED REACHES END OF TAPE
2093                                    ;DURING A READ OR WRITE OPERATION, IT WILL BE REWOUND
2094                                    ;AND FLAGGED AS UNAVAILABLE UNTIL ALL AVAILABLE UNITS
2095                                    ;HAVE REACHED EOT AT WHICH TIME ALL TESTING WILL BE RESUMED
2096                                    ;AT A BLOCK COUNT OF ONE (1).  A MESSAGE WILL BE
2097                                    ;PRINTED ON THE SUPERVISORS CONSOLE AS EACH UNIT REACHES
2098                                    ;EOT AND IS REWOUND.
2099                                    ;*************************************************************
2100
2101  004424  013777  000552  174110  REOT:   MOV     UDES,@C2        ;LOAD COMMAND REGISTER
2102  004432  012777  000011  174050          MOV     #11,@C1         ;DRIVE CLEAR
2103  004440  105777  174056          1$:     TSTB    @DS             ;++G WAIT FOR DRIVE READY
2104  004444  100375                          BPL     1$              ;AWAIT DRY
2105  004446  012777  000007  174034          MOV     #7,@C1          ;START REWIND
2106  004454  005737  000724                  TST     BTFLG           ;SEE IF BAD TAPE OVERFLOW REWIND
2107  004460  001004                          BNE     REOT1A          ;IF SO: BR
2108  004462  013700  000660                  MOV     EOTREC,R0
2109  004466  042700  100000                  BIC     #100000,R0      ;SET RECORD NUMBER OF EOT
2110  004472  005037  000660          REOT1A: CLR     EOTREC          ;++G CLEAR EOT IND & REC CTR
2111  004476  004737  022736                  JSR     PC,PAPRT        ;PRINT HEADER
2112  004502  022737  000002  000724          CMP     #2,BTFLG        ;SEE IF POSITION ERROR
2113  004510  001003                          BNE     REOT1B          ;IF NOT: BR
2114  004512  012704  027146                  MOV     #MSG109,R4      ;SET POSITION ERROR MSG
2115  004516  000406                          BR      REOT1F
2116  004520  022737  000001  000724  REOT1B: CMP     #1,BTFLG        ;SEE IF BAD TAPE OVERFLOW
2117  004526  001004                          BNE     REOT1C          ;IF NOT: BR
2118  004530  012704  026757                  MOV     #MSG106,R4      ;SET BAD TAPE OVERFLOW MSG
2119  004534  104000                  REOT1F: TTOUTT                  ;PRINT REWIND REASON
2120  004536  000412                          BR      REOT1E
2121  004540  012704  025126          REOT1C: MOV     #MSG20,R4       ;SET EOT MSG
2122  004544  104000                  REOT1D: TTOUTT                  ;PRINT MSG
2123  004546  013704  000674                  MOV     UNP,R4
2124  004552  005264  002650                  INC     EOTC0(R4)       ;BUMP CNTR
2125  004556  016403  002650                  MOV     EOTC0(R4),R3
2126  004562  104002                          OCTPP                   ;PRINT EOT CNTR
2127  004564  012704  027004          REOT1E: MOV     #MSG16A,R4
2128  004570  104000                          TTOUTT                  ;PRINT RESTART MSG
2129  004572  005037  000724                  CLR     BTFLG           ;CLEAR BAD TAPE FLAG
2130  004576  004737  004030                  JSR     PC,STP          ;PRINT STATS
2131  004602  004737  007324                  JSR     PC,BTPRT        ;PRINT BAD TAPE STATS
2132  004606  105777  173710          REOT2:  TSTB    @DS             ;++G BRANCH IF UNIT IS READY
2133  004612  100414                          BMI     REOT2A
2134  004614  005337  000666                  DEC     STAL
2135  004620  001372                          BNE     REOT2           ;WAIT DRY
2136  004622  012737  024767  000652          MOV     #MSG6,EMADDR
2137  004630  004737  022736                  JSR     PC,PAPRT        ;PRINT HEADER
2138  004634  012704  026251                  MOV     #MSG60,R4
2139  004640  104000                          TTOUTT                  ;PRINT NO DRIVE READY
2140  004642  104006                          STOPP
2141  004644  105337  005156          REOT2A: DECB    REOTC           ;SEE IF LAST UNIT TO REACH EOT
2142  004650  001410                          BEQ     REOT3           ;IF SO: BR
2143  004652  013700  000674                  MOV     UNP,R0
2144  004656  052760  100000  000746          BIS     #100000,UN1(R0) ;SET EOT FLAG
```

```
2145  004664  005726                          TST     (SP)+           ;RESET STACK POINTER
2146  004666  000137  004360                  JMP     START7          ;GO TO NEXT UNIT
2147  004672  000337  005156          REOT3:  SWAB    REOTC
2148  004676  013700  005156                  MOV     REOTC,R0
2149  004702  000337  005156                  SWAB    REOTC
2150  004706  110037  005156                  MOVB    R0,REOTC        ;RESTORE EOT UNIT COUNTER
2151  004712  005037  000674                  CLR     UNP
2152  004716  013700  000674                  MOV     UNP,R0          ;POINT TO FIRST UNIT
2153  004722  016037  000746  000552  REOT4:  MOV     UN1(R0),UDES    ;LOAD UNIT DESCRIPTION
2154  004730  013777  000552  173604          MOV     UDES,@C2        ;LOAD COMMAND REGISTER
2155  004736  032777  020000  173556  REOT5:  BIT     #20000,@DS
2156  004744  001374                          BNE     REOT5           ;AWAIT PIP RESET
2157  004746  032777  000002  173546          BIT     #2,@DS          ;SEE IF HAVE BOT
2158  004754  001012                          BNE     REOT6           ;IF SO: BR
2159  004756  012700  000001                  MOV     #1,R0
2160  004762  004737  022736                  JSR     PC,PAPRT        ;PRINT HEADER
2161  004766  012704  026042                  MOV     #MSG48,R4
2162  004772  104000                          TTOUTT                  ;PRINT BOT ERROR
2163  004774  104006                          STOPP
2164  004776  013700  000674                  MOV     UNP,R0
2165  005002  042760  100000  000746  REOT6:  BIC     #100000,UN1(R0) ;CLEAR EOT FLAG
2166  005010  062737  000002  000674          ADD     #2,UNP
2167  005016  013700  000674                  MOV     UNP,R0          ;POINT TO NEXT UNIT
2168  005022  022760  177777  000746          CMP     #-1,UN1(R0)     ;++G BRANCH IF NOT LAST UNIT
2169  005030  001334                          BNE     REOT4           ;++
2170  005032  005037  000674          REOT7:  CLR     UNP             ;CLEAR UNIT POINTER
2171  005036  005037  000634                  CLR     TINF            ;CLEAR TTY INPUT FLAG
2172  005042  005737  000734                  TST     ASEQF           ;SEE IF AUTO SEQ
2173  005046  001402                          BEQ     REOTX           ;IF NOT: BR
2174  005050  005726                          TST     (SP)+           ;RESET STACK POINTER
2175  005052  000412                          BR      TEND            ;GO DO END OF PASS ++ C.W
2176
2177  005054  004737  004372          REOTX:  JSR     PC,RANSET       ;GO RESET RANDOM BASE
2178  005060  012737  177777  014406          MOV     #-1,PATS        ;PRESET PATTERN
2179  005066  005037  015150                  CLR     RDFL            ;CLEAR RANDOM FLAG
2180  005072  005737  000570                  TST     SPFLG           ;SEE IF SINGLE PASS
2181  005076  001420                          BEQ     REOTXX          ;IF NOT: BR
2182  005100  012704  026652          TEND:   MOV     #MSG100,R4
2183  005104  104000                          TTOUTT                  ;PRINT END OF PASS
2184  005106  013700  000042                  MOV     @#42,R0         ;GET ACT11 RETURN ADDRESS
2185  005112  001405                          BEQ     HERE            ;BRANCH IF NOT ACT11
2186  005114  000005                          RESET
2187  005116  004710                  SENDAD: JSR     PC,(R0)
2188  005120  000240                          NOP
2189  005122  000240                          NOP
2190  005124  000240                          NOP
2191  005126  005737  000734          HERE:   TST     ASEQF           ;AUTO MODE? ++ C.W
2192  005132  001401                          BEQ     1$              ;BRANCH - IF NO ++ C.W
2193  005134  000207                          RTS     PC              ;RETURN TO AUTO SEQ ++ C.W
2194  005136  104006                  1$:     STOPP
2195  005140  012704  026652          REOTXX: MOV     #MSG100,R4      ;GET END OF PASS MESSAGE ++ C.W
2196  005144  104000                          TTOUTT                  ;PRINT MESSAGE ++ C.W
2197  005146  005237  000744                  INC     EOPB1
2198  005152  000137  003242                  JMP     START           ;RESTART AT BLOCK NUMBER ONE
2199  005156  000000          REOTC:  0                               ;EOT UNIT COUNTER
```

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

005160  032777  001000  173420  RWND:   BIT     #1000,@SWR      ;SEE IF SHOULD REWIND
005166  001001                          BNE     RWNDA           ;IF SO: BR
005170  000207                          RTS     PC              ;ELSE EXIT
005172  013737  000674  000714  RWNDA:  MOV     UNP,UPS         ;SAVE UNIT POINTER
005200  005037  000674                  CLR     UNP             ;CLEAR POINTER
005204  005037  000660                  CLR     EOTREC          ;CLEAR EDT FLAG
005210  000337  005156                  SWAB    REOTC
005214  013700  005156                  MOV     REOTC,R0
005220  000337  005156                  SWAB    REOTC
005224  110037  005156                  MOVB    R0,REOTC        ;RESTORE EOT UNIT COUNTER
005230  013700  000674          RWND0:  MOV     UNP,R0          ;POINT TO UNIT ENTRY
005234  022760  177777  000746          CMP     #-1,UN1(R0)     ;++G BRANCH IF LAST ENTRY
005242  001445                          BEQ     RWND2           ;IF SO: BR
005244  005760  000746                  TST     UN1(R0)         ;++G BRANCH IF ALREADY REWINDING
005250  100433                          BMI     RWND1A          ;++G
005252  016037  000746  000552          MOV     UN1(R0),UDES    ;SET UNIT DESCRIPTION
005260  013777  000552  173254          MOV     UDES,@C2        ;LOAD COMMAND REGISTER
005266  012777  000011  173214          MOV     #11,@C1         ;DRIVE CLEAR
005274  012777  000007  173206          MOV     #7,@C1          ;START REWIND
005302  105777  173214          RWND1:  TSTB    @DS             ;++G WAIT FOR DRIVE READY
005306  100414                          BMI     RWND1A          ;IF DRY: BR
005310  005337  000666                  DEC     STAL
005314  001372                          BNE     RWND1           ;AWAIT DRY
005316  012737  024767  000652          MOV     #MSG6,EMADDR
005324  004737  022736                  JSR     PC,PAPRT        ;PRINT HEADER
005330  012704  026372                  MOV     #MSG70,R4
005334  104000                          TTOUTT                  ;PRINT NO DRIVE READY
005336  104006                          STOPP
005340  042760  100000  000746  RWND1A: BIC     #100000,UN1(R0) ;CLEAR EOT FLAG
005346  062737  000002  000674          ADD     #2,UNP          ;BUMP POINTER
005354  000725                          BR      RWND0           ;++G DO NEXT UNIT
005356  005037  000674          RWND2:  CLR     UNP             ;CLEAR POINTER
005362  013700  000674          RWND3:  MOV     UNP,R0          ;POINT TO UNIT ENTRY
005366  022760  177777  000746          CMP     #-1,UN1(R0)     ;++G BRANCH IF LAST ENTRY
005374  001436                          BEQ     RWNDX           ;IF SO: BR
005376  016037  000746  000552          MOV     UN1(R0),UDES    ;SET UNIT DESCRIPTION
005404  013777  000552  173130          MOV     UDES,@C2        ;LOAD COMMAND REGISTER
005412  032777  020000  173102  RWND4:  BIT     #20000,@DS
005420  001374                          BNE     RWND4           ;AWAIT PIP RESET
005422  032777  000002  173072          BIT     #2,@DS          ;SEE IF HAVE BOT
005430  001407                          BEQ     RWND6           ;IF NOT: BR
005432  062737  000002  000674  RWND5:  ADD     #2,UNP          ;BUMP POINTER
005440  012777  000011  173042          MOV     #11,@C1         ;DRIVE CLEAR
005446  000745                          BR      RWND3           ;++G DO NEXT UNIT
005450  012700  000001          RWND6:  MOV     #1,R0
005454  004737  022736                  JSR     PC,PAPRT        ;PRINT HEADER
005460  012704  026042                  MOV     #MSG48,R4
```

SEQ 0050

```
2256  005464  104000                          TTOUTT                  ;PRINT NO BOT
2257  005466  104006                          STOPP
2258  005470  000760                          BR      RWND5           ;++G DO NEXT UNIT
2259  005472  013737  000714  000674  RWNDX:  MOV     UPS,UNP         ;RESTORE UNIT POINTER
2260  005500  013700  000674                  MOV     UNP,R0
2261  005504  016037  000746  000552          MOV     UN1(R0),UDES    ;RESET UNIT DESCRIPTION
2262  005512  013777  000552  173022          MOV     UDES,@C2
2263  005520  000207                          RTS     PC              ;RETURN TO TEST
2264
```

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

005522  032777  000001  173056  WRITE:  BIT     #1,@SWR          ;SEE IF SHOULD WRITE
005530  001402                          BEQ     WRTE
005532  000137  006320                  JMP     WEX              ;IF NOT: BR
005536  013700  000554          WRTE:   MOV     RCNT,R0          ;R0=RECORD COUNT
005542  012737  024762  000652  W0:     MOV     #MSG5,EMADDR     ;SET ERROR MSG ADDRESS
005550  013777  000556  172740          MOV     FMCNT,@FC        ;LOAD CHAR COUNT
005556  012777  027464  172730          MOV     #WDATA,@BA       ;SET DATA ADDR
005564  112737  000060  000672          MOVB    #60,MTC1         ;SET WRITE OP COMMAND
005572  012737  005604  000662          MOV     #W1,RTRN         ;SET RETURN ADDRESS
005600  000137  021216                  JMP     TAPG             ;GO EXECUTE COMMAND
005604  032777  002000  172710  W1:     BIT     #2000,@DS        ;SEE IF EOT
005612  001414                          BEQ     W2               ;IF NOT AT EOT: BR
005614  005737  000660                  TST     EOTREC           ;++G BRANCH IF WRITTEN PAST EOT
005620  100411                          BMI     W2               ;++G
005622  010037  000660                  MOV     R0,EOTREC        ;SAVE EOT RECORD NUMBER
005626  052737  100000  000660          BIS     #100000,EOTREC   ;++G SET EOT FLAG
005634  005337  000660                  DEC     EOTREC           ;++G ADJUST RECORD COUNT
005640  012700  000002                  MOV     #2,R0            ;++G SET R0 TO WRITE 1 MORE RECORD
005644  032777  010000  172734  W2:     BIT     #10000,@SWR      ;SEE IF SHOULD CHECK ERRORS
005652  001002                          BNE     W3               ;IF NOT: BR
005654  004737  017406                  JSR     PC,ERCHK         ;GO CHECK ERRORS
005660  013737  000574  000666  W3:     MOV     WSTAL,STAL       ;SET DELAY
005666  004737  012304                  JSR     PC,STALL         ;DELAY
```

```
2321  005672  005737  000712                TST     RTYFL           ;SEE IF RETRY TIME
2322  005676  001401                        BEQ     W3A             ;IF NOT: BR
2323  005700  000207                        RTS     PC              ;ELSE RETURN
2324  005702  005737  000706        W3A:    TST     SERFL           ;SEE IF WRITE ERROR
2325  005706  001450                        BEQ     W5              ;IF NOT: BR
2326  005710  013704  000674                MOV     UNP,R4
2327  005714  005264  001070                INC     WTER1(R4)       ;BUMP WRITE ERROR
2328  005720  005037  000706                CLR     SERFL           ;CLEAR STATUS ERROR FLAG
2329  005724  032777  000020  172654        BIT     #20,@SWR        ;SEE IF RETRY
2330  005732  001436                        BEQ     W5              ;IF NOT: BR
2331  005734  013703  000722                MOV     ERSAV,R3
2332  005740  042703  102700                BIC     #102700,R3      ;MASK UNRECOVERABLE ERROR
2333  005744  001410                        BEQ     W4              ;IF SO: BR
2334  005746  004737  022736                JSR     PC,PAPRT        ;PRINT HEADER
2335  005752  012704  026551                MOV     #MSG78,R4
2336  005756  104000                        TTOUTT                  ;PRINT NON-RETRYABLE ERROR TAG
2337  005760  004737  011260                JSR     PC,NRTP         ;PRINT ER FOR NON-RETRYABLE
2338  005764  000421                        BR      W5
2339  005766  013704  000674        W4:     MOV     UNP,R4
2340  005772  005264  001050                INC     RTY1(R4)        ;BUMP RETRY CNTR
2341  005776  032777  002000  172602        BIT     #2000,@SWR      ;SEE IF PRINT ERRORS
2342  006004  001003                        BNE     W4A             ;IF NOT: BR
2343  006006  012704  026265                MOV     #MSG64,R4
2344  006012  104000                        TTOUTT                  ;PRINT ORIGINAL ERROR TAG
2345  006014  005037  000702        W4A:    CLR     RTCNT           ;CLEAR RETRY NUMBER
2346  006020  005037  000700                CLR     RPCNT           ;CLEAR REPEAT COUNTER
2347  006024  004737  006362                JSR     PC,WRTY         ;GO RETRY WRITE ERROR
2348  006030  005037  000712        W5:     CLR     RTYFL           ;CLEAR RETRY COUNTER
2349  006034  005300                        DEC     R0              ;SEE IF DONE ALL
2350  006036  001241                        BNE     W0              ;IF NOT: BR
2351  006040  005737  000564        W6:     TST     TMEX            ;SEE IF TM
2352  006044  001525                        BEQ     WEX             ;IF NOT: BR
2353  006046  005237  000676                INC     TMFLG           ;SET TM FLAG
2354  006052  012737  026172  000652  WTM:   MOV     #MSG54,EMADDR   ;POINT TO TM ERROR MSG
2355  006060  012737  000026  000672        MOV     #26,MTC1        ;SET TM OP CODE
2356  006066  012777  000000  172422        MOV     #0,@FC          ;LOAD FRAME COUNTER
2357  006074  012777  027464  172412        MOV     #WDATA,@BA      ;LOAD BUS ADDRESS
2358  006102  012737  006114  000662        MOV     #WTM0,RTRN      ;SAVE RETURN ADDRESS
2359  006110  000137  021216                JMP     TAPG            ;WRITE TM
2360  006114  032777  010000  172464  WTM0:  BIT     #10000,@SWR     ;SEE IF SHOULD CHECK ERRORS
2361  006122  001076                        BNE     WEX
2362  006124  032777  000004  172370        BIT     #4,@DS          ;SEE IF TM STATUS
2363  006132  001011                        BNE     WTM1            ;IF SO: BR
2364  006134  012737  027464  021132        MOV     #WDATA,CADER    ;SET EXPT BUS ADDRESS
2365  006142  012737  000001  021140        MOV     #1,DRVER        ;INDICATE ERROR
2366  006150  004737  020234                JSR     PC,ERPT         ;PRINT TM ERROR
2367  006154  000404                        BR      WTM2
2368  006156  012703  027464        WTM1:   MOV     #WDATA,R3       ;SET EXPT ADDRESS
2369  006162  004737  017504                JSR     PC,ER2          ;GO CHECK FOR OTHER ERRORS
2370  006166  005737  000712        WTM2:   TST     RTYFL           ;SEE IF RETRY
2371  006172  001401                        BEQ     WTM3            ;IF NOT: BR
2372  006174  000207                        RTS     PC              ;ELSE RETURN TO RETRY ROUTINE
2373  006176  005737  000706        WTM3:   TST     SERFL           ;SEE IF WRITE ERROR
2374  006202  001446                        BEQ     WEX             ;IF NOT: BR
2375  006204  013704  000674                MOV     UNP,R4
2376  006210  005264  001070                INC     WTER1(R4)       ;BUMP WRITE ERROR
```

```
2377  006214  032777  000020  172364          BIT     #20,@SWR        ;SEE IF SHOULD RETRY
2378  006222  001436                          BEQ     WEX             ;IF NOT: BR
2379  006224  013703  000722                  MOV     ERSAV,R3
2380  006230  042703  102700                  BIC     #102700,R3      ;MASK UNRECOVERABLE ERROR
2381  006234  001410                          BEQ     WTM4            ;IF SO: BR
2382  006236  004737  022736                  JSR     PC,PAPRT        ;PRINT HEADER
2383  006242  012704  026551                  MOV     #MSG78,R4
2384  006246  104000                          TTOUTT                  ;PRINT UNRETRYABLE TAG
2385  006250  004737  011260                  JSR     PC,NRTP         ;PRINT ER FOR NON-RETRYABLE
2386  006254  000421                          BR      WEX
2387  006256  005037  000700          WTM4:   CLR     RPCNT           ;CLEAR REPEAT CNTR
2388  006262  013704  000674                  MOV     UNP,R4
2389  006266  005264  001050                  INC     RTY1(R4)        ;BUMP RETRY CNTR
2390  006272  005037  000702                  CLR     RTCNT           ;CLEAR RETRY CNTR
2391  006276  032777  002000  172302          BIT     #2000,@SWR      ;SEE IF PRINT ERRORS
2392  006304  001003                          BNE     WTM4A           ;IF NOT: BR
2393  006306  012704  026265                  MOV     #MSG64,R4
2394  006312  104000                          TTOUTT                  ;PRINT ORIGINAL ERROR TAG
2395  006314  004737  006362          WTM4A:  JSR     PC,WRTY         ;GO DO RETRY
2396  006320  005037  000712          WEX:    CLR     RTYFL           ;CLEAR RETRY FLAG
2397  006324  005037  000676                  CLR     TMFLG           ;CLEAR TAPE MARK FLAG
2398  006330  005737  000660                  TST     EOTREC          ;++G BRANCH IF NOT AT EOT
2399  006334  100011                          BPL     WRWX            ;++G
2400  006336  017703  172244          WRW:    MOV     @SWR,R3
2401  006342  042703  177763                  BIC     #177763,R3
2402  006346  022703  000014                  CMP     #14,R3          ;SEE IF WRITE ONLY
2403  006352  001002                          BNE     WRWX            ;IF NOT: BR
2404  006354  000137  004424                  JMP     REOT            ;ELSE REWIND
2405  006360  000207          WRWX:   RTS     PC                      ;EXIT
```

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

006362  012737  000001  000712  WRTY:   MOV     #1,RTYFL        ;SET RETRY FLAG
006370  004737  006764          WRTY0:  JSR     PC,WRTSB        ;GO SPACE REVERSE FOR REPEAT
006374  005737  000676                  TST     TMFLG           ;SEE IF TAPE MARK TIME
006400  001003                          BNE     WRTYTM          ;IF SO: BR
006402  004737  005542                  JSR     PC,WO           ;REWRITE RECORD
006406  000402                          BR      WRTYR           ;GO ON
006410  004737  006052          WRTYTM: JSR     PC,WTM          ;GO WRITE TAPE MARK AGAIN
006414  005737  000706          WRTYR:  TST     SERFL           ;REWRITE GOOD
006420  001024                          BNE     WRTY2           ;IF NOT: BR
006422  005237  000700                  INC     RPCNT           ;BUMP REPEAT COUNTER
006426  022737  000004  000700        CMP     #4,RPCNT        ;SEE IF FOUR GOOD REPEATS
006434  001355                          BNE     WRTY0           ;IF NOT: REPEAT
006436  032777  002000  172142        BIT     #2000,@SWR      ;SEE IF PRINT
006444  001011                          BNE     WRTY1           ;IF NOT: BR
006446  012704  026744                  MOV     #MSG105,R4
006452  104000                          TTOUTT                  ;PRINT RECOVERED MESSAGE
006454  012704  026307                  MOV     #MSG65,R4
006460  104000                          TTOUTT                  ;PRINT RETRY TAG
006462  013703  000702                  MOV     RTCNT,R3
006466  104002                          OCTPP                   ;PRINT RETRY NUMBER
006470  000207                  WRTY1:  RTS     PC              ;RESUME TESTING
006472  005037  000646          WRTY2:  CLR     TEMP3           ;++G CLEAR RECOVERABLE ERROR FLAG
006476  013703  000722                  MOV     ERSAV,R3        ;GET ER
006502  042703  102700                  BIC     #102700,R3      ;MASK RECOVERABLE BITS
006506  001413                          BEQ     WRTY2A          ;IF RECOVERABLE: BR
006510  004737  022736                  JSR     PC,PAPRT        ;PRINT HEADER
006514  012704  026551                  MOV     #MSG78,R4
006520  104000                          TTOUTT                  ;PRINT NON-RECOVERABLE MSG
006522  004737  011260                  JSR     PC,NRTP         ;PRINT ER
006526  012737  000001  000646        MOV     #1,TEMP3        ;SET FLAG
006534  000407                          BR      WRTY2B
006536  032777  002000  172042  WRTY2A: BIT     #2000,@SWR      ;SEE IF PRINT
006544  001025                          BNE     WRTY3           ;IF NOT: BR
006546  012704  027200                  MOV     #MSG110,R4
006552  104000                          TTOUTT                  ;PRINT BAD TAPE SUSPECT
006554  012704  026307          WRTY2B: MOV     #MSG65,R4
006560  104000                          TTOUTT                  ;PRINT RETRY TAG
006562  013703  000702                  MOV     RTCNT,R3
006566  104002                          OCTPP                   ;PRINT RETRY NUMBER
006570  012704  027222                  MOV     #MSG111,R4
006574  104000                          TTOUTT                  ;PRINT REPEAT TAG
006576  013703  000700                  MOV     RPCNT,R3
006602  104002                          OCTPP                   ;PRINT REPEAT NUMBER
006604  005737  000646                  TST     TEMP3           ;SEE IF DID NON-RECOVERABLE
006610  001403                          BEQ     WRTY3           ;IF NOT: BR
006612  005037  000646                  CLR     TEMP3           ;CLEAR FLAG
006616  000207                          RTS     PC              ;EXIT
006620  005737  000702          WRTY3:  TST     RTCNT           ;SEE IF FIRST RETRY
006624  001004                          BNE     WRTY3A          ;IF NOT: BR
006626  013704  000674                  MOV     UNP,R4
006632  005364  001070                  DEC     WTER1(R4)       ;DECREMENT WRITE ERROR CNTR
```

```
2462  006636  013704  000674          WRTY3A: MOV     UNP,R4          ;GET UNIT NUMBER
2463  006642  016437  001030  000730          MOV     BTADDR(R4),BTPT ;GET ADDRESS OF UNIT BAD TAPE CNTR
2464  006650  017704  172054                  MOV     @BTPT,R4        ;GET COUNTER
2465  006654  005724                          TST     (R4)+           ;SET POINTER OFFSET
2466  006656  010477  172046                  MOV     R4,@BTPT
2467  006662  013703  000730                  MOV     BTPT,R3
2468  006666  060304                          ADD     R3,R4           ;SET ABSOLUTE POINTER
2469  006670  013714  000654                  MOV     BLCNTR,(R4)     ;SET BLOCK NUMBER
2470  006674  062704  000040                  ADD     #40,R4          ;ADD RCNT OFFSET
2471  006700  013714  000554                  MOV     RCNT,(R4)
2472  006704  160014                          SUB     R0,(R4)         ;SET RECORD NUMBER
2473  006706  005214                          INC     (R4)            ;CORRECT RECORD NUMBER
2474  006710  022777  000040  172012          CMP     #40,@BTPT       ;SEE IF TOO MANY BAD SPOTS
2475  006716  001002                          BNE     WRTY4           ;IF NOT: BR
2476  006720  000137  007160                  JMP     BTOV            ;ELSE GO TO BAD TAPE OVERFLOW
2477  006724  005237  000702          WRTY4:  INC     RTCNT           ;BUMP RETRY COUNTER
2478  006730  022737  000004  000702          CMP     #4,RTCNT        ;SEE IF DONE 4 RETRIES
2479  006736  001410                          BEQ     WRTY5           ;IF SO: BR
2480  006740  013704  000674                  MOV     UNP,R4
2481  006744  005264  001050                  INC     RTY1(R4)        ;BUMP RETRY COUNTER
2482  006750  005237  000732                  INC     ERTFL           ;SET ERASE FLAG
2483  006754  000137  006370                  JMP     WRTY0           ;DO NEXT RETRY
2484  006760  000137  007400          WRTY5:  JMP     BTUR            ;ELSE GO TO BAD TAPE UNRECOVERABLE
2485
2486                                  ;WRITE RETRY BACKSPACE-ERASE SUBROUTINE**********
2487
2488  006764  005037  000706          WRTSB:  CLR     SERFL           ;CLEAR FLAG
2489  006770  013737  000576  000666          MOV     TSTAL,STAL
2490  006776  004737  012304                  JSR     PC,STALL        ;DO TURN AROUND DELAY
2491  007002  012737  026320  000652          MOV     #MSG66,EMADDR   ;SET ERROR CODE
2492  007010  012777  177777  171500          MOV     #-1,@FC         ;SET TO BACKSPACE 1 RECORD
2493  007016  012777  033472  171470          MOV     #RDATA,@BA      ;SET BA
2494  007024  004737  012234                  JSR     PC,BKRT         ;GO BACKSPACE
2495  007030  005737  000706                  TST     SERFL           ;SEE IF ERROR
2496  007034  001406                          BEQ     WRTSB1          ;IF NOT: BR
2497  007036  012737  000002  000724  WRTSB0: MOV     #2,BTFLG        ;SET FLAG
2498  007044  022626                          CMP     (SP)+,(SP)+     ;RESET STACK
2499  007046  000137  004424                  JMP     REOT            ;GO REWIND AND REMOVE FROM TESTING
2500  007052  005737  000732          WRTSB1: TST     ERTFL           ;SEE IF SHOULD ERASE
2501  007056  001001                          BNE     WRTSB2          ;IF SO: BR
2502  007060  000207                          RTS     PC              ;RETURN
2503  007062  005037  000732          WRTSB2: CLR     ERTFL           ;CLEAR ERASE FLAG
2504  007066  005037  000700                  CLR     RPCNT           ;CLEAR REPEAT CNTR
2505  007072  005037  000706                  CLR     SERFL           ;CLEAR FLAG
2506  007076  012737  026333  000652          MOV     #MSG67,EMADDR   ;SET ERROR CODE
2507  007104  005077  171406                  CLR     @FC             ;CLEAR FRAME COUNT
2508  007110  012737  000024  000672          MOV     #24,MTC1        ;SET ERASE OP-CODE
2509  007116  012777  027464  171370          MOV     #WDATA,@BA      ;SET BA
2510  007124  012737  007136  000662          MOV     #WRTSB3,RTRN    ;SET RETURN ADDRESS
2511  007132  000137  021216                  JMP     TAPG            ;GO ERASE
2512  007136  012703  027464          WRTSB3: MOV     #WDATA,R3       ;SET EXPT BA
2513  007142  004737  017504                  JSR     PC,ER2          ;GO CHECK ERRORS
2514  007146  005737  000706                  TST     SERFL           ;SEE IF ERROR
2515  007152  001737                          BEQ     WRTSB1          ;IF NOT: BR
2516  007154  000137  007036                  JMP     WRTSB0
2517
```

```
                                        ;BAD TAPE OVERFLOW SUBROUTINE**********
007160  005037  000712          BTOV:   CLR     RTYFL           ;CLEAR RETRY FLAG
007164  012737  000001  000724          MOV     #1,BTFLG        ;SET BAD TAPE OVERFLOW FLAG
007172  005726                          TST     (SP)+           ;++G ADJUST STACK
007174  000137  004424                  JMP     REOT            ;GO REWIND AND REMOVE FROM TESTING
007200  013701  000730          BTOV0:  MOV     BTPT,R1         ;SET TABLE POINTER
007204  005721                          TST     (R1)+
007206  005000                          CLR     R0
007210  010003                  BTOV1:  MOV     R0,R3
007212  000241                          CLC
007214  006003                          ROR     R3              ;R3=R3/2 FOR CORRECT NUMBER
007216  104002                          OCTPP                   ;PRINT ENTRY NUMBER
007220  012704  025054                  MOV     #MSG13,R4
007224  105724                          TSTB    (R4)+           ;SKIP CR/LF
007226  104000                          TTOUTT                  ;PRINT BLOCK NUMBER TAG
007230  011103                          MOV     (R1),R3
007232  104002                          OCTPP                   ;PRINT BLOCK NUMBER
007234  012704  025062                  MOV     #MSG14,R4
007240  104000                          TTOUTT                  ;PRINT RECORD NUMBER TAG
007242  062701  000040                  ADD     #40,R1          ;SET POINTER OFFSET FOR RECOED NUMBER
007246  012103                          MOV     (R1)+,R3
007250  104002                          OCTPP                   ;PRINT RECORD NUMBER
007252  162701  000040                  SUB     #40,R1          ;RESET POINTER FOR BLOCK NUMBER
007256  005720                          TST     (R0)+
007260  020077  171444                  CMP     R0,@BTPT        ;SEE IF DONE
007264  001404                          BEQ     BTOV2           ;IF SO: BR
007266  012704  025405                  MOV     #MSG28,R4
007272  104000                          TTOUTT                  ;DO CR/LF
007274  000745                          BR      BTOV1           ;CONTINUE
007276  005737  000726  BTOV2:  TST     BTSTF           ;SEE IF STAT ONLY PRINT
007302  001007                          BNE     BTOVX           ;IF SO: BR
007304  012703  000041                  MOV     #41,R3          ;SET SIZE OF TABLE
007310  013704  000730                  MOV     BTPT,R4         ;SET POINTER
007314  005024          BTOV3:  CLR     (R4)+           ;CLEAR TABLE
007316  005303                          DEC     R3              ;SEE IF DONE
007320  001375                          BNE     BTOV3           ;IF NOT: BR
007322  000207          BTOVX:  RTS     PC              ;RETURN
```

```
2557
2558
2559
2560  007324  012704  025405          BTPRT:  MOV     #MSG28,R4
2561  007330  104000                          TTOUTT                  ;DO CR/LF
2562  007332  013704  000674                  MOV     UNP,R4
2563  007336  016437  001030  000730          MOV     BTADDR(R4),BTPT ;SET TABLE POINTER
2564  007344  017703  171360                  MOV     @BTPT,R3
2565  007350  000241                          CLC
2566  007352  006003                          ROR     R3              ;CORRECT NUMBER
2567  007354  104002                          OCTPP                   ;PRINT NUMBER OF BAD SPOTS
2568  007356  012704  027234                  MOV     #MSG112,R4
2569  007362  104000                          TTOUTT                  ;PRINT BAD TAPE TAG
2570  007364  005777  171340                  TST     @BTPT           ;SEE IF ANY BAD SPOTS
2571  007370  001001                          BNE     BTPRT1          ;IF SO: BR
2572  007372  000207                          RTS     PC              ;ELSE RETURN
2573  007374  000137  007200          BTPRT1: JMP     BTOVO           ;PRINT STATS
2574
2575
2576
2577  007400  004737  022736          BTUR:   JSR     PC,PAPRT        ;PRINT HEADER
2578  007404  012704  027047                  MOV     #MSG107,R4
2579  007410  104000                          TTOUTT                  ;PRINT UNRECOVERABLE BAD SPOT MSG
2580  007412  000207                          RTS     PC              ;RESUME TESTING
2581
```

;BAD TAPE STATISTIC PRINT********

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

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

007414  012737  000002  000562  RSEQ:   MOV     #2,RDCMD
007422  017704  171160                  MOV     @SWR,R4         ;READ SWITCHES
007426  042704  177763                  BIC     #177763,R4      ;MASK READ BITS
007432  005704                          TST     R4              ;SEE IF BOTH READS
007434  001004                          BNE     RSR             ;IF NOT: BR
007436  032777  000002  171142          BIT     #2,@SWR         ;SEE IF READ REVERSE FIRST
007444  001051                          BNE     RSFR            ;IF NOT: BR
007446  032777  000004  171132  RSR:    BIT     #4,@SWR         ;SEE IF SHOULD READ REVERSE
007454  001005                          BNE     RSF             ;IF NOT: BR
007456  012737  010000  000562          MOV     #10000,RDCMD    ;LOAD READ REVERSE COMMAND
007464  004737  007732                  JSR     PC,READ         ;GO READ REVERSE
007470  032777  000010  171110  RSF:    BIT     #10,@SWR        ;SEE IF SHOULD READ FORWARD
007476  001026                          BNE     RSEX            ;IF NOT: BR
007500  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF HAVE READ REVERSE
007506  001407                          BEQ     RSF0            ;IF NOT: BR
007510  013737  000576  000666          MOV     TSTAL,STAL
007516  004737  012304                  JSR     PC,STALL        ;DO READ STALL
007522  000137  007542                  JMP     RSF1
007526  032777  000001  171052  RSF0:   BIT     #1,@SWR         ;SEE IF WRITE
007534  001002                          BNE     RSF1            ;IF NOT: BR
007536  004737  012056                  JSR     PC,BKSP         ;GO BACKSPACE
007542  012737  000002  000562  RSF1:   MOV     #2,RDCMD        ;LOAD READ FORWARD COMMAND
007550  004737  007732                  JSR     PC,READ         ;GO READ
007554  005737  000660          RSEX:   TST     EOTREC          ;++G BRANCH IF NOT AT EOT
007560  100002                          BPL     1$              ;++G
007562  000137  004424                  JMP     REOT            ;++G ELSE GO REWIND
007566  000207                  1$:     RTS     PC              ;++G EXIT

007570  012737  010000  000562  RSFR:   MOV     #10000,RDCMD
007576  032777  000010  171002          BIT     #10,@SWR        ;SEE IF SHOULD READ FORWARD
007604  001013                          BNE     RSFR1           ;IF NOT: BR
007606  032777  000001  170772          BIT     #1,@SWR         ;SEE IF WRITE
007614  001002                          BNE     RSFR0           ;IF NOT: BR
007616  004737  012056                  JSR     PC,BKSP         ;GO BACKSPACE TO START
007622  012737  000002  000562  RSFR0:  MOV     #2,RDCMD        ;LOAD READ FORWARD COMMAND
007630  004737  007732                  JSR     PC,READ         ;GO READ FORWARD
007634  032777  000004  170744  RSFR1:  BIT     #4,@SWR         ;SEE IF SHOULD READ REVERSE
007642  001344                          BNE     RSEX            ;IF NOT: BR
007644  032737  010000  000562          BIT     #10000,RDCMD
007652  001005                          BNE     RSFR2           ;IF READ REVERSE: BR
```

SEQ 0059

```
2638  007654  013737  000576  000666          MOV     TSTAL,STAL      ;DO READ STALL
2639  007662  004737  012304                  JSR     PC,STALL
2640  007666  012737  010000  000562  RSFR2:  MOV     #10000,RDCMD    ;LOAD READ REVERSE
2641  007674  004737  007732                  JSR     PC,READ         ;GO READ REVERSE
2642  007700  005737  000660                  TST     EOTREC          ;SEE IF AT END OF TAPE
2643  007704  100011                          BPL     RSFRX           ;++G IF NOT: BR
2644  007706  163737  000554  000660          SUB     RCNT,EOTREC
2645  007714  005437  000660                  NEG     EOTREC          ;SET TO PROPER RECORD NUMBER
2646  007720  005237  000660                  INC     EOTREC
2647  007724  000137  004424                  JMP     REOT            ;ELSE GO TO REWIND
2648  007730  000207          RSFRX:  RTS     PC                      ;EXIT
2649
```

```
                                ;*********************************************************
                                ;READ ROUTINE:
                                ;
                                ;THIS ROUTINE PERFORMS THE READ OPERATION DETERMINED
                                ;BY THE READ SEQUENCE ROUTINE ONE RECORD AT A TIME.
                                ;AT THE END OF EACH READ OPERATION THE STATUS REGISTER
                                ;IS SCANNED FOR EITHER END OF TAPE OR BEGINNING OF TAPE.
                                ;IF EOT WAS REACHED, CONTROL WILL BE PASSED TO
                                ;THE EOT SUBROUTINE TO REWIND THE UNIT AND FLAG IT
                                ;UNAVAILABLE UNTIL ALL UNITS HAVE REACHED EOT.
                                ;IF BOT WAS REACHED AN ERROR IS PRINTED AND THE
                                ;PROGRAM WILL HALT.  TESTING MAY BE RESUMED BY PRESSING
                                ;THE CONTINUE SWITCH.
                                ;IF A TAPE MARK IS EXPECTED (TM=1) THEN THE
                                ;READ ROUTINE EXPECTS THE FIRST RECORD OF A
                                ;READ REVERSE TO BE A TM, AND THE LAST RECORD
                                ;OF A READ FORWARD TO BE A TM.  REMEMBER
                                ;THAT THE TM ADDS ONE (1) TO THE TOTAL NUMBER
                                ;OF RECORDS IN A BLOCK.
                                ;CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13) DETERMINE WHETHER
                                ;OR NOT TO CHECK FOR STATUS ERRORS (11) OR DATA ERRORS (13),
                                ;CONSOLE SWITCH FIVE (5) IS USED TO CAUSE A CONTINUOUS
                                ;READ AND SPACE (FORWARD OR REVERSE) OF THE CURRENT
                                ;RECORD ON TAPE (YOZZLE).
                                ;*********************************************************
007732  013700  000554       READ:  MOV     RCNT,R0          ;LOAD REC CNTR
007736  005737  000660              TST     EOTREC           ;SEE IF EOT
007742  100013                      BPL     RDA              ;IF NOT: BR
007744  032737  010000  000562      BIT     #10000,RDCMD     ;SEE IF READ FORWARD
007752  001407                      BEQ     RDA              ;IF SO: BR
007754  042737  100000  000660      BIC     #100000,EOTREC   ;CLEAR FLAG
007762  013703  000660              MOV     EOTREC,R3        ;GET MODIFIED RECORD COUNT
007766  160300                      SUB     R3,R0            ;SET RECORD AT
007770  005200                      INC     R0               ;SET TO PROPER NUMBER OF RECORDS
007772  012737  024767  000652  RDA: MOV     #MSG6,EMADDR     ;SET ERROR MSG ADDRESS
010000  005037  000676              CLR     TMFLG
010004  032737  010000  000562      BIT     #10000,RDCMD
010012  001406                      BEQ     RDO              ;IF READ FORWARD: BR
010014  005737  000564              TST     TMEX             ;SEE IF TM
010020  001403                      BEQ     RDO              ;IF NOT: BR
010022  005237  000676              INC     TMFLG            ;SET TM FLAG
010026  005200                      INC     R0
010030  013777  000556  170460  RDO: MOV     FMCNT,@FC        ;LOAD CHAR CNTR
010036  012777  033472  170450      MOV     #RDATA,@BA       ;LOAD DATA ADDR
010044  032737  010000  000562      BIT     #10000,RDCMD     ;SEE IF READ REVERSE
010052  001417                      BEQ     RD1A             ;IF NOT: BR
010054  013703  000556              MOV     FMCNT,R3
010060  005103                      COM     R3
010062  032737  000020  000552      BIT     #20,UDES         ;SEE IF CORE DUMP
010070  001402                      BEQ     RD1              ;IF NOT: BR
010072  000241                      CLC
010074  006003                      ROR     R3               ;R3 = FC/2
010076  060377  170412      RD1:    ADD     R3,@BA           ;SET REVERSE BUS ADDRESS
010102  012737  000076  000672      MOV     #76,MTC1         ;SET READ REVERSE
010110  000403                      BR      RD1B
```

```
2706  010112  012737  000070  000672  RD1A:   MOV     #70,MTC1        ;SET READ FORWARD
2707  010120  012737  010132  000662  RD1B:   MOV     #RD2,RTRN       ;SET INTERRUPT RETURN ADDRESS
2708  010126  000137  021216          RD1D:   JMP     TAPG            ;GO EXECUTE TAPE COMMAND
2709  010132  032737  010000  000562  RD2:    BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2710  010140  001024                          BNE     RD3             ;IF SO: BR
2711  010142  032777  000020  170352          BIT     #20,@DS
2712  010150  001404                          BEQ     RD2B
2713  010152  032777  000020  170342  RD2A:   BIT     #20,@DS         ;AWAIT SWDN
2714  010160  001374                          BNE     RD2A            ;AWAIT TUR
2715  010162  032777  002000  170332  RD2B:   BIT     #2000,@DS       ;SEE IF EOT
2716  010170  001410                          BEQ     RD3             ;IF NOT: BR
2717  010172  005737  000676                  TST     TMFLG           ;SEE IF TM
2718  010176  001005                          BNE     RD3             ;IF SO: BR
2719  010200  010037  000660                  MOV     R0,EOTREC
2720  010204  052737  100000  000660          BIS     #100000,EOTREC  ;SET EOT FLAG
2721  010212  032777  000002  170302  RD3:    BIT     #2,@DS          ;SEE IF AT LOAD POINT
2722  010220  001410                          BEQ     RD4             ;IF NOT: BR
2723  010222  004737  022736                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
2724  010226  012704  025166                  MOV     #MSG22,R4
2725  010232  104000                          TTOUTT                  ;PRINT BOT ERROR
2726  010234  104006                          STOPP
2727  010236  000137  003160                  JMP     STARTA          ;RESTART
2728  010242  032777  004000  170336  RD4:    BIT     #4000,@SWR      ;SEE IF SHOULD CHECK ERRORS
2729  010250  001121                          BNE     RD5             ;IF NOT: BR
2730  010252  005737  000676                  TST     TMFLG
2731  010256  001472                          BEQ     RD4B            ;IF NO TM EXPT: BR
2732  010260  032777  000004  170234          BIT     #4,@DS
2733  010266  001024                          BNE     RD4A            ;IF TM RECVD: BR
2734  010270  012737  033472  021132          MOV     #RDATA,CADER    ;SAVE EXPT BUS ADDRESS
2735  010276  012737  000002  021140          MOV     #2,DRVER        ;SET TM STATUS ERROR FLAG
2736  010304  004737  020234                  JSR     PC,ERPT         ;GO PRINT TM ERROR
2737  010310  013704  000674                  MOV     UNP,R4
2738  010314  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2739  010322  001403                          BEQ     1$              ;IF NOT: BR
2740  010324  005264  001150                  INC     RDERR1(R4)      ;BUMP READ REVERSE ERROR
2741  010330  000502                          BR      RD6
2742  010332  005264  001110          1$:     INC     RDER1(R4)       ;BUMP READ FORWARD ERROR
2743  010336  000477                          BR      RD6
2744  010340  012703  033472          RD4A:   MOV     #RDATA,R3
2745  010344  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2746  010352  001007                          BNE     RD4A0           ;IF SO: BR
2747  010354  032737  002000  000552          BIT     #2000,UDES      ;SEE IF IN PE
2748  010362  001025                          BNE     RD4A2           ;IF SO: BR
2749  010364  062703  000002                  ADD     #2,R3
2750  010370  000422                          BR      RD4A2
2751  010372  013704  000556          RD4A0:  MOV     FMCNT,R4
2752  010376  005104                          COM     R4
2753  010400  032737  000020  000552          BIT     #20,UDES        ;SEE IF CORE DUMP
2754  010406  001402                          BEQ     RD4A1           ;IF NOT: BR
2755  010410  000241                          CLC
2756  010412  006004                          ROR     R4              ;SET TO FC/2
2757  010414  060403          RD4A1:  ADD     R4,R3           ;SET EXPT BUS ADDRESS
2758  010416  042703  000001                  BIC     #1,R3           ;MAKE EXPT ADDRESS EVEN
2759  010422  032737  002000  000552          BIT     #2000,UDES      ;SEE IF IN PE
2760  010430  001002                          BNE     RD4A2           ;IF SO: BR
2761  010432  162703  000002                  SUB     #2,R3
```

```
2762  010436  004737  017504          RD4A2:  JSR     PC,ER2
2763  010442  000402                  BR      RD4C
2764  010444  004737  017406          RD4B:   JSR     PC,ERCHK        ;GO CHECK ERRORS
2765  010450  005737  000706          RD4C:   TST     SERFL
2766  010454  001417                  BEQ     RD5                     ;IF NO ERROR: BR
2767  010456  013704  000674          MOV     UNP,R4
2768  010462  032737  010000  000562  BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2769  010470  001003                  BNE     RD4D                    ;IF SO: BR
2770  010472  005264  001110          INC     RDER1(R4)               ;BUMP READ FORWARD ERROR
2771  010476  000402                  BR      RD4E
2772  010500  005264  001150          RD4D:   INC     RDERR1(R4)      ;BUMP READ REVERSE ERROR
2773  010504  004737  010706          RD4E:   JSR     PC,RDRTY                ;GO RETRY
2774  010510  005037  000712          CLR     RTYFL                   ;CLEAR RETRY FLAG
2775  010514  032777  020000  170064  RD5:    BIT     #20000,@SWR     ;SEE IF SHOULD DO DATA CHECK
2776  010522  001005                  BNE     RD6                     ;IF NOT; BR
2777  010524  005737  000676          TST     TMFLG
2778  010530  001002                  BNE     RD6
2779  010532  004737  015544          JSR     PC,DCHK                 ;GO CHECK DATA
2780  010536  005037  000706          RD6:    CLR     SERFL           ;CLEAR STATUS ERROR FLAG
2781  010542  004737  014352          JSR     PC,DS3                  ;CLEAR BUFFER
2782  010546  032777  000040  170032  BIT     #40,@SWR                ;SEE IF SHOULD YOZZLE
2783  010554  001402                  BEQ     RD7                     ;IF NOT: BR
2784  010556  004737  011274          JSR     PC,YOZ                  ;ELSE GO YOZZLE
2785  010562  013737  000572  000666  RD7:    MOV     RSTAL,STAL      ;SET DELAY
2786  010570  004737  012304          JSR     PC,STALL                ;STALL
2787  010574  032737  010000  000562  BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2788  010602  001403                  BEQ     RD7A                    ;IF NOT: BR
2789  010604  005037  000676          CLR     TMFLG                   ;CLEAR TAPE MARK FLAG
2790  010610  000405                  BR      RD10
2791  010612  005737  000660          RD7A:   TST     EOTREC          ;SEE IF EOT FOUND
2792  010616  100002                  BPL     RD10                    ;IF NOT: BR
2793  010620  012700  000001          MOV     #1,R0                   ;SET TO EOT
2794  010624  005300                  RD10:   DEC     R0
2795  010626  001402                  BEQ     RD11                    ;IF DONE ALL: BR
2796  010630  000137  010030          JMP     RD0
2797  010634  032737  010000  000562  RD11:   BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2798  010642  001016                  BNE     RDEX                    ;IF SO: BR
2799  010644  005737  000660          TST     EOTREC                  ;SEE IF FOUND EOT
2800  010650  100413                  BMI     RDEX                    ;IF SO: BR
2801  010652  005737  000564          TST     TMEX                    ;SEE IF TM EXPECTED
2802  010656  001410                  BEQ     RDEX                    ;IF NOT: BR
2803  010660  005737  000676          TST     TMFLG                   ;SEE IF TM FOUND
2804  010664  001005                  BNE     RDEX                    ;IF SO: BR
2805  010666  005237  000676          INC     TMFLG                   ;ELSE SET FLAG
2806  010672  005200                  INC     R0                      ;SET RECORD COUNT TO ONE
2807  010674  000137  010030          JMP     RD0                     ;GO READ TM
2808  010700  005037  000676          RDEX:   CLR     TMFLG
2809  010704  000207                  RDX:    RTS     PC              ;EXIT
```

```
2810                                    ;*********************************************************
2811                                    ;READ ERROR RETRY SUBROUTINE:
2812                                    ;
2813                                    ;THIS SUBROUTINE WILL RETRY ALL DATA RELATED
2814                                    ;READ ERRORS UP TO EIGHT (8) TIMES.  IF ALL
2815                                    ;FOUR RETRIES ARE BAD, IT IS CONSIDERED
2816                                    ;A HARD ERROR.  IF ANY ARE GOOD, IT IS A
2817                                    ;SOFT ERROR.  RETRIES MAY BE INHIBITED
2818                                    ;VIA SWITCH FOUR (SW4=0: INHIBIT RETRIES)
2819                                    ;*********************************************************
2820
2821  010706  032777  000020  167672  RDRTY:  BIT     #20,@SWR        ;SEE IF RETRY INHIBITED
2822  010714  001001                   BNE     RDRT0           ;IF NOT: BR
2823  010716  000207                   RTS     PC              ;ELSE RETURN
2824  010720  013703  000722          RDRT0:  MOV     ERSAV,R3
2825  010724  042703  102700                  BIC     #102700,R3      ;MARK NON-RECOVERABLE ERROR BITS
2826  010730  001410                   BEQ     RDRT1           ;IF NOT: BR
2827  010732  004737  022736                  JSR     PC,PAPRT        ;PRINT HEADER
2828  010736  012704  026612                  MOV     #MSG79,R4
2829  010742  104000                           TTOUTT                  ;PRINT NON-RECOVERABLE MESSAGE
2830  010744  004737  011260                  JSR     PC,NRTP         ;PRINT ER FOR NON-RETRYABLE ERROR
2831  010750  000207                   RDRT1A: RTS     PC              ;RETURN
2832  010752  032777  002000  167626  RDRT1:  BIT     #2000,@SWR      ;SEE IF PRINT INHIBITED
2833  010760  001003                   BNE     RDRT1B          ;IF SO: BR
2834  010762  012704  026265                  MOV     #MSG64,R4
2835  010766  104000                           TTOUTT                  ;PRINT ORIGINAL ERROR TAG
2836  010770  005037  000702          RDRT1B: CLR     RTCNT           ;CLEAR RETRY COUNTER
2837  010774  005037  000706          RDRTG:  CLR     SERFL           ;CLEAR STATUS ERROR FLAG
2838  011000  012737  000002  000712          MOV     #2,RTYFL        ;SET READ RETRY FLAG
2839  011006  004737  011274                  JSR     PC,YOZ          ;GO TO YOZZLE TO RETRY READ
2840  011012  005737  000706                  TST     SERFL           ;SEE IF RETRY ERROR
2841  011016  001031                   BNE     RDRT5           ;IF SO: BR
2842  011020  032777  002000  167560          BIT     #2000,@SWR
2843  011026  001011                   BNE     RDRT2
2844  011030  012704  026744                  MOV     #MSG105,R4
2845  011034  104000                           TTOUTT                  ;PRINT RECOVERED MESSAGE
2846  011036  012704  026307                  MOV     #MSG65,R4
2847  011042  104000                           TTOUTT                  ;PRINT RETRY TAG
2848  011044  013703  000702                  MOV     RTCNT,R3
2849  011050  104002                           OCTPP                   ;PRINT RETRY NUMBER
2850  011052  013704  000674          RDRT2:  MOV     UNP,R4
2851  011056  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2852  011064  001003                   BNE     RDRT3           ;IF SO: BR
2853  011066  005264  002670                  INC     RFSOFT(R4)      ;ELSO BUMP FORWARD SOFT ERROR COUNTER
2854  011072  000402                   BR      RDRT4
2855  011074  005264  002710          RDRT3:  INC     RRSOFT(R4)      ;BUMP ERRORS SOFT CNTR
2856  011100  000207                   RDRT4:  RTS     PC              ;RETURN
2857  011102  005037  000646          RDRT5:  CLR     TEMP3           ;++G CLEAR RECOVERABLE ERROR INDICATOR
2858  011106  013703  000722                  MOV     ERSAV,R3        ;GET ER
2859  011112  042703  102700                  BIC     #102700,R3      ;MASK RECOVERABLE BITS
2860  011116  001413                   BEQ     RDRT5A          ;IF RECOVERABLE: BR
2861  011120  004737  022736                  JSR     PC,PAPRT        ;PRINT HEADER
2862  011124  012704  026612                  MOV     #MSG79,R4
2863  011130  104000                           TTOUTT                  ;PRINT NON-RECOVERABLE MSG
2864  011132  004737  011260                  JSR     PC,NRTP         ;PRINT ER
2865  011136  012737  000001  000646          MOV     #1,TEMP3        ;SET FLAG
```

```
2866  011144  000404                        BR      RDRT5B
2867  011146  032777  002000  167432  RDRT5A: BIT     #2000,@SWR      ;SEE IF PRINT INHIBITED
2868  011154  001014                        BNE     RDRT6           ;IF SO: BR
2869  011156  012704  026307          RDRT5B: MOV     #MSG65,R4
2870  011162  104000                        TTOUTT                  ;PRINT RETRY TAG
2871  011164  013703  000702                MOV     RTCNT,R3
2872  011170  104002                        OCTPP                   ;PRINT RETRY NUMBER
2873  011172  005737  000646                TST     TEMP3           ;SEE IF DID NON-RECOVERABLE
2874  011176  001403                        BEQ     RDRT6           ;IF NOT: BR
2875  011200  005037  000646                CLR     TEMP3           ;CLEAR FLAG
2876  011204  000207                        RTS     PC              ;EXIT
2877  011206  005237  000702          RDRT6:  INC     RTCNT
2878  011212  023737  000702  000602        CMP     RTCNT,RETRY     ;SEE IF DONE 8 RETRIES
2879  011220  001265                        BNE     RDRTG           ;IF NOT: BR
2880  011222  012704  027277                MOV     #MSG115,R4
2881  011226  104000                        TTOUTT                  ;PRINT HARD ERROR MESSAGE
2882  011230  013704  000674                MOV     UNP,R4
2883  011234  032737  010000  000562        BIT     #10000,RDCMD    ;SEE IF READ REVERSE
2884  011242  001003                        BNE     RDRT7           ;IF SO: BR
2885  011244  005264  002730                INC     RFHARD(R4)      ;BUMP FORWARD HARD ERROR CNTR
2886  011250  000402                        BR      RDRTX
2887  011252  005264  002750          RDRT7:  INC     RRHARD(R4)      ;BUMP REVERSE HARD ERROR CNTR
2888  011256  000207                  RDRTX:  RTS     PC              ;RETURN
2889
2890  011260  013703  000722          NRTP:   MOV     ERSAV,R3        ;GET ER REGISTER
2891  011264  104002                        OCTPP                   ;PRINT ER
2892  011266  004737  021156                JSR     PC,FRPRT        ;PRINT F OR R
2893  011272  000207                        RTS     PC              ;RETURN
```

```
                                        ;*************************************************************
                                        ;YOZZLE SUBROUTINE:
                                        ;
                                        ;THIS SUBROUTINE, ENTERED VIA SWITCH FIVE (5), IS USED TO PERFORM
                                        ;A CONTINUOUS READ AND SPACE OVER OF THE CURRENT RECORD ON TAPE.
                                        ;FULL STATUS AND DATA CHECKING MAY BE PERFORMED
                                        ;OR NOT VIA CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13).
                                        ;A SOFTWARE DELAY IS PERFORMED BETWEEN EACH READ
                                        ;AND SPACE OPERATION AND MAY BE VARIED BY TYPING
                                        ;CNTRL C ON THE TTY AND ENTERING A VALUE IN RESPONSE
                                        ;TO THE PRINTED REQUEST.
                                        ;*************************************************************
011274  012777  000001  167306  YOZ:    MOV     #1,@TKS         ;SET TTY ENABLE
011302  013737  000600  000666          MOV     YSTAL,STAL
011310  004737  012304                  JSR     PC,STALL        ;DO YOZZLE STALL
011314  012777  177777  167174  YOZ0:   MOV     #-1,@FC         ;SET TO 1 RECORD SPACING
011322  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
011330  001404                          BEQ     YOZA            ;IF NOT: BR
011332  112737  000030  000672          MOVB    #30,MTC1        ;SET TO SPACE FORWARD
011340  000403                          BR      YOZB
011342  112737  000032  000672  YOZA:   MOVB    #32,MTC1        ;SET TO SPACE REVERSE
011350  012737  011370  000662  YOZB:   MOV     #YOZC,RTRN      ;SET RETURN ADDRESS
011356  012737  177775  000666          MOV     #177775,STAL    ;SET TIME MULTIPLIER
011364  000137  021216                  JMP     TAPG            ;GO YOZZLE
011370  005737  000676          YOZC:   TST     TMFLG           ;SEE IF TM
011374  001404                          BEQ     1$              ;IF NOT: BR
011376  012737  040000  000666          MOV     #40000,STAL     ;SET TM STALL
011404  000403                          BR      2$
011406  013737  000600  000666  1$:     MOV     YSTAL,STAL
011414  004737  012304          2$:     JSR     PC,STALL        ;DO YOZZLE STALL
011420  012777  033472  167066          MOV     #RDATA,@BA      ;SET BUS ADDRESS
011426  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
011434  001417                          BEQ     YOZC1           ;IF NOT: BR
011436  013703  000556                  MOV     FMCNT,R3
011442  005103                          COM     R3
011444  032737  000020  000552          BIT     #20,UDES        ;SEE IF CORE DUMP
011452  001402                          BEQ     YOZC0           ;IF NOT: BR
011454  000241                          CLC
011456  006003                          ROR     R3              ;R3 = FC/2
011460  060377  167030          YOZC0:  ADD     R3,@BA          ;SET REVERSE BUS ADDRESS
011464  012737  000076  000672          MOV     #76,MTC1        ;SET READ REVERSE
011472  000403                          BR      YOZC2
011474  012737  000070  000672  YOZC1:  MOV     #70,MTC1        ;SET READ FORWARD
011502  013777  000556  167006  YOZC2:  MOV     FMCNT,@FC       ;SET CHARACTER COUNT
011510  012737  011522  000662          MOV     #YOZD,RTRN      ;SET RETURN ADDRESS
011516  000137  021216                  JMP     TAPG            ;GO READ
011522  032777  004000  167056  YOZD:   BIT     #4000,@SWR      ;SEE IF SHOULD CHECK ERRORS
011530  001051                          BNE     YOZE            ;IF NOT: BR
011532  005737  000676                  TST     TMFLG           ;SEE IF TAPE MARK TIME
011536  001444                          BEQ     YOZD1           ;IF NOT: BR
011540  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
011546  001426                          BEQ     YOZD0           ;IF NOT: BR
011550  012703  033472                  MOV     #RDATA,R3
011554  013704  000556                  MOV     FMCNT,R4
011560  005104                          COM     R4
```

```
2950  011562  032737  000020  000552          BIT     #20,UDES        ;SEE IF CORE DUMP
2951  011570  001402                          BEQ     YOZD4           ;IF NOT: BR
2952  011572  000241                          CLC
2953  011574  006004                          ROR     R4              ;SET TO FC/2
2954  011576  060403                  YOZD4:  ADD     R4,R3           ;SET EXPT BUS ADDRESS
2955  011600  042703  000001                BIC     #1,R3           ;MAKE EXPT ADDRESS EVEN
2956  011604  032737  002000  000552        BIT     #2000,UDES      ;SEE IF PE
2957  011612  001001                          BNE     YOZD2           ;IF SO: BR
2958  011614  005743                          TST     -(R3)           ;SET EXPT BA
2959  011616  004737  017504                YOZD2:  JSR     PC,ER2          ;GO CHECK ERRORS
2960  011622  000430                          BR      YOZF
2961  011624  012703  033472                YOZD0:  MOV     #RDATA,R3
2962  011630  032737  002000  000552        BIT     #2000,UDES      ;SEE IF PE
2963  011636  001001                          BNE     YOZD3           ;IF SO: BR
2964  011640  005723                          TST     (R3)+           ;SET EXPT BA
2965  011642  004737  017504                YOZD3:  JSR     PC,ER2          ;GO CHECK ERRORS
2966  011646  000416                          BR      YOZF
2967  011650  004737  017406                YOZD1:  JSR     PC,ERCHK        ;ELSE GO CHECK ERRORS
2968  011654  032777  020000  166724        YOZE:   BIT     #20000,@SWR     ;SEE IF SHOULD CHECK DATA
2969  011662  001010                          BNE     YOZF            ;IF NOT: BR
2970  011664  005737  000676                  TST     TMFLG           ;SEE IF TAPE MARK
2971  011670  001005                          BNE     YOZF            ;IF SO: BR
2972  011672  005737  000712                  TST     RTYFL           ;SEE IF RETRY
2973  011676  001004                          BNE     YOZF0           ;IF SO: BR
2974  011700  004737  015544                JSR     PC,DCHK         ;ELSE GO CHECK DATA
2975  011704  004737  014352                YOZF:   JSR     PC,DS3          ;GO CLEAR DATA AREA
2976  011710  105777  166674                YOZF0:  TSTB    @TKS            ;SEE IF HAVE NEW STALL VALUE
2977  011714  100032                          BPL     YOZG            ;IF NOT: BR
2978  011716  122777  000203  166666        CMPB    #203,@TKB       ;SEE IF CONT C
2979  011724  001026                          BNE     YOZG            ;IF NOT: BR
2980  011726  012704  025767                MOV     #MSG44,R4
2981  011732  104000                          TTOUTT                  ;PRINT YSTALL REQUEST
2982  011734  013703  000600                MOV     YSTAL,R3
2983  011740  104002                          OCTPP                   ;PRINT PRESENT STALL
2984  011742  010037  000646                MOV     R0,TEMP3        ;SAVE R0(REC CNT)
2985  011746  012705  000600                MOV     #YSTAL,R5       ;SET ADDRESS OF YSTL
2986  011752  012701  000006                MOV     #6,R1           ;SET NUMBER OF CHAR TO INPUT
2987  011756  012702  177777                MOV     #-1,R2          ;SET MAXIMUM LIMIT
2988  011762  012703  002000                MOV     #2000,R3        ;SET MINIMUM LIMIT
2989  011766  004737  023354                JSR     PC,TTR          ;GO GET VALUE
2990  011772  013700  000646                MOV     TEMP3,R0        ;RESTORE R0(REC CNTR)
2991  011776  000137  011274                JMP     YOZ             ;RESTART YOZZLE
2992  012002  122777  000207  166602  YOZG:  CMPB    #207,@TKB       ;CHECK FOR CNTL G
2993  012010  001010                          BNE     YOZI
2994  012012  022737  000176  000606        CMP     #SWREG,SWR      ;IS SWREG SELECTED
2995  012020  001004                          BNE     YOZI
2996  012022  005077  166564                CLR     @TKB            ;CLEAR CNTL G OUT OF BUFFER
2997  012026  004737  024452                JSR     PC,CNTG         ;GO CHANGE SWREG
2998  012032  032777  000040  166546  YOZI:  BIT     #40,@SWR        ;SEE IF SHOULD CONTINUE YOZZLE
2999  012040  001402                          BEQ     YOZH            ;IF NOT: BR
3000  012042  000137  011314                JMP     YOZO
3001  012046  012777  000100  166534  YOZH:  MOV     #100,@TKS       ;SET TTY INTERRUPT ENABLE
3002  012054  000207                          RTS     PC              ;EXIT
3003
```

```
                                               ;*****************************************************
                                               ;BACKSPACE SUBROUTINE:
                                               ;
                                               ;THIS SUBROUTINE IS USED TO PERFORM THE
                                               ;BACKSPACE OPERATION REQUIRED BY THE READ
                                               ;ROUTINE FOR READ FORWARD AFTER WRITING.
                                               ;IF A TAPE MARK IS EXPECTED (TM=1) THEN THE SPACE
                                               ;ROUTINE ASSUMES THAT THE TM WILL BE FIRST WHEN
                                               ;BACKSPACING.  THEREFORE TWO OPERATIONS ARE REQUIRED
                                               ;TO SPACE OVER A BLOCK.  FIRST SPACE OVER THE TM, THEN
                                               ;SPACE OVER THE DATA RECORDS.
                                               ;A CHECK FOR RECORD COUNT ZERO IS MADE AT THE
                                               ;END OF THE SPACE OPERATION TO ASSURE THAT PROPER
                                               ;TAPE POSITIONING WAS DONE.
                                               ;*****************************************************

012056  013737  000576  000666  BKSP:   MOV     TSTAL,STAL
012064  004737  012304                  JSR     PC,STALL                ;DO TURN AROUND STALL
012070  012737  025017  000652          MOV     #MSG10,EMADDR
012076  012777  033472  166410          MOV     #RDATA,@BA
012104  005737  000564                  TST     TMEX                    ;SEE IF TM
012110  001440                          BEQ     B0                      ;IF NOT: BR
012112  012777  177777  166376          MOV     #-1,@FC
012120  012737  000032  000672          MOV     #32,MTC1
012126  012737  012140  000662          MOV     #BKTM,RTRN
012134  000137  021216                  JMP     TAPG                    ;SPACE TO TM
012140  032777  010000  166440  BKTM:   BIT     #10000,@SWR             ;SEE IF SHOULD CHECK ERROR
012146  001021                          BNE     B0                      ;IF NOT: BR
012150  012737  026201  000652          MOV     #MSG55,EMADDR
012156  032777  000004  166336          BIT     #4,@DS                  ;SEE IF TM
012164  001006                          BNE     BKTM0                   ;IF SO: BR
012166  012737  033472  021132          MOV     #RDATA,CADER
012174  004737  020234                  JSR     PC,ERPT                 ;PRINT ERROR
012200  000404                          BR      B0
012202  012703  033472          BKTM0:  MOV     #RDATA,R3
012206  004737  017504                  JSR     PC,ER2
012212  013700  000554          B0:     MOV     RCNT,R0
012216  005100                          COM     R0                      ;BUILD SPACE AMOUNT
012220  005200                          INC     R0
012222  012737  025017  000652          MOV     #MSG10,EMADDR           ;SET ERROR MESG ADDRESS
012230  010077  166262                  MOV     R0,@FC
012234  012737  000032  000672  BKRT:   MOV     #32,MTC1                ;SET SPACE REVERSE
012242  012737  012260  000662          MOV     #B1,RTRN                ;SET RETURN ADDRESS
012250  010037  000666                  MOV     R0,STAL                 ;SET INTERRUPT TIME MULTIPLIER
012254  000137  021216                  JMP     TAPG                    ;GO DO SPACE
012260  012703  033472          B1:     MOV     #RDATA,R3
012264  004737  017504                  JSR     PC,ER2
012270  013737  000576  000666  B2:     MOV     TSTAL,STAL              ;DO STALL
012276  004737  012304                  JSR     PC,STALL                ;STALL
012302  000207                          RTS     PC                      ;EXIT
```

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

012304  005337  000666          STALL:  DEC     STAL
012310  001375                          BNE     STALL                   ;DELAY
012312  000207                          RTS     PC                      ;EXIT
```

```
3078
3079                                    ;*********************************************************
3080                                    ;RANDOM CHARACTER COUNT GENERATOR:
3081                                    ;
3082                                    ;THIS ROUTINE ENTERED VIA CONSOLE SWITCH
3083                                    ;SEVEN (7) IS USED TO GENERATE A RANDOM
3084                                    ;CHARACTER COUNT FOR EACH DATA BLOCK.
3085                                    ;ALL RECORDS WITHIN A GIVEN BLOCK WILL BE
3086                                    ;THE SAME, BUT EACH BLOCK WILL VARY.
3087                                    ;THE LIMITS ARE TWENTY (20) TO FOUR THOUSAND
3088                                    ;(4000) OCTAL CHARACTERS PER RECORD.
3089                                    ;*********************************************************
3090
3091  012314  012701  177760       CCNTR:  MOV     #-20,R1         ;SET HIGH LIMIT
3092  012320  012702  174000               MOV     #-4000,R2       ;SET LOW LIMIT
3093  012324  004737  023322               JSR     PC,RANG         ;GO GENERATE NUMBER
3094  012330  042737  000001  000626       BIC     #1,RANSAV
3095  012336  013737  000626  000556       MOV     RANSAV,FMCNT    ;SET CHAR COUNT
3096  012344  012737  177777  014406       MOV     #-1,PATS        ;PRESET DATA PATTERN
3097  012352  000207                       RTS     PC              ;EXIT
3098
3099                                    ;*********************************************************
3100                                    ;RANDOM RECORD COUNT GENERATOR:
3101                                    ;
3102                                    ;THIS ROUTINE ENTERED VIA CONSOLE SWITCH SIX (6)
3103                                    ;IS USED TO GENERATE A RANDOM NUMBER OF RECORDS
3104                                    ;FOR EACH BLOCK OF DATA.
3105                                    ;THE LIMITS ARE ONE (1) TO FIVE HUNDRED (500) OCTAL
3106                                    ;RECORDS PER BLOCK.
3107                                    ;*********************************************************
3108
3109  012354  012702  000001       RCNTR:  MOV     #1,R2           ;SET LOW LIMIT
3110  012360  012701  000500               MOV     #500,R1         ;SET HIGH LIMIT
3111  012364  004737  023322               JSR     PC,RANG         ;GO GENERATE NUMBER
3112  012370  013737  000626  000554       MOV     RANSAV,RCNT     ;SET RECORD COUNT
3113  012376  000207                       RTS     PC              ;EXIT
3114
3115
```

```
3116
3117                    ;*********************************************************
3118                    ;TEST CONDITION ENTRY ROUTINE:
3119                    ;
3120                    ;THIS ROUTINE IS USED TO ALLOW THE OPERATOR
3121                    ;TO ENTER, AT THE TTY, THE NECESSARY PARAMETERS
3122                    ;TO RUN THE PROGRAM AS HE WISHES.  THE
3123                    ;ROUTINE IS ONLY ENTERED UPON INITIAL STARTING
3124                    ;FROM LOCATION 200(8).
3125                    ;THE MAIN PURPOSE OF THIS ROUTINE IS TO ESTABLISH
3126                    ;A TABLE OF DEVICES TO BE TESTED.  THIS TABLE
3127                    ;CONSISTS OF AN ENTRY FOR EACH OF ONE (1) TO
3128                    ;EIGHT (8) DEVICES.  EACH ENTRY CONTAINS THE
3129                    ;SLAVE NUMBER, DENSITY, PARITY, AND
3130                    ;FORMAT.  THE INFORMATION IS ENTERED
3131                    ;IN RESPONSE TO PRINTED REQUESTS AT THE TTY.
3132                    ;SLAVES MAY BE ENTERED IN ANY ORDER.  EACH
3133                    ;PARAMETER IS CHECKED FOR LEGALITY BEFORE BEING
3134                    ;SET INTO THE TABLE.
3135                    ;THE DRIVE NUMBER REQUEST WILL ALSO CHECK THE MASSBUS
3136                    ;FOR THE PRESENCE OF THE REQUESTED DRIVE.  IF IT IS NOT FOUND,
3137                    ;A NON-EXIST DRIVE MESSAGE WILL BE PRINTED AND ANOTHER DRIVE
3138                    ;REQUEST MADE.  WHEN THE DRIVE IS FOUND, THE RESPONSE IS STORED
3139                    ;AND CONTROL PASSED TO THE SLAVE SELECT ROUTINE.
3140                    ;THE SLAVE SELECT ROUTINE ALSO CHECKS FOR THE PRESENCE OF THE
3141                    ;SLAVE.  IF IT IS NOT PRESENT, A MESSAGE IS PRINTED AND ANOTHER
3142                    ;REQUEST IS ISSUED.  WHEN THE SELECTED SLAVE IS FOUND TO BE
3143                    ;PRESENT, A MESSAGE IS PRINTED IF IT IS A 7 CHANNEL DRIVE
3144                    ;TO ASSIST IN SELECTING DENSITY, PARITY, AND FORMAT.
3145                    ;UPON COMPLETION OF THE DEVICE TABLE, REQUESTS
3146                    ;ARE PRINTED FOR ENTRY OF THE NUMBER OF CHARACTERS
3147                    ;PER RECORD AND THE NUMBER OF RECORDS PER BLOCK.  THE
3148                    ;NEXT REQUEST IS FOR A PATTERN NUMBER TO BE USED
3149                    ;FOR WRITING AND CHECKING OF READ DATA.
3150                    ;FOLLOWING THE PATTERN REQUEST IS THE TAPE MARK OPTION.
3151                    ;RESPONDING TO THE REQUEST (TM=) WITH A ONE (1)
3152                    ;WILL CAUSE THE PROGRAM TO WRITE A TM AT THE
3153                    ;END OF EACH DATA BLOCK AND TO EXPECT THE
3154                    ;TM TO BE DETECTED IN EITHER READ FORWARD AND REVERSE
3155                    ;OR DURING SPACE OPERATION.  A RESPONSE OF ZERO (TM=0)
3156                    ;DISALLOWS WRITTING OF THE TM AND CAUSES THE READ
3157                    ;AND SPACE ROUTINES TO EXPECT NO TM TO BE PRESENT.
3158                    ;THE LAST REQUESTS ARE FOR ENTRY OF THE DESIRED
3159                    ;WRITE, READ, AND TURN AROUND STALLS.
3160                    ;*********************************************************
3161
3162  012400  005737  000634   TINP:   TST     TINF            ;SEE IF SHOULD INPUT FROM TTY
3163  012404  001001           BNE     TINPA           ;IF SO: BR
3164  012406  000207           RTS     PC              ;EXIT
3165  012410  005037  000674   TINPA:  CLR     UNP             ;CLEAR TABLE POINTER
3166  012414  005037  005156           CLR     REOTC           ;CLEAR EOT UNIT COUNTER
3167  012420  012700  000010           MOV     #10,R0          ;SET SIZE OF TABLE
3168  012424  012701  000746           MOV     #UN1,R1         ;SET START OF TABLE
3169  012430  005021           TINPB:  CLR     (R1)+           ;CLEAR TABLE
3170  012432  005300           DEC     R0              ;SEE IF DONE
3171  012434  001375           BNE     TINPB           ;IF NOT: BR
```

```
012436 012704 025461          MOV     #MSG31,R4
012442 005737 000734          TST     ASEQF           ;SEE IF AUTO SEQ
012446 001402                 BEQ     TINPB1          ;IF NOT: BR
012450 012704 025407          MOV     #MSG30,R4       ;SET AUTO SEQ HDR
012454 104000       TINPB1:   TTOUTT                  ;PRINT PROGRAM NAME
012456 005737 014150          TST     SCVFL           ;SEE IF SHORT CONVERSATION
012462 001067                 BNE     TINPC           ;IF SO: BR
012464 012704 026471          MOV     #MSG74,R4
012470 104000                 TTOUTT                  ;REQUEST STARTING REGISTER ADDRESS
012472 013703 000544          MOV     REGS,R3
012476 104002                 OCTPP                   ;PRINT CURRENT REG START
012500 012705 000544          MOV     #REGS,R5        ;SAVE ADDRESS LOCATION
012504 012701 000006          MOV     #6,R1           ;SET SIZE OF ENTRY
012510 012702 176400          MOV     #176400,R2      ;SET UPPER LIMIT
012514 012703 172300          MOV     #172300,R3      ;SET LOWER LIMIT
012520 004737 023354          JSR     PC,TTR          ;GO GET RESPONSE
012524 012704 026514          MOV     #MSG75,R4
012530 104000                 TTOUTT                  ;GO REQUEST VECTOR ADDRESS
012532 013703 000546          MOV     VECT,R3
012536 104002                 OCTPP                   ;PRINT CURRENT VECTOR
012540 012705 000546          MOV     #VECT,R5        ;SET SAVE LOCATION
012544 012701 000003          MOV     #3,R1           ;SET SIZE OF ENTRY
012550 012702 000224          MOV     #224,R2         ;SET UPPER LIMIT
012554 012703 000150          MOV     #150,R3         ;SET LOWER LIMIT
012560 004737 023354          JSR     PC,TTR          ;GO GET RESPONSE
012564 013700 000546          MOV     VECT,R0         ;GET VECTOR ADDRESS
012570 012720 021714          MOV     #MTINT,(R0)+    ;LOAD VECTOR WITH HANDLER ADDRESS
012574 012710 000340          MOV     #340,(R0)       ;LOAD PRIORITY LEVEL
012600 013700 000544          MOV     REGS,R0         ;GET STARTING REGISTER ADDRESS
012604 012701 000016          MOV     #16.,R1         ;SET NUMBER OF REGISTERS
012610 012702 000510          MOV     #C1,R2          ;GET FIRST ADDRESS LOCATION
012614 010022       TINPB0:   MOV     R0,(R2)+        ;BUILD TABLE OF ADDRESSES
012616 062700 000002          ADD     #2,R0           ;BUMP ADDRESS
012622 005301                 DEC     R1              ;SEE IF DONE
012624 001373                 BNE     TINPB0          ;IF NOT: BR
012626 005737 000734          TST     ASEQF           ;SEE IF AUTO SEQ
012632 001403                 BEQ     TINPC           ;IF NOT: BR
012634 005726                 TST     (SP)+           ;RESET STACK POINTER
012636 000137 021732          JMP     ASEQ            ;GO TO AUTO SEQUENCE
012642 012777 000040 165650 TINPC:  MOV     #40,@CS   ;INITIALIZE
012650 012704 026135          MOV     #MSG52,R4
012654 104000                 TTOUTT                  ;REQUEST DRIVE NUMBER
012656 012705 000550          MOV     #DVN,R5         ;GET ADDRESS
012662 012701 000001          MOV     #1,R1           ;SET SIZE OF RESPONSE
012666 012702 000007          MOV     #7,R2           ;SET UPPER LIMIT
012672 012703 000000          MOV     #0,R3           ;SET LOWER LIMIT
012676 004737 023354          JSR     PC,TTR          ;GO GET DRIVE NUMBER
012702 013777 000550 165610   MOV     DVN,@CS
012710 005777 165574          TST     @C1             ;ACCESS DRIVE
012714 032777 010000 165576   BIT     #10000,@CS      ;SEE IF NED
012722 001411                 BEQ     TINPQ           ;IF NOT: BR
012724 012704 026426          MOV     #MSG71,R4
012730 104000                 TTOUTT                  ;PRINT NED
012732 013704 000510          MOV     C1,R4
012736 005204                 INC     R4
012740 152714 000100          BISB    #100,(R4)       ;CLEAR TRE
```

```
3228  012744  000736                      BR      TINPC           ;++G RETRY DVN
3229  012746  012704  025523      TINP0:   MOV     #MSG32,R4
3230  012752  104000                      TTOUTT                  ;PRINT UNIT NUMBER REQUEST
3231  012754  005037  000644              CLR     TEMP2           ;CLEAR BUFFER
3232  012760  012705  000644              MOV     #TEMP2,R5       ;SET UNIT DESCRIPTION BUFFER ADDRESS
3233  012764  012701  000001              MOV     #1,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3234  012770  012702  000007              MOV     #7,R2           ;SET MAXIMUM LIMIT
3235  012774  012703  000000              MOV     #0,R3           ;SET MINIMUM LIMIT
3236  013000  004737  023354              JSR     PC,TTR          ;GO GET UNIT NUMBER
3237  013004  005737  000642              TST     TEMP1           ;SEE IF HAVE NEW PARAMETER
3238  013010  001013                      BNE     TINP0B          ;IF SO: BR
3239  013012  005737  000674              TST     UNP             ;SEE IF FIRST ENTRY
3240  013016  001001                      BNE     TINP0A          ;IF NOT: BR
3241  013020  000752                      BR      TINP0           ;++G ELSE RETRY
3242  013022  013700  000674      TINP0A:  MOV     UNP,R0
3243  013026  012760  177777  000746      MOV     #-1,UN1(R0)     ;SET END UNIT TABLE
3244  013034  000137  013424              JMP     TINP2C          ;GO GET RECORD COUNT
3245  013040  013700  000674      TINP0B:  MOV     UNP,R0
3246  013044  042760  000007  000746      BIC     #7,UN1(R0)      ;CLEAR UNIT NUMBER
3247  013052  004737  014164              JSR     PC,TPOS1        ;GO LOAD UNIT NUMBER TO PROPER POSITION
3248  013056  012777  000040  165434      MOV     '40,@CS
3249  013064  013777  000550  165426      MOV     @N,@CS
3250  013072  016077  000746  165442      MOV     UN1(R0),@C2     ;LOAD UNIT NUMBER
3251  013100  032777  002000  165430  TINP0C: BIT     #2000,@DT       ;SEE  IF SLAVE PRESENT
3252  013106  001005                      BNE     TINP0D          ;IF SO: BR
3253  013110  012704  026214              MOV     #MSG57,R4
3254  013114  104000                      TTOUTT                  ;PRINT NON-EXIST SLAVE
3255  013116  000137  012746              JMP     TINP0           ;REDO
3256  013122  022777  142011  165406  TINP0D: CMP     #142011,@DT     ;++G SEE IF 9TRK TM02,TU16/TE16
3257  013130  001406                      BEQ     TINP0E          ;IF SO: BR
3258  013132  012704  026110              MOV     #MSG50,R4       ;ILLEGAL DRIVE TYPE
3259  013136  104000                      TTOUTT                  ;GO PRINT
3260  013140  017703  165372              MOV     @DT,R3
3261  013144  104002                      OCTPP                   ;PRINT DRIVE TYPE REGISTER
3262  013146  012704  025011      TINP0E:  MOV     #MSG9,R4
3263  013152  104000                      TTOUTT                  ;PRINT SERIAL NUMBER TAG
3264  013154  017703  165360              MOV     @SN,R3
3265  013160  004737  024300              JSR     PC,$NPT         ;PRINT SERIAL NUMBER
3266  013164  012704  025544      TINP1:   MOV     #MSG33,R4
3267  013170  104000                      TTOUTT                  ;PRINT DENSITY REQUEST
3268  013172  005037  000644              CLR     TEMP2           ;CLEAR BUFFER
3269  013176  012701  000001              MOV     #1,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3270  013202  012702  000007              MOV     #7,R2           ;SET MAXIMUM LIMIT
3271  013206  012703  000000              MOV     #0,R3           ;SET MINIMUM LIMIT
3272  013212  004737  023354              JSR     PC,TTR          ;GO GET DENSITY
3273  013216  005737  000642              TST     TEMP1           ;SEE IF HAVE NEW PARAMETER
3274  013222  001407                      BEQ     TINP2           ;IF NOT: BR
3275  013224  042737  003400  000552      BIC     #3400,UDES      ;ELSE CLEAR OLD PARAMETER
3276  013232  012703  000010              MOV     #10,R3          ;SET POSITION FACTOR
3277  013236  004737  014152              JSR     PC,TPOS         ;GO LOAD DENSITY INTO PROPER POSITION
3278  013242  012704  025560      TINP2:   MOV     #MSG34,R4
3279  013246  104000                      TTOUTT                  ;PRINT PARITY REQUEST
3280  013250  005037  000644              CLR     TEMP2           ;CLR BUFFER
3281  013254  012701  000001              MOV     #1,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3282  013260  012702  000001              MOV     #1,R2           ;SET MAXIMUM LIMIT
3283  013264  012703  000000              MOV     #0,R3           ;SET MINIMUM LIMIT
```

```
3284  013270  004737  023354                JSR     PC,TTR          ;GO INPUT PARITY
3285  013274  005737  000642                TST     TEMP1           ;SEE IF HAVE NEW PARAMETER
3286  013300  001407                        BEQ     TINP2A          ;IF NOT: BR
3287  013302  042737  000010  000552        BIC     #10,UDES        ;ELSE CLEAR OLD PARAMETER
3288  013310  012703  000003                MOV     #3,R3           ;SET POSITION FACTOR
3289  013314  004737  014152                JSR     PC,TPOS         ;GO LOAD PARITY TO PROPER POSITION
3290  013320  012704  026157        TINP2A: MOV     #MSG53,R4
3291  013324  104000                        TTOUTT                  ;REQUEST FORMAT
3292  013326  005037  000644                CLR     TEMP2
3293  013332  012701  000002                MOV     #2,R1
3294  013336  012702  000016                MOV     #16,R2
3295  013342  012703  000014                MOV     #14,R3
3296  013346  004737  023354                JSR     PC,TTR          ;GO GET FORMAT
3297  013352  005737  000642                TST     TEMP1           ;SEE IF NEW PARAMETER
3298  013356  001407                        BEQ     TINP2B          ;IF NOT: BR
3299  013360  042737  000170  000552        BIC     #170,UDES
3300  013366  012703  000004                MOV     #4,R3
3301  013372  004737  014152                JSR     PC,TPOS
3302  013376  005237  005156        TINP2B: INC     REOTC           ;BUMP EOT UNIT COUNTER
3303  013402  022737  000016  000674        CMP     #16,UNP         ;SEE IF DONE UNITS
3304  013410  001405                        BEQ     TINP2C          ;IF SO: BR
3305  013412  062737  000002  000674        ADD     #2,UNP          ;POINT TO NEXT UNIT
3306  013420  000137  012746                JMP     TINPO           ;ELSE LOOK FOR NEXT UNIT
3307  013424  005037  000674        TINP2C: CLR     UNP             ;CLEAR UNIT POINTER
3308  013430  013700  005156                MOV     REOTC,R0
3309  013434  000337  005156                SWAB    REOTC
3310  013440  110037  005156                MOVB    R0,REOTC        ;SET UNIT EOT COUNTER
3311  013444  012704  025573        TINP3:  MOV     #MSG35,R4
3312  013450  104000                        TTOUTT                  ;PRINT RECORD COUNT REQUEST
3313  013452  013703  000554                MOV     RCNT,R3
3314  013456  104002                        OCTPP                   ;PRINT RECORD COUNT
3315  013460  012705  000554                MOV     #RCNT,R5        ;SET RECORD COUNT ADDRESS
3316  013464  012701  000006                MOV     #6,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3317  013470  012702  177777                MOV     #-1,R2          ;SET MAXIMUM LIMIT
3318  013474  012703  000001                MOV     #1,R3           ;SET MINIMUM LIMIT
3319  013500  004737  023354                JSR     PC,TTR          ;GO GET RECORD COUNT
3320  013504  013737  000554  000630        MOV     RCNT,RCSAV      ;SAVE RECORD COUNT
3321  013512  012704  025614                MOV     #MSG36,R4
3322  013516  104000                        TTOUTT                  ;PRINT CHARACTER COUNT REQUEST
3323  013520  005437  000556                NEG     FMCNT
3324  013524  013703  000556                MOV     FMCNT,R3
3325  013530  104002                        OCTPP                   ;PRINT CHAR COUNT
3326  013532  012705  000556                MOV     #FMCNT,R5       ;SET CHARACTER COUNT ADDRESS
3327  013536  012701  000006                MOV     #6,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3328  013542  012702  004000                MOV     #4000,R2        ;SET MAXIMUM LIMIT
3329  013546  012703  000004                MOV     #4,R3           ;SET MINIMUM LIMIT
3330  013552  004737  023354                JSR     PC,TTR          ;GO GET CHARACTER COUNT
3331  013556  005437  000556                NEG     FMCNT           ;SET TO TWO'S COMPLIMENT
3332  013562  013737  000556  000632        MOV     FMCNT,FCSAV     ;SAVE FRAME COUNT
3333  013570  012704  025640                MOV     #MSG37,R4       ;PRINT PATTERN NUMBER REQUEST
3334  013574  104000                        TTOUTT
3335  013576  013703  000560                MOV     PATRN,R3
3336  013602  104002                        OCTPP                   ;PRINT PATTERN
3337  013604  005037  014554                CLR     DOFL            ;CLEAR EXTERNAL DATA FLAG
3338  013610  012705  000560                MOV     #PATRN,R5       ;SET PATTERN NUMBER ADDRESS
3339  013614  012701  000002                MOV     #2,R1           ;SET NUMBER OF CHARACTERS TO INPUT
```

```
3340  013620  012702  000015          MOV     #15,R2          ;SET MAXIMUM LIMIT
3341  013624  012703  000000          MOV     #0,R3           ;SET MINIMUM LIMIT
3342  013630  004737  023354          JSR     PC,TTR          ;GO GET PATTERN NUMBER
3343  013634  012704  026354          MOV     #MSG69,R4
3344  013640  104000                  TTOUTT                  ;REQUEST TM
3345  013642  013703  000564          MOV     TMEX,R3
3346  013646  104002                  OCTPP                   ;PRINT CURRENT TM FLAG SETTING
3347  013650  012705  000564          MOV     #TMEX,R5        ;GET TM FLAG ADDRESS
3348  013654  012701  000001          MOV     #1,R1           ;SET SIZE OF RESPONSE
3349  013660  012702  000001          MOV     #1,R2           ;SET UPPER LIMIT
3350  013664  012703  000000          MOV     #0,R3           ;SET LOWER LIMIT
3351  013670  004737  023354          JSR     PC,TTR          ;TM 1=YES
3352  013674  012704  025141          MOV     #MSG21,R4
3353  013700  104000                  TTOUTT                  ;REQUEST INTERCHANGE READ
3354  013702  013703  000566          MOV     INTRF,R3
3355  013706  104002                  OCTPP                   ;PRINT CURRENT SETTING
3356  013710  012705  000566          MOV     #INTRF,R5       ;GET FLAG ADDRESS
3357  013714  012701  000001          MOV     #1,R1           ;SET SIZE OF RESPONSE
3358  013720  012702  000001          MOV     #1,R2           ;SET UPPER LIMIT
3359  013724  012703  000000          MOV     #0,R3           ;SET LOWER LIMIT
3360  013730  004737  023354          JSR     PC,TTR          ;GO GET RESPONSE
3361  013734  012704  025663          MOV     #MSG38,R4
3362  013740  104000                  TTOUTT                  ;REQUEST SINGLE PASS
3363  013742  013703  000570          MOV     SPFLG,R3
3364  013746  104002                  OCTPP                   ;PRINT CURRENT SETTING
3365  013750  012705  000570          MOV     #SPFLG,R5       ;SET ADDRESS OF FLAG
3366  013754  012701  000001          MOV     #1,R1           ;SET SIZE OF RESPONSE
3367  013760  012702  000001          MOV     #1,R2           ;SET UPPER LIMIT
3368  013764  012703  000000          MOV     #0,R3           ;SET LOWER LIMIT
3369  013770  004737  023354          JSR     PC,TTR          ;GO GET RESPONSE
3370  013774  005737  014150  TINP4:   TST     SCVFL           ;SEE IF SHORT CONVERSATION
3371  014000  001060                  BNE     TINPX           ;IF SO: BR
3372  014002  012704  025703          MOV     #MSG40,R4
3373  014006  104000                  TTOUTT                  ;PRINT READ STALL REQUEST
3374  014010  013703  000572          MOV     RSTAL,R3
3375  014014  104002                  OCTPP                   ;PRINT READ STALL
3376  014016  012705  000572          MOV     #RSTAL,R5       ;SET READ STALL ADDRESS
3377  014022  012701  000006          MOV     #6,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3378  014026  012702  177777          MOV     #-1,R2          ;SET MAXIMUM LIMIT
3379  014032  012703  000001          MOV     #1,R3           ;SET MINIMUM LIMIT
3380  014036  004737  023354          JSR     PC,TTR          ;GO GET READ STALL
3381  014042  012704  025731          MOV     #MSG41,R4
3382  014046  104000                  TTOUTT                  ;PRINT WRITE STALL REQUEST
3383  014050  013703  000574          MOV     WSTAL,R3
3384  014054  104002                  OCTPP                   ;PRINT READ STALL
3385  014056  012705  000574          MOV     #WSTAL,R5       ;SET WRITE STALL ADDRESS
3386  014062  012701  000006          MOV     #6,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3387  014066  012702  177777          MOV     #-1,R2          ;SET MAXIMUM LIMIT
3388  014072  012703  000001          MOV     #1,R3           ;SET MINIMUM LIMIT
3389  014076  004737  023354          JSR     PC,TTR          ;GO GET WRITE STALL
3390  014102  012704  025743          MOV     #MSG42,R4
3391  014106  104000                  TTOUTT                  ;PRINT TURN AROUND STALL REQUEST
3392  014110  013703  000576          MOV     TSTAL,R3
3393  014114  104002                  OCTPP                   ;PRINT TA STALL
3394  014116  012705  000576          MOV     #TSTAL,R5       ;SET TURN AROUND STALL ADDRESS
3395  014122  012701  000006          MOV     #6,R1           ;SET NUMBER OF CHARACTERS TO INPUT
```

```
3396  014126  012702  177777                MOV     #-1,R2          ;SET MAXIMUM LIMIT
3397  014132  012703  000001                MOV     #1,R3           ;SET MINIMUM LIMIT
3398  014136  004737  023354                JSR     PC,TTR          ;GO GET TURN AROUND STALL
3399  014142  005037  014150        TINPX:  CLR     SCVFL           ;CLEAR SHORT CONVERSATION FLAG
3400  014146  000207                        RTS     PC              ;EXIT
3401  014150  000000                SCVFL:  0                       ;SHORT CONVERSATION FLAG
3402
3403                                        ;UNIT DESCRIPTION POSITIONING SUBROUTINE**********
3404
3405  014152  000241                TPOS:   CLC
3406  014154  006137  000644                ROL     TEMP2           ;POSITION CHARACTER
3407  014160  005303                        DEC     R3              ;SEE IF DONE
3408  014162  001373                        BNE     TPOS            ;IF NOT: BR
3409  014164  013700  000674        TPOS1:  MOV     UNP,R0          ;LOAD UNIT POINTER
3410  014170  053760  000644  000746        BIS     TEMP2,UN1(R0)   ;LOAD CHARACTER INTO UN1(R0)
3411  014176  000207                        RTS     PC              ;EXIT
3412
```

```
                                ;*********************************************************
                                ;DATA SETUP ROUTINE:
                                ;
                                ;THIS ROUTINE IS USED TO GENERATE INTO THE ENTIRE
                                ;WRITE BUFFER (4000 OCTAL CHARACTERS) THE DATA PATTERN
                                ;SELECTED BY THE OPERATOR.  THERE ARE 15 (8) FIXED
                                ;DATA PATTERNS AVAILABLE AND ONE SELECTION (DATA PATTERN 0)
                                ;WHICH WILL READ ANY PATTERN PRESENTED AT THE
                                ;HIGH SPEED PAPER TAPE READER.  THIS TAPE MUST BE PREPARED
                                ;BY USING THE PROGRAM CALLED DTC. (MAINDEC-11-DZTUF-A-D)
                                ;RANDOM DATA MAY ALSO BE USED VIA CONSOLE
                                ;SWITCH EIGHT (8).
                                ;THIS ROUTINE IS ALSO USED TO CLEAR OUT THE
                                ;READ BUFFER (4000 OCTAL CHARACTERS) BEFORE EACH
                                ;RECORD IS READ.
                                ;*********************************************************
014200  005737  015150        DSUP:   TST     RDFL            ;SEE IF DID RANDOM DATA
014204  001044                BNE     DS1             ;++G F SO BRANCH
014206  005737  000734        DS0:    TST     ASEQF           ;SEE IF AUTO SEQ
014212  001406                BEQ     DS0C            ;IF NOT: BR
014214  005737  000560                TST     PATRN           ;SEE IF AUTO RANDOM
014220  100003                BPL     DS0C            ;IF NOT: BR
014222  004737  015102                JSR     PC,DATR         ;ELSE GO GENERATE RANDOM DATA
014226  000207                RTS     PC              ;RETURN
014230  023737  000560  014406  DS0C:   CMP     PATRN,PATS      ;SEE IF NEW PATTERN
014236  001014                BNE     DS0A            ;IF SO: BR
014240  013703  000552                MOV     UDES,R3         ;GET UNIT DESCRIPTION
014244  042703  177767                BIC     #177767,R3      ;MASK EVEN PARITY
014250  023703  014410                CMP     PARS,R3         ;SEE IF SAME AS LAST TIME
014254  001404                BEQ     DS0B            ;IF SO: BR
014256  010337  014410                MOV     R3,PARS         ;SAVE PARITY
014262  004737  015152                JSR     PC,CRCLRC       ;GO GENERATE EXPT CRC/LRC
014266  000207                DS0B:   RTS     PC
014270  012703  027464        DS0A:   MOV     #WDATA,R3       ;R3 = ADDRS OF WRITE BUFFER
014274  013701  000560                MOV     PATRN,R1        ;R1 = PATTERN SELECTOR
014300  010137  014406                MOV     R1,PATS
014304  062701  000001                ADD     #1,R1           ;BUMP POINTER
014310  006301                ASL     R1              ;++G MAKE PATTERN SELECTOR EVEN
014312  004771  002770                JSR     PC,@DATBL(R1)   ;GO GENERATE PATTERN
014316  032777  010000  164212  DS1:    BIT     #10000,@DT      ;SEE IF 7 CH
014324  001410                BEQ     DS2A            ;IF NOT: BR
014326  012702  002002                MOV     #2002,R2        ;SET BUFFER SIZE
014332  012701  027464                MOV     #WDATA,R1       ;SET START OF BUFFER
014336  042721  140300        DS2:    BIC     #140300,(R1)+   ;MASK FOR 7 CH
014342  005302                DEC     R2              ;SEE IF DONE
014344  001374                BNE     DS2             ;IF NOT: BR
014346  004737  015152        DS2A:   JSR     PC,CRCLRC       ;GO GENERATE EXPT CRC/LRC
014352  012702  002000        DS3:    MOV     #2000,R2        ;R2=BUFFER SIZE
014356  012701  033472                MOV     #RDATA,R1       ;R1=READ DATA START
014362  005021                DS4:    CLR     (R1)+           ;CLEAR BUFFER
014364  005302                DEC     R2              ;SEE IF DONE ALL
014366  001375                BNE     DS4             ;IF NOT: BR
014370  013737  000552  014410        MOV     UDES,PARS       ;GET UNIT DESCRIPTION
014376  042737  177767  014410        BIC     #177767,PARS    ;MASK PARITY
```

```
3469  014404  000207                          RTS     PC              ;EXIT
3470  014406  177777                  PATS:   -1                      ;PATTERN NUMBER SAVE
3471  014410  000000                  PARS:   0
3472
3473                                          ;EXTERNAL DATA INPUT FROM H/S READER (256 CHARACTER MAXIMUM)
3474
3475  014412  005737  014554          DATO:   TST     DOFL            ;++G BRANCH IF EXTERNAL INPUT
3476  014416  001401                          BEQ     1$              ;++G
3477  014420  000207                          RTS     PC              ;++G RETURN
3478  014422  012737  000001  014554  1$:     MOV     #1,DOFL         ;SET EXTERNAL FLAG
3479  014430  005077  164166                  CLR     @PRB            ;CLEAR READER BUFFER
3480  014434  005077  164160                  CLR     @PRS            ;CLEAR READER STATUS
3481  014440  005037  000642                  CLR     TEMP1           ;CLEAR FOR USE AS CHARACTER FLAG
3482  014444  052777  000001  164146  DATOA:  BIS     #1,@PRS         ;START READER
3483  014452  105777  164142          DATOB:  TSTB    @PRS            ;++G SEE IF DONE
3484  014456  100375                          BPL     DATOB           ;++B
3485  014460  005001                          CLR     R1              ;CLEAR SAVE LOCATION
3486  014462  117701  164134                  MOVB    @PRB,R1         ;SAVE CHARACTER
3487  014466  005737  000642                  TST     TEMP1           ;SEE IF HAVE FOUND START CHARACTER
3488  014472  001011                          BNE     DATOC           ;IF SO : BR
3489  014474  105701                          TSTB    R1              ;SEE IF CHARACTER IS 0
3490  014476  001762                          BEQ     DATOA           ;IF SO : BR
3491  014500  012737  000001  000642          MOV     #1,TEMP1        ;ELSE SET CHARACTER FOUND FLAG
3492  014506  010137  000644                  MOV     R1,TEMP2        ;SAVE DATA SIZE
3493  014512  010102                          MOV     R1,R2           ;SAVE DATA SIZE
3494  014514  000753                          BR      DATOA           ;++G GO GET FIRST DATA CHAR
3495  014516  110123                  DATOC:  MOVB    R1,(R3)+        ;LOAD BUFFER
3496  014520  005302                          DEC     R2              ;SEE IF READ ALL
3497  014522  001350                          BNE     DATOA           ;IF NOT : BR
3498  014524  012701  027464          DATOD:  MOV     #WDATA,R1       ;R1 = START OF WRITE BUFFER
3499  014530  013702  000644                  MOV     TEMP2,R2        ;R2 = SIZE OF DATA FIELD
3500  014534  112123                  DATOE:  MOVB    (R1)+,(R3)+     ;REPEAT LOAD OF DATA FIELD
3501  014536  022703  033472                  CMP     #RDATA,R3       ;SEE IF DONE
3502  014542  003001                          BGT     DATOF           ;IF NOT: BR
3503  014544  000207                          RTS     PC              ;++G RETURN
3504  014546  005302                  DATOF:  DEC     R2              ;SEE IF AT END OF DATA FIELD
3505  014550  001371                          BNE     DATOE           ;IF NOT : BR
3506  014552  000764                          BR      DATOD           ;++G ELSE RESTART FILL
3507  014554  000000                  DOFL:   0                       ;EXTERNAL DATA FLAG=1 IF ALREADY DONE
3508
```

```
                                        ;ALL ONES**********
                                        
014556  012701  177777          DAT1:   MOV     #-1,R1          ;R1=DATA
014562  012702  002002          DAT1A:  MOV     #2002,R2        ;R2=WORD COUNT +2
014566  010123                  DAT1B:  MOV     R1,(R3)+        ;LOAD BUFFER
014570  005302                          DEC     R2              ;SEE IF DONE
014572  001375                          BNE     DAT1B           ;IF NOT: BR
014574  000207                          RTS     PC              ;++G RETURN
                                        
                                        ;ALL ZEROS**********
                                        
014576  005001                  DAT2:   CLR     R1              ;R1=DATA
014600  000770                          BR      DAT1A           ;++G LOAD BUFFER
                                        
                                        ;WALKING ONE**********
                                        
014602  012701  000001          DAT3:   MOV     #1,R1           ;R1=DATA
014606  000241                          CLC
014610  012702  004004          DAT3A:  MOV     #4004,R2        ;R2=CHARACTER COUNT+4
014614  110123                  DAT3B:  MOVB    R1,(R3)+        ;LOAD BUFFER
014616  106101                          ROLB    R1              ;SET NEXT CHARACTER
014620  005302                          DEC     R2              ;SEE IF DONE
014622  001374                          BNE     DAT3B           ;IF NOT: BR
014624  000207                          RTS     PC              ;++G RETURN
                                        
                                        ;WALKING ZERO**********
                                        
014626  012701  000376          DAT4:   MOV     #376,R1         ;R1=START OF DATA
014632  000261                          SEC
014634  000765                          BR      DAT3A           ;++G LOAD BUFFER
                                        
                                        ;ALTERNATING ONE/ZERO**********
                                        
014636  012701  052525          DAT5:   MOV     #52525,R1       ;R1=DATA
014642  000747                          BR      DAT1A           ;++G LOAD BUFFER
                                        
                                        ;ALTERNATING ZERO/ONE**********
                                        
014644  012701  125252          DAT6:   MOV     #125252,R1      ;R1=DATA
014650  000744                          BR      DAT1A           ;++G LOAD BUFFER
                                        
                                        ;ONE/ZERO IN ALTERNATING WORDS**********
                                        
014652  012701  125252          DAT7:   MOV     #125252,R1      ;SET WORD 1
014656  012702  052525                  MOV     #52525,R2       ;SET WORD 2
014662  012704  001002                  MOV     #1002,R4        ;SET NUMBER OF ENTRIES
014666  010123                  DAT7A:  MOV     R1,(R3)+        ;LOAD WORD 1
014670  010223                          MOV     R2,(R3)+        ;LOAD WORD 2
014672  005304                          DEC     R4              ;SEE IF DONE
014674  001374                          BNE     DAT7A           ;IF NOT: BR
014676  000207                          RTS     PC              ;++G RETURN
```

```
3562
3563                                    ;WALKING ONE/ALL ONE IN ALTERNATING CHARS****
3564  014700  012702  002002    DAT10:  MOV     #2002,R2        ;SET BUFFER SIZE
3565  014704  012701  000001            MOV     #1,R1           ;SET WALK BASE
3566  014710  000241                    CLC
3567  014712  012713  177400    DAT10A: MOV     #177400,(R3)    ;LOAD ALL ONE BYTE
3568  014716  050123                    BIS     R1,(R3)+        ;LOAD WALK BYTE
3569  014720  106101                    ROLB    R1              ;WALK ONE
3570  014722  005302                    DEC     R2
3571  014724  001372                    BNE     DAT10A          ;DO FULL BUFFER
3572  014726  000207                    RTS     PC              ;++G RETURN
3573
3574                                    ;ALL BITS 0-377*********
3575
3576  014730  005001    DAT11:  CLR     R1              ;R1=STARTING DATA
3577  014732  012702  004004            MOV     #4004,R2        ;R2=CHARACTER COUNT+4
3578  014736  110123    DAT11A: MOVB    R1,(R3)+        ;LOAD BUFFER
3579  014740  105201                    INCB    R1              ;BUMP DATA
3580  014742  005302                    DEC     R2              ;SEE IF DONE
3581  014744  001374                    BNE     DAT11A          ;IF NOT: BR
3582  014746  000207                    RTS     PC              ;++G RETURN
3583
3584                                    ;ALL BITS 377-0**********
3585
3586  014750  012701  000377    DAT12:  MOV     #377,R1         ;R1=STARTING DATA
3587  014754  012702  004004            MOV     #4004,R2        ;R2=CHARACTER COUNT+4
3588  014760  110123    DAT12A: MOVB    R1,(R3)+        ;LOAD BUFFER
3589  014762  105301                    DECB    R1              ;BUMP DATA
3590  014764  005302                    DEC     R2              ;SEE IF DONE
3591  014766  001374                    BNE     DAT12A          ;IF NOT: BR
3592  014770  000207                    RTS     PC              ;++G RETURN
3593
3594                                    ;ALTERNATING CHARACTERS 0 AND 377**********
3595
3596  014772  012701  000377    DAT13:  MOV     #377,R1         ;R1 = DATA
3597  014776  000137  014562            JMP     DAT1A           ;LOAD BUFFER
3598
3599                                    ;WALKING ZERO/ALL ZERO IN ALTERNATING CHARS******
3600
3601  015002  012702  002002    DAT14:  MOV     #2002,R2        ;SET BUFFER SIZE
3602  015006  012701  000376            MOV     #376,R1         ;SET WALK BASE
3603  015012  000261                    SEC
3604  015014  010113    DAT14A: MOV     R1,(R3)         ;LOAD WALK BYTE
3605  015016  042723  177400            BIC     #177400,(R3)+   ;CLEAR HIGH BYTE
3606  015022  106101                    ROLB    R1              ;WALK ZERO BIT
3607  015024  005302                    DEC     R2
3608  015026  001372                    BNE     DAT14A          ;FILL BUFFER
3609  015030  000207                    RTS     PC              ;++G RETURN
3610
```

```
                                        ;AUTO SEQUENCE PATTERN*********
                                
015032  012702  000200          DAT15:  MOV     #200,R2         ;SET NUMBER OF ENTRIES
015036  012701  015062          DAT15A: MOV     #APATS,R1       ;SET START OF PATTERN
015042  012704  000010                  MOV     #10,R4          ;SET SIZE OF PATTERN
015046  012123                  DAT15B: MOV     (R1)+,(R3)+     ;FILL BUFFER
015050  005304                          DEC     R4              ;SEE IF DONE PATTERN
015052  001375                          BNE     DAT15B          ;IF NOT: BR
015054  005302                          DEC     R2              ;SEE IF DONE BUFER
015056  001367                          BNE     DAT15A          ;IF NOT: BR
015060  000207                          RTS     PC              ;++G RETURN
015062  000000                  APATS:  0
015064  177400                          177400
015066  000377                          377
015070  000000                          0
015072  177777                          -1
015074  000377                          377
015076  177400                          177400
015100  177777                          -1
                                
                                        ;RANDOM DATA GENERATOR SUBROUTINE**********
                                
015102  013704  000556          DATR:   MOV     FMCNT,R4        ;SET NUMBER OF FRAMES
015106  012703  027464                  MOV     #WDATA,R3       ;SET ADDRESS OF START OF BUFFER
015112  012701  177777                  MOV     #-1,R1          ;SET HIGH LIMIT
015116  005002                          CLR     R2              ;SET LOW LIMIT
015120  004737  023322          DATR0:  JSR     PC,RANG         ;GO GENERATE NUMBER
015124  013723  000626                  MOV     RANSAV,(R3)+    ;LOAD BUFFER
015130  005204                          INC     R4              ;SEE IF DONE WHOLE BUFFER
015132  001372                          BNE     DATR0           ;IF NOT: BR
015134  004737  014316                  JSR     PC,DS1          ;GO CHECK FOR 7 CH
015140  012737  000001  015150          MOV     #1,RDFL         ;SET RANDOM DATA FLAG
015146  000207                          RTS     PC              ;EXIT
015150  000000                  RDFL:   0                       ;RANDOM DATA SELECT FLAG
```

```
                                ;*************************************************************
                                ;CRC/LRC CHARACTER BUILD;
                                ;
                                ;THIS ROUTINE WILL CONSTRUCT AND SAVE THE EXPECTED
                                ;CRC AND LRC CHARACTERS ACCORDING TO DATA AND
                                ;RECORD SIZE IF OPERATING IN NRZ MODE
                                ;*************************************************************
015152  013700  000556      CRCLRC: MOV     FMCNT,R0        ;SET RECORD SIZE
015156  005400                      NEG     R0
015160  012701  027464              MOV     #WDATA,R1       ;SET START OF BUFFER
015164  005037  015534              CLR     XORS
015170  111104              CL0:    MOVB    (R1),R4         ;GET CHARACTER
015172  004737  015362              JSR     PC,CLP          ;GO GET PARITY OF CHARACTER
015176  004737  015510              JSR     PC,XOR          ;XOR CHARACTER
015202  000241                      CLC
015204  006004                      ROR     R4              ;ROTATE 1 RIGHT
015206  103014                      BCC     CL2             ;IF NO CARRY: BR
015210  052704  000400              BIS     #400,R4         ;SET BIT NINE
015214  000241                      CLC
015216  010405              CL1:    MOV     R4,R5           ;SAVE CHARACTER
015220  042705  177703              BIC     #177703,R5
015224  005105                      COM     R5
015226  042705  177703              BIC     #177703,R5
015232  042704  000074              BIC     #74,R4
015236  050504                      BIS     R5,R4           ;COMPLIMENT BITS 2,3,4,5
015240  010437  015534      CL2:    MOV     R4,XORS
015244  005300                      DEC     R0
015246  001401                      BEQ     CLLAST          ;IF LAST CHARACTER: BR
015250  000747                      BR      CL0             ;++G GET NEXT
015252  013704  015534      CLLAST: MOV     XORS,R4
015256  005137  015534              COM     XORS
015262  042737  177050  015534      BIC     #177050,XORS
015270  042704  177727              BIC     #177727,R4      ;COMPLIMENT ALL BUT BITS 3&5
015274  050437  015534              BIS     R4,XORS
015300  013737  015534  015536      MOV     XORS,EXCRC      ;SAVE EXPECTED CRC
015306  013700  000556              MOV     FMCNT,R0
015312  005400                      NEG     R0
015314  012701  027464              MOV     #WDATA,R1       ;DO EXPT LRC
015320  005037  015534              CLR     XORS
015324  111104              CL3:    MOVB    (R1),R4
015326  004737  015362              JSR     PC,CLP          ;GET PARITY
015332  004737  015510              JSR     PC,XOR          ;XOR CHARACTER
015336  005300                      DEC     R0
015340  001371                      BNE     CL3             ;DO ALL FOR LRC
015342  013704  015536              MOV     EXCRC,R4
015346  004737  015510              JSR     PC,XOR          ;XOR CRC TO DATA
015352  013737  015534  015540      MOV     XORS,EXLRC      ;SAVE EXPT LRC
015360  000207                      RTS     PC              ;RETURN
015362  005704              CLP:    TST     R4              ;SEE IF 0 CHAR
015364  001010                      BNE     CLPE            ;IF NOT: BR
015366  032737  000010  000552      BIT     #10,UDES        ;SEE IF EVEN PARITY
015374  001404                      BEQ     CLPE            ;IF NOT: BR
015376  012704  000420              MOV     #420,R4         ;SET 0 CHAR EVEN PARITY
015402  005201                      INC     R1              ;BUMP POINTER
```

```
3701  015404  000207                    RTS     PC              ;RETURN
3702  015406  005037  015542    CLPE:   CLR     PARCNT          ;CLEAR BIT COUNTER
3703  015412  012703  000010            MOV     #10,R3          ;SET NUMBER OF BITS
3704  015416  032704  000001    CLP0:   BIT     #1,R4           ;SEE IF ONE BIT
3705  015422  001402                    BEQ     CLP1            ;IF NOT: BR
3706  015424  005237  015542            INC     PARCNT          ;BUMP COUNTER
3707  015430  000241            CLP1:   CLC
3708  015432  006004                    ROR     R4              ;ROTATE TO NEXT BIT
3709  015434  005303                    DEC     R3
3710  015436  001367                    BNE     CLP0            ;CONTINUE FOR ALL BITS
3711  015440  112104                    MOVB    (R1)+,R4
3712  015442  042704  177400            BIC     #177400,R4
3713  015446  032737  000001  015542    BIT     #1,PARCNT       ;SEE IF ODD NUMBER OF ONE BITS
3714  015454  001005                    BNE     CLP2            ;IF SO: BR
3715  015456  032737  000010  000552    BIT     #10,UDES        ;SEE IF SHOULD BE EVEN PARITY
3716  015464  001406                    BEQ     CLP3            ;IF NOT: BR
3717  015466  000207                    RTS     PC              ;ELSE EXIT
3718  015470  032737  000010  000552    BIT     #10,UDES        ;SEE IF SHOULD BE ODD PARITY
3719  015476  001001                    BNE     CLP3            ;IF NOT: BR
3720  015500  000207                    RTS     PC              ;ELSE EXIT
3721  015502  052704  000400    CLP3:   BIS     #400,R4         ;SET PARITY BIT
3722  015506  000207                    RTS     PC
3723  015510  010446            XOR:    MOV     R4,-(SP)
3724  015512  043716  015534            BIC     XORS,(SP)
3725  015516  040437  015534            BIC     R4,XORS         ;XOR SUBROUTINE: R4 WITH XORS
3726  015522  052637  015534            BIS     (SP)+,XORS
3727  015526  013704  015534            MOV     XORS,R4
3728  015532  000207                    RTS     PC
3729
3730  015534  000000    XORS:   0               ;XOR SAVE
3731  015536  000000    EXCRC:  0               ;EXPECTED CRC
3732  015540  000000    EXLRC:  0               ;EXPECTED LRC
3733  015542  000000    PARCNT: 0               ;PARITY COUNTER
3734
```

```
                                          ;**********************************************************
                                          ;DATA CHECK SUBROUTINE:
                                          ;
                                          ;THIS SUBROUTINE IS USED TO COMPARE EACH CHARACTER
                                          ;OF DATA READ FROM TAPE WITH THE EXPECTED CHARACTER.
                                          ;ANY ERROR DETECTED WILL CAUSE CONTROL TO BE
                                          ;PASSED TO AN ERROR PRINT SUBROUTINE AND A
                                          ;SUBROUTINE TO ACCUMULATE THE NUMBER OF BITS
                                          ;DROPPED AND PICKED UP FROM EACH CHARACTER.
                                          ;THE NUMBER OF READ ERRORS IS ALSO ACCUMULATED.
                                          ;DATA CHECKING MAY BE TERMINATED BY USE OF
                                          ;CONSOLE SWITCH THIRTEEN (13).
                                          ;**********************************************************

015544  005037  000656          DCHK:   CLR     BBC             ;CLEAR BAD RECORD CNTR
015550  005037  000704                  CLR     DERFL           ;CLEAR DATA ERROR FLAG
015554  013705  000556                  MOV     FMCNT,R5        ;LOAD CHAR COUNT
015560  032737  000020  000552          BIT     #20,UDES        ;SEE IF CORE DUMP
015566  001402                          BEQ     DCHKO           ;IF NOT: BR
015570  000261                          SEC
015572  006005                          ROR     R5              ;R5 = FC/2
015574  012701  027464          DCHKO:  MOV     #WDATA,R1       ;SET WRITE DATA ADDR
015600  012702  033472                  MOV     #RDATA,R2       ;SET READ DATA ADDR
015604  032737  000010  000552          BIT     #10,UDES        ;SEE IF EVEN PARITY
015612  001430                          BEQ     DFOCO           ;IF NOT: BR
015614  032737  000020  000552          BIT     #20,UDES        ;SEE IF CORE DUMP PARITY
015622  001024                          BNE     DFOCO           ;IF SO: BR
015624  032737  002000  000552          BIT     #2000,UDES      ;SEE IF PE MODE
015632  001020                          BNE     DFOCO           ;IF SO: BR
015634  105711                  DFOF:   TSTB    (R1)            ;SEE IF 0 CHAR
015636  001404                          BEQ     DFOD            ;IF SO: BR
015640  005201                          INC     R1              ;BUMP POINTER
015642  005205                  DFOE:   INC     R5              ;SEE IF DONE
015644  001373                          BNE     DFOF            ;IF NOT: BR
015646  000406                          BR      DFOC            ;ELSE CONTINUE
015650  112721  000020          DFOD:   MOVB    #20,(R1)+       ;SET 20 IN PLACE OF 0
015654  012737  177777  014406          MOV     #-1,PATS        ;SET PATTERN GENERATE FLAG
015662  000767                          BR      DFOE
015664  013705  000556          DFOC:   MOV     FMCNT,R5        ;RESET CHAR CNT
015670  012701  027464                  MOV     #WDATA,R1       ;RESET DATA ADDRESS
015674  032737  010000  000562  DFOCO:  BIT     #10000,RDCMD    ;SEE IF READ REVERSE
015702  001462                          BEQ     DFO             ;IF NOT: BR
015704  013704  000556          DFOB:   MOV     FMCNT,R4        ;GET FRAME COUNT
015710  005404                          NEG     R4              ;SET TO WHOLE NUMBER
015712  032737  000020  000552          BIT     #20,UDES        ;SEE IF CORE DUMP
015720  001402                          BEQ     DFOBO           ;IF NOT: BR
015722  000241                          CLC
015724  006004                          ROR     R4              ;SET TO FC/2
015726  060401                  DFOBO:  ADD     R4,R1           ;POINT TO START OF WRITE DATA
015730  060402                          ADD     R4,R2           ;POINT TO START OF READ DATA
015732  032737  000001  000556          BIT     #1,FMCNT        ;SEE IF ODD FRAME COUNT
015740  001401                          BEQ     DFOA            ;IF NOT: BR
015742  105722                          TSTB    (R2)+           ;BUMP POINTER
015744  032737  000020  000552  DFOA:   BIT     #20,UDES        ;SEE IF CORE DUMP
015752  001431                          BEQ     DFOA4           ;IF NOT: BR
```

```
3791  015754  000241                    CLC
3792  015756  132742  000001            BITB    #1,-(R2)        ;SEE IF BIT 0 = 1
3793  015762  001401                    BEQ     DF0A0           ;IF NOT: BR
3794  015764  000261                    SEC
3795  015766  106012            DF0A0:  RORB    (R2)
3796  015770  000241                    CLC
3797  015772  132712  000001            BITB    #1,(R2)
3798  015776  001401                    BEQ     DF0A1
3799  016000  000261                    SEC
3800  016002  106012            DF0A1:  RORB    (R2)            ;POSITION BITS FOR REVERSE CORE DUMP
3801  016004  000241                    CLC
3802  016006  132712  000001            BITB    #1,(R2)
3803  016012  001401                    BEQ     DF0A2
3804  016014  000261                    SEC
3805  016016  106012            DF0A2:  RORB    (R2)
3806  016020  000241                    CLC
3807  016022  132712  000001            BITB    #1,(R2)
3808  016026  001401                    BEQ     DF0A3
3809  016030  000261                    SEC
3810  016032  106012            DF0A3:  RORB    (R2)
3811  016034  005202                    INC     R2              ;RESET POINTER
3812  016036  124142            DF0A4:  CMPB    -(R1),-(R2)     ;TEST DATA CHARACTER
3813  016040  001010                    BNE     DF1             ;IF NOT GOOD: BR
3814  016042  105037  000656            CLRB    BBC             ;CLEAR BAD RECORD COUNTER
3815  016046  000411                    BR      DF2
3816  016050  122122            DF0:    CMPB    (R1)+,(R2)+     ;CHECK DATA
3817  016052  001003                    BNE     DF1             ;IF BAD: BR
3818  016054  105037  000656            CLRB    BBC             ;CLEAR BAD RECORD CNTR
3819  016060  000404                    BR      DF2
3820  016062  004737  016666    DF1:    JSR     PC,DRPKF        ;GO GET DROPS AND PICKS
3821  016066  004737  016160            JSR     PC,DERR         ;GO DO PRINT
3822  016072  005205            DF2:    INC     R5              ;BUMP CHAR CNTR
3823  016074  001405                    BEQ     DF3             ;IF DONE ALL: BR
3824  016076  032737  010000  000562    BIT     #10000,RDCMD    ;SEE IF READ REVERSE
3825  016104  001761                    BEQ     DF0             ;IF NOT: BR
3826  016106  000716                    BR      DF0A            ;ELSE CONTINUE READ REV
3827  016110  005037  000664    DF3:    CLR     HDRFL           ;CLEAR HEADER FLAG
3828  016114  005737  000704            TST     DERFL           ;SEE IF HAD DATA ERROR
3829  016120  001416                    BEQ     DFX             ;IF NOT: BR
3830  016122  005737  000706            TST     SERFL
3831  016126  001013                    BNE     DFX             ;IF NOT DATA ERROR ONLY: BR
3832  016130  013704  000674            MOV     UNP,R4
3833  016134  032737  010000  000562    BIT     #10000,RDCMD    ;SEE IF READ REVERSE
3834  016142  001003                    BNE     DF4             ;IF SO: BR
3835  016144  005264  001130            INC     DATER1(R4)      ;BUMP DATA ERROR FORWARD COUNTER
3836  016150  000402                    BR      DFX
3837  016152  005264  001170    DF4:    INC     DEREV1(R4)      ;BUMP REVERSE DATA ERROR
3838  016156  000207            DFX:    RTS     PC              ;EXIT
3839
```

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

016160  032777  002000  162420  DERR:   BIT     #2000,@SWR      ;SEE IF SHOULD PRINT ERRORS
016166  001067                          BNE     DERR4           ;++G BRANCH IF NOT
016170  005237  000670          DERR0:  INC     PFLG            ;SET PRINT FLAG
016174  005737  000664                  TST     HDRFL           ;SEE IF HAVE PRINTED HEADER
016200  001007                          BNE     DERR0A          ;IF SO: BR
016202  004737  022736                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
016206  012704  024736                  MOV     #MSG1,R4        ;LOAD ERROR MSG ADDR
016212  104000                          TTOUTT                  ;PRINT ERROR
016214  004737  021156                  JSR     PC,FRPRT        ;PRINT F OR R
016220  012704  024755          DERR0A: MOV     #MSG4,R4
016224  104000                          TTOUTT                  ;PRINT CHAR NO. HEADER
016226  010203                          MOV     R2,R3
016230  162703  033472                  SUB     #RDATA,R3       ;POINT TO CHAR
016234  005303                          DEC     R3
016236  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
016244  001402                          BEQ     DERR0B          ;IF NOT: BR
016246  010503                          MOV     R5,R3           ;GET CHAR NUMBER
016250  005103                          COM     R3
016252  104002                  DERR0B: OCTPP                   ;PRINT CHAR NUMBER
016254  012704  024743                  MOV     #MSG2,R4
016260  104000                          TTOUTT                  ;PRINT EXPECTED DATA
016262  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
016270  001402                          BEQ     DERR0C          ;IF NOT: BR
016272  111103                          MOVB    (R1),R3         ;GET CHAR
016274  000401                          BR      DERR0D
016276  114103                  DERR0C: MOVB    -(R1),R3        ;LOAD EXPECTED DATA
016300  004737  024166          DERR0D: JSR     PC,DOUT         ;GO PRINT CHAR
016304  012704  024750                  MOV     #MSG3,R4
```

```
3896  016310  104000                        TTOUTT                  ;PRINT RECIEVED DATA
3897  016312  032737  010000  000562        BIT     #10000,RDCMD    ;SEE IF READ REVERSE
3898  016320  001402                        BEQ     DERR1           ;IF NOT: BR
3899  016322  111203                        MOVB    (R2),R3         ;GET CHAR
3900  016324  000401                        BR      DERR2
3901  016326  114203              DERR1:    MOVB    -(R2),R3
3902  016330  004737  024166      DERR2:    JSR     PC,DOUT         ;PRINT BAD CHAR
3903  016334  032737  010000  000562        BIT     #10000,RDCMD    ;SEE IF READ REVERSE
3904  016342  001001                        BNE     DERR4           ;++G BRANCH IF NOT
3905  016344  122122              DERR3:    CMPB    (R1)+,(R2)+     ;RESET POINTERS
3906  016346  105237  000656      DERR4:    INCB    BBC             ;BUMP BAD RECORD CNTR
3907  016352  122737  000010  000656        CMPB    #10,BBC         ;SEE IF BLD BTH
3908  016360  001120                        BNE     DEREX           ;IF NOT: BR
3909  016362  032777  002000  162216        BIT     #2000,@SWR      ;SEE IF PRINT INHIBIT
3910  016370  001003                        BNE     1$              ;IF SO: BR
3911  016372  012704  025067                MOV     #MSG15,R4
3912  016376  104000                        TTOUTT                  ;PRINT BLD BTH
3913  016400  105037  000656      1$:       CLRB    BBC             ;RESET BAD RECORD CNTR
3914  016404  000337  000656                SWAB    BBC             ;POSITION BLD BTH AMOUNT
3915  016410  105237  000656                INCB    BBC             ;BUMP AMOUNT
3916  016414  122737  000003  000656        CMPB    #3,BBC          ;SEE IF HAD 3 BLD BTHS
3917  016422  101052                        BHI     DERR4B          ;IF NOT: BR
3918  016424  000337  000656                SWAB    BBC             ;REPOSITION BBC
3919  016430  022705  177767                CMP     #177767,R5      ;SEE IF ON LAST EIGHT CHARS
3920  016434  101470                        BLOS    DERR6           ;IF SO: BR
3921  016436  012705  177767                MOV     #177767,R5      ;SET CHAR CNTR TO 8
3922  016442  032737  010000  000562        BIT     #10000,RDCMD    ;SEE IF READ REVERSE
3923  016450  001416                        BEQ     DERR4A          ;IF NOT: BR
3924  016452  012701  027464                MOV     #WDATA,R1       ;GET START OF BUFFER
3925  016456  012702  033472                MOV     #RDATA,R2       ;GET START OF BUFFER
3926  016462  062701  000010                ADD     #10,R1
3927  016466  062702  000010                ADD     #10,R2          ;POINT TO START +10
3928  016472  032737  000001  000556        BIT     #1,FMCNT        ;SEE IF ODD FRAME COUNT
3929  016500  001430                        BEQ     DEREX           ;IF NOT: BR
3930  016502  105722                        TSTB    (R2)+           ;BUMP POINTER
3931  016504  000446                        BR      DEREX
3932  016506  013737  000556  000642  DERR4A: MOV    FMCNT,TEMP1     ;LOAD CHAR COUNT
3933  016514  005437  000642                NEG     TEMP1           ;++G FORM TWO'S COMPLEMENT
3934  016520  162737  000010  000642        SUB     #10,TEMP1       ;POINT TO BUFFER -8
3935  016526  013701  000642                MOV     TEMP1,R1        ;POINT TO NEXT CHAR
3936  016532  062701  027464                ADD     #WDATA,R1       ;POINT TO NEXT WRITE CHAR
3937  016536  013702  000642                MOV     TEMP1,R2        ;POINT TO END OF READ DATA -8 FORWARD
3938  016542  062702  033472                ADD     #RDATA,R2       ;POINT TO NEXT CHAR
3939  016546  000425                        BR      DEREX           ;EXIT
3940  016550  000337  000656      DERR4B:   SWAB    BBC             ;REPOSITION BBC
3941  016554  062705  000024                ADD     #24,R5          ;SKIP 20 CHARS
3942  016560  103416                        BCS     DERR6           ;IF EXCEED RECORD SIZE: BR
3943  016562  032737  010000  000562        BIT     #10000,RDCMD    ;SEE IF READ REVERSE
3944  016570  001405                        BEQ     DERR5           ;IF NOT: BR
3945  016572  162701  000024                SUB     #24,R1
3946  016576  162702  000024                SUB     #24,R2          ;RESET POINTERS
3947  016602  000407                        BR      DEREX
3948  016604  062701  000024      DERR5:    ADD     #24,R1          ;SKIP 20 CHARS
3949  016610  062702  000024                ADD     #24,R2          ;SKIP FORWARD 20 CHARS
3950  016614  000402                        BR      DEREX
3951  016616  012705  177777      DERR6:    MOV     #-1,R5          ;SET TO EOR
```

```
3952  016622  005777  161760          DEREX:  TST     @SWR            ;++G BRANCH IF CONTINUE ON ERROR
3953  016626  100012                          BPL     DEREX1          ;++G
3954  016630  104006                          STOPP
3955  016632  005737  000670                  TST     PFLG            ;SEE IF PRINTED
3956  016636  001006                          BNE     DEREX1          ;IF SO: BR
3957  016640  032777  002000  161740          BIT     #2000,@SWR      ;SEE IF SHOULD PRINT
3958  016646  001002                          BNE     DEREX1          ;IF NOT: BR
3959  016650  000137  016170                  JMP     DERRO           ;ELSE PRINT
3960  016654  005037  000670          DEREX1: CLR     PFLG            ;CLEAR FLAG
3961  016660  005237  000704                  INC     DERFL           ;BUMP DATA ERROR FLAG
3962  016664  000207                          RTS     PC              ;RETURN
3963
```

```
                                        ;*************************************************************
                                        ;DROPS AND PICKS SUBROUTINE:
                                        ;
                                        ;THIS SUBROUTINE IS USED TO ACCUMULATE FROM
                                        ;EACH BAD DATA CHARACTER FOUND THE NUMBER
                                        ;OF BITS WHICH WERE EITHER DROPPED OR PICKED UP.
                                        ;TWO COUNTERS PER SLAVE ARE USED TO ACCUMULATE THIS
                                        ;INFORMATION AND CAN STORE UP TO 32K DROPS
                                        ;OR PICKS BEFORE OVERFLOWING.  IF OVERFLOW IS
                                        ;ABOUT TO OCCUR, THESE ACCUMULATORS ARE
                                        ;PRINTED IN OCTAL AND RESET TO ZERO.
                                        ;THE CONTENTS OF THE ACCUMULATORS MAY BE
                                        ;DISPLAYED AT ANY TIME BY SETTING CONSOLE
                                        ;SWITCH FOURTEEN TO A ONE (1). THE PRINTOUT WILL OCCUR
                                        ;AT THE END OF THE CURRENT BLOCK CYCLE.
                                        ;*************************************************************

016666  005037  000642          DRPKF:  CLR     TEMP1
016672  005037  000644                  CLR     TEMP2
016676  005037  000646                  CLR     TEMP3
016702  111137  000642                  MOVB    (R1),TEMP1      ;LOAD GOOD CHAR
016706  111237  000644                  MOVB    (R2),TEMP2      ;LOAD BAD CHAR
016712  013704  000674                  MOV     UNP,R4
016716  016437  000770  000720          MOV     PIK1(R4),BPKP
016724  016437  001010  000716          MOV     DRP1(R4),BDPP
016732  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
016740  001005                          BNE     DRPK            ;IF SO: BR
016742  124142                          CMPB    -(R1),-(R2)     ;POINT TO CHAR
016744  112137  000642                  MOVB    (R1)+,TEMP1     ;LOAD GOOD CHAR
016750  112237  000644                  MOVB    (R2)+,TEMP2     ;LOAD BAD CHAR
016754  004737  016766          DRPK:   JSR     PC,DROP         ;GET DROPS
016760  004737  017204                  JSR     PC,PICK         ;GET PICKS
016764  000207                          RTS     PC              ;EXIT
016766  113703  000642          DROP:   MOVB    TEMP1,R3        ;R3 = GOOD CHAR
016772  113704  000644                  MOVB    TEMP2,R4        ;R4 = BAD CHAR
016776  140403                  DPC:    BICB    R4,R3           ;GET DROPS/PICKS
017000  001001                          BNE     DPCG            ;IF SOME: BR
017002  000207                          RTS     PC              ;RETURN
017004  012737  000010  000710  DPCG:   MOV     #10,BCNT        ;SET NUMBER TO CHECK
017012  132703  000001          DPC0:   BITB    #1,R3           ;SEE IF DROPPED OR PICKED THIS BIT
017016  001455                          BEQ     DPC2            ;IF NOT: BR
017020  105737  000646                  TSTB    TEMP3           ;SEE IF ON PICKS
017024  001016                          BNE     DPC1            ;IF SO: BR
017026  005277  161664                  INC     @BDPP           ;BUMP DROP CNTR
017032  005777  161660                  TST     @BDPP
017036  100045                          BPL     DPC2            ;IF NO OVERFLOW: BR
017040  032777  002000  161540          BIT     #2000,@SWR      ;SEE IF HAVE PRINTED DATA
017046  001402                          BEQ     DPC0A           ;IF SO: BR
017050  004737  022736                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
017054  004737  017250          DPC0A:  JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
017060  000415                          BR      DPC2A
017062  005277  161632          DPC1:   INC     @BPKP           ;BUMP PICK CNTR
017066  005777  161626                  TST     @BPKP           ;SEE IF OVERFLOW
017072  100027                          BPL     DPC2            ;IF NOT: BR
017074  032777  002000  161504          BIT     #2000,@SWR      ;SEE IF HAVE PRINTED DATA
```

```
4020  017102  001402                        BEQ     DPC1A           ;IF SO: BR
4021  017104  004737  022736                JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
4022  017110  004737  017250        DPC1A:  JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
4023  017114  013704  000674        DPC2A:  MOV     UNP,R4
4024  017120  016403  001010                MOV     DRP1(R4),R3     ;SET DROP POINTER
4025  017124  016404  000770                MOV     PIK1(R4),R4     ;SET PICK POINTER
4026  017130  012737  000010  000710        MOV     #10,BCNT        ;SET NUMBER OF BITS
4027  017136  005023                DPC2B:  CLR     (R3)+           ;CLEAR DROPS
4028  017140  005024                        CLR     (R4)+           ;CLEAR PICK
4029  017142  005337  000710                DEC     BCNT            ;SEE IF DONE
4030  017146  001373                        BNE     DPC2B           ;IF NOT: BR
4031  017150  000207                        RTS     PC              ;EXIT
4032  017152  000241                DPC2:   CLC
4033  017154  106003                        RORB    R3              ;GET NEXT BIT
4034  017156  005337  000710                DEC     BCNT            ;SEE IF DONE
4035  017162  001407                        BEQ     DPC3
4036  017164  062737  000002  000720        ADD     #2,BPKP
4037  017172  062737  000002  000716        ADD     #2,BDPP
4038  017200  000704                        BR      DPC0            ;++G CONTINUE
4039  017202  000207                DPC3:   RTS     PC              ;RETURN
4040  017204  013704  000674        PICK:   MOV     UNP,R4          ;GET UNIT POINTER
4041  017210  016437  000770  000720        MOV     PIK1(R4),BPKP   ;SET PICK POINTER
4042  017216  016437  001010  000716        MOV     DRP1(R4),BDPP   ;SET DROP POINTER
4043  017224  113704  000642                MOVB    TEMP1,R4        ;R4 = GOOD CHAR
4044  017230  113703  000644                MOVB    TEMP2,R3        ;R3 = BAD CHAR
4045  017234  112737  000001  000646        MOVB    #1,TEMP3        ;SET PICK FLAG
4046  017242  004737  016776                JSR     PC,DPC          ;GO CHECK PICKS
4047  017246  000207                        RTS     PC              ;EXIT
4048  017250  012704  025363        DPPRT:  MOV     #MSG26,R4
4049  017254  104000                        TTOUTT                  ;PRINT DROP HEADER
4050  017256  013704  000674                MOV     UNP,R4
4051  017262  016437  001010  000716        MOV     DRP1(R4),BDPP   ;SET DROP POINTER
4052  017270  016437  000770  000720        MOV     PIK1(R4),BPKP   ;SET PICK POINTER
4053  017276  062737  000016  000716        ADD     #16,BDPP
4054  017304  062737  000016  000720        ADD     #16,BPKP
4055  017312  012737  000010  000710        MOV     #10,BCNT        ;SET NUMBER TO PRINT
4056  017320  017703  161372        DPPRT0: MOV     @BDPP,R3
4057  017324  104002                        OCTPP                   ;PRINT DROPS
4058  017326  005337  000710                DEC     BCNT            ;SEE IF DONE
4059  017332  001404                        BEQ     DPPRT1          ;IF NOT: BR
4060  017334  162737  000002  000716        SUB     #2,BDPP         ;BUMP POINTER
4061  017342  000766                        BR      DPPRT0          ;CONTINUE FOR ALL 8 BITS
4062  017344  012737  000010  000710  DPPRT1: MOV     #10,BCNT        ;SET NUMBER TO PRINT
4063  017352  012704  025374                MOV     #MSG27,R4
4064  017356  104000                        TTOUTT                  ;PRINT PICK HEADER
4065  017360  017703  161334        DPPRT2: MOV     @BPKP,R3
4066  017364  104002                        OCTPP                   ;PRINT PICKS
4067  017366  005337  000710                DEC     BCNT            ;SEE IF DONE
4068  017372  001404                        BEQ     DPPRTX          ;IF SO: BR
4069  017374  162737  000002  000720        SUB     #2,BPKP         ;BUMP POINTER
4070  017402  000766                        BR      DPPRT2          ;CONTINUE FOR ALL 8 BITS
4071  017404  000207                DPPRTX: RTS     PC              ;RETURN
```

```
                                        ;*********************************************************
                                        ;STATUS CHECK SUBROUTINE:
                                        ;
                                        ;THIS SUBROUTINE IS USED TO PERFORM A CHECK OF
                                        ;BOTH THE MASSBUS CONTROLLER (RH11) AND THE TAPE
                                        ;CONTROLLER (TM02).  THE RH11 IS CHECKED FOR ERRORS
                                        ;AS REFLECTED IN REGISTERS CS1 AND CS2 AND ALSO THAT
                                        ;THE BUS ADDRESS (BA) AND WORD COUNT (WC) ARE
                                        ;CORRECT.  THE TM02 IS CHECKED FOR DRIVE STATIS (DS),
                                        ;DRIVE ERRORS (ER), AND PROPER FRAME COUNT.  THE SPECIAL
                                        ;CHECK CHARACTERS (CRC+LRC) ARE ALSO CHECKED WHEN
                                        ;APPROPIATE (IE: NRZ READ OR WRITE).  CERTAIN TYPES
                                        ;OF DRIVE ERRORS IN PE OPERATION WILL BE ACCOMPANIED
                                        ;BY THE DISPLAY OF THE DEAD TRACK REGISTER (CC).  THESE
                                        ;TYPES ARE ER BITS 15,10,7,6.  THE PRINTOUTS OF BAD
                                        ;CRC,LRC,FC, AND BA WILL SHOW BOTH THE EXPECTED AND
                                        ;RECEIVED VALUES (IE: EXPT-RCVD).  ONLY THOSE REGISTERS
                                        ;WHICH ARE IN ERROR WILL BE PRINTED AND ALL PRINTOUTS
                                        ;ARE IN OCTAL FORMAT WITH NO LEADING ZEROS.  AS IN
                                        ;DATA ERRORS, STATUS ERRORS ARE PRECEEDED BY HEADER
                                        ;DESCRIBING THE HARDWARE UNDER TEST, THE BLOCKING
                                        ;INFORMATION, AND THE ERROR TYPE.
                                        ;*********************************************************
017406  013703  000556          ERCHK:  MOV     FMCNT,R3        ;GET FRAME COUNT
017412  032703  000001                  BIT     #1,R3           ;SEE IF ODD
017416  001401                          BEQ     ER0             ;IF NOT: BR
017420  005303                          DEC     R3              ;BUMP COUNT
017422  005403                  ER0:    NEG     R3
017424  032737  000020  000552          BIT     #20,UDES        ;SEE IF CORE DUMP
017432  001402                          BEQ     ER0B            ;IF NOT: BR
017434  000241                          CLC
017436  006003                          ROR     R3              ;SET TO FC/2
017440  032737  000010  000672  ER0B:   BIT     #10,MTC1        ;SEE IF WRITE OP
017446  001414                          BEQ     ER1             ;IF SO: BR
017450  032737  010000  000562          BIT     #10000,RDCMD
017456  001405                          BEQ     ER0A
017460  012703  033472                  MOV     #RDATA,R3
017464  162703  000002                  SUB     #2,R3           ;SET POINTER
017470  000405                          BR      ER2
017472  062703  033472          ER0A:   ADD     #RDATA,R3       ;BUILD EXPT READ ADDRESS
017476  000402                          BR      ER2
017500  062703  027464          ER1:    ADD     #WDATA,R3       ;BUILD EXPT WRITE ADDRESS
017504  010337  021132          ER2:    MOV     R3,CADER        ;SAVE ADDRESS
017510  012704  000007                  MOV     #7,R4
017514  012701  021134                  MOV     #BAER,R1
017520  005021                  ER2A0:  CLR     (R1)+           ;CLEAR FLAGS
017522  005304                          DEC     R4
017524  001375                          BNE     ER2A0
017526  020377  160762                  CMP     R3,@BA          ;SEE IF ADDRESS OK
017532  001402                          BEQ     ER2A1           ;IF SO: BR
017534  005237  021134                  INC     BAER            ;SET BUS ADDRESS ERROR
017540  032737  000010  000672  ER2A1:  BIT     #10,MTC1        ;SEE IF WRITE OPER
017546  001006                          BNE     ER2B            ;IF NOT: BR
017550  005777  160742          ER2A:   TST     @FC             ;SEE IF FC=0
```

```
4128  017554  001441                          BEQ     ER3             ;IF SO: BR
4129  017556  005237  021142                  INC     FCER            ;SET FC ERROR
4130  017562  000436                          BR      ER3             ;++G
4131  017564  032737  000040  000672  ER2B:   BIT     #40,MTC1        ;SEE IF SPACE OPER
4132  017572  001766                          BEQ     ER2A            ;IF SO: BR
4133  017574  005737  000676                  TST     TMFLG           ;SEE IF TM TIME
4134  017600  001011                          BNE     ER2D            ;IF SO: BR
4135  017602  013703  000556                  MOV     FMCNT,R3
4136  017606  005403                          NEG     R3              ;R3 = EXPT RECORD SIZE
4137  017610  020377  160702          ER2C:   CMP     R3,@FC          ;SEE IF FC = EXPT
4138  017614  001421                          BEQ     ER3             ;IF SO: BR
4139  017616  005237  021142                  INC     FCER            ;SET FC ERROR FLAG
4140  017622  000416                          BR      ER3             ;++G
4141  017624  032737  002000  000552  ER2D:   BIT     #2000,UDES      ;SEE IF PE
4142  017632  001346                          BNE     ER2A            ;IF SO: BR
4143  017634  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
4144  017642  001003                          BNE     ER2E            ;IF SO: BR
4145  017644  012703  000002                  MOV     #2,R3
4146  017650  000757                          BR      ER2C            ;LOOK FOR EXPT = 2
4147  017652  012703  000001          ER2E:   MOV     #1,R3
4148  017656  000754                          BR      ER2C            ;GO CHECK FC FOR TM
4149  017660  032777  160000  160622  ER3:    BIT     #160000,@C1     ;SEE IF COUNT ERROR
4150  017666  001441                          BEQ     ER4
4151  017670  017703  160624                  MOV     @CS,R3          ;GET CONT STATUS REG
4152  017674  042703  000307                  BIC     #307,R3         ;MASK OUT IR,OR,UNIT NO.
4153  017700  005703                          TST     R3              ;SEE IF ANY OTHER ERRORS
4154  017702  001407                          BEQ     ER3A            ;IF NOT: BR
4155  017704  005737  000676                  TST     TMFLG           ;SEE IF TAPE MARK TIME
4156  017710  001426                          BEQ     ER3B            ;IF NOT: BR
4157  017712  042703  001000                  BIC     #1000,R3        ;MASK MISSED TRANS
4158  017716  005703                          TST     R3              ;SEE IF ANY OTHER ERRORS
4159  017720  001022                          BNE     ER3B            ;IF SO: BR
4160  017722  032777  060000  160560  ER3A:   BIT     #60000,@C1      ;SEE IF EITHER TRE OR MCPE
4161  017730  001420                          BEQ     ER4             ;IF NOT: BR
4162  017732  005737  000676                  TST     TMFLG           ;SEE IF TM TIME
4163  017736  001413                          BEQ     ER3B            ;IF NOT: BR
4164  017740  017703  160560                  MOV     @ER,R3          ;GET ERROR REGISTER
4165  017744  032737  000010  000552          BIT     #10,UDES        ;SEE IF EVEN PARITY
4166  017752  001402                          BEQ     ER3A1           ;IF NOT: BR
4167  017754  042703  000100                  BIC     #100,R3         ;MASK PAR
4168  017760  042703  001000          ER3A1:  BIC     #1000,R3        ;MASK FCE
4169  017764  001402                          BEQ     ER4             ;IF NO ERRORS EXCEPT FCE: BR
4170  017766  005237  021136          ER3B:   INC     CONER           ;SET CONT ERROR FLAG
4171  017772  032777  040000  160522  ER4:    BIT     #40000,@DS      ;SEE IF DRIVE ERROR
4172  020000  001420                          BEQ     ER6             ;IF NOT: BR
4173  020002  005737  000676                  TST     TMFLG           ;SEE IF TAPE MARK TIME
4174  020006  001413                          BEQ     ER4A            ;IF NOT: BR
4175  020010  017703  160510                  MOV     @ER,R3          ;GET ER
4176  020014  032737  000010  000552          BIT     #10,UDES        ;SEE IF EVEN PARITY
4177  020022  001402                          BEQ     ER4A1           ;IF NOT: BR
4178  020024  042703  000100                  BIC     #100,R3         ;MASK PAR
4179  020030  042703  001000          ER4A1:  BIC     #1000,R3        ;MASK OUT FCE
4180  020034  001402                          BEQ     ER6             ;++G & BR IF NO OTHER ERR BITS ARE SET
4181  020036  005237  021140          ER4A:   INC     DRVER           ;SET DRIVER ERROR FLAG
4182  020042  032737  002000  000552  ER6:    BIT     #2000,UDES
4183  020050  001071                          BNE     ERPT            ;IF IN PE MODE: BR
```

```
 4184  020052  C32777  020000  160526          BIT     #20000,@SWR     ;SEE IF NO DATA CHECK
 4185  020060  001065                          BNE     ERPT            ;IF NOT: BR (ALLOW READ OF UNKNOWN TAPES)
 4186  020062  032737  000040  000672          BIT     #40,MTC1        ;SEE IF WRITE OR READ OP
 4187  020070  001461                          BEQ     ERPT            ;IF NOT: BR
 4188  020072  005737  000676                  TST     TMFLG           ;SEE IF TAPE MARK TIME
 4189  020076  001413                          BEQ     ER6A            ;IF NOT: BR
 4190  020100  013737  015536  021154          MOV     EXCRC,CRCSV     ;SAVE CRC
 4191  020106  013737  015540  021152          MOV     EXLRC,LRCSV     ;SAVE LRC
 4192  020114  005037  015536                  CLR     EXCRC
 4193  020120  012737  000023  015540          MOV     #23,EXLRC       ;SET CRC/LRC FOR TM
 4194  020126  032737  000060  000552  ER6A:   BIT     #60,UDES        ;SEE IF FORMAT 14
 4195  020134  001037                          BNE     ERPT            ;IF NOT: BR
 4196  020136  017703  160366                  MOV     @CC,R3          ;GET CRC CHARACTER
 4197  020142  042703  177000                  BIC     #177000,R3
 4198  020146  023703  015536                  CMP     EXCRC,R3
 4199  020152  001402                          BEQ     ER7
 4200  020154  005237  021146                  INC     CRCER           ;SET ERROR FLAG
 4201  020160  017703  160350          ER7:    MOV     @MR,R3          ;GET LRC
 4202  020164  000303                          SWAB    R3
 4203  020166  005703                          TST     R3
 4204  020170  100002                          BPL     ER10
 4205  020172  052703  000400                  BIS     #400,R3
 4206  020176  042703  177000          ER10:   BIC     #177000,R3
 4207  020202  023703  015540                  CMP     EXLRC,R3
 4208  020206  001412                          BEQ     ERPT            ;IF LRC GOOD: BR
 4209  020210  010337  021150                  MOV     R3,ACTLRC       ;SAVE ACTUAL LRC
 4210  020214  005237  021144                  INC     LRCER           ;SET LRC ERROR FLAG
 4211  020220  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 4212  020226  001402                          BEQ     ERPT            ;IF NOT: BR
 4213  020230  005037  021144                  CLR     LRCER           ;ELSE CLEAR LRC ERROR
 4214  020234  012703  000006          ERPT:   MOV     #6,R3
 4215  020240  005037  000706                  CLR     SERFL           ;CLEAR ERROR FLAG
 4216  020244  005037  000722                  CLR     ERSAV
 4217  020250  012704  021134                  MOV     #BAER,R4
 4218  020254  005724                  ERPTT:  TST     (R4)+           ;SEE IF ANY ERROR
 4219  020256  001004                          BNE     ERPTG           ;IF SO: BR
 4220  020260  005303                          DEC     R3
 4221  020262  001374                          BNE     ERPTT
 4222  020264  000137  021076                  JMP     ERPX1
 4223  020270  005237  000706          ERPTG:  INC     SERFL           ;SET ERROR FLAG
 4224  020274  017737  160224  000722          MOV     @ER,ERSAV       ;SAVE ERROR REGISTER
 4225  020302  032777  002000  160276          BIT     #2000,@SWR      ;SEE IF PRINT
 4226  020310  001420                          BEQ     ERPTO           ;IF SO: BR
 4227  020312  022737  000002  000712          CMP     #2,RTYFL        ;SEE IF READ RETRY
 4228  020320  001006                          BNE     ERPTG1          ;IF NOT: BR
 4229  020322  013703  000702                  MOV     RTCNT,R3
 4230  020326  005203                          INC     R3              ;BUMP RETRY COUNT
 4231  020330  020337  000602                  CMP     R3,RETRY        ;SEE IF LAST RETRY
 4232  020334  001406                          BEQ     ERPTO           ;IF SO: BR
 4233  020336  022737  000002  021140  ERPTG1: CMP     #2,DRVER        ;SEE IF TM STATUS ERROR
 4234  020344  001402                          BEQ     ERPTO           ;IF SO: BR
 4235  020346  000137  021000                  JMP     ERPXO
 4236  020352  005237  000670          ERPTO:  INC     PFLG
 4237  020356  004737  022736                  JSR     PC,PAPRT        ;PRINT HEADER
 4238  020362  013704  000652                  MOV     EMADDR,R4
 4239  020366  104000                          TTOUTT                  ;PRINT ERROR TYPE
```

```
4240  020370  004737  021156                JSR     PC,FRPRT        ;PRINT F OR R
4241  020374  005737  000676                TST     TMFLG
4242  020400  001407                        BEQ     ERPT1
4243  020402  027737  026172  000652        CMP     #MSG54,EMADDR
4244  020410  001403                        BEQ     ERPT1
4245  020412  012704  026210                MOV     #MSG56,R4       ;PRINT TM
4246  020416  104000                        TTOUTT
4247  020420  005737  021136        ERPT1:  TST     CONER
4248  020424  001414                        BEQ     ERPT2           ;IF NO CONT ERROR: BR
4249  020426  012704  025213                MOV     #MSG23,R4
4250  020432  104000                        TTOUTT                  ;PRINT C1 TAG
4251  020434  017703  160050                MOV     @C1,R3
4252  020440  104002                        OCTPP                   ;PRINT CONTROL 1
4253  020442  012704  025240                MOV     #MSG23D,R4      ;PRINT CS TAG
4254  020446  104000                        TTOUTT
4255  020450  017703  160044                MOV     @CS,R3
4256  020454  104002                        OCTPP                   ;PRINT CONT STATUS
4257  020456  005737  021140        ERPT2:  TST     DRVER
4258  020462  001414                        BEQ     ERPT3           ;IF SO DRIVE ERROR: BR
4259  020464  012704  025246                MOV     #MSG23E,R4
4260  020470  104000                        TTOUTT                  ;PRINT DS TAG
4261  020472  017703  160024                MOV     @DS,R3
4262  020476  104002                        OCTPP                   ;PRINT DRIVE STATUS
4263  020500  012704  025253                MOV     #MSG23F,R4
4264  020504  104000                        TTOUTT                  ;PRINT ER TAG
4265  020506  017703  160012                MOV     @ER,R3
4266  020512  104002                        OCTPP                   ;PRINT DRIVE ERROR
4267  020514  005737  021134        ERPT3:  TST     BAER
4268  020520  001416                        BEQ     ERPT4           ;IF NO BA ERROR: BR
4269  020522  012704  025226                MOV     #MSG23B,R4
4270  020526  104000                        TTOUTT                  ;PRINT BA TAG
4271  020530  017703  157760                MOV     @BA,R3
4272  020534  104002                        OCTPP                   ;PRINT BUS ADDRESS
4273  020536  012737  000255  000636        MOV     #255,TOB
4274  020544  004737  023670                JSR     PC,TOG          ;PRINT /
4275  020550  013703  021132                MOV     CADER,R3
4276  020554  104002                        OCTPP                   ;PRINT EXPT BUS ADDRESS
4277  020556  005737  021142        ERPT4:  TST     FCER
4278  020562  001406                        BEQ     ERPT5           ;IF NO FC ERROR: BR
4279  020564  012704  025233                MOV     #MSG23C,R4
4280  020570  104000                        TTOUTT                  ;PRINT FC TAG
4281  020572  017703  157720                MOV     @FC,R3
4282  020576  104002                        OCTPP                   ;PRINT FRAME COUNT
4283  020600  012704  025221        ERPT5:  MOV     #MSG23A,R4
4284  020604  104000                        TTOUTT                  ;PRINT WC TAG
4285  020606  017703  157700                MOV     @WC,R3
4286  020612  104002                        OCTPP                   ;PRINT WORD COUNT
4287  020614  005737  021146                TST     CRCER
4288  020620  001420                        BEQ     ERPT5A          ;IF NO CRC ERROR: BR
4289  020622  012704  026235                MOV     #MSG58,R4
4290  020626  104000                        TTOUTT                  ;PRINT CRC TAG
4291  020630  017703  157674                MOV     @CC,R3
4292  020634  042703  177000                BIC     #177000,R3
4293  020640  104002                        OCTPP                   ;PRINT ACTUAL CRC
4294  020642  012737  000255  000636        MOV     #255,TOB
4295  020650  004737  023670                JSR     PC,TOG
```

```
4296  020654  013703  015536                    MOV     EXCRC,R3
4297  020660  104002                            OCTPP                   ;PRINT EXPECTED CRC
4298  020662  005737  021144          ERPT5A:   TST     LRCER
4299  020666  001416                            BEQ     ERPT6           ;IF NO LRC ERROR: BR
4300  020670  012704  026243                    MOV     #MSG59,R4
4301  020674  104000                            TTOUTT                  ;PRINT LRC TAG
4302  020676  013703  021150                    MOV     ACTLRC,R3
4303  020702  104002                            OCTPP                   ;PRINT ACTUAL LRC
4304  020704  012737  000255  000636            MOV     #255,TOB
4305  020712  004737  023670                    JSR     PC,TOG
4306  020716  013703  015540                    MOV     EXLRC,R3
4307  020722  104002                            OCTPP                   ;PRINT EXPECTED LRC
4308  020724  005737  021140          ERPT6:    TST     DRVER
4309  020730  001422                            BEQ     ERPT7           ;IF NO DRIVE ERROR: BR
4310  020732  032737  002000  000552            BIT     #2000,UDES
4311  020740  001416                            BEQ     ERPT7           ;IF NO PE: BR
4312  020742  017704  157556                    MOV     @ER,R4
4313  020746  042704  075477                    BIC     #75477,R4       ;MASK OUT ALL BUT BITS 15,10,7,6
4314  020752  005704                            TST     R4
4315  020754  001410                            BEQ     ERPT7           ;IF NO CONDITIONALS SET: BR
4316  020756  012704  025265                    MOV     #MSG23H,R4
4317  020762  104000                            TTOUTT                  ;PRINT CC TAG
4318  020764  017703  157540                    MOV     @CC,R3
4319  020770  042703  177000                    BIC     #177000,R3      ;MASK CC
4320  020774  104002                            OCTPP                   ;PRINT CHECK CHARACTERS
4321  020776  000240                  ERPT7:    NOP
4322  021000  032777  100000  157600  ERPX0:    BIT     #100000,@SWR    ;SEE IF STOP ON ERROR
4323  021006  001412                            BEQ     ERPX            ;IF NOT: BR
4324  021010  104006                            STOPP
4325  021012  005737  000670                    TST     PFLG            ;SEE IF HAVE PRINTED
4326  021016  001006                            BNE     ERPX            ;IF SO: BR
4327  021020  032777  002000  157560            BIT     #2000,@SWR      ;SEE IF SHOULD PRINT
4328  021026  001002                            BNE     ERPX            ;IF NOT: BR
4329  021030  000137  020352                    JMP     ERPTO           ;PRINT ERROR
4330  021034  005037  000670          ERPX:     CLR     PFLG
4331  021040  012777  000011  157442            MOV     #11,@C1         ;DRIVE CLEAR
4332  021046  017704  157454                    MOV     @AS,R4
4333  021052  010477  157450                    MOV     R4,@AS          ;CLEAR AS
4334  021056  013704  000510                    MOV     C1,R4
4335  021062  005204                            INC     R4
4336  021064  152714  000100                    BISB    #100,(R4)       ;RESET TRE
4337  021070  013777  000552  157444            MOV     UDES,@C2        ;RESET TC
4338  021076  032737  000040  000672  ERPX1:    BIT     #40,@TC1
4339  021104  001411                            BEQ     ERPX2           ;IF NOT READ/WRITE OP: BR
4340  021106  005737  000676                    TST     TMFLG
4341  021112  001406                            BEQ     ERPX2           ;IF NOT TM TIME: BR
4342  021114  013737  021154  015536            MOV     CRCSV,EXCRC     ;RESTORE CRC
4343  021122  013737  021152  015540            MOV     LRCSV,EXLRC     ;RESTORE LRC
4344  021130  000207                  ERPX2:    RTS     PC              ;EXIT
4345  021132  000000                  CADER:    0                       ;EXPT ADDRESS SAVE
4346  021134  000000                  BAER:     0
4347  021136  000000                  CONER:    0
4348  021140  000000                  DRVER:    0
4349  021142  000000                  FCER:     0
4350  021144  000000                  LRCER:    0
4351  021146  000000                  CRCER:    0
```

```
4352  021150  000000                    ACTLRC: 0
4353  021152  000000                    LRCSV:  0
4354  021154  000000                    CRCSV:  0
4355
4356                                    ;*************************************************************
4357                                    ;F FOR FORWARD/R FOR REVERSE PRINT SUBROUTINE:
4358                                    ;
4359                                    ;THIS SUBROUTINE IS USED TO PRINT OUT THE
4360                                    ;TAPE DIRECTION USED WHEN ANY ERROR IS
4361                                    ;DETECTED IN STATUS OF READ OR WRITE, DATA, OR
4362                                    ;SPACING OPERATIONS.
4363                                    ;*************************************************************
4364
4365  021156  032737  000010  000672    FRPRT:  BIT     #10,MTC1        ;SEE IF WRITE COMMAND
4366  021164  001413                            BEQ     FREX            ;IF SO: BR
4367  021166  032737  000002  000672            BIT     #2,MTC1         ;SEE IF REVERSE
4368  021174  001404                            BEQ     FRO             ;IF NOT: BR
4369  021176  012704  025123                    MOV     #MSG17,R4
4370  021202  104000                            TTOUTT                  ;PRINT R
4371  021204  000403                            BR      FREX
4372  021206  012704  025120            FRO:    MOV     #MSG16,R4
4373  021212  104000                            TTOUTT                  ;PRINT F
4374  021214  000207                    FREX:   RTS     PC              ;EXIT
4375
```

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

021216  005037  000642          TAPG:    CLR     TEMP1
021222  013777  000550  157270           MOV     DVN,@CS          ;SET DRIVE NO.
021230  032777  010000  157264  TAPG0:   BIT     #10000,@DS       ;SEE IF HAVE MOL
021236  001026                           BNE     TAPG3            ;IF SO: BR
021240  005237  000642                   INC     TEMP1            ;SEE IF TIMED OUT
021244  001371                           BNE     TAPG0            ;WAIT FOR READY
021246  004737  022736                   JSR     PC,PAPRT         ;PRINT CYCLE NUMBER
021252  032737  000010  000672           BIT     #10,MTC1         ;SEE IF WRITE OP
021260  001004                           BNE     TAPG1            ;IF NOT: BR
021262  012704  024762                   MOV     #MSG5,R4
021266  104000                           TTOUTT                   ;PRINT WRITE ERR
021270  000405                           BR      TAPG2
021272  012704  024767          TAPG1:   MOV     #MSG6,R4
021276  104000                           TTOUTT                   ;PRINT READ ERR
021300  004737  021156                   JSR     PC,FRPRT         ;PRINT F OR R
021304  012704  025343          TAPG2:   MOV     #MSG25,R4
021310  104000                           TTOUTT                   ;PRINT NO MOL ERR
021312  104006                           STOPP
021314  032777  020000  157200  TAPG3:   BIT     #20000,@DS       ;SEE IF PIP RESET
021322  001411                           BEQ     TAPG3F           ;IF SO: BR
021324  004737  022736                   JSR     PC,PAPRT         ;PRINT HEADER
021330  012704  027322                   MOV     #MSG116,R4
021334  104000                           TTOUTT                   ;PRINT REWINDING MESSAGE
021336  032777  020000  157156  1$:      BIT     #20000,@DS
021344  001374                           BNE     1$               ;AWAIT PIP RESET
021346  022737  000026  000672  TAPG3F:  CMP     #26,MTC1         ;SEE IF WRITE TM
021354  001003                           BNE     TAPG3A           ;IF NOT: BR
```

```
4432  021356  012704  177777              MOV     #-1,R4          ;ELSE SET FC FOR -1
4433  021362  000406                      BR      TAPG3B
4434  021364  013704  000556      TAPG3A: MOV     FMCNT,R4
4435  021370  032704  000001              BIT     #1,R4
4436  021374  001401                      BEQ     TAPG3B
4437  021376  005304                      DEC     R4
4438  021400  000261              TAPG3B: SEC
4439  021402  006004                      ROR     R4              ;SET WC = FC/2 FOR NORMAL FORMAT
4440  021404  032737  000020  000552      BIT     #20,UDES        ;SEE IF CORE DUMP FORMAT
4441  021412  001402                      BEQ     TAPG3C          ;IF NOT: BR
4442  021414  000261                      SEC
4443  021416  006004                      ROR     R4              ;SET WC = FC/4 FOR CORE DUMP
4444  021420  010477  157066      TAPG3C: MOV     R4,@WC          ;SET WORD COUNT
4445  021424  012777  000011  157056      MOV     #11,@C1         ;DRIVE CLEAR
4446  021432  017777  157060  157056      MOV     @FC,@FC         ;RESET FC LOADED
4447  021440  005737  000566              TST     INTRF           ;SEE IF INTERCHANGE READ
4448  021444  001407                      BEQ     TAPG3D          ;IF NOT: BR
4449  021446  032737  000040  000672      BIT     #40,MTC1        ;SEE IF READ OP
4450  021454  001403                      BEQ     TAPG3D          ;IF NOT: BR
4451  021456  012777  000003  157050      MOV     #3,@MR          ;SET INTERCHANGE READ MAINT. MODE
4452  021464  013704  000672      TAPG3D: MOV     MTC1,R4         ;GET COMMAND
4453  021470  042704  177707              BIC     #177707,R4      ;MASK OP CODE
4454  021474  022704  000030              CMP     #30,R4          ;SEE IF SPACE OP CODE
4455  021500  001403                      BEQ     TAPG3E          ;IF SO: BR
4456  021502  012737  177740  000666      MOV     #-40,STAL       ;SET INTERRUPT DELAY MULT TO 40
4457  021510  052737  000101  000672      BIS     #101,MTC1       ;SET INTERRUPT ENABLE AND GO
4458  021516  000240              TAPG3E: NOP
4459  021520  013777  000672  156762      MOV     MTC1,@C1        ;EXECUTE COMMAND
4460  021526  005077  157052              CLR     @PSW            ;CLEAR PRIORITY
4461  021532  005037  000642              CLR     TEMP1
4462  021536  005237  000642      TAPG4:  INC     TEMP1           ;SEE IF HAVE TIMED OUT
4463  021542  001375                      BNE     TAPG4           ;IF NOT: BR
4464  021544  005237  000666              INC     STAL
4465  021550  001372                      BNE     TAPG4           ;DO TIME DELAY MULTIPLIER
4466  021552  012777  000340  157024      MOV     #340,@PSW       ;RESET PRIORITY
4467  021560  032777  002000  157020      BIT     #2000,@SWR      ;SEE IF SHOULD PRINT ERRORS
4468  021566  001012                      BNE     TAPG6           ;IF NOT: BR
4469  021570  004737  022736              JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
4470  021574  013704  000652              MOV     EMADDR,R4
4471  021600  104000                      TTOUTT.
4472  021602  004737  021156              JSR     PC,FRPRT        ;PRINT F OR R
4473  021606  012704  025323              MOV     #MSG24,R4
4474  021612  104000                      TTOUTT                  ;PRINT NO INTERRUPT
4475  021614  005777  156766      TAPG6:  TST     @SWR            ;++G BRANCH IF CONTINUE ON ERROR
4476  021620  100001                      BPL     TAPG7
4477  021622  104006                      STOPP
4478  021624  000137  021716      TAPG7:  JMP     @INTA           ;RETURN TO CALLING ROUTINE
4479
```

SEQ 0098

```
4480
4481                                        ;TTY INTERRUPT HANDLER**********
4482
4483  021630  012777  000340  156746  TTINT:  MOV     #340,@PSW       ;RESET PSW
4484  021636  017746  156750          MOV     @TKB,-(SP)      ;++G GET CHARACTER
4485  021642  042716  000200          BIC     #200,(SP)       ;++G STRIP PARITY BIT
4486  021646  122716  000003          CMPB    #3,(SP)         ;++G SEE IF CONT C
4487  021652  001412                  BEQ     TTINTO          ;IF SO: BR
4488  021654  122716  000007          CMPB    #7,(SP)         ;++G CHECK FOR CNTL G
4489  021660  001013                  BNE     RETURN
4490  021662  027737  000176  000606  CMP     #SWREG,SWR      ;IS SOFTWARE SWITCH REGISTER USED
4491  021670  001007                  BNE     RETURN          ;NO, GET OUT
4492  021672  004737  024452          JSR     PC,CNTG         ;GO CHANGE SWREG
4493  021676  000404                  BR      RETURN          ;++G GO TO EXIT
4494
4495  021700  010046          TTINTO: MOV     R0,-(SP)        ;++G SAVE R0(REC CNTR)
4496  021702  004737  013774          JSR     PC,TINP4        ;GO GET STALL VALUES
4497  021706  012600                  MOV     (SP)+,R0        ;++G RESTORE R0(REC CNTR)
4498  021710  005726          RETURN: TST     (SP)+           ;++G POP CHAR OFF STACK
4499  021712  000002                  RTI                     ;RETURN
4500
4501                                        ;MAG TAPE INTERRUPT HANDLER**********
4502
4503  021714  000240          MTINT:  NOP
4504  021716  042777  000037  156610  MTINTA: BIC     #37,@MR         ;CLEAR MAINT MODE
4505  021724  013716  000662          MOV     RTRN,(SP)       ;++G GET RETURN ADDRESS
4506  021730  000002                  RTI                     ;++G RETURN
```

SEQ 0099

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

021732  012704  027132          ASEQ:   MOV     #MSG108,R4
021736  104000                          TTOUTT                  ;PRINT NRZ ONLY REQUEST
021740  012705  000650                  MOV     #NRZOF,R5       ;SET ADDRESS OF FLAG
021744  012701  000001                  MOV     #1.R1           ;SET SIZE OF ENTRY
021750  012702  000001                  MOV     #1,R2           ;SET UPPER LIMIT
021754  012703  000000                  MOV     #0,R3           ;SET LOWER LIMIT
021760  004737  023354                  JSR     PC,TTR          ;GO GET RESPONSE
021764  012704  026727                  MOV     #MSG104,R4
021770  104000                          TTOUTT                  ;REQUEST CONT OR NOT
021772  012705  000742                  MOV     #ASEQCF,R5      ;SET ADDRESS OF ENTRY
021776  012701  000001                  MOV     #1.R1           ;SET SIZE OF ENTRY
022002  012702  000001                  MOV     #1,R2           ;SET UPPER LIMIT
022006  012703  000000                  MOV     #0,R3           ;SET LOWER LIMIT
022012  004737  023354                  JSR     PC,TTR          ;GO GET INPUT
022016  005037  000736          ASEQ0:  CLR     ADRVN           ;CLEAR DRV NUM
022022  004737  022162          ASEQ1:  JSR     PC,HRDS         ;GO SELECT HARDWARE CONFIGURATION
022026  005737  000042                  TST     @#42            ;AUTO MODE? ++ C.W
022032  001404                          BEQ     1$              ;BRANCH - IF NO ++ C.W
022034  012737  000001  000742          MOV     #1,ASEQCF       ;SET AUTO SEQ FLAG ++ C.W
022042  000414                          BR      2$              ;DO AUTO SEQ TESTS ++ C.W
022044  012704  026673          1$:     MOV     #MSG101,R4
022050  104000                          TTOUTT                  ;PRINT DIVIDER
022052  012704  026707                  MOV     #MSG102,R4
022056  104000                          TTOUTT                  ;PRINT TM02 NUMBER
022060  013703  000736                  MOV     ADRVN,R3
022064  104002                          OCTPP                   ;PRINT TM02
022066  012704  026716                  MOV     #MSG103,R4
022072  104000                          TTOUTT                  ;PRINT SLAVE HDR
022074  012700  000746          2$:     MOV     #UN1,R0         ;POINT TO START OF SLAVE TABLE
022100  005710                  ASEQ2:  TST     (R0)            ;SEE IF END
022102  100403                          BMI     ASEQ3           ;IF SO: BR
022104  012003                          MOV     (R0)+,R3
022106  104002                          OCTPP                   ;PRINT SLAVE TABLE
022110  000773                          BR      ASEQ2           ;DO ALL
022112  004737  022366          ASEQ3:  JSR     PC,AMOD1        ;GO DO MODE 1(NRZ)
022116  004737  022562                  JSR     PC,AMOD2        ;GO DO MODE 2(PE)
022122  027737  000007  000736  ASEQ4:  CMP     #7,ADRVN        ;SEE IF DONE ALL DRIVES
022130  001403                          BEQ     ASEQX           ;IF SO: BR
022132  005237  000736                  INC     ADRVN           ;BUMP DRIVE NUMBER
022136  000751                          BR      ASEQ1           ;CONTINUE
022140  005737  000742          ASEQX:  TST     ASEQCF          ;CONTINUOUS AUTO SEQUENCE? ++ C.W
022144  001405                          BEQ     1$              ;BRANCH - IF NO ++ C.W
022146  004737  005100                  JSR     PC,TEND         ;GO DO ACT END OF PASS
022152  005737  000734                  TST     ASEQF           ;CONTINUE
022156  001317                          BNE     ASEQ0           ;GO START AGAIN
022160  000000          1$:     HALT
```

```
4562
4563                                        ;SUBROUTINE TO SELECT AUTO SEQUENCE HARDWARE********
4564
4565  022162  005037  005156        HRDS:   CLR     REOTC           ;CLEAR EOT UNIT CNTR
4566  022166  005037  000642                CLR     TEMP1
4567  022172  012777  000040  156320        MOV     #40,@CS         ;INIT
4568  022200  013777  000736  156312        MOV     ADRVN,@CS       ;SET DRIVE
4569  022206  017701  156324                MOV     @DT,R1          ;READ DRIVE TYPE
4570  022212  032777  010000  156300        BIT     #10000,@CS      ;TEST FOR NON-EXISTANT DRIVE
4571  022220  001403                        BEQ     HRDS1           ;IF DRIVE AVAIL: BR
4572  022222  005726              HRDS0:    TST     (SP)+           ;RESET STACK POINTER
4573  022224  000137  022122                JMP     @SEQ4           ;GO SEE IF TRIED ALL DRIVES
4574  022230  042701  002007        HRDS1:  BIC     #2007,R1        ;MASK SLAVE TYPE
4575  022234  022701  140010                CMP     #140010,R1      ;++G SEE IF TU16/TE16 TAPE
4576  022240  001370                        BNE     HRDS0           ;IF NOT: BR
4577  022242  005000                        CLR     R0
4578  022244  012701  000746                MOV     #UN1,R1         ;SET START OF SLAVE TABLE
4579  022250  005737  003040                TST     CHNFLG          ;++G BRANCH IF NOT IN CHAIN MODE
4580  022254  001410                        BEQ     HRDS2
4581  022256  122737  000006  000041        CMPB    #6,@#41         ;++G BRANCH IF NOT LOADED VIA TMDP
4582  022264  001004                        BNE     HRDS2
4583  022266  005737  000736                TST     ADRVN           ;++G BRANCH IF NOT DRIVE 0
4584  022272  001001                        BNE     HRDS2           ;++G
4585  022274  005200                        INC     R0              ;++G DO NOT TEST DRIVE 0 SLAVE 0
4586                                                                ;++G IF TMDP CHAIN
4587  022276  010077  156240        HRDS2:  MOV     R0,@C2          ;SELECT SLAVE
4588  022302  032777  010000  156212        BIT     #10000,@DS      ;SEE IF SLAVE AVAIL FOR TEST(MOL)
4589  022310  001403                        BEQ     HRDS3           ;IF NOT: BR
4590  022312  005237  000642                INC     TEMP1           ;SET SLAVE FOUND FLAG
4591  022316  010021                        MOV     R0,(R1)+        ;LOAD SLAVE TABLE
4592  022320  022700  000007        HRDS3:  CMP     #7,R0           ;SEE IF DONE ALL SLAVES
4593  022324  001402                        BEQ     HRDS4           ;IF SO: BR
4594  022326  005200                        INC     R0              ;ELSE BUMP SLAVE NUMBER
4595  022330  000762                        BR      HRDS2           ;CONTINUE SELECTION
4596  022332  005737  000642        HRDS4:  TST     TEMP1           ;SEE IF FOUND ANY SLAVES
4597  022336  001731                        BEQ     HRDS0           ;IF NOT: BR
4598  022340  013737  000642  005156        MOV     TEMP1,REOTC     ;SET NUMBER OF UNITS
4599  022346  000337  000642                SWAB    TEMP1
4600  022352  053737  000642  005156        BIS     TEMP1,REOTC     ;SET EOT CNTR
4601  022360  012711  177777                MOV     #-1,(R1)        ;TERMINATE SLAVE TABLE
4602  022364  000207                        RTS     PC              ;RETURN TO SEQ
```

```
4603
4604                                        ;SUBROUTINE TO SELECT NRZ AUTO TEST MODE**********
4605
4606  022366  005037  000654        AMOD1:  CLR     BLCNTR          ;ASSURE BLOCK COUNTER IS 0
4607  022372  012701  000746                MOV     #UN1,R1         ;GET START OF SLAVE TABLE
4608  022376  052721  001700        AMOD1A: BIS     #1700,(R1)+     ;SET ALL SLAVE TO NRZ,NORM,ODD
4609  022402  005111                        COM     (R1)
4610  022404  001402                        BEQ     AMOD1B          ;IF FILLED ALL SLAVES: BR
4611  022406  005111                        COM     (R1)
4612  022410  000772                        BR      AMOD1A          ;ELSE DO ALL
4613  022412  005111                AMOD1B: COM     (R1)
4614  022414  004737  005172              JSR     PC,RWNDA        ;GO REWIND ALL AVAIL SLAVES
4615  022420  012737  000006  000740      MOV     #6,ABLCNT       ;SET NUMBER OF BLOCKS FOR MODE 1
4616  022426  012737  174000  000556      MOV     #-4000,FMCNT    ;SET FC = 4000
4617  022434  012737  000100  000554      MOV     #100,RCNT       ;SET REC CNTR = 100
4618  022442  013737  000736  000550      MOV     ADRVN,DVN       ;SELECT DRIVE
4619  022450  012737  000001  000560      MOV     #1,PATRN        ;SELECT PATTERN 1
4620  022456  005037  000564              CLR     TMEX            ;ASSURE NO TMK
4621  022462  005037  000566              CLR     INTRF           ;ASSURE NORMAL READ
4622  022466  004737  003414              JSR     PC,STAUTO       ;GO DO AUTO MODE 1
4623  022472  012737  000010  000560      MOV     #10,PATRN       ;SELECT PATTERN 10
4624  022500  004737  003414              JSR     PC,STAUTO       ;GO DO PATTERN 10
4625  022504  012737  000014  000560      MOV     #14,PATRN       ;SELECT PATTERN 14
4626  022512  004737  003414              JSR     PC,STAUTO
4627  022516  005737  000650              TST     NRZOF           ;SEE IF NRZ ONLY
4628  022522  001411                        BEQ     AMOD1C          ;IF NOT: BR
4629  022524  012737  177777  000740      MOV     #-1,ABLCNT      ;FORCE TO EOT
4630  022532  012737  153624  000624      MOV     #153624,RANBAS
4631  022540  012737  032561  000626      MOV     #32561,RANSAV   ;RESET RANDOM DATA BASE
4632  022546  012737  177777  000560  AMOD1C: MOV     #-1,PATRN       ;SELECT AUTO RANDOM DATA
4633  022554  004737  003414              JSR     PC,STAUTO
4634  022560  000207                        RTS     PC              ;RETURN TO SEQ
```

```
4635
4636                                    ;SUBROUTINE TO SELECT PE AUTO TEST MODE**********
4637
4638  022562  005737  003040    AMOD2:  TST     CHNFLG          ;++G BRANCH IF IN CHAIN MODE
4639  022566  001003                    BNE     1$              ;++G
4640  022570  005737  000650            TST     NRZOF           ;SEE IF NRZ ONLY
4641  022574  001057                    BNE     AMOD2X          ;IF SO: BR
4642  022576  005037  000654    1$:     CLR     BLCNTR          ;CLEAR BLOCK CNTR
4643  022602  012701  000746            MOV     #UN1,R1         ;SET START OF SLAVE TABLE
4644  022606  042711  001700    AMOD2A: BIC     #1700,(R1)      ;CLEAR NRZ
4645  022612  052721  002300            BIS     #2300,(R1)+     ;SET TO PE NORM, ODD
4646  022616  005111                    COM     (R1)            ;SEE IF END OF TABLE
4647  022620  001402                    BEQ     AMOD2B          ;IF SO: BR
4648  022622  005111                    COM     (R1)
4649  022624  000770                    BR      AMOD2A          ;CONTINUE
4650  022626  005111            AMOD2B: COM     (R1)
4651  022630  004737  005172            JSR     PC,RWNDA        ;REWIND ALL SLAVES
4652  022634  012737  000006  000740    MOV     #6,ABLCNT       ;SET AUTO BLOCK COUNT
4653  022642  012737  174000  000556    MOV     #-4000,FMCNT    ;SET FC = 4000
4654  022650  012737  000100  000554    MOV     #100,RCNT       ;SET REC CNTR TO 100
4655  022656  012737  000010  000560    MOV     #10,PATRN       ;SELECT PATTERN 10
4656  022664  004737  003414            JSR     PC,$TAUTO       ;GO DO AUTO SEQ
4657  022670  012737  000014  000560    MOV     #14,PATRN       ;SELECT PATTERN 14
4658  022676  004737  003414            JSR     PC,$TAUTO
4659  022702  012737  000015  000560    MOV     #15,PATRN       ;SELECT PATTERN 15
4660  022710  004737  003414            JSR     PC,$TAUTO
4661  022714  012737  177777  000740    MOV     #-1,ABLCNT      ;FORCE TO END OF TAPE
4662  022722  012737  177777  000560    MOV     #-1,PATRN       ;SELECT AUTO RANDOM DATA
4663  022730  004737  003414            JSR     PC,$TAUTO
4664  022734  000207            AMOD2X: RTS     PC              ;RETURN TO SEQ
4665
4666
```

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

022736  012704  025040           PAPRT:  MOV     #MSG12,R4
022742  104000                           TTOUTT                          ;PRINT DRIVE HEADER
022744  013703  000550                   MOV     DVN,R3
022750  104002                           OCTPP                           ;PRINT DRIVE NUMBER
022752  012704  025024                   MOV     #MSG11,R4
022756  104000                           TTOUTT                          ;PRINT UNIT HEADER
022760  013703  000552                   MOV     UDES,R3
022764  042703  177770                   BIC     #177770,R3
022770  104002                           OCTPP                           ;PRINT UNIT NUMBER
022772  012704  026251                   MOV     #MSG60,R4
022776  104000                           TTOUTT                          ;PRINT DENSITY TAG
023000  013703  000552                   MOV     UDES,R3
023004  000303                           SWAB    R3
023006  042703  177770                   BIC     #177770,R3
023012  104002                           OCTPP                           ;PRINT DENSITY
023014  012704  026255                   MOV     #MSG61,R4
023020  104000                           TTOUTT                          ;PRINT PARITY TAG
023022  005003                           CLR     R3
023024  032737  000010  000552           BIT     #10,UDES
023032  001402                           BEQ     PAPRT0
023034  012703  000001                   MOV     #1,R3
023040  104002                   PAPRT0: OCTPP                           ;PRINT PARITY
023042  012704  026261                   MOV     #MSG62,R4
023046  104000                           TTOUTT                          ;PRINT FORMAT TAG
023050  013703  000552                   MOV     UDES,R3
023054  000241                           CLC
023056  006003                           ROR     R3
023060  006003                           ROR     R3
023062  006003                           ROR     R3                      ;PONTION FORMAT
023064  006003                           ROR     R3
023066  042703  177760                   BIC     #177760,R3
023072  104002                           OCTPP                           ;PRINT FORMAT
023074  012704  025001                   MOV     #MSGB,R4
023100  104000                           TTOUTT                          ;PRINT PATRN TAG
023102  032777  000400  155476           BIT     #400,@SWR               ;SEE IF RANDOM DATA
023110  001406                           BEQ     PAPRTB                  ;IF NOT: BR
023112  012737  000122  000636   PAPRTA: MOV     #122,TOB
023120  004737  023670                   JSR     PC,TOG                  ;PRINT R
023124  000411                           BR      PAPRTD
023126  005737  000734           PAPRTB: TST     ASEQF                   ;SEE IF AUTO SEQ
```

```
 4723  023132  001403                        BEQ     PAPRTC          ;IF NOT: BR
 4724  023134  005737  000560                TST     PATRN           ;SEE IF AUTO RANDOM
 4725  023140  100764                        BMI     PAPRTA          ;IF SO: BR
 4726  023142  013703  000560        PAPRTC: MOV     PATRN,R3
 4727  023146  104002                        OCTPP                   ;PRINT PATRN NUMBER
 4728  023150  012704  025054        PAPRTD: MOV     #MSG13,R4
 4729  023154  104000                        TTOUTT                  ;PRINT BLOCK NO. HEADER
 4730  023156  013703  000654                MOV     BLCNTR,R3
 4731  023162  104002                        OCTPP                   ;PRINT NUMBER
 4732  023164  012704  025062                MOV     #MSG14,R4
 4733  023170  104000                        TTOUTT                  ;PRINT REC NO. HEADER
 4734  023172  010003                        MOV     R0,R3
 4735  023174  032737  000010  000672        BIT     #10,MTC1        ;SEE IF WRITE OPERATION
 4736  023202  001404                        BEQ     PAPRT1          ;IF SO: BR
 4737  023204  032737  010000  000562        BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 4738  023212  001016                        BNE     PAPRT3          ;IF SO: BR
 4739  023214  013703  000554        PAPRT1: MOV     RCNT,R3
 4740  023220  005737  000676                TST     TMFLG           ;SEE IF TAPE MARK TIME
 4741  023224  001010                        BNE     PAPRT2          ;IF SO: BR
 4742  023226  022737  012260  000662        CMP     #B1,RTRN
 4743  023234  001003                        BNE     PAPRTY          ;IF NOT BACK SPACE: BR
 4744  023236  005737  000712                TST     RTYFL
 4745  023242  001402                        BEQ     PAPRT3          ;IF NOT RETRY: BR
 4746  023244  160003                PAPRTY: SUB     R0,R3           ;GET RECORD NUMBER
 4747  023246  005203                PAPRT2: INC     R3
 4748  023250  104002                PAPRT3: OCTPP                   ;PRINT RECORD NUMBER
 4749  023252  012737  000055  000636        MOV     #55,TOB         ;LOAD DASH (-)
 4750  023260  004737  023670                JSR     PC,TOG          ;PRINT DASH (-)
 4751  023264  013703  000554                MOV     RCNT,R3
 4752  023270  104002                        OCTPP                   ;PRINT RECORD COUNT
 4753  023272  012704  024774                MOV     #MSG7,R4
 4754  023276  104000                        TTOUTT                  ;PRINT RECORD SIZE HEADER
 4755  023300  013703  000556                MOV     FMCNT,R3        ;GET CHARACTER COUNT
 4756  023304  005303                        DEC     R3
 4757  023306  005103                        COM     R3              ;REMOVE TWOS COMPLEMENT
 4758  023310  104002                        OCTPP                   ;PRINT RECORD SIZE
 4759  023312  012737  000001  000664        MOV     #1,HDRFL        ;SET HEADER FLAG
 4760  023320  000207                        RTS     PC              ;RETURN
 4761
```

```
4762
4763                                    ;*************************************************************
4764                                    ;RANDOM NUMBER GENERATOR SUBROUTINE:
4765                                    ;
4766                                    ;THIS SUBROUTINE IS USED TO GENERATE THE RANDOM
4767                                    ;NUMBERS REQUIRED FOR USE AS RANDOM DATA,
4768                                    ;RECORD COUNT, AND CHARACTER COUNT.
4769                                    ;*************************************************************
4770
4771  023322  063737  000626  000624  RANG:  ADD     RANSAV,RANBAS
4772  023330  063737  000624  000626         ADD     RANBAS,RANSAV   ;GET NEW NUMBER
4773  023336  023701  000626                 CMP     RANSAV,R1       ;SEE IF NUMBER TOO BIG
4774  023342  101367                         BHI     RANG            ;IF SO: BR
4775  023344  020237  000626                 CMP     R2,RANSAV       ;SEE IF NUMBER TOO SMALL
4776  023350  101364                         BHI     RANG            ;IF SO: BR
4777  023352  000207                         RTS     PC              ;EXIT
4778
```

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

023354  005037  000642          TTR:    CLR     TEMP1           ;CLEAR FIRST CHARACTER FLAG
023360  005000                          CLR     R0
023362  104010                  TTR0:   TTINN                   ;GO READ CHARACTER
023364  122737  000015  000640          CMPB    #15,TIB         ;++G SEE IF CR
023372  001004                          BNE     TTR1            ;IF NOT: BR
023374  005737  000642                  TST     TEMP1           ;SEE IF FIRST CHARACTER
023400  001436                          BEQ     TTR5            ;IF SO: BR
023402  000426                          BR      TTR2            ;++G ELSE GO LOAD VALUE
023404  122737  000060  000640  TTR1:   CMPB    #60,TIB         ;++G SEE IF CHAR IS LESS THAN 0
023412  101401                          BLOS    TTR1A           ;IF NOT: BR
023414  000431                          BR      TINER           ;++G ELSE GO TO ERROR
023416  122737  000070  000640  TTR1A:  CMPB    #70,TIB         ;++G SEE IF CHAR IS GREATER THAN 7
023424  101001                          BHI     TTR1B           ;IF NOT: BR
023426  000424                          BR      TINER           ;++G ELSE GO TO ERROR
023430  005237  000642          TTR1B:  INC     TEMP1           ;SET FIRST CHARACTER FLAG
023434  006300                          ASL     R0
023436  006300                          ASL     R0              ;SHIFT 3 LEFT
023440  006300                          ASL     R0
023442  042737  177770  000640          BIC     #177770,TIB     ;STRIP ASCII
023450  053700  000640                  BIS     TIB,R0          ;LOAD CHARACTER
023454  005301                          DEC     R1              ;SEE IF DONE
023456  001341                          BNE     TTR0            ;IF NOT: BR
023460  020002                  TTR2:   CMP     R0,R2           ;SEE IF EXCEEDED MAXIMUM LIMIT
023462  101401                          BLOS    TTR3            ;IF NOT: BR
023464  000405                          BR      TINER           ;++G ELSE GO TO ERROR
023466  020300                  TTR3:   CMP     R3,R0           ;SEE IF BELOW MINIMUM LIMIT
023470  101401                          BLOS    TTR4            ;IF NOT: BR
023472  000402                          BR      TINER           ;++G ELSE GO TO ERROR
023474  010015                  TTR4:   MOV     R0,(R5)         ;LOAD VALUE
023476  000207                  TTR5:   RTS     PC              ;EXIT
023500  012704  025763          TINER:  MOV     #MSG43,R4
023504  104000                          TTOUTT                  ;PRINT?
023506  162716  000020                  SUB     #20,(SP)        ;RESET SP TO START OF VALUE ROUTINE
023512  000207                          RTS     PC              ;REDO VALUE ENTRY
```

```
4830
4831                                        ;TTY READ SUBROUTINE**********
4832
4833  023514  005277  155070        TTIN:   INC     @TKS
4834  023520  105777  155064        TTIN1:  TSTB    @TKS
4835  023524  100375                        BPL     TTIN1
4836  023526  017737  155060  000640        MOV     @TKB,TIB
4837  023534  042737  000200  000640        BIC     #200,TIB        ;++G STRIP PARITY BIT
4838  023542  105777  155046        TTIN2:  TSTB    @TPS
4839  023546  100375                        BPL     TTIN2
4840  023550  113777  000640  155040        MOVB    TIB,@TPB
4841  023556  000207                        RTS     PC
4842
4843                                        ;TTY OUTPUT SUBROUTINE************
4844
4845  023560  112437  000636        TTOUT:  MOVB    (R4)+,TOB
4846  023564  122737  000043  000636        CMPB    #43,TOB
4847  023572  001444                        BEQ     TEX
4848  023574  122737  000045  000636        CMPB    #45,TOB
4849  023602  001407                        BEQ     TCRLF
4850  023604  122737  000041  000636        CMPB    #41,TOB
4851  023612  001435                        BEQ     TBELL           ;DO BELL
4852  023614  004737  023670                JSR     PC,TOG
4853  023620  000757                        BR      TTOUT
4854  023622  112737  000015  000636  TCRLF:  MOVB    #15,TOB
4855  023630  004737  023670                JSR     PC,TOG
4856  023634  012703  000006                MOV     #6,R3
4857  023640  005037  000636        TCRLFA: CLR     TOB
4858  023644  004737  023670                JSR     PC,TOG
4859  023650  005303                        DEC     R3
4860  023652  001372                        BNE     TCRLFA          ;DO FILLERS
4861  023654  112737  000012  000636        MOVB    #12,TOB
4862  023662  004737  023670                JSR     PC,TOG
4863  023666  000734                        BR      TTOUT
4864
4865  023670  105777  154720        TOG:    TSTB    @TPS
4866  023674  100375                        BPL     TOG
4867  023676  113777  000636  154712        MOVB    TOB,@TPB
4868  023704  000207                TEX:    RTS     PC
4869  023706  012703  000002        TBELL:  MOV     #2,R3
4870  023712  012737  000007  000636  TBELA:  MOV     #7,TOB
4871  023720  004737  023670                JSR     PC,TOG
4872  023724  005303                        DEC     R3
4873  023726  001371                        BNE     TBELA
4874  023730  000713                        BR      TTOUT
4875
4876
```

```
                                            ;OCTAL OUTPUT SUBROUTINE************

023732  005037  024164          OCTP:   CLR     OFL                     ;CLEAR FLAG FOR LEADING ZERO
023736  000403                          BR      OCTPE1
023740  012737  000001  024164  OCTPE:  MOV     #1,OFL
023746  010304                  OCTPE1: MOV     R3,R4                   ;SEE IF NUMBER IS ZERO
023750  001006                          BNE     OCTPO                   ;IF NOT ZERO: BR
023752  005737  024164                  TST     OFL
023756  001003                          BNE     OCTPO
023760  004737  024144                  JSR     PC,OCTPG1               ;ELSE PRINT ZERO
023764  000450                          BR      OCTP3                   ;++G SPACE AND EXIT
023766  032704  100000          OCTPO:  BIT     #100000,R4              ;SEE IF MSD = 1
023772  001406                          BEQ     OCTP1                   ;IF NOT: BR
023774  012704  000001                  MOV     #1,R4
024000  004737  024122                  JSR     PC,OCTPG                ;PRINT 1
024004  000137  024016                  JMP     OCTP2
024010  005004                  OCTP1:  CLR     R4
024012  004737  024122                  JSR     PC,OCTPG                ;PRINT 0
024016  010304                  OCTP2:  MOV     R3,R4
024020  006004                          ROR     R4
024022  006004                          ROR     R4
024024  006004                          ROR     R4                      ;POSITION DIGIT
024026  006004                          ROR     R4
024030  000304                          SWAB    R4
024032  004737  024122                  JSR     PC,OCTPG                ;PRINT DIGIT 2
024036  010304                          MOV     R3,R4
024040  006004                          ROR     R4
024042  000304                          SWAB    R4
024044  004737  024122                  JSR     PC,OCTPG                ;PRINT DIGIT 3
024050  010304                          MOV     R3,R4
024052  006104                          ROL     R4
024054  006104                          ROL     R4
024056  000304                          SWAB    R4
024060  004737  024122                  JSR     PC,OCTPG                ;PRINT DIGIT 4
024064  010304                          MOV     R3,R4
024066  006004                          ROR     R4
024070  006004                          ROR     R4
024072  006004                          ROR     R4
024074  004737  024122                  JSR     PC,OCTPG
024100  010304                          MOV     R3,R4
024102  004737  024122                  JSR     PC,OCTPG                ;PRINT DIGIT 5
024106  012737  000240  000636  OCTP3:  MOV     #240,TOB
024114  004737  023670                  JSR     PC,TOG                  ;PRINT SPACE
024120  000207                          RTS     PC                      ;EXIT
024122  042704  177770          OCTPG:  BIC     #177770,R4
024126  001004                          BNE     OCTPGO
024130  005737  024164                  TST     OFL
024134  001001                          BNE     OCTPGO
024136  000207                          RTS     PC
024140  005237  024164          OCTPGO: INC     OFL
024144  052704  000260          OCTPG1: BIS     #260,R4
024150  010437  000636                  MOV     R4,TOB
024154  004737  023670                  JSR     PC,TOG
024160  010304                          MOV     R3,R4
024162  000207                          RTS     PC
024164  000000                  OFL:    0                               ;FIRST CHAR FLAG
```

```
                                        ;DATA CHARACTER OUTPUT SUBROUTINE************
024166  005037  000636          DOUT:   CLR     TOB
024172  012704  000010                  MOV     #10,R4          ;SET NUMBER TO PRINT
024176  110337  000636                  MOVB    R3,TOB
024202  105777  154406          DOUT1:  TSTB    @TPS
024206  100375                  BPL     DOUT1
024210  132737  000200  000636          BITB    #200,TOB
024216  105737  000636                  TSTB    TOB             ;++G
024222  100004                  BPL     DOUT2           ;++G
024224  012777  000061  154364          MOV     #061,@TPB
024232  000403                  BR      DOUT3
024234  012777  000060  154354  DOUT2:  MOV     #060,@TPB
024242  006137  000636          DOUT3:  ROL     TOB
024246  005304                  DEC     R4
024250  001354                  BNE     DOUT1
024252  000207                  RTS     PC
024254  013703  000646          DOUTD:  MOV     TEMP3,R3
024260  000303                  SWAB    R3
024262  004737  024166                  JSR     PC,DOUT
024266  013703  000646                  MOV     TEMP3,R3
024272  004737  024166                  JSR     PC,DOUT
024276  000207                  RTS     PC

                                        ;++G TU16/TE16 SERIAL NUMBER PRINT SUBROUTINE*******
024300  010304          SNPT:   MOV     R3,R4
024302  000304                  SWAB    R4
024304  006004                  ROR     R4
024306  006004                  ROR     R4
024310  006004                  ROR     R4
024312  006004                  ROR     R4
024314  004737  024356          JSR     PC,SNPG         ;PRINT FIRST DIGIT
024320  010304                  MOV     R3,R4
024322  000304                  SWAB    R4
024324  004737  024356          JSR     PC,SNPG         ;PRINT SECOND DIGIT
024330  010304                  MOV     R3,R4
024332  006004                  ROR     R4
024334  006004                  ROR     R4
024336  006004                  ROR     R4
024340  006004                  ROR     R4
024342  004737  024356          JSR     PC,SNPG         ;PRINT THIRD DIGIT
024346  010304                  MOV     R3,R4
024350  004737  024356          JSR     PC,SNPG         ;PRINT FOURTH DIGIT
024354  000207                  RTS     PC              ;EXIT
024356  012737  000260  000636  SNPG:   MOV     #260,TOB        ;SET NUMBER BASE
024364  042704  177760                  BIC     #177760,R4      ;MASK NUMBER
024370  050437  000636                  BIS     R4,TOB          ;BUILD DIGIT
024374  004737  023670                  JSR     PC,TOG          ;GO TYPE
024400  000207                  RTS     PC              ;RETURN

                                        ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ^G TO ALLOW CHANGING
                                        ;OF LOC.176.
                                        ;CALL IS BY WAY OF CKSWRR
```

```
                                                        ;LOCATIONS USED:
                                                TEMPST: .WORD   0
                                                COUNT:  .WORD   0
                                                RDSW:   .WORD   0
000176  000606          CKSWR:  CMP     #SWREG,SWR              ;SOFTWARE SWITCH REG PRESENT
                                BNE     OUT                     ;NO, GET OUT
154164                          TSTB    @TKS                    ;YES, WAIT FOR
                                BPL     OUT                     ;READY, GET CHARACTER
154160  000640                  MOV     @TKB,TIB                ;AND STRIP OFF
177600  000640                  BIC     #177600,TIB             ;THE GARBAGE
000007  000640                  CMP     #7,TIB          ;IS IT A <^G>
                                BNE     OUT
027436                  CNTG:   MOV     #$CNTG,R4
                                TTOUTT
027442                  CNTLU:  MOV     #SMSWR,R4
                                TTOUTT
154114                          MOV     @SWR,R3
023740                          JSR     PC,OCTPE
027451                          MOV     #SANEW,R4
                                TTOUTT
024402                  $READ:  CLR     TEMPST
000007  024404                  MOV     #7,COUNT
                        1$:     TTINN                   ;GO  READ A CHARACTER
000025  000640                  CMPB    #25,TIB         ;IS IT A ^U?
                                BNE     2$              ;BRANCH IF NOT
                        3$:     BR CNTLU                ;START OVER
000015  000640          2$:     CMPB    #15,TIB         ;IS IT A <CR>?
                                BNE     4$              ;BRANCH IF NOT
000200  024406                  MOV     #200,RDSW
027461                          MOV     #MCRLF,R4
                                TTOUTT
000007  024404                  CMP     #7,COUNT        ;WAS IT FIRST CHARACTER
                                BNE     7$              ;CHANGE SWR IF NOT FIRST ONE
                                BR      OUT             ;GET OUT
000060  000640          4$:     CMPB    #60,TIB
                                BGT     5$
000067  000640                  CMPB    #67,TIB
                                BGE     6$
025763                  5$:     MOV     #MSG43,R4
                                TTOUTT
                                BR      3$              ;START OVER IF NOT LEGAL CHARACTER
024402                  6$:     ASL     TEMPST
                                ASL     TEMPST
                                ASL     TEMPST
000060  000640                  BICB    #60,TIB         ;GET NITTY-GRITTY
000640  024402                  BISB    TIB,TEMPST
024404                          DEC     COUNT           ;ONLY WANT 6   DIGITS
                                BEQ     5$
                                BR      1$
024402  153720          7$:     MOV     TEMPST,@SWR     ;CHANGE SWITCH REGISTER CONTENTS
000207                  OUT:    RTS     PC              ;RETURN TO BODY OF PROGRAM
                        ;HALT HANDLER**********


000000                  STOP:   HALT
104004                          CKSWRR                  ;CHECK FOR CONTROL G
000207                          RTS     PC
```

```
                                                  ;TRAP HANDLER**************
024676  016677  000002  153700  TRAP30: MOV      2(6),@PSW          ;ADJUST PSW
024704  011666  000002                  MOV      @SP,2(6)           ;PLACE RETURN ADDRESS OVER PSW
024710  162716  000002                  SUB      #2,@SP             ;SUB. 2 FROM RETURN ADDRESS
024714  013646                          MOV      @(6)+,-(6)
024716  062716  120724                  ADD      #TABLE-104000,@SP          ;GET SUBROUTINE STARTING ADDRESS
024722  013607                          MOV      @(SP)+,PC          ;GO TO SUBROUTINE
024724  023560                  TABLE:  TTOUT
024726  025732                          OCTP
024730  024410                          CKSWR
024732  024670                          STOP
024734  023514                          TTIN
        104000                  TTOUTT= 104000
        104002                  OCTPP=  104002
        104004                  CKSWRR= 104004
        104006                  STOPP=  104006
        104010                  TTINN=  104010
```

```
                                        ;ERROR MESSAGES**********
024736 042052 020105     043  MSG1:   .ASCII  /*DE #/
024743     045 035507  021440  MSG2:   .ASCII  /%G; #/
024750 041045 020073     043  MSG3:   .ASCII  /%B; #/
024755     045 047103  021440  MSG4:   .ASCII  /%CN #/
024762 053452 020105     043  MSG5:   .ASCII  /*WE #/
024767     052 042522  021440  MSG6:   .ASCII  /*RE #/
024774 051052 020123     043  MSG7:   .ASCII  /*RS #/
025001     052 040520  051124  MSG8:   .ASCII  /*PATRN #/
025006 020116     043
025011     040 047123  020072  MSG9:   .ASCII  / SN: #/
025016     043
025017     052 042523  021440  MSG10:  .ASCII  /*SE #/
025024 051452 040514  042526  MSG11:  .ASCII  /*SLAVE NO. #/
025032 047040 027117  021440
025040 042045 044522  042526  MSG12   .ASCII  /%DRIVE NO. #/
025046 047040 027117  021440
025054 025045 047102  021440  MSG13:  .ASCII  /%*BN #/
025062 051052 020116     043  MSG14:  .ASCII  /*RN #/
025067     045 020041  020040  MSC15:  .ASCII  /%!          BAD RECORD%%#/
025074 020040 020040  020040
025102 041040 042101  051040
025110 041505 051117  022504
025116 021445
025120 043040     043        MSG16:  .ASCII  / F#/
025123     040 021522         MSG17:  .ASCII  / R#/
025126 020041 047505  020124  MSG20:  .ASCII  /! EOT NO: #/
025134 047516 020072     043
025141     045 047111  042524  MSG21:  .ASCII  /%INTERCHANGE READ = #/
025146 041522 040510  043516
025154 020105 042522  042101
025162 036440 021440
025166 020445 046111  042514  MSG22:  .ASCII  /%!ILLEGAL BOT: HALT%#/
025174 040507 020114  047502
025202 033124 044040  046101
025210 022524     043
```

```
025213      045  051503  020061  MSG23:   .ASCII  /%CS1 #/
025220      043
025221      045  041527  021440  MSG23A:  .ASCII  /%WC #/
025226   041045  020101     043  MSG23B:  .ASCII  /%BA #/
025233      045  041506  021440  MSG23C:  .ASCII  /%FC #/
025240   041445  031123  021440  MSG23D:  .ASCII  /%CS2 #/
025246   042045  020123     043  MSG23E:  .ASCII  /%DS #/
025253      045  051105  021440  MSG23F:  .ASCII  /%ER #/
025260   040445  020123     043  MSG23G:  .ASCII  /%AS #/
025265      045  045503  021440  MSG23H:  .ASCII  /%CK #/
025272   042045  020102     043  MSG23I:  .ASCII  /%DB #/
025277      045  051115  021440  MSG23J:  .ASCII  /%MR #/
025304   042045  020124     043  MSG23K:  .ASCII  /%DT #/
025311      045  041524  021440  MSG23L:  .ASCII  /%TC #/
025316   051445  020116     043  MSG23M:  .ASCII  /%SN #/
025323      045  047041  020117  MSG24:   .ASCII  /%!NO INTERRUPT%#/
025330   047111  042524  051122
025336   050125  022524     043
025343      045  047041  020117  MSG25:   .ASCII  /%!NO MOL: HALT%#/
025350   047515  035114  044040
025356   046101  022524     043
025363      045  051104  050117  MSG26:   .ASCII  /%DROPS: #/
025370   035123  021440
025374   050045  041511  051513  MSG27:   .ASCII  /%PICKS: #/
025402   020072     043
025405      045     043          MSG28:   .ASCII  /%#/
025407      045  052045  030115  MSG30:   .ASCII  '%%TM02-TU16/TE16 AUTO SEQUENCE (CZTUAIO)%#'    ;++G
025414   026462  052524  033061
025422   052057  030505  020066
025430   052501  047524  051440
025436   050505  042525  041516
025444   020105  041450  052132
025452   040525  030111  022451
025460      043
025461      045  041445  052132  MSG31:   .ASCII  '%%CZTUAIO TM02-TU16/TE16 RELIAB%%#'
025466   040525  030111  052040
025474   030115  026462  052524
```

```
 5178  025502  033061  052057  030505
 5179  025510  020066  042522  044514
 5180  025516  041101  022445     043
 5181
 5182  025523     045  046123  053101  MSG32:  .ASCII  /%SLAVE NUMBER = #/
 5183  025530  020105  052516  041115
 5184  025536  051105  036440  021440
 5185
 5186  025544  042045  047105  044523  MSG33:  .ASCII  /%DENSITY = #/
 5187  025552  054524  036440  021440
 5188
 5189  025560  050045  051101  052111  MSG34:  .ASCII  /%PARITY = #/
 5190  025566  020131  020075     043
 5191
 5192  025573     045  042522  047503  MSG35:  .ASCII  /%RECORD COUNT = #/
 5193  025600  042122  041440  052517
 5194  025606  052116  036440  021440
 5195
 5196  025614  041445  040510  040522  MSG36:  .ASCII  /%CHARACTER COUNT = #/
 5197  025622  052103  051105  041440
 5198  025630  052517  052116  036440
 5199  025636  021440
 5200
 5201  025640  050045  052101  042524  MSG37:  .ASCII  /%PATTERN NUMBER = #/
 5202  025646  047122  047040  046525
 5203  025654  042502  020122  020075
 5204  025662     043
 5205  025663     045  044523  043516  MSG38:  .ASCII  /%SINGLE PASS = #/
 5206  025670  042514  050040  051501
 5207  025676  020123  020075     043
 5208  025703     045  047105  042524  MSG40:  .ASCII  /%ENTER STALLS%READ = #/
 5209  025710  020122  052123  046101
 5210  025716  051514  051045  040505
 5211  025724  020104  020075     043
 5212
 5213  025731     045  051127  052111  MSG41:  .ASCII  /%WRITE = #/
 5214  025736  020105  020075     043
 5215
 5216  025743     045  052524  047122  MSG42:  .ASCII  /%TURN AROUND = #/
 5217  025750  040440  047522  047125
 5218  025756  020104  020075     043
 5219
 5220  025763     045  022477     043  MSG43:  .ASCII  /%?%#/
 5221
 5222  025767     045  047105  042524  MSG44:  .ASCII  /%ENTER YOZZLE STALL = #/
 5223  025774  020122  047531  055132
 5224  026002  042514  051440  040524
 5225  026010  046114  036440  021440
 5226
 5227  026016  042445  051122  040440  MSG45:  .ASCII  /%ERR AMT #/
 5228  026024  052115  021440
 5229
 5230  026030  043045  020103     043  MSG46:  .ASCII  /%FC #/
 5231
 5232  026035     045  040503  021440  MSG47:  .ASCII  /%CA #/
 5233
```

```
5234  026042  020445  047516  041040  MSG48:  .ASCII  /%!NO BOT ON REWIND: HALT%#/
5235  026050  052117  047440  020116
5236  026056  042522  044527  042116
5237  026064  020075  040510  052114
5238  026072  021445
5239
5240  026074  047040  052117  040440  MSG49:  .ASCII  / NOT AVAIL #/
5241  026102  040526  046111  021440
5242  026110  044440  046114  043505  MSG50:  .ASCII  / ILLEGAL DRIVE TYPE #/
5243  026116  046101  042040  044522
5244  026124  042526  052040  050131
5245  026132  020105      043
5246  026135      045  042045  044522  MSG52:  .ASCII  /%%DRIVE NUMBER = #/
5247  026142  042526  047040  046525
5248  026150  042502  020122  020075
5249  026156      043
5250
5251  026157      045  047506  046522  MSG53:  .ASCII  /%FORMAT = #/
5252  026164  052101  036440  021440
5253
5254  026172  053452  020105  046524  MSG54:  .ASCII  /*WE TM#/
5255  026200      043
5256
5257  026201      052  042523  052040  MSG55:  .ASCII  /*SE TM#/
5258  026206  021515
5259
5260  026210  052040  021515          MSG56:  .ASCII  / TM#/
5261
5262  026214  047040  047117  042455  MSG57:  .ASCII  / NON-EXIST SLAVE#/
5263  026222  044530  052123  051440
5264  026230  040514  042526      043
5265  026235      045  051103  020103  MSG58:  .ASCII  /%CRC #/
5266  026242      043
5267  026243      045  051114  020103  MSG59:  .ASCII  /%LRC #/
5268  026250      043
5269  026251      052  020104      043  MSG60:  .ASCII  /*D #/
5270  026255      052  020120      043  MSG61:  .ASCII  /*P #/
5271  026261      052  020106      043  MSG62:  .ASCII  /*F #/
5272
5273  026265      045  047452  044522  MSG64:  .ASCII  /%*ORIGINAL ERROR*#/
5274  026272  044507  040516  020114
5275  026300  051105  047522  025122
5276  026306      043
5277
5278  026307      045  042522  051124  MSG65:  .ASCII  /%RETRY: #/
5279  026314  035131  021440
5280
5281  026320  020452  042523  051040  MSG66:  .ASCII  /*!SE RTRY #/
5282  026326  051124  020131      043
5283
5284  026333      052  042441  040522  MSG67:  .ASCII  /*!ERASE#/
5285  026340  042523      043
5286
5287  026343      045  042522  042522  MSG68:  .ASCII  /%REREV: #/
5288  026350  035126  021440
5289  026354  052045  050101  020105  MSG69:  .ASCII  /%TAPE MARK = #/
```

```
 5290  026362  040515  045522  036440
 5291  026370  021440
 5292
 5293  026372  020445  047516  042040  MSG70:  .ASCII  /%!NO DRY FROM REWIND: HALT%#/
 5294  026400  054522  043040  047522
 5295  026406  020115  042522  044527
 5296  026414  042116  020072  040510
 5297  026422  052114  021445
 5298  026426  047040  047117  042455  MSG71:  .ASCII  / NON-EXIST DRIVE#/
 5299  026434  044530  052123  042040
 5300  026442  044522  042526      043
 5301  026447      045  042522  053506  MSG72:  .ASCII  /%REFWD: #/
 5302  026454  035104  021440
 5303  026460  053445  042524  051122  MSG73:  .ASCII  /%WTERR: #/
 5304  026466  020072      043
 5305  026471      044  042522  044507  MSG74:  .ASCII  /%REGISTER START = #/
 5306  026476  052123  051105  051440
 5307  026504  040524  052122  036440
 5308  026512  021440
 5309  026514  053045  041505  047524  MSG75:  .ASCII  /%VECTOR = #/
 5310  026522  020122  020075      043
 5311  026527      045  042504  042522  MSG76:  .ASCII  /%DEREV: #/
 5312  026534  035126  021440
 5313  026540  042045  043105  042127  MSG77:  .ASCII  /%DEFWD: #/
 5314  026546  020072      043
 5315  026551      045  047041  047117  MSG78:  .ASCII  /%!NON-RETRYABLE WRITE ERROR: ER #/
 5316  026556  051055  052105  054522
 5317  026564  041101  042514  053440
 5318  026572  044522  042524  042440
 5319  026600  051122  051117  020072
 5320  026606  051105  021440
 5321  026612  020445  047516  026516  MSG79:  .ASCII  /%!NON-RETRYABLE READ ERROR: ER #/
 5322  026620  042522  051124  040531
 5323  026626  046102  020105  042522
 5324  026634  042101  042440  051122
 5325  026642  051117  020072  051105
 5326  026650  021440
 5327  026652  020445  042441  042116  MSG100: .ASCII  /%!!END OF PASS %#/
 5328  026660  047440  020106  040520
 5329  026666  051523  022440      043
 5330  026673      045  025052  025052  MSG101: .ASCII  /%*********%#/
 5331  026700  025052  025052  022452
 5332  026706      045
 5333  026707      052  046524  031060  MSG102: .ASCII  /*TM02 #/
 5334  026714  021440
 5335  026716  051452  040514  042526  MSG103: .ASCII  /*SLAVES #/
 5336  026724  020123      043
 5337  026727      045  052501  047524  MSG104: .ASCII  /%AUTO CONT: #/
 5338  026734  041440  047117  035124
 5339  026742  021440
 5340  026744  051045  041505  053117  MSG105: .ASCII  /%RECOVERED#/
 5341  026752  051105  042105      043
 5342  026757      052  020441  040502  MSG106: .ASCII  /*!!BAD TAPE OVERFLOW#/
 5343  026764  020104  040524  042520
 5344  026772  047440  042526  043122
 5345  027000  047514  021527
```

SEQ 0117

```
5346  027004  051045  053505  047111  MSG16A: .ASCII  /%REWIND TAPE; RESTART AT BLOCK ONE#/
5347  027012  020104  040524  042520
5348  027020  020073  042522  052123
5349  027026  051101  020124  052101
5350  027034  041040  047514  045503
5351  027042  047440  042516     043
5352  027047     045  020441  047125  MSG107: .ASCII  /%!!UNRECOVERABLE BAD SPOT/
5353  027054  042522  047503  042526
5354  027062  040522  046102  020105
5355  027070  040502  020104  050123
5356  027076  052117
5357  027100  041045  042101  051040          .ASCII  /%BAD RECORD LEFT ON TAPE%#/
5358  027106  041505  051117  020104
5359  027114  042514  052106  047440
5360  027122  020116  040524  042520
5361  027130  021445
5362  027132  047045  055122  047440  MSG108: .ASCII  /%NRZ ONLY: #/
5363  027140  046116  035131  021440
5364  027146  020452  050041  051517  MSG109: .ASCII  /*!!POSITION LOST IN RETRY#/
5365  027154  052111  047511  020116
5366  027162  047514  052123  044440
5367  027170  020116  042522  051124
5368  027176  021531
5369  027200  051445  051525  042520  MSG110: .ASCII  /%SUSPECT BAD TAPE#/
5370  027206  052103  041040  042101
5371  027214  052040  050101  021505
5372  027222  051045  050105  040505  MSG111: .ASCII  /%REPEAT: #/
5373  027230  035124  021440
5374  027234  041040  042101  052040  MSG112: .ASCII  / BAD TAPE SPOTS%#/
5375  027242  050101  020105  050123
5376  027250  052117  022523     043
5377
5378  027255     045  051440  043117  MSG113: .ASCII  /% SOFT: #/
5379  027262  035124  021440
5380
5381  027266  020045  040510  042122  MSG114: .ASCII  /% HARD: #/
5382  027274  020072     043
5383
5384  027277     045  020441  040510  MSG115: .ASCII  /%!!HARD READ ERROR#/
5385  027304  042122  051040  040505
5386  027312  020104  051105  047522
5387  027320  021522
5388  027322  020445  047125  052111  MSG116: .ASCII  /%!UNIT REWINDING: TEST WILL START AT BOT#/
5389  027330  051040  053505  047111
5390  027336  044504  043516  020072
5391  027344  042524  052123  053440
5392  027352  046111  020114  052123
5393  027360  051101  020124  052101
5394  027366  041040  052117     043
5395  027373     045  042522  047515  MSG120: .ASCII  /%REMOVE TMDP FROM UNIT UNDER TEST%#/
5396  027400  042526  052040  042115
5397  027406  020120  051106  045517
5398  027414  052440  044516  020124
5399  027423  047125  042504  020123
5400  027430  042524  052123  021445
5401  027436  057045  021507          SCNTG:  .ASCII  /%^G#/
```

```
5402  027442  051445  051127  020075  SMSWR:  .ASCII  /%SWR= #/
5403  027450     043
5404  027451     040  047040  053505  SMNEW:  .ASCII  /  NEW= #/
5405  027456  020075     043
5406  027461     045     043          MCRLF:  .ASCII  /%#/
5407
5408           027464                       .EVEN
5409  027464  000000                WDATA:  0                       ;WRITE BUFFER
5410
5411           033472                       .=.+4004
5412  033472  000000                RDATA:  0                       ;READ BUFFER
5413
5414           000001                       .END
```

```
ABLCNT  000740          1604#  1956   4615*  4629*  4652*  4661*
ACTLRC  021150          4209*  4302   4352#
ADRVN   000736          1603#  4530*  4540   4552   4554*  4568   4583   4618
AMOD1   022366          4550   4606#
AMOD1A  022376          4608#  4612
AMOD1B  022412          4610   4613#
AMOD1C  022546          4628   4632#
AMOD2   022562          4551   4638#
AMOD2A  022606          4644#  4649
AMOD2B  022626          4647   4650#
AMOD2X  022734          4641   4664#
APATS   015062          3614   3622#
AS      000526          1529#  4332   4333*
ASEQ    021732          3210   4516#
ASEQCF  000742          1605#  4525   4534*  4556
ASEQF   000734          1511*  1602#  1865*  1884*  1933   1954   2172   2191   3173   3207   3433   4559   4722
ASEQX   022140          4553   4556#
ASEQ0   022016          1935   4530#  4560
ASEQ1   022022          4531#  4555
ASEQ2   022100          4545#  4549
ASEQ3   022112          4546   4550#
ASEQ4   022122          4552#  4573
BA      000514          1524#  2304*  2357*  2493*  2509*  2694*  2703*  2925*  2934*  3024*  4122   4271
BAER    021134          4118   4124*  4217   4267   4346#
BBC     000656          1579#  3750*  3814*  3818*  3906*  3907   3913*  3914*  3915*  3916   3918*  3940*
BCNT    000710          1592#  4003*  4026*  4029*  4034*  4055*  4058*  4062*  4067*
BDPP    000716          1595#  3989*  4008*  4009   4037*  4042*  4051*  4053*  4056   4060*
BD00    001410          1630   1733#
BD10    001430          1631   1735#
BD20    001450          1632   1737#
BD30    001470          1633   1739#
BD40    001510          1634   1741#
BD50    001530          1635   1743#
BD60    001550          1636   1745#
BD70    001570          1637   1747#
BKRT    012234          2494   3046#
BKSP    012056          2618   2631   3021#
BKTM    012140          3029   3031#
BKTM0   012202          3035   3039#
BLCNTR  000654          1578#  1912*  1953*  1956   1958*  2469   4606*  4642*  4730
BPKP    000720          1596#  3988*  4016*  4017   4036*  4041*  4052*  4054*  4065   4069*
BP00    001210          1622   1717#
BP10    001230          1623   1719#
BP20    001250          1624   1721#
BP30    001270          1625   1723#
BP40    001310          1626   1725#
BP50    001330          1627   1727#
BP60    001350          1628   1729#
BP70    001370          1629   1731#
BTADDR  001030          1641#  2463   2563
BTFLG   000724          1598#  2106   2112   2116   2129*  2497*  2521*
BTOV    007160          2476   2520#
BTOVX   007322          2549   2555#
BTOV0   007200          2524#  2573
BTOV1   007210          2527#  2547
BTOV2   007276          2544   2548#
```

```
BTOV3   007314          2552#   2554
BTPRT   007324          2064    2131    2560#
BTPRT1  007374          2571    2573#
BTPT    000730          1600#   2463*   2464    2466*   2467    2474    2524    2543    2551    2563*   2564    2570
BTSTF   000726          1599#   2063*   2065*   2548
BTUR    007400          2484    2577#
BT00    001610          1641    1754#
BT01    001714          1642    1756#
BT02    002020          1643    1758#
BT03    002124          1644    1760#
BT04    002230          1645    1762#
BT05    002334          1646    1764#
BT06    002440          1647    1766#
BT07    002544          1648    1768#
B0      012212          3026    3032    3038    3041#
B1      012260          3047    3050#   4742
B2      012270          3052#
CADER   021132          2364*   2734*   3036*   4116*   4275    4345#
CC      000530          1530#   4196    4291    4318
CCNTR   012314          1968    3091#
CHNFLG  003040          1867#   1876*   4579    4638
CKSWR   024410          4993#   5057
CKSWRR= 104004          5043    5062#
CLLAST  015252          3674    3676#
CLP     015362          3659    3687    3695#
CLPE    015406          3696    3698    3702#
CLP0    015416          3704#   3710
CLP1    015430          3705    3707#
CLP2    015470          3714    3718#
CLP3    015502          3716    3719    3721#
CL0     015170          3658#   3675
CL1     015216          3666#
CL2     015240          3663    3672#
CL3     015324          3686#   3690
CNTG    024452          2997    4492    5001#
CNTLU   024460          1932    5003#   5014
CONER   021136          4170*   4247    4347#
COUNT   024404          4991#   5010*   5020    5035*
CRCER   021146          4200*   4287    4351#
CRCLRC  015152          3446    3461    3654#
CRCSV   021154          4190*   4342    4354#
CS      000520          1526#   1937*   1978*   3211*   3219*   3221    3248*   3249*   4151    4255    4406*   4567*   4568*
                        4570
C1      000510          1522#   2102*   2105*   2226*   2227*   2251*   3202    3220    3225    4149    4160    4251    4331*
                        4334    4445*   4459*
C2      000542          1535#   1979*   2101*   2154*   2225*   2245*   2262*   3250*   4337*   4587*
DATA0   002772          1830#
DATA1   002774          1831#
DATA10  003012          1838#
DATA11  003014          1839#
DATA12  003016          1840#
DATA13  003020          1841#
DATA14  003022          1842#
DATA15  003024          1843#
DATA2   002776          1832#
DATA3   003000          1833#
```

```
DATA4   003002  1834#
DATA5   003004  1835#
DATA6   003006  1836#
DATA7   003010  1837#
DATBL   002770  1829#  3453
DATER1  001130  1685#  2040  3835*
DATR    015102  1971   3437  3633#
DATRO   015120  3637#  3640
DATO    014412  1830   3475#
DATOA   014444  3482#  3490  3494  3497
DATOB   014452  3483#  3484
DATOC   014516  3488   3495#
DATOD   014524  3498#  3506
DATOE   014534  3500#  3505
DATOF   014546  3502   3504#
DAT1    014556  1831   3511#
DAT1A   014562  3512#  3521  3544  3549  3597
DAT1B   014566  3513#  3515
DAT10   014700  1836   3564#
DAT10A  014712  3567#  3571
DAT11   014730  1839   3576#
DAT11A  014736  3578#  3581
DAT12   014750  1840   3586#
DAT12A  014760  3588#  3591
DAT13   014772  1841   3596#
DAT14   015002  1842   3601#
DAT14A  015014  3604#  3608
DAT15   015032  1843   3613#
DAT15A  015036  3614#  3620
DAT15B  015046  3616#  3618
DAT2    014576  1832   3520#
DAT3    014602  1833   3525#
DAT3A   014610  3527#  3538
DAT3B   014614  3528#  3531
DAT4    014626  1834   3536#
DAT5    014636  1835   3543#
DAT6    014644  1836   3548#
DAT7    014652  1837   3553#
DAT7A   014666  3556#  3559
DB      000532  1531#
DCHK    015544  2779   2974  3750#
DCHKO   015574  3754   3757#
DEREV1  001170  1707#  2060  3837*
DEREX   016622  3908   3929  3931  3939  3947  3950  3952#
DEREX1  016654  3953   3956  3958  3960#
DERFL   000704  1590#  3751* 3828  3961*
DERR    016160  3821   3838#
DERRO   016170  3870#  3959
DERROA  016220  3872   3877#
DERROB  016252  3883   3886#
DERROC  016276  3890   3893#
DERROD  016300  3892   3894#
DERR1   016326  3898   3901#
DERR2   016330  3900   3902#
DERR3   016344  3905#
DERR4   016346  3869   3904  3906#
```

```
DERR4A  016506          3923    3932#
DERR4B  016550          3917    3940#
DERR5   016604          3944    3948#
DERR6   016616          3920    3942    3951#
DFX     016156          3829    3831    3836    3838#
DF0     016050          3777    3816#   3825
DF0A    015744          3787    3789#   3826
DF0A0   015766          3793    3795#
DF0A1   016002          3798    3800#
DF0A2   016016          3803    3805#
DF0A3   016032          3808    3810#
DF0A4   016036          3790    3812#
DF0B    015704          3778#
DF0B0   015726          3781    3784#
DF0C    015664          3770    3774#
DF0C0   015674          3760    3762    3764    3776#
DF0D    015650          3766    3771#
DF0E    015642          3768#   3773
DF0F    015634          3765#   3769
DF1     016062          3813    3817    3820#
DF2     016072          3815    3819    3822#
DF3     016110          3823    3827#
DF4     016152          3834    3837#
DOUT    024166          3894    3902    4936#   4953    4955
DOUTD   024254          4951#
DOUT1   024202          4939#   4940    4949
DOUT2   024234          4943    4946#
DOUT3   024242          4945    4947#
DPC     016776          4000#   4046
DPCG    017004          4001    4003#
DPC0    017012          4004#   4038
DPC0A   017054          4012    4014#
DPC1    017062          4007    4016#
DPC1A   017110          4020    4022#
DPC2    017152          4005    4010    4018    4032#
DPC2A   017114          4015    4023#
DPC2B   017136          4027#   4030
DPC3    017202          4035    4039#
DPPRT   017250          2011    4014    4022    4048#
DPPRTX  017404          4068    4071#
DPPRT0  017320          4056#   4061
DPPRT1  017344          4059    4062#
DPPRT2  017360          4065#   4070
DROP    016766          3995    3998#
DRPK    016754          3991    3995#
DRPKF   016666          3820    3982#
DRP1    001010          1630#   3989    4024    4042    4051
DRP2    001012          1631#
DRP3    001014          1632#
DRP4    001016          1633#
DRP5    001020          1634#
DRP6    001022          1635#
DRP7    001024          1636#
DRP8    001026          1637#
DRVER   021140          2365*   2735*   4181*   4233    4257    4308    4348#
DS      000522          1527#   1980    1994    2070    2103    2132    2155    2157    2228    2246    2248    2308    2362
```

```
                    2711    2713    2715    2721    2732    3034    4171    4261    4407    4423    4428    4588
DSUP    014200      1997    3431#
DSO     014206      3433#
DSOA    014270      3440    3448#
DSOB    014266      3444    3447#
DSOC    014230      3434    3436    3439#
DS1     014316      3432    3454#   3641
DS2     014336      3458#   3460
DS2A    014346      3455    3461#
DS3     014352      2781    2975    3462#
DS4     014362      3464#   3466
DT      000536      1533#   3251    3256    3260    3454    4569
DVN     000550      1541#   1978    3214    3219    3249    4406    4618*   4685
DOFL    014554      3337*   3475    3478*   3507#
EMADDR  000652      1577#   2136*   2232*   2302*   2354*   2491*   2506*   2685*   3023*   3033*   3044*   4238    4243
                    4470
EOPB1   000744      1606#   1911*   1927    1929*   2197*
EOTCO   002650      1773#   2124*   2125
EOTREC  000660      1580#   2108    2110*   2214*   2310    2312*   2313*   2314*   2398    2621    2642    2644*   2645*
                    2646*   2677    2681*   2682    2719*   2720*   2791    2799
ER      000524      1528#   4164    4175    4224    4265    4312
ERCHK   017406      2318    2764    2967    4097#
ERPT    020234      2366    2736    3037    4183    4185    4187    4195    4208    4212    4214#
ERPTG   020270      4219    4223#
ERPTG1  020336      4228    4233#
ERPTT   020254      4218#   4221
ERPTO   020352      4226    4232    4234    4236#   4329
ERPT1   020420      4242    4244    4247#
ERPT2   020456      4248    4257#
ERPT3   020514      4258    4267#
ERPT4   020556      4268    4277#
ERPT5   020600      4278    4283#
ERPT5A  020662      4288    4298#
ERPT6   020724      4299    4308#
ERPT7   020776      4309    4311    4315    4321#
ERPX    021034      4323    4326    4328    4330#
ERPX0   021000      4235    4322#
ERPX1   021076      4322    4338#
ERPX2   021130      4339    4341    4344#
ERSAV   000722      1597#   2331    2379    2433    2824    2858    2890    4216*   4224*
ERTFL   000732      1601#   2482*   2500    2503*
ER0     017422      4099    4101#
ER0A    017472      4109    4113#
ER0B    017440      4103    4106#
ER1     017500      4107    4115#
ER10    020176      4204    4206#
ER2     017504      2369    2513    2762    2959    2965    3040    3051    4112    4114    4116#
ER2A    017550      4127#   4132    4142
ER2A0   017520      4119#   4121
ER2A1   017540      4123    4125#
ER2B    017564      4126    4131#
ER2C    017610      4137#   4146    4148
ER2D    017624      4134    4141#
ER2E    017652      4144    4147#
ER3     017660      4128    4130    4138    4140    4149#
ER3A    017722      4154    4160#
```

```
ER3A1   017760  4166  4168#
ER3B    017766  4156  4159  4163  4170#
ER4     017772  4150  4161  4169  4171#
ER4A    020036  4174  4181#
ER4A1   020030  4177  4179#
ER6     020042  4172  4180  4182#
ER6A    020126  4189  4194#
ER7     020160  4199  4201#
EXCRC   015536  3681* 3691  3731# 4190  4192* 4198  4296  4342*
EXLRC   015540  3693* 3732# 4191  4193* 4207  4306  4343*
FC      000516  1525# 2303* 2356* 2492* 2507* 2693* 2910* 2938* 3027* 3045* 4127  4137  4281
                4446*
FCER    021142  4129* 4139* 4277  4349#
FCSAV   000632  1566# 2086  3332*
FMCNT   000556  1544# 2086* 2303  2693  2697  2751  2928  2938  2948  3095* 3323* 3324  3326
                3331* 3332  3633  3654  3682  3752  3774  3778  3786  3928  3932  4097  4135
                4434  4616* 4653* 4755
FREX    021214  4366  4371  4374#
FRPRT   021156  2892  3876  4240  4365#  4419  4472
FRO     021206  4368  4372#
HDRFL   000664  1582# 3827* 3871  4759*
HERE    005126  2185  2191#
HRDS    022162  4531  4565#
HRDS0   022222  4572# 4576  4597
HRDS1   022230  4571  4574#
HRDS2   022276  4580  4582  4584  4587#  4595
HRDS3   022320  4589  4592#
HRDS4   022332  4593  4596#
INTRF   000566  1548# 3354  3356  4447  4621*
LRCER   021144  4210* 4213* 4298  4350#
LRCSV   021152  4191* 4343  4353#
MCRLF   027461  5018  5406#
MR      000534  1532# 4201  4451* 4504*
MSG1    024736  3874  5069#
MSG10   025017  3023  3044  5087#
MSG100  026652  2182  2195  5327#
MSG101  026673  4536  5330#
MSG102  026707  4538  5333#
MSG103  026716  4542  5335#
MSG104  026727  4523  5337#
MSG105  026744  2425  2844  5340#
MSG106  026757  2118  5342#
MSG107  027047  2578  5352#
MSG108  027132  4516  5362#
MSG109  027146  2114  5364#
MSG11   025024  4687  5089#
MSG110  027200  2444  5369#
MSG111  027222  2450  5372#
MSG112  027234  2568  5374#
MSG113  027255  2027  2047  5378#
MSG114  027266  2052  5381#
MSG115  027277  2880  5384#
MSG116  027322  4426  5388#
MSG12   025040  4683  5092#
MSG120  027373  1881  5395#
MSG13   025054  2531  4728  5095#
```

```
MSG14   025062            2536    4732    5097#
MSG15   025067            3911    5099#
MSG16   025120            4372    5105#
MSG16A  027004            2127    5346#
MSG17   025123            4369    5107#
MSG2    024743            3887    5071#
MSG20   025126            2121    5109#
MSG21   025141            3352    5113#
MSG22   025166            2724    5118#
MSG23   025213            4249    5123#
MSG23A  025221            4283    5126#
MSG23B  025226            4269    5128#
MSG23C  025233            4279    5130#
MSG23D  025240            4253    5132#
MSG23E  025246            4259    5134#
MSG23F  025253            4263    5136#
MSG23G  025260            5138#
MSG23H  025265            4316    5140#
MSG23I  025272            5142#
MSG23J  025277            5144#
MSG23K  025304            5146#
MSG23L  025311            5148#
MSG23M  025316            5150#
MSG24   025323            4473    5152#
MSG25   025343            4420    5156#
MSG26   025363            4048    5160#
MSG27   025374            4063    5163#
MSG28   025405            2545    2560    5166#
MSG3    024750            3895    5073#
MSG30   025407            3175    5167#
MSG31   025461            3172    5175#
MSG32   025523            3229    5182#
MSG33   025544            3266    5186#
MSG34   025560            3278    5189#
MSG35   025573            3311    5192#
MSG36   025614            3321    5196#
MSG37   025640            3333    5201#
MSG38   025663            3361    5205#
MSG4    024755            3877    5075#
MSG40   025703            3372    5208#
MSG41   025731            3381    5213#
MSG42   025743            3390    5216#
MSG43   025763            4826    5027    5220#
MSG44   025767            2980    5222#
MSG45   026016            5227#
MSG46   026030            5230#
MSG47   026035            5232#
MSG48   026042            2161    2255    5234#
MSG49   026074            1985    2075    5240#
MSG5    024762            2302    4414    5077#
MSG50   026110            3258    5242#
MSG52   026135            3212    5246#
MSG53   026157            3290    5251#
MSG54   026172            2354    4243    5254#
MSG55   026201            3033    5257#
MSG56   026210            4245    5260#
```

```
MSG57   026214          3253    5262#
MSG58   026235          4289    5265#
MSG59   026243          4300    5267#
MSG6    024767          2136    2232    2685    4417    5079#
MSG60   026251          2138    4692    5269#
MSG61   026255          4698    5270#
MSG62   026261          4705    5271#
MSG64   026265          2343    2393    2834    5273#
MSG65   026307          2012    2427    2446    2846    2869    5278#
MSG66   026320          2491    5281#
MSG67   026333          2506    5284#
MSG68   026343          2042    5287#
MSG69   026354          3343    5289#
MSG7    024774          4753    5081#
MSG70   026372          2234    5293#
MSG71   026426          3223    5298#
MSG72   026447          2022    5301#
MSG73   026460          2017    5303#
MSG74   026471          3179    5305#
MSG75   026514          3188    5309#
MSG76   026527          2057    5311#
MSG77   026540          2037    5313#
MSG78   026551          2335    2383    2437    5315#
MSG79   026612          2828    2862    5321#
MSG8    025001          4715    5083#
MSG9    025011          3262    5085#
MTC1    000672          1585#   2305*   2355*   2508*   2704*   2706*   2913*   2915*   2935*   2937*   3028*   3046*   4106
                        4125    4131    4186    4338    4365    4367    4412    4430    4449    4452    4457*   4459    4735
MTINT   021714          1505    3198    4503#
MTINTA  021716          4478    4504#
NOP   = 000240          1461#
NRTP    011260          2337    2385    2439    2830    2864    2890#
NRZOF   000650          1576#   4518    4627    4640
OCTP    023732          4879#   5056
OCTPE   023740          4881#   5006
OCTPE1  023746          4880    4882#
OCTPG   024122          4891    4894    4901    4905    4910    4915    4917    4921#
OCTPG0  024140          4922    4924    4926#
OCTPG1  024144          4886    4927#
OCTPP = 104002          2016    2021    2026    2031    2036    2041    2046    2051    2056    2061    2126    2430    2449
                        2453    2530    2535    2540    2567    2649    2872    2891    2983    3182    3191    3261    3314
                        3325    3336    3346    3355    3364    3375    3384    3393    3886    4057    4066    4252    4256
                        4262    4266    4272    4276    4282    4286    4293    4297    4303    4307    4320    4541    4548
                        4686    4691    4697    4704    4714    4727    4731    4748    4752    4758    5061#
OCTP0   023766          4883    4885    4888#
OCTP1   024010          4889    4893#
OCTP2   024016          4892    4895#
OCTP3   024106          4887    4918#
OFL     024164          4879*   4881*   4884    4923    4926*   4932#
OUT     024666          4994    4996    5000    5022    5039#
PAPRT   022736          1984    2008    2074    2111    2137    2160    2233    2254    2334    2382    2436    2577    2723
                        2827    2861    3873    4013    4021    4237    4411    4425    4469    4683#
PAPRTA  023112          4719#   4725
PAPRTB  023126          4718    4722#
PAPRTC  023142          4723    4726#
PAPRTD  023150          4721    4728#
```

```
PAPRTY  023244          4743    4746#
PAPRT0  023040          4702    4704#
PAPRT1  023214          4736    4739#
PAPRT2  023246          4741    4747#
PAPRT3  023250          4738    4745    4748#
PARCNT  015542          3702*   3706*   3713    3733#
PARS    014410          3443    3445*   3467*   3468*   3471#
PATRN   000560          1545#   3335    3338    3435    3439    3449    4619*   4623*   4625*   4632*   4655*   4657*   4659*
                        4662*   4724    4726
PATS    014406          1910*   2178*   3096*   3439    3450*   3470#   3772*
PFLG    000670          1584#   3870*   3955    3960*   4236*   4325    4330*
PICK    017204          3996    4040#
PIK1    000770          1622#   3988    4025    4041    4052
PIK2    000772          1623#
PIK3    000774          1624#
PIK4    000776          1625#
PIK5    001000          1626#
PIK6    001002          1627#
PIK7    001004          1628#
PIK8    001006          1629#
PRB     000622          1562#   3479*   3486
PRS     000620          1561#   3480*   3482*   3483
PSW     000604          1555#   1914*   4460*   4466*   4483*   5049*
RANBAS  000624          1563#   2083*   4630*   4771*   4772
RANG    023322          3093    3111    3637    4771#   4774    4776
RANSAV  000626          1564#   2084*   3094*   3095    3112    3638    4631*   4771    4772*   4773    4775
RANSET  004372          1904    2083#   2177
RCNT    000554          1543#   2085*   2301    2471    2644    2676    3041    3112*   3313    3315    3320    4617*   4654*
                        4739    4751
RCNTR   012354          1974    3109#
RCSAV   000630          1565#   2085    3320*
RDA     007772          2678    2680    2685#
RDATA   033472          2493    2694    2734    2744    2925    2947    2961    3024    3036    3039    3050    3463    3501
                        3758    3880    3925    3938    4110    4113    5412#
RDCMD   000562          1546#   2598*   2607*   2611    2619*   2626*   2632*   2636    2640*   2679    2687    2695    2709
                        2738    2745    2768    2787    2797    2851    2883    2911    2926    2945    3776    3824    3833
                        3882    3889    3897    3903    3922    3943    3990    4108    4143    4211    4737
RDERR1  001150          1696#   2045    2740*   2772*
RDER1   001110          1674#   2025    2742*   2770*
RDER2   001112          1675#
RDER3   001114          1676#
RDER4   001116          1677#
RDER5   001120          1678#
RDER6   001122          1679#
RDER7   001124          1680#
RDER8   001126          1681#
RDEX    010700          2798    2800    2802    2804    2808#
RDFL    015150          1499*   1889*   2179*   3431    3642*   3644#
RDRTG   010774          2837#   2879
RDRTX   011256          2886    2888#
RDRTY   010706          2773    2821#
RDRT0   010720          2822    2824#
RDRT1   010752          2826    2832#
RDRT1A  010750          2831#
RDRT1B  010770          2833    2836#
RDRT2   011052          2843    2850#
```

```
RDRT3   011074          2852    2855#
RDRT4   011100          2854    2856#
RDRT5   011102          2841    2857#
RDRT5A  011146          2860    2867#
RDRT5B  011156          2866    2869#
RDRT6   011206          2868    2874    2877#
RDRT7   011252          2884    2887#
RDSW    024406          4992#   5017*
RDX     010704          2809#
RD0     010030          2688    2690    2693#   2796    2807
RD1     010076          2700    2703#
RD1A    010112          2696    2706#
RD1B    010120          2705    2707#
RD1D    010126          2708#
RD10    010624          2790    2792    2794#
RD11    010634          2795    2797#
RD2     010132          2707    2709#
RD2A    010152          2713#   2714
RD2B    010162          2712    2715#
RD3     010212          2710    2716    2718    2721#
RD4     010242          2722    2728#
RD4A    010340          2733    2744#
RD4A0   010372          2746    2751#
RD4A1   010414          2754    2757#
RD4A2   010436          2748    2750    2760    2762#
RD4B    010444          2731    2764#
RD4C    010450          2763    2765#
RD4D    010500          2769    2772#
RD4E    010504          2771    2773#
RD5     010514          2729    2766    2775#
RD6     010536          2741    2743    2776    2778    2780#
RD7     010562          2783    2785#
RD7A    010612          2788    2791#
READ    007732          2608    2620    2633    2641    2676#
REGS    000544          1539#   3181    3183    3200
REOT    004424          2101#   2404    2499    2523    2623    2647
REOTC   005156          1944    1946*   2141*   2147*   2148    2149*   2150*   2199#   2215*   2216    2217*   2218*   3166*
                        3302*   3308    3309*   3310*   4565*   4598*   4600*
REOTX   005054          2173    2177#
REOTXX  005140          2181    2195#
REOT1A  004472          2107    2110#
REOT1B  004520          2113    2116#
REOT1C  004540          2117    2121#
REOT1D  004544          2122#
REOT1E  004564          2120    2127#
REOT1F  004534          2115    2119#
REOT2   004606          2132#   2135
REOT2A  004644          2133    2141#
REOT3   004672          2142    2147#
REOT4   004722          2153#   2169
REOT5   004736          2155#   2156
REOT6   005002          2158    2165#
REOT7   005032          2170#
RETRY   000602          1554#   2878    4231
RETURN  021710          4489    4491    4493    4498#
RFHARD  002730          1807#   2035    2885*
```

```
RFSOFT  002670          1784#  2030   2853*
RPCNT   000700          1588#  2346*  2387*  2420*  2421   2452   2504*
RRHARD  002750          1818#  2055   2887*
RRSOFT  002710          1795#  2050   2655*
RSEQ    007414          2002   2598#
RSEX    007554          2610   2621#  2635
RSF     007470          2606   2609#
RSFR    007570          2604   2626#
RSFRX   007730          2643   2648#
RSFR0   007622          2630   2632#
RSFR1   007634          2628   2634#
RSFR2   007666          2637   2640#
RSF0    007526          2612   2616#
RSF1    007542          2615   2617   2619#
RSR     007446          2602   2605#
RSTAL   000572          1550#  2785   3374   3376
RTCNT   000702          1589#  2345*  2390*  2429   2448   2458   2477*  2478   2836*  2848   2871   2877*  2878
                        4229
RTRN    000662          1581#  2306*  2358*  2510*  2707*  2916*  2939*  3029*  3047*  4505   4742
RTYFL   000712          1593#  2321   2348*  2370   2396*  2411*  2520*  2774*  2838*  2972   4227   4744
RTY1    001050          1652#  1905   2015   2340*  2389*  2481*
RTY2    001052          1653#
RTY3    001054          1654#
RTY4    001056          1655#
RTY5    001060          1656#
RTY6    001062          1657#
RTY7    001064          1658#
RTY8    001066          1659#
RWND    005160          1998   2209#
RWNDA   005172          2210   2212#  4614   4651
RWNDX   005472          2243   2259#
RWND0   005230          2219#  2239
RWND1   005302          2228#  2231
RWND1A  005340          2223   2229   2237#
RWND2   005356          2221   2240#
RWND3   005362          2241#  2252
RWND4   005412          2246#  2247
RWND5   005432          2250#  2258
RWND6   005450          2249   2253#
SCVFL   014150          1516*  3177   3370   3399*  3401#
SERFL   000706          1591#  2324   2328*  2373   2418   2488*  2495   2505*  2514   2765   2780*  2837*  2840
                        3830   4215*  4223*
SN      000540          1534#  3264
SNPG    024356          4966   4969   4975   4977   4979#
SNPT    024300          3265   4960#
SPFLG   000570          1549#  2180   3363   3365
STAL    000666          1583#  1982*  2000*  2003*  2072*  2134*  2230*  2319*  2489*  2613*  2638*  2785*  2908*
                        2917*  2921*  2923*  3021*  3048*  3052*  3075*  4456*  4464*
STALL   012304          2001   2004   2320   2490   2614   2639   2786   2909   2924   3022   3053   3075#  3076
START   003026          1493   1517   1864#
STARTA  003160          1500   1897#  2727
STARTB  003164          1890   1898#
STARTC  003152          1496   1893#
STARTD  003250          1894   1913#
STARTE  003242          1912#  2198
STARTF  003222          1907#  1909
```

```
STARTO  003174          1900#  1902
START1  003454          1947#  2079
START2  003572          1966   1967   1969#
START3  003606          1970   1972#
START4  003622          1952   1973   1975#  1983   1988
START5  004312          2006   2066#
START6  004330          2070#  2073
START7  004360          1977   1996   2071   2078#  2146
START8  004366          2069   2079#
STAR1A  003466          1949#
STAR1B  003506          1950   1953#
STAR1C  003542          1955   1957   1962#
STAR4A  003702          1981   1989#
STAR40  003634          1976   1978#
STAUT   003136          1512   1880   1883   1888#
STAUTO  003414          1938#  4622   4624   4626   4633   4656   4658   4660   4663
STOP    024670          5042#  5058
STOPP = 104006          1987   2077   2140   2163   2194   2236   2257   2726   3954   4324   4422   4477   5063#
STP     004030          2009   2011#  2130
STPX    004276          2010   2063#
SUSWR   003262          1885   1915#
SWR     000606          1556#  1874*  1875*  1918   1922*  1925   1965   1969   1972   2005   2066   2209   2298
                        2316   2329   2341   2360   2377   2391   2400   2423   2442   2599   2603   2605   2609
                        2616   2627   2629   2634   2728   2775   2782   2821   2832   2842   2867   2941   2968
                        2994   2998   3031   3868   3909   3952   3957   4011   4019   4184   4225   4322   4327
                        4467   4475   4490   4717   4993   5005   5036*
SWREG   000176          1488#  1874   1922   1925   2994   4490   4993
TABLE   024724          5053   5055#
TAPG    021216          2307   2359   2511   2708   2918   2940   3030   3049   4405#
TAPGO   021230          4407#  4410
TAPG1   021272          4413   4417#
TAPG2   021304          4416   4420#
TAPG3   021314          4408   4423#
TAPG3A  021364          4431   4434#
TAPG3B  021400          4433   4436   4438#
TAPG3C  021420          4441   4444#
TAPG3D  021464          4448   4450   4452#
TAPG3E  021510          4455   4457#
TAPG3F  021346          4424   4430#
TAPG4   021536          4462#  4463   4465
TAPG5   021552          4466#
TAPG6   021614          4468   4475#
TAPG7   021624          4476   4478#
TBELA   023712          4870#  4873
TBELL   023706          4851   4869#
TCRLF   023622          4849   4854#
TCRLFA  023640          4857#  4860
TEMPST  024402          4990#  5009*  5030*  5031*  5032*  5034*  5038
TEMP1   000642          1573#  3237   3273   3285   3297   3481*  3487   3491*  3932*  3933*  3934*  3935   3937
                        3982*  3985*  3993*  3998   4043   4405*  4409*  4461*  4462*  4566*  4590*  4596   4598
                        4599*  4600   4796*  4801   4810*
TEMP2   000644          1574#  3231*  3232   3268*  3280*  3292*  3406*  3410   3492*  3499   3983*  3986*  3994*
                        3999   4044
TEMP3   000646          1575#  2432*  2440*  2454   2456*  2857*  2865*  2873   2875*  2984*  2990   3984*  4006
                        4045*  4951   4954
TEND    005100          2175   2182#  4558
```

TEX	023704	4847	4868#											
TIB	000640	1572#	4799	4804	4807	4814*	4815	4836*	4837*	4840	4997*	4998*	4999	5012
		5015	5023	5025	5033*	5034								
TINER	023500	4806	4809	4820	4823	4826#								
TINF	000634	1570#	1888*	1893*	1897*	2171*	3162							
TINP	012400	1936	3162#											
TINPA	012410	3163	3165#											
TINPB	012430	3169#	3171											
TINPBO	012614	3203#	3206											
TINPB1	012454	3174	3176#											
TINPC	012642	3178	3208	3211#	3228									
TINPX	014142	3371	3399#											
TINPO	012746	3222	3229#	3241	3255	3306								
TINPOA	013022	3240	3242#											
TINPOB	013040	3238	3245#											
TINPOC	013100	3251#												
TINPOD	013122	3252	3256#											
TINPOE	013146	3257	3262#											
TINP1	013164	3266#												
TINP2	013242	3274	3278#											
TINP2A	013320	3286	3290#											
TINP2B	013376	3298	3302#											
TINP2C	013424	3244	3304	3307#										
TINP3	013444	3311#												
TINP4	013774	3370#	4496											
TKB	000612	1558#	2978	2992	2996*	4484	4836	4997						
TKS	000610	1557#	1947*	2907*	2976	3001*	4833*	4834	4995					
TMEX	000564	1547#	2351	2689	2801	3025	3345	3347	4620*					
TMFLG	000676	1587#	2353*	2397*	2413	2686*	2691*	2717	2730	2777	2789*	2803	2805*	2808*
		2919	2943	2970	4133	4155	4162	4173	4188	4241	4340	4740		
TOB	000636	1571#	1898	4273*	4294*	4304*	4719*	4749*	4845*	4846	4848	4850	4854*	4857*
		4861*	4867	4870*	4918*	4928*	4936*	4938*	4941	4942	4947*	4979*	4981*	
TOG	023670	4274	4295	4305	4720	4750	4852	4855	4858	4862	4865#	4866	4871	4919
		4929	4982											
TPB	000616	1560#	4840*	4867*	4944*	4946*								
TPOS	014152	3277	3289	3301	3405#	3408								
TPOS1	014164	3247	3409#											
TPS	000614	1559#	4838	4865	4939									
TRAP30	024676	1465	5049#											
TSTAL	000576	1552#	2000	2003	2489	2613	2638	3021	3052	3392	3394			
TTIN	023514	4833#	5059											
TTINN =	104010	4798	5011	5064#										
TTINT	021630	1481	4483#											
TTINTO	021700	4487	4495#											
TTIN1	023520	4834#	4835											
TTIN2	023542	4838#	4839											
TTOUT	023560	4845#	4853	4863	4874	5055								
TTOUTT=	104000	1882	1986	2013	2018	2023	2028	2033	2038	2043	2048	2053	2058	2076
		2119	2122	2128	2139	2162	2183	2196	2235	2256	2336	2344	2384	2394
		2426	2428	2438	2445	2447	2451	2533	2537	2546	2561	2569	2579	2725
		2829	2835	2845	2847	2863	2870	2881	2981	3176	3180	3189	3213	3224
		3230	3254	3259	3263	3267	3279	3291	3312	3322	3334	3344	3353	3362
		3373	3382	3391	3875	3878	3888	3896	3912	4049	4064	4239	4246	4250
		4254	4260	4264	4270	4280	4284	4290	4301	4317	4370	4373	4415	4418
		4421	4427	4471	4474	4517	4524	4537	4539	4543	4684	4688	4693	4699
		4706	4716	4729	4733	4754	4827	5002	5004	5008	5019	5028	5060#	

```
TTR     023354  2989  3187  3196  3218  3236  3272  3284  3296  3319  3330  3342  3351  3360
                3369  3380  3389  3398  4522  4529  4796#
TTR0    023362  4798# 4817
TTR1    023404  4800  4804#
TTR1A   023416  4805  4807#
TTR1B   023430  4808  4810#
TTR2    023460  4803  4818#
TTR3    023466  4819  4821#
TTR4    023474  4822  4824#
TTR5    023476  4802  4825#
UDES    000552  1542# 1951* 1964* 1979  1989  2101  2153* 2154  2224* 2225  2244* 2245  2261*
                2262  2699  2747  2753  2759  2930  2950  2956  2962  3275* 3287* 3299* 3441
                3467  3697  3715  3718  3753  3759  3761  3763  3780  3789  4102  4141  4165
                4176  4182  4194  4310  4337  4440  4689  4694  4701  4707
UNP     000674  1586# 1948  1959* 1962* 2014  2019  2024  2029  2034  2039  2044  2049  2054
                2059  2078* 2123  2143  2151* 2152  2164  2166* 2167  2170* 2212  2213* 2219
                2238* 2240* 2241  2250* 2259* 2260  2326  2339  2375  2388  2460  2462  2480
                2562  2737  2767  2850  2882  3165* 3239  3242  3245  3303  3305* 3307* 3409
                3832  3987  4023  4040  4050
UNX     000766  1618#
UN1     000746  1610# 1939  1941* 1949  1951  1964  1975  2144* 2153  2165* 2168  2220  2222
                2224  2237* 2242  2244  2261  3168  3243* 3246* 3250  3410* 4544  4578  4607
                4643
UN2     000750  1611#
UN3     000752  1612#
UN4     000754  1613#
UN5     000756  1614#
UN6     000760  1615#
UN7     000762  1616#
UN8     000764  1617#
UPS     000714  1594#  2212* 2259
VECT    000546  1540#  3190  3192  3197
WC      000512  1523#  4285  4444*
WDATA   027464  2304  2357  2364  2368  2509  2512  3448  3457  3498  3634  3656  3684  3757
                3775  3924  3936  4115  5409#
WEX     006320  2300  2352  2361  2374  2378  2386  2396#
WRITE   005552  1999  2298#
WRTE    005536  2299  2301#
WRTSB   006764  2412  2488#
WRTSB0  007036  2497#  2516
WRTSB1  007052  2496  2500#  2515
WRTSB2  007062  2501  2503#
WRTSB3  007136  2510  2512#
WRTY    006362  2347  2395  2411#
WRTYR   006414  2416  2418#
WRTYTM  006410  2414  2417#
WRTY0   006370  2412#  2422  2483
WRTY1   006470  2424  2431#
WRTY2   006472  2419  2432#
WRTY2A  006536  2435  2442#
WRTY2B  006524  2441  2446#
WRTY3   006620  2443  2455  2458#
WRTY3A  006636  2439  2462#
WRTY4   006724  2475  2477#
WRTY5   006760  2479  2484#
WRW     006336  2400#
```

```
WRWX    006360          2399    2403    2405#
WSTAL   000574          1551#   2319    3383    3385
WTER1   001070          1663#   2020    2327*   2376*   2461*
WTER2   001072          1664#
WTER3   001074          1665#
WTER4   001076          1666#
WTER5   001100          1667#
WTER6   001102          1668#
WTER7   001104          1669#
WTER8   001106          1670#
WTM     006052          2354#   2417
WTM0    006114          2358    2360#
WTM1    006156          2363    2368#
WTM2    006166          2367    2370#
WTM3    006176          2371    2373#
WTM4    006256          2381    2387#
WTM4A   006314          2392    2395#
W0      005542          2302#   2350    2415
W1      005604          2306    2308#
W2      005644          2309    2311    2316#
W3      005660          2317    2319#
W3A     005702          2322    2324#
W4      005766          2333    2339#
W4A     006014          2342    2345#
W5      006030          2325    2330    2338    2348#
W6      006040          2351#
XOR     015510          3660    3688    3692    3723#
XORS    015534          3657*   3672*   3676    3677*   3678*   3680*   3681    3685*   3693    3724    3725*   3726*   3727
                        3730#
YOZ     011274          2784    2839    2907#   2991
YOZA    011342          2912    2915#
YOZB    011350          2914    2916#
YOZC    011370          2916    2919#
YOZC0   011460          2931    2934#
YOZC1   011474          2927    2937#
YOZC2   011502          2936    2938#
YOZD    011522          2939    2941#
YOZD0   011624          2946    2961#
YOZD1   011650          2944    2967#
YOZD2   011616          2957    2959#
YOZD3   011642          2963    2965#
YOZD4   011576          2951    2954#
YOZE    011654          2942    2968#
YOZF    011704          2960    2966    2969    2971    2975#
YOZF0   011710          2973    2976#
YOZG    012002          2977    2979    2992#
YOZH    012046          2999    3001#
YOZI    012032          2993    2995    2998#
YOZO    011314          2910#   3000
YSTAL   000600          1553#   2908    2923    2982    2985
$CNTG   027436          5001    5401#
$ENDAD  005116          1473    1869    1930    2187#
$MNEW   027451          5007    5404#
$MSWR   027442          5003    5402#
$READ   024504          5009#
$SVPC =  000034         1471#   1476
```

```
.     = 033474          1464#  1466#  1471   1472#  1474#  1476#  1480#  1487#  1492#  1495#  1498#  1504#  1510#
                        1515#  1519#  1718#  1720#  1722#  1724#  1726#  1728#  1730#  1732#  1734#  1736#  1738#
                        1740#  1742#  1744#  1746#  1748#  1755#  1757#  1759#  1761#  1763#  1765#  1767#  1769#
                        1829   5408#  5411#
```

```
$CHAIN   1358#   1866
.$ACT1   1358#   1469
.$EOP    1358#   2184

. ABS.  033474     000

 ERRORS DETECTED:  0

 CZTUAI,CZTUAI.LST/CRF/SOL/NL:TOC=CZTUAI.SML/ML,CZTUAI.P11
 RUN-TIME:  4  9  1 SECONDS
 RUN-TIME RATIO: 59/15=3.7
 CORE USED:  15K  (30 PAGES)
```