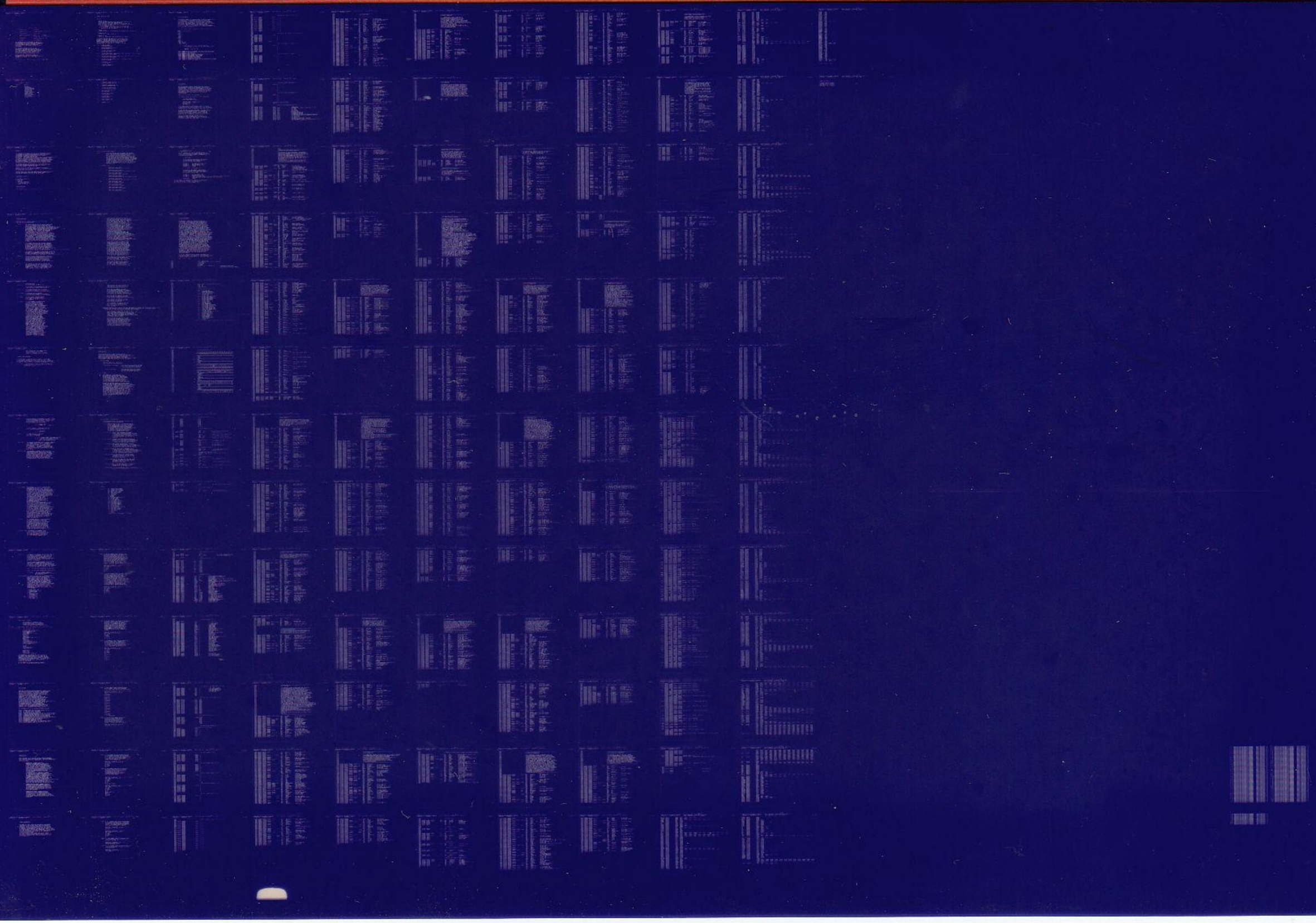
TM03,TU45 TM03/TU45 DATA RELIABAH-E497B-MC
CZTURBO FICHE 1 OF 1



.REM %

IDENTIFICATION

PRODUCT CODE:	AC-E496B-MC
PRODUCT TITLE:	CZTURB0 TM03/TU45 DATA RELIB
DATE CREATED:	25 MAY 1978
UPDATE INFORMATION:	DATE AUTHOR 29 FEB 1980 VIJAY ANANDWALA
MAINTAINER:	COMPUTER SPECIAL SYSTEMS
AUTHOR:	CSS DIAGNOSTICS

TABLE OF CONTENTS

1. ABSTRACT

THIS PROGRAM IS DESIGNED TO BE USED BY AN EXPERIENCED ENGINEER /TECHNICIAN FOR EVALUATION AND DEBUGGING OF MAG TAPE DRIVES. THE PROGRAM IS CAPABLE OF EXERCISING THE TU45 MAGNETIC ON A MASSBUS THROUGH THE TM03 MAG TAPE CONTROLLER. ANY COMBINATION OF TM03'S & TU45'S UP TO A MAXIMUM OF EIGHT (8), MAY BE TESTED BY A SINGLE EXECUTION OF THE PROGRAM. THIS FLEXIBILITY IS POSSIBLE BECAUSE THE PROGRAM HAS NO FIXED PARAMETERS OR TESTING SEQUENCE. THE ENTIRE TEST PLAN, INCLUDING PARAMETERS AND OPERATING SEQUENCE, IS DETERMINED BY THE OPERATOR THROUGH RESPONSES TO TELETYPE REQUESTS AND SETTING OF CONSOLE SWITCHES.

THE PROGRAM PROVIDES FOR TESTING OF ALL TAPE DRIVE FUNCTIONS SUCH AS WRITING,READING,REWINDING,TAPE POSITIONING,EOT - BOT SENSING AND ASSUMES A GOOD RH AND TM03.

HOWEVER; THE RH AND TM03 ARE TESTED SOMEWHAT INTRINSICALLY DURING THE TEST CYCLE IN ORDER TO PROVIDE FULL INFORMATION ABOUT ANY ERROR CONDITIONS DETECTED.

DURING A TEST CYCLE, CHECKS ARE MADE FOR STATUS ERRORS,DATA ERRORS, POSITION ERRORS,WORD COUNT AND CURRENT MEMORY ADDRESS ERRORS WHEREVER APPLICABLE AS DETECTED BY THE RH OR TM03.

2. REQUIREMENTS (HARDWARE)

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

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING BINARY TAPES

4. STARTING PROCEDURE

THERE ARE FOUR (4) STARTING ADDRESSES THAT MAY BE USED;
200(8),204(8),210(8),AND 240(8):

A. 200(8): THIS ADDRESS MUST BE USED ON INITIAL START FROM LOAD AS ALL PARAMETERS ARE ENTERED FROM HERE. REQUESTS ARE PRINTED ON THE TELETYPE FOR ENTPY OF RH STARTING ADDRESS, VECTOR ADDRESS, DRIVE NUMBER(TM03 ADDRESS), SLAVE NUMBER, DENSITY,PARITY,FORMAT,RECORD COUNT,CHARACTER COUNT,PATTERN NUMBER,TAPE MARK AND STALL FOR READ,WRITE, AND TURNAROUND. ALL REPONSES SHOULD BE MADE IN OCTAL AND WITHIN THE LIMITS OF THE PARAMETER. A QUESTION MARK (?) WILL BE TYPED IF ANY CHARACTER ENTERED IS NOT BETWEEN 0 THRU 7 (OCTAL). THE CHARACTER MAY BE RETYPED FOLLOWING THE QUESTION MARK. IF THE RESPONSE IS NOT WITHIN ITS LIMITS. A QUESTION MARK (?) IS TYPED AND THE ENTIRE RESPONSE MAY BE RENTERED. SOME RESPONSES REQUIRE MORE THAN ONE (1) CHARACTER, BUT NONE REQUIRES MORE THAN SIX (6). RESPONSES OF MORE THAN ONE CHARACTER NEED NOT HAVE LEADING ZEROS AND SHOULD BE TERMINATED BY A CARRIAGE RETURN IF LESS THAN THE MAXIMUM NUMBER OF CHARACTERS IS INPUT.

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

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

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

THE FOLLOWING IS AN EXPLANATION OF THE INITIAL START (200 OCTAL) REQUESTS AND RESPONSES:

REGISTER START: THE RESPONSE REQUIRED FOR THIS REQUEST IS TO ENTER THE ADDRESS OF THE FIRST RH REGISTER (CS1) AS A SIX DIGIT UNIBUS ADDRESS.

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

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

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

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

4.1 AUTOMATIC MODE OPERATION

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

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

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

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

PARITY: THE PARITY REQUEST IS RESPONDED TO BY ONE (1) OCTAL CHARACTER AND MUST BE EITHER 0 OR 1.

A. 1 = EVEN PARITY
B. 0 = ODD PARITY

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

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

RECORD COUNT: THIS REQUEST IS RESPONDED TO BY A SIX (6) CHARACTER OCTAL NUMBER FROM 1 TO 177777. REMEMBER LEADING ZEROS ARE NOT REQUIRED AND IF LESS THAN SIX CHARACTERS ARE ENTERED, A CARRIAGE RETURN WILL TERMINATE THE RESPONSE. THE RECORD COUNT IS USED IN CONJUNCTION 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. (CZTUTA0) ANY CONFIGURATION
OF BITS AND CHARACTERS MAY BE USED AND A LIMIT
OF 377(8) CHARATERS IS IMPOSED. WHEN EXTERNAL
DATA IS INPUT, THE ENTIRE WRITE BUFFER IN
CORE IS FILLED WITH THE PATTERN SO THAT ANY
SIZE RECORD MAY BE USED. DATA PATTERN
PATTERN ZERO (0) EXTERNAL PAPER TAPE NEED ONLY
BE READ ONCE AT INITIAL START OF 200(8), AND
NEED NOT BE READ AGAIN UNLESS OVERWRITTEN BY
RANDOM DATA. BE SURE TO LOAD THE READER
BEFORE PRESSING START.

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

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

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

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

READ: THE TIME DELAY BETWEEN EACH RECORD READ

WRITE: THE TIME DELAY BETWEEN EACH RECORD WRITTEN

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

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

A. RECORD COUNT = 200
B. CHARACTER COUNT = 4000
C. PATTERN NUMBER = 1
D. TM=1
E. INTERCHANGE READ = 0
F. SINGLE PASS = 0
G. CRC CORRECTION = 0
H. READ STALL = 10
I. WRITE STALL = 10
J. TURN AROUND STALL = 10

```
SAMPLE START AT 200(8):

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

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

        TU45 TAPE DRIVE TEST

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

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

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

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

5. DATA PATTERNS

THERE ARE FIFTEEN DATA PATTERN GENERATORS STORED IN CORE
AND ANY ONE OF THESE MAY BE SELECTED. THE ONE UNIQUE
CASE IS PATTERN ZERO(0); SELECTION OF PATTERN ZERO(0)
REQUIRES THAT A PREVIOUSLY PREPARED PAPER TAPE BE
ENTERED AT THE HIGH SPEED READER. THIS TAPE CONTAINS
A DATA PATTERN OF NO MORE THAN 377 OCTAL
CHARACTERS. THE FIRST CHARACTER READ IN IS THE
NUMBER OF ACTUAL DATA CHARACTERS THAT ARE CONTAINED
ON THE TAPE. EACH DATA CHARACTER MAY BE ANY
COMBINATION OF BITS AND WILL BE LOADED INTO CORE
AS THEY APPEAR ON THE TAPE. NO MATTER HOW MANY
CHARACTERS ARE ON TAPE, THE ENTIRE WRITE BUFFER (4000 CHARACTERS)
WILL BE FILLED WITH THE PATTERN ENTERED SO THAT ANY
SIZE RECORD CAN BE USED. (SEE DTC CZTUTA0)
THE PROGRAM GENERATES A CYLIC REDUNDENCY CHECK
CHARACTER (CRC) AND A LONGITUDINAL REDUNDENCY CHECK
CHARACTER (LRC) FOR COMPARISONS AGAINST THE CRC AND LRC
GENERATED BY THE HARDWARE IN NRZI READS OR WRITES.

THE FOLLOWING IS A LIST OF THE DATA PATTERNS AVAILABLE:

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

6. RANDOMIZATION

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

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

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

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

7. DYNAMIC PARAMETERS:

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

8. CONSOLE SWITCH SETTINGS

CONTROL:

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

2) CONTROL A <^A>;
   ALTERNATES USAGE OF THE SWR BETWEEN THE HARDWARE SWR & SOFTWARE SWR.

3) CONTROL B <^B>;
   SEE SECTION 7 DYNAMIC PARAMETERS

4) CONTROL U <^U>;
   DELETES ALL CHARACTERS TYPED IN RESPONSE TO A REQUEST.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

SWITCH EXPLANATION AND EXAMPLES:

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

EXAMPLES: 0-3

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

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

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

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

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

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

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

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

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

SW4: SWITCH FOUR (4), WHEN SET TO A ONE (1), WILL CAUSE ANY DATA RELATED ERROR TO BE RETRIED. THE WRITE RETRY SCHEME CONSISTS OF REWRITING THE RECORD IN THE SAME SPOT ON TAPE FOUR (4) TIMES. IF ALL FOUR (4) REPEATS ARE SUCCESSFUL, THE RECORD IS CONSIDERED AS RECOVERED, AND A TAPE WRITE ERROR IS LOGGED. IF ANY OF THE FOUR (4) REPEATS IS UNSUCESSFUL, A SKIP ERASE IS DONE, A SUPECTED BAD TAPE SPOT IS LOGGED AT THIS BLOCK AND RECORD NUMBER, AND A SECOND RETRY OF FOUR REPEATS IS DONE. IF AFTER FOUR (4) RETRIES, THE RECORD CANNOT BE RECOVERED A NOTIFICATION IS PRINTED, AND TESTING IS RESUMED ON THE NEXT RECORD.
IF 20(8) BAD TAPE SPOTS ARE FOUND, THE SLAVE WILL BE REWOUND AND REMOVED FROM TESTING WITH AN APPROPRIATE MESSAGE PRINTED.
THE READ RETRY SCHEME CONSISTS OF REREADING THE RECORD UP TO EIGHT TIMES. IF ALL EIGHT REREADS ARE BAD, IT IS A HARD ERROR. IF ANY REREAD IS SUCCESSFUL, THIS IS A SOFT ERROR.
IF THE ORIGINAL ERROR IS OF THE NON-RETRYABLE TYPE (IE: ILF,RMR,ILR,NEF,CBUSPE), THE RETRY SCHEME IS NOT ENTERED AND A MESSAGE IS PRINTED.

SW5: SWITCH FIVE (5) WHEN SET DURING A READ FORWARD OR REVERSE WILL CAUSE THE TAPE TO CONTINUOUSLY READ THE CURRENT RECORD BY SPACING EITHER FORWARD OR REVERSE AND REREADING THAT RECORD. THIS TAPE MOVEMENT IS CALLED YOZZLING. THERE IS A SOFTWARE DELAY EXECUTED BETWEEN EACH SPACE/READ OF THE RECORD AND IT MAY BE VARIED BY TYPING CONTROL C ON THE TELETYPE DURING THE EXECUTION OF THE YOZZLE AND RESPONDING TO THE PRINTED REQUEST WITH A SIX (6) DIGIT VALUE. THE YOZZLE STALL IS PRESET TO A VALUE OF 3000 IN THE PROGRAM TO PREVENT EXCESSIVE TAPE WEAR, BUT MAY BE SET TO ANY VALUE THROUGH THE TELETYPE.

SW6-8: THESE THREE (3) SWITCHES CONTROL THE RANDOMIZATION OF DATA AND BLOCK SIZE AND MAY BE SET AND RESET AT ANY TIME. THE ACTUAL CHANGE WILL TAKE PLACE BETWEEN BLOCK CYCLES.

SW9: SWITCH NINE (9) WHEN SET WILL CAUSE ALL AVAILABLE TAPE UNITS TO BE REWOUND AT THE END OF THE CURRENT BLOCK CYCLE. TESTING WILL BE RESUMED AT A BLOCK COUNT OF ONE (1) WHEN ALL UNITS HAVE REACHED BOT.

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

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

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

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

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

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

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

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

G 2

9. ERROR PRINTOUTS

THERE ARE THREE TYPES OF ERROR PRINTOUTS MADE BY THE PROGRAM; OPERATION ERRORS, DATA ERRORS, AND CONDITION ERRORS. EACH ERROR MESSAGE PRINTED IS PROCEEDED BY A TWO LINE HEADER WHICH CONTAINS THE DRIVE NUMBER, SLAVE NUMBER, DENSITY, PARITY, AND FORMAT ON THE FIRST LINE, AND THE BLOCK NUMBER, RECORD NUMBER, RECORD SIZE, AND ERROR TYPE ON THE SECOND.

A. OPERATION ERRORS:

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

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

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

B. DATA ERRORS:

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

BECAUSE DATA RECORDS CAN BE UP TO FOUR THOUSAND CHARACTERS LONG, AN ERROR CONDITION WHICH WILL CAUSE THE ENTIRE RECORD TO READ INCORRECTLY COULD CAUSE A VERY LENGTHY PRINTOUT. THEREFORE, A COUNTER OF SUCCESSIVE BAD CHARACTERS IS EMPLOYED. IF TEN (10) CHARACTERS IN SUCCESSION ARE BAD, A NOTIFICATION IS PRINTED (BAD RECORD) AND THE NEXT TWENTY FIVE (25) CHARACTERS ARE SKIPPED BEFORE CHECKING IS RESUMED. IF THE BAD RECORD CONDITION OCCURS THREE (3) TIMES IN ONE RECORD, THE REST OF THE RECORD IS SKIPPED, DOWN TO THE LAST TEN (10) CHARACTERS WHICH WILL BE CHECKED. THE SKIPPING AND RESUMPTION OF CHECKING WILL ONLY BE DONE ON RECORDS WHICH ARE LONG ENOUGH TO ALLOW IT.

C. CONDITION ERRORS: (CATASTROPHIC)

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

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

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

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

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

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

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

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

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

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

D. EXAMPLES:

GLOSSARY:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

10. STATISTICS PRINTOUT

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

STATISTICS PRINT EXAMPLE (A HEADER WILL PRECEED THE STATS)

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

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

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

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

11. AUTO SEQUENCE

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

SAMPLE START AT 240(8): AUTO SEQUENCE.

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

```
        TU45 AUTO SEQUENCE TEST
        ENTER CONDITIONS IN OCTAL

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

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

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

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

AUTO SEQUENCE TEST PLAN:

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

1. NRZ CYCLE:

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

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

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

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

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

IN CHAIN MODE THIS SEQUENCE TEST PLAN WILL BE EXECUTED ONE TIME AND CONTROL WILL RETURN TO THE MONITOR. SPECIFY NUMBER OF PASSES DESIRED IN CHAIN COMMAND STRING.

12. TESTING PROCEDURES

AS PREVIOUSLY STATED THIS PROGRAM CONTAINS NO FIXED TESTS. THE ENTIRE TEST CYCLE TO BE EXECUTED IS DESCRIBED BY THE OPERATOR THOUGH RESPONSES TO TELETYPE REQUESTS FOR PARAMETERS AND CONSOLE SWITCH SETTINGS FOR OPERATION. THE OPERATION SELECTED WILL BE EXECUTED WITH THE PARAMETERS ENTERED CONTINUOUSLY ON EACH AVAILABLE UNIT, ONE BLOCK AT A TIME, UNTIL STOPPED BY THE OPERATOR. THE OPERATION MAY BE CHANGED DYNAMICALLY BY CHANGING THE CONSOLE SWITCHES AT ANY TIME. THE PROGRAM WILL ATTEMPT TO PERFORM ANY OPERATION SET AND THEREFORE CAUTION SHOULD BE TAKEN TO ASSURE THAT THE UNIT IS CAPABLE OF PERFORMING AS REQUESTED. FOR INSTANCE, ONE SHOULD NOT ATTEMPT TO PERFORM READ OPERATIONS ON A TAPE WHICH HAS NOT BEEN WRITTEN AS THE DATA, IF ANY, IS UNPREDICTABLE. HOWEVER, IF A TAPE HAS BEEN WRITTEN WITH THIS PROGRAM, IT CAN BE READ AS OFTEN AS DESIRED WITHOUT BEING REWRITTEN. THIS IS A GOOD PROCEDURE TO USE FOR TESTING TAPE COMPATABILITY. SCOPING OF TAPE UNITS BECOMES SIMPLE; BY SETTING THE DESIRED OPERATION AND ITS PARAMETER, A UNIT MAY BE CONTINUOUSLY EXERCISED IN ANY MANNER DESIRED. BY USING THE VARIOUS ERROR CONTROL SWITCHES AND ENTERING THE NEEDED STALL, ANY FUNCTION CAN BE SCOPED RATHER EASILY. RELIABILITY TESTING CAN BE PREFORMED BY USE OF THE RANDOMIZATION CAPABILITY. PERHAPS A CYCLE OF RANDOM TESTING MIGHT BE SET UP AND ALLOWED TO RUN FOR SOME PERIOD OF TIME, THE STATISTICAL COLLECTION OF DROPS AND PICKS IS THEN SIGNIFICANT. INTERMITTANT PROBLEMS CAN BE FOUND BY SETTING THE DESIRED OPERATION IN MOTION AND DISALLOWING ERROR PRINTOUTS WHILE ALLOWING A HALT ON ERROR. THE ERROR THAT CAUSED THE HALT CAN BE PRINTED BY RESETTING CONSOLE SWITCH TEN AND PRESSING CONTINUE. IF SOME PARTICULAR DATA PATTERN SHOULD BE CAUSING DATA ERROR, USE OF THE YOZZLE SWITCH AND ITS ASSOCIATED STALL WILL TO ALLOW SCOPING OF THIS PARTICULAR RECORD.

AS YOU SEE, THERE ARE MYRIAD TESTING PROCEDURES WHICH COULD BE PERFORMED. THE PARAMETERS, TAPE OPERATIONS, ERROR EXAMINATION AND REPORTING ARE ALL AT YOUR DISCRETION.

TRY IT, YOU'LL LIKE IT.

```
%
2632                                    .LIST   BIN,LOC,SEQ
2633                                    .TITLE TM03/TU45 DATA RELIABILITY PROGRAM
2634                                    ;++B CZTURB0
2635                                    ;21 FEB 1977
2636                                    ;R. BARNES
2637                                    ;REVISED (++B) J.G.ADAMS MAY 1977       1)INCORRECT RECORD COUNT
2638                                    ;++B                                    STORED WHEN EOT REACHED ON WRITE
2639                                    ;++B                                    2)ADJUST STACK PTR ON BAD TAPE OVFLW
```

```
2640
2641                                    .MCALL  .$ACT11,.$EOP,$SAVE,$RESTORE,$CHAIN
2642                                    .NLIST  MC
2643                                    .LIST   ME
2644                                    .ENABLE ABS,AMA
2645
2646                                    ;CONSOLE SWITCHES***********
2647
2648                                    ;SW15: 1=STOP ON ERROR
2649                                    ;       0=CONTINUE ON ERROR
2650                                    ;SW14: 1=PRINT READ/WRITE STATS
2651                                    ;       0=DO NOT PRINT STATS
2652                                    ;SW13: 1=DO NOT CHECK DATA
2653                                    ;       0=CHECK DATA
2654                                    ;SW12: 1=DO NOT CHECK WRITE ERRORS
2655                                    ;       0=CHECK WRITE ERRORS
2656                                    ;SW11: 1=DO NOT CHECK READ ERRORS
2657                                    ;       0=CHECK READ ERRORS
2658                                    ;SW10: 1=DO NOT PRINT ERRORS
2659                                    ;       0=PRINT ERRORS
2660                                    ;SW9:  1=REWIND TAPE
2661                                    ;       0=DO NOT REWIND
2662                                    ;SW8:  1=USE RANDOM DATA
2663                                    ;       0=USE FIXED DATA PATTERN
2664                                    ;SW7:  1=USE RANDOM CHARACTER COUNT
2665                                    ;       0=USE FIXED CHAR COUNT
2666                                    ;SW6:  1=USE RANDOM RECORD COUNT
2667                                    ;       0=USE FIXED RECORD COUNT
2668                                    ;SW5:  1-YOZZLE ON CURRENT RECORD
2669                                    ;       0=DO NOT YOZZLE
2670                                    ;SW4:  1=DO BOTH READ AND WRITE RETRIES
2671                                    ;       0=INHIBIT RETRIES
2672                                    ;SW3:  1=DO NOT READ FORWARD
2673                                    ;       0=READ FORWARD
2674                                    ;SW2:  1=DO NOT READ REVERSE
2675                                    ;       0=READ REVERSE
2676                                    ;SW1:  1=READ FORWARD FIRST
2677                                    ;       0=READ REVERSE FIRST
2678                                    ;SW0:  1=DO NOT WRITE
2679                                    ;       0=WRITE
2680                                ;IF SWR <15::00> = 177777 OR NOT AVAILABLE USE SOFTWARE SWITCH REGISTER
```

```
2682                            ;TU45 REGISTER BITS**********
2683
2684                            ;15 ;14 ;13 ;12 ;11 ;10 ;09 ;08 ;07 ;06 ;05 ;04 ;03 ;02 ;01 ;00 ;
2685                            ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2686                    ;CS1:   ;SC ;TRE;MCP;SPR;DNA;PSL;A17;A16;RDY;IE ;FUN;FUN;FUN;FUN;FUN;GO ;
2687                            ;  ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;
2688                            ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2689                    ;WC:    ; WORD                                                          ;
2690                            ; COUNT                                                         ;
2691                            ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2692                    ;BA:    ; BUS                                                           ;
2693                            ; ADDRESS                                                       ;
2694                            ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2695                    ;FC:    ; FRAME                                                         ;
2696                            ; COUNT                                                         ;
2697                            ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2698                    ;CS2:   ;DLT;WLE;UPE;NED;NEM;PGE;MXF;MDP;OR ;IR ;CLR;PAT;BAI;U2 ;U1 ;U0 ;
2699                            ;  ;   ;   ;   ;   ;   ;   ;ERR;  ;   ;   ;   ;   ;   ;   ;   ;
2700                            ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2701                    ;DS:    ;ATA;ERR;PIP;MOL;WRL;EOT;SPR;DPR;DRY;SSC;PES;SDN;IDB;EOF;BOT;SLA;
2702                            ;  ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;
2703                            ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2704                    ;ER:    ;CDE;UNS;OPI;DTE;NEF;CS ;FCE;NSG;PEF;INC;DAT;FMT;CNT;RMR;ILR;ILF;
2705                            ;CRC;   ;   ;   ;ITM;   ;   ;LRC;VPE;BPE;   ;BPE;   ;   ;   ;
2706                            ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2707                    ;ATS:   ; ATTENTION                                                     ;
2708                            ; SUMMARY                                                       ;
2709                            ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2710                    ;CC:    ; CHECK                                                         ;
2711                            ; CHARACTER                                                     ;
2712                            ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2713                    ;DB:    ; DATA                                                          ;
2714                            ; BUFFER                                                        ;
2715                            ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2716                    ;MR:    ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;WRT;MM ;OP ;OP ;OP ;OP ;MM ;
2717                            ; 7 ; 6 ; 5 ; 4 ; 3 ; 2 ; 1 ; 0 ; P ;CLK;CLK; 4 ; 3 ; 2 ; 1 ;GO ;
2718                            ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2719                    ;DT:    ; DRIVE     ; 7 ;   ;SLV;                                      ;
2720                            ; TYPE      ;CH ;   ;PR ;                                      ;
2721                            ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2722                    ;SN:    ; SERIAL                                                        ;
2723                            ; NUMBER                                                        ;
2724                            ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2725                    ;TC:    ;ACL;FCS;TCW;ENA;SPR;DEN;DEN;DEN;FMT;FMT;FMT;FMT;EVN;SSN;SSN;SSN;
2726                            ;   ;   ;   ;DT ;   ; 4 ; 2 ; 1 ; 8 ; 4 ; 2 ; 1 ;PAR; 4 ; 2 ; 1 ;
2727                            ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

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

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

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

           000020                       .=20
  000020   023754                       .WORD   TTOUT           ;SET IOT TRAP TO TTOUT ROUTINE
  000022   000340                       .WORD   340             ;PRIORITY LEVEL 7

           000004                       TYPE=IOT                ;EQUATE TYPE TO AN IOT INSTRUCTION
           000034                       .=34
  000034   024126                       .WORD   OCTP            ;SET TRAP TRAP TO OCTP ROUTINE
  000036   000340                       .WORD   340
           104400                       TYPOCT=TRAP             ;EQUATE TYPOCT TO TRAP INSTRUCTION

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

                                        ;TTY INTERRUPT VECTOR**********
           000060                       .=60
  000060   021556                       .WORD   TTINT           ;TTY INTERRUPT HANDLER ADDRESS
  000062   000340                       .WORD   340             ;PRIORITY LEVEL 7

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

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

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

           000210                       .=210
  000210   005037   015064              CLR     RDFL
  000214   000137   003160              JMP     STARTA          ;USE FIXED PARAMETERS; NEW DATA

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

           000224                       .-224
```

```
 2784  000224  022006                  MTINT                   ;MAG TAPE INTERRUPT HANDLER ADDRESS
 2785  000226  000340                  340
 2786
 2787                                  ;AUTO SEQUENCE START**********
 2788
 2789          000240                  .=240
 2790  000240  005237  000736          INC     AGEQF           ;SET AUTO SEQUENCE FLAG
 2791  000244  000137  003136          JMP     STAUT           ;GO TO START OF AUTO SEQUENCE
```

```
2793                                            ;SHORT CONVERSATION RESTART********
2794
2795            000300                          .=300
2796  000300    005237  014076                INC     SCVFL                   ;SET SHORT CONVERSATION FLAG
2797  000304    000137  003026                JMP     START                   ;ENTER SHORT PARAMETER LIST
2798
2799            000510                          .=510
2800                                            ;TU45 REGISTER EQUIVS**********
2801
2802  000510    172440                C1:     172440
2803  000512    172442                WC:     172442
2804  000514    172444                BA:     172444
2805  000516    172446                FC:     172446
2806  000520    172450                CS:     172450
2807  000522    172452                DS:     172452
2808  000524    172454                ER:     172454
2809  000526    172456                AS:     172456
2810  000530    172460                CC:     172460
2811  000532    172462                DB:     172462
2812  000534    172464                MR:     172464
2813  000536    172466                DT:     172466
2814  000540    172470                SN:     172470
2815  000542    172472                TC:     172472
2816
2817                                            ;CONSTANTS**********
2818
2819  000544    172440                REGS:   172440          ;STARTING REGISTER ADDRESS (CS1)
2820  000546    000224                VECT:   224             ;VECTOR ADDRESS (RH INTERRUPT)
2821  000550    000000                DVN:    0               ;DRIVE NUMBER
2822  000552    000000                UDES:   0               ;UNIT DESCRIPTION (PARITY,DENSITY,UNIT,FORMAT)
2823  000554    000200                RCNT:   200             ;RECORD COUNTER
2824  000556    174000                FMCNT:  174000          ;NUMBER OF CHAR (4 - 4000) OCTAL IN TWOS COMPLEMENT
2825  000560    000001                PATRN:  1               ;DATA PATTERN SELECTOR (0 - 15) OCTAL
2826  000562    000002                RDCMD:  2               ;READ COMMAND
2827  000564    000001                TMEX:   1               ;TAPE MARK FLAG: 1=TM 0=NO TM
2828  000566    000000                CRCC:   0               ;CRC CORRECTION FLAG (YES=1,NO=0)
2829  000570    000000                INTRF:  0               ;INTERCHANGE READ 1=YES 0=NO
2830  000572    000000                SPFLG:  0               ;SINGLE PASS 1=YES 0=NO
2831  000574    000010                RSTAL:  10              ;READ STALL
2832  000576    000010                WSTAL:  10              ;WRITE STALL
2833  000600    000010                TSTAL:  10              ;TURN AROUND STAL
2834  000602    002000                YSTAL:  2000            ;YOZZLE STAL
2835  000604    000010                RETRY:  10              ;READ RETRY NUMBER
2836  000606    177776                PSW:    177776          ;PROCESSOR STATUS
2837  000610    177570                SWR:    177570          ;CONSOLE SWITCHES
2838  000612    177560                TKS:    177560          ;TTY READ STATUS REGISTER
2839  000614    177562                TKB:    177562          ;TTY READ BUFFER
2840  000616    177564                TPS:    177564          ;TTY PUNCH STATUS REGISTER
2841  000620    177566                TPB:    177566          ;TTY PUNCH OUTPUT REGISTER
2842  000622    177550                PRS:    177550          ;H/S READER STATUS REGISTER
2843  000624    177552                PRB:    177552          ;H/S READER BUFFER
2844  000626    153624                RANBAS: 153624          ;RANDOM NUMBER GENERATOR BASE
2845  000630    032561                RANSAV: 032561          ;RANDOM NUMBER BUFFER
2846  000632    000200                RCSAV:  200             ;RECORD COUNT SAVE
2847  000634    174000                FCSAV:  174000          ;FRAME COUNT SAVE
```

```
2849
2850                                    ;FLAGS AND COUNTERS**********
2851
2852  000636  000000    TINF:   0       ;TTY ENTRY FLAG
2853  000640            STFLG:
2854  000640  000000    TOB:    0       ;TTY OUTPUT BUFFER
2855  000642  000000    TIB:    0       ;TTY INPUT BUFFER
2856  000644  000000    TEMP1:  0       ;TEMP STORAGE
2857  000646  000000    TEMP2:  0       ;TEMP STORAGE
2858  000650  000000    TEMP3:  0       ;TEMP STORAGE
2859  000652  000000    NRZOF:  0       ;NRZ ONLY FLAG
2860  000654  000000    EMADDR: 0       ;ERROR MSG ADDRESS STORAGE
2861  000656  000000    BLCNTR: 0       ;BLOCK COUNTER
2862  000660  000000    BBC:    0       ;BAD RECORD COUNTER
2863  000662  000000    EOTREC: 0       ;EOT FLAG
2864  000664  000000    RTRN:   0       ;INTERRUUPT RETURN STORAGE
2865  000666  000000    HDRFL:  0       ;HEADER FLAG
2866  000670  000000    STAL:   0       ;DELAY STORAGE
2867  000672  000000    PFLG:   0       ;PRINT FLAG
2868  000674  000000    MTC1:   0       ;MAG TAPE CONT REGISTER BUFFER
2869  000676  000000    UNP:    0       ;UNIT TABLE POINTER
2870  000700  000000    TMFLG:  0       ;TAPE MARK FLAG
2871  000702  000000    RPCNT:  0       ;REPEAT COUNTER
2872  000704  000000    RTCNT:  0       ;RETRY COUNTER
2873  000706  000000    DERFL:  0       ;DATA ERROR FLAG
2874  000710  000000    SERFL:  0       ;STATUS ERROR FLAG
2875  000712  000000    BCNT:   0       ;BIT COUNTER
2876  000714  000000    RTYFL:  0       ;RETRY FLAG
2877  000716  000000    UPS:    0       ;UNIT POINTER SAVE
2878  000720  000000    BDPP:   0       ;BITS DROPPED POINTER
2879  000722  000000    BPKP:   0       ;BITS PICKED POINTER
2880  000724  000000    ERSAV:  0       ;ERROR SAVE LOC
2881  000726  000000    BTFLG:  0       ;BAD TAPE FLAG
2882  000730  000000    BTSTF:  0       ;STATISTIC PRINT FLAG
2883  000732  000000    BTPT:   0       ;BAD TAPE POINTER
2884  000734  000000    ERTFL:  0       ;ERASE FLAG
2885  000736            ENDFLG:
2886  000736  000000    ASEQF:  0       ;AUTO SEQ FLAG
2887  000740  000000    ADRVN:  0       ;UTO SEQ DRIVE NUMBER
2888  000742  000000    ABLCNT: 0       ;AUTO BLOCK COUNTER
2889  000744  000000    ASEQCF: 0       ;AUTO SEQ CONTINUOUS FLAG
```

```
2891
2892                                        ;UNIT ORDER AND DESCRIPTION TABLE **********
2893
2894  000746  000000                UN1:    0               ;THIS TABLE IS LOADED
2895  000750  000000                UN2:    0               ;WITH UNIT NUMBERS AND
2896  000752  000000                UN3:    0               ;THEIR DESCRIPTIONS IN
2897  000754  000000                UN4:    0               ;THE ORDER THAT THEY
2898  000756  000000                UN5:    0               ;WILL BE TESTED
2899  000760  000000                UN6:    0
2900  000762  000000                UN7:    0
2901  000764  000000                UN8:    0
2902  000766  177777                UNX:    -1
2903
2904                                        ;UNIT DROPS AND PICKS POINTERS*********
2905
2906  000770  001210                PIK1:   BP00
2907  000772  001230                PIK2:   BP10
2908  000774  001250                PIK3:   BP20
2909  000776  001270                PIK4:   BP30
2910  001000  001310                PIK5:   BP40
2911  001002  001330                PIK6:   BP50
2912  001004  001350                PIK7:   BP60
2913  001006  001370                PIK8:   BP70
2914  001010  001410                DRP1:   BD00
2915  001012  001430                DRP2:   BD10
2916  001014  001450                DRP3:   BD20
2917  001016  001470                DRP4:   BD30
2918  001020  001510                DRP5:   BD40
2919  001022  001530                DRP6:   BD50
2920  001024  001550                DRP7:   BD60
2921  001026  001570                DRP8:   BD70
2922
2923                                        ;UNIT BAD TAPE POINTERS**********
2924
2925  001030  001610                BTADDR: BT00
2926  001032  001714                        BT01
2927  001034  002020                        BT02
2928  001036  002124                        BT03
2929  001040  002230                        BT04
2930  001042  002334                        BT05
2931  001044  002440                        BT06
2932  001046  002544                        BT07
2933
2934                                        ;UNIT WRITE RETRY COUNTER**********
2935
2936                                ;SET START OF STATISTICS TABLE
2937  001050                        STTBL:
2938  001050  000000                RTY1:   0
2939  001052  000000                RTY2:   0
2940  001054  000000                RTY3:   0
2941  001056  000000                RTY4:   0
2942  001060  000000                RTY5:   0
2943  001062  000000                RTY6:   0
2944  001064  000000                RTY7:   0
2945  001066  000000                RTY8:   0
2946
```

```
 2947                                           ;UNIT WRITE ERRORS**********
 2948
 2949  001070  000000                 WTER1:  0
 2950  001072  000000                 WTER2:  0
 2951  001074  000000                 WTER3:  0
 2952  001076  000000                 WTER4:  0
 2953  001100  000000                 WTER5:  0
 2954  001102  000000                 WTER6:  0
 2955  001104  000000                 WTER7:  0
 2956  001106  000000                 WTER8:  0
 2957
 2958                                           ;UNIT READ FORWARD ERRORS**********
 2959
 2960  001110  000000                 RDER1:  0
 2961  001112  000000                 RDER2:  0
 2962  001114  000000                 RDER3:  0
 2963  001116  000000                 RDER4:  0
 2964  001120  000000                 RDER5:  0
 2965  001122  000000                 RDER6:  0
 2966  001124  000000                 RDER7:  0
 2967  001126  000000                 RDER8:  0
 2968
 2969                                           ;UNIT DATA ERRORS FORWARD**********
 2970
 2971  001130  000000                 DATER1: 0
 2972  001132  000000                         0
 2973  001134  000000                         0
 2974  001136  000000                         0
 2975  001140  000000                         0
 2976  001142  000000                         0
 2977  001144  000000                         0
 2978  001146  000000                         0
 2979
 2980                                           ;UNIT READ REVERSE ERRORS**********
 2981
 2982  001150  000000                 RDERR1: 0
 2983  001152  000000                         0
 2984  001154  000000                         0
 2985  001156  000000                         0
 2986  001160  000000                         0
 2987  001162  000000                         0
 2988  001164  000000                         0
 2989  001166  000000                         0
 2990
 2991                                           ;UNIT DATA ERRORS REVERSE**********
 2992
 2993  001170  000000                 DEREV1: 0
 2994  001172  000000                         0
 2995  001174  000000                         0
 2996  001176  000000                         0
 2997  001200  000000                         0
 2998  001202  000000                         0
 2999  001204  000000                         0
 3000  001206  000000                         0
```

```
3002                                        ;DROPS + PICKS PER CHANNEL PER UNIT**********
3003
3004  001210  000000                BP00:   0
3005          001230                        .=.+16
3006  001230  000000                BP10:   0
3007          001250                        .=.+16
3008  001250  000000                BP20:   0
3009          001270                        .=.+16
3010  001270  000000                BP30:   0
3011          001310                        .=.+16
3012  001310  000000                BP40:   0
3013          001330                        .=.+16
3014  001330  000000                BP50:   0
3015          001350                        .=.+16
3016  001350  000000                BP60:   0
3017          001370                        .=.+16
3018  001370  000000                BP70:   0
3019          001410                        .=.+16
3020  001410  000000                BD00:   0
3021          001430                        .=.+16
3022  001430  000000                BD10:   0
3023          001450                        .=.+16
3024  001450  000000                BD20:   0
3025          001470                        .=.+16
3026  001470  000000                BD30:   0
3027          001510                        .=.+16
3028  001510  000000                BD40:   0
3029          001530                        .=.+16
3030  001530  000000                BD50:   0
3031          001550                        .=.+16
3032  001550  000000                BD60:   0
3033          001570                        .=.+16
3034  001570  000000                BD70:   0
3035          001610                        .=.+16
3036
3037
```

```
3039
3040                                        ;UNIT BAD TAPE COUNTER:16 PER SLAVE**********
3041
3042  001610  000000                BT00:   0
3043          001714                        .=.+102
3044  001714  000000                BT01:   0
3045          002020                        .=.+102
3046  002020  000000                BT02:   0
3047          002124                        .=.+102
3048  002124  000000                BT03:   0
3049          002230                        .=.+102
3050  002230  000000                BT04:   0
3051          002334                        .=.+102
3052  002334  000000                BT05:   0
3053          002440                        .=.+102
3054  002440  000000                BT06:   0
3055          002544                        .=.+102
3056  002544  000000                BT07:   0
3057          002650                        .=.+102
3058
3059                                        ;UNIT END OF TAPE COUNTERS 1 PER SLAVE*******
3060
3061  002650  000000                EOTC0:  0
3062  002652  000000                        0
3063  002654  000000                        0
3064  002656  000000                        0
3065  002660  000000                        0
3066  002662  000000                        0
3067  002664  000000                        0
3068  002666  000000                        0
3069
3070                                        ;UNIT READ FORWARD SOFT ERROR*******
3071
3072  002670  000000                RFSOFT: 0
3073  002672  000000                        0
3074  002674  000000                        0
3075  002676  000000                        0
3076  002700  000000                        0
3077  002702  000000                        0
3078  002704  000000                        0
3079  002706  000000                        0
3080
3081                                        ;UNIT READ REVERSE SOFT ERROR*******
3082
3083  002710  000000                RRSOFT: 0
3084  002712  000000                        0
3085  002714  000000                        0
3086  002716  000000                        0
3087  002720  000000                        0
3088  002722  000000                        0
3089  002724  000000                        0
3090  002726  000000                        0
3091
```

```
3093
3094                                        ;UNIT READ FORWARD HARD ERROR*******
3095
3096  002730  000000                RFHARD: 0
3097  002732  000000                        0
3098  002734  000000                        0
3099  002736  000000                        0
3100  002740  000000                        0
3101  002742  000000                        0
3102  002744  000000                        0
3103  002746  000000                        0
3104
3105                                        ;UNIT READ REVERSE HARD ERROR*******
3106
3107  002750  000000                RRHARD: 0
3108  002752  000000                        0
3109  002754  000000                        0
3110  002756  000000                        0
3111  002760  000000                        0
3112  002762  000000                        0
3113  002764  000000                        0
3114  002766  000000                        0
3115                                  ;SET END OF STATISTICS TABLE
3116  002770                        ENDTBL:
3117
3118                                        ;DATA PATTERN GENERATORS************
3119
3120  002770  002770                DATBL:  .                       ;ENTRY TABLE
3121  002772  014336                DATA0:  DAT0                    ;EXTERNAL INPUT FROM H/S READER(SEE CZTUTA0)
3122  002774  014476                DATA1:  DAT1                    ;ALL ONES
3123  002776  014516                DATA2:  DAT2                    ;ALL ZEROS
3124  003000  014522                DATA3:  DAT3                    ;WALKING ONE
3125  003002  014546                DATA4:  DAT4                    ;WALKING ZERO
3126  003004  014556                DATA5:  DAT5                    ;ALTERNATING ONE/ZERO
3127  003006  014564                DATA6:  DAT6                    ;ALTERNATING ZERO/ONE
3128  003010  014572                DATA7:  DAT7                    ;ALTERNATING ONE/ZERO IN ALTERNATING CHARACTERS
3129  003012  014620                DATA10: DAT10                   ;WALKING ONE/ALL ONE IN ALTERNATING CHARACTERS
3130  003014  014650                DATA11: DAT11                   ;ALL BITS 0-377
3131  003016  014670                DATA12: DAT12                   ;ALL BITS 377-0
3132  003020  014712                DATA13: DAT13                   ;ALTERNATING CHARACTERS  0 AND 377
3133  003022  014722                DATA14: DAT14                   ;WALKING ZERO/ALL ZERO IN ALTERNATING CHARACTERS
3134  003024  014752                DATA15: DAT15                   ;AUTO SEQUENCE PATTERN 0,0,-1,-1,-1,0,0
3135
```

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


                                ;START 200, & 300**********
003026  012706  000500          START:  MOV     #500,SP         ;SET STACK PTR
003032  005037  000736                  CLR     ASEQF           ;CLEAR AUTO SEQUENCE FLAG
003036  005027                          CLR     (PC)+           ;;CLEAR CHAIN INDICATOR
(1)     003040  000000          CHNFLG: .WORD   0               ;;CHAIN MODE INDICATOR
(1)                                                             ;;1/0 = CHAIN/NOT CHAIN MODE
(1)     003042  022737  005022  000042          CMP     #$ENDAD,@#42    ;;BRANCH IF LOADED VIA ACT11 CHAIN MODE
(1)     003050  001404                  BEQ     50$
(1)     003052  005737  000042          TST     @#42            ;;BRANCH IF IN DUMP MODE
(1)     003056  001413                  BEQ     52$
(1)     003060  000406                  BR      51$
(1)     003062  012737  000176  000610  50$:    MOV     #SWREG,SWR      ;;INVOKE SOFTWARE SWR
(1)     003070  012777  100000  175512          MOV     #100000,@SWR    ;;WITH HALT ON ERROR SET
(1)     003076  005237  003040  51$:    INC     CHNFLG          ;;SET CHNFLG = CHAIN MODE
(1)     003102  000137  003126          JMP     3$              ;;GO TO CHAIN ADDRESS
(1)     003106                  52$:
3158    003106  122737  000006  000041          CMPB    #6,@#41         ;BRANCH IF LOADED VIA TMDP
3159    003114  001010                  BNE     STAUT
3160    003116  012704  027356                  MOV     #MSG120,R4      ;ADVISE USER TO REMOVE TMDP FROM SLAVE
3161    003122  000004                  TYPE
3162    003124  000404                  BR      STAUT
3163    003126  005237  000736  3$:     INC     ASEQF           ;SET AUTO SEQUENCE FLAG
3164    003132  000137  022056          JMP     ASEQ0           ;GO TO AUTO SEQUENCER
3165
3166                                    ;START 240**********
3167    003136  012737  000001  000636  STAUT:  MOV     #1,TINF         ;SET TTY ENTRY FLAG
3168    003144  005037  015064          CLR     RDFL            ;CLEAR RANDOM DATA FLAG
3169    003150  000405                  BR      STARTB
3170
3171                                    ;START 204**********
3172    003152  005037  000636  STARTC: CLR     TINF            ;CLEAR TTY INPUT FLAG
3173    003156  000432                  BR      STARTD
3174
3175                                    ;START 210**********
3176    003160  005037  000636  STARTA: CLR     TINF            ;CLEAR TTY ENTRY FLAG
3177    003164  012700  000640  STARTB: MOV     #STFLG,R0       ;GET STARTING ADDRESS OF FLAGS
3178    003170  012701  000076          MOV     #ENDFLG-STFLG,R1
3179    003174  105020          1$:     CLRB    (R0)+           ;CLEAR FLAGS AND COUNTERS
3180    003176  005301                  DEC     R1
```

```
3181  003200  001375                        BNE     1$
3182  003202  012706  000500                MOV     #500,SP         ;SET STACK POINTER
3183  003206  004737  004276                JSR     PC,RANSET       ;GO RESET RANDOM BASE
3184  003212  012700  001050                MOV     #STTBL,R0       ;GET STARTING ADDRESS OF STAT TABLE
3185  003216  012701  001720                MOV     #ENDTBL-STTBL,R1        ;AND # OF BYTES IN TABLE
3186  003222  105020                2$:     CLRB    (R0)+           ;CLEAR STATISTIC COUNTERS
3187  003224  005301                        DEC     R1
3188  003226  001375                        BNE     2$
3189  003230  012737  177777  014330        MOV     #-1,PATS        ;PRESET PATTERN
3190  003236  012737  000001  000656  STARTE: MOV    #1,BLCNTR       ;PRESET BLOCK COUNTER
3191  003244  013746  000004          STARTD: MOV    @#4,-(SP)       ;SAVE ERROR TRAP VECTOR
3192  003250  013746  000006                MOV     @#6,-(SP)
3193  003254  022737  000176  000610        CMP     #SWREG,SWR      ;BRANCH IF SOFTWARE SWR
3194  003262  001413                        BEQ     2$              ;ALREADY SELECTED
3195  003264  012737  003310  000004        MOV     #1$,@#4         ;SET TIMEOUT TRAP TO 1$ BELOW
3196  003272  005037  000006                CLR     @#6
3197  003276  022777  177777  175304        CMP     #177777,@SWR    ;BRANCH IF SWR = 177777 TRAP
3198  003304  001402                        BEQ     2$              ;IF NOT AVAIL (1$) OTHERWISE
3199  003306  000404                        BR      3$              ;GO TO 3$
3200  003310  022626                1$:     CMP     (SP)+,(SP)+     ;RESET STACK
3201  003312  012737  000176  000610  2$:     MOV     #SWREG,SWR      ;SET SWR = SOFTWARE SWR
3202  003320  012637  000006          3$:     MOV     (SP)+,@#6       ;RESTORE ERROR TRAP
3203  003324  012637  000004                MOV     (SP)+,@#4
3204  003330  012706  000500                MOV     #500,SP
3205  003334  004737  012226                JSR     PC,TINP         ;GO GET PARAMETERS FROM TTY
3206  003340  012777  000040  175152        MOV     #40,@CS         ;INITIALIZE
3207  003346  005000                STAUTO: CLR     R0              ;POINT TO FIRST ENTRY
3208  003350  022760  177777  000746  1$:     CMP     #-1,UN1(R0)     ;BRANCH IF LAST ENTRY
3209  003356  001406                        BEQ     2$
3210  003360  042760  100000  000746        BIC     #100000,UN1(R0) ;CLEAR EOT FLAG
3211  003366  062700  000002                ADD     #2,R0           ;POINT TO NEXT UNIT ENTRY
3212  003372  000766                        BR      1$              ;CONTINUE CLEARING
3213  003374  013703  005054          2$:     MOV     REOTC,R3
3214  003400  000303                        SWAB    R3
3215  003402  110337  005054                MOVB    R3,REOTC        ;RESTORE EOT CNTR
3216  003406  012777  000100  175176  START1: MOV    #100,@TKS       ;SET KEYBOARD IE BIT
3217  003414  013700  000676                MOV     UNP,R0          ;R0 = UNIT TABLE POINTER
3218  003420  022760  177777  000746  STAR1A: CMP    #-1,UN1(R0)     ;BRANCH IF LAST ENTRY
3219  003426  001404                        BEQ     STAR1B
3220  003430  016037  000746  000552        MOV     UN1(R0),UDES    ;LOAD NEXT UNIT DESCRIPTION
3221  003436  000446                        BR      START4
3222  003440  005237  000656          STAR1B: INC    BLCNTR          ;BUMP BLOCK COUNTER
3223  003444  005737  000736                TST     ASEQF           ;SEE IF AUTO SEQ
3224  003450  001411                        BEQ     STAR1C          ;IF NOT: BR
3225  003452  023737  000656  000742        CMP     BLCNTR,ABLCNT   ;SEE IF DONE SEQ
3226  003460  001005                        BNE     STAR1C          ;IF NOT: BR
3227  003462  005037  000656                CLR     BLCNTR          ;RESET BLOCK CNTR
3228  003466  005037  000676                CLR     UNP             ;RESET UNIT POINTER
3229  003472  000207                        RTS     PC              ;RETURN TO AUTO SEQ
3230  003474  005037  000676          STAR1C: CLR    UNP
3231  003500  005000                        CLR     R0
3232  003502  016037  000746  000552        MOV     UN1(R0),UDES    ;LOAD FIRST UNIT DESCRIPTION
3233  003510  032777  000200  175072        BIT     #200,@SWR       ;SEE IF RANDOM RECORD SIZE
3234  003516  001402                        BEQ     START2          ;IF NOT: BR
3235  003520  004737  012142                JSR     PC,CCNTR        ;GO GENERATE RANDOM RECORD SIZE
3236  003524  032777  000400  175056  START2: BIT    #400,@SWR       ;SEE IF RANDOM DATA
```

```
3237  003532  001402                          BEQ     START3          ;IF NOT: BR
3238  003534  004737  015022                  JSR     PC,DATR         ;GO GENERATE RANDOM DATA
3239  003540  032777  000100  175042  START3: BIT     #100,@SWR       ;SEE IF RANDOM RECORD COUNT
3240  003546  001402                          BEQ     START4          ;IF NOT: BR
3241  003550  004737  012202                  JSR     PC,RCNTR        ;GO GENERATE RANDOM RECORD COUNT
3242  003554  005760  000746          START4: TST     UN1(R0)         ;SEE IF REACHED EOT
3243  003560  100002                          BPL     STAR40          ;IF NOT: BR
3244  003562  000137  004264                  JMP     START7          ;ELSE GO TO NEXT UNIT
3245  003566  013777  000550  174724  STAR40: MOV     DVN,@CS         ;SET DRIVE NUMBER
3246  003574  013777  000552  174740          MOV     UDES,@TC        ;SET UNIT NUMBER
3247  003602  105777  174714                  TSTB    @DS             ;SEE IF UNIT AVAIL
3248  003606  100412                          BMI     STAR4A          ;IF SO: BR
3249  003610  005337  000670                  DEC     STAL
3250  003614  001357                          BNE     START4          ;AWAIT TUR
3251  003616  004737  022672                  JSR     PC,PAPRT        ;PRINT HEADER
3252  003622  012704  026056                  MOV     #MSG49,R4
3253  003626  000004                          TYPE                    ;TYPE MSG
3254  003630  000000                          HALT                    ;STOP
3255  003632  000750                          BR      START4          ;RETRY
3256  003634  004737  014124          STAR4A: JSR     PC,DSUP         ;GO SET UP WRITE DATA
3257  003640  004737  005426                  JSR     PC,INIT         ;INIT SLAVE
3258  003644  004737  005056                  JSR     PC,RWND         ;REWIND
3259  003650  004737  005542                  JSR     PC,WRITE        ;WRITE
3260  003654  013737  000600  000670          MOV     TSTAL,STAL      ;SET TURN AROUND DELAY
3261  003662  004737  012132                  JSR     PC,STALL                ;DELAY
3262  003666  004737  007432                  JSR     PC,RSEQ         ;GO TO READ SEQUENCER
3263  003672  013737  000600  000670          MOV     TSTAL,STAL      ;SET TURN AROUND DELAY
3264  003700  004737  012132                  JSR     PC,STALL        ;DELAY
3265  003704  032777  040000  174676          BIT     #40000,@SWR     ;SEE IF SHOULD PRINT STATISTICS
3266  003712  001541                          BEQ     START5          ;IF NOT: BR
3267  003714  012700  000001                  MOV     #1,R0           ;SET RECORD COUNTER TO 1
3268  003720  004737  022672                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
3269  003724  004737  003734                  JSR     PC,STP          ;GO PRINT STATS
3270  003730  000137  004202                  JMP     STPX
3271  003734  004737  017160  STP:    JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
3272  003740  012704  026270                  MOV     #MSG65,R4
3273  003744  000004                          TYPE                    ;TYPE MSG
3274  003746  013704  000676                  MOV     UNP,R4
3275  003752  016403  001050                  MOV     RTY1(R4),R3
3276  003756  104400                          TYPOCT                  ;PRINT RETRIES
3277  003760  012704  026441                  MOV     #MSG73,R4
3278  003764  000004                          TYPE                    ;TYPE MSG
3279  003766  013704  000676                  MOV     UNP,R4
3280  003772  016403  001070                  MOV     WTER1(R4),R3
3281  003776  104400                          TYPOCT                  ;PRINT WRITE ERRORS
3282  004000  012704  026430                  MOV     #MSG72,R4
3283  004004  000004                          TYPE                    ;TYPE MSG
3284  004006  013704  000676                  MOV     UNP,R4
3285  004012  016403  001110                  MOV     RDER1(R4),R3
3286  004016  104400                          TYPOCT                  ;PRINT READ FORWARD ERRORS
3287  004020  012704  027233                  MOV     #MSG113,R4
3288  004024  000004                          TYPE                    ;TYPE MSG
3289  004026  013704  000676                  MOV     UNP,R4
3290  004032  016403  002670                  MOV     RFSOFT(R4),R3
3291  004036  104400                          TYPOCT                  ;PRINT FORWARD SOFT ERRORS
3292  004040  012704  027244                  MOV     #MSG114,R4
```

```
3293  004044  000004                          TYPE                    ;TYPE MSG
3294  004046  013704  000676                  MOV     UNP,R4
3295  004052  016403  002730                  MOV     RFHARD(R4),R3
3296  004056  104400                          TYPOCT                  ;PRINT HARD FORWARE ERRORS
3297  004060  012704  026521                  MOV     #MSG77,R4
3298  004064  000004                          TYPE                    ;TYPE MSG
3299  004066  013704  000676                  MOV     UNP,R4
3300  004072  016403  001130                  MOV     DATER1(R4),R3
3301  004076  104400                          TYPOCT                  ;PRINT DATA ERROR FORWARD NUMBER
3302  004100  012704  026324                  MOV     #MSG68,R4
3303  004104  000004                          TYPE                    ;TYPE MSG
3304  004106  013704  000676                  MOV     UNP,R4
3305  004112  016403  001150                  MOV     RDERR1(R4),R3
3306  004116  104400                          TYPOCT                  ;PRINT REVESE ERROR NUMBER
3307  004120  012704  027233                  MOV     #MSG113,R4
3308  004124  000004                          TYPE                    ;TYPE MSG
3309  004126  013704  000676                  MOV     UNP,R4
3310  004132  016403  002710                  MOV     RRSOFT(R4),R3
3311  004136  104400                          TYPOCT                  ;PRINT REVERSE SOFT ERPOR
3312  004140  012704  027244                  MOV     #MSG114,R4
3313  004144  000004                          TYPE                    ;TYPE MSG
3314  004146  013704  000676                  MOV     UNP,R4
3315  004152  016403  002750                  MOV     RRHARD(R4),R3
3316  004156  104400                          TYPOCT
3317  004160  012704  026510                  MOV     #MSG76,R4
3318  004164  000004                          TYPE                    ;TYPE MSG
3319  004166  013704  000676                  MOV     UNP,R4
3320  004172  016403  001170                  MOV     DEREV1(R4),R3
3321  004176  104400                          TYPOCT                  ;PRINT DATA REVERSE ERROR NUMBER
3322  004200  000207                          RTS     PC              ;RETURN
3323  004202  005237  000730          STPX:   INC     BTSTF           ;SET STAT ONLY PRINT
3324  004206  004737  007342                  JSR     PC,BTPRT        ;PRINT BAD TAPE STATS
3325  004212  005037  000730                  CLR     BTSTF           ;CLEAR FLAG
3326  004216  017700  174366          START5: MOV     @SWR,R0         ;LOAD SWR
3327  004222  042700  177762                  BIC     #177762,R0      ;MASK READ/WRITE SWITCHES
3328  004226  022700  000015                  CMP     #15,R0          ;SEE IF HAVE READ OR WRITE
3329  004232  001417                          BEQ     START8          ;IF NOT: BR
3330  004234  105777  174262          START6: TSTB    @DS             ;SEE IF HAVE UNIT READY
3331  004240  100411                          BMI     START7          ;IF SO: BR
3332  004242  005337  000670                  DEC     STAL
3333  004246  001372                          BNE     START6          ;DELAY FOR TUR
3334  004250  004737  022672                  JSR     PC,PAPRT        ;PRINT HEADER
3335  004254  012704  026056                  MOV     #MSG49,R4
3336  004260  000004                          TYPE                    ;TYPE MSG
3337  004262  000000                          HALT                    ;STOP
3338  004264  062737  000002  000676  START7: ADD     #2,UNP          ;POINT TO NEXT UNIT
3339  004272  000137  003406          START8: JMP     START1          ;CONTINUE
3340
3341                                          ;RANDOM BASE RESET**********
3342
3343  004276  012737  153624  000626  RANSET: MOV     #153624,RANBAS  ;RESET BASE
3344  004304  012737  032561  000630          MOV     #32561,RANSAV   ;RESET BUFFER
3345  004312  013737  000632  000554          MOV     RCSAV,RCNT      ;RESET RECORD COUNT
3346  004320  013737  000634  000556          MOV     FCSAV,FMCNT     ;RESET FRAME COUNT
3347  004326  000207                          RTS     PC
3348
```

```
3350                                    ;*****************************************
3351                                    ;REWIND FROM EOT
3352                                    ;
3353                                    ;WHEN ANY TRANSPORT BEING TESTED REACHES END OF TAPE
3354                                    ;DURING A READ OR WRITE OPERATION, IT WILL BE REWOUND
3355                                    ;AND FLAGGED AS UNAVAILABLE UNTIL ALL AVAILABLE UNITS
3356                                    ;HAVE REACHED EOT AT WHICH TIME ALL TESTING WILL BE RESUMED
3357                                    ;AT A BLOCK COUNT OF ONE (1).  A MESSAGE WILL BE
3358                                    ;PRINTED ON THE SUPERVISORS CONSOLE AS EACH UNIT REACHES
3359                                    ;EOT AND IS REWOUND.
3360                                    ;*****************************************
3361
3362  004330  013777  000552  174204  REOT:   MOV     UDES,@TC        ;LOAD TAPE CONTROL REGISTER
3363  004336  012777  000011  174144          MOV     #11,@C1         ;DRIVE CLEAR
3364  004344  105777  174152          1$:     TSTB    @DS             ;WAIT FOR DRY
3365  004350  100375                          BPL     1$
3366  004352  012777  000007  174130          MOV     #7,@C1          ;START REWIND
3367  004360  005737  000726                  TST     BTFLG           ;SEE IF BAD TAPE OVERFLOW REWIND
3368  004364  001004                          BNE     REOT1A          ;IF SO: BR
3369  004366  013700  000662                  MOV     EOTREC,R0
3370  004372  042700  100000                  BIC     #100000,R0      ;SET RECORD NUMBER OF EOT
3371  004376  005037  000662          REOT1A: CLR     EOTREC          ;CLEAR EOT INDICATOR & REC COUNT
3372  004402  004737  022672                  JSR     PC,PAPRT        ;PRINT HEADER
3373  004406  022737  000002  000726          CMP     #2,BTFLG        ;SEE IF POSITION ERROR
3374  004414  001003                          BNE     REOT1B          ;IF NOT: BR
3375  004416  012704  027124                  MOV     #MSG109,R4      ;SET POSITION ERROR MSG
3376  004422  000406                          BR      REOT1F
3377  004424  022737  000001  000726  REOT1B: CMP     #1,BTFLG        ;SEE IF BAD TAPE OVERFLOW
3378  004432  001004                          BNE     REOT1C          ;IF NOT: BR
3379  004434  012704  026752                  MOV     #MSG106,R4      ;SET BAD TAPE OVERFLOW MSG
3380  004440  000004                  REOT1F: TYPE                    ;TYPE MSG
3381  004442  000412                          BR      REOT1E
3382  004444  012704  024756          REOT1C: MOV     #MSG20,R4       ;SET EOT MSG
3383  004450  000004                          TYPE                    ;TYPE MSG
3384  004452  013704  000676                  MOV     UNP,R4
3385  004456  005264  002650                  INC     EOTC0(R4)       ;BUMP CNTR
3386  004462  016403  002650                  MOV     EOTC0(R4),R3
3387  004466  104400                          TYPOCT                  ;PRINT EOT CNTR
3388  004470  012704  026777          REOT1E: MOV     #MSG16A,R4
3389  004474  000004                          TYPE                    ;TYPE MSG
3390  004476  005037  000726                  CLR     BTFLG           ;CLEAR BAD TAPE FLAG
3391  004502  004737  003734                  JSR     PC,STP          ;PRINT STATS
3392  004506  004737  007342                  JSR     PC,BTPRT        ;PRINT BAD TAPE STATS
3393  004512  105777  174004          REOT2:  TSTB    @DS             ;BRANCH IF DRY SET
3394  004516  100414                          BMI     REOT2A
3395  004520  005337  000670                  DEC     STAL
3396  004524  001372                          BNE     REOT2           ;WAIT DRY
3397  004526  012737  024615  000654          MOV     #MSG6,EMADDR
3398  004534  004737  022672                  JSR     PC,PAPRT        ;PRINT HEADER
3399  004540  012704  026232                  MOV     #MSG60,R4
3400  004544  000004                          TYPE                    ;TYPE MSG
3401  004546  000000                          HALT
3402  004550  105337  005054          REOT2A: DECB    REOTC           ;SEE IF LAST UNIT TO REACH EOT
3403  004554  001410                          BEQ     REOT3           ;IF SO: BR
3404  004556  013700  000676                  MOV     UNP,R0
3405  004562  052760  100000  000746          BIS     #100000,UN1(R0) ;SET EOT FLAG
```

```
3406  004570  005726                          TST     (SP)+           ;RESET STACK POINTER
3407  004572  000137  004264                  JMP     START7          ;GO TO NEXT UNIT
3408  004576  000337  005054          REOT3:  SWAB    REOTC
3409  004602  013700  005054                  MOV     REOTC,R0
3410  004606  000337  005054                  SWAB    REOTC
3411  004612  110037  005054                  MOVB    R0,REOTC        ;RESTORE EOT UNIT COUNTER
3412  004616  005037  000676                  CLR     UNP
3413  004622  013700  000676                  MOV     UNP,R0          ;POINT TO FIRST UNIT
3414  004626  016037  000746  000552  REOT4:  MOV     UN1(R0),UDES    ;LOAD UNIT DESCRIPTION
3415  004634  013777  000552  173700          MOV     UDES,@TC        ;LOAD COMMAND REGISTER
3416  004642  032777  020000  173652  REOT5:  BIT     #20000,@DS
3417  004650  001374                          BNE     REOT5           ;AWAIT PIP RESET
3418  004652  032777  000002  173642          BIT     #2,@DS          ;SEE IF HAVE BOT
3419  004660  001012                          BNE     REOT6           ;IF SO: BR
3420  004662  012700  000001                  MOV     #1,R0
3421  004666  004737  022672                  JSR     PC,PAPRT        ;PRINT HEADER
3422  004672  012704  026023                  MOV     #MSG48,R4
3423  004676  000004                          TYPE                    ;TYPE MSG
3424  004700  000000                          HALT
3425  004702  013700  000676                  MOV     UNP,R0
3426  004706  042760  100000  000746  REOT6:  BIC     #100000,UN1(R0) ;CLEAR EOT FLAG
3427  004714  062737  000002  000676          ADD     #2,UNP
3428  004722  013700  000676                  MOV     UNP,R0          ;POINT TO NEXT UNIT
3429  004726  022760  177777  000746          CMP     #-1,UN1(R0)     ;BRANCH IF NOT LAST UNIT
3430  004734  001334                          BNE     REOT4
3431  004736  005037  000676          REOT7:  CLR     UNP             ;CLEAR UNIT POINTER
3432  004742  005037  000636                  CLR     TINF            ;CLEAR TTY INPUT FLAG
3433  004746  005737  000736                  TST     ASEQF           ;SEE IF AUTO SEQ
3434  004752  001402                          BEQ     REOTX           ;IF NOT: BR
3435  004754  005726                          TST     (SP)+           ;RESET STACK POINTER
3436  004756  000207                          RTS     PC              ;RETURN TO AUTO SEQ
3437  004760  004737  004276          REOTX:  JSR     PC,RANSET       ;GO RESET RANDOM BASE
3438  004764  012737  177777  014330          MOV     #-1,PATS        ;PRESET PATTERN
3439  004772  005037  015064                  CLR     RDFL            ;CLEAR RANDOM FLAG
3440  004776  005737  000572                  TST     SPFLG           ;SEE IF SINGLE PASS
3441  005002  001422                          BEQ     REOTXX          ;IF NOT: BR
3442  005004  012704  026633          TEND:   MOV     #MSG100,R4
3443  005010  000004                          TYPE                    ;TYPE MSG
3444  005012  013700  000042                  MOV     @#42,R0         ;GET ACT11 RETURN ADDRESS
 (1)  005016  001405                          BEQ     HERE            ;BRANCH IF NOT ACT11
 (1)  005020  000005                          RESET
 (1)  005022  004710          $ENDAD:  JSR     PC,(R0)
 (1)  005024  000240                          NOP
 (1)  005026  000240                          NOP
 (1)  005030  000240                          NOP
 (1)  005032  000240          HERE:   NOP
3445  005034  005737  003040                  TST     CHNFLG          ;BRANCH IF NOT CHAIN MODE
3446  005040  001402                          BEQ     1$
3447  005042  000137  022056                  JMP     ASEQ0           ;RETURN TO AUTO SEQUENCER
3448  005046  000000          1$:     HALT
3449  005050  000137  003236          REOTXX: JMP     STARTE          ;RESTART AT BLOCK NUMBER ONE
3450  005054  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.
                                        ;*********************************************************

005056  032777  001000  173524  RWND:   BIT     #1000,@SWR      ;SEE IF SHOULD REWIND
005064  001001                          BNE     RWNDA           ;IF SO: BR
005066  000207                          RTS     PC              ;ELSE EXIT
005070  013737  000676  000716  RWNDA:  MOV     UNP,UPS         ;SAVE UNIT POINTER
005076  005037  000676                  CLR     UNP             ;CLEAR POINTER
005102  005037  000662                  CLR     EOTREC          ;CLEAR EOT FLAG
005106  000337  005054                  SWAB    REOTC
005112  013700  005054                  MOV     REOTC,R0
005116  000337  005054                  SWAB    REOTC
005122  110037  005054                  MOVB    R0,REOTC        ;RESTORE EOT UNIT COUNTER
005126  013700  000676          RWND0:  MOV     UNP,R0          ;POINT TO UNIT ENTRY
005132  022760  177777  000746          CMP     #-1,UN1(R0)     ;BRANCH IF LAST ENTRY
005140  001445                          BEQ     RWND2
005142  005760  000746                  TST     UN1(R0)         ;SEE IF ALREADY REWINDING
005146  100433                          BMI     RWND1A          ;IF SO: BR
005150  016037  000746  000552          MOV     UN1(R0),UDES    ;SET UNIT DESCRIPTION
005156  013777  000552  173356          MOV     UDES,@TC        ;LOAD COMMAND REGISTER
005164  012777  000011  173316          MOV     #11,@C1         ;DRIVE CLEAR
005172  012777  000007  173310          MOV     #7,@C1          ;START REWIND
005200  105777  173316          1$:     TSTB    @DS
005204  100414                          BMI     RWND1A          ;IF DRY: BR
005206  005337  000670                  DEC     STAL
005212  001372                          BNE     1$              ;AWAIT DRY
005214  012737  024615  000654          MOV     #MSG6,EMADDR
005222  004737  022672                  JSR     PC,PAPRT        ;PRINT HEADER
005226  012704  026353                  MOV     #MSG70,R4
005232  000004                          TYPE                    ;TYPE MSG
005234  000000                          HALT
005236  042760  100000  000746  RWND1A: BIC     #100000,UN1(R0) ;CLEAR EOT FLAG
005244  062737  000002  000676          ADD     #2,UNP          ;BUMP POINTER
005252  000725                          BR      RWND0           ;DO NEXT UNIT
005254  005037  000676          RWND2:  CLR     UNP             ;CLEAR POINTER
005260  013700  000676          RWND3:  MOV     UNP,R0          ;POINT TO UNIT ENTRY
005264  022760  177777  000746          CMP     #-1,UN1(R0)     ;BRANCH IF LAST ENTRY
005272  001441                          BEQ     RWNDX
005274  016037  000746  000552          MOV     UN1(R0),UDES    ;SET UNIT DESCRIPTION
005302  013777  000552  173232          MOV     UDES,@TC        ;LOAD COMMAND REGISTER
005310  032777  020000  173204  1$:     BIT     #20000,@DS
005316  001374                          BNE     1$              ;AWAIT PIP RESET
005320  013777  000552  173214          MOV     UDES,@TC        ;LOAD UNIT DESCRIPTION
005326  032777  000002  173166          BIT     #2,@DS          ;SEE IF HAVE BOT
005334  001407                          BEQ     RWND6           ;IF NOT: BR
005336  062737  000002  000676  RWND5:  ADD     #2,UNP          ;BUMP POINTER
005344  012777  000011  173136          MOV     #11,@C1         ;DRIVE CLEAR
005352  000742                          BR      RWND3           ;DO NEXT UNIT
005354  012700  000001          RWND6:  MOV     #1,R0
005360  004737  022672                  JSR     PC,PAPRT        ;PRINT HEADER
```

```
3508  005364  012704  026023                  MOV     #MSG48,R4
3509  005370  000004                          TYPE                    ;TYPE MSG
3510  005372  000000                          HALT
3511  005374  000760                          BR      RWND5           ;DO NEXT UNIT
3512  005376  013737  000716  000676  RWNDX:  MOV     UPS,UNP         ;RESTORE UNIT POINTER
3513  005404  013700  000676                  MOV     UNP,R0
3514  005410  016037  000746  000552          MOV     UN1(R0),UDES    ;RESET UNIT DESCRIPTION
3515  005416  013777  000552  173116          MOV     UDES,@TC
3516  005424  000207                          RTS     PC              ;RETURN TO TEST
3517
3518
3519                                          ;*******************************************************
3520                                          ;INITIALIZE SELECTED SALVE
3521                                          ;THIS ROUTINE REWINDS AND SETS THE PROPER DENSITY IF
3522                                          ;THE DENSITY REQUIRED FOR THE TEST IS DIFFERENT FROM
3523                                          ;THE DENSITY AT WHICH THE SLAVE IS SELECTED.
3524                                          ;*******************************************************
3525
3526  005426  013746  000552          INIT:   MOV     UDES,-(SP)      ;GET UNIT DESCRIPTION
3527  005432  013777  000550  173060          MOV     DVN,@CS         ;LOAD DRIVE #
3528  005440  011677  173076                  MOV     (SP),@TC        ;LOAD SLAVE # & SLAVE DESCRIPTION
3529  005444  042716  174377                  BIC     #174377,(SP)    ;CLEAR ALL BUT DENSITY BITS
3530  005450  022726  001400                  CMP     #1400,(SP)+     ;BRANCH IF NOT NRZ
3531  005454  001005                          BNE     1$
3532  005456  032777  000040  173036          BIT     #40,@DS         ;BRANCH IF SLAVE IS IN PE MODE
3533  005464  001425                          BEQ     4$              ;PES = 0
3534  005466  000404                          BR      2$
3535  005470  032777  000040  173024  1$:     BIT     #40,@DS         ;BRANCH IF SLAVE IS IN PE MODE
3536  005476  001020                          BNE     4$              ;PES = 1
3537  005500  012777  000011  173002  2$:     MOV     #11,@C1         ;CLEAR DRIVE
3538  005506  105777  173010          20$:    TSTB    @DS             ;WAIT FOR READY
3539  005512  100375                          BPL     20$
3540  005514  012777  000007  172766          MOV     #7,@C1          ;LOAD REWIND COMMAND
3541  005522  105777  172774          10$:    TSTB    @DS             ;WAIT FOR READY
3542  005526  100375                          BPL     10$
3543  005530  032777  020000  172764  3$:     BIT     #20000,@DS      ;WAIT FOR PIP=0
3544  005536  001374                          BNE     3$
3545  005540  000207                  4$:     RTS     PC
```

```
3547                                    ;*********************************************************
3548                                    ;WRITE ROUTINE:
3549                                    ;
3550                                    ;THIS ROUTINE IS USED TO WRITE ONTO TAPE THE BLOCK
3551                                    ;OF DATA DESCRIBED BY THE OPERATOR AND SET UP
3552                                    ;IN THE SEQUENCE FORMATTER.  THE TAPE UNIT TO BE USED
3553                                    ;HAS BEEN ASSIGNED BY THE SEQUENCE FORMATTER AND
3554                                    ;ITS PARAMETERS SET IN A UNIT DESCRIPTION WORD.
3555                                    ;AS EACH RECORD OF THE BLOCK IS WRITTEN, IT IS CHECKED
3556                                    ;FOR STATUS ERRORS, WORD COUNT ZERO, AND CORRECT CURRENT
3557                                    ;MEMORY ADDRESS.  IF THE WRITE OPERATION RESULTS IN
3558                                    ;ANY ERROR CONDITION, A WRITE RETRY OF THAT OPERATION
3559                                    ;MAY BE DONE BY SETTING SWITCH FOUR (4) TO A ONE (1).
3560                                    ;THE RETRY CONSISTS OF A BACKSPACE, ERASE FORWARD, AND
3561                                    ;REWRITE OF THE RECORD.  (SEE WRITE RETRY SUBROUTINE)
3562                                    ;AFTER ALL DATA RECORDS IN THE BLOCK HAVE BEEN
3563                                    ;WRITTEN, THE WRITE ROUTINE WILL EXECUTE A WRITE
3564                                    ;TAPE MARK COMMAND IF THE TTY RESPONSE TM=1 WAS
3565                                    ;MADE AT INITIAL START.  THE TM IS COUNTED AS TOTAL
3566                                    ;DATA RECORDS PLUS ONE (IE: IF 100 DATA RECORDS; TM=RECORD 101)
3567                                    ;IF THE WRITE OPERATION (DATA OR TM) CAUSES THE SELECTED SLAVE
3568                                    ;TO REACH END OF TAPE (EOT) AND THERE IS TO BE NO READING DONE,
3569                                    ;(SW2 AND SW3 SET TO A 1) THEN THE SLAVE IS REWOUND AND
3570                                    ;FLAGGED AS UNAVAILABLE FOR TESTING UNTIL ALL SLAVES HAVE
3571                                    ;REACHED EOT AND BEEN REWOUND AT WHICH TIME TESTING IS
3572                                    ;RESUMED ON ALL AVAILABLE SLAVES.
3573                                    ;WRITE RETRY MAY BE ALLOWED VIA CONSOLE SWITCH FOUR (4).
3574                                    ;ERROR CHECKING MAY BE DISALLOWED VIA CONSOLE SWITCH
3575                                    ;TWELVE (12).
3576                                    ;WRITING TO TAPE MAY BE DISALLOWED VIA CONSOLE SWITCH
3577                                    ;ZERO (0).
3578                                    ;*********************************************************
3579
3580  005542  032777  000001  173040  WRITE:  BIT     #1,@SWR          ;SEE IF SHOULD WRITE
3581  005550  001402                          BEQ     WRTE
3582  005552  000137  006340                  JMP     WEX              ;IF NOT: BR
3583  005556  013700  000554          WRTE:   MOV     RCNT,R0          ;R0=RECORD COUNT
3584  005562  012737  024610  000654  W0:     MOV     #MSG5,EMADDR     ;SET ERROR MSG ADDRESS
3585  005570  013777  000556  172720          MOV     FMCNT,@FC        ;LOAD CHAR COUNT
3586  005576  012777  027452  172710          MOV     #WDATA,@BA       ;SET DATA ADDR
3587  005604  112737  000060  000674          MOVB    #60,MTC1         ;SET WRITE OP COMMAND
3588  005612  012737  005624  000664          MOV     #W1,RTRN         ;SET RETURN ADDRESS
3589  005620  000137  021144                  JMP     TAPG             ;GO EXECUTE COMMAND
3590  005624  032777  002000  172670  W1:     BIT     #2000,@DS        ;SEE IF EOT
3591  005632  001414                          BEQ     W2               ;IF NOT AT EOT: BR
3592  005634  005737  000662                  TST     EOTREC           ;BRANCH IF WRITTEN PAST EOT
3593  005640  100411                          BMI     W2
3594  005642  010037  000662                  MOV     R0,EOTREC        ;SAVE RECORD COUNT
3595  005646  052737  100000  000662          BIS     #100000,EOTREC   ;++B SET EOT INDICATOR
3596  005654  005337  000662                  DEC     EOTREC           ;++B ADJUST RECORD COUNT
3597  005660  012700  000002                  MOV     #2,R0            ;SET TO WRITE 1 LAST RECORD
3598  005664  032777  010000  172716  W2:     BIT     #10000,@SWR      ;SEE IF SHOULD CHECK ERRORS
3599  005672  001002                          BNE     W3               ;IF NOT: BR
3600  005674  004737  017316                  JSR     PC,ERCHK         ;GO CHECK ERRORS
3601  005700  013737  000576  000670  W3:     MOV     WSTAL,STAL       ;SET DELAY
3602  005706  004737  012132                  JSR     PC,STALL         ;DELAY
```

```
3603  005712  005737  000714          TST     RTYFL           ;SEE IF RETRY TIME
3604  005716  001401                  BEQ     W3A             ;IF NOT: BR
3605  005720  000207                  RTS     PC              ;ELSE RETURN
3606  005722  005737  000710  W3A:    TST     SERFL           ;SEE IF WRITE ERROR
3607  005726  001450                  BEQ     W5              ;IF NOT: BR
3608  005730  013704  000676          MOV     UNP,R4
3609  005734  005264  001070          INC     WTER1(R4)       ;BUMP WRITE ERROR
3610  005740  005037  000710          CLR     SERFL           ;CLEAR STATUS ERROR FLAG
3611  005744  032777  000020  172636  BIT     #20,@SWR        ;SEE IF RETRY
3612  005752  001436                  BEQ     W5              ;IF NOT: BR
3613  005754  013703  000724          MOV     ERSAV,R3
3614  005760  042703  102700          BIC     #102700,R3      ;MASK UNRECOVERABLE ERROR
3615  005764  001410                  BEQ     W4              ;IF SO: BR
3616  005766  004737  022672          JSR     PC,PAPRT        ;PRINT HEADER
3617  005772  012704  026532          MOV     #MSG78,R4
3618  005776  000004                  TYPE                    ;TYPE MSG
3619  006000  004737  011252          JSR     PC,NRTP         ;PRINT ER FOR NON-RETRYABLE
3620  006004  000421                  BR      W5
3621  006006  013704  000676  W4:     MOV     UNP,R4
3622  006012  005264  001050          INC     RTY1(R4)        ;BUMP RETRY CNTR
3623  006016  032777  002000  172564  BIT     #2000,@SWR      ;SEE IF PRINT ERRORS
3624  006024  001003                  BNE     W4A             ;IF NOT: BR
3625  006026  012704  026246          MOV     #MSG64,R4
3626  006032  000004                  TYPE                    ;TYPE MSG
3627  006034  005037  000704  W4A:    CLR     RTCNT           ;CLEAR RETRY NUMBER
3628  006040  005037  000702          CLR     RPCNT           ;CLEAR REPEAT COUNTER
3629  006044  004737  006402          JSR     PC,WRTY         ;GO RETRY WRITE ERROR
3630  006050  005037  000714  W5:     CLR     RTYFL           ;CLEAR RETRY COUNTER
3631  006054  005300                  DEC     R0              ;SEE IF DONE ALL
3632  006056  001241                  BNE     W0              ;IF NOT: BR
3633  006060  005737  000564  W6:     TST     TMEX            ;SEE IF TM
3634  006064  001525                  BEQ     WEX             ;IF NOT: BR
3635  006066  005237  000700          INC     TMFLG           ;SET TM FLAG
3636  006072  012737  026153  000654  WTM:    MOV     #MSG54,EMADDR   ;POINT TO TM ERROR MSG
3637  006100  012737  000026  000674          MOV     #26,MTC1        ;SET TM OP CODE
3638  006106  012777  000000  172402          MOV     #0,@FC          ;LOAD FRAME COUNTER
3639  006114  012777  027452  172372          MOV     #WDATA,@BA      ;LOAD BUS ADDRESS
3640  006122  012737  006134  000664          MOV     #WTM0,RTRN      ;SAVE RETURN ADDRESS
3641  006130  000137  021144                  JMP     TAPG            ;WRITE TM
3642  006134  032777  010000  172446  WTM0:   BIT     #10000,@SWR     ;SEE IF SHOULD CHECK ERRORS
3643  006142  001076                  BNE     WEX
3644  006144  032777  000004  172350          BIT     #4,@DS          ;SEE IF TM STATUS
3645  006152  001011                  BNE     WTM1            ;IF SO: BR
3646  006154  012737  027452  021064          MOV     #WDATA,CADER    ;SET EXPT BUS ADDRESS
3647  006162  012737  000001  021072          MOV     #1,DRVER        ;INDICATE ERROR
3648  006170  004737  020136                  JSR     PC,ERPT         ;PRINT TM ERROR
3649  006174  000404                  BR      WTM2
3650  006176  012703  027452  WTM1:   MOV     #WDATA,R3       ;SET EXPT ADDRESS
3651  006202  004737  017412          JSR     PC,ER2          ;GO CHECK FOR OTHER ERRORS
3652  006206  005737  000714  WTM2:   TST     RTYFL           ;SEE IF RETRY
3653  006212  001401                  BEQ     WTM3            ;IF NOT: BR
3654  006214  000207                  RTS     PC              ;ELSE RETURN TO RETRY ROUTINE
3655  006216  005737  000710  WTM3:   TST     SERFL           ;SEE IF WRITE ERROR
3656  006222  001446                  BEQ     WEX             ;IF NOT: BR
3657  006224  013704  000676          MOV     UNP,R4
3658  006230  005264  001070          INC     WTER1(R4)       ;BUMP WRITE ERROR
```

```
3659  006234  032777  000020  172346          BIT     #20,@SWR        ;SEE IF SHOULD RETRY
3660  006242  001436                          BEQ     WEX             ;IF NOT: BR
3661  006244  013703  000724                  MOV     ERSAV,R3
3662  006250  042703  102700                  BIC     #102700,R3      ;MASK UNRECOVERABLE ERROR
3663  006254  001410                          BEQ     WTM4            ;IF SO: BR
3664  006256  004737  022672                  JSR     PC,PAPRT        ;PRINT HEADER
3665  006262  012704  026532                  MOV     #MSG78,R4
3666  006266  000004                          TYPE                    ;TYPE MSG
3667  006270  004737  011252                  JSR     PC,NRTP         ;PRINT ER FOR NON-RETRYABLE
3668  006274  000421                          BR      WEX
3669  006276  005037  000702          WTM4:   CLR     RPCNT           ;CLEAR REPEAT CNTR
3670  006302  013704  000676                  MOV     UNP,R4
3671  006306  005264  001050                  INC     RTY1(R4)        ;BUMP RETRY CNTR
3672  006312  005037  000704                  CLR     RTCNT           ;CLEAR RETRY CNTR
3673  006316  032777  002000  172264          BIT     #2000,@SWR      ;SEE IF PRINT ERRORS
3674  006324  001003                          BNE     WTM4A           ;IF NOT: BR
3675  006326  012704  026246                  MOV     #MSG64,R4
3676  006332  000004                          TYPE                    ;TYPE MSG
3677  006334  004737  006402          WTM4A:  JSR     PC,WRTY         ;GO DO RETRY
3678  006340  005037  000714          WEX:    CLR     RTYFL           ;CLEAR RETRY FLAG
3679  006344  005037  000700                  CLR     TMFLG           ;CLEAR TAPE MARK FLAG
3680  006350  005737  000662                  TST     EOTREC          ;BRANCH IF NOT AT EOT
3681  006354  100011                          BPL     WRWX
3682  006356  017703  172226          WRW:    MOV     @SWR,R3
3683  006362  042703  177763                  BIC     #177763,R3
3684  006366  022703  000014                  CMP     #14,R3          ;SEE IF WRITE ONLY
3685  006372  001002                          BNE     WRWX            ;IF NOT: BR
3686  006374  000137  004330                  JMP     REOT            ;ELSE REWIND
3687  006400  000207          WRWX:   RTS     PC              ;EXIT
```

```
3689                                    ;*************************
3690                                    ;WRITE ERROR RETRY
3691                                    ;
3692                                    ;*************************
3693
3694  006402  012737  000001  000714  WRTY:   MOV     #1,RTYFL        ;SET RETRY FLAG
3695  006410  004737  007004          WRTY0:  JSR     PC,WRTSB        ;GO SPACE REVERSE FOR REPEAT
3696  006414  005737  000700                  TST     TMFLG           ;SEE IF TAPE MARK TIME
3697  006420  001003                          BNE     WRTYTM          ;IF SO: BR
3698  006422  004737  005562                  JSR     PC,W0           ;REWRITE RECORD
3699  006426  000402                          BR      WRTYR           ;GO ON
3700  006430  004737  006072          WRTYTM: JSR     PC,WTM          ;GO WRITE TAPE MARK AGAIN
3701  006434  005737  000710          WRTYR:  TST     SERFL           ;REWRITE GOOD
3702  006440  001024                          BNE     WRTY2           ;IF NOT: BR
3703  006442  005237  000702                  INC     RPCNT           ;BUMP REPEAT COUNTER
3704  006446  022737  000003  000702          CMP     #3,RPCNT        ;SEE IF THREE GOOD REPEATS
3705  006454  001355                          BNE     WRTY0           ;IF NOT: REPEAT
3706  006456  032777  002000  172124          BIT     #2000,@SWR      ;SEE IF PRINT
3707  006464  001011                          BNE     WRTY1           ;IF NOT: BR
3708  006466  012704  026737                  MOV     #MSG105,R4
3709  006472  000004                          TYPE                    ;TYPE MSG
3710  006474  012704  026270                  MOV     #MSG65,R4
3711  006500  000004                          TYPE                    ;TYPE MSG
3712  006502  013703  000704                  MOV     RTCNT,R3
3713  006506  104400                          TYPOCT                  ;PRINT RETRY NUMBER
3714  006510  000207          WRTY1:  RTS     PC                      ;RESUME TESTING
3715  006512  013703  000724          WRTY2:  MOV     ERSAV,R3        ;GET ER
3716  006516  005037  000650                  CLR     TEMP3           ;CLEAR RECOVERABLE ERROR INDICATOR
3717  006522  042703  102700                  BIC     #102700,R3      ;MASK RECOVERABLE BITS
3718  006526  001413                          BEQ     WRTY2A          ;IF RECOVERABLE: BR
3719  006530  004737  022672                  JSR     PC,PAPRT        ;PRINT HEADER
3720  006534  012704  026532                  MOV     #MSG78,R4
3721  006540  000004                          TYPE                    ;TYPE MSG
3722  006542  004737  011252                  JSR     PC,NRTP         ;PRINT ER
3723  006546  012737  000001  000650          MOV     #1,TEMP3        ;SET FLAG
3724  006554  000407                          BR      WRTY2B
3725  006556  032777  002000  172024  WRTY2A: BIT     #2000,@SWR      ;SEE IF PRINT
3726  006564  001025                          BNE     WRTY3           ;IF NOT: BR
3727  006566  012704  027156                  MOV     #MSG110,R4
3728  006572  000004                          TYPE                    ;TYPE MSG
3729  006574  012704  026270          WRTY2B: MOV     #MSG65,R4
3730  006600  000004                          TYPE                    ;TYPE MSG
3731  006602  013703  000704                  MOV     RTCNT,R3
3732  006606  104400                          TYPOCT                  ;PRINT RETRY NUMBER
3733  006610  012704  027200                  MOV     #MSG111,R4
3734  006614  000004                          TYPE                    ;TYPE MSG
3735  006616  013703  000702                  MOV     RPCNT,R3
3736  006622  104400                          TYPOCT                  ;PRINT REPEAT NUMBER
3737  006624  005737  000650                  TST     TEMP3           ;SEE IF DID NON-RECOVERABLE
3738  006630  001403                          BEQ     WRTY3           ;IF NOT: BR
3739  006632  005037  000650                  CLR     TEMP3           ;CLEAR FLAG
3740  006636  000207                          RTS     PC              ;EXIT
3741  006640  005737  000704          WRTY3:  TST     RTCNT           ;SEE IF FIRST RETRY
3742  006644  001004                          BNE     WRTY3A          ;IF NOT: BR
3743  006646  013704  000676                  MOV     UNP,R4
3744  006652  005364  001070                  DEC     WTER1(R4)       ;DECREMENT WRITE ERROR CNTR
```

```
3745  006656  013704  000676          WRTY3A: MOV     UNP,R4          ;GET UNIT NUMBER
3746  006662  016437  001030  000732          MOV     BTADDR(R4),BTPT ;GET ADDRESS OF UNIT BAD TAPE CNTR
3747  006670  017704  172036                   MOV     @BTPT,R4        ;GET COUNTER
3748  006674  005724                          TST     (R4)+           ;SET POINTER OFFSET
3749  006676  010477  172030                   MOV     R4,@BTPT
3750  006702  013703  000732                   MOV     BTPT,R3
3751  006706  060304                          ADD     R3,R4           ;SET ABSOLUTE POINTER
3752  006710  013714  000656                   MOV     BLCNTR,(R4)     ;SET BLOCK NUMBER
3753  006714  062704  000040                   ADD     #40,R4          ;ADD RCNT OFFSET
3754  006720  013714  000554                   MOV     RCNT,(R4)
3755  006724  160014                          SUB     R0,(R4)         ;SET RECORD NUMBER
3756  006726  005214                          INC     (R4)            ;CORRECT RECORD NUMBER
3757  006730  022777  000040  171774          CMP     #40,@BTPT       ;SEE IF TOO MANY BAD SPOTS
3758  006736  001002                          BNE     WRTY4           ;IF NOT: BR
3759  006740  000137  007200                   JMP     BTOV            ;ELSE GO TO BAD TAPE OVERFLOW
3760  006744  005237  000704          WRTY4:  INC     RTCNT           ;BUMP RETRY COUNTER
3761  006750  022737  000004  000704          CMP     #4,RTCNT        ;SEE IF DONE 4 RETRIES
3762  006756  001410                          BEQ     WRTY5           ;IF SO: BR
3763  006760  013704  000676                   MOV     UNP,R4
3764  006764  005264  001050                   INC     RTY1(R4)        ;BUMP RETRY COUNTER
3765  006770  005237  000734                   INC     ERTFL           ;SET ERASE FLAG
3766  006774  000137  006410                   JMP     WRTY0           ;DO NEXT RETRY
3767  007000  000137  007416          WRTY5:  JMP     BTUR            ;ELSE GO TO BAD TAPE UNRECOVERABLE
3768
3769                                          ;WRITE RETRY BACKSPACE-ERASE SUBROUTINE**********
3770
3771  007004  005037  000710          WRTSB:  CLR     SERFL           ;CLEAR FLAG
3772  007010  013737  000600  000670          MOV     TSTAL,STAL
3773  007016  004737  012132                   JSR     PC,STALL        ;DO TURN AROUND DELAY
3774  007022  012737  026301  000654          MOV     #MSG66,EMADDR   ;SET ERROR CODE
3775  007030  012777  177777  171460          MOV     #-1,@FC         ;SET TO BACKSPACE 1 RECORD
3776  007036  012777  033460  171450          MOV     #RDATA,@BA      ;SET BA
3777  007044  004737  012062                   JSR     PC,BKRT         ;GO BACKSPACE
3778  007050  005737  000710                   TST     SERFL           ;SEE IF ERROR
3779  007054  001406                          BEQ     WRTSB1          ;IF NOT: BR
3780  007056  012737  000002  000726  WRTSB0: MOV     #2,BTFLG        ;SET FLAG
3781  007064  022626                          CMP     (SP)+,(SP)+     ;RESET STACK
3782  007066  000137  004330                   JMP     REOT            ;GO REWIND AND REMOVE FROM TESTING
3783  007072  005737  000734          WRTSB1: TST     ERTFL           ;SEE IF SHOULD ERASE
3784  007076  001001                          BNE     WRTSB2          ;IF SO: BR
3785  007100  000207                          RTS     PC              ;RETURN
3786  007102  005037  000734          WRTSB2: CLR     ERTFL           ;CLEAR ERASE FLAG
3787  007106  005037  000702                   CLR     RPCNT           ;CLEAR REPEAT CNTR
3788  007112  005037  000710                   CLR     SERFL           ;CLEAR FLAG
3789  007116  012737  026314  000654          MOV     #MSG67,EMADDR   ;SET ERROR CODE
3790  007124  005077  171366                   CLR     @FC             ;CLEAR FRAME COUNT
3791  007130  012737  000024  000674          MOV     #24,MTC1        ;SET ERASE OP-CODE
3792  007136  012777  027452  171350          MOV     #WDATA,@BA      ;SET BA
3793  007144  012737  007156  000664          MOV     #WRTSB3,RTRN    ;SET RETURN ADDRESS
3794  007152  000137  021144                   JMP     TAPG            ;GO ERASE
3795  007156  012703  027452          WRTSB3: MOV     #WDATA,R3       ;SET EXPT BA
3796  007162  004737  017412                   JSR     PC,ER2          ;GO CHECK ERRORS
3797  007166  005737  000710                   TST     SERFL           ;SEE IF ERROR
3798  007172  001737                          BEQ     WRTSB1          ;IF NOT: BR
3799  007174  000137  007056                   JMP     WRTSB0
3800
```

```
3801                                        ;BAD TAPE OVERFLOW SUBROUTINE**********
3802
3803  007200  005037  000714        BTOV:   CLR     RTYFL           ;CLEAR RETRY FLAG
3804  007204  012737  000001  000726        MOV     #1,BTFLG        ;SET BAD TAPE OVERFLOW FLAG
3805  007212  005726                        TST     (SP)+           ;++B ADJUST STACK PTR
3806  007214  000137  004330                JMP     REOT            ;GO REWIND AND REMOVE FROM TESTING
3807  007220  013701  000732        BTOV0:  MOV     BTPT,R1         ;SET TABLE POINTER
3808  007224  005721                        TST     (R1)+
3809  007226  005000                        CLR     R0
3810  007230  010003                BTOV1:  MOV     R0,R3
3811  007232  000241                        CLC
3812  007234  006003                        ROR     R3              ;R3=R3/2 FOR CORRECT NUMBER
3813  007236  104400                        TYPOCT                  ;PRINT ENTRY NUMBER
3814  007240  012704  024705                MOV     #MSG13+1,R4
3815  007244  000004                        TYPE                    ;TYPE MSG
3816  007246  011103                        MOV     (R1),R3
3817  007250  104400                        TYPOCT                  ;PRINT BLOCK NUMBER
3818  007252  012704  024712                MOV     #MSG14,R4
3819  007256  000004                        TYPE                    ;TYPE MSG
3820  007260  062701  000040                ADD     #40,R1          ;SET POINTER OFFSET FOR RECOED NUMBER
3821  007264  012103                        MOV     (R1)+,R3
3822  007266  104400                        TYPOCT                  ;PRINT RECORD NUMBER
3823  007270  162701  000040                SUB     #40,R1          ;RESET POINTER FOR BLOCK NUMBER
3824  007274  005720                        TST     (R0)+
3825  007276  020077  171430                CMP     R0,@BTPT        ;SEE IF DONE
3826  007302  001404                        BEQ     BTOV2           ;IF SO: BR
3827  007304  012704  025237                MOV     #MSG28,R4
3828  007310  000004                        TYPE                    ;TYPE MSG
3829  007312  000746                        BR      BTOV1           ;CONTINUE
3830  007314  005737  000730        BTOV2:  TST     BTSTF           ;SEE IF STAT ONLY PRINT
3831  007320  001007                        BNE     BTOVX           ;IF SO: BR
3832  007322  012703  000041                MOV     #41,R3          ;SET SIZE OF TABLE
3833  007326  013704  000732                MOV     BTPT,R4         ;SET POINTER
3834  007332  005024                BTOV3:  CLR     (R4)+           ;CLEAR TABLE
3835  007334  005303                        DEC     R3              ;SEE IF DONE
3836  007336  001375                        BNE     BTOV3           ;IF NOT: BR
3837  007340  000207                BTOVX:  RTS     PC              ;RETURN
3838
```

```
3840
3841                                        ;BAD TAPE STATISTIC PRINT********
3842
3843  007342  012704  025237        BTPRT: MOV     #MSG28,R4
3844  007346  000004                       TYPE                    ;TYPE MSG
3845  007350  013704  000676               MOV     UNP,R4
3846  007354  016437  001030  000732       MOV     BTADDR(R4),BTPT ;SET TABLE POINTER
3847  007362  017703  171344               MOV     @BTPT,R3
3848  007366  000241                       CLC
3849  007370  006003                       ROR     R3              ;CORRECT NUMBER
3850  007372  104400                       TYPOCT                  ;PRINT NUMBER OF BAD SPOTS
3851  007374  012704  027212               MOV     #MSG112,R4
3852  007400  000004                       TYPE                    ;TYPE MSG
3853  007402  005777  171324               TST     @BTPT           ;SEE IF ANY BAD SPOTS
3854  007406  001001                       BNE     BTPRT1          ;IF SO: BR
3855  007410  000207                       RTS     PC              ;ELSE RETURN
3856  007412  000137  007220      BTPRT1: JMP     BTOV0           ;PRINT STATS
3857
3858                                        ;BAD TAPE UNRECOVERABLE SUBROUTINE**********
3859
3860  007416  004737  022672        BTUR:  JSR     PC,PAPRT        ;PRINT HEADER
3861  007422  012704  027040               MOV     #MSG107,R4
3862  007426  000004                       TYPE                    ;TYPE MSG
3863  007430  000207                       RTS     PC              ;RESUME TESTING
3864
```

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

007432  012737  000002  000562  RSEQ:   MOV     #2,RDCMD
007440  017704  171144                  MOV     @SWR,R4         ;READ SWITCHES
007444  042704  177763                  BIC     #177763,R4      ;MASK READ BITS & SEE IF BOTH READS
007450  001004                          BNE     RSR             ;IF NOT: BR
007452  032777  000002  171130          BIT     #2,@SWR         ;SEE IF READ REVERSE FIRST
007460  001050                          BNE     RSFR            ;IF NOT: BR
007462  032777  000004  171120  RSR:    BIT     #4,@SWR         ;SEE IF SHOULD READ REVERSE
007470  001005                          BNE     RSF             ;IF NOT: BR
007472  012737  010000  000562          MOV     #10000,RDCMD    ;LOAD READ REVERSE COMMAND
007500  004737  007744                  JSR     PC,READ         ;GO READ REVERSE
007504  032777  000010  171076  RSF:    BIT     #10,@SWR        ;SEE IF SHOULD READ FORWARD
007512  001025                          BNE     RSEX            ;IF NOT: BR
007514  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF HAVE READ REVERSE
007522  001406                          BEQ     RSF0            ;IF NOT: BR
007524  013737  000600  000670          MOV     TSTAL,STAL
007532  004737  012132                  JSR     PC,STALL        ;DO READ STALL
007536  000406                          BR      RSF1
007540  032777  000001  171042  RSF0:   BIT     #1,@SWR         ;SEE IF WRITE
007546  001002                          BNE     RSF1            ;IF NOT: BR
007550  004737  011706                  JSR     PC,BKSP         ;GO BACKSPACE
007554  012737  000002  000562  RSF1:   MOV     #2,RDCMD        ;LOAD READ FORWARD COMMAND
007562  004737  007744                  JSR     PC,READ         ;GO READ
007566  005737  000662          RSEX:   TST     EOTREC          ;BRANCH IF NOT AT EOT
007572  100002                          BPL     1$
007574  000137  004330                  JMP     REOT            ;ELSE GO TO REWIND
007600  000207                  1$:     RTS     PC              ;EXIT

007602  012737  010000  000562  RSFR:   MOV     #10000,RDCMD
007610  032777  000010  170772          BIT     #10,@SWR        ;SEE IF SHOULD READ FORWARD
007616  001013                          BNE     RSFR1           ;IF NOT: BR
007620  032777  000001  170762          BIT     #1,@SWR         ;SEE IF WRITE
007626  001002                          BNE     RSFR0           ;IF NOT: BR
007630  004737  011706                  JSR     PC,BKSP         ;GO BACKSPACE TO START
007634  012737  000002  000562  RSFR0:  MOV     #2,RDCMD        ;LOAD READ FORWARD COMMAND
007642  004737  007744                  JSR     PC,READ         ;GO READ FORWARD
007646  032777  000004  170734  RSFR1:  BIT     #4,@SWR         ;SEE IF SHOULD READ REVERSE
007654  001344                          BNE     RSEX            ;IF NOT: BR
007656  032737  010000  000562          BIT     #10000,RDCMD
007664  001005                          BNE     RSFR2           ;IF READ REVERSE: BR
007666  013737  000600  000670          MOV     TSTAL,STAL      ;DO READ STALL
```

```
3922  007674  004737  012132                  JSR     PC,STALL
3923  007700  012737  010000  000562  RSFR2:  MOV     #10000,RDCMD    ;LOAD READ REVERSE
3924  007706  004737  007744                  JSR     PC,READ         ;GO READ REVERSE
3925  007712  005737  000662                  TST     EOTREC          ;SEE IF AT END OF TAPE
3926  007716  100011                          BPL     RSFRX           ;IF NOT: BR
3927  007720  163737  000554  000662          SUB     RCNT,EOTREC
3928  007726  005437  000662                  NEG     EOTREC          ;SET TO PROPER RECORD NUMBER
3929  007732  005237  000662                  INC     EOTREC
3930  007736  000137  004330                  JMP     REOT            ;ELSE GO TO REWIND
3931  007742  000207                  RSFRX:  RTS     PC              ;EXIT
3932
```

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

007744  013700  000554          READ:   MOV     RCNT,R0         ;LOAD REC CNTR
007750  005737  000662                  TST     EOTREC          ;SEE IF EOT
007754  100013                        BPL     RDA             ;IF NOT: BR
007756  032737  010000  000562        BIT     #10000,RDCMD    ;SEE IF READ FORWARD
007764  001407                        BEQ     RDA             ;IF SO: BR
007766  042737  100000  000662        BIC     #100000,EOTREC  ;CLEAR FLAG
007774  013703  000662                MOV     EOTREC,R3       ;GET MODIFIED RECORD COUNT
010000  160300                        SUB     R3,R0           ;SET RECORD AT
010002  005200                        INC     R0              ;SET TO PROPER NUMBER OF RECORDS
010004  012737  024615  000654  RDA:  MOV     #MSG6,EMADDR    ;SET ERROR MSG ADDRESS
010012  005037  000700                CLR     TMFLG
010016  032737  010000  000562        BIT     #10000,RDCMD
010024  001406                        BEQ     RD0             ;IF READ FORWARD: BR
010026  005737  000564                TST     TMEX            ;SEE IF TM
010032  001403                        BEQ     RD0             ;IF NOT: BR
010034  005237  000700                INC     TMFLG           ;SET TM FLAG
010040  005200                        INC     R0
010042  013777  000556  170446  RD0:  MOV     FMCNT,@FC       ;LOAD CHAR CNTR
010050  012777  033460  170436        MOV     #RDATA,@BA      ;LOAD DATA ADDR
010056  032737  010000  000562        BIT     #10000,RDCMD    ;SEE IF READ REVERSE
010064  001417                        BEQ     RD1A            ;IF NOT: BR
010066  013703  000556                MOV     FMCNT,R3
010072  005103                        COM     R3
010074  032737  000020  000552        BIT     #20,UDES        ;SEE IF CORE DUMP
010102  001402                        BEQ     RD1             ;IF NOT: BR
010104  000241                        CLC
010106  006003                        ROR     R3              ;R3 = FC/2
010110  060377  170400          RD1:  ADD     R3,@BA          ;SET REVERSE BUS ADDRESS
010114  012737  000076  000674        MOV     #76,MTC1        ;SET READ REVERSE
010122  000403                        BR      RD1B
```

```
3990  010124  012737  000070  000674  RD1A:   MOV     #70,MTC1        ;SET READ FORWARD
3991  010132  012737  010144  000664  RD1B:   MOV     #RD2,RTRN       ;SET INTERRUPT RETURN ADDRESS
3992  010140  000137  021144          RD1D:   JMP     TAPG            ;GO EXECUTE TAPE COMMAND
3993  010144  032737  010000  000562  RD2:    BIT     #10000,RDCMD    ;SEE IF READ REVERSE
3994  010152  001014                          BNE     RD3             ;IF SO: BR
3995  010154  032777  002000  170340          BIT     #2000,@DS       ;SEE IF EOT
3996  010162  001410                          BEQ     RD3             ;IF NOT: BR
3997  010164  005737  000700                  TST     TMFLG           ;SEE IF TM
3998  010170  001005                          BNE     RD3             ;IF SO: BR
3999  010172  010037  000662                  MOV     R0,EOTREC
4000  010176  052737  100000  000662          BIS     #100000,EOTREC  ;SET EOT FLAG
4001  010204  032777  000002  170310  RD3:    BIT     #2,@DS          ;SEE IF AT LOAD POINT
4002  010212  001410                          BEQ     RD4             ;IF NOT: BR
4003  010214  004737  022672                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
4004  010220  012704  025016                  MOV     #MSG22,R4
4005  010224  000004                          TYPE                    ;TYPE MSG
4006  010226  000000                          HALT
4007  010230  000137  003160                  JMP     STARTA          ;RESTART
4008  010234  032777  004000  170346  RD4:    BIT     #4000,@SWR      ;SEE IF SHOULD CHECK ERRORS
4009  010242  001121                          BNE     RD5             ;IF NOT: BR
4010  010244  005737  000700                  TST     TMFLG
4011  010250  001472                          BEQ     RD4B            ;IF NO TM EXPT: BR
4012  010252  032777  000004  170242          BIT     #4,@DS
4013  010260  001024                          BNE     RD4A            ;IF TM RECVD: BR
4014  010262  012737  033460  021064          MOV     #RDATA,CADER    ;SAVE EXPT BUS ADDRESS
4015  010270  012737  000002  021072          MOV     #2,DRVER        ;SET TM STATUS ERROR FLAG
4016  010276  004737  020136                  JSR     PC,ERPT         ;GO PRINT TM ERROR
4017  010302  013704  000676                  MOV     UNP,R4
4018  010306  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
4019  010314  001403                          BEQ     1$              ;IF NOT: BR
4020  010316  005264  001150                  INC     RDERR1(R4)      ;BUMP READ REVERSE ERROR
4021  010322  000502                          BR      RD6
4022  010324  005264  001110          1$:     INC     RDER1(R4)       ;BUMP READ FORWARD ERROR
4023  010330  000477                          BR      RD6
4024  010332  012703  033460          RD4A:   MOV     #RDATA,R3
4025  010336  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
4026  010344  001007                          BNE     RD4A0           ;IF SO: BR
4027  010346  032737  002000  000552          BIT     #2000,UDES      ;SEE IF IN PE
4028  010354  001025                          BNE     RD4A2           ;IF SO: BR
4029  010356  062703  000002                  ADD     #2,R3
4030  010362  000422                          BR      RD4A2
4031  010364  013704  000556          RD4A0:  MOV     FMCNT,R4
4032  010370  005104                          COM     R4
4033  010372  032737  000020  000552          BIT     #20,UDES        ;SEE IF CORE DUMP
4034  010400  001402                          BEQ     RD4A1           ;IF NOT: BR
4035  010402  000241                          CLC
4036  010404  006004                          ROR     R4              ;SET TO FC/2
4037  010406  060403          RD4A1:  ADD     R4,R3           ;SET EXPT BUS ADDRESS
4038  010410  042703  000001                  BIC     #1,R3           ;MAKE EXPT ADDRESS EVEN
4039  010414  032737  002000  000552          BIT     #2000,UDES      ;SEE IF IN PE
4040  010422  001002                          BNE     RD4A2           ;IF SO: BR
4041  010424  162703  000002                  SUB     #2,R3
4042  010430  004737  017412          RD4A2:  JSR     PC,ER2
4043  010434  000402                          BR      RD4C
4044  010436  004737  017316          RD4B:   JSR     PC,ERCHK        ;GO CHECK ERRORS
4045  010442  005737  000710          RD4C:   TST     SERFL
```

```
4046  010446  001417                          BEQ     RD5             ;IF NO ERROR: BR
4047  010450  013704  000676                  MOV     UNP,R4
4048  010454  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
4049  010462  001003                          BNE     RD4D            ;IF SO: BR
4050  010464  005264  001110                  INC     RDER1(R4)       ;BUMP READ FORWARD ERROR
4051  010470  000402                          BR      RD4E
4052  010472  005264  001150          RD4D:   INC     RDERR1(R4)      ;BUMP READ REVERSE ERROR
4053  010476  004737  010700          RD4E:   JSR     PC,RDRTY        ;GO RETRY
4054  010502  005037  000714                  CLR     RTYFL           ;CLEAR RETRY FLAG
4055  010506  032777  020000  170074  RD5:    BIT     #20000,@SWR     ;SEE IF SHOULD DO DATA CHECK
4056  010514  001005                          BNE     RD6             ;IF NOT; BR
4057  010516  005737  000700                  TST     TMFLG
4058  010522  001002                          BNE     RD6
4059  010524  004737  015446                  JSR     PC,DCHK         ;GO CHECK DATA
4060  010530  005037  000710          RD6:    CLR     SERFL           ;CLEAR STATUS ERROR FLAG
4061  010534  004737  014272                  JSR     PC,DS3          ;CLEAR BUFFER
4062  010540  032777  000040  170042          BIT     #40,@SWR        ;SEE IF SHOULD YOZZLE
4063  010546  001402                          BEQ     RD7             ;IF NOT: BR
4064  010550  004737  011266                  JSR     PC,YOZ          ;ELSE GO YOZZLE
4065  010554  013737  000574  000670  RD7:    MOV     RSTAL,STAL      ;SET DELAY
4066  010562  004737  012132                  JSR     PC,STALL        ;STALL
4067  010566  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
4068  010574  001403                          BEQ     RD7A            ;IF NOT: BR
4069  010576  005037  000700                  CLR     TMFLG           ;CLEAR TAPE MARK FLAG
4070  010602  000405                          BR      RD10
4071  010604  005737  000662          RD7A:   TST     EOTREC          ;SEE IF EOT FOUND
4072  010610  100002                          BPL     RD10            ;IF NOT: BR
4073  010612  012700  000001                  MOV     #1,R0           ;SET TO EOT
4074  010616  005300                  RD10:   DEC     R0
4075  010620  001402                          BEQ     RD11            ;IF DONE ALL: BR
4076  010622  000137  010042                  JMP     RD0
4077  010626  032737  010000  000562  RD11:   BIT     #10000,RDCMD    ;SEE IF READ REVERSE
4078  010634  001016                          BNE     RDEX            ;IF SO: BR
4079  010636  005737  000662                  TST     EOTREC          ;SEE IF FOUND EOT
4080  010642  100413                          BMI     RDEX            ;IF SO: BR
4081  010644  005737  000564                  TST     TMEX            ;SEE IF TM EXPECTED
4082  010650  001410                          BEQ     RDEX            ;IF NOT: BR
4083  010652  005737  000700                  TST     TMFLG           ;SEE IF TM FOUND
4084  010656  001005                          BNE     RDEX            ;IF SO: BR
4085  010660  005237  000700                  INC     TMFLG           ;ELSE SET FLAG
4086  010664  005200                          INC     R0              ;SET RECORD COUNT TO ONE
4087  010666  000137  010042                  JMP     RD0             ;GO READ TM
4088  010672  005037  000700          RDEX:   CLR     TMFLG
4089  010676  000207                  RDX:    RTS     PC              ;EXIT
```

```
                                        ;*********************************************************
                                        ;READ ERROR RETRY SUBROUTINE:
                                        ;
                                        ;THIS SUBROUTINE WILL RETRY ALL DATA RELATED
                                        ;READ ERRORS UP TO EIGHT (8) TIMES.  IF ALL
                                        ;FOUR RETRIES ARE BAD, IT IS CONSIDERED
                                        ;A HARD ERROR.  IF ANY ARE GOOD, IT IS A
                                        ;SOFT ERROR.  RETRIES MAY BE INHIBITED
                                        ;VIA SWITCH FOUR (SW4=0: INHIBIT RETRIES)
                                        ;*********************************************************

010700  032777  000020  167702  RDRTY:  BIT     #20,@SWR        ;SEE IF RETRY INHIBITED
010706  001001                          BNE     RDRT0           ;IF NOT: BR
010710  000207                          RTS     PC              ;ELSE RETURN
010712  013703  000724          RDRT0:  MOV     ERSAV,R3
010716  042703  102700                  BIC     #102700,R3      ;MARK NON-RECOVERABLE ERROR BITS
010722  001410                          BEQ     RDRT1           ;IF NOT: BR
010724  004737  022672                  JSR     PC,PAPRT        ;PRINT HEADER
010730  012704  026573                  MOV     #MSG79,R4
010734  000004                          TYPE                    ;TYPE MSG
010736  004737  011252                  JSR     PC,NRTP         ;PRINT ER FOR NON-RETRYABLE ERROR
010742  000207          RDRT1A: RTS     PC                      ;RETURN
010744  032777  002000  167636  RDRT1:  BIT     #2000,@SWR      ;SEE IF PRINT INHIBITED
010752  001003                          BNE     RDRT1B          ;IF SO: BR
010754  012704  026246                  MOV     #MSG64,R4
010760  000004                          TYPE                    ;TYPE MSG
010762  005037  000704          RDRT1B: CLR     RTCNT           ;CLEAR RETRY COUNTER
010766  005037  000710          RDRTG:  CLR     SERFL           ;CLEAR STATUS ERROR FLAG
010772  012737  000002  000714          MOV     #2,RTYFL        ;SET READ RETRY FLAG
011000  004737  011266                  JSR     PC,YOZ          ;GO TO YOZZLE TO RETRY READ
011004  005737  000710                  TST     SERFL           ;SEE IF RETRY ERROR
011010  001031                          BNE     RDRT5           ;IF SO: BR
011012  032777  002000  167570          BIT     #2000,@SWR
011020  001011                          BNE     RDRT2
011022  012704  026737                  MOV     #MSG105,R4
011026  000004                          TYPE                    ;TYPE MSG
011030  012704  026270                  MOV     #MSG65,R4
011034  000004                          TYPE                    ;TYPE MSG
011036  013703  000704                  MOV     RTCNT,R3
011042  104400                          TYPOCT                  ;PRINT RETRY NUMBER
011044  013704  000676          RDRT2:  MOV     UNP,R4
011050  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
011056  001003                          BNE     RDRT3           ;IF SO: BR
011060  005264  002670                  INC     RFSOFT(R4)      ;ELSO BUMP FORWARD SOFT ERROR COUNTER
011064  000402                          BR      RDRT4
011066  005264  002710          RDRT3:  INC     RRSOFT(R4)      ;BUMP ERRORS SOFT CNTR
011072  000207          RDRT4:  RTS     PC                      ;RETURN
011074  013703  000724          RDRT5:  MOV     ERSAV,R3        ;GET ER
011100  005037  000650                  CLR     TEMP3           ;CLEAR RECOVERABLE ERROR INDICATOR
011104  042703  102700                  BIC     #102700,R3      ;MASK RECOVERABLE BITS
011110  001413                          BEQ     RDRT5A          ;IF RECOVERABLE. BR
011112  004737  022672                  JSR     PC,PAPRT        ;PRINT HEADER
011116  012704  026573                  MOV     #MSG79,R4
011122  000004                          TYPE                    ;TYPE MSG
011124  004737  011252                  JSR     PC,NRTP         ;PRINT ER
011130  012737  000001  000650          MOV     #1,TEMP3        ;SET FLAG
```

```
 4147  011136  000404                          BR      RDRT5B
 4148  011140  032777  002000  167442  RDRT5A: BIT     #2000,@SWR      ;SEE IF PRINT INHIBITED
 4149  011146  001014                          BNE     RDRT6           ;IF SO: BR
 4150  011150  012704  026270          RDRT5B: MOV     #MSG65,R4
 4151  011154  000004                          TYPF                    ;TYPE MSG
 4152  011156  013703  000704                  MOV     RTCNT,R3
 4153  011162  104400                          TYPOCT                  ;PRINT RETRY NUMBER
 4154  011164  005737  000650                  TST     TEMP3           ;SEE IF DID NON-RECOVERABLE
 4155  011170  001403                          BEQ     RDRT6           ;IF NOT: BR
 4156  011172  005037  000650                  CLR     TEMP3           ;CLEAR FLAG
 4157  011176  000207                          RTS     PC              ;EXIT
 4158  011200  005237  000704          RDRT6:  INC     RTCNT
 4159  011204  023737  000704  000604          CMP     RTCNT,RETRY     ;SEE IF DONE 8 RETRIES
 4160  011212  001265                          BNE     RDRTG           ;IF NOT: BR
 4161  011214  012704  027255                  MOV     #MSG115,R4
 4162  011220  000004                          TYPE                    ;TYPE MSG
 4163  011222  013704  000676                  MOV     UNP,R4
 4164  011226  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 4165  011234  001003                          BNE     RDRT7           ;IF SO: BR
 4166  011236  005264  002730                  INC     RFHARD(R4)      ;BUMP FORWARD HARD ERROR CNTR
 4167  011242  000402                          BR      RDRTX
 4168  011244  005264  002750          RDRT7:  INC     RRHARD(R4)      ;BUMP REVERSE HARD ERROR CNTR
 4169  011250  000207                  RDRTX:  RTS     PC              ;RETURN
 4170
 4171  011252  013703  000724          NRTP:   MOV     ERSAV,R3        ;GET ER REGISTER
 4172  011256  104400                          TYPOCT                  ;PRINT ER
 4173  011260  004737  021110                  JSR     PC,FRPRT        ;PRINT F OR R
 4174  011264  000207                          RTS     PC              ;RETURN
```

```
4176                                    ;*********************************************************
4177                                    ;YOZZLE SUBROUTINE:
4178                                    ;
4179                                    ;THIS SUBROUTINE, ENTERED VIA SWITCH FIVE (5), IS USED TO PERFORM
4180                                    ;A CONTINUOUS READ AND SPACE OVER OF THE CURRENT RECORD ON TAPE.
4181                                    ;FULL STATUS AND DATA CHECKING MAY BE PERFORMED
4182                                    ;OR NOT VIA CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13).
4183                                    ;A SOFTWARE DELAY IS PERFORMED BETWEEN EACH READ
4184                                    ;AND SPACE OPERATION AND MAY BE VARIED BY TYPING
4185                                    ;CNTRL C ON THE TTY AND ENTERING A VALUE IN RESPONSE
4186                                    ;TO THE PRINTED REQUEST.
4187                                    ;*********************************************************
4188
4189  011266  013737  000602  000670  YOZ:    MOV     YSTAL,STAL
4190  011274  004737  012132          YOZ0:   JSR     PC,STALL        ;DO YOZZLE STALL
4191  011300  012777  177777  167210          MOV     #-1,@FC         ;SET TO 1 RECORD SPACING
4192  011306  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
4193  011314  001404                          BEQ     YOZA            ;IF NOT: BR
4194  011316  112737  000030  000674          MOVB    #30,MTC1        ;SET TO SPACE FORWARD
4195  011324  000403                          BR      YOZB
4196  011326  112737  000032  000674  YOZA:   MOVB    #32,MTC1        ;SET TO SPACE REVERSE
4197  011334  012737  011354  000664  YOZB:   MOV     #YOZC,RTRN      ;SET RETURN ADDRESS
4198  011342  012737  177775  000670          MOV     #177775,STAL    ;SET TIME MULTIPLIER
4199  011350  000137  021144                  JMP     TAPG            ;GO YOZZLE
4200  011354  005737  000700          YOZC:   TST     TMFLG           ;SEE IF TM
4201  011360  001404                          BEQ     1$              ;IF NOT: BR
4202  011362  012737  040000  000670          MOV     #40000,STAL     ;SET TM STALL
4203  011370  000403                          BR      2$
4204  011372  013737  000602  000670  1$:     MOV     YSTAL,STAL
4205  011400  004737  012132          2$:     JSR     PC,STALL        ;DO YOZZLE STALL
4206  011404  012777  033460  167102          MOV     #RDATA,@BA      ;SET BUS ADDRESS
4207  011412  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
4208  011420  001416                          BEQ     YOZC1           ;IF NOT: BR
4209  011422  013703  000556                  MOV     FMCNT,R3
4210  011426  005103                          COM     R3
4211  011430  032737  000020  000552          BIT     #20,UDES        ;SEE IF CORE DUMP
4212  011436  001401                          BEQ     YOZC0           ;IF NOT: BR
4213  011440  006203                          ASR     R3              ;R3 = FC/2
4214  011442  060377  167046          YOZC0:  ADD     R3,@BA          ;SET REVERSE BUS ADDRESS
4215  011446  012737  000076  000674          MOV     #76,MTC1        ;SET READ REVERSE
4216  011454  000403                          BR      YOZC2
4217  011456  012737  000070  000674  YOZC1:  MOV     #70,MTC1        ;SET READ FORWARD
4218  011464  013777  000556  167024  YOZC2:  MOV     FMCNT,@FC       ;SET CHARACTER COUNT
4219  011472  012737  011504  000664          MOV     #YOZD,RTRN      ;SET RETURN ADDRESS
4220  011500  000137  021144                  JMP     TAPG            ;GO READ
4221  011504  032777  004000  167076  YOZD:   BIT     #4000,@SWR      ;SEE IF SHOULD CHECK ERRORS
4222  011512  001050                          BNE     YOZE            ;IF NOT: BR
4223  011514  005737  000700                  TST     TMFLG           ;SEE IF TAPE MARK TIME
4224  011520  001443                          BEQ     YOZD1           ;IF NOT: BR
4225  011522  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
4226  011530  001425                          BEQ     YOZD0           ;IF NOT: BR
4227  011532  012703  033460                  MOV     #RDATA,R3
4228  011536  013704  000556                  MOV     FMCNT,R4
4229  011542  005104                          COM     R4
4230  011544  032737  000020  000552          BIT     #20,UDES        ;SEE IF CORE DUMP
4231  011552  001401                          BEQ     YOZD4           ;IF NOT: BR
```

```
4232  011554  006204                          ASR     R4              ;SET TO FC/2
4233  011556  060403                  YOZD4:  ADD     R4,R3           ;SET EXPT BUS ADDRESS
4234  011560  042703  000001                  BIC     #1,R3           ;MAKE EXPT ADDRESS EVEN
4235  011564  032737  002000  000552          BIT     #2000,UDES      ;SEE IF PE
4236  011572  001001                          BNE     YOZD2           ;IF SO: BR
4237  011574  005743                          TST     -(R3)           ;SET EXPT BA
4238  011576  004737  017412          YOZD2:  JSR     PC,ER2          ;GO CHECK ERRORS
4239  011602  000430                          BR      YOZF
4240  011604  012703  033460          YOZD0:  MOV     #RDATA,R3
4241  011610  032737  002000  000552          BIT     #2000,UDES      ;SEE IF PE
4242  011616  001001                          BNE     YOZD3           ;IF SO: BR
4243  011620  005723                          TST     (R3)+           ;SET EXPT BA
4244  011622  004737  017412          YOZD3:  JSR     PC,ER2          ;GO CHECK ERRORS
4245  011626  000416                          BR      YOZF
4246  011630  004737  017316          YOZD1:  JSR     PC,ERCHK        ;ELSE GO CHECK ERRORS
4247  011634  005737  000714          YOZE:   TST     RTYFL           ;SEE IF RETRY
4248  011640  001013                          BNE     YOZG            ;IF SO: BR
4249  011642  032777  020000  166740          BIT     #20000,@SWR     ;SEE IF SHOULD CHECK DATA
4250  011650  001005                          BNE     YOZF            ;IF NOT: BR
4251  011652  005737  000700                  TST     TMFLG           ;SEE IF TAPE MARK
4252  011656  001002                          BNE     YOZF            ;IF SO: BR
4253  011660  004737  015446                  JSR     PC,DCHK         ;ELSE GO CHECK DATA
4254  011664  004737  014272          YOZF:   JSR     PC,DS3          ;GO CLEAR DATA AREA
4255  011670  032777  000040  166712  YOZG:   BIT     #40,@SWR        ;SEE IF SHOULD CONTINUE YOZZLE
4256  011676  001402                          BEQ     YOZH            ;IF NOT: BR
4257  011700  000137  011300                  JMP     YOZO
4258  011704  000207                  YOZH:   RTS     PC              ;EXIT
4259
```

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

011706  013737  000600  000670  BKSP:   MOV     TSTAL,STAL
011714  004737  012132                  JSR     PC,STALL        ;DO TURN AROUND STALL
011720  012737  024645  000654          MOV     #MSG10,EMADDR
011726  012777  033460  166560          MOV     #RDATA,@BA
011734  005737  000564                  TST     TMEX            ;SEE IF TM
011740  001440                          BEQ     B0              ;IF NOT: BR
011742  012777  177777  166546          MOV     #-1,@FC
011750  012737  000032  000674          MOV     #32,MTC1
011756  012737  011770  000664          MOV     #BKTM,RTRN
011764  000137  021144                  JMP     TAPG            ;SPACE TO TM
011770  032777  010000  166612  BKTM:   BIT     #10000,@SWR     ;SEE IF SHOULD CHECK ERROR
011776  001021                          BNE     B0              ;IF NOT: BR
012000  012737  026162  000654          MOV     #MSG55,EMADDR
012006  032777  000004  166506          BIT     #4,@DS          ;SEE IF TM
012014  001006                          BNE     BKTM0           ;IF SO: BR
012016  012737  033460  021064          MOV     #RDATA,CADER
012024  004737  020136                  JSR     PC,ERPT         ;PRINT ERROR
012030  000404                          BR      B0
012032  012703  033460          BKTM0:  MOV     #RDATA,R3
012036  004737  017412                  JSR     PC,ER2
012042  013700  000554          B0:     MOV     RCNT,R0
012046  005400                          NEG     R0              ;BUILD SPACE AMOUNT
012050  012737  024645  000654          MOV     #MSG10,EMADDR   ;SET ERROR MESG ADDRESS
012056  010077  166434                  MOV     R0,@FC
012062  012737  000032  000674  BKRT:   MOV     #32,MTC1        ;SET SPACE REVERSE
012070  012737  012106  000664          MOV     #B1,RTRN        ;SET RETURN ADDRESS
012076  010037  000670                  MOV     R0,STAL         ;SET INTERRUPT TIME MULTIPLIER
012102  000137  021144                  JMP     TAPG            ;GO DO SPACE
012106  012703  033460          B1:     MOV     #RDATA,R3
012112  004737  017412                  JSR     PC,ER2
012116  013737  000600  000670  B2:     MOV     TSTAL,STAL      ;DO STALL
012124  004737  012132                  JSR     PC,STALL        ;STALL
012130  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.
                                        ;*********************************************************

012132  005337  000670        STALL:  DEC     STAL
012136  001375                        BNE     STALL           ;DELAY
012140  000207                        RTS     PC              ;EXIT
```

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

012142  012701  177760          CCNTR:  MOV     #-20,R1         ;SET HIGH LIMIT
012146  012702  175000                  MOV     #-3000,R2       ;SET LOW LIMIT
012152  004737  023254                  JSR     PC,RANG         ;GO GENERATE NUMBER
012156  042737  000001  000630          BIC     #1,RANSAV
012164  013737  000630  000556          MOV     RANSAV,FMCNT    ;SET CHAR COUNT
012172  012737  177777  014330          MOV     #-1,PATS        ;PRESET DATA PATTERN
012200  000207                          RTS     PC              ;EXIT

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

012202  012702  000001          RCNTR:  MOV     #1,R2           ;SET LOW LIMIT
012206  012701  000500                  MOV     #500,R1         ;SET HIGH LIMIT
012212  004737  023254                  JSR     PC,RANG         ;GO GENERATE NUMBER
012216  013737  000630  000554          MOV     RANSAV,RCNT     ;SET RECORD COUNT
012224  000207                          RTS     PC              ;EXIT
```

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

012226  005737  000636        TINP:   TST     TINF            ;SEE IF SHOULD INPUT FROM TTY
012232  001002                        BNE     1$              ;IF SO. BR
012234  000137  013714                JMP     TINP4           ;GET SWITCHES
012240  005037  000676        1$:     CLR     UNP             ;CLEAR TABLE POINTER
012244  005037  005054                CLR     REOTC           ;CLEAR EOT UNIT COUNTER
012250  012700  000010                MOV     #10,R0          ;SET SIZE OF TABLE
012254  012701  000746                MOV     #UN1,R1         ;SET START OF TABLE
012260  005021                3$:     CLR     (R1)+           ;CLEAR TABLE
012262  005300                        DEC     R0              ;SEE IF DONE
012264  001375                        BNE     3$              ;IF NOT: BR
```

```
4431  012266  012704  025306                MOV     #MSG31,R4
4432  012272  005737  000736                TST     ASEQF           ;SEE IF ALTO SEQ
4433  012276  001402                        BEQ     4$              ;IF NOT: BR
4434  012300  012704  025241                MOV     #MSG30,R4       ;SET AUTO SEQ HDR
4435  012304  010446              4$:       MOV     R4,-(SP)        ;SAVE ADDRESS OF MESSAGE
4436  012306  000004                        TYPE                    ;TYPE MSG
4437  012310  105036                        CLRB    @(SP)+          ;DO NOT TYPE TITLE ON RESTART
4438  012312  012704  025363                MOV     #MSG31A,R4      ;TYPE INSTRUCTION
4439  012316  000004                        TYPE
4440  012320  105037  025363                CLRB    MSG31A          ;DO NOT TYPE STARTUP INSTRUCTIONS ON RESTART
4441  012324  005737  014076                TST     SCVFL           ;SEE IF SHORT CONVERSATION
4442  012330  001067                        BNE     6$              ;IF SO: BR
4443  012332  012704  026452                MOV     #MSG74,R4
4444  012336  000004                        TYPE                    ;TYPE MSG
4445  012340  013703  000544                MOV     REGS,R3
4446  012344  104400                        TYPOCT                  ;PRINT CURRENT REG START
4447  012346  012705  000544                MOV     #REGS,R5        ;SAVE ADDRESS LOCATION
4448  012352  012701  000007                MOV     #7,R1           ;SET SIZE OF ENTRY
4449  012356  012702  176400                MOV     #176400,R2      ;SET UPPER LIMIT
4450  012362  012703  172300                MOV     #172300,R3      ;SET LOWER LIMIT
4451  012366  004737  023442                JSR     PC,TTR          ;GO GET RESPONSE
4452  012372  012704  026475                MOV     #MSG75,R4
4453  012376  000004                        TYPE                    ;TYPE MSG
4454  012400  013703  000546                MOV     VECT,R3
4455  012404  104400                        TYPOCT                  ;PRINT CURRENT VECTOR
4456  012406  012705  000546                MOV     #VECT,R5        ;SET SAVE LOCATION
4457  012412  012701  000004                MOV     #4,R1           ;SET SIZE OF ENTRY
4458  012416  012702  000224                MOV     #224,R2         ;SET UPPER LIMIT
4459  012422  012703  000150                MOV     #150,R3         ;SET LOWER LIMIT
4460  012426  004737  023442                JSR     PC,TTR          ;GO GET RESPONSE
4461  012432  013700  000546                MOV     VECT,R0         ;GET VECTOR ADDRESS
4462  012436  012720  022006                MOV     #MTINT,(R0)+    ;LOAD VECTOR WITH HANDLER ADDRESS
4463  012442  012710  000340                MOV     #340,(R0)       ;LOAD PRIORITY LEVEL
4464  012446  013700  000544                MOV     REGS,R0         ;GET STARTING REGISTER ADDRESS
4465  012452  012701  000016                MOV     #16,R1          ;SET NUMBER OF REGISTERS
4466  012456  012702  000510                MOV     #C1,R2          ;GET FIRST ADDRESS LOCATION
4467  012462  010022              5$:       MOV     R0,(R2)+        ;BUILD TABLE OF ADDRESSES
4468  012464  062700  000002                ADD     #2,R0           ;BUMP ADDRESS
4469  012470  005301                        DEC     R1              ;SEE IF DONE
4470  012472  001373                        BNE     5$              ;IF NOT: BR
4471  012474  005737  000736                TST     ASEQF           ;SEE IF AUTO SEQ
4472  012500  001403                        BEQ     6$              ;IF NOT: BR
4473  012502  005726                        TST     (SP)+           ;RESET STACK POINTER
4474  012504  000137  022024                JMP     ASEQ            ;GO TO AUTO SEQUENCE
4475  012510  012777  000040  166002  6$:   MOV     #40,@CS         ;INITIALIZE
4476  012516  012704  026117                MOV     #MSG52,R4
4477  012522  000004                        TYPE                    ;TYPE MSG
4478  012524  012705  000550                MOV     #DVN,R5         ;GET ADDRESS
4479  012530  012701  000002                MOV     #2,R1           ;SET SIZE OF RESPONSE
4480  012534  012702  000007                MOV     #7,R2           ;SET UPPER LIMIT
4481  012540  012703  000000                MOV     #0,R3           ;SET LOWER LIMIT
4482  012544  004737  023442                JSR     PC,TTR          ;GO GET DRIVE NUMBER
4483  012550  013777  000550  165742        MOV     DVN,@CS
4484  012556  005777  165726                TST     @C1             ;ACCESS DRIVE
4485  012562  032777  010000  165730        BIT     #10000,@CS      ;SEE IF NED
4486  012570  001411                        BEQ     TINP0           ;IF NOT: BR
```

```
4487  012572  012704  026407            MOV     #MSG71,R4
4488  012576  000004                    TYPE                    ;TYPE MSG
4489  012600  013704  000510            MOV     C1,R4
4490  012604  005204                    INC     R4
4491  012606  152714  000100            BISB    #100,(R4)       ;CLEAR TRE
4492  012612  000736                    BR      6$              ;RETRY DVN
4493  012614  012704  025450  TINP0:   MOV     #MSG32,R4
4494  012620  000004                    TYPE                    ;TYPE MSG
4495  012622  005037  000646            CLR     TEMP2           ;CLEAR BUFFER
4496  012626  012705  000646            MOV     #TEMP2,R5       ;SET UNIT DESCRIPTION BUFFER ADDRESS
4497  012632  012701  000002            MOV     #2,R1           ;SET NUMBER OF CHARACTERS TO INPUT
4498  012636  012702  000007            MOV     #7,R2           ;SET MAXIMUM LIMIT
4499  012642  012703  000000            MOV     #0,R3           ;SET MINIMUM LIMIT
4500  012646  004737  023442            JSR     PC,TTR          ;GO GET UNIT NUMBER
4501  012652  005737  000644            TST     TEMP1           ;SEE IF HAVE NEW PARAMETER
4502  012656  001012                    BNE     TINP0B          ;IF SO: BR
4503  012660  005737  000676            TST     UNP             ;SEE IF FIRST ENTRY
4504  012664  001753                    BEQ     TINP0
4505  012666  013700  000676            MOV     UNP,R0
4506  012672  012760  177777  000746    MOV     #-1,UN1(R0)     ;SET END UNIT TABLE
4507  012700  000137  013300            JMP     TINP2C          ;GO GET RECORD COUNT
4508  012704  013700  000676  TINP0B:   MOV     UNP,R0
4509  012710  042760  000007  000746    BIC     #7,UN1(R0)      ;CLEAR UNIT NUMBER
4510  012716  004737  014110            JSR     PC,TPOS1        ;GO LOAD UNIT NUMBER TO PROPER POSITION
4511  012722  012777  000040  165570    MOV     #40,@CS
4512  012730  013777  000550  165562    MOV     DVN,@CS
4513  012736  016077  000746  165576    MOV     UN1(R0),@TC     ;LOAD UNIT NUMBER
4514  012744  032777  002000  165564  TINP0C: BIT     #2000,@DT       ;SEE  IF SLAVE PRESENT
4515  012752  001004                    BNE     TINP0D          ;IF SO: BR
4516  012754  012704  026175            MOV     #MSG57,R4
4517  012760  000004                    TYPE                    ;TYPE MSG
4518  012762  000714                    BR      TINP0           ;REDO
4519  012764  017703  165546  TINP0D:   MOV     @DT,R3          ;GET CONTENTS OF DT REG
4520  012770  042703  000007            BIC     #7,R3           ;CLEAR DRIVE TYPE #
4521  012774  022703  142050            CMP     #142050,R3      ;SEE IF 9TRK TM03,TU45
4522  013000  001410                    BEQ     TINP0E          ;IF SO: BR
4523  013002  012704  026072            MOV     #MSG50,R4       ;ILLEGAL DRIVE TYPE
4524  013006  000004                    TYPE                    ;TYPE MSG
4525  013010  017703  165522            MOV     @DT,R3
4526  013014  042703  000007            BIC     #7,R3           ;CLEAR SLAVE #
4527  013020  104400                    TYPOCT                  ;PRINT DRIVE TYPE REGISTER
4528  013022  012704  024637  TINP0E:   MOV     #MSG9,R4
4529  013026  000004                    TYPE                    ;TYPE MSG
4530  013030  017703  165504            MOV     @SN,R3
4531  013034  004737  024442            JSR     PC,SNPT         ;PRINT SERIAL NUMBER
4532  013040  012704  025471  TINP1:   MOV     #MSG33,R4
4533  013044  000004                    TYPE                    ;TYPE MSG
4534  013046  005037  000646            CLR     TEMP2           ;CLEAR BUFFER
4535  013052  012701  000002            MOV     #2,R1           ;SET NUMBER OF CHARACTERS TO INPUT
4536  013056  012702  000004            MOV     #4,R2           ;SET MAXIMUM LIMIT
4537  013062  012703  000003            MOV     #3,R3           ;SET MINIMUM LIMIT
4538  013066  004737  023442            JSR     PC,TTR          ;GO GET DENSITY
4539  013072  005737  000644            TST     TEMP1           ;SEE IF HAVE NEW PARAMETER
4540  013076  001407                    BEQ     TINP2           ;IF NOT: BR
4541  013100  042737  003400  000552    BIC     #3400,UDES      ;ELSE CLEAR OLD PARAMETER
4542  013106  012703  000010            MOV     #10,R3          ;SET POSITION FACTOR
```

```
4543  013112  004737  014100              JSR     PC,TPOS         ;GO LOAD DENSITY INTO PROPER POSITION
4544  013116  012704  025505      TINP2:  MOV     #MSG34,R4
4545  013122  000004                      TYPE                    ;TYPE MSG
4546  013124  005037  000646              CLR     TEMP2           ;CLR BUFFER
4547  013130  012701  000002              MOV     #2,R1           ;SET NUMBER OF CHARACTERS TO INPUT
4548  013134  012702  000001              MOV     #1,R2           ;SET MAXIMUM LIMIT
4549  013140  012703  000000              MOV     #0,R3           ;SET MINIMUM LIMIT
4550  013144  004737  023442              JSR     PC,TTR          ;GO INPUT PARITY
4551  013150  005737  000644              TST     TEMP1           ;SEE IF HAVE NEW PARAMETER
4552  013154  001407                      BEQ     TINP2A          ;IF NOT: BR
4553  013156  042737  000010  000552      BIC     #10,UDES        ;ELSE CLEAR OLD PARAMETER
4554  013164  012703  000003              MOV     #3,R3           ;SET POSITION FACTOR
4555  013170  004737  014100              JSR     PC,TPOS         ;GO LOAD PARITY TO PROPER POSITION
4556  013174  012704  026140      TINP2A: MOV     #MSG53,R4
4557  013200  000004                      TYPE                    ;TYPE MSG
4558  013202  005037  000646              CLR     TEMP2
4559  013206  012701  000003              MOV     #3,R1
4560  013212  012702  000017              MOV     #17,R2
4561  013216  012703  000000              MOV     #0,R3
4562  013222  004737  023442              JSR     PC,TTR          ;GO GET FORMAT
4563  013226  005737  000644              TST     TEMP1           ;SEE IF NEW PARAMETER
4564  013232  001407                      BEQ     TINP2B          ;IF NOT: BR
4565  013234  042737  000170  000552      BIC     #170,UDES
4566  013242  012703  000004              MOV     #4,R3
4567  013246  004737  014100              JSR     PC,TPOS
4568  013252  005237  005054      TINP2B: INC     REOTC           ;BUMP EOT UNIT COUNTER
4569  013256  022737  000016  000676      CMP     #16,UNP         ;SEE IF DONE UNITS
4570  013264  001405                      BEQ     TINP2C          ;IF SO: BR
4571  013266  062737  000002  000676      ADD     #2,UNP          ;POINT TO NEXT UNIT
4572  013274  000137  012614              JMP     TINP0           ;ELSE LOOK FOR NEXT UNIT
4573  013300  005037  000676      TINP2C: CLR     UNP             ;CLEAR UNIT POINTER
4574  013304  013700  005054              MOV     REOTC,R0
4575  013310  000337  005054              SWAB    REOTC
4576  013314  110037  005054              MOVB    R0,REOTC        ;SET UNIT EOT COUNTER
4577  013320  012704  025520      TINP3:  MOV     #MSG35,R4
4578  013324  000004                      TYPE                    ;TYPE MSG
4579  013326  013703  000554              MOV     RCNT,R3
4580  013332  104400                      TYPOCT                  ;PRINT RECORD COUNT
4581  013334  012705  000554              MOV     #RCNT,R5        ;SET RECORD COUNT ADDRESS
4582  013340  012701  000007              MOV     #7,R1           ;SET NUMBER OF CHARACTERS TO INPUT
4583  013344  012702  177777              MOV     #-1,R2          ;SET MAXIMUM LIMIT
4584  013350  012703  000001              MOV     #1,R3           ;SET MINIMUM LIMIT
4585  013354  004737  023442              JSR     PC,TTR          ;GO GET RECORD COUNT
4586  013360  013737  000554  000632      MOV     RCNT,RCSAV      ;SAVE RECORD COUNT
4587  013366  012704  025541              MOV     #MSG36,R4
4588  013372  000004                      TYPE                    ;TYPE MSG
4589  013374  005437  000556              NEG     FMCNT
4590  013400  013703  000556              MOV     FMCNT,R3
4591  013404  104400                      TYPOCT                  ;PRINT CHAR COUNT
4592  013406  012705  000556              MOV     #FMCNT,R5       ;SET CHARACTER COUNT ADDRESS
4593  013412  012701  000007              MOV     #7,R1           ;SET NUMBER OF CHARACTERS TO INPUT
4594  013416  012702  004000              MOV     #4000,R2        ;SET MAXIMUM LIMIT
4595  013422  012703  000004              MOV     #4,R3           ;SET MINIMUM LIMIT
4596  013426  004737  023442              JSR     PC,TTR          ;GO GET CHARACTER COUNT
4597  013432  005437  000556              NEG     FMCNT           ;SET TO TWO'S COMPLIMENT
4598  013436  013737  000556  000634      MOV     FMCNT,FCSAV     ;SAVE FRAME COUNT
```

```
4599  013444  012704  025560          MOV     #MSG37,R4       ;PRINT PATTERN NUMBER REQUEST
4600  013450  000004                  TYPE                    ;TYPE MSG
4601  013452  013703  000560          MOV     PATRN,R3
4602  013456  104400                  TYPOCT                  ;PRINT PATTERN
4603  013460  005037  014474          CLR     DOFL            ;CLEAR EXTERNAL DATA FLAG
4604  013464  012705  000560          MOV     #PATRN,R5       ;SET PATTERN NUMBER ADDRESS
4605  013470  012701  000003          MOV     #3,R1           ;SET NUMBER OF CHARACTERS TO INPUT
4606  013474  012702  000015          MOV     #15,R2          ;SET MAXIMUM LIMIT
4607  013500  012703  000000          MOV     #0,R3           ;SET MINIMUM LIMIT
4608  013504  004737  023442          JSR     PC,TTR          ;GO GET PATTERN NUMBER
4609  013510  012704  026335          MOV     #MSG69,R4
4610  013514  000004                  TYPE                    ;TYPE MSG
4611  013516  013703  000564          MOV     TMEX,R3
4612  013522  104400                  TYPOCT                  ;PRINT CURRENT TM FLAG SETTING
4613  013524  012705  000564          MOV     #TMEX,R5        ;GET TM FLAG ADDRESS
4614  013530  012701  000002          MOV     #2,R1           ;SET SIZE OF RESPONSE
4615  013534  012702  000001          MOV     #1,R2           ;SET UPPER LIMIT
4616  013540  012703  000000          MOV     #0,R3           ;SET LOWER LIMIT
4617  013544  004737  023442          JSR     PC,TTR          ;TM 1=YES
4618  013550  012704  024771          MOV     #MSG21,R4
4619  013554  000004                  TYPE                    ;TYPE MSG
4620  013556  013703  000570          MOV     INTRF,R3
4621  013562  104400                  TYPOCT                  ;PRINT CURRENT SETTING
4622  013564  012705  000570          MOV     #INTRF,R5       ;GET FLAG ADDRESS
4623  013570  012701  000002          MOV     #2,R1           ;SET SIZE OF RESPONSE
4624  013574  012702  000001          MOV     #1,R2           ;SET UPPER LIMIT
4625  013600  012703  000000          MOV     #0,R3           ;SET LOWER LIMIT
4626  013604  004737  023442          JSR     PC,TTR          ;GO GET RESPONSE
4627  013610  012704  025603          MOV     #MSG38,R4
4628  013614  000004                  TYPE                    ;TYPE MSG
4629  013616  013703  000572          MOV     SPFLG,R3
4630  013622  104400                  TYPOCT                  ;PRINT CURRENT SETTING
4631  013624  012705  000572          MOV     #SPFLG,R5       ;SET ADDRESS OF FLAG
4632  013630  012701  000002          MOV     #2,R1           ;SET SIZE OF RESPONSE
4633  013634  012702  000001          MOV     #1,R2           ;SET UPPER LIMIT
4634  013640  012703  000000          MOV     #0,R3           ;SET LOWER LIMIT
4635  013644  004737  023442          JSR     PC,TTR          ;GO GET RESPONSE
4636  013650  012704  025623  TINP3A: MOV     #MSG39,R4
4637  013654  000004                  TYPE                    ;TYPE MSG
4638  013656  013703  000566          MOV     CRCC,R3
4639  013662  104400                  TYPOCT
4640  013664  012705  000566          MOV     #CRCC,R5
4641  013670  012701  000002          MOV     #2,R1
4642  013674  012702  000001          MOV     #1,R2
4643  013700  012703  000000          MOV     #0,R3
4644  013704  004737  023442          JSR     PC,TTR
4645  013710  004737  023306          JSR     PC,GTSWR        ;GET SWITCHES
4646  013714  005737  014076  TINP4:  TST     SCVFL           ;BRANCH IF SHORT CONVERSATION
4647  013720  001063                  BNE     TINPX
4648  013722  005737  000636  1$:     TST     TINF            ;BRANCH IF NO TTY INPUT
4649  013726  001460                  BEQ     TINPX
4650  013730  012704  025663          MOV     #MSG40,R4
4651  013734  000004                  TYPE                    ;TYPE MSG
4652  013736  013703  000574          MOV     RSTAL,R3
4653  013742  104400                  TYPOCT                  ;PRINT READ STALL
4654  013744  012705  000574          MOV     #RSTAL,R5       ;SET READ STALL ADDRESS
```

```
 4655  013750  012701  000007                MOV     #7,R1           ;SET NUMBER OF CHARACTERS TO INPUT
 4656  013754  012702  177777                MOV     #-1,R2          ;SET MAXIMUM LIMIT
 4657  013760  012703  000001                MOV     #1,R3           ;SET MINIMUM LIMIT
 4658  013764  004737  023442                JSR     PC,TTR          ;GO GET READ STALL
 4659  013770  012704  025712                MOV     #MSG41,R4
 4660  013774  000004                        TYPE                    ;TYPE MSG
 4661  013776  013703  000576                MOV     WSTAL,R3
 4662  014002  104400                        TYPOCT                  ;PRINT READ STALL
 4663  014004  012705  000576                MOV     #WSTAL,R5       ;SET WRITE STALL ADDRESS
 4664  014010  012701  000007                MOV     #7,R1           ;SET NUMBER OF CHARACTERS TO INPUT
 4665  014014  012702  177777                MOV     #-1,R2          ;SET MAXIMUM LIMIT
 4666  014020  012703  000001                MOV     #1,R3           ;SET MINIMUM LIMIT
 4667  014024  004737  023442                JSR     PC,TTR          ;GO GET WRITE STALL
 4668  014030  012704  025724                MOV     #MSG42,R4
 4669  014034  000004                        TYPE                    ;TYPE MSG
 4670  014036  013703  000600                MOV     TSTAL,R3
 4671  014042  104400                        TYPOCT                  ;PRINT TA STALL
 4672  014044  012705  000600                MOV     #TSTAL,R5       ;SET TURN AROUND STALL ADDRESS
 4673  014050  012701  000007                MOV     #7,R1           ;SET NUMBER OF CHARACTERS TO INPUT
 4674  014054  012702  177777                MOV     #-1,R2          ;SET MAXIMUM LIMIT
 4675  014060  012703  000001                MOV     #1,R3           ;SET MINIMUM LIMIT
 4676  014064  004737  023442                JSR     PC,TTR          ;GO GET TURN AROUND STALL
 4677  014070  005037  014076        TINPX:  CLR     SCVFL           ;CLEAR SHORT CONVERSATION FLAG
 4678  014074  000207                        RTS     PC              ;EXIT
 4679  014076  000000                SCVFL:  0                       ;SHORT CONVERSATION FLAG
 4680
 4681                                        ;UNIT DESCRIPTION POSITIONING SUBROUTINE**********
 4682
 4683  014100  006337  000646        TPOS:   ASL     TEMP2           ;POSITION CHARACTER
 4684  014104  005303                        DEC     R3              ;SEE IF DONE
 4685  014106  001374                        BNE     TPOS            ;IF NOT: BR
 4686  014110  013700  000676        TPOS1:  MOV     UNP,R0          ;LOAD UNIT POINTER
 4687  014114  053760  000646  000746        BIS     TEMP2,UN1(R0)   ;LOAD CHARACTER INTO UN1(R0)
 4688  014122  000207                        RTS     PC              ;EXIT
 4689
```

```
                                ;*********************************************************
                                ;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. (CZTUTA0)
                                ;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.
                                ;*********************************************************

014124  005737  015064        DSUP:   TST     RDFL            ;SEE IF DID RANDOM DATA
014130  001056                        BNE     DS2A            ;IF NOT: BR
014132  005737  000736        DS0:    TST     ASEQF           ;SEE IF AUTO SEQ
014136  001412                        BEQ     DS0C            ;IF NOT: BR
014140  005737  000560                TST     PATRN           ;SEE IF AUTO RANDOM
014144  100007                        BPL     DS0C            ;IF NOT: BR
014146  004737  015022                JSR     PC,DATR         ;ELSE GO GENERATE RANDOM DATA
014152  005037  015064                CLR     RDFL            ;RESET RANDOM DATA FLAG
014156  004737  015066                JSR     PC,CRCLRC       ;GO GENERATE EXPT CRC/LRC
014162  000207                        RTS     PC              ;RETURN
014164  023737  000560  014330  DS0C:   CMP     PATRN,PATS      ;SEE IF NEW PATTERN
014172  001^20                        BNE     DS0A            ;IF SO: BR
014174  013,03  000552                MOV     UDES,R3         ;GET UNIT DESCRIPTION
014200  042703  175767                BIC     #175767,R3      ;MASK EVEN PARITY :DEN 2 BIT
014204  023703  014332                CMP     PARS,R3         ;SEE IF SAME AS LAST TIME
014210  001403                        BEQ     DS0B            ;IF SO: BR
014212  010337  014332                MOV     R3,PARS         ;SAVE PARITY
014216  000403                        BR      TWO
014220  005737  014334        DS0B:   TST     CLF             ;EXPT CRC/LRC DONE?
014224  001002                        BNE     ONE             ;IF SO :BR
014226  004737  015066        TWO:    JSR     PC,CRCLRC       ;GO GENERATE EXPT CRC/LRC
014232  000207                ONE:    RTS     PC
014234  005037  014334        DS0A:   CLR     CLF             ;CLEAR EXPT CRC/LRC CAL.DONE FLAG
014240  012703  027452                MOV     #WDATA,R3       ;R3 = ADDRS OF WRITE BUFFER
014244  013701  000560                MOV     PATRN,R1        ;R1 = PATTERN SELECTOR
014250  010137  014330                MOV     R1,PATS
014254  062701  000001                ADD     #1,R1           ;BUMP POINTER
014260  006301                        ASL     R1              ;MAKE PATTERN SELECTOR EVEN
014262  004771  002770                JSR     PC,@DATBL(R1)   ;GO GENERATE PATTERN
014266  004737  015066        DS2A:   JSR     PC,CRCLRC       ;GO GENERATE EXPT CRC/LRC
014272  013702  000556        DS3:    MOV     FMCNT,R2        ;R2=BUFFER SIZE
014276  006202                        ASR     R2              ;R2=FRAME CMT/2
014300  012701  033460                MOV     #RDATA,R1       ;R1=READ DATA START
014304  005021                DS4:    CLR     (R1)+           ;CLEAR BUFFER
014306  005202                        INC     R2              ;SEE IF DONE ALL
014310  001375                        BNE     DS4             ;IF NOT: BR
014312  013737  000552  014332        MOV     UDES,PARS       ;GET UNIT DESCRIPTION
014320  042737  175767  014332        BIC     #175767,PARS    ;MASK PARITY :DEN 2 BIT
```

L 6

SEQ 0076

```
 4747  014326  000207                    RTS     PC              ;EXIT
 4748  014330  177777            PATS:   -1                      ;PATTERN NUMBER SAVE
 4749  014332  000000            PARS:   0
 4750  014334  000000            CLF:    0
 4751
 4752
```

```
4754
4755                                        ;EXTERNAL DATA INPUT FROM H/S READER (256 CHARACTER MAXIMUM)
4756
4757  014336  005737  014474        DATO:   TST     DOFL            ;SEE IF SHOULD DO EXTERNAL INPUT
4758  014342  001351                        BNE     DS2A            ;IF NOT: BR
4759  014344  012737  000001  014474        MOV     #1,DOFL         ;SET EXTERNAL FLAG
4760  014352  005077  164244                CLR     @PRS            ;CLEAR READER STATUS
4761  014356  005037  000644                CLR     TEMP1           ;CLEAR FOR USE AS CHARACTER FLAG
4762  014362  052777  000001  164232  DATOA:  BIS     #1,@PRS         ;START READER
4763  014370  105777  164226        DATOB:  TSTB    @PRS            ;SEE IF DONE
4764  014374  100375                        BPL     DATOB           ;IF NOT : BR
4765  014376  005001                        CLR     R1              ;CLEAR SAVE LOCATION
4766  014400  117701  164220                MOVB    @PRB,R1         ;SAVE CHARACTER
4767  014404  005737  000644                TST     TEMP1           ;SEE IF HAVE FOUND START CHARACTER
4768  014410  001011                        BNE     DATOC           ;IF SO : BR
4769  014412  105701                        TSTB    R1              ;SEE IF CHARACTER IS 0
4770  014414  001762                        BEQ     DATOA           ;IF SO : BR
4771  014416  012737  000001  000644        MOV     #1,TEMP1        ;ELSE SET CHARACTER FOUND FLAG
4772  014424  010137  000646                MOV     R1,TEMP2        ;SAVE DATA SIZE
4773  014430  010102                        MOV     R1,R2           ;SAVE DATA SIZE
4774  014432  000753                        BR      DATOA           ;GO GET FIRST DATA CHAR
4775  014434  110123        DATOC:  MOVB    R1,(R3)+        ;LOAD BUFFER
4776  014436  005302                        DEC     R2              ;SEE IF READ ALL
4777  014440  001350                        BNE     DATOA           ;IF NOT : BR
4778  014442  012701  027452        DATOD:  MOV     #WDATA,R1       ;R1 = START OF WRITE BUFFER
4779  014446  013702  000646                MOV     TEMP2,R2        ;R2 = SIZE OF DATA FIELD
4780  014452  112123        DATOE:  MOVB    (R1)+,(R3)+     ;REPEAT LOAD OF DATA FIELD
4781  014454  022703  033460                CMP     #RDATA,R3       ;SEE IF DONE
4782  014460  003002                        BGT     DATOF           ;IF NOT: BR
4783  014462  000137  014266                JMP     DS2A            ;EXIT
4784  014466  005302        DATOF:  DEC     R2              ;SEE IF AT END OF DATA FIELD
4785  014470  001370                        BNE     DATOE           ;IF NOT : BR
4786  014472  000763                        BR      DATOD           ;ELSE RESTART FILL
4787  014474  000000        DOFL:   0                               ;EXTERNAL DATA FLAG=1 IF ALREADY DONE
4788
```

```
4790                                        ;ALL ONES**********
4791
4792  014476  012701  177777    DAT1:   MOV     #-1,R1          ;R1=DATA
4793  014502  012702  002002    DAT1A:  MOV     #2002,R2        ;R2=WORD COUNT +2
4794  014506  010123            1$:     MOV     R1,(R3)+        ;LOAD BUFFER
4795  014510  005302                    DEC     R2              ;SEE IF DONE
4796  014512  001375                    BNE     1$              ;IF NOT: BR
4797  014514  000207                    RTS     PC
4798
4799                                        ;ALL ZEROS**********
4800
4801  014516  005001    DAT2:   CLR     R1              ;R1=DATA
4802  014520  000770            BR      DAT1A           ;LOAD BUFFER
4803
4804                                        ;WALKING ONE**********
4805
4806  014522  012701  000001    DAT3:   MOV     #1,R1           ;R1=DATA
4807  014526  000241            CLC
4808  014530  012702  004004    DAT3A:  MOV     #4004,R2        ;R2=CHARACTER COUNT+4
4809  014534  110123            1$:     MOVB    R1,(R3)+        ;LOAD BUFFER
4810  014536  106101                    ROLB    R1              ;SET NEXT CHARACTER
4811  014540  005302                    DEC     R2              ;SEE IF DONE
4812  014542  001374                    BNE     1$              ;IF NOT: BR
4813  014544  000207                    RTS     PC
4814
4815                                        ;WALKING ZERO**********
4816
4817  014546  012701  000376    DAT4:   MOV     #376,R1         ;R1=START OF DATA
4818  014552  000261            SEC
4819  014554  000765            BR      DAT3A           ;LOAD BUFFER
4820
4821                                        ;ALTERNATING ONE/ZERO**********
4822
4823
4824  014556  012701  052525    DAT5:   MOV     #52525,R1       ;R1=DATA
4825  014562  000747            BR      DAT1A           ;LOAD BUFFER
4826
4827                                        ;ALTERNATING ZERO/ONE**********
4828
4829  014564  012701  125252    DAT6:   MOV     #125252,R1      ;R1=DATA
4830  014570  000744            BR      DAT1A           ;LOAD BUFFER
4831
4832                                        ;ONE/ZERO IN ALTERNATING WORDS**********
4833
4834  014572  012701  125252    DAT7:   MOV     #125252,R1      ;SET WORD 1
4835  014576  012702  052525            MOV     #52525,R2       ;SET WORD 2
4836  014602  012704  001002            MOV     #1002,R4        ;SET NUMBER OF ENTRIES
4837  014606  010123            1$:     MOV     R1,(R3)+        ;LOAD WORD 1
4838  014610  010223                    MOV     R2,(R3)+        ;LOAD WORD 2
4839  014612  005304                    DEC     R4              ;SEE IF DONE
4840  014614  001374                    BNE     1$              ;IF NOT: BR
4841  014616  000207                    RTS     PC
4842
```

```
4844                                        ;WALKING ONE/ALL ONE IN ALTERNATING CHARS****
4845
4846  014620  012702  002002    DAT10:  MOV     #2002,R2        ;SET BUFFER SIZE
4847  014624  012701  000001            MOV     #1,R1           ;SET WALK BASE
4848  014630  000241                    CLC
4849  014632  012713  177400    1$:     MOV     #177400,(R3)    ;LOAD ALL ONE BYTE
4850  014636  050123                    BIS     R1,(R3)+        ;LOAD WALK BYTE
4851  014640  106101                    ROLB    R1              ;WALK ONE
4852  014642  005302                    DEC     R2
4853  014644  001372                    BNE     1$              ;DO FULL BUFFER
4854  014646  000207                    RTS     PC
4855
4856                                        ;ALL BITS 0-37/**********
4857
4858  014650  005001    DAT11:  CLR     R1              ;R1=STARTING DATA
4859  014652  012702  004004            MOV     #4004,R2        ;R2=CHARACTER COUNT+4
4860  014656  110123    1$:     MOVB    R1,(R3)+        ;LOAD BUFFER
4861  014660  105201                    INCB    R1              ;BUMP DATA
4862  014662  005302                    DEC     R2              ;SEE IF DONE
4863  014664  001374                    BNE     1$              ;IF NOT: BR
4864  014666  000207                    RTS     PC              ;RETURN
4865
4866                                        ;ALL BITS 377-0**********
4867
4868  014670  012701  000377    DAT12:  MOV     #377,R1         ;R1=STARTING DATA
4869  014674  012702  004004            MOV     #4004,R2        ;R2=CHARACTER COUNT+4
4870  014700  110123    1$:     MOVB    R1,(R3)+        ;LOAD BUFFER
4871  014702  105301                    DECB    R1              ;BUMP DATA
4872  014704  005302                    DEC     R2              ;SEE IF DONE
4873  014706  001374                    BNE     1$              ;IF NOT: BR
4874  014710  000207                    RTS     PC              ;RETURN
4875
4876                                        ;ALTERNATING CHARACTERS 0 AND 377**********
4877
4878  014712  012701  000377    DAT13:  MOV     #377,R1         ;R1 = DATA
4879  014716  000137  014502            JMP     DAT1A           ;LOAD BUFFER
4880
4881                                        ;WALKING ZERO/ALL ZERO IN ALTERNATING CHARS******
4882
4883  014722  012702  002002    DAT14:  MOV     #2002,R2        ;SET BUFFER SIZE
4884  014726  012701  000376            MOV     #376,R1         ;SET WALK BASE
4885  014732  000261                    SEC
4886  014734  010113    1$:     MOV     R1,(R3)         ;LOAD WALK BYTE
4887  014736  042723  177400            BIC     #177400,(R3)+   ;CLEAR HIGH BYTE
4888  014742  106101                    ROLB    R1              ;WALK ZERO BIT
4889  014744  005302                    DEC     R2
4890  014746  001372                    BNE     1$              ;FILL BUFFER
4891  014750  000207                    RTS     PC              ;RETURN
4892
```

```
                                                        ;AUTO SEQUENCE PATTERN*********
                                                        
014752  012702  000200          DAT15:  MOV     #200,R2         ;SET NUMBER OF ENTRIES
014756  012701  015002          1$:     MOV     #APATS,R1       ;SET START OF PATTERN
014762  012704  000010                  MOV     #10,R4          ;SET SIZE OF PATTERN
014766  012123                  2$:     MOV     (R1)+,(R3)+     ;FILL BUFFER
014770  005304                          DEC     R4              ;SEE IF DONE PATTERN
014772  001375                          BNE     2$              ;IF NOT: BR
014774  005302                          DEC     R2              ;SEE IF DONE BUFER
014776  001367                          BNE     1$              ;IF NOT: BR
015000  000207                          RTS     PC              ;RETURN
                                                        
015002  000000                  APATS:  0
015004  177400                          177400
015006  000377                          377
015010  000000                          0
015012  177777                          -1
015014  000377                          377
015016  177400                          177400
015020  177777                          -1
                                                        
                                                        ;RANDOM DATA GENERATOR SUBROUTINE**********
                                                        
015022  013704  000556          DATR:   MOV     FMCNT,R4        ;SET NUMBER OF FRAMES
015026  012703  027452                  MOV     #WDATA,R3       ;SET ADDRESS OF START OF BUFFER
015032  012701  177777                  MOV     #-1,R1          ;SET HIGH LIMIT
015036  005002                          CLR     R2              ;SET LOW LIMIT
015040  004737  023254          1$:     JSR     PC,RANG         ;GO GENERATE NUMBER
015044  013723  000630                  MOV     RANSAV,(R3)+    ;LOAD BUFFER
015050  005204                          INC     R4              ;SEE IF DONE WHOLE BUFFER
015052  001372                          BNE     1$              ;IF NOT: BR
015054  012737  000001  015064          MOV     #1,RDFL         ;SET RANDOM DATA FLAG
015062  000207                          RTS     PC              ;EXIT
015064  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
                                        ;*************************************************************

015066  032737  002000  000552  CRCLRC: BIT     #2000,UDES
015074  001105                          BNE     CL4             ;IF IN PE MODE: BR
015076  012737  177777  014334          MOV     #-1,CLF         ;SET EXPT CRCLRC CAL.FLAG
015104  013700  000556                  MOV     FMCNT,R0        ;SET RECORD SIZE
015110  005400                          NEG     R0
015112  012701  027452                  MOV     #WDATA,R1       ;SET START OF BUFFER
015116  005037  015440                  CLR     XORS
015122  111104                  CL0:    MOVB    (R1),R4         ;GET CHARACTER
015124  004737  015312                  JSR     PC,CLP          ;GO GET PARITY OF CHARACTER
015130  004737  015414                  JSR     PC,XOR          ;XOR CHARACTER
015134  000241                          CLC
015136  006004                          ROR     R4              ;ROTATE 1 RIGHT
015140  103014                          BCC     CL2             ;IF NO CARRY: BR
015142  052704  000400                  BIS     #400,R4         ;SET BIT NINE
015146  000241                          CLC
015150  010405                  CL1:    MOV     R4,R5           ;SAVE CHARACTER
015152  042705  177703                  BIC     #177703,R5
015156  005105                          COM     R5
015160  042705  177703                  BIC     #177703,R5
015164  042704  000074                  BIC     #74,R4
015170  050504                          BIS     R5,R4           ;COMPLIMENT BITS 2,3,4,5
015172  010437  015440          CL2:    MOV     R4,XORS
015176  005300                          DEC     R0
015200  001350                          BNE     CL0             ;BRANCH IF NOT LAST CHAR
015202  013704  015440          CLLAST: MOV     XORS,R4
015206  005137  015440                  COM     XORS
015212  042737  177050  015440        BIC     #177050,XORS
015220  042704  177727                  BIC     #177727,R4      ;COMPLIMENT ALL BUT BITS 3&5
015224  050437  015440                  BIS     R4,XORS
015230  013737  015440  015442        MOV     XORS,EXCRC      ;SAVE EXPECTED CRC
015236  013700  000556                  MOV     FMCNT,R0
015242  005400                          NEG     R0
015244  012701  027452                  MOV     #WDATA,R1       ;DO EXPT LRC
015250  005037  015440                  CLR     XORS
015254  111104                  CL3:    MOVB    (R1),R4
015256  004737  015312                  JSR     PC,CLP          ;GET PARITY
015262  004737  015414                  JSR     PC,XOR          ;XOR CHARACTER
015266  005300                          DEC     R0
015270  001371                          BNE     CL3             ;DO ALL FOR LRC
015272  013704  015442                  MOV     EXCRC,R4
015276  004737  015414                  JSR     PC,XOR          ;XOR CRC TO DATA
015302  013737  015440  015444        MOV     XORS,EXLRC      ;SAVE EXPT LRC
015310  000207                  CL4:    RTS     PC              ;RETURN
015312  005704                  CLP:    TST     R4              ;SEE IF 0 CHAR
015314  001010                          BNE     CLPE            ;IF NOT: BR
015316  032737  000010  000552        BIT     #10,UDES        ;SEE IF EVEN PARITY
015324  001404                          BEQ     CLPE            ;IF NOT: BR
```

```
 4988  015326  012704  000420            MOV     #420,R4         ;SET 0 CHAR EVEN PARITY
 4989  015332  005201                    INC     R1              ;BUMP POINTER
 4990  015334  000207                    RTS     PC              ;RETURN
 4991  015336  005046            CLPE:   CLR     -(SP)           ;CLEAR WORLING SPACE ON STACK
 4992  015340  106304            1$:     ASLB    R4              ;SHIFT DATA
 4993  015342  005516                    ADC     (SP)            ;ADDUP # OF 1 BITS
 4994  015344  105704                    TSTB    R4              ;BRANCH IF ALL 0'S LEFT
 4995  015346  001374                    BNE     1$
 4996  015350  112104                    MOVB    (R1)+,R4
 4997  015352  042704  177400            BIC     #177400,R4
 4998  015356  106026                    RORB    (SP)+           ;BRANCH IF ODD # OF 1 BITS
 4999  015360  103405                    BCS     CLP2
 5000  015362  032737  000010  000552    BIT     #10,UDES        ;SEE IF SHOULD BE EVEN PARITY
 5001  015370  001406                    BEQ     CLP3            ;IF NOT: BR
 5002  015372  000207                    RTS     PC              ;ELSE EXIT
 5003  015374  032737  000010  000552    BIT     #10,UDES        ;SEE IF SHOULD BE ODD PARITY
 5004  015402  001001                    BNE     CLP3            ;IF NOT: BR
 5005  015404  000207                    RTS     PC              ;ELSE EXIT
 5006  015406  052704  000400    CLP3:   BIS     #400,R4         ;SET PARITY BIT
 5007  015412  000207                    RTS     PC
 5008
 5009  015414  010446            XOR:    MOV     R4,-(SP)
 5010  015416  043716  015440            BIC     XORS,(SP)
 5011  015422  040437  015440            BIC     R4,XORS         ;XOR SUBROUTINE: R4 WITH XORS
 5012  015426  052637  015440            BIS     (SP)+,XORS
 5013  015432  013704  015440            MOV     XORS,R4
 5014  015436  000207                    RTS     PC
 5015
 5016  015440  000000            XORS:   0                       ;XOR SAVE
 5017  015442  000000            EXCRC:  0                       ;EXPECTED CRC
 5018  015444  000000            EXLRC:  0                       ;EXPECTED LRC
 5019
```

```
5021
5022                                    ;*****************************************************
5023                                    ;DATA CHECK SUBROUTINE:
5024                                    ;
5025                                    ;THIS SUBROUTINE IS USED TO COMPARE EACH CHARACTER
5026                                    ;OF DATA READ FROM TAPE WITH THE EXPECTED CHARACTER.
5027                                    ;ANY ERROR DETECTED WILL CAUSE CONTROL TO BE
5028                                    ;PASSED TO AN ERROR PRINT SUBROUTINE AND A
5029                                    ;SUBROUTINE TO ACCUMULATE THE NUMBER OF BITS
5030                                    ;DROPPED AND PICKED UP FROM EACH CHARACTER.
5031                                    ;THE NUMBER OF READ ERRORS IS ALSO ACCUMULATED.
5032                                    ;DATA CHECKING MAY BE TERMINATED BY USE OF
5033                                    ;CONSOLE SWITCH THIRTEEN (13).
5034                                    ;*****************************************************
5035
5036  015446  005037  000660        DCHK:   CLR     BBC             ;CLEAR BAD RECORD CNTR
5037  015452  005037  000706                CLR     DERFL           ;CLEAR DATA ERROR FLAG
5038  015456  013705  000556                MOV     FMCNT,R5        ;LOAD CHAR COUNT
5039  015462  032737  000020  000552        BIT     #20,UDES        ;SEE IF CORE DUMP
5040  015470  001401                        BEQ     DCHK0           ;IF NOT: BR
5041  015472  006205                        ASR     R5              ;R5 = FC/2
5042  015474  012701  027452        DCHK0:  MOV     #WDATA,R1       ;SET WRITE DATA ADDR
5043  015500  012702  033460                MOV     #RDATA,R2       ;SET READ DATA ADDR
5044  015504  032737  000010  000552        BIT     #10,UDES        ;SEE IF EVEN PARITY
5045  015512  001430                        BEQ     DF0C0           ;IF NOT: BR
5046  015514  032737  000020  000552        BIT     #20,UDES        ;SEE IF CORE DUMP PARITY
5047  015522  001024                        BNE     DF0C0           ;IF SO: BR
5048  015524  032737  002000  000552        BIT     #2000,UDES      ;SEE IF PE MODE
5049  015532  001020                        BNE     DF0C0           ;IF SO: BR
5050  015534  105711                DF0F:   TSTB    (R1)            ;SEE IF 0 CHAR
5051  015536  001404                        BEQ     DF0D            ;IF SO: BR
5052  015540  005201                        INC     R1              ;BUMP POINTER
5053  015542  005205                DF0E:   INC     R5              ;SEE IF DONE
5054  015544  001373                        BNE     DF0F            ;IF NOT: BR
5055  015546  000406                        BR      DF0C            ;ELSE CONTINUE
5056  015550  112721  000020        DF0D:   MOVB    #20,(R1)+       ;SET 20 IN PLACE OF 0
5057  015554  012737  177777  014330        MOV     #-1,PATS        ;SET PATTERN GENERATE FLAG
5058  015562  000767                        BR      DF0E
5059  015564  013705  000556        DF0C:   MOV     FMCNT,R5        ;RESET CHAR CNT
5060  015570  012701  027452                MOV     #WDATA,R1       ;RESET DATA ADDRESS
5061  015574  032737  010000  000562 DF0C0:  BIT     #10000,RDCMD    ;SEE IF READ REVERSE
5062  015602  001462                        BEQ     DF0             ;IF NOT: BR
5063  015604  013704  000556        DF0B:   MOV     FMCNT,R4        ;GET FRAME COUNT
5064  015610  005404                        NEG     R4              ;SET TO WHOLE NUMBER
5065  015612  032737  000020  000552        BIT     #20,UDES        ;SEE IF CORE DUMP
5066  015620  001402                        BEQ     DF0B0           ;IF NOT: BR
5067  015622  000241                        CLC
5068  015624  006004                        ROR     R4              ;SET TO FC/2
5069  015626  060401                DF0B0:  ADD     R4,R1           ;POINT TO START OF WRITE DATA
5070  015630  060402                        ADD     R4,R2           ;POINT TO START OF READ DATA
5071  015632  032737  000001  000556        BIT     #1,FMCNT        ;SEE IF ODD FRAME COUNT
5072  015640  001401                        BEQ     DF0A            ;IF NOT: BR
5073  015642  105722                        TSTB    (R2)+           ;BUMP POINTER
5074  015644  032737  000020  000552 DF0A:   BIT     #20,UDES        ;SEE IF CORE DUMP
5075  015652  001431                        BEQ     DF0A4           ;IF NOT: BR
5076  015654  000241                        CLC
```

```
 5077  015656  132742  000001          BITB    #1,-(R2)        ;SEE IF BIT 0 = 1
 5078  015662  001401                  BEQ     DF0A0           ;IF NOT: BR
 5079  015664  000261                  SEC
 5080  015666  106012          DF0A0:  RORB    (R2)
 5081  015670  000241                  CLC
 5082  015672  132712  000001          BITB    #1,(R2)
 5083  015676  001401                  BEQ     DF0A1
 5084  015700  000261                  SEC
 5085  015702  106012          DF0A1:  RORB    (R2)            ;POSITION BITS FOR REVERSE CORE DUMP
 5086  015704  000241                  CLC
 5087  015706  132712  000001          BITB    #1,(R2)
 5088  015712  001401                  BEQ     DF0A2
 5089  015714  000261                  SEC
 5090  015716  106012          DF0A2:  RORB    (R2)
 5091  015720  000241                  CLC
 5092  015722  132712  000001          BITB    #1,(R2)
 5093  015726  001401                  BEQ     DF0A3
 5094  015730  000261                  SEC
 5095  015732  106012          DF0A3:  RORB    (R2)
 5096  015734  005202                  INC     R2              ;RESET POINTER
 5097  015736  124142          DF0A4:  CMPB    -(R1),-(R2)     ;TEST DATA CHARACTER
 5098  015740  001010                  BNE     DF1             ;IF NOT GOOD: BR
 5099  015742  105037  000660          CLRB    BBC             ;CLEAR BAD RECORD COUNTER
 5100  015746  000411                  BR      DF2
 5101  015750  122122          DF0:    CMPB    (R1)+,(R2)+     ;CHECK DATA
 5102  015752  001003                  BNE     DF1             ;IF BAD: BR
 5103  015754  105037  000660          CLRB    BBC             ;CLEAR BAD RECORD CNTR
 5104  015760  000404                  BR      DF2
 5105  015762  004737  016574  DF1:    JSR     PC,DRPKF        ;GO GET DROPS AND PICKS
 5106  015766  004737  016060          JSR     PC,DERR         ;GO DO PRINT
 5107  015772  005205          DF2:    INC     R5              ;BUMP CHAR CNTR
 5108  015774  001405                  BEQ     DF3             ;IF DONE ALL: BR
 5109  015776  032737  010000  000562  BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 5110  016004  001761                  BEQ     DF0             ;IF NOT: BR
 5111  016006  000716                  BR      DF0A            ;ELSE CONTINUE READ REV
 5112  016010  005037  000666  DF3:    CLR     HDRFL           ;CLEAR HEADER FLAG
 5113  016014  005737  000706          TST     DERFL           ;SEE IF HAD DATA ERROR
 5114  016020  001416                  BEQ     DFX             ;IF NOT: BR
 5115  016022  005737  000710          TST     SERFL
 5116  016026  001013                  BNE     DFX             ;IF NOT DATA ERROR ONLY: BR
 5117  016030  013704  000676          MOV     UNP,R4
 5118  016034  032737  010000  000562  BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 5119  016042  001003                  BNE     DF4             ;IF SO: BR
 5120  016044  005264  001130          INC     DATER1(R4)      ;BUMP DATA ERROR FORWARD COUNTER
 5121  016050  000402                  BR      DFX
 5122  016052  005264  001170  DF4:    INC     DEREV1(R4)      ;BUMP REVERSE DATA ERROR
 5123  016056  000207          DFX:    RTS     PC              ;EXIT
 5124
```

```
5126
5127                            ;*************************************************************
5128                            ;DATA ERROR SUBROUTINE:
5129                            ;
5130                            ;THIS SUBROUTINE IS USED TO PRINT OUT ANY
5131                            ;ERRORS FOUND DURING THE DATA CHECK.
5132                            ;EACH CHARACTER FOUND BAD WILL BE PRINTED
5133                            ;IN BIT FORMAT ALONG WITH ITS EXPECTED CHARACTER.
5134                            ;AN ERROR HEADER CONSISTING OF THE UNIT NUMBER,
5135                            ;BLOCK NUMBER, RECORD NUMBER, SIZE OF RECORD, AND
5136                            ;ERROR TYPE (READ FORWARD, READ REVERSE, WRITE, ETC)
5137                            ;IS PRINTED ONLY ONCE FOR EACH RECORD FOUND BAD.
5138                            ;A COUNT IS MADE OF THE NUMBER OF SUCCESSIVE BAD
5139                            ;CHARACTERS, AND IF TEN (10) SUCCESSIVE BAD CHARACTERS
5140                            ;ARE FOUND IN A SINGLE RECORD, A MESSAGE INDICATING
5141                            ;A BAD RECORD CONDITION IS PRINTED AND THE NEXT
5142                            ;TWENTY (20) CHARACTERS ARE SKIPPED BEFORE CHECKING
5143                            ;IS RESUMED.  IF THE BAD RECORD CONDITION IS FOUND
5144                            ;THREE TIMES IN A RECORD, ALL REMAINING DATA IS
5145                            ;SKIPPED EXCEPT THE FINAL TEN (10) CHARACTERS.
5146                            ;THIS SKIPPING IS OF COURSE ONLY POSSIBLE IN
5147                            ;RECORDS WHICH CONTAIN A SUFFICIENT NUMBER OF CHARACTERS.
5148                            ;PRINTING OF ERRORS MAY BE DISALLOWED AT ANY TIME
5149                            ;BY SETTING CONSOLE SWITCH TEN (10) TO A ONE.
5150                            ;THE OPERATOR MAY CAUSE THE PROGRAM TO HALT ON ANY ERROR
5151                            ;BY SETTING CONSOLE SWITCH FIFTEEN (15) TO A ONE.
5152                            ;*************************************************************
5153
5154  016060  032777  002000  162522  DERR:   BIT     #2000,@SWR      ;BRANCH IF NO ERROR
5155  016066  001067                    BNE     DERR4           ;PRINTOUT DESIRED
5156  016070  005237  000672          DERR0:  INC     PFLG            ;SET PRINT FLAG
5157  016074  005737  000666                  TST     HDRFL           ;SEE IF HAVE PRINTED HEADER
5158  016100  001007                    BNE     DERR0A          ;IF SO: BR
5159  016102  004737  022672                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
5160  016106  012704  024564                  MOV     #MSG1,R4        ;LOAD ERROR MSG ADDR
5161  016112  000004                    TYPE                    ;TYPE MSG
5162  016114  004737  021110                  JSR     PC,FRPRT        ;PRINT F OR R
5163  016120  012704  024603          DERR0A: MOV     #MSG4,R4
5164  016124  000004                    TYPE                    ;TYPE MSG
5165  016126  010203                    MOV     R2,R3
5166  016130  162703  033460                  SUB     #RDATA,R3       ;POINT TO CHAR
5167  016134  005303                    DEC     R3
5168  016136  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
5169  016144  001402                    BEQ     DERR0B          ;IF NOT: BR
5170  016146  010503                    MOV     R5,R3           ;GET CHAR NUMBER
5171  016150  005103                    COM     R3
5172  016152  104400          DERR0B: TYPOCT                  ;PRINT CHAR NUMBER
5173  016154  012704  024571                  MOV     #MSG2,R4
5174  016160  000004                    TYPE                    ;TYPE MSG
5175  016162  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
5176  016170  001402                    BEQ     DERR0C          ;IF NOT: BR
5177  016172  111103                    MOVB    (R1),R3         ;GET CHAR
5178  016174  000401                    BR      DERR0D
5179  016176  114103          DERR0C: MOVB    -(R1),R3        ;LOAD EXPECTED DATA
5180  016200  004737  024342  DERR0D: JSR     PC,DOUT         ;GO PRINT CHAR
5181  016204  012704  024576                  MOV     #MSG3,R4
```

```
5182  016210  000004                          TYPE                    ;TYPE MSG
5183  016212  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
5184  016220  001402                          BEQ     DERR1           ;IF NOT: BR
5185  016222  111203                          MOVB    (R2),R3         ;GET CHAR
5186  016224  000401                          BR      DERR2
5187  016226  114203                  DERR1:  MOVB    -(R2),R3
5188  016230  004737  024342          DERR2:  JSR     PC,DOUT         ;PRINT BAD CHAR
5189  016234  032737  010000  000562          BIT     #10000,RDCMD    ;BRANCH IF NOT READ
5190  016242  001001                          BNE     DERR4           ;REVERSE
5191  016244  122122                  DERR3:  CMPB    (R1)+,(R2)+     ;RESET POINTERS
5192  016246  105237  000660          DERR4:  INCB    BBC             ;BUMP BAD RECORD CNTR
5193  016252  122737  000010  000660          CMPB    #10,BBC         ;SEE IF BLD BTH
5194  016260  001123                          BNE     DEREX           ;IF NOT: BR
5195  016262  032777  002000  162320          BIT     #2000,@SWR      ;SEE IF PRINT INHIBIT
5196  016270  001003                          BNE     1$              ;IF SO: BR
5197  016272  012704  024717                  MOV     #MSG15,R4
5198  016276  000004                          TYPE                    ;TYPE MSG
5199  016300  105037  000660          1$:     CLRB    BBC             ;RESET BAD RECORD CNTR
5200  016304  000337  000660                  SWAB    BBC             ;POSITION BLD BTH AMOUNT
5201  016310  105237  000660                  INCB    BBC             ;BUMP AMOUNT
5202  016314  122737  000003  000660          CMPB    #3,BBC          ;SEE IF HAD 3 BLD BTHS
5203  016322  101054                          BHI     DERR4B          ;IF NOT: BR
5204  016324  000337  000660                  SWAB    BBC             ;REPOSITION BBC
5205  016330  022705  177767                  CMP     #177767,R5      ;SEE IF ON LAST EIGHT CHARS
5206  016334  101473                          BLOS    DERR6           ;IF SO: BR
5207  016336  012705  177767                  MOV     #177767,R5      ;SET CHAR CNTR TO 8
5208  016342  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
5209  016350  001416                          BEQ     DERR4A          ;IF NOT: BR
5210  016352  012701  027452                  MOV     #WDATA,R1       ;GET START OF BUFFER
5211  016356  012702  033460                  MOV     #RDATA,R2       ;GET START OF BUFFER
5212  016362  062701  000010                  ADD     #10,R1
5213  016366  062702  000010                  ADD     #10,R2          ;POINT TO START +10
5214  016372  032737  000001  000556          BIT     #1,FMCNT        ;SEE IF ODD FRAME COUNT
5215  016400  001453                          BEQ     DEREX           ;IF NOT: BR
5216  016402  105722                          TSTB    (R2)+           ;BUMP POINTER
5217  016404  000451                          BR      DEREX
5218  016406  013737  000556  000644  DERR4A: MOV     FMCNT,TEMP1     ;LOAD CHAR COUNT
5219  016414  005137  000644                  COM     TEMP1
5220  016420  005237  000644                  INC     TEMP1
5221  016424  162737  000010  000644          SUB     #10,TEMP1       ;POINT TO BUFFER -8
5222  016432  013701  000644                  MOV     TEMP1,R1        ;POINT TO NEXT CHAR
5223  016436  062701  027452                  ADD     #WDATA,R1       ;POINT TO NEXT WRITE CHAR
5224  016442  013702  000644                  MOV     TEMP1,R2        ;POINT TO END OF READ DATA -8 FORWARD
5225  016446  062702  033460                  ADD     #RDATA,R2       ;POINT TO NEXT CHAR
5226  016452  000426                          BR      DEREX           ;EXIT
5227  016454  000337  000660          DERR4B: SWAB    BBC             ;REPOSITION BBC
5228  016460  000241                          CLC
5229  016462  062705  000024                  ADD     #24,R5          ;SKIP 20 CHARS
5230  016466  103416                          BCS     DERR6           ;IF EXCEED RECORD SIZE: BR
5231  016470  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
5232  016476  001405                          BEQ     DERR5           ;IF NOT: BR
5233  016500  162701  000024                  SUB     #24,R1
5234  016504  162702  000024                  SUB     #24,R2          ;RESET POINTERS
5235  016510  000407                          BR      DEREX
5236  016512  062701  000024          DERR5:  ADD     #24,R1          ;SKIP 20 CHARS
5237  016516  062702  000024                  ADD     #24,R2          ;SKIP FORWARD 20 CHARS
```

```
 5238  016522  000402                        BR      DEREX
 5239  016524  012705  177777        DERR6:  MOV     #-1,R5          ;SET TO EOR
 5240  016530  005777  162054        DEREX:  TST     @SWR            ;BRANCH IF NOT HALT ON ERROR
 5241  016534  100012                        BPL     DEREX1
 5242  016536  000000                        HALT
 5243  016540  005737  000672                TST     PFLG            ;SEE IF PRINTED
 5244  016544  001006                        BNE     DEREX1          ;IF SO: BR
 5245  016546  032777  002000  162034        BIT     #2000,@SWR      ;SEE IF SHOULD PRINT
 5246  016554  001002                        BNE     DEREX1          ;IF NOT: BR
 5247  016556  000137  016070                JMP     DERR0           ;ELSE PRINT
 5248  016562  005037  000672        DEREX1: CLR     PFLG            ;CLEAR FLAG
 5249  016566  005237  000706                INC     DERFL           ;BUMP DATA ERROR FLAG
 5250  016572  000207                        RTS     PC              ;RETURN
 5251
```

```
5253
5254                                    ;*************************************************************
5255                                    ;DROPS AND PICKS SUBROUTINE:
5256                                    ;
5257                                    ;THIS SUBROUTINE IS USED TO ACCUMULATE FROM
5258                                    ;EACH BAD DATA CHARACTER FOUND THE NUMBER
5259                                    ;OF BITS WHICH WERE EITHER DROPPED OR PICKED UP.
5260                                    ;TWO COUNTERS PER SLAVE ARE USED TO ACCUMULATE THIS
5261                                    ;INFORMATION AND CAN STORE UP TO 32K DROPS
5262                                    ;OR PICKS BEFORE OVERFLOWING.  IF OVERFLOW IS
5263                                    ;ABOUT TO OCCUR, THESE ACCUMULATORS ARE
5264                                    ;PRINTED IN OCTAL AND RESET TO ZERO.
5265                                    ;THE CONTENTS OF THE ACCUMULATORS MAY BE
5266                                    ;DISPLAYED AT ANY TIME BY SETTING CONSOLE
5267                                    ;SWITCH FOURTEEN TO A ONE (1). THE PRINTOUT WILL OCCUR
5268                                    ;AT THE END OF THE CURRENT BLOCK CYCLE.
5269                                    ;*************************************************************
5270
5271  016574  005037  000644          DRPKF:  CLR     TEMP1
5272  016600  005037  000646                  CLR     TEMP2
5273  016604  005037  000650                  CLR     TEMP3
5274  016610  111137  000644                  MOVB    (R1),TEMP1      ;LOAD GOOD CHAR
5275  016614  111237  000646                  MOVB    (R2),TEMP2      ;LOAD BAD CHAR
5276  016620  013704  000676                  MOV     UNP,R4
5277  016624  016437  000770  000722          MOV     PIK1(R4),BPKP
5278  016632  016437  001010  000720          MOV     DRP1(R4),BDPP
5279  016640  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
5280  016646  001005                          BNE     DRPK            ;IF SO: BR
5281  016650  124142                          CMPB    -(R1),-(R2)     ;POINT TO CHAR
5282  016652  112137  000644                  MOVB    (R1)+,TEMP1     ;LOAD GOOD CHAR
5283  016656  112237  000646                  MOVB    (R2)+,TEMP2     ;LOAD BAD CHAR
5284  016662  004737  016674          DRPK:   JSR     PC,DROP         ;GET DROPS
5285  016666  004737  017114                  JSR     PC,PICK         ;GET PICKS
5286  016672  000207                          RTS     PC              ;EXIT
5287  016674  113703  000644          DROP:   MOVB    TEMP1,R3        ;R3 = GOOD CHAR
5288  016700  113704  000646                  MOVB    TEMP2,R4        ;R4 = BAD CHAR
5289  016704  140403                  DPC:    BICB    R4,R3           ;GET DROPS/PICKS
5290  016706  001001                          BNE     DPCG            ;IF SOME: BR
5291  016710  000207                          RTS     PC              ;RETURN
5292  016712  012737  000010  000712  DPCG:   MOV     #10,BCNT        ;SET NUMBER TO CHECK
5293  016720  132703  000001          DPC0:   BITB    #1,R3           ;SEE IF DROPPED OR PICKED THIS BIT
5294  016724  001455                          BEQ     DPC2            ;IF NOT: BR
5295  016726  105737  000650                  TSTB    TEMP3           ;SEE IF ON PICKS
5296  016732  001016                          BNE     DPC1            ;IF SO: BR
5297  016734  005277  161760                  INC     @BDPP           ;BUMP DROP CNTR
5298  016740  005777  161754                  TST     @BDPP
5299  016744  100045                          BPL     DPC2            ;IF NO OVERFLOW: BR
5300  016746  032777  002000  161634          BIT     #2000,@SWR      ;SEE IF HAVE PRINTED DATA
5301  016754  001402                          BEQ     DPC0A           ;IF SO: BR
5302  016756  004737  022672                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
5303  016762  004737  017160          DPC0A:  JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
5304  016766  000415                          BR      DPC2A
5305  016770  005277  161726          DPC1:   INC     @BPKP           ;BUMP PICK CNTR
5306  016774  005777  161722                  TST     @BPKP           ;SEE IF OVERFLOW
5307  017000  100027                          BPL     DPC2            ;IF NOT: BR
5308  017002  032777  002000  161600          BIT     #2000,@SWR      ;SEE IF HAVE PRINTED DATA
```

```
 5309  017010  001402                          BEQ     DPC1A           ;IF SO: BR
 5310  017012  004737  022672                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
 5311  017016  004737  017160          DPC1A:  JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
 5312  017022  013704  000676          DPC2A:  MOV     UNP,R4
 5313  017026  016403  001010                  MOV     DRP1(R4),R3     ;SET DROP POINTER
 5314  017032  016404  000770                  MOV     PIK1(R4),R4     ;SET PICK POINTER
 5315  017036  012737  000010  000712          MOV     #10,BCNT        ;SET NUMBER OF BITS
 5316  017044  005023                  DPC2B:  CLR     (R3)+           ;CLEAR DROPS
 5317  017046  005024                          CLR     (R4)+           ;CLEAR PICK
 5318  017050  005337  000712                  DEC     BCNT            ;SEE IF DONE
 5319  017054  001373                          BNE     DPC2B           ;IF NOT: BR
 5320  017056  000207                          RTS     PC              ;EXIT
 5321  017060  000241                  DPC2:   CLC
 5322  017062  106003                          RORB    R3              ;GET NEXT BIT
 5323  017064  005337  000712                  DEC     BCNT            ;SEE IF DONE
 5324  017070  001410                          BEQ     DPC3
 5325  017072  062737  000002  000722          ADD     #2,BPKP
 5326  017100  062737  000002  000720          ADD     #2,BDPP
 5327  017106  000137  016720                  JMP     DPC0            ;CONTINUE
 5328  017112  000207                  DPC3:   RTS     PC              ;RETURN
 5329  017114  013704  000676          PICK:   MOV     UNP,R4          ;GET UNIT POINTER
 5330  017120  016437  000770  000722          MOV     PIK1(R4),BPKP   ;SET PICK POINTER
 5331  017126  016437  001010  000720          MOV     DRP1(R4),BDPP   ;SET DROP POINTER
 5332  017134  113704  000644                  MOVB    TEMP1,R4        ;R4 = GOOD CHAR
 5333  017140  113703  000646                  MOVB    TEMP2,R3        ;R3 = BAD CHAR
 5334  017144  112737  000001  000650          MOVB    #1,TEMP3        ;SET PICK FLAG
 5335  017152  004737  016704                  JSR     PC,DPC          ;GO CHECK PICKS
 5336  017156  000207                          RTS     PC              ;EXIT
 5337  017160  012704  025215          DPPRT:  MOV     #MSG26,R4
 5338  017164  000004                          TYPE                    ;TYPE MSG
 5339  017166  013704  000676                  MOV     UNP,R4
 5340  017172  016437  001010  000720          MOV     DRP1(R4),BDPP   ;SET DROP POINTER
 5341  017200  016437  000770  000722          MOV     PIK1(R4),BPKP   ;SET PICK POINTER
 5342  017206  062737  000016  000720          ADD     #16,BDPP
 5343  017214  062737  000016  000722          ADD     #16,BPKP
 5344  017222  012737  000010  000712          MOV     #10,BCNT        ;SET NUMBER TO PRINT
 5345  017230  017703  161464          DPPRT0: MOV     @BDPP,R3
 5346  017234  104400                          TYPOCT                  ;PRINT DROPS
 5347  017236  005337  000712                  DEC     BCNT            ;SEE IF DONE
 5348  017242  001404                          BEQ     DPPRT1          ;IF NOT: BR
 5349  017244  162737  000002  000720          SUB     #2,BDPP         ;BUMP POINTER
 5350  017252  000766                          BR      DPPRT0          ;CONTINUE FOR ALL 8 BITS
 5351  017254  012737  000010  000712  DPPRT1: MOV     #10,BCNT        ;SET NUMBER TO PRINT
 5352  017262  012704  025226                  MOV     #MSG27,R4
 5353  017266  000004                          TYPE                    ;TYPE MSG
 5354  017270  017703  161426          DPPRT2: MOV     @BPKP,R3
 5355  017274  104400                          TYPOCT                  ;PRINT PICKS
 5356  017276  005337  000712                  DEC     BCNT            ;SEE IF DONE
 5357  017302  001404                          BEQ     DPPRTX          ;IF SO: BR
 5358  017304  162737  000002  000722          SUB     #2,BPKP         ;BUMP POINTER
 5359  017312  000766                          BR      DPPRT2          ;CONTINUE FOR ALL 8 BITS
 5360  017314  000207                  DPPRTX: RTS     PC              ;RETURN
```

```
5362
5363                                    ;*********************************************************
5364                                    ;STATUS CHECK SUBROUTINE:
5365                                    ;
5366                                    ;THIS SUBROUTINE IS USED TO PERFORM A CHECK OF
5367                                    ;BOTH THE MASSBUS CONTROLLER (RH11) AND THE TAPE
5368                                    ;CONTROLLER (TM03).  THE RH11 IS CHECKED FOR ERRORS
5369                                    ;AS REFLECTED IN REGISTERS CS1 AND CS2 AND ALSO THAT
5370                                    ;THE BUS ADDRESS (BA) AND WORD COUNT (WC) ARE
5371                                    ;CORRECT.  THE TM03 IS CHECKED FOR DRIVE STATIS (DS),
5372                                    ;DRIVE ERRORS (ER), AND PROPER FRAME COUNT.  THE SPECIAL
5373                                    ;CHECK CHARACTERS (CRC+LRC) ARE ALSO CHECKED WHEN
5374                                    ;APPROPIATE (IE: NRZ READ OR WRITE).  CERTAIN TYPES
5375                                    ;OF DRIVE ERRORS IN PE OPERATION WILL BE ACCOMPANIED
5376                                    ;BY THE DISPLAY OF THE DEAD TRACK REGISTER (CC).  THESE
5377                                    ;TYPES ARE ER BITS 15,10,7,6.  THE PRINTOUTS OF BAD
5378                                    ;CRC,LRC,FC, AND BA WILL SHOW BOTH THE EXPECTED AND
5379                                    ;RECEIVED VALUES (IE: EXPT-RCVD).  ONLY THOSE REGISTERS
5380                                    ;WHICH ARE IN ERROR WILL BE PRINTED AND ALL PRINTOUTS
5381                                    ;ARE IN OCTAL FORMAT WITH NO LEADING ZEROS.  AS IN
5382                                    ;DATA ERRORS, STATUS ERRORS ARE PRECEEDED BY HEADER
5383                                    ;DESCRIBING THE HARDWARE UNDER TEST, THE BLOCKING
5384                                    ;INFORMATION, AND THE ERROR TYPE.
5385                                    ;*********************************************************
5386
5387  017316  013703  000556        ERCHK:  MOV     FMCNT,R3        ;GET FRAME COUNT
5388  017322  032703  000001                BIT     #1,R3           ;SEE IF ODD
5389  017326  001401                        BEQ     ER0             ;IF NOT: BR
5390  017330  005303                        DEC     R3              ;BUMP COUNT
5391  017332  005403                ER0:    NEG     R3
5392  017334  032737  000020  000552        BIT     #20,UDES        ;SEE IF CORE DUMP
5393  017342  001401                        BEQ     ER0B            ;IF NOT: BR
5394  017344  006203                        ASR     R3              ;SET TO FC/2
5395  017346  032737  000010  000674  ER0B:  BIT     #10,MTC1        ;SEE IF WRITE OP
5396  017354  001414                        BEQ     ER1             ;IF SO: BR
5397  017356  032737  010000  000562        BIT     #10000,RDCMD
5398  017364  001405                        BEQ     ER0A
5399  017366  012703  033460                MOV     #RDATA,R3
5400  017372  162703  000002                SUB     #2,R3           ;SET POINTER
5401  017376  000405                        BR      ER2
5402  017400  062703  033460        ER0A:   ADD     #RDATA,R3       ;BUILD EXPT READ ADDRESS
5403  017404  000402                        BR      ER2
5404  017406  062703  027452        ER1:    ADD     #WDATA,R3       ;BUILD EXPT WRITE ADDRESS
5405  017412  010337  021064        ER2:    MOV     R3,CADER        ;SAVE ADDRESS
5406  017416  012704  000007                MOV     #7,R4
5407  017422  012701  021066                MOV     #BAER,R1
5408  017426  005021                ER2A0:  CLR     (R1)+           ;CLEAR FLAGS
5409  017430  005304                        DEC     R4
5410  017432  001375                        BNE     ER2A0
5411  017434  020377  161054                CMP     R3,@BA          ;SEE IF ADDRESS OK
5412  017440  001402                        BEQ     ER2A1           ;IF SO: BR
5413  017442  005237  021066                INC     BAER            ;SET BUS ADDRESS ERROR
5414  017446  032737  000010  000674  ER2A1:  BIT     #10,MTC1        ;SEE IF WRITE OPER
5415  017454  001006                        BNE     ER2B            ;IF NOT: BR
5416  017456  005777  161034        ER2A:   TST     @FC             ;SEE IF FC=0
5417  017462  001441                        BEQ     ER3             ;IF SO: BR
```

```
 5418  017464  005237  021074                 INC     FCER            ;SET FC ERROR
 5419  017470  000436                         BR      ER3
 5420  017472  032737  000040  000674  ER2B:  BIT     #40,MTC1        ;SEE IF SPACE OPER
 5421  017500  001766                         BEQ     ER2A            ;IF SO: BR
 5422  017502  005737  000700                 TST     TMFLG           ;SEE IF TM TIME
 5423  017506  001011                         BNE     ER2D            ;IF SO: BR
 5424  017510  013703  000556                 MOV     FMCNT,R3
 5425  017514  005403                         NEG     R3              ;R3 = EXPT RECORD SIZE
 5426  017516  020377  160774          ER2C:  CMP     R3,@FC          ;SEE IF FC = EXPT
 5427  017522  001421                         BEQ     ER3             ;IF SO: BR
 5428  017524  005237  021074                 INC     FCER            ;SET FC ERROR FLAG
 5429  017530  000416                         BR      ER3
 5430  017532  032737  002000  000552  ER2D:  BIT     #2000,UDES      ;SEE IF PE
 5431  017540  001346                         BNE     ER2A            ;IF SO: BR
 5432  017542  032737  010000  000562         BIT     #10000,RDCMD    ;SEE IF READ REVERSE
 5433  017550  001003                         BNE     ER2E            ;IF SO: BR
 5434  017552  012703  000002                 MOV     #2,R3
 5435  017556  000757                         BR      ER2C            ;LOOK FOR EXPT = 2
 5436  017560  012703  000001          ER2E:  MOV     #1,R3
 5437  017564  000754                         BR      ER2C            ;GO CHECK FC FOR TM
 5438  017566  032777  160000  160714  ER3:   BIT     #160000,@C1     ;SEE IF COUNT ERROR
 5439  017574  001437                         BEQ     ER4
 5440  017576  017703  160716                 MOV     @CS,R3          ;GET CONT STATUS REG
 5441  017602  042703  000307                 BIC     #307,R3         ;MASK OUT IR,OR,UNIT NO. & SEE IF OTHER ERRORS
 5442  017606  001406                         BEQ     ER3A            ;IF NOT: BR
 5443  017610  005737  000700                 TST     TMFLG           ;SEE IF TAPE MARK TIME
 5444  017614  001425                         BEQ     ER3B            ;IF NOT: BR
 5445  017616  042703  001000                 BIC     #1000,R3        ;MASK MISSED TRANS & BR IF OTHER ERRORS
 5446  017622  001022                         BNE     ER3B
 5447  017624  032777  060000  160656  ER3A:  BIT     #60000,@C1      ;SEE IF EITHER TRE OR MCPE
 5448  017632  001420                         BEQ     ER4             ;IF NOT: BR
 5449  017634  005737  000700                 TST     TMFLG           ;SEE IF TM TIME
 5450  017640  001413                         BEQ     ER3B            ;IF NOT: BR
 5451  017642  017703  160656                 MOV     @ER,R3          ;GET ERROR REGISTER
 5452  017646  032737  000010  000552         BIT     #10,UDES        ;SEE IF EVEN PARITY
 5453  017654  001402                         BEQ     ER3A1           ;IF NOT: BR
 5454  017656  042703  000100                 BIC     #100,R3         ;MASK PAR
 5455  017662  042703  001000          ER3A1: BIC     #1000,R3        ;MASK FCE
 5456  017666  001402                         BEQ     ER4             ;IF NO ERRORS EXCEPT FCE: BR
 5457  017670  005237  021070          ER3B:  INC     CONER           ;SET CONT ERROR FLAG
 5458  017674  032777  040000  160620  ER4:   BIT     #40000,@DS      ;SEE IF DRIVE ERROR
 5459  017702  001420                         BEQ     ER6             ;IF NOT: BR
 5460  017704  005737  000700                 TST     TMFLG           ;SEE IF TAPE MARK TIME
 5461  017710  001413                         BEQ     ER4A            ;IF NOT: BR
 5462  017712  017703  160606                 MOV     @ER,R3          ;GET ER
 5463  017716  032737  000010  000552         BIT     #10,UDES        ;SEE IF EVEN PARITY
 5464  017724  001402                         BEQ     ER4A1           ;IF NOT: BR
 5465  017726  042703  000100                 BIC     #100,R3         ;MASK PAR
 5466  017732  042703  001000          ER4A1: BIC     #1000,R3        ;MASK OUT FCE & BRANCH IF
 5467  017736  001402                         BEQ     ER6             ;NO OTHER ERRORS
 5468  017740  005237  021072          ER4A:  INC     DRVER           ;SET DRIVER ERROR FLAG
 5469  017744  032737  002000  000552  ER6:   BIT     #2000,UDES
 5470  017752  001071                         BNE     ERPT            ;IF IN PE MODE: BR
 5471  017754  032777  020000  160626         BIT     #20000,@SWR     ;SEE IF NO DATA CHECK
 5472  017762  001065                         BNE     ERPT            ;IF NOT: BR (ALLOW READ OF UNKNOWN TAPES)
 5473  017764  032737  000040  000674         BIT     #40,MTC1        ;SEE IF WRITE OR READ OP
```

```
5474  017772  001461                          BEQ     ERPT            ;IF NOT: BR
5475  017774  005737  000700                  TST     TMFLG           ;SEE IF TAPE MARK TIME
5476  020000  001413                          BEQ     ER6A            ;IF NOT: BR
5477  020002  013737  015442  021106          MOV     EXCRC,CRCSV     ;SAVE CRC
5478  020010  013737  015444  021104          MOV     EXLRC,LRCSV     ;SAVE LRC
5479  020016  005037  015442                  CLR     EXCRC
5480  020022  012737  000023  015444          MOV     #23,EXLRC       ;SET CRC/LRC FOR TM
5481  020030  032737  000060  000552  ER6A:    BIT     #60,UDES        ;SEE IF FORMAT 14
5482  020036  001037                          BNE     ERPT            ;IF NOT: BR
5483  020040  017703  160464                  MOV     @CC,R3          ;GET CRC CHARACTER
5484  020044  042703  177000                  BIC     #177000,R3
5485  020050  023703  015442                  CMP     EXCRC,R3
5486  020054  001402                          BEQ     ER7             ;IF CRC GOOD: BR
5487  020056  005237  021100                  INC     CRCER           ;SET ERROR FLAG
5488  020062  017703  160446          ER7:    MOV     @MR,R3          ;GET LRC
5489  020066  000303                          SWAB    R3
5490  020070  005703                          TST     R3
5491  020072  100002                          BPL     ER10
5492  020074  052703  000400                  BIS     #400,R3
5493  020100  042703  177000          ER10:   BIC     #177000,R3
5494  020104  023703  015444                  CMP     EXLRC,R3
5495  020110  001412                          BEQ     ERPT            ;IF LRC GOOD: BR
5496  020112  010337  021102                  MOV     R3,ACTLRC       ;SAVE ACTUAL LRC
5497  020116  005237  021076                  INC     LRCER           ;SET LRC ERROR FLAG
5498  020122  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
5499  020130  001402                          BEQ     ERPT            ;IF NOT: BR
5500  020132  005037  021076                  CLR     LRCER           ;ELSE CLEAR LRC ERROR
5501  020136  012703  000006          ERPT:   MOV     #6,R3
5502  020142  005037  000710                  CLR     SERFL           ;CLEAR ERROR FLAG
5503  020146  005037  000724                  CLR     ERSAV
5504  020152  012704  021066                  MOV     #BAER,R4
5505  020156  005724                  ERPTT:  TST     (R4)+           ;SEE IF ANY ERROR
5506  020160  001004                          BNE     ERPTG           ;IF SO: BR
5507  020162  005303                          DEC     R3
5508  020164  001374                          BNE     ERPTT
5509  020166  000137  021020                  JMP     ERPX1
5510  020172  005237  000710          ERPTG:  INC     SERFL           ;SET ERROR FLAG
5511  020176  017737  160322  000724          MOV     @ER,ERSAV       ;SAVE ERROR REGISTER
5512  020204  032777  002000  160376          BIT     #2000,@SWR      ;SEE IF PRINT
5513  020212  001420                          BEQ     ERPT0           ;IF SO: BR
5514  020214  022737  000002  000714          CMP     #2,RTYFL        ;SEE IF READ RETRY
5515  020222  001006                          BNE     ERPTG1          ;IF NOT: BR
5516  020224  013703  000704                  MOV     RTCNT,R3
5517  020230  005203                          INC     R3              ;BUMP RETRY COUNT
5518  020232  020337  000604                  CMP     R3,RETRY        ;SEE IF LAST RETRY
5519  020236  001406                          BEQ     ERPT0           ;IF SO: BR
5520  020240  022737  000002  021072  ERPTG1: CMP     #2,DRVER        ;SEE IF TM STATUS ERROR
5521  020246  001402                          BEQ     ERPT0           ;IF SO: BR
5522  020250  000137  020700                  JMP     ERPX0
5523  020254  005237  000672          ERPT0:  INC     PFLG
5524  020260  004737  022672                  JSR     PC,PAPRT        ;PRINT HEADER
5525  020264  013704  000654                  MOV     EMADDR,R4
5526  020270  000004                          TYPE                    ;TYPE MSG
5527  020272  004737  021110                  JSR     PC,FRPRT        ;PRINT F OR R
5528  020276  005737  000700                  TST     TMFLG
5529  020302  001407                          BEQ     ERPT1
```

```
5530  020304  022737  026153  000654          CMP     #MSG54,EMADDR
5531  020312  001403                          BEQ     ERPT1
5532  020314  012704  026171                  MOV     #MSG56,R4       ;PRINT TM
5533  020320  000004                          TYPE                    ;TYPE MSG
5534  020322  005737  021070          ERPT1:  TST     CONER
5535  020326  001414                          BEQ     ERPT2           ;IF NO CONT ERROR: BR
5536  020330  012704  025045                  MOV     #MSG23,R4
5537  020334  000004                          TYPE                    ;TYPE MSG
5538  020336  017703  160146                  MOV     @C1,R3
5539  020342  104400                          TYPOCT                  ;PRINT CONTROL 1
5540  020344  012704  025072                  MOV     #MSG23D,R4      ;PRINT CS TAG
5541  020350  000004                          TYPE                    ;TYPE MSG
5542  020352  017703  160142                  MOV     @CS,R3
5543  020356  104400                          TYPOCT                  ;PRINT CONT STATUS
5544  020360  005737  021072          ERPT2:  TST     DRVER
5545  020364  001414                          BEQ     ERPT3           ;IF SO DRIVE ERROR: BR
5546  020366  012704  025100                  MOV     #MSG23E,R4
5547  020372  000004                          TYPE                    ;TYPE MSG
5548  020374  017703  160122                  MOV     @DS,R3
5549  020400  104400                          TYPOCT                  ;PRINT DRIVE STATUS
5550  020402  012704  025105                  MOV     #MSG23F,R4
5551  020406  000004                          TYPE                    ;TYPE MSG
5552  020410  017703  160110                  MOV     @ER,R3
5553  020414  104400                          TYPOCT                  ;PRINT DRIVE ERROR
5554  020416  005737  021066          ERPT3:  TST     BAER
5555  020422  001416                          BEQ     ERPT4           ;IF NO BA ERROR: BR
5556  020424  012704  025060                  MOV     #MSG23B,R4
5557  020430  000004                          TYPE                    ;TYPE MSG
5558  020432  017703  160056                  MOV     @BA,R3
5559  020436  104400                          TYPOCT                  ;PRINT BUS ADDRESS
5560  020440  012737  000255  000640          MOV     #255,TOB
5561  020446  004737  024064                  JSR     PC,TOG          ;PRINT /
5562  020452  013703  021064                  MOV     CADER,R3
5563  020456  104400                          TYPOCT                  ;PRINT EXPT BUS ADDRESS
5564  020460  005737  021074          ERPT4:  TST     FCER
5565  020464  001406                          BEQ     ERPT5           ;IF NO FC ERROR: BR
5566  020466  012704  025065                  MOV     #MSG23C,R4
5567  020472  000004                          TYPE                    ;TYPE MSG
5568  020474  017703  160016                  MOV     @FC,R3
5569  020500  104400                          TYPOCT                  ;PRINT FRAME COUNT
5570  020502  012704  025053          ERPT5:  MOV     #MSG23A,R4
5571  020506  000004                          TYPE                    ;TYPE MSG
5572  020510  017703  157776                  MOV     @WC,R3
5573  020514  104400                          TYPOCT                  ;PRINT WORD COUNT
5574  020516  005737  021100                  TST     CRCER
5575  020522  001420                          BEQ     ERPT5A          ;IF NO CRC ERROR: BR
5576  020524  012704  026216                  MOV     #MSG58,R4
5577  020530  000004                          TYPE                    ;TYPE MSG
5578  020532  017703  157772                  MOV     @CC,R3
5579  020536  042703  177000                  BIC     #177000,R3
5580  020542  104400                          TYPOCT                  ;PRINT ACTUAL CRC
5581  020544  012737  000255  000640          MOV     #255,TOB
5582  020552  004737  024064                  JSR     PC,TOG
5583  020556  013703  015442                  MOV     EXCRC,R3
5584  020562  104400                          TYPOCT                  ;PRINT EXPECTED CRC
5585  020564  005737  021076          ERPT5A: TST     LRCER
```

```
5586  020570  001416                        BEQ     ERPT6           ;IF NO LRC ERROR: BR
5587  020572  012704  026224                MOV     #MSG59,R4
5588  020576  000004                        TYPE                    ;TYPE MSG
5589  020600  013703  021102                MOV     ACTLRC,R3
5590  020604  104400                        TYPOCT                  ;PRINT ACTUAL LRC
5591  020606  012737  000255  000640        MOV     #255,TOB
5592  020614  004737  024064                JSR     PC,TOG
5593  020620  013703  015444                MOV     EXLRC,R3
5594  020624  104400                        TYPOCT                  ;PRINT EXPECTED LRC
5595  020626  005737  021072        ERPT6:  TST     DRVER
5596  020632  001421                        BEQ     ERPT7           ;IF NO DRIVE ERROR: BR
5597  020634  032737  002000  000552        BIT     #2000,UDES
5598  020642  001415                        BEQ     ERPT7           ;IF NO PE: BR
5599  020644  017704  157654                MOV     @ER,R4
5600  020650  042704  075477                BIC     #75477,R4       ;MASK OUT ALL BUT BITS 15,10,7,6
5601  020654  001410                        BEQ     ERPT7           ;IF NO CONDITIONALS SET: BR
5602  020656  012704  025117                MOV     #MSG23H,R4
5603  020662  000004                        TYPE                    ;TYPE MSG
5604  020664  017703  157640                MOV     @CC,R3
5605  020670  042703  177000                BIC     #177000,R3      ;MASK CC
5606  020674  104400                        TYPOCT                  ;PRINT CHECK CHARACTERS
5607  020676  000240        ERPT7:  NOP
5608  020700  005777  157704        ERPX0:  TST     @SWR            ;BRANCH IF NOT HALT ON ERROR
5609  020704  100012                        BPL     ERPX
5610  020706  000000                        HALT
5611  020710  005737  000672                TST     PFLG            ;SEE IF HAVE PRINTED
5612  020714  001006                        BNE     ERPX            ;IF SO: BR
5613  020716  032777  002000  157664        BIT     #2000,@SWR      ;SEE IF SHOULD PRINT
5614  020724  001002                        BNE     ERPX            ;IF NOT: BR
5615  020726  000137  020254                JMP     ERPTO           ;PRINT ERROR
5616  020732  005037  000672        ERPX:   CLR     PFLG
5617  020736  005737  000566                TST     CRCC            ;BRANCH IF CRC ERROR
5618  020742  001007                        BNE     1$              ;CORRECTION DESIRED
5619  020744  012777  000040  157546        MOV     #40,@CS         ;ELSE INIT
5620  020752  013777  000550  157540        MOV     DVN,@CS         ;RESET DRIVE NUMBER
5621  020760  000414                        BR      2$
5622  020762  012777  000011  157520  1$:   MOV     #11,@C1         ;DRIVE CLEAR
5623  020770  017704  157532                MOV     @AS,R4
5624  020774  010477  157526                MOV     R4,@AS          ;CLEAR AS
5625  021000  013704  000510                MOV     C1,R4
5626  021004  005204                        INC     R4
5627  021006  152714  000100                BISB    #100,(R4)       ;RESET TRE
5628  021012  013777  000552  157522  2$:   MOV     UDES,@TC        ;RESET TC
5629  021020  032737  000040  000674  ERPX1: BIT     #40,MTC1
5630  021026  001415                        BEQ     ERPX2           ;IF NOT READ/WRITE OP: BR
5631  021030  005737  000700                TST     TMFLG
5632  021034  001412                        BEQ     ERPX2           ;IF NOT TM TIME: BR
5633  021036  032737  002000  000552        BIT     #2000,UDES      ;CHECK UDES
5634  021044  001006                        BNE     ERPX2           ;BR IF PE
5635  021046  013737  021106  015442        MOV     CRCSV,EXCRC     ;RESTORE CRC
5636  021054  013737  021104  015444        MOV     LRCSV,EXLRC     ;RESTORE LRC
5637  021062  000207        ERPX2:  RTS     PC                      ;EXIT
5638  021064  000000        CADER:  0                               ;EXPT ADDRESS SAVE
5639  021066  000000        BAER:   0
5640  021070  000000        CONER:  0
5641  021072  000000        DRVER:  0
```

```
5642  021074  000000                    FCER:   0
5643  021076  000000                    LRCER:  0
5644  021100  000000                    CRCER:  0
5645  021102  000000                    ACTLRC: 0
5646  021104  000000                    LRCSV:  0
5647  021106  000000                    CRCSV:  0
5648
5649                                    ;*********************************************************
5650                                    ;F FOR FORWARD/R FOR REVERSE PRINT SUBROUTINE:
5651                                    ;
5652                                    ;THIS SUBROUTINE IS USED TO PRINT OUT THE
5653                                    ;TAPE DIRECTION USED WHEN ANY ERROR IS
5654                                    ;DETECTED IN STATUS OF READ OR WRITE, DATA, OR
5655                                    ;SPACING OPERATIONS.
5656                                    ;*********************************************************
5657
5658  021110  032737  000010  000674    FRPRT:  BIT     #10,MTC1        ;SEE IF WRITE COMMAND
5659  021116  001411                            BEQ     2$              ;IF SO: BR
5660  021120  012704  024753                    MOV     #MSG17,R4       ;SET TO TYPE REVERSE MSG
5661  021124  032737  000002  000674            BIT     #2,MTC1         ;BRANCH IF REVERSE
5662  021132  001002                            BNE     1$
5663  021134  012704  024750                    MOV     #MSG16,R4       ;SET FORWARD MESSAGE
5664  021140  000004                    1$:     TYPE                    ;TYPE MSG
5665  021142  000207                    2$:     RTS     PC              ;EXIT
5666
```

```
5668
5669                            ;*************************************************************
5670                            ;TAPE COMMAND EXECUTE SUBROUTINE:
5671                            ;
5672                            ;THIS SUBROUTINE IS USED TO EXECUTE THE
5673                            ;MAG TAPE COMMAND DESCRIBED BY THE READ
5674                            ;OR WRITE ROUTINE.  THE FINAL COMMAND IS
5675                            ;SENT TO THE DEVICE REGISTER ALONG WITH THE
5676                            ;INTERRUPT ENABLE AND GO BITS.
5677                            ;ONCE THE COMMAND IS ISSUED, AN INTERRUPT
5678                            ;TIMER IS STARTED AND IF NO INTERRUPT IS RETURNED
5679                            ;BEFORE TIME OUT OCCURS, AN ERROR WILL BE
5680                            ;PRINTED AND THE PROGRAM STOPPED.  TESTING MAY
5681                            ;BE RESUMED BY PRESSING THE CONTINUE SWITCH.
5682                            ;TWO INTERRUPT HANDLERS ARE USED, ONE FOR MAG TAPE
5683                            ;AND ANOTHER FOR TELETYPE (TTY).
5684                            ;UPON RECEIPT OF A MAG TAPE INTERRUPT, HOUSEKEEPING
5685                            ;IS PERFORMED AND CONTROL RETURNED TO THE CALLING
5686                            ;ROUTINE (READ,WRITE,ETC).
5687                            ;RECEIPT OF A TTY INTERRUPT WILL CAUSE THE
5688                            ;PROGRAM TO CHECK FOR ENTRY OF A CNTRL C CHARACTER.
5689                            ;IF NOT CNTRL C, THEN CONTINUATION OF WAIT FOR MAG
5690                            ;TAPE INTERRUPT IS RETURNED.  IF, HOWEVER, THE TTY
5691                            ;INTERRUPT WAS CAUSED BY ENTRY OF A CNTRL C,
5692                            ;THEN AT THIS TIME REQUESTS FOR NEW STALL VALUES
5693                            ;ARE PRINTED AND THE RESPONSES ENTERED.  RESUMPTION
5694                            ;OF TAPE INTERRUPT WAIT IS THEN RESUMED.
5695                            ;*************************************************************
5696
5697  021144  005037  000644        TAPG:   CLR     TEMP1
5698  021150  013777  000550  157342        MOV     DVN,@CS         ;SET DRIVE NO.
5699  021156  032777  010000  157336 TAPG0:  BIT     #10000,@DS      ;SEE IF HAVE MOL
5700  021164  001026                BNE     TAPG3           ;IF SO: BR
5701  021166  005237  000644        INC     TEMP1           ;SEE IF TIMED OUT
5702  021172  001371                BNE     TAPG0           ;WAIT FOR READY
5703  021174  004737  022672        JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
5704  021200  032737  000010  000674        BIT     #10,MTC1        ;SEE IF WRITE OP
5705  021206  001004                BNE     TAPG1           ;IF NOT: BR
5706  021210  012704  024610        MOV     #MSG5,R4
5707  021214  000004                TYPE                    ;TYPE MSG
5708  021216  000405                BR      TAPG2
5709  021220  012704  024615        TAPG1:  MOV     #MSG6,R4
5710  021224  000004                TYPE                    ;TYPE MSG
5711  021226  004737  021110        JSR     PC,FRPRT        ;PRINT F OR R
5712  021232  012704  025175        TAPG2:  MOV     #MSG25,R4
5713  021236  000004                TYPE                    ;TYPE MSG
5714  021240  000000                HALT
5715  021242  032777  020000  157252 TAPG3:  BIT     #20000,@DS      ;SEE IF PIP RESET
5716  021250  001411                BEQ     TAPG3F          ;IF SO: BR
5717  021252  004737  022672        JSR     PC,PAPRT        ;PRINT HEADER
5718  021256  012704  027300        MOV     #MSG116,R4
5719  021262  000004                TYPE                    ;TYPE MSG
5720  021264  032777  020000  157230 1$:     BIT     #20000,@DS
5721  021272  001374                BNE     1$              ;AWAIT PIP RESET
5722  021274  022737  000026  000674 TAPG3F: CMP     #26,MTC1        ;SEE IF WRITE TM
5723  021302  001003                BNE     TAPG3A          ;IF NOT: BR
```

```
5724  021304  012704  177777          MOV     #-1,R4          ;ELSE SET FC FOR -1
5725  021310  000406                  BR      TAPG3B
5726  021312  013704  000556  TAPG3A: MOV     FMCNT,R4
5727  021316  032704  000001          BIT     #1,R4
5728  021322  001401                  BEQ     TAPG3B
5729  021324  005304                  DEC     R4
5730  021326  000261  TAPG3B: SEC
5731  021330  006004                  ROR     R4              ;SET WC = FC/2 FOR NORMAL FORMAT
5732  021332  032737  000020  000552  BIT     #20,UDES        ;SEE IF CORE DUMP FORMAT
5733  021340  001402                  BEQ     TAPG3C          ;IF NOT: BR
5734  021342  000261                  SEC
5735  021344  006004                  ROR     R4              ;SET WC = FC/4 FOR CORE DUMP
5736  021346  010477  157140  TAPG3C: MOV     R4,@WC          ;SET WORD COUNT
5737  021352  012777  000011  157130  MOV     #11,@C1         ;DRIVE CLEAR
5738  021360  017777  157132  157130  MOV     @FC,@FC         ;RESET FC LOADED
5739  021366  005737  000570          TST     INTRF           ;SEE IF INTERCHANGE READ
5740  021372  001407                  BEQ     TAPG3D          ;IF NOT: BR
5741  021374  032737  000040  000674  BIT     #40,MTC1        ;SEE IF READ OP
5742  021402  001403                  BEQ     TAPG3D          ;IF NOT: BR
5743  021404  012777  000003  157122  MOV     #3,@MR          ;SET INTERCHANGE READ MAINT. MODE
5744  021412  013704  000674  TAPG3D: MOV     MTC1,R4         ;GET COMMAND
5745  021416  042704  177707          BIC     #177707,R4      ;MASK OP CODE
5746  021422  022704  000030          CMP     #30,R4          ;SEE IF SPACE OP CODE
5747  021426  001403                  BEQ     TAPG3E          ;IF SO: BR
5748  021430  012737  177740  000670  MOV     #-40,STAL       ;SET INTERRUPT DELAY MULT TO 40
5749  021436  052737  000101  000674  TAPG3E: BIS     #101,MTC1       ;SET INTERRUPT ENABLE AND GO
5750  021444  000240                  NOP
5751  021446  013777  000674  157034  MOV     MTC1,@C1        ;EXECUTE COMMAND
5752  021454  005077  157126          CLR     @PSW            ;CLEAR PRIORITY
5753  021460  005037  000644          CLR     TEMP1
5754  021464  005237  000644  TAPG4:  INC     TEMP1           ;SEE IF HAVE TIMED OUT
5755  021470  001375                  BNE     TAPG4           ;IF NOT: BR
5756  021472  005237  000670          INC     STAL
5757  021476  001372                  BNE     TAPG4           ;DO TIME DELAY MULTIPLIER
5758  021500  012777  000340  157100  TAPG5:  MOV     #340,@PSW       ;RESET PRIORITY
5759  021506  032777  002000  157074  BIT     #2000,@SWR      ;SEE IF SHOULD PRINT ERRORS
5760  021514  001012                  BNE     TAPG6           ;IF NOT: BR
5761  021516  004737  022672          JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
5762  021522  013704  000654          MOV     EMADDR,R4
5763  021526  000004                  TYPE                    ;TYPE MSG
5764  021530  004737  021110          JSR     PC,FRPRT        ;PRINT F OR R
5765  021534  012704  025155          MOV     #MSG24,R4
5766  021540  000004                  TYPE                    ;TYPE MSG
5767  021542  005777  157042  TAPG6:  TST     @SWR            ;BRANCH IF NOT HALT ON ERROR
5768  021546  100001                  BPL     TAPG7
5769  021550  000000                  HALT
5770  021552  000137  022010  TAPG7:  JMP     MTINTA          ;RETURN TO CALLING ROUTINE
5771
```

```
5773
5774                                            ;TTY INTERRUPT HANDLER
5775  021556  017746  157032          TTINT:  MOV     @TKB,-(SP)      ;GET CHARACTER
5776  021562  042716  000200                  BIC     #200,(SP)       ;STRIP PARITY BIT
5777  021566  122716  000003                  CMPB    #3,(SP)         ;BRANCH IF NOT ^C
5778  021572  001005                          BNE     1$
5779  021574  000005                          RESET                   ;RESET ALL I/O
5780  021576  005077  157004                  CLR     @PSW            ;CLEAR PSW
5781  021602  000137  000200                  JMP     @#200           ;RESTART PROGRAM
5782  021606  122716  000001          1$:     CMPB    #1,(SP)         ;BRANCH IF NOT ^A
5783  021612  001016                          BNE     2$
5784  021614  022737  000176  000610        CMP     #SWREG,SWR      ;BRANCH IF HARDWARE SWR IS INVOKED
5785  021622  001015                          BNE     3$
5786  021624  012737  177570  000610        MOV     #177570,SWR     ;INVOKE HARWARE SWR
5787  021632  004737  023376                  JSR     PC,.SAVE        ;SAVE REGISTERS ON THE STACK
5788  021636  012704  027424                  MOV     #MSG121,R4      ;TYPE 'HARDWARE SWR IN USE'
5789  021642  000004                          TYPE
5790  021644  004737  023420                  JSR     PC,.RESTORE     ;RESTORE REGISTERS
5791  021650  022716  000007          2$:     CMP     #7,(SP)         ;BRANCH IF NOT ^G
5792  021654  001005                          BNE     4$
5793  021656  012737  000176  000610  3$:     MOV     #SWREG,SWR      ;INVOKE SOFTWARE SWR
5794  021664  004737  023306                  JSR     PC,GTSWR        ;GET SWITCHES
5795  021670  022716  000002          4$:     CMP     #2,(SP)         ;BRANCH IF NOT ^B
5796  021674  001042                          BNE     6$
5797  021676  004737  023376                  JSR     PC,.SAVE        ;SAVE REGISTERS ON THE STACK
5798  021702  005237  014076                  INC     SCVFL           ;SET FLAG
5799  021706  004737  013650                  JSR     PC,TINP3A       ;GO CHECK CRC CORRECTION
5800  021712  032777  000040  156670        BIT     #40,@SWR        ;BRANCH IF NOT YOZZLING
5801  021720  001426                          BEQ     5$
5802  021722  012704  025750                  MOV     #MSG44,R4       ;REQUEST NEW YOZZLE STALL
5803  021726  000004                          TYPE                    ;TYPE MSG
5804  021730  013703  000602                  MOV     YSTAL,R3
5805  021734  104400                          TYPOCT                  ;PRINT PRESENT STALL
5806  021736  012705  000602                  MOV     #YSTAL,R5       ;SET ADDRESS OF YSTL
5807  021742  012701  000007                  MOV     #7,R1           ;SET NUMBER OF CHAR TO INPUT
5808  021746  012702  177777                  MOV     #-1,R2          ;SET MAXIMUM LIMIT
5809  021752  012703  002000                  MOV     #2000,R3        ;SET MINIMUM LIMIT
5810  021756  004737  023442                  JSR     PC,TTR          ;GO GET VALUE
5811  021762  004737  023420                  JSR     PC,.RESTORE     ;RESTORE REGISTERS
5812  021766  005726                          TST     (SP)+           ;POP CHARACTER OF THE STACK
5813  021770  012716  011266                  MOV     #YOZ,(SP)       ;RETURN TO 'YOZ'
5814  021774  000002                          RTI                     ;RETURN TO YOZ
5815  021776  004737  023420          5$:     JSR     PC,.RESTORE
5816  022002  005726                  6$:     TST     (SP)+           ;POP CHARACTER OFF THE STACK
5817  022004  000002                          RTI                     ;RETURN
5818
5819                                            ;MAG TAPE INTERRUPT HANDLER
5820  022006  000240                  MTINT:  NOP
5821  022010  042777  000037  156516  MTINTA: BIC     #37,@MR         ;CLEAR MAINT MODE
5822  022016  013716  000664                  MOV     RTRN,(SP)       ;SET RETURN TO (RTRN)
5823  022022  000002                          RTI                     ;RETURN
```

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

022024  012704  026722        ASEQ:   MOV     #MSG104,R4
022030  000004                        TYPE                    ;TYPE MSG
022032  012705  000744                MOV     #ASEQCF,R5      ;SET ADDRESS OF ENTRY
022036  012701  000002                MOV     #2,R1           ;SET SIZE OF ENTRY
022042  012702  000001                MOV     #1,R2           ;SET UPPER LIMIT
022046  012703  000000                MOV     #0,R3           ;SET LOWER LIMIT
022052  004737  023442                JSR     PC,TTR          ;GO GET INPUT
022056  005037  000740        ASEQ0:  CLR     ADRVN           ;CLEAR DRV NUM
022062  004737  022212        ASEQ1:  JSR     PC,HRDS         ;GO SELECT HARDWARE CONFIGURATION
022066  012704  026654                MOV     #MSG101,R4
022072  000004                        TYPE                    ;TYPE MSG
022074  012704  026703                MOV     #MSG102,R4
022100  000004                        TYPE                    ;TYPE MSG
022102  013703  000740                MOV     ADRVN,R3
022106  104400                        TYPOCT                  ;PRINT TM03
022110  012704  026712                MOV     #MSG103,R4
022114  000004                        TYPE                    ;TYPE MSG
022116  012700  000746                MOV     #UN1,R0         ;POINT TO START OF SLAVE TABLE
022122  005710        ASEQ2:  TST     (R0)            ;SEE IF END
022124  100403                BMI     ASEQ3           ;IF SO: BR
022126  012003                MOV     (R0)+,R3
022130  104400                TYPOCT                  ;PRINT SLAVE TABLE
022132  000773                BR      ASEQ2           ;DO ALL
022134  004737  022376        ASEQ3:  JSR     PC,AMOD1        ;GO DO MODE 1(NRZ)
022140  012700  001050                MOV     #STTBL,R0       ;POINTER TO THE TABLE
022144  012701  001720                MOV     #ENDTBL-STTBL,R1 ;AND #OF BYTES IN TABLE
022150  105020                CLRB    (R0)+           ;CLEAR STATISTIC COUNTER
022152  005301                DEC     R1
022154  001375                BNE     .-4
022156  004737  022536                JSR     PC,AMOD2        ;GO DO MODE 2(PE)
022162  022737  000007  000740 ASEQ4:  CMP     #7,ADRVN        ;SEE IF DONE ALL DRIVES
022170  001403                BEQ     ASEQX           ;IF SO: BR
022172  005237  000740                INC     ADRVN           ;BUMP DRIVE NUMBER
022176  000731                BR      ASEQ1           ;CONTINUE
022200  005737  000744        ASEQX:  TST     ASEQCF          ;SEE IF CONTINUOUS AUTO SEQ
022204  001324                BNE     ASEQ0           ;++B CONTINUE TESTING
022206  000137  005004                JMP     TEND
```

```
5872
5873                                        ;SUBROUTINE TO SELECT AUTO SEQUENCE HARDWARE********
5874
5875  022212  005037  005054          HRDS: CLR   REOTC           ;CLEAR EOT UNIT CNTR
5876  022216  005037  000644                CLR   TEMP1
5877  022222  012777  000040  156270        MOV   #40,@CS         ;INIT
5878  022230  013777  000740  156262        MOV   ADRVN,@CS       ;SET DRIVE
5879  022236  032777  010000  156254        BIT   #10000,@CS      ;TEST FOR NON-EXISTANT DRIVE
5880  022244  001403                        BEQ   2$              ;IF DRIVE AVAIL: BR
5881  022246  005726                  1$:   TST   (SP)+           ;RESET STACK POINTER
5882  022250  000137  022162              JMP   ASEQ4           ;GO SEE IF TRIED ALL DRIVES
5883  022254  005000                  2$:   CLR   R0
5884  022256  012701  000746              MOV   #UN1,R1         ;SET START OF SLAVE TABLE
5885  022262  005737  003040              TST   CHNFLG          ;BRANCH IF NOT IN CHAIN MODE
5886  022266  001410                        BEQ   3$
5887  022270  122737  000006  000041        CMPB  #6,@#41         ;BRANCH IF NOT LOADED VIA TMDP
5888  022276  001004                        BNE   3$
5889  022300  005737  000740              TST   ADRVN           ;BRANCH IF NOT DRIVE 0
5890  022304  001001                        BNE   3$
5891  022306  005200                        INC   R0              ;DO NOT TEST SLAVE 0
5892  022310  010077  156226          3$:   MOV   R0,@TC          ;SELECT SLAVE
5893  022314  032777  010000  156200        BIT   #10000,@DS      ;SEE IF SLAVE AVAIL FOR TEST(MOL)
5894  022322  001403                        BEQ   4$              ;IF NOT: BR
5895  022324  005237  000644              INC   TEMP1           ;SET SLAVE FOUND FLAG
5896  022330  010021                        MOV   R0,(R1)+        ;LOAD SLAVE TABLE
5897  022332  005200                  4$:   INC   R0              ;STEP TO NEXT SLAVE
5898  022334  022700  000010              CMP   #10,R0          ;BRANCH IF ALL SLAVE NOT DONE
5899  022340  001363                        BNE   3$
5900  022342  005737  000644          5$:   TST   TEMP1           ;SEE IF FOUND ANY SLAVES
5901  022346  001737                        BEQ   1$              ;IF NOT: BR
5902  022350  013737  000644  005054        MOV   TEMP1,REOTC     ;SET NUMBER OF UNITS
5903  022356  000337  000644              SWAB  TEMP1
5904  022362  053737  000644  005054        BIS   TEMP1,REOTC     ;SET EOT CNTR
5905  022370  012711  177777              MOV   #-1,(R1)        ;TERMINATE SLAVE TABLE
5906  022374  000207                        RTS   PC              ;RETURN TO SEQ
```

```
5908
5909                                        ;SUBROUTINE TO SELECT NRZ AUTO TEST MODE**********
5910
5911  022376  005037  000656          AMOD1:  CLR     BLCNTR          ;ASSURE BLOCK COUNTER IS 0
5912  022402  012701  000746                  MOV     #UN1,R1         ;GET START OF SLAVE TABLE
5913  022406  052721  001700          1$:     BIS     #1700,(R1)+     ;SET ALL SLAVE TO NRZ,NORM,ODD
5914  022412  022711  177777                  CMP     #-1,(R1)        ;LOOP UNTIL REACED END OF TABLE
5915  022416  001373                          BNE     1$
5916  022420  004737  005070                  JSR     PC,RWNDA        ;GO REWIND ALL AVAIL SLAVES
5917  022424  012737  000006  000742          MOV     #6,ABLCNT       ;SET NUMBER OF BLOCKS FOR MODE 1
5918  022432  012737  174000  000556          MOV     #-4000,FMCNT    ;SET FC = 4000
5919  022440  012737  000100  000554          MOV     #100,RCNT       ;SET REC CNTR = 100
5920  022446  013737  000740  000550          MOV     ADRVN,DVN       ;SELECT DRIVE
5921  022454  012737  000001  000560          MOV     #1,PATRN        ;SELECT PATTERN 1
5922  022462  005037  000564                  CLR     TMEX            ;ASSURE NO TMK
5923  022466  005037  000570                  CLR     INTRF           ;ASSURE NORMAL READ
5924  022472  004737  003346                  JSR     PC,STAUTO       ;GO DO AUTO MODE 1
5925  022476  012737  000010  000560          MOV     #10,PATRN       ;SELECT PATTERN 10
5926  022504  004737  003346                  JSR     PC,STAUTO       ;GO DO PATTERN 10
5927  022510  012737  000014  000560          MOV     #14,PATRN       ;SELECT PATTERN 14
5928  022516  004737  003346                  JSR     PC,STAUTO
5929  022522  012737  177777  000560  3$:     MOV     #-1,PATRN       ;SELECT AUTO RANDOM DATA
5930  022530  004737  003346                  JSR     PC,STAUTO
5931  022534  000207                          RTS     PC              ;RETURN TO SEQ
```

```
5933
5934                                        ;SUBROUTINE TO SELECT PE AUTO TEST MODE**********
5935
5936  022536  005037  000656        AMOD2:  CLR     BLCNTR          ;CLEAR BLOCK CNTR
5937  022542  012701  000746                MOV     #UN1,R1         ;SET START OF SLAVE TABLE
5938  022546  042711  001700        1$:     BIC     #1700,(R1)      ;CLEAR NRZ
5939  022552  052721  002300                BIS     #2300,(R1)+     ;SET TO PE NORM, ODD
5940  022556  022711  177777                CMP     #-1,(R1)        ;LOOP UNTIL END OF TABLE
5941  022562  001371                        BNE     1$
5942  022564  004737  005070                JSR     PC,RWNDA        ;REWIND ALL SLAVES
5943  022570  012737  000006  000742        MOV     #6,ABLCNT       ;SET AUTO BLOCK COUNT
5944  022576  012737  174000  000556        MOV     #-4000,FMCNT    ;SET FC = 4000
5945  022604  012737  000100  000554        MOV     #100,RCNT       ;SET REC CNTR TO 100
5946  022612  012737  000010  000560        MOV     #10,PATRN       ;SELECT PATTERN 10
5947  022620  004737  003346                JSR     PC,STAUTO       ;GO DO AUTO SEQ
5948  022624  012737  000014  000560        MOV     #14,PATRN       ;SELECT PATTERN 14
5949  022632  004737  003346                JSR     PC,STAUTO
5950  022636  012737  000015  000560        MOV     #15,PATRN       ;SELECT PATTERN 15
5951  022644  004737  003346                JSR     PC,STAUTO
5952  022650  012737  177777  000742        MOV     #-1,ABLCNT      ;FORCE TO END OF TAPE
5953  022656  012737  177777  000560        MOV     #-1,PATRN       ;SELECT AUTO RANDOM DATA
5954  022664  004737  003346                JSR     PC,STAUTO
5955  022670  000207                3$:     RTS     PC              ;RETURN TO SEQ
5956
5957
```

```
5959
5960                                    ;*********************************************************
5961                                    ;ERROR HEADER PRINT SUBROUTINE:
5962                                    ;
5963                                    ;THIS ROUTINE IS USED TO PRINT OUT A HEADER
5964                                    ;WITH EACH ERROR MESSAGE.  THE PRINT IS IN TWO
5965                                    ;LINES AND CONTAINS THE FOLLOWING INFORMATION.
5966                                    ;LINE 1:  DRIVE NO. SLAVE NO. DENSITY PARITY FORMAT
5967                                    ;LINE 2:  CURRENT BLOCK NUMBER, RECORD NUMBER IN
5968                                    ;WHICH THE ERROR OCCURED PLUS THE TOTAL NUMBER
5969                                    ;OF RECORDS IN THIS BLOCK, THE RECORD SIZE (NUMBER
5970                                    ;OF CHARACTERS), AND THE ERROR TYPE (READ,WRITE, SPACE, ETC)
5971                                    ;PLUS THE TAPE DIRECTION (FORWARD OR REVERSE).
5972                                    ;ALL NUMBERS ARE IN OCTAL.
5973                                    ;*********************************************************
5974
5975  022672  012704  024666       PAPRT:  MOV     #MSG12,R4
5976  022676  000004                       TYPE                    ;TYPE MSG
5977  022700  013703  000550               MOV     DVN,R3
5978  022704  104400                       TYPOCT                  ;PRINT DRIVE NUMBER
5979  022706  012704  024652               MOV     #MSG11,R4
5980  022712  000004                       TYPE                    ;TYPE MSG
5981  022714  013703  000552               MOV     UDES,R3
5982  022720  042703  177770               BIC     #177770,R3
5983  022724  104400                       TYPOCT                  ;PRINT UNIT NUMBER
5984  022726  012704  026232               MOV     #MSG60,R4
5985  022732  000004                       TYPE                    ;TYPE MSG
5986  022734  013703  000552               MOV     UDES,R3
5987  022740  000303                       SWAB    R3
5988  022742  042703  177770               BIC     #177770,R3
5989  022746  104400                       TYPOCT                  ;PRINT DENSITY
5990  022750  012704  026236               MOV     #MSG61,R4
5991  022754  000004                       TYPE                    ;TYPE MSG
5992  022756  005003                       CLR     R3
5993  022760  032737  000010  000552       BIT     #10,UDES
5994  022766  001402                       BEQ     PAPRT0
5995  022770  012703  000001               MOV     #1,R3
5996  022774  104400               PAPRT0: TYPOCT                  ;PRINT PARITY
5997  022776  012704  026242               MOV     #MSG62,R4
5998  023002  000004                       TYPE                    ;TYPE MSG
5999  023004  013703  000552               MOV     UDES,R3
6000  023010  000241                       CLC
6001  023012  006003                       ROR     R3
6002  023014  006003                       ROR     R3
6003  023016  006003                       ROR     R3              ;PONTION FORMAT
6004  023020  006003                       ROR     R3
6005  023022  042703  177760               BIC     #177760,R3
6006  023026  104400                       TYPOCT                  ;PRINT FORMAT
6007  023030  012704  024627               MOV     #MSG8,R4
6008  023034  000004                       TYPE                    ;TYPE MSG
6009  023036  032777  000400  155544       BIT     #400,@SWR       ;SEE IF RANDOM DATA
6010  023044  001406                       BEQ     PAPRTB          ;IF NOT: BR
6011  023046  012737  000122  000640 PAPRTA: MOV     #122,TOB
6012  023054  004737  024064               JSR     PC,TOG          ;PRINT R
6013  023060  000411                       BR      PAPRTD
6014  023062  005737  000736       PAPRTB: TST     ASEQF           ;SEE IF AUTO SEQ
```

```
6015  023066  001403                          BEQ     PAPRTC          ;IF NOT: BR
6016  023070  005737  000560                  TST     PATRN           ;SEE IF AUTO RANDOM
6017  023074  100764                          BMI     PAPRTA          ;IF SO: BR
6018  023076  013703  000560          PAPRTC: MOV     PATRN,R3
6019  023102  104400                          TYPOCT                  ;PRINT PATRN NUMBER
6020  023104  012704  024704          PAPRTD: MOV     #MSG13,R4
6021  023110  000004                          TYPE                    ;TYPE MSG
6022  023112  013703  000656                  MOV     BLCNTR,R3
6023  023116  104400                          TYPOCT                  ;PRINT NUMBER
6024  023120  012704  024712                  MOV     #MSG14,R4
6025  023124  000004                          TYPE                    ;TYPE MSG
6026  023126  010003                          MOV     R0,R3
6027  023130  032737  000010  000674          BIT     #10,MTC1        ;SEE IF WRITE OPERATION
6028  023136  001404                          BEQ     PAPRT1          ;IF SO: BR
6029  023140  032737  010000  000562          BIT     #10000,RDCMD    ;SEE IF READ REVERSE
6030  023146  001016                          BNE     PAPRT3          ;IF SO: BR
6031  023150  013703  000554          PAPRT1: MOV     RCNT,R3
6032  023154  005737  000700                  TST     TMFLG           ;SEE IF TAPE MARK TIME
6033  023160  001010                          BNE     PAPRT2          ;IF SO: BR
6034  023162  022737  012106  000664          CMP     #B1,RTRN
6035  023170  001003                          BNE     PAPRTY          ;IF NOT BACK SPACE: BR
6036  023172  005737  000714                  TST     RTYFL
6037  023176  001402                          BEQ     PAPRT3          ;IF NOT RETRY: BR
6038  023200  160003          PAPRTY: SUB     R0,R3           ;GET RECORD NUMBER
6039  023202  005203          PAPRT2: INC     R3
6040  023204  104400          PAPRT3: TYPOCT                  ;PRINT RECORD NUMBER
6041  023206  012737  000055  000640          MOV     #55,TOB         ;LOAD DASH (-)
6042  023214  004737  024064                  JSR     PC,TOG          ;PRINT DASH (-)
6043  023220  013703  000554                  MOV     RCNT,R3
6044  023224  104400                          TYPOCT                  ;PRINT RECORD COUNT
6045  023226  012704  024622                  MOV     #MSG7,R4
6046  023232  000004                          TYPE                    ;TYPE MSG
6047  023234  013703  000556                  MOV     FMCNT,R3        ;GET CHARACTER COUNT
6048  023240  005403                          NEG     R3              ;FORM TWO'S COMPLEMENT
6049  023242  104400                          TYPOCT                  ;PRINT RECORD SIZE
6050  023244  012737  000001  000666          MOV     #1,HDRFL        ;SET HEADER FLAG
6051  023252  000207                          RTS     PC              ;RETURN
6052
```

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

023254  063737  000630  000626  RANG:   ADD     RANSAV,RANBAS
023262  063737  000626  000630          ADD     RANBAS,RANSAV   ;GET NEW NUMBER
023270  023701  000630                  CMP     RANSAV,R1       ;SEE IF NUMBER TOO BIG
023274  101367                          BHI     RANG            ;IF SO: BR
023276  020237  000630                  CMP     R2,RANSAV       ;SEE IF NUMBER TOO SMALL
023302  101364                          BHI     RANG            ;IF SO: BR
023304  000207                          RTS     PC              ;EXIT

                                    ;SUBROUTINE TO GET NEW SOFTWARE SWR

023306  022737  000176  000610  GTSWR:  CMP     #SWREG,SWR      ;BRANCH IF SOFTWARE SWR
023314  001027                          BNE     1$              ;NOT INVOKED
023316  004737  023376                  JSR     PC,.SAVE        ;SAVE REGISTERS ON THE STACK
023322  012704  024544                  MOV     #$MSWR,R4       ;TYPE 'SWR = '
023326  000004                          TYPE                    ;TYPE MSG
023330  017703  155254                  MOV     @SWR,R3         ;GET CURRENT SWR
023334  104400                          TYPOCT
023336  012704  024554                  MOV     #$MNEW,R4       ;ASK FOR NEW SETTING
023342  000004                          TYPE                    ;TYPE MSG
023344  013705  000610                  MOV     SWR,R5          ;TTR ROUTINE RETURNS VALUE TO (R5)
023350  012701  000007                  MOV     #7,R1           ;LIMIT RESPONSE TO 7 CHARS
023354  012702  177777                  MOV     #177777,R2      ;BETWEEN 0 AND 177777
023360  012703  000000                  MOV     #0,R3
023364  004737  023442                  JSR     PC,TTR          ;GET RESPONSE
023370  004737  023420                  JSR     PC,.RESTORE     ;RESTORE REGISTERS
023374  000207              1$:         RTS     PC              ;RETURN

                                    ;;ROUTINE TO SAVE REGISTERS ON THE STACK
 (1)    023376  010546              .SAVE:  MOV     %5,-(SP)        ;;R5 IS SAVED AT 12(SP)
 (1)    023400  010446                      MOV     %4,-(SP)        ;;R4 IS SAVED AT 10(SP)
 (1)    023402  010346                      MOV     %3,-(SP)        ;;R3 IS SAVED AT 6(SP)
 (1)    023404  010246                      MOV     %2,-(SP)        ;;R2 IS SAVED AT 4(SP)
 (1)    023406  010146                      MOV     %1,-(SP)        ;;R1 IS SAVED AT 2(SP)
 (1)    023410  010046                      MOV     %0,-(SP)        ;;R0 IS SAVED AT (SP)
 (1)    023412  016646  000014              MOV     14(SP),-(SP)    ;;PUSH RETURN PC ON THE STACK
 (1)    023416  000207                      RTS     PC              ;;RETURN TO CALLER
 (1)

                                    ;;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
 (1)    023420  012666  000014      .RESTORE:MOV    (SP)+,14(SP)    ;;STORE RETURN PC ON STACK
 (1)    023424  012600                      MOV     (SP)+,%0
 (1)    023426  012601                      MOV     (SP)+,%1
 (1)    023430  012602                      MOV     (SP)+,%2
 (1)    023432  012603                      MOV     (SP)+,%3
 (1)    023434  012604                      MOV     (SP)+,%4
 (1)    023436  012605                      MOV     (SP)+,%5
 (1)    023440  000207                      RTS     PC              ;;RETURN
 (1)
```

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

023442  010146                  TTR:    MOV     R1,-(SP)        ;SAVE CHAR COUNT
023444  011601                  10$:    MOV     (SP),R1         ;RESTORE CHAR COUNT (FOR ^U)
023446  005037  000644                  CLR     TEMP1           ;CLEAR FIRST CHARACTER FLAG
023452  005000                          CLR     R0
023454  004737  023674          1$:     JSR     PC,TTIN         ;GO READ CHARACTER
023460  122737  000003  000642          CMPB    #3,TIB          ;BRANCH IF NOT ^C
023466  001003                          BNE     11$
023470  000005                          RESET
023472  000137  000200                  JMP     @#200   ;RESTART AT 200
023476  122737  000015  000642  11$:    CMPB    #15,TIB         ;SEE IF CR
023504  001004                          BNE     2$              ;IF NOT: BR
023506  005737  000644                  TST     TEMP1           ;SEE IF FIRST CHARACTER
023512  001457                          BEQ     9$              ;IF SO: BR
023514  000451                          BR      6$              ;ELSE GO LOAD VALUE
023516  122737  000025  000642  2$:     CMPB    #25,TIB         ;BRANCH IF NOT CONTROL U
023524  001004                          BNE     21$
023526  012704  025237                  MOV     #MSG28,R4       ;TYPE <CR><LF>
023532  000004                          TYPE                    ;TYPE MSG
023534  000743                          BR      10$
023536  122737  000177  000642  21$:    CMPB    #177,TIB                ;BRANCH IF NOT 'RUBOUT'
023544  001011                          BNE     3$
023546  000241                          CLC                     ;REMOVE LAST CHARACTER
023550  006000                          ROR     R0
023552  006200                          ASR     R0
023554  006200                          ASR     R0
023556  012704  027354                  MOV     #MSG118,R4      ;TYPE '\'
023562  000004                          TYPE                    ;TYPE MSG
023564  005201                          INC     R1              ;DEC CHAR RECEIVED COUNT
023566  000732                          BR      1$              ;GET NEXT CHARACTER
023570  122737  000060  000642  3$:     CMPB    #60,TIB         ;SEE IF CHAR IS LESS THAN 0
023576  101027                          BHI     TINER
023600  122737  000070  000642  4$:     CMPB    #70,TIB         ;SEE IF CHAR IS GREATER THAN 7
023606  101423                          BLOS    TINER
023610  005237  000644          5$:     INC     TEMP1           ;SET FIRST CHARACTER FLAG
023614  006300                          ASL     R0
023616  006300                          ASL     R0              ;SHIFT 3 LEFT
023620  006300                          ASL     R0
023622  042737  177770  000642          BIC     #177770,TIB     ;STRIP ASCII
023630  053700  000642                  BIS     TIB,R0          ;LOAD CHARACTER
```

```
6149  023634  005301                  DEC     R1              ;SEE IF DONE
6150  023636  001306                  BNE     1$              ;IF NOT: BR
6151  023640  020002          6$:     CMP     R0,R2           ;SEE IF EXCEEDED MAXIMUM LIMIT
6152  023642  101005                  BHI     TINER
6153  023644  020300          7$:     CMP     R3,R0           ;SEE IF BELOW MINIMUM LIMIT
6154  023646  101003                  BHI     TINER
6155  023650  010015          8$:     MOV     R0,(R5)         ;LOAD VALUE
6156  023652  005726          9$:     TST     (SP)+           ;POP CHAR COUNT OFF STACK
6157  023654  000207                  RTS     PC              ;EXIT
6158
6159  023656  012704  025744  TINER:  MOV     #MSG43,R4
6160  023662  000004                  TYPE                    ;TYPE MSG
6161  023664  005726                  TST     (SP)+           ;POP CHAR COUNT OFF STACK
6162  023666  162716  000020          SUB     #20,(SP)        ;RESET SP TO START OF VALUE ROUTINE
6163  023672  000207                  RTS     PC              ;REDO VALUE ENTRY
```

```
6165
6166                                        ;TTY READ SUBROUTINE**********
6167
6168  023674  017746  154706          TTIN: MOV     @PSW,   -(SP)   ;SAVE CURRENT PSW
6169  023700  052777  000340  154700        BIS     #340,   @PSW    ;SET TO BR7 TO PREVENT INTRPT
6170  023706  005277  154700                INC     @TKS
6171  023712  105777  154674          1$:   TSTB    @TKS
6172  023716  100375                        BPL     1$
6173  023720  012677  154662                MOV     (SP)+,  @PSW    ;RESTORE PSW ,OK TO INTRPT NOW
6174  023724  017737  154664  000642        MOV     @TKB,TIB
6175  023732  042737  000200  000642        BIC     #200,TIB        ;STRIP PARITY BIT
6176  023740  013737  000642  000640        MOV     TIB,TOB         ;MOVE CHAR TO TTY OUTPUT BFR
6177  023746  004737  024064                JSR     PC,TOG          ;ECHO CHARACTER
6178  023752  000207                        RTS     PC
6179
6180                                        ;TTY OUTPUT SUBROUTINE************
6181
6182  023754  112437  000640          TTOUT: MOVB   (R4)+,TOB
6183  023760  105737  000640                TSTB    TOB
6184  023764  001436                        BEQ     3$
6185  023766  122737  000045  000640        CMPB    #45,TOB
6186  023774  001407                        BEQ     1$
6187  023776  122737  000041  000640        CMPB    #41,TOB
6188  024004  001436                        BEQ     TBELL           ;DO BELL
6189  024006  004737  024064                JSR     PC,TOG
6190  024012  000760                        BR      TTOUT
6191  024014  112737  000015  000640  1$:   MOVB    #15,TOB
6192  024022  004737  024064                JSR     PC,TOG
6193  024026  012703  000006                MOV     #6,R3
6194  024032  005037  000640          2$:   CLR     TOB
6195  024036  004737  024064                JSR     PC,TOG
6196  024042  005303                        DEC     R3
6197  024044  001372                        BNE     2$              ;DO FILLERS
6198  024046  112737  000012  000640        MOVB    #12,TOB
6199  024054  004737  024064                JSR     PC,TOG
6200  024060  000735                        BR      TTOUT
6201  024062  000002                  3$:   RTI                     ;RETURN
6202
6203  024064  105777  154526          TOG:  TSTB    @TPS
6204  024070  100375                        BPL     TOG
6205  024072  113777  000640  154520        MOVB    TOB,@TPB
6206  024100  000207                        RTS     PC              ;RETURN
6207
6208  024102  012703  000002          TBELL: MOV    #2,R3
6209  024106  012737  000007  000640  1$:   MOV     #7,TOB
6210  024114  004737  024064                JSR     PC,TOG
6211  024120  005303                        DEC     R3
6212  024122  001371                        BNE     1$
6213  024124  000713                        BR      TTOUT
6214
6215
```

```
6217                                    ;OCTAL OUTPUT SUBROUTINE************
6218
6219  024126  005037  024340    OCTP:   CLR     OFL             ;CLEAR FLAG FOR LEADING ZERO
6220  024132  010304                    MOV     R3,R4           ;SEE IF NUMBER IS ZERO
6221  024134  001003                    BNE     OCTP0           ;IF NOT ZERO: BR
6222  024136  004737  024320            JSR     PC,OCTPG1       ;ELSE PRINT ZERO
6223  024142  000447                    BR      OCTP3           ;SPACE AND EXIT
6224  024144  005704            OCTP0:  TST     R4              ;BRANCH IF MSD = 0
6225  024146  100006                    BPL     OCTP1
6226  024150  012704  000001            MOV     #1,R4
6227  024154  004737  024276            JSR     PC,OCTPG        ;PRINT 1
6228  024160  000137  024172            JMP     OCTP2
6229  024164  005004            OCTP1:  CLR     R4
6230  024166  004737  024276            JSR     PC,OCTPG        ;PRINT 0
6231  024172  010304            OCTP2:  MOV     R3,R4
6232  024174  006004                    ROR     R4
6233  024176  006004                    ROR     R4
6234  024200  006004                    ROR     R4              ;POSITION DIGIT
6235  024202  006004                    ROR     R4
6236  024204  000304                    SWAB    R4
6237  024206  004737  024276            JSR     PC,OCTPG        ;PRINT DIGIT 2
6238  024212  010304                    MOV     R3,R4
6239  024214  006004                    ROR     R4
6240  024216  000304                    SWAB    R4
6241  024220  004737  024276            JSR     PC,OCTPG        ;PRINT DIGIT 3
6242  024224  010304                    MOV     R3,R4
6243  024226  006104                    ROL     R4
6244  024230  006104                    ROL     R4
6245  024232  000304                    SWAB    R4
6246  024234  004737  024276            JSR     PC,OCTPG        ;PRINT DIGIT 4
6247  024240  010304                    MOV     R3,R4
6248  024242  006004                    ROR     R4
6249  024244  006004                    ROR     R4
6250  024246  006004                    ROR     R4
6251  024250  004737  024276            JSR     PC,OCTPG
6252  024254  010304                    MOV     R3,R4
6253  024256  004737  024276            JSR     PC,OCTPG        ;PRINT DIGIT 5
6254  024262  012737  000240  000640 OCTP3:  MOV     #240,TOB
6255  024270  004737  024064            JSR     PC,TOG          ;PRINT SPACE
6256  024274  000002                    RTI                     ;EXIT
6257  024276  042704  177770    OCTPG:  BIC     #177770,R4
6258  024302  001004                    BNE     OCTPG0
6259  024304  005737  024340            TST     OFL
6260  024310  001001                    BNE     OCTPG0
6261  024312  000207                    RTS     PC
6262
6263  024314  005237  024340    OCTPG0: INC     OFL
6264  024320  052704  000260    OCTPG1: BIS     #260,R4
6265  024324  010437  000640            MOV     R4,TOB
6266  024330  004737  024064            JSR     PC,TOG
6267  024334  010304                    MOV     R3,R4
6268  024336  000207                    RTS     PC
6269  024340  000000            OFL:    0                       ;FIRST CHAR FLAG
6270
```

```
6272
6273                                            ;DATA CHARACTER OUTPUT SUBROUTINE***********
6274
6275  024342  012704  000010          DOUT:   MOV     #10,R4          ;SET NUMBER TO PRINT
6276  024346  110337  000640                  MOVB    R3,@DB
6277  024352  105777  154240          1$:     TSTB    @TPS
6278  024356  100375                          BPL     1$
6279  024360  105737  000640                  TSTB    DB
6280  024364  100004                          BPL     2$
6281  024366  012777  000061  154224          MOV     #061,@TPB
6282  024374  000403                          BR      3$
6283  024376  012777  000060  154214  2$:     MOV     #060,@TPB
6284  024404  006337  000640          3$:     ASL     DB
6285  024410  005304                          DEC     R4
6286  024412  001357                          BNE     1$
6287  024414  000207                          RTS     PC
6288
6289  024416  013703  000650          DOUTD:  MOV     TEMP3,R3
6290  024422  000303                          SWAB    R3
6291  024424  004737  024342                  JSR     PC,DOUT
6292  024430  013703  000650                  MOV     TEMP3,R3
6293  024434  004737  024342                  JSR     PC,DOUT
6294  024440  000207                          RTS     PC
6295
6296                                            ;TU45 SERIAL NUMBER PRINT SUBROUTINE*******
6297
6298  024442  010304          SNPT:   MOV     R3,R4
6299  024444  000304                  SWAB    R4
6300  024446  006004                  ROR     R4
6301  024450  006004                  ROR     R4
6302  024452  006004                  ROR     R4
6303  024454  006004                  ROR     R4
6304  024456  004737  024520          JSR     PC,SNPG         ;PRINT FIRST DIGIT
6305  024462  010304                  MOV     R3,R4
6306  024464  000304                  SWAB    R4
6307  024466  004737  024520          JSR     PC,SNPG         ;PRINT SECOND DIGIT
6308  024472  010304                  MOV     R3,R4
6309  024474  006004                  ROR     R4
6310  024476  006004                  ROR     R4
6311  024500  006004                  ROR     R4
6312  024502  006004                  ROR     R4
6313  024504  004737  024520          JSR     PC,SNPG         ;PRINT THIRD DIGIT
6314  024510  010304                  MOV     R3,R4
6315  024512  004737  024520          JSR     PC,SNPG         ;PRINT FOURTH DIGIT
6316  024516  000207                  RTS     PC              ;EXIT
6317  024520  012737  000260  000640  SNPG:   MOV     #260,@DB        ;SET NUMBER BASE
6318  024526  042704  177760          BIC     #177760,R4      ;MASK NUMBER
6319  024532  050437  000640          BIS     R4,@DB          ;BUILD DIGIT
6320  024536  004737  024064          JSR     PC,TOG          ;GO TYPE
6321  024542  000207                  RTS     PC              ;RETURN
6322
```

```
6324
6325                                                    ;ERROR MESSAGES**********
6326
6327  024544  051445  051127  036440  $MSWR:  .ASCIZ  /%SWR = /
      024552  000040
6328  024554  047040  053505  036440  $MNEW:  .ASCIZ  / NEW = /
      024562  000040
6329  024564  042052  020105     000  MSG1:   .ASCIZ  /*DE /
6330  024571     045  035507  000040  MSG2:   .ASCIZ  /%G; /
6331  024576  041045  020073     000  MSG3:   .ASCIZ  /%B; /
6332  024603     045  047103  000040  MSG4:   .ASCIZ  /%CN /
6333  024610  053452  020105     000  MSG5:   .ASCIZ  /*WE /
6334  024615     052  042522  000040  MSG6:   .ASCIZ  /*RE /
6335  024622  051052  020123     000  MSG7:   .ASCIZ  /*RS /
6336  024627     052  040520  051124  MSG8:   .ASCIZ  /*PATRN /
      024634  020116     000
6337  024637     045  047123  020072  MSG9:   .ASCIZ  /%SN: /
      024644     000
6338  024645     052  042523  000040  MSG10:  .ASCIZ  /*SE /
6339  024652  051452  040514  042526  MSG11:  .ASCIZ  /*SLAVE NO. /
      024660  047040  027117  000040
6340  024666  022445  042045  044522  MSG12:  .ASCIZ  /%%%DRIVE NO. /
      024674  042526  047040  027117
      024702  000040
6341  024704  025045  047102  000040  MSG13:  .ASCIZ  /%*BN /
6342  024712  051052  020116     000  MSG14:  .ASCIZ  /*RN /
6343  024717     045  020041  020040  MSG15:  .ASCIZ  /%!          BAD RECORD%%/
      024724  020040  020040  020040
      024732  041040  042101  051040
      024740  041505  051117  022504
      024746  000045
6344  024750  043040     000          MSG16:  .ASCIZ  / F/
6345  024753     040  000122          MSG17:  .ASCIZ  / R/
6346  024756  020041  047505  020124  MSG20:  .ASCIZ  /! EOT NO: /
      024764  047516  020072     000
6347
6348  024771     045  047111  042524  MSG21:  .ASCIZ  /%INTERCHANGE READ = /
      024776  041522  040510  043516
      025004  020105  042522  042101
      025012  036440  000040
6349  025016  020445  046111  042514  MSG22:  .ASCIZ  /%!ILLEGAL BOT: HALT%%%/
      025024  040507  020114  047502
      025032  035124  044040  046101
      025040  022524  022445     000
6350  025045     045  051503  020061  MSG23:  .ASCIZ  /%CS1 /
      025052     000
6351  025053     045  041527  000040  MSG23A: .ASCIZ  /%WC /
6352  025060  041045  020101     000  MSG23B: .ASCIZ  /%BA /
6353  025065     045  041506  000040  MSG23C: .ASCIZ  /%FC /
6354  025072  041445  031123  000040  MSG23D: .ASCIZ  /%CS2 /
6355  025100  042045  020123     000  MSG23E: .ASCIZ  /%DS /
6356  025105     045  051105  000040  MSG23F: .ASCIZ  /%ER /
6357  025112  040445  020123     000  MSG23G: .ASCIZ  /%AS /
6358  025117     045  045503  000040  MSG23H: .ASCIZ  /%CK /
6359  025124  042045  020102     000  MSG23I: .ASCIZ  /%DB /
6360  025131     045  051115  000040  MSG23J: .ASCIZ  /%MR /
```

```
 6361  025136  042045  020124     000  MSG23K: .ASCIZ  /%DT /
 6362  025143     045  041524  000040  MSG23L: .ASCIZ  /%TC /
 6363  025150  051445  020116     000  MSG23M: .ASCIZ  /%SN /
 6364  025155     045  047041  020117  MSG24:  .ASCIZ  /%!NO INTERRUPT%/
       025162  047111  042524  051122
       025170  050125  022524     000
 6365  025175     045  047041  020117  MSG25:  .ASCIZ  /%!NO MOL: HALT%/
       025202  047515  035114  044040
       025210  046101  022524     000
 6366  025215     045  051104  050117  MSG26:  .ASCIZ  /%DROPS: /
       025222  035123  000040
 6367  025226  050045  041511  051513  MSG27:  .ASCIZ  /%PICKS: /
       025234  020072     000
 6368  025237     045     000          MSG28:  .ASCIZ  /%/
 6369  025241     045  052045  047515  MSG30:  .ASCIZ  '%%TM03-TU45 AUTO SEQUENCE (CZTURB0)%';++8
       025246  026463  052524  032464
       025254  040440  052125  020117
       025262  042523  052521  047105
       025270  042503  024040  055103
       025276  052524  041122  024460
       025304  000045
 6370  025306  022445  046524  031460  MSG31:  .ASCIZ  '%%TM03-TU45 DATA RELIABILITY TEST (CZTURB0)%';++8
       025314  052055  032125  020065
       025322  040504  040524  051040
       025330  046105  040511  044502
       025336  044514  054524  052040
       025344  051505  020124  041450
       025352  052132  051125  030102
       025360  022451     000
 6371  025363     124  050131  020105  MSG31A: .ASCIZ  /TYPE <CR> TO TERMINATE ALL REQUESTS & ^C TO RESTART%/
       025370  041474  037122  052040
       025376  020117  042524  046522
       025404  047111  052101  020105
       025412  046101  020114  042522
       025420  052521  051505  051524
       025426  023040  057040  020103
       025434  047524  051040  051505
       025442  040524  052122  000045
 6372  025450  051445  040514  042526  MSG32:  .ASCIZ  /%SLAVE NUMBER = /
       025456  047040  046525  042502
       025464  020122  020075     000
 6373  025471     045  042504  051516  MSG33:  .ASCIZ  /%DENSITY = /
       025476  052111  020131  020075
       025504     000
 6374  025505     045  040520  044522  MSG34:  .ASCIZ  /%PARITY = /
       025512  054524  036440  000040
 6375  025520  051045  041505  051117  MSG35:  .ASCIZ  /%RECORD COUNT - /
       025526  020104  047503  047125
       025534  020124  020075     000
 6376  025541     045  044103  051101  MSG36:  .ASCIZ  /%CHAR COUNT = /
       025546  041440  052517  052116
       025554  036440  000040
 6377  025560  050045  052101  042524  MSG37:  .ASCIZ  /%PATTERN NUMBER   /
       025566  047122  047040  046525
       025574  042502  020122  020075
       025602     000
```

```
 6378  025603  045     044523  043516  MSG38:  .ASCIZ  /%SINGLE PASS = /
       025610  042514  050040  051501
       025616  020123  020075  000
 6379  025623  045     051103  020103  MSG39:  .ASCIZ  /%CRC CORRECTION (YES=1,NO=0) = /
       025630  047503  051122  041505
       025636  044524  047117  024040
       025644  042531  036523  026061
       025652  047516  030075  020051
       025660  020075  000
 6380  025663  045     042445  052116  MSG40:  .ASCIZ  /%%ENTER STALLS%READ = /
       025670  051105  051440  040524
       025676  046114  022523  042522
       025704  042101  036440  000040
 6381  025712  053445  044522  042524  MSG41:  .ASCIZ  /%WRITE = /
       025720  036440  000040
 6382
 6383  025724  052045  051125  020116  MSG42:  .ASCIZ  /%TURN AROUND = /
       025732  051101  052517  042116
       025740  036440  000040
 6384  025744  037445  000045          MSG43:  .ASCIZ  /%?%/
 6385  025750  042445  052116  051105  MSG44:  .ASCIZ  /%ENTER YOZZLE STALL = /
       025756  054440  055117  046132
       025764  020105  052123  046101
       025772  020114  020075  000
 6386  025777  045     051105  020122  MSG45:  .ASCIZ  /%ERR AMT /
       026004  046501  020124  000
 6387  026011  045     041506  000040  MSG46:  .ASCIZ  /%FC /
 6388  026016  041445  020101  000     MSG47:  .ASCIZ  /%CA /
 6389  026023  045     047041  020117  MSG48:  .ASCIZ  /%!NO BOT ON REWIND: HALT%%/
       026030  047502  020124  047117
       026036  051040  053505  047111
       026044  035104  044040  046101
       026052  022524  000045
 6390  026056  047045  052117  040440  MSG49:  .ASCIZ  /%NOT AVAIL /
       026064  040526  046111  000040
 6391  026072  044445  046114  043505  MSG50:  .ASCIZ  /%ILLEGAL DRIVE TYPE /
       026100  046101  042040  044522
       026106  042526  052040  050131
       026114  020105  000
 6392  026117  045     051104  053111  MSG52:  .ASCIZ  /%DRIVE NUMBER = /
       026124  020105  052516  041115
       026132  051105  036440  000040
 6393  026140  043045  051117  040515  MSG53:  .ASCIZ  /%FORMAT = /
       026146  020124  020075  000
 6394  026153  052     042527  052040  MSG54:  .ASCIZ  /*WE TM/
       026160  000115
 6395  026162  051452  020105  046524  MSG55:  .ASCIZ  /*SE TM/
       026170  000
 6396  026171  040     046524  000     MSG56:  .ASCIZ  / TM/
 6397  026175  045     047516  026516  MSG57:  .ASCIZ  /%NON-EXIST SLAVE/
       026202  054105  051511  020124
       026210  046123  053101  000105
 6398  026216  041445  041522  000040  MSG58:  .ASCIZ  /%CRC /
 6399  026224  046045  041522  000040  MSG59:  .ASCIZ  /%LRC /
 6400  026232  042052  000040          MSG60:  .ASCIZ  /*D /
 6401  026236  050052  000040          MSG61:  .ASCIZ  /*P /
```

```
 6402  026242  043052  000040          MSG62:  .ASCIZ  /*F /
 6403  026246  025045  051117  043511  MSG64:  .ASCIZ  /%*ORIGINAL ERROR*/
       026254  047111  046101  042440
       026262  051122  051117  000052
 6404  026270  051045  052105  054522  MSG65:  .ASCIZ  /%RETRY: /
       026276  020072     000
 6405  026301     052  051441  020105  MSG66:  .ASCIZ  /*!SE RTRY /
       026306  052122  054522  000040
 6406  026314  020452  051105  051501  MSG67:  .ASCIZ  /*!ERASE/
       026322  000105
 6407  026324  051045  051105  053105  MSG68:  .ASCIZ  /%REREV: /
       026332  020072     000
 6408  026335     045  040524  042520  MSG69:  .ASCIZ  /%TAPE MARK = /
       026342  046440  051101  020113
       026350  020075     000
 6409  026353     045  047041  020117  MSG70:  .ASCIZ  /%.NO DRY FROM REWIND: HALT%/
       026360  051104  020131  051106
       026366  046517  051040  053505
       026374  047111  035104  044040
       026402  046101  022524     000
 6410  026407     045  047516  026516  MSG71:  .ASCIZ  /%NON-EXIST DRIVE/
       026414  054105  051511  020124
       026422  051104  053111  000105
 6411  026430  051045  043105  042127  MSG72:  .ASCIZ  /%REFWD: /
       026436  020072     000
 6412  026441     045  052127  051105  MSG73:  .ASCIZ  /%WTERR: /
       026446  035122  000040
 6413  026452  051045  043505  051511  MSG74:  .ASCIZ  /%REGISTER START = /
       026460  042524  020122  052123
       026466  051101  020124  020075
       026474     000
 6414  026475     045  042526  052103  MSG75:  .ASCIZ  /%VECTOR = /
       026502  051117  036440  000040
 6415  026510  042045  051105  053105  MSG76:  .ASCIZ  /%DEREV: /
       026516  020072     000
 6416  026521     045  042504  053506  MSG77:  .ASCIZ  /%DEFWD: /
       026526  035104  000040
 6417  026532  020445  047516  026516  MSG78:  .ASCIZ  /%!NON-RETRYABLE WRITE ERROR: ER /
       026540  042522  051124  040531
       026546  046102  020105  051127
       026554  052111  020105  051105
       026562  047522  035122  042440
       026570  020122     000
 6418  026573     045  047041  047117  MSG79:  .ASCIZ  /%!NON-RETRYABLE READ ERROR: ER /
       026600  051055  052105  054522
       026606  041101  042514  051040
       026614  040505  020104  051105
       026622  047522  035122  042440
       026630  020122     000
 6419  026633     045  020441  047105  MSG100: .ASCIZ  /%!!END OF PASS %/
       026640  020104  043117  050040
       026646  051501  020123  000045
 6420  026654  022445  025052  025052  MSG101: .ASCIZ  /%%**********************/
       026662  025052  025052  025052
       026670  025052  025052  025052
       026676  025052  025052     000
```

```
 6421  026703  000052  046524  031460  MSG102: .ASCIZ  /*TM03 /
       026710  000040
 6422  026712  051452  040514  042526  MSG103: .ASCIZ  /*SLAVE /
       026720  000040
 6423  026722  040445  052125  020117  MSG104: .ASCIZ  /%AUTO CONT: /
       026730  047503  052116  020072
       026736  000000
 6424  026737  000045  042522  047503  MSG105: .ASCIZ  /%RECOVERED/
       026744  042526  042522  000104
 6425  026752  020452  041041  042101  MSG106: .ASCIZ  /*!.BAD TAPE OVERFLOW/
       026760  052040  050101  020105
       026766  053117  051105  046106
       026774  053517  000000
 6426  026777  000045  042522  044527  MSG16A: .ASCIZ  /%REWIND TAPE; RESTART AT BLOCK 1/
       027004  042116  052040  050101
       027012  035505  051040  051505
       027020  040524  052122  040440
       027026  020124  046102  041517
       027034  020113  000061
 6427  027040  020445  052441  051116  MSG107: .ASCIZ  /%!!UNRECOVERABLE BAD SPOT/
       027046  041505  053117  051105
       027054  041101  042514  041040
       027062  042101  051440  047520
       027070  000124
 6428  027072  041045  042101  051040          .ASCIZ  /%BAD RECORD LEFT ON TAPE%/
       027100  041505  051117  020104
       027106  042514  052106  047440
       027114  020116  040524  042520
       027122  000045
 6429  027124  020452  050041  051517  MSG109: .ASCIZ  /*!!POSITION LOST IN RETRY/
       027132  052111  047511  020116
       027140  047514  052123  044440
       027146  020116  042522  051124
       027154  000131
 6430  027155  051445  051525  042520  MSG110: .ASCIZ  /%SUSPECT BAD TAPE/
       027164  052103  041040  042101
       027172  052040  050101  000105
 6431  027200  051045  050105  040505  MSG111: .ASCIZ  /%REPEAT: /
       027206  035124  000040
 6432  027212  041040  042101  052040  MSG112: .ASCIZ  / BAD TAPE SPOTS%/
       027220  050101  020105  050123
       027226  052117  022523  000000
 6433
 6434  027233  000045  051440  043117  MSG113: .ASCIZ  /% SOFT: /
       027240  035124  000040
 6435
 6436  027244  020045  040510  042122  MSG114: .ASCIZ  /% HARD: /
       027252  020072  000000
 6437
 6438  027255  000045  020441  040510  MSG115: .ASCIZ  /%!!HARD READ ERROR/
       027262  042122  051040  040505
       027270  020104  051105  047522
       027276  000122
 6439  027300  020445  047125  052111  MSG116: .ASCIZ  /%!UNIT IS REWINDING: TEST WILL START AT BOT/
       027306  044440  020123  042522
       027314  044527  042116  047111
```

```
                027322  035107  052040  051505
                027330  020124  044527  046114
                027336  051440  040524  052122
                027344  040440  020124  047502
                027352  000124
 6440           027354  000134                  MSG118: .ASCIZ  /\/
 6441           027356  051045  046505  053117  MSG120: .ASCIZ  /%REMOVE TMDP FROM SLAVE TO BE TESTED%/
                027364  020105  046524  050104
                027372  043040  047522  020115
                027400  046123  053101  020105
                027406  047524  041040  020105
                027414  042524  052123  042105
                027422  000045
 6442           027424  044045  051101  053504  MSG121: .ASCIZ  /%HARDWARE SWR IN USE%/
                027432  051101  020105  053523
                027440  020122  047111  052440
                027446  042523  000045
 6443
 6444                                            .EVEN
 6445           027452  000000                  WDATA:  0                       ;WRITE BUFFER
 6446
 6447                   033460                           .=.+4004
 6448           033460  000000                  RDATA:  0                       ;READ BUFFER
 6449
 6450                   000001                           .END
```

ABLCNT	000742	2888#	3225	5917*	5943*	5952*						
ACTLRC	021102	5496*	5589	5645#								
ADRVN	000740	2887#	5841*	5847	5864	5866*	5878	5889	5920			
AMOD1	022376	5857	5911#									
AMOD2	022536	5863	5936#									
APATS	015002	4900	4909#									
AS	000526	2809#	5623	5624*								
ASEQ	022024	4474	5834#									
ASEQCF	000744	2889#	5836	5868								
ASEQF	000736	2790*	2886#	3156*	3163*	3223	3433	4432	4471	4711	6014	
ASEQX	022200	5865	5868#									
ASEQ0	022056	3164	3447	5841#	5869							
ASEQ1	022062	5842#	5867									
ASEQ2	022122	5852#	5856									
ASEQ3	022134	5853	5857#									
ASEQ4	022162	5864#	5882									
BA	000514	2804#	3586*	3639*	3776*	3792*	3978*	3987*	4206*	4214*	4281*	5411	5558
BAER	021066	5407	5413*	5504	5554	5639#							
BBC	000660	2862#	5036*	5099*	5103*	5192*	5193	5199*	5200*	5201*	5202	5204*	5227*
BCNT	000712	2875#	5292*	5315*	5318*	5323*	5344*	5347*	5351*	5356*			
BDPP	000720	2878#	5278*	5297*	5298	5326*	5331*	5340*	5342*	5345	5349*		
BD00	001410	2914	3020#										
BD10	001430	2915	3022#										
BD20	001450	2916	3024#										
BD30	001470	2917	3026#										
BD40	001510	2918	3028#										
BD50	001530	2919	3030#										
BD60	001550	2920	3032#										
BD70	001570	2921	3034#										
BKRT	012062	3777	4302#										
BKSP	011706	3901	3914	4278#									
BKTM	011770	4286	4288#										
BKTM0	012032	4292	4296#										
BLCNTR	000656	2861#	3190*	3222*	3225	3227*	3752	5911*	5936*	6022			
BPKP	000722	2879#	5277*	5305*	5306	5325*	5330*	5341*	5343*	5354	5358*		
BP00	001210	2906	3004#										
BP10	001230	2907	3006#										
BP20	001250	2908	3008#										
BP30	001270	2909	3010#										
BP40	001310	2910	3012#										
BP50	001330	2911	3014#										
BP60	001350	2912	3016#										
BP70	001370	2913	3018#										
BTADDR	001030	2925#	3746	3846									
BTFLG	000726	2881#	3367	3373	3377	3390*	3780*	3804*					
BTOV	007200	3759	3803#										
BTOVX	007340	3831	3837#										
BTOV0	007220	3807#	3856										
BTOV1	007230	3810#	3829										
BTOV2	007314	3826	3830#										
BTOV3	007332	3834#	3836										
BTPRT	007342	3324	3392	3843#									
BTPRT1	007412	3854	3856#										
BTPT	000732	2883#	3746*	3747	3749*	3750	3757	3807	3825	3833	3846*	3847	3853
BTSTF	000730	2882#	3323*	3325*	3830								
BTUR	007416	3767	3860#										

```
BT00    001610          2925    3042#
BT01    001714          2926    3044#
BT02    002020          2927    3046#
BT03    002124          2928    3048#
BT04    002230          2929    3050#
BT05    002334          2930    3052#
BT06    002440          2931    3054#
BT07    002544          2932    3056#
B0      012042          4283    4289    4295    4298#
B1      012106          4303    4306#   6034
B2      012116          4308#
CADER   021064          3646*   4014*   4293*   5405*   5562    5638#
CC      000530          2810#   5483    5578    5604
CCNTR   012142          3235    4349#
CHNFLG  003040          3157#*  3445    5885
CLF     014334          4727    4731*   4750#   4943*
CLLAST  015202          4965#
CLP     015312          4949    4976    4984#
CLPE    015336          4985    4987    4991#
CLP2    015374          4999    5003#
CLP3    015406          5001    5004    5006#
CL0     015122          4948#   4964
CL1     015150          4956#
CL2     015172          4953    4962#
CL3     015254          4975#   4979
CL4     015310          4942    4983#
CONER   021070          5457*   5534    5640#
CRCC    000566          2828#   4638    4640    5617
CRCER   021100          5487*   5574    5644#
CRCLRC  015066          4717    4729    4738    4941#
CRCSV   021106          5477*   5635    5647#
CS      000520          2806#   3206*   3245*   3527*   4475*   4483*   4485    4511*   4512*   5440    5542    5619*   5620*
                        5698*   5877*   5878*   5879
C1      000510          2802#   3363*   3366*   3478*   3479*   3504*   3537*   3540*   4466    4484    4489    5438    5447
                        5538    5622*   5625    5737*   5751*
DATA0   002772          3121#
DATA1   002774          3122#
DATA10  003012          3129#
DATA11  003014          3130#
DATA12  003016          3131#
DATA13  003020          3132#
DATA14  003022          3133#
DATA15  003024          3134#
DATA2   002776          3123#
DATA3   003000          3124#
DATA4   003002          3125#
DATA5   003004          3126#
DATA6   003006          3127#
DATA7   003010          3128#
DATBL   002770          3120#   4737
DATER1  001130          2971#   3300    5120*
DATR    005022          3238    4715    4920#
DAT0    014336          3121    4757#
DAT0A   014362          4762#   4770    4774    4777
DAT0B   014370          4763#   4764
DAT0C   014434          4768    4775#
```

```
DAT0D   014442          4778#   4786
DAT0E   014452          4780#   4785
DAT0F   014466          4782    4784#
DAT1    014476          3122    4792#
DAT1A   014502          4793#   4802    4825    4830    4879
DAT10   014620          3129    4846#
DAT11   014650          3130    4858#
DAT12   014670          3131    4868#
DAT13   014712          3132    4878#
DAT14   014722          3133    4883#
DAT15   014752          3134    4899#
DAT2    014516          3123    4801#
DAT3    014522          3124    4806#
DAT3A   014530          4808#   4819
DAT4    014546          3125    4817#
DAT5    014556          3126    4824#
DAT6    014564          3127    4829#
DAT7    014572          3128    4834#
DB      000532          2811#
DCHK    015446          4059    4253    5036#
DCHK0   015474          5040    5042#
DEREV1  001170          2993#   3320    5122*
DEREX   016530          5194    5215    5217    5226    5235    5238    5240#
DEREX1  016562          5241    5244    5246    5248#
DERFL   000706          2873#   5037*   5113    5249*
DERR    016060          5106    5154#
DERR0   016070          5156#   5247
DERR0A  016120          5158    5163#
DERR0B  016152          5169    5172#
DERR0C  016176          5176    5179#
DERR0D  016200          5178    5180#
DERR1   016226          5184    5187#
DERR2   016230          5186    5188#
DERR3   016244          5191#
DERR4   016246          5155    5190    5192#
DERR4A  016406          5209    5218#
DERR4B  016454          5203    5227#
DERR5   016512          5232    5236#
DERR6   016524          5206    5230    5239#
DFX     016056          5114    5116    5121    5123#
DF0     015750          5062    5101#   5110
DF0A    015644          5072    5074#   5111
DF0A0   015666          5078    5080#
DF0A1   015702          5083    5085#
DF0A2   015716          5088    5090#
DF0A3   015732          5093    5095#
DF0A4   015736          5075    5097#
DF0B    015604          5063#
DF0B0   015626          5066    5069#
DF0C    015564          5055    5059#
DF0C0   015574          5045    5047    5049    5061#
DF0D    015550          5051    5056#
DF0E    015542          5053#   5058
DF0F    015534          5050#   5054
DF1     015762          5098    5102    5105#
DF2     015772          5100    5104    5107#
```

```
DF3     016010          5108    5112#
DF4     016052          5119    5122#
DOUT    024342          5180    5188    6275#   6291    6293
DOUTD   024416          6289#
DPC     016704          5289#   5335
DPCG    016712          5290    5292#
DPCO    016720          5293#   5327
DPCOA   016762          5301    5303#
DPC1    016770          5296    5305#
DPC1A   017016          5309    5311#
DPC2    017060          5294    5299    5307    5321#
DPC2A   017022          5304    5312#
DPC2B   017044          5316#   5319
DPC3    017112          5324    5328#
DPPRT   017160          3271    5303    5311    5337#
DPPRTX  017314          5357    5360#
DPPRTO  017230          5345#   5350
DPPRT1  017254          5348    5351#
DPPRT2  017270          5354#   5359
DROP    016674          5284    5287#
DRPK    016662          5280    5284#
DRPKF   016574          5105    5271#
DRP1    001010          2914#   5278    5313    5331    5340
DRP2    001012          2915#
DRP3    001014          2916#
DRP4    001016          2917#
DRP5    001020          2918#
DRP6    001022          2919#
DRP7    001024          2920#
DRP8    001026          2921#
DRVER   021072          3647*   4015*   5468*   5520    5544    5595    5641#
DS      000522          2807#   3247    3330    3364    3393    3416    3418    3480    3498    3501    3532    3535    3538
                        3541    3543    3590    3644    3995    4001    4012    4291    5458    5548    5699    5715    5720
                        5893
DSUP    014124          3256    4709#
DSO     014132          4711#
DSOA    014234          4720    4731#
DSOB    014220          4724    4727#
DSOC    014164          4712    4714    4719#
DS2A    014266          4710    4738#   4758    4783
DS3     014272          4061    4254    4739#
DS4     014304          4742#   4744
DT      000536          2813#   4514    4519    4525
DVN     000550          2821#   3245    3527    4478    4483    4512    5620    5698    5920*   5977
DOFL    014474          4603*   4757    4759*   4787#
EMADDR  000654          2860#   3397*   3484*   3584*   3636*   3774*   3789*   3969*   4280*   4290*   4300*   5525    5530
                        5762
ENDFLG  000736          2885#   3178
ENDTBL  002770          3116#   3185    5859
EOTCO   002650          3061#   3385*   3386
EOTREC  000662          2863#   3369    3371*   3466*   3592    3594*   3595*   3596*   3680    3904    3925    3927*   3928*
                        3929*   3961    3965*   3966    3999*   4000*   4071    4079
ER      000524          2808#   5451    5462    5511    5552    5599
ERCHK   017316          3600    4044    4246    5387#
ERPT    020136          3648    4016    4294    5470    5472    5474    5482    5495    5499    5501#
ERPTG   020172          5506    5510#
```

```
ERPTG1  020240  5515  5520#
ERPTT   020156  5505# 5508
ERPT0   020254  5513  5519  5521  5523#  5615
ERPT1   020322  5529  5531  5534#
ERPT2   020360  5535  5544#
ERPT3   020416  5545  5554#
ERPT4   020460  5555  5564#
ERPT5   020502  5565  5570#
ERPT5A  020564  5575  5585#
ERPT6   020626  5586  5595#
ERPT7   020676  5596  5598  5601  5607#
ERPX    020732  5609  5612  5614  5616#
ERPX0   020700  5522  5608#
ERPX1   021020  5509  5629#
ERPX2   021062  5630  5632  5634  5637#
ERSAV   000724  2880# 3613  3661  3715  4105  4138  4171  5503* 5511*
ERTFL   000734  2884# 3765* 3783  3786*
ER0     017332  5389  5391#
ER0A    017400  5398  5402#
ER0B    017346  5393  5395#
ER1     017406  5396  5404#
ER10    020100  5491  5493#
ER2     017412  3651  3796  4042  4238  4244  4297  4307  5401  5403  5405#
ER2A    017456  5416# 5421  5431
ER2A0   017426  5408# 5410
ER2A1   017446  5412  5414#
ER2B    017472  5415  5420#
ER2C    017516  5426# 5435  5437
ER2D    017532  5423  5430#
ER2E    017560  5433  5436#
ER3     017566  5417  5419  5427  5429  5438#
ER3A    017624  5442  5447#
ER3A1   017662  5453  5455#
ER3B    017670  5444  5446  5450  5457#
ER4     017674  5439  5448  5456  5458#
ER4A    017740  5461  5468#
ER4A1   017732  5464  5466#
ER6     017744  5459  5467  5469#
ER6A    020030  5476  5481#
ER7     020062  5486  5488#
EXCRC   015442  4970* 4980  5017# 5477  5479* 5485  5583  5635*
EXLRC   015444  4982* 5018# 5478  5480* 5494  5593  5636*
FC      000516  2805# 3585* 3638* 3775* 3790* 3977* 4191* 4218* 4284* 4301* 5416  5426  5568
                5738*
FCER    021074  5418* 5428* 5564  5642#
FCSAV   000634  2847# 3346  4598*
FMCNT   000556  2824# 3346* 3585  3977  3981  4031  4209  4218  4228  4353* 4589* 4590  4592
                4597* 4598  4739  4920  4944  4971  5038  5059  5063  5071  5214  5218  5387
                5424  5726  5918* 5944* 6047
FRPRT   021110  4173  5162  5527  5658# 5711  5764
GTSWR   023306  4645  5794  6073#
HDRFL   000666  2865# 5112* 5157  6050*
HERE    005032  3444#
HRDS    022212  5842  5875#
INIT    005426  3257  3526#
INTRF   000570  2829# 4620  4622  5739  >>23*
```

LRCER	021076	5497*	5500*	5585	5643#
LRCSV	021104	5478*	5636	5646#	
MR	000534	2812#	5488	5743*	5821*
MSG1	024564	5160	6329#		
MSG10	024645	4280	4300	6338#	
MSG100	026633	3442	6419#		
MSG101	026654	5843	6420#		
MSG102	026703	5845	6421#		
MSG103	026712	5849	6422#		
MSG104	026722	5834	6423#		
MSG105	026737	3708	4125	6424#	
MSG106	026752	3379	6425#		
MSG107	027040	3861	6427#		
MSG109	027124	3375	6429#		
MSG11	024652	5979	6339#		
MSG110	027156	3727	6430#		
MSG111	027200	3733	6431#		
MSG112	027212	3851	6432#		
MSG113	027233	3287	3307	6434#	
MSG114	027244	3292	3312	6436#	
MSG115	027255	4161	6438#		
MSG116	027300	5718	6439#		
MSG118	027354	6135	6440#		
MSG12	024666	5975	6340#		
MSG120	027356	3160	6441#		
MSG121	027424	5788	6442#		
MSG13	024704	3814	6020	6341#	
MSG14	024712	3818	6024	6342#	
MSG15	024717	5197	6343#		
MSG16	024750	5663	6344#		
MSG16A	026777	3388	6426#		
MSG17	024753	5660	6345#		
MSG2	024571	5173	6330#		
MSG20	024756	3382	6346#		
MSG21	024771	4618	6348#		
MSG22	025016	4004	6349#		
MSG23	025045	5536	6350#		
MSG23A	025053	5570	6351#		
MSG23B	025060	5556	6352#		
MSG23C	025065	5566	6353#		
MSG23D	025072	5540	6354#		
MSG23E	025100	5546	6355#		
MSG23F	025105	5550	6356#		
MSG23G	025112	6357#			
MSG23H	025117	5602	6358#		
MSG23I	025124	6359#			
MSG23J	025131	6360#			
MSG23K	025136	6361#			
MSG23L	025143	6362#			
MSG23M	025150	6363#			
MSG24	025155	5765	6364#		
MSG25	025175	5712	6365#		
MSG26	025215	5337	6366#		
MSG27	025226	5352	6367#		
MSG28	025237	3827	3843	6126	6368#
MSG3	024576	5181	6331#		

```
MSG30   025241          4434    6369#
MSG31   025306          4431    6370#
MSG31A  025363          4438    4440*   6371#
MSG32   025450          4493    6372#
MSG33   025471          4532    6373#
MSG34   025505          4544    6374#
MSG35   025520          4577    6375#
MSG36   025541          4587    6376#
MSG37   025560          4599    6377#
MSG38   025603          4627    6378#
MSG39   025623          4636    6379#
MSG4    024603          5163    6332#
MSG40   025663          4650    6380#
MSG41   025712          4659    6381#
MSG42   025724          4668    6383#
MSG43   025744          6159    6384#
MSG44   025750          5802    6385#
MSG45   025777          6386#
MSG46   026011          6387#
MSG47   026016          6388#
MSG48   026023          3422    3508    6389#
MSG49   026056          3252    3335    6390#
MSG5    024610          3584    5706    6333#
MSG50   026072          4523    6391#
MSG52   026117          4476    6392#
MSG53   026140          4556    6393#
MSG54   026153          3056    3530    6394#
MSG55   026162          4290    6395#
MSG56   026171          5532    6396#
MSG57   026175          4516    6397#
MSG58   026216          5576    6398#
MSG59   026224          5587    6399#
MSG6    024615          3397    3484    3969    5709    6334#
MSG60   026232          3399    5984    6400#
MSG61   026236          5990    6401#
MSG62   026242          5997    6402#
MSG64   026246          3625    3675    4115    6403#
MSG65   026270          3272    3710    3729    4127    4150    6404#
MSG66   026301          3774    6405#
MSG67   026314          3789    6406#
MSG68   026324          3302    6407#
MSG69   026335          4609    6408#
MSG7    024622          6045    6335#
MSG70   026353          3486    6409#
MSG71   026407          4487    6410#
MSG72   026430          3282    6411#
MSG73   026441          3277    6412#
MSG74   026452          4443    6413#
MSG75   026475          4452    6414#
MSG76   026510          3317    6415#
MSG77   026521          3297    6416#
MSG78   026532          3617    3665    3720    6417#
MSG79   026573          4109    4143    6418#
MSG8    024627          6007    6336#
MSG9    024637          4528    6337#
MTC1    000674          2868#   3587*   3637*   3791*   3988*   3990*   4194*   4196*   4215*   4217*   4285*   4302*   5395
```

```
                                5414    5420    5473    5629    5658    5661    5704    5722    5741    5744    5749*   5751    6027
MTINT   022006                  2784    4462    5820#
MTINTA  022010                  5770    5821#
NOP   = 000240                  2739#
NRTP    011252                  3619    3667    3722    4111    4145    4171#
NRZOF   000652                  2859#
OCTP    024126                  2756    6219#
OCTPG   024276                  6227    6230    6237    6241    6246    6251    6253    6257#
OCTPG0  024314                  6258    6260    6263#
OCTPG1  024320                  6222    6264#
OCTP0   024144                  6221    6224#
OCTP1   024164                  6225    6229#
OCTP2   024172                  6228    6231#
OCTP3   024262                  6223    6254#
OFL     024340                  6219*   6259    6263*   6269#
ONE     014232                  4728    4730#
PAPRT   022672                  3251    3268    3334    3372    3398    3421    3485    3507    3616    3664    3719    3860    4003
                                4108    4142    5159    5302    5310    5524    5703    5717    5761    5975#
PAPRTA  023046                  6011#   6017
PAPRTB  023062                  6010    6014#
PAPRTC  023076                  6015    6018#
PAPRTD  023104                  6013    6020#
PAPRTY  023200                  6035    6038#
PAPRT0  022774                  5994    5996#
PAPRT1  023150                  6028    6031#
PAPRT2  023202                  6033    6039#
PAPRT3  023204                  6030    6037    6040#
PARS    014332                  4723    4725*   4745*   4746*   4749#
PATRN   000560                  2825#   4601    4604    4713    4719    4733    5921*   5925*   5927*   5929*   5946*   5948*   5950*
                                5953*   6016    6018
PATS    014330                  3189*   3438*   4354*   4719    4734*   4748#   5057*
PFLG    000672                  2867#   5156*   5243    5248*   5523*   5611    5616*
PICK    017114                  5285    5329#
PIK1    000770                  2906#   5277    5314    5330    5341
PIK2    000772                  2907#
PIK3    000774                  2908#
PIK4    000776                  2909#
PIK5    001000                  2910#
PIK6    001002                  2911#
PIK7    001004                  2912#
PIK8    001006                  2913#
PRB     000624                  2843#   4766
PRS     000622                  2842#   4760*   4762*   4763
PSW     000606                  2836#   5752*   5758*   5780*   6168    6169*   6173*
RANBAS  000626                  2844#   3343*   6063*   6064
RANG    023254                  4351    4369    4924    6063#   6066    6068
RANSAV  000630                  2845#   3344*   4352*   4353    4370    4925    6063    6064*   6065    6067
RANSET  004276                  3183    3343#   3437
RCNT    000554                  2823#   3345*   3583    3754    3927    3960    4298    4370*   4579    4581    4586    5919*   5945*
                                6031    6043
RCNTR   012202                  3241    4367#
RCSAV   000632                  2846#   3345    4586*
RDA     010004                  3962    3964    3969#
RDATA   033460                  3776    3978    4014    4024    4206    4227    4240    4281    4293    4296    4306    4741    4781
                                5043    5166    5211    5225    5399    5402    6448#
RDCMD   000562                  2826#   3882*   3890*   3894    3902*   3909*   3915*   3919    3923*   3963    3971    3979    3993
```

		4018	4025	4048	4067	4077	4132	4164	4192	4207	4225	5061	5109	5118
		5168	5175	5183	5189	5208	5231	5279	5397	5432	5498	6029		
RDERR1	001150	2982#	3305	4020*	4052*									
RDER1	001110	2960#	3285	4022*	4050*									
RDER2	001112	2961#												
RDER3	001114	2962#												
RDER4	001116	2963#												
RDER5	001120	2964#												
RDER6	001122	2965#												
RDER7	001124	2966#												
RDER8	001126	2967#												
RDEX	010672	4078	4080	4082	4084	4088#								
RDFL	015064	2778*	3168*	3439*	4709	4716*	4928*	4930#						
RDRTG	010766	4118#	4160											
RDRTX	011250	4167	4169#											
RDRTY	010700	4053	4102#											
RDRT0	010712	4103	4105#											
RDRT1	010744	4107	4113#											
RDRT1A	010742	4112#												
RDRT1B	010762	4114	4117#											
RDRT2	011044	4124	4131#											
RDRT3	011066	4133	4136#											
RDRT4	011072	4135	4137#											
RDRT5	011074	4122	4138#											
RDRT5A	011140	4141	4148#											
RDRT5B	011150	4147	4150#											
RDRT6	011200	4149	4155	4158#										
RDRT7	011244	4165	4168#											
RDX	010676	4089#												
RD0	010042	3972	3974	3977#	4076	4087								
RD1	010110	3984	3987#											
RD1A	010124	3980	3990#											
RD1B	010132	3989	3991#											
RD1D	010140	3992#												
RD10	010616	4070	4072	4074#										
RD11	010626	4075	4077#											
RD2	010144	3991	3993#											
RD3	010204	3994	3996	3998	4001#									
RD4	010234	4002	4008#											
RD4A	010332	4013	4024#											
RD4A0	010364	4026	4031#											
RD4A1	010406	4034	4037#											
RD4A2	010430	4028	4030	4040	4042#									
RD4B	010436	4011	4044#											
RD4C	010442	4043	4045#											
RD4D	010472	4049	4052#											
RD4E	010476	4051	4053#											
RD5	010506	4009	4046	4055#										
RD6	010530	4021	4023	4056	4058	4060#								
RD7	010554	4063	4065#											
RD7A	010604	4068	4071#											
READ	007744	3891	3903	3916	3924	3960#								
REGS	000544	2819#	4445	4447	4464									
REOT	004330	3362#	3686	3782	3806	3906	3930							
REOTC	005054	3213	3215*	3402*	3408*	3409	3410*	3411*	3450#	3467*	3468	3469*	3470*	4425*
		4568*	4574	4575*	4576*	5875*	5902*	5904*						

```
REOTX   004760          3434    3437#
REOTXX  005050          3441    3449#
REOT1A  004376          3368    3371#
REOT1B  004424          3374    3377#
REOT1C  004444          3378    3382#
REOT1E  004470          3381    3388#
REOT1F  004440          3376    3380#
REOT2   004512          3393#   3396
REOT2A  004550          3394    3402#
REOT3   004576          3403    3408#
REOT4   004626          3414#   3430
REOT5   004642          3416#   3417
REOT6   004706          3419    3426#
REOT7   004736          3431#
RETRY   000604          2835#   4159    5518
RFHARD  002730          3096#   3295    4166*
RFSOFT  002670          3072#   3290    4134*
RPCNT   000702          2871#   3628*   3669*   3703*   3704    3735    3787*
RRHARD  002750          3107#   3315    4168*
RRSOFT  002710          3083#   3310    4136*
RSEQ    007432          3262    3882#
RSEX    007566          3893    3904#   3918
RSF     007504          3889    3892#
RSFR    007602          3887    3909#
RSFRX   007742          3926    3931#
RSFR0   007634          3913    3915#
RSFR1   007646          3911    3917#
RSFR2   007700          3920    3923#
RSF0    007540          3895    3899#
RSF1    007554          3898    3900    3902#
RSR     007462          3885    3888#
RSTAL   000574          2831#   4065    4652    4654
RTCNT   000704          2872#   3627*   3672*   3712    3731    3741    3760*   3761    4117*   4129    4152    4158*   4159
                        5516
RTRN    000664          2864#   3588*   3640*   3793*   3991*   4197*   4219*   4286*   4303*   5822    6034
RTYFL   000714          2876#   3603    3630*   3652    3678*   3694*   3803*   4054*   4119*   4247    5514    6036
RTY1    001050          2938#   3275    3622*   3671*   3764*
RTY2    001052          2939#
RTY3    001054          2940#
RTY4    001056          2941#
RTY5    001060          2942#
RTY6    001062          2943#
RTY7    001064          2944#
RTY8    001066          2945#
RWND    005056          3258    3461#
RWNDA   005070          3462    3464#   5916    5942
RWNDX   005376          3495    3512#
RWND0   005126          3471#   3491
RWND1A  005236          3475    3481    3489#
RWND2   005254          3473    3492#
RWND3   005260          3493#   3505
RWND5   005336          3503#   3511
RWND6   005354          3502    3506#
SCVFL   014076          2796*   4441    4646    4677*   4679#   5798*
SERFL   000710          2874#   3606    3610*   3655    3701    3771*   3778    3788*   3797    4045    4060*   4118*   4121
                        5115    5502*   5510*
```

```
SN      000540  2814#  4530
SNPG    024520  6304   6307   6313   6315   6317#
SNPT    024442  4531   6298#
SPFLG   000572  2830#  3440   4629   4631
STAL    000670  2866#  3249*  3260*  3263*  3332*  3395*  3482*  3601*  3772*  3896*  3921*  4065*  4189*
                4198*  4202*  4204*  4278*  4304*  4308*  4332*  5748*  5756*
STALL   012132  3261   3264   3602   3773   3897   3922   4066   4190   4205   4279   4309   4332#  4333
START   003026  2772   2797   3155#
STARTA  003160  2779   3176#  4007
STARTB  003164  3169   3177#
STARTC  003152  2775   3172#
STARTD  003244  3173   3191#
STARTE  003236  3190#  3449
START1  003406  3216#  3339
START2  003524  3234   3236#
START3  003540  3237   3239#
START4  003554  3221   3240   3242#  3250   3255
START5  004216  3266   3326#
START6  004234  3330#  3333
START7  004264  3244   3331   3338#  3407
START8  004272  3329   3339#
STAR1A  003420  3218#
STAR1B  003440  3219   3222#
STAR1C  003474  3224   3226   3230#
STAR4A  003634  3248   3256#
STAR40  003566  3243   3245#
STAUT   003136  2791   3159   3162   3167#
STAUTO  003346  3207#  5924   5926   5928   5930   5947   5949   5951   5954
STFLG   000640  2853#  3177   3178
STP     003734  3269   3271#  3391
STPX    004202  3270   3323#
STTBL   001050  2937#  3184   3185   5858   5859
SWR     000610  2837#  3157*  3193   3197   3201*  3233   3236   3239   3265   3326   3461   3580   3598
                3611   3623   3642   3659   3673   3682   3706   3725   3883   3886   3888   3892   3899
                3910   3912   3917   4008   4055   4062   4102   4113   4123   4148   4221   4249   4255
                4288   5154   5195   5240   5245   5300   5308   5471   5512   5608   5613   5759   5767
                5784   5786*  5793*  5800   6009   6073   6078   6082
SWREG   000176  2768#  3157   3193   3201   5784   5793   6073
TAPG    021144  3589   3641   3794   3992   4199   4220   4287   4305   5697#
TAPG0   021156  5699#  5702
TAPG1   021220  5705   5709#
TAPG2   021232  5708   5712#
TAPG3   021242  5700   5715#
TAPG3A  021312  5723   5726#
TAPG3B  021326  5725   5728   5730#
TAPG3C  021346  5733   5736#
TAPG3D  021412  5740   5742   5744#
TAPG3E  021436  5747   5749#
TAPG3F  021274  5716   5722#
TAPG4   021464  5754#  5755   5757
TAPG5   021500  5758#
TAPG6   021542  5760   5767#
TAPG7   021552  5768   5770#
TBELL   024102  6188   6208#
TC      000542  2815#  3246*  3362*  3415*  3477*  3497*  3500*  3515*  3528*  4513*  5628*  5892*
TEMP1   000644  2856#  4501   4539   4551   4563   4761*  4767   4771*  5218*  5219*  5220*  5221*  5222
```

		5224	5271*	5274*	5282*	5287	5332	5697*	5701*	5753*	5754*	5876*	5895*	5900
		5902	5903*	5904	6112*	6121	6143*							
TEMP2	000646	2857#	4495*	4496	4534*	4546*	4558*	4683*	4687	4772*	4779	5272*	5275*	5283*
		5288	5333											
TEMP3	000650	2858#	3716*	3723*	3737	3739*	4139*	4146*	4154	4156*	5273*	5295	5334*	6289
		6292												
TEND	005004	3442#	5870											
TIB	000642	2855#	6115	6119	6124	6129	6139	6141	6147*	6148	6174*	6175*	6176	
TINER	023656	6140	6142	6152	6154	6159#								
TINF	000636	2852#	3167*	3172*	3176*	3432*	4421	4648						
TINP	012226	3205	4421#											
TINPX	014070	4647	4649	4677#										
TINP0	012614	4486	4493#	4504	4518	4572								
TINP0B	012704	4502	4508#											
TINP0C	012744	4514#												
TINP0D	012764	4515	4519#											
TINP0E	013022	4522	4528#											
TINP1	013040	4532#												
TINP2	013116	4540	4544#											
TINP2A	013174	4552	4556#											
TINP2B	013252	4564	4568#											
TINP2C	013300	4507	4570	4573#										
TINP3	013320	4577#												
TINP3A	013650	4636#	5799											
TINP4	013714	4423	4646#											
TKB	000614	2839#	5775	6174										
TKS	000612	2838#	3216*	6170*	6171									
TMEX	000564	2827#	3633	3973	4081	4282	4611	4613	5922*					
TMFLG	000700	2870#	3635*	3679*	3696	3970*	3975*	3997	4010	4057	4069*	4083	4085*	4088*
		4200	4223	4251	5422	5443	5449	5460	5475	5528	5631	6032		
TOB	000640	2854#	5560*	5581*	5591*	6011*	6041*	6176*	6182*	6183	6185	6187	6191*	6194*
		6198*	6205	6209*	6254*	6265*	6276*	6279	6284*	6317*	6319*			
TOG	024064	5561	5582	5592	6012	6042	6177	6189	6192	6195	6199	6203#	6204	6210
		6255	6266	6320										
TPB	000620	2841#	6205*	6281*	6283*									
TPOS	014100	4543	4555	4567	4683#	4685								
TPOS1	014110	4510	4686#											
TPS	000616	2840#	6203	6277										
TSTAL	000600	2833#	3260	3263	3772	3896	3921	4278	4308	4670	4672			
TTIN	023674	6114	6168#											
TTINT	021556	2762	5775#											
TTOUT	023754	2751	6182#	6190	6200	6213								
TTR	023442	4451	4460	4482	4500	4538	4550	4562	4585	4596	4608	4617	4626	4635
		4644	4658	4667	4676	5810	5840	6086	6110#					
TWO	014226	4726	4729#											
TYPE =	000004	2754#	3161	3253	3273	3278	3283	3288	3293	3298	3303	3308	3313	3318
		3336	3380	3383	3389	3400	3423	3443	3487	3509	3618	3626	3666	3676
		3709	3711	3721	3728	3730	3734	3815	3819	3828	3844	3852	3862	4005
		4110	4116	4126	4128	4144	4151	4162	4436	4439	4444	4453	4477	4488
		4494	4517	4524	4529	4533	4545	4557	4578	4588	4600	4610	4619	4628
		4637	4651	4660	4669	5161	5164	5174	5182	5198	5338	5353	5526	5533
		5537	5541	5547	5551	5557	5567	5571	5577	5588	5603	5664	5707	5710
		5713	5719	5763	5766	5789	5803	5835	5844	5846	5850	5976	5980	5985
		5991	5998	6008	6021	6025	6046	6077	6081	6127	6136	6160		
TYPOCT-	104400	2758#	3276	3281	3286	3291	3296	3301	3306	3311	3316	3321	3387	3713
		3732	3736	3813	3817	3822	3850	4130	4153	4172	4446	4455	4527	4580

		4591	4602	4612	4621	4630	4639	4653	4662	4671	5172	5346	5355	5539
		5543	5549	5553	5559	5563	5569	5573	5580	5584	5590	5594	5606	5805
		5848	5855	5978	5983	5989	5996	6006	6019	6023	6040	6044	6049	6079
UDES	000552	2822#	3220*	3232*	3246	3362	3414*	3415	3476*	3477	3496*	3497	3500	3514*
		3515	3526	3983	4027	4033	4039	4211	4230	4235	4241	4541*	4553*	4565*
		4721	4745	4941	4986	5000	5003	5039	5044	5046	5048	5065	5074	5392
		5430	5452	5463	5469	5481	5597	5628	5633	5732	5981	5986	5993	5999
UNP	000676	2869#	3217	3228*	3230*	3274	3279	3284	3289	3294	3299	3304	3309	3314
		3319	3338*	3384	3404	3412*	3413	3425	3427*	3428	3431*	3464	3465*	3471
		3490*	3492*	3493	3503*	3512*	3513	3608	3621	3657	3670	3743	3745	3763
		3845	4017	4047	4131	4163	4424*	4503	4505	4508	4569	4571*	4573*	4686
		5117	5276	5312	5329	5339								
UNX	000766	2902#												
UN1	000746	2894#	3208	3210*	3218	3220	3232	3242	3405*	3414	3426*	3429	3472	3474
		3476	3489*	3494	3496	3514	4427	4506*	4509*	4513	4687*	5851	5884	5912
		5937												
UN2	000750	2895#												
UN3	000752	2896#												
UN4	000754	2897#												
UN5	000756	2898#												
UN6	000760	2899#												
UN7	000762	2900#												
UN8	000764	2901#												
UPS	000716	2877#	3464*	3512										
VECT	000546	2820#	4454	4456	4461									
WC	000512	2803#	5572	5736*										
WDATA	027452	3586	3639	3646	3650	3792	3795	4732	4778	4921	4946	4973	5042	5060
		5210	5223	5404	6445#									
WEX	006340	3582	3634	3643	3656	3660	3668	3678#						
WRITE	005542	3259	3580#											
WRTE	005556	3581	3583#											
WRTSB	007004	3695	3771#											
WRTSB0	007056	3780#	3799											
WRTSB1	007072	3779	3783#	3798										
WRTSB2	007102	3784	3786#											
WRTSB3	007156	3793	3795#											
WRTY	006402	3629	3677	3694#										
WRTYR	006434	3699	3701#											
WRTYTM	006430	3697	3700#											
WRTY0	006410	3695#	3705	3766										
WRTY1	006510	3707	3714#											
WRTY2	006512	3702	3715#											
WRTY2A	006556	3718	3725#											
WRTY2B	006574	3724	3729#											
WRTY3	006640	3726	3738	3741#										
WRTY3A	006656	3742	3745#											
WRTY4	006744	3758	3760#											
WRTY5	007000	3762	3767#											
WRW	006356	3682#												
WRWX	006400	3681	3685	3687#										
WSTAL	000576	2832#	3601	4661	4663									
WTER1	001070	2949#	3280	3609*	3658*	3744*								
WTER2	001072	2950#												
WTER3	001074	2951#												
WTER4	001076	2952#												
WTER5	001100	2953#												

```
WTER6   001102          2954#
WTER7   001104          2955#
WTER8   001106          2956#
WTM     006072          3636#   3700
WTM0    006134          3640    3642#
WTM1    006176          3645    3650#
WTM2    006206          3649    3652#
WTM3    006216          3653    3655#
WTM4    006276          3663    3669#
WTM4A   006334          3674    3677#
W0      005562          3584#   3632    3698
W1      005624          3588    3590#
W2      005664          3591    3593    3598#
W3      005700          3599    3601#
W3A     005722          3604    3606#
W4      006006          3615    3621#
W4A     006034          3624    3627#
W5      006050          3607    3612    3620    3630#
W6      006060          3633#
XOR     015414          4950    4977    4981    5009#
XORS    015440          4947*   4962*   4965    4966*   4967*   4969*   4970    4974*   4982    5010    5011*   5012*   5013
                        5016#
YOZ     011266          4064    4120    4189#   5813
YOZA    011326          4193    4196#
YOZB    011334          4195    4197#
YOZC    011354          4197    4200#
YOZC0   011442          4212    4214#
YOZC1   011456          4208    4217#
YOZC2   011464          4216    4218#
YOZD    011504          4219    4221#
YOZD0   011604          4226    4240#
YOZD1   011630          4224    4246#
YOZD2   011576          4236    4238#
YOZD3   011622          4242    4244#
YOZD4   011556          4231    4233#
YOZE    011634          4222    4247#
YOZF    011664          4239    4245    4250    4252    4254#
YOZG    011670          4248    4255#
YOZH    011704          4256    4258#
YOZO    011300          4191#   4257
YSTAL   000602          2834#   4189    4204    5804    5806
$ENDAD  005022          2759    3157    3444#
$MNEW   024554          6080    6328#
$MSWR   024544          6076    6327#
$SVPC   =  000040       2759#
.       =  033462       2744#   2748    2750#   2755#   2759#   2761#   2767#   2771#   2774#   2777#   2783#   2789#   2795#
                        2799#   3005#   3007#   3009#   3011#   3013#   3015#   3017#   3019#   3021#   3023#   3025#   3027#
                        3029#   3031#   3033#   3035#   3043#   3045#   3047#   3049#   3051#   3053#   3055#   3057#   3120
                        5862    6447#
.RESTO  023420          5790    5811    5815    6087    6091#
.SAVE   023376          5787    5797    6075    6090#
```

```
DTBOOT   1243#
GETANS    767#
LDPDR     515#
LPDP11   1268#
PSPTAG    746#
REGBOX    132#
RESLDR    873#
SAVLDR    855#
SVTK$    1141#
$CATCH   1124#
$CHAIN     89#   2641#   3157
$CHNMO    105#
$CNV16    606#
$CNV18    635#
$CNV48    704#
$CPCHK    897#
$CPREG     17#
$CPVEC    167#
$FPREG     46#
$GETAN    771#
$KMMRE    347#
$KWPDR    998#
$KW11     929#
$LCTRL      2#
$LPREG    186#
$MAMFO   1176#
$MMBIT    207#
$MMREG    264#
$PDRBI    385#
$POWER    439#
$PSWBI    147#
$RECO     795#
$RESLD    876#
$RESTO    476#   2641#   6091
$SAVE     464#   2641#   6090
$SAVID    858#
$SETIB    508#
$SHIFT    489#
$SMMRE    309#
$STINS      8#
$STKPT    202#
$ST200   1136#
$SVTK    1146#
$SWOPT     56#
$TCDRV   1029#
$TCREG    192#
$TRAPS    402#
$TYPE     518#
$TYPEF    592#
$UMMRE    272#
$VECTA   1163#
.$ACT1     67#   2641#   2759
.$EOP      78#   2641#   3444

. ABS.  033462     000
```

ERRORS DETECTED: 0

CZTURB.CZTURB/CRF=CZTURB.P11
RUN-TIME: 23 40 4 SECONDS
RUN-TIME RATIO: 173/68=2.5
CORE USED: 15K (29 PAGES)