

# TM03/TE16

DATA RELIABILITY TEST  
MD-11-DZTED-A

EP-DZTED-A-DL-A  
COPYRIGHT © 1977  
FICHE 1 OF 1

JUN 1977  
**digital**  
MADE IN USA

The image displays a grid of 100 small data tables, arranged in 10 rows and 10 columns. Each table contains various data points, likely test results, organized in a structured format with headers and rows of values. The data is presented in a consistent layout across all tables, suggesting a systematic data collection process. The tables are densely packed and cover most of the left and center portions of the page.

A small data table located in the bottom right corner of the page, containing a few rows of data. It appears to be a continuation or a summary of the data presented in the main grid.

801

EOF1DZTECASE0

00010000

770526

POP10 411

HR1DZTEDASE0

00010000

770526

.REY %

## IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZTED-A-D  
PRODUCT TITLE: TM03/TE16 DATA RELIABILITY PROGRAM  
DATE CREATED: 21 FEB 1977  
MAINTAINER: DIAGNOSTIC GROUP  
AUTHOR: J. G. ADAMS

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (c) 1977, BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

PARAGRAPH	SUBJECT	PAGE
1.	ABSTRACT	3
2.	REQUIREMENTS	3
3.	LOADING PROCEDURE	3
4.	STARTING PROCEDURE	4
5.	DATA PATTERNS	11
6.	RANDOMIZATION	12
7.	DYNAMIC PARAMETERS	13
8.	CONSOLE SWITCH	14
9.	ERROR PRINTOUTS	19
10.	STATISTICS PRINTOUT	27
11.	AUTO SEQUENCE	28
12.	TESTING PROCEDURES	30
13.	LISTING	32

1. ABSTRACT  
-----

THIS PROGRAM IS DESIGNED TO BE USED BY AN EXPERIENCED ENGINEER /TECHNICIAN FOR EVALUATION AND DEBUGGING OF MAG TAPE DRIVES. THE PROGRAM IS CAPABLE OF EXERCISING THE TE16 MAGNETIC ON A MASSBUS THROUGH THE TMO3 MAG TAPE CONTROLLER. ANY COMBINATION OF TMO3'S & TE16'S UP TO A MAXIMUM OF EIGHT (8), MAY BE TESTED BY A SINGLE EXECUTION OF THE PROGRAM. THIS FLEXIBILITY IS POSSIBLE BECAUSE THE PROGRAM HAS NO FIXED PARAMETERS OR TESTING SEQUENCE. THE ENTIRE TEST PLAN, INCLUDING PARAMETERS AND OPERATING SEQUENCE, IS DETERMINED BY THE OPERATOR THROUGH RESPONSES TO TELETYPE REQUESTS AND SETTING OF CONSOLE SWITCHES.

THE PROGRAM PROVIDES FOR TESTING OF ALL TAPE DRIVE FUNCTIONS SUCH AS WRITING, READING, REWINDING, TAPE POSITIONING, EOT - BOT SENSING AND ASSUMES A GOOD RH AND TMO3.

HOWEVER; THE RH AND TMO3 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 TMO3.

2. REQUIREMENTS (HARDWARE)  
-----

- A. ANY PDP-11 PROCESSOR
- B. 8K OF CORE
- C. TELETYPE
- D. TMO3 TAPE CONTROLLER
- E. 1 TO 8 MAG TAPE DRIVES
- F. MASSBUS CONTROLLER

3. LOADING PROCEDURE  
-----

USE STANDARD PROCEDURE FOR LOADING BINARY TAPES

4. STARTING PROCEDURE  
-----

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

- A. 200(8): THIS ADDRESS MUST BE USED ON INITIAL START FROM LOAD AS ALL PARAMETERS ARE ENTERED FROM HERE. REQUESTS ARE PRINTED ON THE TELETYPE FOR ENTRY OF RH STARTING ADDRESS, VECTOR ADDRESS, DRIVE NUMBER (TM03 ADDRESS), SLAVE NUMBER, DENSITY, PARITY, FORMAT, RECORD COUNT, CHARACTER COUNT, PATTERN NUMBER, TAPE MARK AND STALL FOR READ, WRITE, AND TURNAROUND. ALL RESPONSES 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 REENTERED. SOME RESPONSES REQUIRE MORE THAN ONE (1) CHARACTER, BUT NONE REQUIRES MORE THAN SIX (6). RESPONSES OF MORE THAN ONE CHARACTER NEED NOT HAVE LEADING ZEROS AND SHOULD BE TERMINATED BY A CARRIAGE RETURN IF LESS THAN THE MAXIMUM NUMBER OF CHARACTERS IS INPUT.
- B. 204(8): THIS ADDRESS SHOULD BE USED ANYTIME A RESTART OF THE PROGRAM IS NECESSARY AND THE PARAMETERS ENTERED AT THE INITIAL START OF 200(8) NEED NOT BE CHANGED. ALSO NOTE THAT ANY DATA PATTERN WHICH HAD BEEN GENERATED BY SETTING THE RANDOM DATA SWITCH (CONSOLE SWITCH EIGHT) WILL NOT BE OVERWRITTEN AND THEREFORE IS HELD IN CORE FOR USE UNTIL CONSOLE SWITCH EIGHT(8) IS AGAIN SET AND THAT ALL STATISTICS WILL BE RETAINED.
- C. 210(8): THIS ADDRESS IS THE SAME AS USING 204(8) IN THAT THE PREVIOUSLY SET PARAMETERS ARE USED; HOWEVER, THE DATA PATTERN IS RETURNED TO THE FIXED PATTERN ORIGINALLY CALLED FOR AT THE 200(8) START AND ALL STATISTICS ARE CLEARED TO ZERO.
- D. 240(8): THIS IS A SPECIAL ADDRESS WHICH WILL CAUSE THE PROGRAM TO EXECUTE A PREDETERMINED TEST PLAN ON ALL AVAILABLE DRIVES AND SLAVES. THE ONLY INPUT REQUIRED BY THE OPERATOR IS A RESPONSE TO REQUESTS FOR THE RH ADDRESS, VECTOR ADDRESS, CONTINUOUS OPERATION OF THE SEQUENCE, AND NRZ ONLY.
- E. 300(8): THIS ADDRESS IS TO BE USED AS A RESTART ONLY AND WILL PERFORM JUST AS IN 200(8) EXCEPT THAT THE PARAMETER INPUT LIST IS SHORTENED. THE SHORT PARAMETER LIST CONSISTS OF DRIVE NUMBER, SLAVE NUMBER, DENSITY, PARITY, FORMAT, RECORD COUNT, CHARACTER COUNT, PATTERN, TAPE MARK, AND

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

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

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

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

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

SLAVE NUMBER: THE SLAVE NUMBER IS ENTERED AS ONE  
(1) OCTAL CHARACTER AND MUST BE  
WITHIN THE LIMITS OF 0 THROUGH 7.  
WHEN THE SLAVE NUMBER HAS BEEN  
ENTERED AND IS LEGAL, THE PROGRAM TESTS  
FOR THE PRESENCE OF A SLAVE OF THAT  
NUMBER. IF THE SLAVE IS AVAILABLE,  
A PRINTOUT OF ? 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 WROU, A  
MESSAGE IS PRINTED FOR INFORMATION ONLY.  
IF THE SLAVE IS NOT AVAILABLE,  
A MESSAGE STATING SO WILL BE  
PRINTED AND A NEW SLAVE NUMBER  
REQUEST WILL BE ISSUED. WHEN A  
GOOD SLAVE NUMBER HAS BEEN ENTERED,  
REQUESTS FOR OPERATING DENSITY,  
PARITY AND FORMAT ARE MADE FOR THAT  
SLAVE AND SHOULD BE RESPONDED TO  
ACCORDING TO THAT PARTICULAR SLAVE'S  
NEEDS. AS MANY AS EIGHT (8) SLAVE  
NUMBER REQUESTS MAY BE USED, 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

H01

TMD3/TE16 DATA RELIABILITY PROGRAM  
DZTEDA.P11 07-APR-77 13:36

MACY11 27(1006) 07-APR-77 13:36 PAGE 6

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 DENSITY 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. RESETTNG 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 THROUGH THE READER IS DONE BY PREPARING A PAPER TAPE WITH A PROGRAM CALLED DTC. (MAINDEC-11-DZTUF-A-0) ANY CONFIGURATION OF BITS AND CHARACTERS MAY BE USED AND A LIMIT OF 377(8) CHARACTERS 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 PARAMETERS EXCEPT FOR THE SLAVE DESCRIPTION VALUES (SLAVE NUMBER, DENSITY, PARITY AND FORMAT) HAVE NOMINAL VALUES ALREADY STORED IN THE PROGRAM. COUNT, CHARACTER COUNT, TAPE MARK AND STALLS) IS TYPED. ITS PRESENT STORED VALUE IS ALSO PRINTED. IF THESE VALUES NEED NOT BE CHANGED, SIMPLY TYPE A CARRIAGE RETURN AS RESPONSE AND NO CHANGE WILL BE MADE. EACH START OF THE PROGRAM AT 200(8) WILL SHOW THE CURRENT VALUES OF THESE PARAMETERS AS PER THE LAST ENTRY. WHEN A FRESH LOAD OF THE PAPER TAPE IS DONE, THE PARAMETERS WILL REFLECT THE FIXED VALUES STORED IN THE PROGRAM.

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

SAMPLE START AT 200(8):

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

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

TE16 TAPE DRIVE TEST

REGISTER START=172440(172440)  
VECTOR ADDRESS=224(CR)  
DRIVE NUMBER (4)  
SLAVE NUMBER=(5) SN: 5009  
DENSITY=(3)  
PARITY=(0)  
FORMAT=(14)  
SLAVE NUMBER=(2) 9 CHAN SN: 0022  
DENSITY=(3)  
PARITY=(1)  
FORMAT=(15)  
SLAVE NUMBER=(CR)  
RECORD COUNT=100 (500)(CR)  
CHARACTER COUNT=200 (38)?(7)(CR)  
PATTERN NUMBER=1 (22)  
?  
(6)(CR)  
TH=(0)  
INTERCHANGE READ=(1)  
SINGLE PASS=(0)  
  
ENTER STALLS  
READ=1 (CR)  
WRITE=1 (CR)  
TURN AROUND=1 (3000)(CR)

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

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

5. DATA PATTERNS

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

THE FOLLOWING IS A LIST OF THE DATA PATTERNS AVAILABLE:

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

6. RANDOMIZATION  
-----

THERE ARE THREE (3) VALUES THAT MAY BE GENERATED RANDOMLY: DATA, CHARACTER COUNT, AND RECORD COUNT. THESE ARE NORMALLY SET TO SOME FIXED VALUE BUT MAY BE RANDOMIZED BY SETTING THE APPROPRIATE 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 PERFORMED 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 UNSUCCESSFUL, A SKIP ERASE IS DONE, A SUSPECTED BAD TAPE SPOT IS LOGGED AT THIS BLOCK AND RECORD NUMBER, AND A SECOND RETRY OF FOUR REPEATS IS DONE. IF AFTER FOUR (4) RETRIES, THE RECORD CANNOT BE RECOVERED A NOTIFICATION IS PRINTED, AND TESTING IS RESUMED ON THE NEXT RECORD. IF 20(8) BAD TAPE SPOTS ARE FOUND, THE SLAVE WILL BE REWOUND AND REMOVED FROM TESTING WITH AN APPROPRIATE MESSAGE PRINTED. THE READ RETRY SCHEME CONSISTS OF REREADING THE RECORD UP TO EIGHT TIMES. IF ALL EIGHT REREADS ARE BAD, IT IS A HARD ERROR. IF ANY REREAD IS SUCCESSFUL, THIS IS A SOFT ERROR. IF THE ORIGINAL ERROR IS OF THE NON-RETRYABLE TYPE (IE: ILF, RMR, ILR, NEF, CBUSPE), THE RETRY SCHEME IS NOT ENTERED AND A MESSAGE IS PRINTED.
- SW5: SWITCH FIVE (5) WHEN SET DURING A READ FORWARD OR REVERSE WILL CAUSE THE TAPE TO CONTINUOUSLY READ THE CURRENT RECORD BY SPACING EITHER FORWARD OR REVERSE AND REREADING THAT RECORD. THIS TAPE MOVEMENT IS CALLED YOZZLING. THERE IS A SOFTWARE DELAY EXECUTED BETWEEN EACH SPACE/READ OF THE RECORD AND IT MAY BE VARIED BY TYPING CONTROL C ON THE TELETYPE DURING THE EXECUTION OF THE YOZZLE AND RESPONDING TO THE PRINTED REQUEST WITH A SIX (6) DIGIT VALUE. THE YOZZLE STALL IS PRESET TO A VALUE OF 3000 IN THE PROGRAM TO PREVENT EXCESSIVE TAPE WEAR, BUT MAY BE SET TO ANY VALUE THROUGH THE TELETYPE.
- SW6-8: THESE THREE (3) SWITCHES CONTROL THE RANDOMIZATION OF DATA AND BLOCK SIZE AND MAY BE SET AND RESET AT ANY TIME. THE ACTUAL CHANGE WILL TAKE PLACE BETWEEN BLOCK CYCLES
- SW9: SWITCH NINE (9) WHEN SET WILL CAUSE ALL AVAILABLE TAPE UNITS TO BE REWOUND AT THE END OF THE CURRENT BLOCK CYCLE. TESTING WILL BE RESUMED AT A BLOCK COUNT OF ONE (1) WHEN ALL UNITS HAVE REACHED BOT.

- SW10-13: THESE SWITCHES ARE USED TO CONTROL THE ERROR HANDLING TO BE DONE ON THE TAPE OPERATION DESCRIBED BY SWITCHES 0-3.
- A. SWITCH TEN (10) WHEN SET TO A ONE WILL DISALLOW ANY ERROR PRINTOUTS MADE ON THE OPERATION IN PROGRESS. CATASTROPHIC FAILURES AND INFORMATION PRINTOUTS WILL STILL OCCUR. IE: UNIT NOT AVAILABLE, ILLEGAL BOT, DROP OR PICK OVERFLOW, AND EOT REWIND.
  - B. SWITCH ELEVEN (11) WHEN SET TO A ONE WILL DISALLOW THE CHECKING FOR STATUS ERRORS ON READ (FORWARD OR REVERSE) OPERATIONS.
  - C. SWITCH TWELVE (12) WHEN SET TO A ONE WILL DISALLOW THE CHECKING FOR STATUS ERRORS ON WRITE OPERATIONS.
  - D. SWITCH THIRTEEN (13) WHEN SET TO A ONE WILL DISALLOW THE CHECKING OF READ DATA. THIS SWITCH HAS NO EFFECT ON STATUS CHECKING.

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

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

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

9. ERROR PRINTOUTS

THERE ARE THREE TYPES OF ERROR PRINTOUTS MADE BY THE PROGRAM: OPERATION ERRORS, DATA ERRORS, AND CONDITION ERRORS. EACH ERROR MESSAGE PRINTED IS PRECEDED 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 TMO3 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 APPROPRIATE 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 UNTILL ARE  
RESTARTED AT BLOCK ONE.
8. HARD READ ERROR: IF ANY HARD READ ERROR IS ENCOUNTERED  
DURING A RETRY, A MESSAGE IS PRINTED  
REGARDLESS OF THE SETTING OF SW10.
9. NON-RETRYABLE: IF ANY NON-RETRYABLE ERROR IS ENCOUNTERED, A  
MESSAGE IS PRINTED REGARDLESS OF THE SETTING OF SW10.

## D. EXAMPLES:

## GLOSSARY:

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

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

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

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

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



EXAMPLE 3: IN THIS EXAMPLE, THE HEADER IS THE SAME AS  
IN EXAMPLE TWO (2) EXCEPT THAT THE ERROR TYPE  
REFLECTS A READ ERROR IN THE FORWARD  
DIRECTION. IT IS NORMAL FOR THE SYSTEM  
TO DETECT AN ERROR IN THE FORWARD AND  
REVERSE DIRECTION AT THE SAME RECORD.  
REMEMBER THAT IN REVERSE OPERATIONS THE  
RECORD NUMBER IS COUNTED DOWN SO THAT  
RECORD NUMBER TEN (10) WILL SHOW 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 ACCOMPANYING  
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 TMO3. THE ONLY OPERATOR RESPONSE IS TO THE TYPED REQUESTS FOR THE RH ADDRESS, VECTOR, CONTINUOUS OR SINGLE CYCLE, AND NRZ ONLY. ALL SWITCHES REMAIN ACTIVE AND MAY BE USED NORMALLY; HOWEVER THE IDEA IS TO LEAVE ALL SWITCHES DOWN AND ALLOW FULL EXECUTION OF THE TEST PLAN FOR SYSTEM CHECKOUT.

SAMPLE START AT 240(8): AUTO SEQUENCE.

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

TE16 AUTO SEQUENCE TEST  
ENTER CONDITIONS IN OCTAL

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

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

AS EACH TMO3 AND ITS SLAVES ARE FOUND, A DIVIDER LINE OF ASTERICKS WILL BE PRINTED FOLLOWED BY A PRINTOUT OF THE TMO3 AND ITS SLAVES BEING TESTED. AS EACH TMO3 AND ITS SLAVES ARE FINISHED, ANOTHER DIVIDER IS PRINTED BEFORE TESTING IS RESUMED ON THE NEXT AVAILABLE DRIVE.

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

AUTO SEQUENCE TEST PLAN:

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

1. NRZ CYCLE:

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

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

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

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

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

12. TESTING PROCEDURES  
-----

AS PREVIOUSLY STATED THIS PROGRAM CONTAINS NO FIXED TESTS. THE ENTIRE TEST CYCLE TO BE EXECUTED IS DESCRIBED BY THE OPERATOR THROUGH 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 PERFORMED BY USE OF THE RANDOMIZATION CAPABILITY. PERHAPS A CYCLE OF RANDOM TESTING MIGHT BE SET UP AND ALLOWED TO RUN FOR SOME PERIOD OF TIME, THE STATISTICAL COLLECTION OF DROPS AND PICKS IS THEN SIGNIFICANT. INTERMITTANT PROBLEMS CAN BE FOUND BY SETTING THE DESIRED OPERATION IN MOTION AND DISALLOWING ERROR PRINTOUTS WHILE ALLOWING A HALT ON ERROR. THE ERROR THAT CAUSED THE HALT CAN BE PRINTED BY RESETTING CONSOLE SWITCH TEN AND PRESSING CONTINUE. IF SOME PARTICULAR DATA PATTERN SHOULD BE CAUSING DATA ERROR, USE OF THE YOZZLE SWITCH AND ITS ASSOCIATED STALL WILL TO ALLOW SCOPING OF THIS PARTICULAR RECORD.

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

TRY IT, YOU'LL LIKE IT.

1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315

```
.LIST BIN,LOC,SEQ
.TITLE TM03/TE16 DATA RELIABILITY PROGRAM
;MAINDEC-1-DZTED-A-D
;21 FEB 1977
;R. BARNES
.MCALL .SACT11, .SEOP, $SAVE, $RESTORE, $CHAIN
.NLIST MC
.LIST ME
```



1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352

.ENABLE ABS,AMA

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

;SW15: 1=STOP ON ERROR  
0=CONTINUE ON ERROR  
;SW14: 1=PRINT READ/WRITE STATS  
0=DO NOT PRINT STATS  
;SW13: 1=DO NOT CHECK DATA  
0=CHECK DATA  
;SW12: 1=DO NOT CHECK WRITE ERRORS  
0=CHECK WRITE ERRORS  
;SW11: 1=DO NOT CHECK READ ERRORS  
0=CHECK READ ERRORS  
;SW10: 1=DO NOT PRINT ERRORS  
0=PRINT ERRORS  
;SW9: 1=REWIND TAPE  
0=DO NOT REWIND  
;SW8: 1=USE RANDOM DATA  
0=USE FIXED DATA PATTERN  
;SW7: 1=USE RANDOM CHARACTER COUNT  
0=USE FIXED CHAR COUNT  
;SW6: 1=USE RANDOM RECORD COUNT  
0=USE FIXED RECORD COUNT  
;SW5: 1=YOZZLE ON CURRENT RECORD  
0=DO NOT YOZZLE  
;SW4: 1=DO BOTH READ AND WRITE RETRIES  
0=INHIBIT RETRIES  
;SW3: 1=DO NOT READ FORWARD  
0=READ FORWARD  
;SW2: 1=DO NOT READ REVERSE  
0=READ REVERSE  
;SW1: 1=READ FORWARD FIRST  
0=READ REVERSE FIRST  
;SW0: 1=DO NOT WRITE  
0=WRITE

;IF SWR <15::00> = 177777 OR NOT AVAILABLE USE SOFTWARE SWITCH REGISTER



```

1399                                     ;REGISTER EQUIVS*****
1400
1401                                000000      R0=%0
1402                                000001      R1=%1
1403                                000002      R2=%2
1404                                000003      R3=%3
1405                                000004      R4=%4
1406                                000005      R5=%5
1407                                000006      SP=%6
1408                                000007      PC=%7
1409                                000240      NOP=240
1410
1411                                     ;TRAP CATCHERS*****
1412
1413                                .=20
1414                                000020      .WORD TTOUT      ;SET IOT TRAP TO TTOUT ROUTINE
1415                                000022      000340      .WORD 340          ;PRIORITY LEVEL 7
1416
1417                                000004      TYPE=IOT      ;EQUATE TYPE TO AN IOT INSTRUCTION
1418                                000034      .=34
1419                                000034      024006      .WORD OCTP      ;SET TRAP TRAP TO OCTP ROUTINE
1420                                000036      000340      .WORD 340
1421                                104400      TYPOCT=TRAP    ;EQUATE TYPOCT TO TRAP INSTRUCTION
1422
1423                                     ;ACT11 HOOK *****
1424                                000040      $SVPC=.          ;SAVE CURRENT LOCATION CTR
1425                                000046      .=46
1426                                000046      005022      .WORD SENDAD    ;SET LOCATION 46
1427                                000052      .=52
1428                                000052      000000      .WORD 0          ;SET LOCATION 52 = 0
1429                                000040      .=$SVPC        ;RESTORE LOCATION CTR
1430
1431                                     ;TTY INTERRUPT VECTOR*****
1432                                .=60
1433                                000060      021466      .WORD TTINT     ;TTY INTERRUPT HANDLER ADDRESS
1434                                000062      000340      .WORD 340     ;PRIORITY LEVEL 7
1435
1436                                     ;SOFTWARE SWITCH REGISTER*****
1437                                     ;INVOKED IF SWR <15::00> = 177777 OR NOT AVAILABLE
1438                                .=176
1439                                000176      000000      SWREG: .WORD 0
1440
1441                                     ;START ADDRESS*****
1442                                .=200
1443                                000200      000137      003026      JMP START      ;ENTER PARAMETERS VIA TTY
1444
1445                                .=204
1446                                000204      000137      003152      JMP STARTC     ;USE FIXED PARAMETERS; HOLD DATA
1447
1448                                .=210
1449                                000210      005037      015022      CLR RDFL
1450                                000214      000137      003160      JMP STARTA     ;USE FIXED PARAMETERS; NEW DATA
1451
1452                                     ;MAG TAPE INTERRUPT VECTOR*****
1453
1454                                000224      .=224
  
```

1455	000224	021716	
1456	000226	000340	
1457			
1458			
1459			
1460		000240	
1461	000240	005237	000736
1462	000244	000137	003136

```

MTINT           ;MAG TAPE INTERRUPT HANDLER ADDRESS
340
;AUTO SEQUENCE START*****
.=240
INC  ASEQF      ;SET AUTO SEQUENCE FLAG
JMP  STAUT      ;GO TO START OF AUTO SEQUENCE

```

```

1463 ;SHORT CONVERSATION RESTART*****
1464
1465      000300      000300      . =300
1466 000300 005237 014062      INC      SCVFL      ;SET SHORT CONVERSATION FLAG
1467 000304 000137 003026      JMP      START      ;ENTER SHORT PARAMETER LIST
1468
1469      000510      . =510
1470 ;TU16 REGISTER EQUIVS*****
1471
1472 000510 172440      C1:      172440
1473 000512 172442      MC:      172442
1474 000514 172444      BA:      172444
1475 000516 172446      FC:      172446
1476 000520 172450      CS:      172450
1477 000522 172452      DS:      172452
1478 000524 172454      ER:      172454
1479 000526 172456      AS:      172456
1480 000530 172460      CC:      172460
1481 000532 172462      DB:      172462
1482 000534 172464      MR:      172464
1483 000536 172466      DT:      172466
1484 000540 172470      SN:      172470
1485 000542 172472      TC:      172472
1486
1487 ;CONSTANTS*****
1488
1489 000544 172440      REGS:     172440      ;STARTING REGISTER ADDRESS (CS1)
1490 000546 000224      VECT:     224        ;VECTOR ADDRESS (RH INTERRUPT)
1491 000550 000000      DVN:      0          ;DRIVE NUMBER
1492 000552 000000      UDES:     0          ;UNIT DESCRIPTION (PARITY,DENSITY,UNIT,FORMAT)
1493 000554 000100      RCNT:     100       ;RECORD COUNTER
1494 000556 177400      FMCNT:    177400    ;NUMBER OF CHAR (4 - 4000) OCTAL IN TWOS COMPLEMENT
1495 000560 000001      PATRN:    1          ;DATA PATTERN SELECTOR (0 - 15) OCTAL
1496 000562 000002      RDCMD:    2          ;READ COMMAND
1497 000564 000001      TMEX:     1          ;TAPE MARK FLAG: 1=TM 0=NO TM
1498 000566 000000      CRCC:     0          ;CRC CORRECTION FLAG (YES=1 NO=0)
1499 000570 000000      INTRF:    0          ;INTERCHANGE READ 1=YES 0=NO
1500 000572 000000      SPFLG:    0          ;SINGLE PASS 1=YES 0=NO
1501 000574 000001      RSTAL:    1          ;READ STALL
1502 000576 000001      WSTAL:    1          ;WRITE STALL
1503 000600 000001      TSTAL:    1          ;TURN AROUND STAL
1504 000602 002000      YSTAL:    2000      ;YOZZLE STAL
1505 000604 000010      RETRY:    10         ;READ RETRY NUMBER
1506 000606 177776      PSW:      177776    ;PROCESSOR STATUS
1507 000610 177570      SWR:      177570    ;CONSOLE SWITCHES
1508 000612 177560      TKS:      177560    ;TTY READ STATUS REGISTER
1509 000614 177562      TKB:      177562    ;TTY READ BUFFER
1510 000616 177564      TPS:      177564    ;TTY PUNCH STATUS REGISTER
1511 000620 177566      TPB:      177566    ;TTY PUNCH OUTPUT REGISTER
1512 000622 177550      PRS:      177550    ;H/S READER STATUS REGISTER
1513 000624 177552      PRB:      177552    ;H/S READER BUFFER
1514 000626 153624      RANBAS:   153624    ;RANDOM NUMBER GENERATOR BASE
1515 000630 032561      RANSAY:   032561    ;RANDOM NUMBER BUFFER
1516 000632 000100      RCSAY:    100       ;RECORD COUNT SAVE
1517 000634 177400      FCSAY:    177400    ;FRAME COUNT SAVE

```

1518					
1519					
1520					
1521	000636	000000	TINF:	0	;TTY ENTRY FLAG
1522	000640		STFLG:		
1523	000640	000000	T08:	0	;TTY OUTPUT BUFFER
1524	000642	000000	TIB:	0	;TTY INPUT BUFFER
1525	000644	000000	TEMP1:	0	;TEMP STORAGE
1526	000646	000000	TEMP2:	0	;TEMP STORAGE
1527	000650	000000	TEMP3:	0	;TEMP STORAGE
1528	000652	000000	NRZOF:	0	;NRZ ONLY FLAG
1529	000654	000000	EMADDR:	0	;ERROR MSG ADDRESS STORAGE
1530	000656	000000	BLCNTR:	0	;BLOCK COUNTER
1531	000660	000000	BBC:	0	;BAD RECORD COUNTER
1532	000662	000000	EOTREC:	0	;EOT FLAG
1533	000664	000000	RTRN:	0	;INTERRUPT RETURN STORAGE
1534	000666	000000	HDRFL:	0	;HEADER FLAG
1535	000670	000000	STAL:	0	;DELAY STORAGE
1536	000672	000000	PLG:	0	;PRINT FLAG
1537	000674	000000	MTC1:	0	;MAG TAPE CONT REGISTER BUFFER
1538	000676	000000	UNP:	0	;UNIT TABLE POINTER
1539	000700	000000	TMFLG:	0	;TAPE MARK FLAG
1540	000702	000000	RPCNT:	0	;REPEAT COUNTER
1541	000704	000000	RTCNT:	0	;RETRY COUNTER
1542	000706	000000	DERFL:	0	;DATA ERROR FLAG
1543	000710	000000	SERFL:	0	;STATUS ERROR FLAG
1544	000712	000000	BCNT:	0	;BIT COUNTER
1545	000714	000000	RTYFL:	0	;RETRY FLAG
1546	000716	000000	UPS:	0	;UNIT POINTER SAVE
1547	000720	000000	BOPP:	0	;BITS DROPPED POINTER
1548	000722	000000	BPKP:	0	;BITS PICKED POINTER
1549	000724	000000	ERSAV:	0	;ERROR SAVE LOC
1550	000726	000000	BTFLG:	0	;BAD TAPE FLAG
1551	000730	000000	BTSTF:	0	;STATISTIC PRINT FLAG
1552	000732	000000	BTPT:	0	;BAD TAPE POINTER
1553	000734	000000	ERTFL:	0	;ERASE FLAG
1554	000736		ENDFLG:		
1555	000736	000000	ASEGF:	0	;AUTO SEQ FLAG
1556	000740	000000	ADRVN:	0	;UTO SEQ DRIVE NUMBER
1557	000742	000000	ABLNT:	0	;AUTO BLOCK COUNTER
1558	000744	000001	ASEGCF:	1	;AUTO SEQ CONTINUOUS FLAG

;FLAGS AND COUNTERS\*\*\*\*\*

1559  
1560  
1561  
1562 000746 000000  
1563 000750 000000  
1564 000752 000000  
1565 000754 000000  
1566 000756 000000  
1567 000760 000000  
1568 000762 000000  
1569 000764 000000  
1570 000766 177777  
1571  
1572  
1573  
1574 000770 001210  
1575 000772 001230  
1576 000774 001250  
1577 000776 001270  
1578 001000 001310  
1579 001002 001330  
1580 001004 001350  
1581 001006 001370  
1582 001010 001410  
1583 001012 001430  
1584 001014 001450  
1585 001016 001470  
1586 001020 001510  
1587 001022 001530  
1588 001024 001550  
1589 001026 001570  
1590  
1591  
1592  
1593 001030 001610  
1594 001032 001714  
1595 001034 002020  
1596 001036 002124  
1597 001040 002230  
1598 001042 002334  
1599 001044 002440  
1600 001046 002544  
1601  
1602  
1603  
1604  
1605 001050  
1606 001050 000000  
1607 001052 000000  
1608 001054 000000  
1609 001056 000000  
1610 001060 000000  
1611 001062 000000  
1612 001064 000000  
1613 001066 000000  
1614

;UNIT ORDER AND DESCRIPTION TABLE \*\*\*\*\*

UN1: 0  
UN2: 0  
UN3: 0  
UN4: 0  
UN5: 0  
UN6: 0  
UN7: 0  
UN8: 0  
UNX: -1

;THIS TABLE IS LOADED  
;WITH UNIT NUMBERS AND  
;THEIR DESCRIPTIONS IN  
;THE ORDER THAT THEY  
;WILL BE TESTED

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

PIK1: BP00  
PIK2: BP10  
PIK3: BP20  
PIK4: BP30  
PIK5: BP40  
PIK6: BP50  
PIK7: BP60  
PIK8: BP70  
DRP1: B000  
DRP2: B010  
DRP3: B020  
DRP4: B030  
DRP5: B040  
DRP6: B050  
DRP7: B060  
DRP8: B070

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

BTADDR: BT00  
BT01  
BT02  
BT03  
BT04  
BT05  
BT06  
BT07

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

;SET START OF STATISTICS TABLE

STTBL:  
RTY1: 0  
RTY2: 0  
RTY3: 0  
RTY4: 0  
RTY5: 0  
RTY6: 0  
RTY7: 0  
RTY8: 0

```

1615
1616
1617 001070 000000
1618 001072 000000
1619 001074 000000
1620 001076 000000
1621 001100 000000
1622 001102 000000
1623 001104 000000
1624 001106 000000
1625
1626 ;
1627
1628 001110 000000
1629 001112 000000
1630 001114 000000
1631 001116 000000
1632 001120 000000
1633 001122 000000
1634 001124 000000
1635 001126 000000
1636
1637
1638
1639 001130 000000
1640 001132 000000
1641 001134 000000
1642 001136 000000
1643 001140 000000
1644 001142 000000
1645 001144 000000
1646 001146 000000
1647
1648
1649
1650 001150 000000
1651 001152 000000
1652 001154 000000
1653 001156 000000
1654 001160 000000
1655 001162 000000
1656 001164 000000
1657 001166 000000
1658
1659
1660
1661 001170 000000
1662 001172 000000
1663 001174 000000
1664 001176 000000
1665 001200 000000
1666 001202 000000
1667 001204 000000
1668 001206 000000

```

```

;UNIT WRITE ERRORS*****
WTER1: 0
WTER2: 0
WTER3: 0
WTER4: 0
WTER5: 0
WTER6: 0
WTER7: 0
WTER8: 0

;UNIT READ FORWARD ERRORS*****
RDER1: 0
RDER2: 0
RDER3: 0
RDER4: 0
RDER5: 0
RDER6: 0
RDER7: 0
RDER8: 0

;UNIT DATA ERRORS FORWARD*****
DATER1: 0
0
0
0
0
0
0
0
0

;UNIT READ REVERSE ERRORS*****
RDERR1: 0
0
0
0
0
0
0
0
0

;UNIT DATA ERRORS REVERSE*****
DEREV1: 0
0
0
0
0
0
0
0

```



```

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

```

1705  
 1706  
 1707  
 1708 001610 000000  
 1709 001714 001714  
 1710 001714 000000  
 1711 002020 002020  
 1712 002020 000000  
 1713 002124 002124  
 1714 002124 000000  
 1715 002230 002230  
 1716 002230 000000  
 1717 002334 002334  
 1718 002334 000000  
 1719 002440 002440  
 1720 002440 000000  
 1721 002544 002544  
 1722 002544 000000  
 1723 002650 002650  
 1724  
 1725  
 1726  
 1727 002650 000000  
 1728 002652 000000  
 1729 002654 000000  
 1730 002656 000000  
 1731 002660 000000  
 1732 002662 000000  
 1733 002664 000000  
 1734 002666 000000  
 1735  
 1736  
 1737  
 1738 002670 000000  
 1739 002672 000000  
 1740 002674 000000  
 1741 002676 000000  
 1742 002700 000000  
 1743 002702 000000  
 1744 002704 000000  
 1745 002706 000000  
 1746  
 1747  
 1748  
 1749 002710 000000  
 1750 002712 000000  
 1751 002714 000000  
 1752 002716 000000  
 1753 002720 000000  
 1754 002722 000000  
 1755 002724 000000  
 1756 002726 000000  
 1757

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

BT00: 0  
 .=.+102  
 BT01: 0  
 .=.+102  
 BT02: 0  
 .=.+102  
 BT03: 0  
 .=.+102  
 BT04: 0  
 .=.+102  
 BT05: 0  
 .=.+102  
 BT06: 0  
 .=.+102  
 BT07: 0  
 .=.+102

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

EOTCO: 0  
 0  
 0  
 0  
 0  
 0  
 0  
 0  
 0

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

RFSOFT: 0  
 0  
 0  
 0  
 0  
 0  
 0  
 0  
 0

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

RRSOFT: 0  
 0  
 0  
 0  
 0  
 0  
 0  
 0  
 0

1758  
1759  
1760  
1761 002730 000000  
1762 002732 000000  
1763 002734 000000  
1764 002736 000000  
1765 002740 000000  
1766 002742 000000  
1767 002744 000000  
1768 002746 000000  
1769  
1770  
1771  
1772 002750 000000  
1773 002752 000000  
1774 002754 000000  
1775 002756 000000  
1776 002760 000000  
1777 002762 000000  
1778 002764 000000  
1779 002766 000000  
1780  
1781 002770  
1782  
1783  
1784  
1785 002770 002770  
1786 002772 014274  
1787 002774 014434  
1788 002776 014454  
1789 003000 014460  
1790 003002 014504  
1791 003004 014514  
1792 003006 014522  
1793 003010 014530  
1794 003012 014556  
1795 003014 014606  
1796 003016 014626  
1797 003020 014650  
1798 003022 014660  
1799 003024 014710  
1800

;UNIT READ FORWARD HARD ERROR\*\*\*\*\*

RFHARD: 0  
0  
0  
0  
0  
0  
0  
0

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

RRHARD: 0  
0  
0  
0  
0  
0  
0  
0

;SET END OF STATISTICS TABLE  
ENDTBL:

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

DATBL: .  
DATA0: DAT0  
DATA1: DAT1  
DATA2: DAT2  
DATA3: DAT3  
DATA4: DAT4  
DATA5: DAT5  
DATA6: DAT6  
DATA7: DAT7  
DATA10: DAT10  
DATA11: DAT11  
DATA12: DAT12  
DATA13: DAT13  
DATA14: DAT14  
DATA15: DAT15

;ENTRY TABLE  
;EXTERNAL INPUT FROM H/S READER(SEE MAINDEC-11-DZTUF)  
;ALL ONES  
;ALL ZEROS  
;WALKING ONE  
;WALKING ZERO  
;ALTERNATING ONE/ZERO  
;ALTERNATING ZERO/ONE  
;ALTERNATING ONE/ZERO IN ALTERNATING CHARACTERS  
;WALKING ONE/ALL ONE IN ALTERNATING CHARACTERS  
;ALL BITS 0-377  
;ALL BITS 377-0  
;ALTERNATING CHARACTERS 0 AND 377  
;WALKING ZERO/ALL ZERO IN ALTERNATING CHARACTERS  
;AUTO SEQUENCE PATTERN 0,0,-1,-1,-1,0,0

1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856

003026 012706 000500  
003032 005037 000736  
003036 005027  
003040 000000  
003042 022737 005022 000042  
003050 001404  
003052 005737 000042  
003056 001413  
003060 000406  
003062 012737 000176 000610  
003070 012777 100000 175512  
003076 005237 003040  
003102 000137 003126  
003106 122737 000006 000041  
003114 001010  
003116 012704 027232  
003122 000004  
003124 000404  
003126 005237 000736  
003132 000137 021766  
003136 012737 000001 000636  
003144 005037 01502.  
003150 000405  
003152 005037 000636  
003156 000432  
003160 005037 000636  
003164 012700 000640  
003170 012701 000076  
003174 105020  
003176 005301

```

.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*****
START:  MOV    #500,SP          ;SET STACK PTR
        CLR    ASEQF          ;CLEAR AUTO SEQUENCE FLAG
        CLR    (PC)+          ;CLEAR CHAIN INDICATOR
CHNFLG: .WORD  0              ;CHAIN MODE INDICATOR
                                ;1/0 = CHAIN/NOT CHAIN MODE
                                ;;BRANCH IF LOADED VIA ACT11 CHAIN MODE
        CMP    #SENDAD,2#42
        BEQ   50$
        TST   2#42            ;;BRANCH IF IN DUMP MODE
        BEQ   52$
        BR    51$
50$:   MOV    #SWREG,SWR      ;; INVOKE SOFTWARE SWR
        MOV   #100000,2SWR   ;; WITH HALT ON ERROR SET
51$:   INC    CHNFLG         ;; SET CHNFLG = CHAIN MODE
        JMP   3$            ;; GO TO CHAIN ADDRESS
52$:   CMPB   #6,2#41        ;BRANCH IF LOADED VIA TMDP
        BNE   STAUT
        MOV   #MSG120,R4    ;ADVISE USER TO REMOVE TMDP FROM SLAVE
        TYPE
        BR    STAUT
3$:    INC    ASEQF          ;SET AUTO SEQUENCE FLAG
        JMP   ASEQ0        ;GO TO AUTO SEQUENCER

;START 240*****
STAUT: MOV    #1,TINF        ;SET TTY ENTRY FLAG
        CLR   R0FL          ;CLEAR RANDOM DATA FLAG
        BR   STARTB

;START 204*****
STARTC: CLR   TINF          ;CLEAR TTY INPUT FLAG
        BR   STARTD

;START 210*****
STARTA: CLR   TINF          ;CLEAR TTY ENTRY FLAG
STARTB: MOV   #STFLG,R0     ;GET STARTING ADDRESS OF FLAGS
        MOV   #ENDFLG-STFLG,R1
1$:    CLRB   (R0)+         ;CLEAR FLAGS AND COUNTERS
        DEC   R1

```

1857	003200	001375				BNE	1\$		
1858	003202	012706	000500			MOV	#500, SP	;	SET STACK POINTER
1859	003206	004737	004276			JSR	PC, RANSET	;	GO RESET RANDOM BASE
1860	003212	012700	001050			MOV	#STBL, R0	;	GET STARTING ADDRESS OF STAT TABLE
1861	003216	012701	001720			MOV	#ENDTBL - STBL, R1	;	AND # OF BYTES IN TABLE
1862	003222	105020			2\$:	CLRB	(R0)+	;	CLEAR STATISTIC COUNTERS
1863	003224	005301				DEC	R1		
1864	003226	001375				BNE	2\$		
1865	003230	012737	177777	014270		MOV	#-1, PATS	;	PRESET PATTERN
1866	003236	012737	000001	000656	STARTE:	MOV	#1, BLCNTR	;	PRESET BLOCK COUNTER
1867	003244	013746	000004		STARTD:	MOV	#4, -(SP)	;	SAVE ERROR TRAP VECTOR
1868	003250	013746	000006			MOV	#6, -(SP)		
1869	003254	022737	000176	000610		CMP	#SWREG, SWR	;	BRANCH IF SOFTWARE SWR
1870	003262	001413				BEQ	2\$	;	ALREADY SELECTED
1871	003264	012737	003310	000004		MOV	#1\$, #4	;	SET TIMEOUT TRAP TO 1\$ BELOW
1872	003272	005037	000006			CLR	#6		
1873	003276	022777	177777	175304		CMP	#177777, #SWR	;	BRANCH IF SWR = 177777 TRAP
1874	003304	001402				BEQ	2\$	;	IF NOT AVAIL (1\$) OTHERWISE
1875	003306	000404				BR	3\$	;	GO TO 3\$
1876	003310	022626			1\$:	CMP	(SP)+, (SP)+	;	RESET STACK
1877	003312	012737	000176	000610	2\$:	MOV	#SWREG, SWR	;	SET SWR = SOFTWARE SWR
1878	003320	012637	000006		3\$:	MOV	(SP)+, #6	;	RESTORE ERROR TRAP
1879	003324	012637	000004			MOV	(SP)+, #4		
1880	003330	012706	000500			MOV	#500, SP		
1881	003334	004737	012212			JSR	PC, T1NP	;	GO GET PARAMETERS FROM TTY
1882	003340	012777	000040	175152		MOV	#40, #CS	;	INITIALIZE
1883	003346	005000			STAUTO:	CLR	R0	;	POINT TO FIRST ENTRY
1884	003350	022760	177777	000746	1\$:	CMP	#-1, UN1(R0)	;	BRANCH IF LAST ENTRY
1885	003356	001406				BEQ	2\$		
1886	003360	042760	100000	000746		BIC	#100000, UN1(R0)	;	CLEAR EOT FLAG
1887	003366	062700	000002			ADD	#2, R0	;	POINT TO NEXT UNIT ENTRY
1888	003372	000766				BR	1\$	;	CONTINUE CLEARING
1889	003374	013703	005054		2\$:	MOV	REOTC, R3		
1890	003400	000303				SWAB	R3		
1891	003402	110337	005054			MOV	R3, REOTC	;	RESTORE EOT CNTR
1892	003406	012777	000100	175176	START1:	MOV	#100, #TKS	;	SET KEYBOARD IE BIT
1893	003414	013700	000676			MOV	UNP, R0	;	R0 = UNIT TABLE POINTER
1894	003420	022760	177777	000746	STAR1A:	CMP	#-1, UN1(R0)	;	BRANCH IF LAST ENTRY
1895	003426	001404				BEQ	STAR1B		
1896	003430	016037	000746	000552		MOV	UN1(R0), UDES	;	LOAD NEXT UNIT DESCRIPTION
1897	003436	000446				BR	STAR14		
1898	003440	005237	000656		STAR1B:	INC	BLCNTR	;	BUMP BLOCK COUNTER
1899	003444	005737	000736			TST	ASEOF	;	SEE IF AUTO SEQ
1900	003450	001411				BEQ	STAR1C	;	IF NOT: BR
1901	003452	023737	000656	000742		CMP	BLCNTR, #BLCNT	;	SEE IF DONE SEQ
1902	003460	001005				BNE	STAR1C	;	IF NOT: BR
1903	003462	005037	000656			CLR	BLCNTR	;	RESET BLOCK CNTR
1904	003466	005037	000676			CLR	UNP	;	RESET UNIT POINTER
1905	003472	000207				RTS	PC	;	RETURN TO AUTO SEQ
1906	003474	005037	000676		STAR1C:	CLR	UNP		
1907	003500	005000				CLR	R0		
1908	003502	016037	000746	000552		MOV	UN1(R0), UDES	;	LOAD FIRST UNIT DESCRIPTION
1909	003510	032777	000200	175072		BIT	#200, #SWR	;	SEE IF RANDOM RECORD SIZE
1910	003516	001402				BEQ	STAR12	;	IF NOT: BR
1911	003520	004737	012126			JSR	PC, CNTR	;	GO GENERATE RANDOM RECORD SIZE
1912	003524	032777	000400	175056	START2:	BIT	#400, #SWR	;	SEE IF RANDOM DATA

1913	003532	001402			BEQ	START3		; IF NOT: BR
1914	003534	004737	014760		JSR	PC,DATA		; GO GENERATE RANDOM DATA
1915	003540	032777	000100	175042	START3: BIT	#100,JSWR		; SEE IF RANDOM RECORD COUNT
1916	003546	001402			BEQ	START4		; IF NOT: BR
1917	003550	004737	012166		JSR	PC,RCNTR		; GO GENERATE RANDOM RECORD COUNT
1918	003554	005760	000746		START4: TST	UN1(R0)		; SEE IF REACHED EOT
1919	003560	100002			BPL	STAR40		; IF NOT: BR
1920	003562	000137	004264		JMP	START7		; ELSE GO TO NEXT UNIT
1921	003566	013777	000550	174724	STAR40: MOV	DVN,ACS		; SET DRIVE NUMBER
1922	003574	013777	000552	174740	MOV	UDES,ATC		; SET UNIT NUMBER
1923	003602	105777	174714		TSTB	ACS		; SEE IF UNIT AVAIL
1924	003606	100412			BMI	STAR4A		; IF SO: BR
1925	003610	005337	000670		DEC	STAL		
1926	003614	001357			BNE	START4		; AWAIT TUR
1927	003616	004737	022570		JSR	PC,PAPRT		; PRINT HEADER
1928	003622	012704	025731		MOV	#MSG49,R4		
1929	003626	000004			TYPE			; TYPE MSG
1930	003630	000000			HALT			; STOP
1931	003632	000750			BR	START4		; RETRY
1932	003634	004737	014110		STAR4A: JSR	PC,DSUP		; GO SET UP WRITE DATA
1933	003640	004737	005426		JSR	PC,INIT		; INIT SLAVE
1934	003644	004737	005056		JSR	PC,RUND		; REWIND
1935	003650	004737	005534		JSR	PC,WRITE		; WRITE
1936	003654	013737	000600	000670	MOV	TSTAL,STAL		; SET TURN AROUND DELAY
1937	003662	004737	012116		JSR	PC,STALL		; DELAY
1938	003666	004737	007416		JSR	PC,RSEQ		; GO TO READ SEQUENCER
1939	003672	013737	000600	000670	MOV	TSTAL,STAL		; SET TURN AROUND DELAY
1940	003700	004737	012116		JSR	PC,STALL		; DELAY
1941	003704	032777	040000	174676	BIT	#40000,JSWR		; SEE IF SHOULD PRINT STATISTICS
1942	003712	001541			BEQ	START5		; IF NOT: BR
1943	003714	012700	000001		MOV	#1,R0		; SET RECORD COUNTER TO 1
1944	003720	004737	022570		JSR	PC,PAPRT		; PRINT CYCLE NUMBER
1945	003724	004737	003734		JSR	PC,STP		; GO PRINT STAT
1946	003730	000137	004202		JMP	STPX		
1947	003734	004737	017100		STAR4: JSR	PC,DPPRT		; PRINT DROPS AND PICKS
1948	003740	012704	026143		MOV	#MSG65,R4		
1949	003744	000004			TYPE			; TYPE MSG
1950	003746	013704	000676		MOV	UNP,R4		
1951	003752	016403	001050		MOV	RTY1(R4),R3		
1952	003756	104400			TYPOCT			; PRINT RETRIES
1953	003760	012704	026314		MOV	#MSG73,R4		
1954	003764	000004			TYPE			; TYPE MSG
1955	003766	013704	000676		MOV	UNP,R4		
1956	003772	016403	001070		MOV	WTER1(R4),R3		
1957	003776	104400			TYPOCT			; PRINT WRITE ERRORS
1958	004000	012704	026303		MOV	#MSG72,R4		
1959	004004	000004			TYPE			; TYPE MSG
1960	004006	013704	000676		MOV	UNP,R4		
1961	004012	016403	001110		MOV	RDER1(R4),R3		
1962	004016	104400			TYPOCT			; PRINT READ FORWARD ERRORS
1963	004020	012704	027107		MOV	#MSG113,R4		
1964	004024	000004			TYPE			; TYPE MSG
1965	004026	013704	000676		MOV	UNP,R4		
1966	004032	016403	002670		MOV	RFSOFT(R4),R3		
1967	004036	104400			TYPOCT			; PRINT FORWARD SOFT ERRORS
1968	004040	012704	027120		MOV	#MSG114,R4		

```

1969 004044 000004          TYPE          ;TYPE MSG
1970 004046 013704 000676  MOV          UNP,R4
1971 004052 016403 002730  MOV          RFHARD(R4),R3
1972 004056 104400          TYPOCT       ;PRINT HARD FORWARE ERRORS
1973 004060 012704 026374  MOV          #MSG77,R4
1974 004064 000004          TYPE          ;TYPE MSG
1975 004066 013704 000676  MOV          UNP,R4
1976 004072 016403 001130  MOV          DATERR1(R4),R3
1977 004076 104400          TYPOCT       ;PRINT DATA ERROR FORWARD NUMBER
1978 004100 012704 026177  MOV          #MSG68,R4
1979 004104 000004          TYPE          ;TYPE MSG
1980 004106 013704 000676  MOV          UNP,R4
1981 004112 016403 001150  MOV          RDEERR1(R4),R3
1982 004116 104400          TYPOCT       ;PRINT REVESE ERROR NUMBER
1983 004120 012704 027107  MOV          #MSG113,R4
1984 004124 000004          TYPE          ;TYPE MSG
1985 004126 013704 000676  MOV          UNP,R4
1986 004132 016403 002710  MOV          RRSOFT(R4),R3
1987 004136 104400          TYPOCT       ;PRINT REVERSE SOFT ERROR
1988 004140 012704 027120  MOV          #MSG114,R4
1989 004144 000004          TYPE          ;TYPE MSG
1990 004146 013704 000676  MOV          UNP,R4
1991 004152 016403 002750  MOV          RRHARD(R4),R3
1992 004156 104400          TYPOCT
1993 004160 012704 026363  MOV          #MSG76,R4
1994 004164 000004          TYPE          ;TYPE MSG
1995 004166 013704 000676  MOV          UNP,R4
1996 004172 016403 001170  MOV          DERREV1(R4),R3
1997 004176 104400          TYPOCT       ;PRINT DATA REVERSE ERROR NUMBER
1998 004200 000207          RTS          PC          ;RETURN
1999 004202 005237 000730  STPX: INC     BTSTF     ;SET STAT ONLY PRINT
2000 004206 004737 007326  JSR     PC,BTPRT  ;PRINT BAD TAPE STATS
2001 004212 005037 000730  CLR     BTSTF     ;CLEAR FLAG
2002 004216 017700 174366  START5: MOV    @SWR,RO ;LOAD SWR
2003 004222 042700 177762  BIC    #177762,RO ;MASK READ/WRITE SWITCHES
2004 004226 022700 000015  CMP    #15,RO    ;SEE IF HAVE READ OR WRITE
2005 004232 001417          BEQ     START8    ;IF NOT: BR
2006 004234 105777 174262  START6: TSTB   @DS   ;SEE IF HAVE UNIT READY
2007 004240 100411          BMI     START7    ;IF SO: BR
2008 004242 005337 000670  DEC     STAL      ;DELAY FOR TUR
2009 004246 001372          BNE     START6    ;PRINT HEADER
2010 004250 004737 022570  JSR     PC,PAPRT
2011 004254 012704 025731  MOV    #MSG49,R4
2012 004260 000004          TYPE          ;TYPE MSG
2013 004262 000000          HALT         ;STOP
2014 004264 062737 000002 000676 START7: ADD    #2,UNP  ;POINT TO NEXT UNIT
2015 004272 000137 003406  START8: JMP     START1 ;CONTINUE
2016
2017
2018          ;RANDOM BASE RESET*****
2019 004276 012737 153624 000626 RANSET: MOV    #153624,RANBAS ;RESET BASE
2020 004304 012737 032561 000630      MOV    #32561,RANSABV ;RESET BUFFER
2021 004312 013737 000632 000554      MOV    RCSAV,RCNT    ;RESET RECORD COUNT
2022 004320 013737 000634 000556      MOV    FCSAV,FMCNT  ;RESET FRAME COUNT
2023 004326 000207          RTS          PC
2024

```

```

2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037 004330 013777 000552 174204
2038 004336 012777 000011 174144
2039 004344 105777 174152
2040 004350 100375
2041 004352 012777 000007 174130
2042 004360 005737 000726
2043 004364 001004
2044 004366 013700 000662
2045 004372 042700 100000
2046 004376 005037 000662
2047 004402 004737 022570
2048 004406 022737 000002 000726
2049 004414 001003
2050 004416 012704 027000
2051 004422 000406
2052 004424 022737 000001 000726
2053 004432 001004
2054 004434 012704 026626
2055 004440 000004
2056 004442 000412
2057 004444 012704 024636
2058 004450 000004
2059 004452 013704 000676
2060 004456 005264 002650
2061 004462 016403 002650
2062 004466 104400
2063 004470 012704 026653
2064 004474 000004
2065 004476 005037 000726
2066 004502 004737 003734
2067 004506 004737 007326
2068 004512 105777 174004
2069 004516 100414
2070 004520 005337 000670
2071 004524 001372
2072 004526 012737 024475 000654
2073 004534 004737 022570
2074 004540 012704 026105
2075 004544 000004
2076 004546 000000
2077 004550 105337 005054
2078 004554 001410
2079 004556 013700 000676
2080 004562 052760 100000 000746

```

```

*****
REWIND FROM EOT:
*****
WHEN ANY TRANSPORT BEING TESTED REACHES END OF TAPE
DURING A READ OR WRITE OPERATION, IT WILL BE REWOUND
AND FLAGGED AS UNAVAILABLE UNTIL ALL AVAILABLE UNITS
HAVE REACHED EOT AT WHICH TIME ALL TESTING WILL BE RESUMED
AT A BLOCK COUNT OF ONE (1). A MESSAGE WILL BE
PRINTED ON THE SUPERVISORS CONSOLE AS EACH UNIT REACHES
EOT AND IS REWOUND.
*****
REOT:  MOV      UDES, JTC      ;LOAD TAPE CONTROL REGISTER
        MOV      #11, JCI      ;DRIVE CLEAR
IS:    TSTB     JDS           ;WAIT FOR DRY
        BPL      IS
        MOV      #7, JCI      ;START REWIND
        TST      BTFLG        ;SEE IF BAD TAPE OVERFLOW REWIND
        BNE     REOT1A        ;IF SO: BR
        MOV      EOTREC, RO
        BIC      #100000, RO   ;SET RECORD NUMBER OF EOT
REOT1A: CLR      EOTREC        ;CLEAR EOT INDICATOR & REC COUNT
        JSR      PC, PAPRT     ;PRINT HEADER
        CMP      #2, BTFLG     ;SEE IF POSITION ERROR
        BNE     REOT1B        ;IF NOT: BR
        MOV      #MSG109, R4   ;SET POSITION ERROR MSG
        BR      REOT1F
REOT1B: CMP      #1, BTFLG     ;SEE IF BAD TAPE OVERFLOW
        BNE     REOT1C        ;IF NOT: BR
        MOV      #MSG106, R4   ;SET BAD TAPE OVERFLOW MSG
REOT1F: TYPE
REOT1C: MOV      #MSG20, R4    ;SET EOT MSG
        TYPE
        MOV      UNP, R4
        INC      EOTCO(R4)     ;BUMP CNTR
        MOV      EOTCO(R4), R3
REOT1E: MOV      #MSG16A, R4   ;PRINT EOT CNTR
        TYPE
        CLR      BTFLG        ;CLEAR BAD TAPE FLAG
        JSR      PC, STP       ;PRINT STATS
        JSR      PC, BTPRT     ;PRINT BAD TAPE STATS
REOT2: TSTB     JDS           ;BRANCH IF DRY SET
        BMI     REOT2A
        DEC     STAL
        BNE     REOT2
        MOV      #MSG6, EMADDR  ;WAIT DRY
        JSR      PC, PAPRT     ;PRINT HEADER
        MOV      #MSG60, R4
        TYPE
REOT2A: DECB    REOTC          ;SEE IF LAST UNIT TO REACH EOT
        BEQ     REOT3
        MOV      UNP, RO
        BIS      #100000, UN1(RO) ;SET EOT FLAG

```



2081	004570	005726			TST	(SP)+		;RESET STACK POINTER
2082	004572	000137	004264		JMP	START7		;GO TO NEXT UNIT
2083	004576	000337	005054		REOT3: SWAB	REOTC		
2084	004602	013700	005054		MOV	REOTC,R0		
2085	004606	000337	005054		SWAB	REOTC		
2086	004612	110037	005054		MOV	R0,REOTC		;RESTORE EOT UNIT COUNTER
2087	004616	005037	000676		CLR	UNP		
2088	004622	013700	000676		MOV	UNP,R0		;POINT TO FIRST UNIT
2089	004626	016037	000746	000552	REOT4: MOV	UNI(R0),UDES		;LOAD UNIT DESCRIPTION
2090	004634	013777	000552	173700	MOV	UDES,ATC		;LOAD COMMAND REGISTER
2091	004642	032777	020000	173652	REOT5: BIT	#20000,205		
2092	004650	001374			BNE	REOT5		;AWAIT PIP RESET
2093	004652	032777	000002	173642	BIT	#2,205		;SEE IF HAVE BOT
2094	004650	001012			BNE	REOT6		;IF SO: BR
2095	004662	012700	000001		MOV	#1,R0		
2096	004666	004737	022570		JSR	PC,PAPRT		;PRINT HEADER
2097	004672	012704	025676		MOV	#MSG48,R4		
2098	004676	000004			TYPE			;TYPE MSG
2099	004700	000000			HALT			
2100	004702	013700	000676		MOV	UNP,R0		
2101	004706	042760	100000	000746	REOT6: BIC	#100000,UNI(R0)		;CLEAR EOT FLAG
2102	004714	062737	000002	000676	ADD	#2,UNP		
2103	004722	013700	000676		MOV	UNP,R0		;POINT TO NEXT UNIT
2104	004726	022760	177777	000746	CHP	#-1,UNI(R0)		;BRANCH IF NOT LAST UNIT
2105	004734	001334			BNE	REOT4		
2106	004736	005037	000676		REOT7: CLR	UNP		;CLEAR UNIT POINTER
2107	004742	005037	000636		CLR	TINF		;CLEAR TTY INPUT FLAG
2108	004746	005737	000736		TST	ASEQF		;SEE IF AUTO SEQ
2109	004752	001402			BEQ	REOTX		;IF NOT: BR
2110	004754	005726			TST	(SP)+		;RESET STACK POINTER
2111	004756	000207			RTS	PC		;RETURN TO AUTO SEQ
2112	004760	004737	004276		REOTX: JSR	PC,RANSET		;GO RESET RANDOM BASE
2113	004764	012737	177777	014270	MOV	#-1,PATS		;PRESET PATTERN
2114	004772	005037	015022		CLR	RDFL		;CLEAR RANDOM FLAG
2115	004776	005737	000572		TST	SPFLG		;SEE IF SINGLE PASS
2116	005002	001422			BEQ	REOTXX		;IF NOT: BR
2117	005004	012704	026506		TEND: MOV	#MSG100,R4		
2118	005010	000004			TYPE			;TYPE MSG
2119	005012	013700	000042		MOV	#42,R0		;GET ACT11 RETURN ADDRESS
2120	005016	001405			BEQ	HERE		;BRANCH IF NOT ACT11
2121	005020	000005			RESET			
2122	005022	004710			SENDAD: JSR	PC,(R0)		
2123	005024	000240			NOP			
2124	005026	000240			NOP			
2125	005030	000240			NOP			
2126	005032	000240			NOP			
2127	005034	005737	003040		HERE: TST	CHNFLG		;BRANCH IF NOT CHAIN MODE
2128	005040	001402			BEQ	IS		
2129	005042	000137	021766		JMP	ASEQ0		;RETURN TO AUTO SEQUENCER
2130	005046	000000			IS: HALT			
2131	005050	000137	003236		REOTXX: JMP	STARTE		;RESTART AT BLOCK NUMBER ONE
2132	005054	000000			REOTC: 0			;EOT UNIT COUNTER

2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188

005056 032777 001000 173524  
005064 001001  
005066 000207  
000070 013737 000676 000716  
000076 005037 000676  
005102 005037 000662  
005106 000337 005054  
005112 013700 005054  
005116 000337 005054  
005122 110037 005054  
005126 013700 000676  
005132 022760 177777 000746  
005140 001445  
005142 005760 000746  
005146 100433  
005150 016037 000746 000552  
005156 013777 000552 173356  
005164 012777 000011 173316  
005172 012777 000007 173310  
005200 105777 173316  
005204 100414  
005206 005337 000670  
005212 001372  
005214 012737 024475 000654  
005222 004737 022570  
005226 012704 026226  
005232 000004  
005234 000000  
005236 042760 100000 000746  
000044 062737 000002 000676  
005252 000725  
005254 005037 000676  
005260 013700 000676  
005264 022760 177777 000746  
005272 001441  
005274 016037 000746 000552  
005302 013777 000552 173232  
005310 032777 020000 173204  
005316 001374  
005320 013777 000552 173214  
005326 032777 000002 173166  
005334 001407  
005336 062737 000002 000676  
005344 012777 000011 173136  
005352 000742  
005354 012700 000001  
005360 004737 022570

```
*****  
;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.  
*****  
RWIND: BIT #1000,2SWR ;SEE IF SHOULD REWIND  
BNE RWINDA ;IF SO: BR  
RTS PC ;ELSE EXIT  
RWINDA: MOV UNP,UPS ;SAVE UNIT POINTER  
CLR UNP ;CLEAR POINTER  
CLR EOTREC ;CLEAR EDT FLAG  
SWAB REOTC  
MOV REOTC,RO  
SWAB REOTC  
MOVVB RO,REOTC ;RESTORE EOT UNIT COUNTER  
RWIND0: MOV UNP,RO ;POINT TO UNIT ENTRY  
CMP #-1,UNI(RO) ;BRANCH IF LAST ENTRY  
BEQ RWIND2  
TST UNI(RO) ;SEE IF ALREADY REWINDING  
BMI RWIND1A ;IF SO: BR  
MOV UNI(RO),UDES ;SET UNIT DESCRIPTION  
MOV UDES,2TC ;LOAD COMMAND REGISTER  
MOV #11,2C1 ;DRIVE CLEAR  
MOV #7,2C1 ;START REWIND  
1S: TSTB 2DS  
BMI RWIND1A ;IF DRY: BR  
DEC STAL  
BNE 1S ;AWAIT DRY  
MOV #MSG6,EMADDR  
JSR PC,PAPRT ;PRINT HEADER  
MOV #MSG70,R4  
TYPE ;TYPE MSG  
HALT  
RWIND1A: BIC #100000,UNI(RO) ;CLEAR EOT FLAG  
ADD #2,UNP ;BUMP POINTER  
BR RWIND0 ;DO NEXT UNIT  
RWIND2: CLR UNP ;CLEAR POINTER  
RWIND3: MOV UNP,RO ;POINT TO UNIT ENTRY  
CMP #-1,UNI(RO) ;BRANCH IF LAST ENTRY  
BEQ RWINDX  
MOV UNI(RO),UDES ;SET UNIT DESCRIPTION  
MOV UDES,2TC ;LOAD COMMAND REGISTER  
1S: BIT #20000,2DS  
BNE 1S ;AWAIT PIP RESET  
MOV UDES,2TC ;LOAD UNIT DESCRIPTION  
BIT #2,2DS ;SEE IF HAVE BOT  
BEQ RWIND6 ;IF NOT: BR  
RWIND5: ADD #2,UNP ;BUMP POINTER  
MOV #11,2C1 ;DRIVE CLEAR  
BR RWIND3 ;DO NEXT UNIT  
RWIND6: MOV #1,RO  
JSR PC,PAPRT ;PRINT HEADER
```

```

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

```

```

:*****
:INITIALIZE SELECTED SALVE
:THIS ROUTINE REWINDS AND SETS THE PROPER DENSITY IF
:THE DENSITY REQUIRED FOR THE TEST IS DIFFERENT FROM
:THE DENSITY AT WHICH THE SLAVE IS SELECTED.
:*****

```

2225  
2226  
2227  
2228  
2229  
2230  
2231  
2232  
2233  
2234  
2235  
2236  
2237  
2238  
2239  
2240  
2241  
2242  
2243  
2244  
2245  
2246  
2247  
2248  
2249  
2250  
2251  
2252  
2253  
2254  
2255  
2256  
2257  
2258  
2259  
2260  
2261  
2262  
2263  
2264  
2265  
2266  
2267  
2268  
2269  
2270  
2271  
2272  
2273  
2274  
2275  
2276  
2277  
2278  
2279  
2280

005534 032777 000001 173046  
005542 001402  
005544 000137 006326  
005550 013700 000554  
005554 012737 024470 000654  
005562 013777 000556 172726  
005570 012777 027326 172716  
005576 112737 000060 000674  
005604 012737 005616 000664  
005612 000137 021054  
005616 032777 002000 172676  
005624 001412  
005626 005737 000662  
005632 100407  
005634 010037 000662  
005640 062737 100001 000662  
005646 012700 000002  
005652 032777 010000 172730  
005660 001002  
005662 004737 017236  
005666 013737 000576 000670  
005674 004737 012116

```
*****  
WRITE ROUTINE:  
THIS ROUTINE IS USED TO WRITE ONTO TAPE THE BLOCK  
OF DATA DESCRIBED BY THE OPERATOR AND SET UP  
IN THE SEQUENCE FORMATTER. THE TAPE UNIT TO BE USED  
MAY BE ASSIGNED BY THE SEQUENCE FORMATTER AND  
ITS PARAMETERS SET IN A UNIT DESCRIPTION WORD.  
AS EACH RECORD OF THE BLOCK IS WRITTEN, IT IS CHECKED  
FOR STATUS ERRORS, WORD COUNT ZERO, AND CORRECT CURRENT  
MEMORY ADDRESS. IF THE WRITE OPERATION RESULTS IN  
ANY ERROR CONDITION, A WRITE RETRY OF THAT OPERATION  
MAY BE DONE BY SETTING SWITCH FOUR (4) TO A ONE (1).  
THE RETRY CONSISTS OF A BACKSPACE, ERASE FORWARD, AND  
REWRITE OF THE RECORD. (SEE WRITE RETRY SUBROUTINE)  
AFTER ALL DATA RECORDS IN THE BLOCK HAVE BEEN  
WRITTEN, THE WRITE ROUTINE WILL EXECUTE A WRITE  
TAPE MARK COMMAND IF THE TTY RESPONSE TM=1 WAS  
MADE AT INITIAL START. THE TM IS COUNTED AS TOTAL  
DATA RECORDS PLUS ONE (IE: IF 100 DATA RECORDS TM=RECORD 101)  
IF THE WRITE OPERATION (DATA OR TM) CAUSES THE ELECTED SLAVE  
TO REACH END OF TAPE (EOT) AND THERE IS TO BE NO READING DONE,  
(SW2 AND SW3 SET TO A 1) THEN THE SLAVE IS REWOUND AND  
FLAGGED AS UNAVAILABLE FOR TESTING UNTIL ALL SLAVES HAVE  
REACHED EOT AND BEEN REWOUND AT WHICH TIME TESTING IS  
RESUMED ON ALL AVAILABLE SLAVES.  
WRITE RETRY MAY BE ALLOWED VIA CONSOLE SWITCH FOUR (4).  
ERROR CHECKING MAY BE DISALLOWED VIA CONSOLE SWITCH  
TWELVE (12).  
WRITING TO TAPE MAY BE DISALLOWED VIA CONSOLE SWITCH  
ZERO (0).  
*****
```

```
WRITE: BIT #1,JSWR ;SEE IF SHOULD WRITE  
BEQ WRITE  
JMP WEX ;IF NOT: BR  
MOV RCNT,RO ;RO=RECORD COUNT  
W1: MOV #MSG5,EMADDR ;SET ERROR MSG ADDRESS  
MOV FMCNT,CFC ;LOAD CHAR COUNT  
MOV #DATA,2BA ;SET DATA ADDR  
MOV #60,MTC1 ;SET WRITE OP COMMAND  
MOV #W1,RTRN ;SET RETURN ADDRESS  
JMP TAPC ;GO EXECUTE COMMAND  
W1: BIT #2000,2DS ;SEE IF EOT  
BEQ W2 ;IF NOT AT EOT: BR  
TST EOTREC ;BRANCH IF WRITTEN PAST EOT  
BMI W2  
W2: MOV RO,EOTREC ;SAVE RECORD COUNT  
ADD #100001,EOTREC ;SET EOT INDICATOR & ADD 1 TO # OF  
;RECORDS WRITTEN  
MOV #2,RO ;SET TO WRITE 1 LAST RECORD  
W2: BIT #10000,JSWR ;SEE IF SHOULD CHECK ERRORS  
EINE W3 ;IF NOT: BR  
JSR PC,ERCHK ;GO CHECK ERRORS  
W3: MOV WSTAL,STAL ;SET DELAY  
JSR PC,STALL ;DELAY
```

2281	005700	005737	000714		TST	RTYFL	:SEE IF RETRY TIME
2282	005704	001401			BEQ	W3A	:IF NOT: BR
2283	005706	000207			RTS	PC	:ELSE RETURN
2284	005710	005737	000710	W3A:	TST	SERFL	:SEE IF WRITE ERROR
2285	005714	001450			BEQ	W5	:IF NOT: BR
2286	005716	013704	000676		MOV	UNP R4	
2287	005722	005264	001070		INC	WTER1(R4)	:BUMP WRITE ERROR
2288	005726	005037	000710		CLR	SERFL	:CLEAR STATUS ERROR FLAG
2289	005732	032777	000020	172650	BIT	#20,JSWR	:SEE IF RETRY
2290	005740	001436			BEQ	W5	:IF NOT: BR
2291	005742	013703	000724		MOV	ERSAV,R3	
2292	005746	042703	102700		BIC	#102700,R3	:MASK UNRECOVERABLE ERROR
2293	005752	001410			BEQ	W4	:IF SO: BR
2294	005754	004737	022570		JSR	PC,PAPRT	:PRINT HEADER
2295	005760	012704	026405		MOV	#MSG78,R4	
2296	005764	000004			TYPE		:TYPE MSG
2297	005766	004737	011236		JSR	PC,NRTP	:PRINT ER FOR NON-RETRYABLE
2298	005772	000421			BR	W5	
2299	005774	013704	000676	W4:	MOV	UNP R4	
2300	006000	005264	001050		INC	RTY1(R4)	:BUMP RETRY CNTR
2301	006004	032777	002000	172576	BIT	#2000,JSWR	:SEE IF PRINT ERRORS
2302	006012	001003			BNE	W4A	:IF NOT: BR
2303	006014	012704	026121		MOV	#MSG64,R4	
2304	006020	000004			TYPE		:TYPE MSG
2305	006022	005037	000704	W4A:	CLR	RTCNT	:CLEAR RETRY NUMBER
2306	006026	005037	000702		CLR	RPCNT	:CLEAR REPEAT COUNTER
2307	006032	004737	006370		JSR	PC,WRTY	:GO RETRY WRITE ERROR
2308	006036	005037	000714	W5:	CLR	RTYFL	:CLEAR RETRY COUNTER
2309	006042	005300			DEC	RD	:SEE IF DONE ALL
2310	006044	001243			BNE	W0	:IF NOT: BR
2311	006046	005737	000564	W6:	TST	TMEX	:SEE IF TM
2312	006052	001525			BEQ	WEX	:IF NOT: BR
2313	006054	005237	000700		INC	TMFLG	:SET TM FLAG
2314	006060	012737	026026	000654	MOV	#MSG54,EMADDR	:POINT TO TM ERROR MSG
2315	006066	012737	000026	000674	MOV	#26,MTCT	:SET TM OP CODE
2316	006074	012777	000000	172414	MOV	#0,SFC	:LOAD FRAME COUNTER
2317	006102	012777	027326	172404	MOV	#WDATA,2BA	:LOAD BUS ADDRESS
2318	006110	012737	006122	000664	MOV	#WTMO,RTRN	:SAVE RETURN ADDRESS
2319	006116	000137	021054		JMP	TAPG	:WRITE TM
2320	006122	032777	010000	172460	WTMO:	BIT	#10000,JSWR
2321	006130	001076			BNE	WEX	:SEE IF SHOULD CHECK ERRORS
2322	006132	032777	000004	172362	BIT	#4,205	:SEE IF TM STATUS
2323	006140	001011			BNE	WTM1	:IF SO: BR
2324	006142	012737	027326	020774	MOV	#WDATA,CADER	:SET EXPT BUS ADDRESS
2325	006150	012737	000001	021002	MOV	#1,DRVER	:INDICATE ERFOR
2326	006156	004737	020056		JSR	PC,ERPT	:PRINT TM ERROR
2327	006162	000404			BR	WTM2	
2328	006164	012703	027326	WTM1:	MOV	#WDATA,P3	:SET EXPT ADDRESS
2329	006170	004737	017332		JSR	PC,ER2	:GO CHECK FOR OTHER ERRORS
2330	006174	005737	000714	WTM2:	TST	RTYFL	:SEE IF RETRY
2331	006200	001401			BEQ	WTM3	:IF NOT: BR
2332	006202	000207			RTS	PC	:ELSE RETURN TO RETRY ROUTINE
2333	006204	005737	000710	WTM3:	TST	SERFL	:SEE IF WRITE ERROR
2334	006210	001446			BEQ	WEX	:IF NOT: BR
2335	006212	013704	000676		MOV	UNP R4	
2336	006216	005264	001070		INC	WTER1(R4)	:BUMP WRITE ERROR

2337	006222	032777	000020	172360	BIT	#20, @SWR	: SEE IF SHOULD RETRY
2338	006230	001436			BEQ	MEX	: IF NOT: BR
2339	011232	013703	000724		MOV	ERSAV, R3	
2340	011236	042703	102700		BIC	#102700, R3	: MASK UNRECOVERABLE ERROR
2341	006242	001410			BEQ	WTM4	: IF SO: BR
2342	011244	004737	022570		JSR	PC, PAPRT	: PRINT HEADER
2343	011250	012704	026405		MOV	#MSG78, R4	
2344	006254	000004			TYPE		: TYPE MSG
2345	006256	004737	011236		JSR	PC, MRTF	: PRINT ER FOR NON-RETRYABLE
2346	011252	000421			BR	MEX	
2347	011264	005037	000702		WTM4: CLR	RPCNT	: CLEAR REPEAT CNTR
2348	006270	013704	000676		MOV	UNP, R4	
2349	006274	006264	001050		INC	RTY1(R4)	: BUMP RETRY CNTR
2350	006300	005037	000704		CLR	RTCNT	: CLEAR RETRY CNTR
2351	006304	032777	002000	172276	BIT	#2000, @SWR	: SEE IF PRINT ERRORS
2352	006312	001003			BNE	WTM4A	: IF NOT: BR
2353	006314	012704	026121		MOV	#MSG64, R4	
2354	006320	000004			TYPE		: TYPE MSG
2355	006322	004737	006370		WTM4A: JSR	PC, WRTY	: GO DO RETRY
2356	006326	005037	000714		WEX: CLR	RTYFL	: CLEAR RETRY FLAG
2357	006332	005037	000700		CLR	TMFLG	: CLEAR TAPE MARK FLAG
2358	006336	005737	000662		TST	EOTREC	: BRANCH IF NOT AT EOT
2359	006342	100011			BPL	WRWX	
2360	006344	017703	172240		WRW: MOV	@SWR, R3	
2361	006350	042703	177763		BIC	#177763, R3	
2362	006354	022703	000014		CMP	#14, R3	: SEE IF WRITE ONLY
2363	006360	001002			BNE	WRWX	: IF NOT: BR
2364	006362	000137	004330		JMP	REOT	: ELSE REWIND
2365	006366	000207			WRWX: RTS	PC	: EXIT

```

2366                                     :*****
2367                                     :WRITE ERROR RETRY
2368                                     :*****
2369
2370
2371 006370 012737 000001 000714 WRTY:  MOV  #1,RTYFL      ;SET RETRY FLAG
2372 006376 004737 006772 WRTY0: JSR  PC,WRTSB    ;GO SPACE REVERSE FOR REPEAT
2373 006402 005737 000700      TST  TMFLG      ;SEE IF TAPE MARK TIME
2374 006406 001003      BNE  WRTYTM     ;IF SO: BR
2375 006410 004737 005554      JSR  PC,W0      ;REWRITE RECORD
2376 006414 000402      BR   WRTYR     ;GO ON
2377 006416 004737 006060 WRTYTM: JSR PC,WTM   ;GO WRITE TAPE MARK AGAIN
2378 006422 005737 000710 WRTYR:  TST  SEFL   ;REWRITE GOOD
2379 006426 001024      BNE  WRTY2     ;IF NOT: BR
2380 006430 005237 000702      INC  RPCNT     ;BUMP REPEAT COUNTER
2381 006434 022737 000004 000702  CMP  #4,RPCNT ;SEE IF FOUR GOOD REPEATS
2382 006442 001355      BNE  WRTY0     ;IF NOT: REPEAT
2383 006444 032777 002000 172136 BIT  #2000,@SWR ;SEE IF PRINT
2384 006452 001011      BNE  WRTY1     ;IF NOT: BR
2385 006454 012704 026613      MOV  #MSG105,R4
2386 006460 000004      TYPE  ;TYPE MSG
2387 006462 012704 026143      MOV  #MSG65,R4
2388 006466 000004      TYPE  ;TYPE MSG
2389 006470 013703 000704      MOV  RTCNT,R3
2390 006474 104400 WRTY1:  TYPOCT ;PRINT RETRY NUMBER
2391 006476 000207      RTS  PC        ;RESUME TESTING
2392 006500 013703 000724 WRTY2:  MOV  ERSV,R3 ;GET ER
2393 006504 005037 000650      CLR  TEMP3    ;CLEAR RECOVERABLE ERROR INDICATOR
2394 006510 042703 102700      BIC  #102700,R3 ;MASK RECOVERABLE BITS
2395 006514 001413      BEQ  WRTY2A   ;IF RECOVERABLE: BR
2396 006516 004737 022570      JSR  PC,PAPRT ;PRINT HEADER
2397 006522 012704 026405      MOV  #MSG78,R4
2398 006526 000004      TYPE  ;TYPE MSG
2399 006530 004737 011236      JSR  PC,NRTP  ;PRINT ER
2400 006534 012737 000001 000650  MOV  #1,TEMP3 ;SET FLAG
2401 006542 000407      BR   WRTY2B   ;SEE IF PRINT
2402 006544 032777 002000 172036 WRTY2A: BIT  #2000,@SWR ;IF NOT: BR
2403 006552 001025      BNE  WRTY3    ;SEE IF PRINT
2404 006554 012704 027032      MOV  #MSG110,R4
2405 006560 000004      TYPE  ;TYPE MSG
2406 006562 012704 026143 WRTY2B: MOV  #MSG65,R4
2407 006566 000004      TYPE  ;TYPE MSG
2408 006570 013703 000704      MOV  RTCNT,R3 ;PRINT RETRY NUMBER
2409 006574 104400      TYPOCT
2410 006576 012704 027054      MOV  #MSG111,R4
2411 006602 000004      TYPE  ;TYPE MSG
2412 006604 013703 000702      MOV  RPCNT,R3 ;PRINT REPEAT NUMBER
2413 006610 104400      TYPOCT
2414 006612 005737 000650      TST  TEMP3    ;SEE IF DID NON-RECOVERABLE
2415 006616 001403      BEQ  WRTY3    ;IF NOT: BR
2416 006620 005037 000650      CLR  TEMP3    ;CLEAR FLAG
2417 006624 000207      RTS  PC        ;EXIT
2418 006626 005737 000704 WRTY3:  TST  RTCNT   ;SEE IF FIRST RETRY
2419 006632 001004      BNE  WRTY3A   ;IF NOT: BR
2420 006634 013704 000676      MOV  UNP,R4
2421 006640 005364 001070      DEC  WTER1(R4) ;DECREMENT WRITE ERROR CNTR

```

```

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

```



```

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

2535  
2536  
2537  
2538  
2539  
2531  
2532  
2533  
2534  
2535  
2536  
2537  
2538  
2539  
2530  
2531  
2532  
2533  
2534  
2535  
2536  
2537  
2538  
2539  
2530  
2531  
2532  
2533  
2534  
2535  
2536  
2537  
2538  
2539  
2530  
2531  
2532  
2533  
2534  
2535  
2536  
2537  
2538  
2539

007326	012704	025117	
007332	000004		
007334	013704	000676	
007340	016437	001030	000732
007346	017703	171360	
007352	000241		
007354	006003		
007356	104400		
007360	012704	027066	
007364	000004		
007366	005777	171340	
007372	001001		
007374	000207		
007376	000137	007204	
007402	004737	022570	
007406	012704	026714	
007412	000004		
007414	000207		

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

```

BTPRT:  MOV    #MSG28,R4
        TYPE                   ;TYPE MSG
        MOV    UNP,R4
        MOV    BTA00R(R4),BTPT ;SET TABLE POINTER
        MOV    @BTPT,R3
        CLC
        ROR    R3                ;CORRECT NUMBER
        TYPOCT                   ;PRINT NUMBER OF BAD SPOTS
        MOV    #MSG112,R4
        TYPE                   ;TYPE MSG
        TST    @BTPT              ;SEE IF ANY BAD SPOTS
        BNE    BTPRT1            ;IF SO: BR
        RTS    PC                ;ELSE RETURN
BTPRT1: JMP    BTOVO             ;PRINT STATS

```

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

```

BTUR:  JSR    PC,PAPRT           ;PRINT HEADER
        MOV    #MSG107,R4
        TYPE                   ;TYPE MSG
        RTS    PC                ;RESUME TESTING

```

2540  
2541  
2542  
2543  
2544  
2545  
2546  
2547  
2548  
2549  
2550  
2551  
2552  
2553  
2554  
2555  
2556  
2557  
2558  
2559  
2560  
2561  
2562  
2563  
2564  
2565  
2566  
2567  
2568  
2569  
2570  
2571  
2572  
2573  
2574  
2575  
2576  
2577  
2578  
2579  
2580  
2581  
2582  
2583  
2584  
2585  
2586  
2587  
2588  
2589  
2590  
2591  
2592  
2593  
2594  
2595

007416 012737 000002 000562 RSEQ:  
007424 017704 171160  
007430 042704 177763  
007434 001004  
007436 032777 000002 171144  
007444 001050  
007446 032777 000004 171134 RSR:  
007454 001005  
007456 012737 010000 000562  
007464 004737 007730  
007470 032777 000010 171112 RSF:  
007476 001025  
007500 032737 010000 000562  
007506 001406  
007510 013737 000600 000670  
007516 004737 012116  
007522 000406  
007524 032777 000001 171056 RSFO:  
007532 001002  
007534 004737 011672  
007540 012737 000002 000562 RSF1:  
007546 004737 007730  
007552 005737 000662 RSEX:  
007556 100002  
007560 000137 004330  
007564 000207  
1\$:  
007566 012737 010000 000562 RSFR:  
007574 032777 000010 171006  
007602 001013  
007604 032777 000001 170776  
007612 001002  
007614 004737 011672  
007620 012737 000002 000562 RSFR0:  
007626 004737 007730  
007632 032777 000004 170750 RSFR1:  
007640 001344  
007642 032737 010000 000562  
007650 001005  
007652 013737 000600 000670

```
*****  
: 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.  
*****  
: MOV #2, RDCMD  
: MOV @SWR, R4 ; READ SWITCHES  
: BIC #177763, R4 ; MASK READ BITS & SEE IF BOTH READS  
: BNE RSR ; IF NOT: BR  
: BIT #2, @SWR ; SEE IF READ REVERSE FIRST  
: BNE RSFR ; IF NOT: BR  
: BIT #4, @SWR ; SEE IF SHOULD READ REVERSE  
: BNE RSF ; IF NOT: BR  
: MOV #10000, RDCMD ; LOAD READ REVERSE COMMAND  
: JSR PC, READ ; GO READ REVERSE  
: BIT #10, @SWR ; SEE IF SHOULD READ FORWARD  
: BNE RSEX ; IF NOT: BR  
: BIT #10000, RDCMD ; SEE IF HAVE READ REVERSE  
: BEQ RSFO ; IF NOT: BR  
: MOV TSTAL, STAL  
: JSR PC, STALL ; DO READ STALL  
: BR RSF1  
: BIT #1, @SWR ; SEE IF WRITE  
: BNE RSF1 ; IF NOT: BR  
: JSR PC, BKSP ; GO BACKSPACE  
: MOV #2, RDCMD ; LOAD READ FORWARD COMMAND  
: JSR PC, READ ; GO READ  
: TST EOTREC ; BRANCH IF NOT AT EOT  
: BPL 1$  
: JMP REOT ; ELSE GO TO REWIND  
: RTS PC ; EXIT  
: MOV #10000, RDCMD  
: BIT #10, @SWR ; SEE IF SHOULD READ FORWARD  
: BNE RSFR1 ; IF NOT: BR  
: BIT #1, @SWR ; SEE IF WRITE  
: BNE RSFR0 ; IF NOT: BR  
: JSR PC, BKSP ; GO BACKSPACE TO START  
: MOV #2, RDCMD ; LOAD READ FORWARD COMMAND  
: JSR PC, READ ; GO READ FORWARD  
: BIT #4, @SWR ; SEE IF SHOULD READ REVERSE  
: BNE RSEX ; IF NOT: BR  
: BIT #10000, RDCMD ; IF READ REVERSE: BR  
: BNE RSFR2 ; DO READ STALL  
: MOV TSTAL, STAL
```

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

2607  
2608  
2609  
2610  
2611  
2612  
2613  
2614  
2615  
2616  
2617  
2618  
2619  
2620  
2621  
2622  
2623  
2624  
2625  
2626  
2627  
2628  
2629  
2630  
2631  
2632  
2633  
2634  
2635  
2636  
2637  
2638  
2639  
2640  
2641  
2642  
2643  
2644  
2645  
2646  
2647  
2648  
2649  
2650  
2651  
2652  
2653  
2654  
2655  
2656  
2657  
2658  
2659  
2660  
2661  
2662

007730 013700 000554  
007734 005737 000662  
007740 100013  
007742 032737 010000 000562  
007750 001407  
007752 042737 100000 000662  
007760 013703 000662  
007764 160300  
007766 005200  
007770 012737 024475 000654  
007776 005037 000700  
010002 032737 010000 000562  
010010 001406  
010012 005737 000564  
010016 001403  
010020 005237 000700  
010024 005200  
010026 013777 000556 170462  
010034 012777 033334 170452  
010042 032737 010000 000562  
010050 001417  
010052 013703 000556  
010056 005103  
010060 032737 000020 000552  
010066 001402  
010070 000241  
010072 006003  
010074 060377 170414  
010100 012737 000076 000674  
010106 000403

\*\*\*\*\*  
: 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).  
:\*\*\*\*\*

READ: MOV RCNT,R0 ;LOAD REC CNTR  
TST EOTREC ;SEE IF EOT  
BPL RDA ;IF NOT: BR  
BIT #10000,RDCMD ;SEE IF READ FORWARD  
BEQ RDA ;IF SO: BR  
BIC #10000,EOTREC ;CLEAR FLAG  
MOV EOTREC,R3 ;GET MODIFIED RECORD COUNT  
SUB R3,R0 ;SET RECORD AT  
INC R0 ;SET TO PROPER NUMBER OF RECORDS  
RDA: MOV #MSG6,EMADDR ;SET ERROR MSG ADDRESS  
CLR TMFLG  
BIT #10000,RDCMD  
BEQ R0 ;IF READ FORWARD: BR  
TST TMEX ;SEE IF TM  
BEQ R0 ;IF NOT: BR  
INC TMFLG ;SET TM FLAG  
INC R0  
R0: MOV FMCNT,2FC ;LOAD CHAR CNTR  
MOV #RDATA,2BA ;LOAD DATA ADDR  
BIT #10000,RDCMD ;SEE IF READ REVERSE  
BEQ R0A ;IF NOT: BR  
MOV FMCNT,R3  
COM R3  
BIT #20,UDES ;SEE IF CORE DUMP  
BEQ R01 ;IF NOT: BR  
CLC  
R01: ROR R3 ;R3 = FC/2  
ADD R3,2BA ;SET REVERSE BUS ADDRESS  
MOV #76, MTC1 ;SET READ REVERSE  
BR R01B

2663	010110	012737	000070	000674	RD1A:	MOV	#70, MTC1	; SET READ FORWARD
2664	010116	012737	010130	000664	RD1B:	MOV	#RD2, RTRN	; SET INTERRUPT RETURN ADDRESS
2665	010124	000137	021054		RD1D:	JMP	TAPG	; GO EXECUTE TAPE COMMAND
2666	010130	032737	010000	000562	RD2:	BIT	#10000, RDCMD	; SEE IF READ REVERSE
2667	010136	001014				BNE	R03	; IF S0: BR
2668	010140	032777	002000	170354		BIT	#2000, JDS	; SEE IF EOT
2669	010146	001410				BEQ	R03	; IF NOT: BR
2670	010150	005737	000700			TST	TMFLG	; SEE IF TM
2671	010154	001005				BNE	R03	; IF S0: BR
2672	010156	010037	000662			MOV	R0, EOTREC	
2673	010162	052737	100000	000662		BIS	#10000, EOTREC	; SET EOT FLAG
2674	010170	032777	000002	170324	RD3:	BIT	#2, JDS	; SEE IF AT LOAD POINT
2675	010176	001410				BEQ	R04	; IF NOT: BR
2676	010200	004737	022570			JSR	PC, PAPRT	; PRINT CYCLE NUMBER
2677	010204	012704	024676			MOV	#MSG22, R4	
2678	010210	000004				TYPE		; TYPE MSG
2679	010212	000000				HALT		
2680	010214	000137	003160			JMP	STARTA	; RESTART
2681	010220	032777	004000	170362	RD4:	BIT	#4000, JSWR	; SEE IF SHOULD CHECK ERRORS
2682	010226	001121				BNE	R05	; IF NOT: BR
2683	010230	005737	000700			TST	TMFLG	
2684	010234	001472				BEQ	R04B	; IF NO TM EXPT: BR
2685	010236	032777	000004	170256		BIT	#4, JDS	
2686	010244	001024				BNE	R04A	; IF TM RECVD: BR
2687	010246	012737	033334	020774		MOV	#RDATA, CADER	; SAVE EXPT BUS ADDRESS
2688	010254	012737	000002	021002		MOV	#2, DRIVER	; SET TM STATUS ERROR FLAG
2689	010262	004737	020056			JSR	PC, ERPT	; GO PRINT TM ERROR
2690	010266	013704	000676			MOV	UMP, R4	
2691	010272	032737	010000	000562		BIT	#10000, RDCMD	; SEE IF READ REVERSE
2692	010300	001403				BEQ	1\$	; IF NOT: BR
2693	010302	005264	001150			INC	RDERR1(R4)	; BUMP READ REVERSE ERROR
2694	010306	000502				BR	R06	
2695	010310	005264	001110		1\$:	INC	RDERR1(R4)	; BUMP READ FORWARD ERROR
2696	010314	000477				BR	R06	
2697	010316	012703	033334		RD4A:	MOV	#RDATA, R3	
2698	010322	032737	010000	000562		BIT	#10000, RDCMD	; SEE IF READ REVERSE
2699	010330	001007				BNE	R04A0	; IF S0: BR
2700	010332	032737	002000	000552		BIT	#2000, UDES	; SEE IF IN PE
2701	010340	001025				BNE	R04A2	; IF S0: BR
2702	010342	052703	000002			ADD	#2, R3	
2703	010346	000422				BR	R04A2	
2704	010350	013704	000556		RD4A0:	MOV	FMCNT, R4	
2705	010354	005104				COM	R4	
2706	010356	032737	000020	000552		BIT	#20, UDES	; SEE IF CORE DUMP
2707	010364	001402				BEQ	R04A1	; IF NOT: BR
2708	010366	000241				CLC		
2709	010370	006004				ROR	R4	; SET TO FC/2
2710	010372	060403			RD4A1:	ADD	R4, R3	; SET EXPT BUS ADDRESS
2711	010374	042703	000001			BIC	#1, R3	; MAKE EXPT ADDRESS EVEN
2712	010400	032737	002000	000552		BIT	#2000, UDES	; SEE IF IN PE
2713	010406	001002				BNE	R04A2	; IF S0: BR
2714	010410	162703	000002			SUB	#2, R3	
2715	010414	004737	017332		RD4A2:	JSR	PC, ER2	
2716	010420	000402				BR	R04C	
2717	010422	004737	017236		RD4B:	JSR	PC, ERCHK	; GO CHECK ERRORS
2718	010426	005737	000710		RD4C:	TST	SERFL	

# K05

TMD3/TE16 DATA RELIABILITY PROGRAM  
 DZTEDA.P11 07-APR-77 13:36

MACY11 27(1006) 07-APR-77 13:36 PAGE 61

2719	010432	001417				BEQ	R05		; IF NO ERROR: BR
2720	010434	013704	000676			MOV	UNP, R4		
2721	010440	032737	010000	000562		BIT	#10000, R0CMD		; SEE IF READ REVERSE
2722	010446	001003				BNE	R04D		; IF SO: BR
2723	010450	005264	001110			INC	R0ERR1(R4)		; BUMP READ FORWARD ERROR
2724	010454	000402				BR	R04E		
2725	010456	005264	001150		RD4D:	INC	R0ERR1(R4)		; BUMP READ REVERSE ERROR
2726	010462	004737	010664		RD4E:	JSR	PC, R0RTY		; GO RETRY
2727	010466	005037	000714			CLR	R1YFL		; CLEAR RETRY FLAG
2728	010472	032777	020000	170110	RD5:	BIT	#20000, #SWR		; SEE IF SHOULD DO DATA CHECK
2729	010500	001005				BNE	R06		; IF NOT; BR
2730	010502	005737	000700			TST	TMFLG		
2731	010506	001002				BNE	R06		
2732	010510	004737	015366			JSR	PC, DCHK		; GO CHECK DATA
2733	010514	005037	000710		RD6:	CLR	SRFL		; CLEAR STATUS ERROR FLAG
2734	010520	004737	014232			JSR	PC, D53		; CLEAR BUFFER
2735	010524	032777	000040	170056		BIT	#40, #SWR		; SEE IF SHOULD YOZZLE
2736	010532	001402				BEQ	R07		; IF NOT: BR
2737	010534	004737	011252			JSR	PC, YOZ		; ELSE GO YOZZLE
2738	010540	013737	000574	000670	RD7:	MOV	R0STAL, STAL		; SET DELAY
2739	010546	004737	012116			JSR	PC, STALL		; STALL
2740	010552	032737	010000	000562		BIT	#10000, R0CMD		; SEE IF READ REVERSE
2741	010560	001403				BEQ	R07A		; IF NOT: BR
2742	010562	005037	000700			CLR	TMFLG		; CLEAR TAPE MARK FLAG
2743	010566	000405				BR	R010		
2744	010570	005737	000662		RD7A:	TST	E0TREC		; SEE IF EOT FOUND
2745	010574	100002				BPL	R010		; IF NOT: BR
2746	010576	012700	000001			MOV	#1, R0		; SET TO EOT
2747	010602	005300			RD10:	DEC	R0		
2748	010604	001402				BEQ	R011		; IF DONE ALL: BR
2749	010606	000137	010026			JMP	R00		
2750	010612	032737	010000	000562	RD11:	BIT	#10000, R0CMD		; SEE IF READ REVERSE
2751	010620	001016				BNE	R0EX		; IF SO: BR
2752	010622	005737	000662			TST	E0TREC		; SEE IF FOUND EOT
2753	010626	100413				BMI	R0EX		; IF SO: BR
2754	010630	005737	000564			TST	TMEX		; SEE IF TM EXPECTED
2755	010634	001410				BEQ	R0EX		; IF NOT: BR
2756	010636	005737	000700			TST	TMFLG		; SEE IF TM FOUND
2757	010642	001005				BNE	R0EX		; IF SO: BR
2758	010644	005237	000700			INC	TMFLG		; ELSE SET FLAG
2759	010650	005200				INC	R0		; SET RECORD COUNT TO ONE
2760	010652	000137	010026			JMP	R00		; GO READ TM
2761	010656	005037	000700		RDEX:	CLR	TMFLG		
2762	010662	000207			RDX:	RTS	PC		; EXIT

```

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

```



2819	011122	000404				BR	RDR75B	
2820	011124	032777	002000	167456	RDR75A:	BIT	#2000,JSWR	;SEE IF PRINT INHIBITED
2821	011132	001014				BNE	RDR76	;IF SO: BR
2822	011134	012704	026143		RDR75B:	MOV	#MSG65,R4	
2823	011140	000004				TYPE		;TYPE MSG
2824	011142	013703	000704			MOV	RTCNT,R3	
2825	011146	104400				TYPOCT		;PRINT RETRY NUMBER
2826	011150	005737	000650			TST	TEMP3	;SEE IF DID NON-RECOVERABLE
2827	011154	001403				BEQ	RDR76	;IF NOT: BR
2828	011156	005037	000650			CLR	TEMP3	;CLEAR FLAG
2829	011162	000207				RTS	PC	;EXIT
2830	011164	005237	000704		RDR76:	INC	RTCNT	
2831	011170	023737	000704	000604		CHP	RTCNT,RETRY	;SEE IF DONE 8 RETRIES
2832	011176	001265				BNE	RDR7G	;IF NOT: BR
2833	011200	012704	027131			MOV	#MSG115,R4	
2834	011204	000004				TYPE		;TYPE MSG
2835	011206	013704	000676			MOV	UNP,R4	
2836	011212	032737	010000	000562		BIT	#10000,RDCMD	;SEE IF READ REVERSE
2837	011220	001003				BNE	RDR77	;IF SO: BR
2838	011222	005264	002730			INC	RFHARD(R4)	;BUMP FORWARD HARD ERROR CNTR
2839	011226	000402				BR	RDR7X	
2840	011230	005264	002750		RDR77:	INC	RFHARD(R4)	;BUMP REVERSE HARD ERROR CNTR
2841	011234	000207			RDR7X:	RTS	PC	;RETURN
2842								
2843	011236	013703	000724		NRTP:	MOV	ERSAV,R3	;GET ER REGISTER
2844	011242	104400				TYPOCT		;PRINT ER
2845	011244	004737	021020			JSR	PC,FRPRT	;PRINT F OR R
2846	011250	000207				RTS	PC	;RETURN

```

2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860 011252 013737 000602 000670 YOZ:
2861 011260 004737 012116 JSR
2862 011264 012777 177777 167224 YOZO: MOV
2863 011272 032737 010000 000562 BIT
2864 011300 001404 BEQ
2865 011302 112737 000030 000674 MOV
2866 011310 000403 BR
2867 011312 112737 000032 000674 YOZA: MOV
2868 011320 012737 011340 000664 YOZB: MOV
2869 011326 012737 177775 000670 MOV
2870 011334 000137 021054 JMP
2871 011340 005737 000700 YOZC: TST
2872 011344 001404 BEQ
2873 011346 012737 040000 000670 MOV
2874 011354 000403 BR
2875 011356 013737 000602 000670 15: MOV
2876 011364 004737 012116 25: JSR
2877 011370 012777 033334 167116 MOV
2878 011376 032737 010000 000562 BIT
2879 011404 001416 BEQ
2880 011406 013703 000556 MOV
2881 011412 005103 COM
2882 011414 032737 000020 000552 BIT
2883 011422 001401 BEQ
2884 011424 006203 ASR
2885 011426 060377 167062 YOZCO: ADD
2886 011432 012737 000076 000674 MOV
2887 011440 000403 BR
2888 011442 012737 000070 000674 YOZC1: MOV
2889 011450 013777 000556 167040 YOZC2: MOV
2890 011456 012737 011470 000664 MOV
2891 011464 000137 021054 JMP
2892 011470 032777 004000 167112 YOZD: BIT
2893 011476 001050 BNE
2894 011500 005737 000700 TST
2895 011504 001443 BEQ
2896 011506 032737 010000 000562 BIT
2897 011514 001425 BEQ
2898 011516 012703 033334 MOV
2899 011522 013704 000556 MOV
2900 011526 005104 COM
2901 011530 032737 000020 000552 BIT
2902 011536 001401 BEQ

```

```

*****
:YOZZLE SUBROUTINE:

```

```

:THIS SUBROUTINE, ENTERED VIA SWITCH FIVE (5), IS USED TO PERFORM
:A CONTINUOUS READ AND SPACE OVER OF THE CURRENT RECORD ON TAPE.
:FULL STATUS AND DATA CHECKING MAY BE PERFORMED
:OR NOT VIA CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13).
:A SOFTWARE DELAY IS PERFORMED BETWEEN EACH READ
:AND SPACE OPERATION AND MAY BE VARIED BY TYPING
:CNTRL C ON THE TTY AND ENTERING A VALUE IN RESPONSE
:TO THE PRINTED REQUEST.

```

```

*****

```

```

:DO YOZZLE STALL
:SET TO 1 RECORD SPACING
:SEE IF READ REVERSE
:IF NOT: BR
:SET TO SPACE FORWARD
:SET TO SPACE REVERSE
:SET RETURN ADDRESS
:SET TIME MULTIPLIER
:GO YOZZLE
:SEE IF TM
:IF NOT: BR
:SET TM STALL
:DO YOZZLE STALL
:SET BUS ADDRESS
:SEE IF READ REVERSE
:IF NOT: BR
:SEE IF CORE DUMP
:IF NOT: BR
:R3 = FC/2
:SET REVERSE BUS ADDRESS
:SET READ REVERSE
:SET READ FORWARD
:SET CHARACTER COUNT
:SET RETURN ADDRESS
:GO READ
:SEE IF SHOULD CHECK ERRORS
:IF NOT: BR
:SEE IF TAPE MARK TIME
:IF NOT: BR
:SEE IF READ REVERSE
:IF NOT: BR
:SEE IF CORE DUMP
:IF NOT: BR

```

2903	011540	006204			ASR	R4		: SET TO FC/2
2904	011542	060403			ROO	R4, R3		: SET EXPT BUS ADDRESS
2905	011544	042703	000001		BIC	R1, R3		: MAKE EXPT ADDRESS EVEN
2906	011550	032737	002000	000552	BIT	R2000, UDES		: SEE IF PE
2907	011556	001001			BNE	YOZD2		: IF SO: BR
2908	011560	005743			TST	-(R3)		: SET EXPT BA
2909	011562	004737	017332		JSR	PC, ER2		: GO CHECK ERRORS
2910	011567	000430			BR	YOZF		
2911	011570	012703	033334		MOV	R0DATA, R3		
2912	011574	032737	002000	000552	BIT	R2000, UDES		: SEE IF PE
2913	011602	001001			BNE	YOZD3		: IF SO: BR
2914	011604	005723			TST	(R3)+		: SET EXPT BA
2915	011606	004737	017332		JSR	PC, ER2		: GO CHECK ERRORS
2916	011612	000416			BR	YOZF		
2917	011614	004737	017236		JSR	PC, ERCHK		: ELSE GO CHECK ERRORS
2918	011620	005737	000714		YOZE:	TST		: SEE IF RETRY
2919	011624	001013			BNE	YOZG		: IF SO: BR
2920	011626	032777	020000	166754	BIT	R2000, JSWR		: SEE IF SHOULD CHECK DATA
2921	011634	001005			BNE	YOZF		: IF NOT: BR
2922	011636	005737	000700		TST	TMFLG		: SEE IF TAPE MARK
2923	011642	001002			BNE	YOZF		: IF SO: BR
2924	011644	004737	015366		JSR	PC, DCHK		: ELSE GO CHECK DATA
2925	011650	004737	014232		YOZF:	JSR		: GO CLEAR DATA AREA
2926	011654	032777	000040	166726	YOZG:	BIT		: SEE IF SHOULD CONTINUE YOZZLE
2927	011662	001402			BEQ	YOZH		: IF NOT: BR
2928	011664	000137	011264		JMP	YOZO		
2929	011670	000207			YOZH:	RTS		: EXIT
2930								

2931  
2932  
2933  
2934  
2935  
2936  
2937  
2938  
2939  
2940  
2941  
2942  
2943  
2944  
2945  
2946  
2947  
2948  
2949  
2950  
2951  
2952  
2953  
2954  
2955  
2956  
2957  
2958  
2959  
2960  
2961  
2962  
2963  
2964  
2965  
2966  
2967  
2968  
2969  
2970  
2971  
2972  
2973  
2974  
2975  
2976  
2977  
2978  
2979  
2980  
2981

011672 013737 000600 000670  
011700 004737 012116  
011704 012737 024525 000654  
011712 012777 033334 166574  
011720 005737 000564  
011724 001440  
011726 012777 177777 166562  
011734 012737 000032 000674  
011742 012737 011754 000664  
011750 000137 021054  
011754 032777 010000 166626  
011762 001021  
011764 012737 026035 000654  
011772 032777 000004 166522  
012000 001006  
012002 012737 033334 020774  
012010 004737 020056  
012014 000404  
012016 012703 033334  
012022 004737 017332  
012026 013700 000554  
012032 005400  
012034 012737 024525 000654  
012042 010077 166450  
012046 012737 000032 000674  
012054 012737 012072 000664  
012062 010037 000670  
012066 000137 021054  
012072 012703 033334  
012076 004737 017332  
012102 013737 000600 000670  
012110 004737 012116  
012114 000207

\*\*\*\*\*  
: 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.  
\*\*\*\*\*

BKSP: MOV TSTAL, STAL  
JSR PC, STALL ; DO TURN AROUND STALL  
MOV #MSG10, EMADDR  
MOV #RDATA, @BA  
TST TMEX ; SEE IF TM  
BEQ BO ; IF NOT: BR  
MOV #-1, @FC  
MOV #32, MTC1  
MOV #BKTM, RTN  
JMP TAPG ; SPACE TO TM  
BIT #10000, @SWR ; SEE IF SHOULD CHECK ERROR  
BNE BO ; IF NOT: BR  
MOV #MSG55, EMADDR  
BIT #4, @OS ; SEE IF TM  
BNE BKTMO ; IF SO: BR  
MOV #RDATA, CADER  
JSR PC, ERPT ; PRINT ERROR  
BR BO  
BKTMO: MOV #RDATA, R3  
JSR PC, FR2  
BO: MOV RONT, RO  
NEG RO ; BUILD SPACE AMOUNT  
MOV #MSG10, EMADDR ; SET ERROR MSG ADDRESS  
MOV RO, @FC  
BKRT: MOV #32, MTC1 ; SET SPACE REVERSE  
MOV #B1, RTN ; SET RETURN ADDRESS  
MOV RO, STAL ; SET INTERRUPT TIME MULTIPLIER  
JMP TAPG ; GO DO SPACE  
B1: MOV #RDATA, R3  
JSR PC, ER2  
B2: MOV TSTAL, STAL ; DO STALL  
JSR PC, STALL ; STALL  
RTS PC ; EXIT

2982  
2983  
2984  
2985  
2986  
2987  
2988  
2989  
2990  
2991  
2992  
2993  
2994  
2995  
2996  
2997  
2998  
2999  
3000  
3001  
3002  
3003

012116 005337 000670  
012122 001375  
012124 000207

STALL: DEC STAL  
BNE STAL ;DELAY  
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(B) 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.  
*****
```

3004  
3005  
3006  
3007  
3008  
3009  
3010  
3011  
3012  
3013  
3014  
3015  
3016  
3017  
3018  
3019  
3020  
3021  
3022  
3023  
3024  
3025  
3026  
3027  
3028  
3029  
3030  
3031  
3032  
3033  
3034  
3035  
3036  
3037  
3038  
3039  
3040  
3041

012126	012701	177760	
012132	012702	175000	
012136	004737	023152	
012142	042737	000001	000630
012150	013737	000630	000556
012156	012737	177777	014270
012164	000207		
012166	012702	000001	
012172	012701	000500	
012176	004737	023152	
012202	013737	000630	000554
012210	000207		

```

CCNTR:  MOV    #20,R1          ;SET HIGH LIMIT
        MOV    #3000,R2      ;SET LOW LIMIT
        JSR    PC,RANG       ;GO GENERATE NUMBER
        BIC    #1,RANSV      ;
        MOV    RANSV,FMcnt    ;SET CHAR COUNT
        MOV    #-1,PATs      ;PRESET DATA PATTERN
        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.
*****

```

```

*****
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.
*****

```

```

RCNTR:  MOV    #1,R2          ;SET LOW LIMIT
        MOV    #500,R1       ;SET HIGH LIMIT
        JSR    PC,RANG       ;GO GENERATE NUMBER
        MOV    RANSV,RCNT    ;SET RECORD COUNT
        RTS    PC            ;EXIT

```

3042  
3043  
3044  
3045  
3046  
3047  
3048  
3049  
3050  
3051  
3052  
3053  
3054  
3055  
3056  
3057  
3058  
3059  
3060  
3061  
3062  
3063  
3064  
3065  
3066  
3067  
3068  
3069  
3070  
3071  
3072  
3073  
3074  
3075  
3076  
3077  
3078  
3079  
3080  
3081  
3082  
3083  
3084  
3085  
3086  
3087  
3088  
3089  
3090  
3091  
3092  
3093  
3094  
3095  
3096  
3097

```

*****
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(B).
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 WRITING 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.
*****

```

```

012212 005737 000636
012216 001002
012220 000137 013700
012224 005037 000676
012230 005037 005054
012234 012700 000010
012240 012701 000746
012244 005021
012246 005300
012250 001375

```

```

TINP: TST TINP
      BNE IS ;SEE IF SHOULD INPUT FROM TTY
      JMP TINP4 ;IF SO: BR
      CLR UNP ;GET SWITCHES
      CLR REOTC ;CLEAR TABLE POINTER
      MOV #10,RO ;CLEAR EOT UNIT COUNTER
      MOV #UN1,R1 ;SET SIZE OF TABLE
      CLR (R1)+ ;SET START OF TABLE
      DEC RO ;CLEAR TABLE
      BNE 35 ;SEE IF DONE
          ;IF NOT: BR

```

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



3154	012556	012704	026262		MOV	#MSG71,R4	
3155	012562	000004			TYPE		;TYPE MSG
3156	012564	013704	000510		MOV	C1,R4	
3157	012570	005204			INC	R4	
3158	012572	152714	000100		BISB	#100,(R4)	;CLEAR TRE
3159	012576	000736			BR	65	;RETRY DVN
3160	012600	012704	025323	TINPO:	MOV	#MSG32,R4	
3161	012604	000004			TYPE		;TYPE MSG
3162	012606	005037	000646		CLR	TEMP2	;CLEAR BUFFER
3163	012612	012705	000646		MOV	#TEMP2,R5	;SET UNIT DESCRIPTION BUFFER ADDRESS
3164	012616	012701	000002		MOV	#2,R1	;SET NUMBER OF CHARACTERS TO INPUT
3165	012622	012702	000007		MOV	#7,R2	;SET MAXIMUM LIMIT
3166	012626	012703	000000		MOV	#0,R3	;SET MINIMUM LIMIT
3167	012632	004737	023340		JSR	PC,TTR	;GO GET UNIT NUMBER
3168	012636	005737	000644		TST	TEMP1	;SEE IF HAVE NEW PARAMETER
3169	012642	001012			BNE	TINPOB	;IF SO: BR
3170	012644	005737	000676		TST	UNP	;SEE IF FIRST ENTRY
3171	012650	001753			BEQ	TINPO	
3172	012652	013700	000676		MOV	UNP,R0	
3173	012656	012760	177777	000746	MOV	#-1,UNI(R0)	;SET END UNIT TABLE
3174	012664	000137	013264		JMP	TINP2C	;GO GET RECORD COUNT
3175	012670	013700	000676		TINPOB:	MOV	UNP,R0
3176	012674	042760	000007	000746	BIC	#7,UNI(R0)	;CLEAR UNIT NUMBER
3177	012702	004737	014074		JSR	PC,TPOS1	;GO LOAD UNIT NUMBER TO PROPER POSITION
3178	012706	012777	000740	165604	MOV	#40,ACS	
3179	012714	013777	000550	165576	MOV	DVN,ACS	
3180	012722	016077	000746	165612	MOV	UNI(R0),ATC	;LOAD UNIT NUMBER
3181	012730	032777	002000	165600	TINPOC:	BIT	#2000,AT
3182	012736	001004			BNE	TINPOD	;SEE IF SLAVE PRESENT
3183	012740	012704	026050		MOV	#MSG57,R4	;IF SO: BR
3184	012744	000004			TYPE		;TYPE MSG
3185	012746	000714			BR	TINPO	;REDO
3186	012750	017703	165562		TINPOD:	MOV	AT,R3
3187	012754	042703	000007		BIC	#7,R3	;GET CONTENTS OF DT REG
3188	012760	022703	142050		CMF	#142050,R3	;CLEAR DRIVE TYPE #
3189	012764	001410			BEQ	TINPOE	;SEE IF 9TRK TM03,TE16
3190	012766	012704	025745		MOV	#MSG50,R4	;IF SO: BR
3191	012772	000004			TYPE		;ILLEGAL DRIVE TYPE
3192	012774	017703	165536		MOV	AT,R3	;TYPE MSG
3193	013000	042703	000007		BIC	#7,R3	;CLEAR SLAVE #
3194	013004	104400			TYPOCT		;PRINT DRIVE TYPE REGISTER
3195	013006	012704	024517		TINPOE:	MOV	#MSG9,R4
3196	013012	000004			TYPE		;TYPE MSG
3197	013014	017703	165520		MOV	ASN,R3	
3198	013020	004737	024322		JSR	PC,SNPT	;PRINT SERIAL NUMBER
3199	013024	012704	025344		TINP1:	MOV	#MSG33,R4
3200	013030	000004			TYPE		;TYPE MSG
3201	013032	005037	000646		CLR	TEMP2	;CLEAR BUFFER
3202	013036	012701	000002		MOV	#2,R1	;SET NUMBER OF CHARACTERS TO INPUT
3203	013042	012702	000004		MOV	#4,R2	;SET MAXIMUM LIMIT
3204	013046	012703	000003		MOV	#3,R3	;SET MINIMUM LIMIT
3205	013052	004737	023340		JSR	PC,TTR	;GO GET DENSITY
3206	013056	005737	000644		TST	TEMP1	;SEE IF HAVE NEW PARAMETER
3207	013062	001407			BEQ	TINP2	;IF NOT: BR
3208	013064	042737	003400	000552	BIC	#3400,UDES	;ELSE CLEAR OLD PARAMETER
3209	013072	012703	000010		MOV	#10,R3	;SET POSITION FACTOR

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

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

```

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

```

3357  
3358  
3359  
3360  
3361  
3362  
3363  
3364  
3365  
3366  
3367  
3368  
3369  
3370  
3371  
3372  
3373  
3374  
3375  
3376  
3377  
3378  
3379  
3380  
3381  
3382  
3383  
3384  
3385  
3386  
3387  
3388  
3389  
3390  
3391  
3392  
3393  
3394  
3395  
3396  
3397  
3398  
3399  
3400  
3401  
3402  
3403  
3404  
3405  
3406  
3407  
3408  
3409  
3410  
3411

014110 005737 015022  
014114 001044  
014116 005737 000736  
014122 001406  
014124 005737 000560  
014130 100003  
014132 004737 014760  
014136 000207  
014140 023737 000560 014270  
014146 001014  
014150 013703 000552  
014154 042703 177767  
014160 023703 014272  
014164 001404  
014166 010337 014272  
014172 004737 015024  
014176 000207  
014200 012703 027326  
014204 013701 000560  
014210 010137 014270  
014214 062701 000001  
014220 006301  
014222 004771 002770  
014226 004737 015024  
014232 013702 000556  
014236 006202  
014240 012701 033334  
014244 005021  
014246 005202  
014250 001375  
014252 013737 000552 014272  
014260 042737 177767 014272  
014266 000207  
014270 177777  
014272 000000

DSUP: TST ROFL  
BNE DS2A  
DSO: TST ASEQF  
BEQ DSOC  
TST PATRN  
BPL DSOC  
JSR PC,DATR  
RTS PC  
DSOC: CMP PATRN,PATS  
BNE DSOA  
MOV UDES,R3  
BIC #177767,R3  
CMP PARS,R3  
BEQ DS08  
MOV R3,PARS  
JSR PC,CRCLRC  
PC  
DS08: RTS PC  
DSOA: MOV #DATA,R3  
MOV PATRN,R1  
MOV R1,PATS  
ADD #1,R1  
ASL R1  
JSR PC,@DATBL(R1)  
DS2A: JSR PC,CRCLRC  
DS3: MOV FMCNT,R2  
ASR R2  
MOV #RDATA,R1  
DS4: CLR (R1)+  
INC R2  
BNE DS4  
MOV UDES,PARS  
BIC #177767,PARS  
RTS PC  
PATS: -1  
PARS: 0

```
*****  
: DATA SETUP ROUTINE:  
: THIS ROUTINE IS USED TO GENERATE INTO THE ENTIRE  
: WRITE BUFFER (4000 OCTAL CHARACTERS) THE DATA PATTERN  
: SELECTED BY THE OPERATOR. THERE ARE 15 (8) FIXED  
: DATA PATTERNS AVAILABLE AND ONE SELECTION (DATA PATTERN 0)  
: WHICH WILL READ ANY PATTERN PRESENTED AT THE  
: HIGH SPEED PAPER TAPE READER. THIS TAPE MUST BE PREPARED  
: BY USING THE PROGRAM CALLED DTC. (MAINDEC-11-DZTUF-A-D)  
: RANDOM DATA MAY ALSO BE USED VIA CONSOLE  
: SWITCH EIGHT (8).  
: THIS ROUTINE IS ALSO USED TO CLEAR OUT THE  
: READ BUFFER (4000 OCTAL CHARACTERS) BEFORE EACH  
: RECORD IS READ.  
*****  
: SEE IF DID RANDOM DATA  
: IF NOT: BR  
: SEE IF AUTO SEQ  
: IF NOT: BR  
: SEE IF AUTO RANDOM  
: IF NOT: BR  
: ELSE GO GENERATE RANDOM DATA  
: RETURN  
: SEE IF NEW PATTERN  
: IF SO: BR  
: GET UNIT DESCRIPTION  
: MASK EVEN PARITY  
: SEE IF SAME AS LAST TIME  
: IF SO: BR  
: SAVE PARITY  
: GO GENERATE EXPT CRC/LRC  
: R3 = ADDRS OF WRITE BUFFER  
: R1 = PATTERN SELECTOR  
: BUMP POINTER  
: MAKE PATTERN SELECTOR EVEN  
: GO GENERATE PATTERN  
: GO GENERATE EXPT CRC/LRC  
: R2=BUFFER SIZE  
: R2=FRAME CMT/2  
: R1=READ DATA START  
: CLEAR BUFFER  
: SEE IF DONE ALL  
: IF NOT: BR  
: GET UNIT DESCRIPTION  
: MASK PARITY  
: EXIT  
: PATTERN NUMBER SAVE
```

# M06

TM03/TE16 DATA RELIABILITY PROGRAM  
DZTEDA.P11 07-APR-77 13:36

MACY11 27(1006) 07-APR-77 13:36 PAGE 76

```

3412
3413
3414
3415 014274 005737 014432
3416 014300 001352
3417 014302 012737 000001 014432
3418 014310 005077 164306
3419 014314 005037 000644
3420 014320 052777 000001 164274
3421 014326 105777 164270
3422 014332 100375
3423 014334 005001
3424 014336 117701 164262
3425 014342 005737 000644
3426 014346 001011
3427 014350 105701
3428 014352 001762
3429 014354 012737 000001 000644
3430 014362 010137 000646
3431 014366 010102
3432 014370 000753
3433 014372 110123
3434 014374 005302
3435 014376 001350
3436 014400 012701 027326
3437 014404 013702 000646
3438 014410 112123
3439 014412 022703 033334
3440 014416 003002
3441 014420 000137 014226
3442 014424 005302
3443 014426 001370
3444 014430 000763
3445 014432 000000
3446

```

;EXTERNAL DATA INPUT FROM H/S READER (256 CHARACTER MAXIMUM)

```

DATO:  TST      DOFL      ;SEE IF SHOULD DO EXTERNAL INPUT
       BNE     DS2A      ;IF NOT: BR
       MOV     #1,DOFL   ;SET EXTERNAL FLAG
       CLR     @PRS      ;CLEAR READER STATUS
       CLR     TEMP1     ;CLEAR FOR USE AS CHARACTER FLAG
DATOA:  BIS     #1,@PRS   ;START READER
DATOB:  TSTB   @PRS      ;SEE IF DONE
       BPL     DATOB     ;IF NOT: BR
       CLR     R1        ;CLEAR SAVE LOCATION
       MOVB   @PRB,R1    ;SAVE CHARACTER
       TST    TEMP1     ;SEE IF HAVE FOUND START CHARACTER
       BNE    DATOC     ;IF SO: BR
       TSTB   R1        ;SEE IF CHARACTER IS 0
       BEQ    DATOA     ;IF SO: BR
       MOV    #1,TEMP1   ;ELSE SET CHARACTER FOUND FLAG
       MOV    R1,TEMP2   ;SAVE DATA SIZE
       MOV    R1,R2      ;SAVE DATA SIZE
       BR     DATOA     ;GO GET FIRST DATA CHAR
DATOC:  MOVB   R1,(R3)+  ;LOAD BUFFER
       DEC    R2        ;SEE IF READ ALL
       BNE   DATOA     ;IF NOT: BR
       MOV   @DATA,R1   ;R1 = START OF WRITE BUFFER
       MOV   TEMP2,R2   ;R2 = SIZE OF DATA FIELD
DATOE:  MOVB   (R1)+(R3)+;REPEAT LOAD OF DATA FIELD
       CMP   @DATA,R3   ;SEE IF DONE
       BGT   DATOF     ;IF NOT: BR
       JMP   DS2A      ;EXIT
DATOF:  DEC    R2       ;SEE IF AT END OF DATA FIELD
       BNE   DATOE     ;IF NOT: BR
       BR   DATOD     ;ELSE RESTART FILL
DOFL:   0              ;EXTERNAL DATA FLAG=1 IF ALREADY DONE

```



```

3530 014556 012702 002002
3531 014557 012701 000001
3532 014558 000241
3533 014559 012713 177400
3534 014574 050123
3535 014576 106101
3536 014600 005302
3537 014602 001372
3538 014604 000207
3539
3540
3541 014606 005001
3542 014610 012702 004004
3543 014614 110123
3544 014616 105201
3545 014620 005302
3546 014622 001374
3547 014624 000207
3548
3549
3550
3551
3552
3553
3554 014626 012701 000377
3555 014632 012702 004004
3556 014636 110123
3557 014640 105301
3558 014642 005302
3559 014644 001374
3560 014646 000207
3561
3562
3563
3564 014650 012701 000377
3565 014654 000137 014440
3566
3567
3568
3569 014660 012702 002002
3570 014664 012701 000376
3571 014670 000261
3572 014672 010113
3573 014674 042723 177400
3574 014700 106101
3575 014702 005302
3576 014704 001372
3577 014706 000207
3578

```

;WALKING ONE/ALL ONE IN ALTERNATING CHARS\*\*\*\*

```

DAT10: MOV #2002,R2 ;SET BUFFER SIZE
        MOV #1,R1 ;SET WALK BASE
        CLC
IS: MOV #177400,(R3) ;LOAD ALL ONE BYTE
     BIS R1,(R3)+ ;LOAD WALK BYTE
     ROLB R1 ;WALK ONE
     DEC R2
     BNE IS ;DO FULL BUFFER
     RTS PC

```

;ALL BITS 0-377\*\*\*\*\*

```

DAT11: CLR R1 ;R1=STARTING DATA
        MOV #4004,R2 ;R2=CHARACTER COUNT+4
IS: MOVB R1,(R3)+ ;LOAD BUFFER
     INCB R1 ;BUMP DATA
     DEC R2 ;SEE IF DONE
     BNE IS ;IF NOT: BR
     RTS PC ;RETURN

```

;ALL BITS 377-0\*\*\*\*\*

```

DAT12: MOV #377,R1 ;R1=STARTING DATA
        MOV #4004,R2 ;R2=CHARACTER COUNT+4
IS: MOVB R1,(R3)+ ;LOAD BUFFER
     DECB R1 ;BUMP DATA
     DEC R2 ;SEE IF DONE
     BNE IS ;IF NOT: BR
     RTS PC ;RETURN

```

;ALTERNATING CHARACTERS 0 AND 377\*\*\*\*\*

```

DAT13: MOV #377,R1 ;R1 = DATA
        JMP DAT1A ;LOAD BUFFER

```

;WALKING ZERO/ALL ZERO IN ALTERNATING CHARS\*\*\*\*\*

```

DAT14: MOV #2002,R2 ;SET BUFFER SIZE
        MOV #376,R1 ;SET WALK BASE
        SEC
IS: MOV R1,(R3) ;LOAD WALK BYTE
     BIC #177400,(R3)+ ;CLEAR HIGH BYTE
     ROLB R1 ;WALK ZERO BIT
     DEC R2
     BNE IS ;FILL BUFFER
     RTS PC ;RETURN

```



```

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

```

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

```



```

3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683 015366 005037 000660
3684 015372 005037 000706
3685 015376 013705 000556
3686 015402 032737 000020 000552
3687 015410 001401
3688 015412 006205
3689 015414 012701 027326
3690 015420 012702 033334
3691 015424 032737 000010 000552
3692 015432 001430
3693 015434 032737 000020 000552
3694 015442 001024
3695 015444 032737 002000 000552
3696 015452 001020
3697 015454 105711
3698 015456 001404
3699 015460 005201
3700 015462 005205
3701 015464 001373
3702 015466 000406
3703 015470 112721 000020
3704 015474 012737 177777 014270
3705 015502 000767
3706 015504 013705 000556
3707 015510 012701 027326
3708 015514 032737 010000 000562
3709 015522 001462
3710 015524 013704 000556
3711 015530 005404
3712 015532 032737 000020 000552
3713 015540 001402
3714 015542 000241
3715 015544 006004
3716 015546 060401
3717 015550 060402
3718 015552 032737 000001 000556
3719 015560 001401
3720 015562 105722
3721 015564 032737 000020 000552
3722 015572 001431
3723 015574 000241

```

```

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

```

```

DCHK: CLR BBC ; CLEAR BAD RECORD CNTR
      CLR DERFL ; CLEAR DATA ERROR FLAG
      MOV FMCNT,RS ; LOAD CHAR COUNT
      BIT #20,UDES ; SEE IF CORE DUMP
      BEQ DCHK0 ; IF NOT: BR
      ASR RS ; RS = FC/2
DCHK0: MOV #WDATA,R1 ; SET WRITE DATA ADDR
      MOV #RDATA,R2 ; SET READ DATA ADDR
      BIT #10,UDES ; SEE IF EVEN PARITY
      BEQ DF0C0 ; IF NOT: BR
      BIT #20,UDES ; SEE IF CORE DUMP PARITY
      BNE DF0C0 ; IF SO: BR
      BIT #2000,UDES ; SEE IF PE MODE
      BNE DF0C0 ; IF SO: BR
DF0F: TSTB (R1) ; SEE IF 0 CHAR
      BEQ DF0D ; IF SO: BR
      INC R1 ; BUMP POINTER
DF0E: INC RS ; SEE IF DONE
      BNE DF0F ; IF NOT: BR
      BR DF0C ; ELSE CONTINUE
DF0D: MOVB #20,(R1)+ ; SET 20 IN PLACE OF 0
      MOV #-1,PATS ; SET PATTERN GENERATE FLAG
      BR DF0E
DF0C: MOV FMCNT,RS ; RESET CHAR CNT
      MOV #WDATA,R1 ; RESET DATA ADDRESS
DF0C0: BIT #10000,R0CMD ; SEE IF READ REVERSE
      BEQ DF0 ; IF NOT: BR
DF0B: MOV FMCNT,R4 ; GET FRAME COUNT
      NEG R4 ; SET TO WHOLE NUMBER
      BIT #20,UDES ; SEE IF CORE DUMP
      BEQ DF0B0 ; IF NOT: BR
DF0B0: ROR R4 ; SET TO FC/2
      ADD R4,R1 ; POINT TO START OF WRITE DATA
      ADD R4,R2 ; POINT TO START OF READ DATA
      BIT #1,FMCNT ; SEE IF ODD FRAME COUNT
      BEQ DF0A ; IF NOT: BR
      TSTB (R2)+ ; BUMP POINTER
DF0A: BIT #20,UDES ; SEE IF CORE DUMP
      BEQ DF0A4 ; IF NOT: BR
      CLC

```

3724	015576	132742	000001		BITB	#1 -(R2)	;SEE IF BIT 0 = 1
3725	015602	001401			BEQ	DFJAO	;IF NOT: BR
3726	015604	000261			SEC		
3727	015606	106012		DF0A0:	RORB	(R2)	
3728	015610	000241			CLC		
3729	015612	132712	000001		BITB	#1 (R2)	
3730	015616	001401			BEQ	DF0A1	
3731	015620	000261			SEC		
3732	015622	106012		DF0A1:	RORB	(R2)	;POSITION BITS FOR REVERSE CORE DUMP
3733	015624	000241			CLC		
3734	015626	132712	000001		BITB	#1 (R2)	
3735	015632	001401			BEQ	DF0A2	
3736	015634	000261			SEC		
3737	015636	106012		DF0A2:	RORB	(R2)	
3738	015640	000241			CLC		
3739	015642	132712	000001		BITB	#1 (R2)	
3740	015646	001401			BEQ	DF0A3	
3741	015650	000261			SEC		
3742	015652	106012		DF0A3:	RORB	(R2)	
3743	015654	005202			INC	R2	;RESET POINTER
3744	015656	124142		DF0A4:	CMPB	-(R1), -(R2)	;TEST DATA CHARACTER
3745	015660	001010			BNE	DF1	;IF NOT GOOD: BR
3746	015662	105037	000660		CLRB	BBC	;CLEAR BAD RECORD COUNTER
3747	015666	000411			BR	DF2	
3748	015670	122122		DF0:	CMPB	(R1)+, (R2)+	;CHECK DATA
3749	015672	001003			BNE	DF1	;IF BAD: BR
3750	015674	105037	000660		CLRB	BBC	;CLEAR BAD RECORD CNTR
3751	015700	000404			BR	DF2	
3752	015702	004737	016514	DF1:	JSR	PC, DRPKF	;GO GET DROPS AND PICKS
3753	015706	004737	016000		JSR	PC, DERR	;GO DO PRINT
3754	015712	005205		DF2:	INC	R5	;BUMP CHAR CNTR
3755	015714	001405			BEQ	DF3	;IF DONE ALL: BR
3756	015716	032737	010000	000562	BIT	#10000, RDCMD	;SEE IF READ REVERSE
3757	015724	001761			BEQ	DF0	;IF NOT: BR
3758	015726	000716			BR	DF0A	;ELSE CONTINUE READ REV
3759	015730	005037	000666	DF3:	CLR	HDRFL	;CLEAR HEADER FLAG
3760	015734	005737	000706		TST	DERFL	;SEE IF HAD DATA ERROR
3761	015740	001416			BEQ	DFX	;IF NOT: BR
3762	015742	005737	000710		TST	SERFL	
3763	015746	001013			BNE	DFX	;IF NOT DATA ERROR ONLY: BR
3764	015750	013704	000676		MOV	UNP, R4	
3765	015754	032737	010000	000562	BIT	#10000, RDCMD	;SEE IF READ REVERSE
3766	015762	001003			BNE	DF4	;IF SO: BR
3767	015764	005264	001130		INC	DATER1(R4)	;BUMP DATA ERROR FORWARD COUNTER
3768	015770	000402			BR	DFX	
3769	015772	005264	001170	DF4:	INC	DEREV1(R4)	;BUMP REVERSE DATA ERROR
3770	015776	000207		DFX:	RTS	PC	;EXIT
3771							

3772  
3773  
3774  
3775  
3776  
3777  
3778  
3779  
3780  
3781  
3782  
3783  
3784  
3785  
3786  
3787  
3788  
3789  
3790  
3791  
3792  
3793  
3794  
3795  
3796  
3797  
3798  
3799

```

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

```

3800	016000	032777	002000	162602	DERR:	BIT	#2000, @SWR	; BRANCH IF NO ERROR
3801	016006	001067				BNE	DERR4	; PRINTOUT DESIRED
3802	016010	005237	000672		DERRO:	INC	PFLG	; SET PRINT FLAG
3803	016014	005737	000666			TST	HDRFL	; SEE IF HAVE PRINTED HEADER
3804	016020	001007				BNE	DERROA	; IF SO: BR
3805	016022	004737	022570			JSR	PC, PAPRT	; PRINT CYCLE NUMBER
3806	016026	012704	024444			MOV	#MSG1, R4	; LOAD ERROR MSG ADDR
3807	016032	000004				TYPE		; TYPE MSG
3808	016034	004737	021020			JSR	PC, FRPRT	; PRINT F OR R
3809	016040	012704	024463		DERROA:	MOV	#MSG4, R4	
3810	016044	000004				TYPE		; TYPE MSG
3811	016046	010203				MOV	R2, R3	
3812	016050	162703	033334			SUB	#RDATA, R3	; POINT TO CHAR
3813	016054	005303				DEC	R3	
3814	016056	032737	010000	000562		BIT	#10000, RDCMD	; SEE IF READ REVERSE
3815	016064	001402				BEQ	DERROB	; IF NOT: BR
3816	016066	010503				MOV	R5, R3	; GET CHAR NUMBER
3817	016070	005103				COM	R3	
3818	016072	104400			DERROB:	TYPOCT		; PRINT CHAR NUMBER
3819	016074	012704	024451			MOV	#MSG2, R4	
3820	016100	000004				TYPE		; TYPE MSG
3821	016102	032737	010000	000562		BIT	#10000, RDCMD	; SEE IF READ REVERSE
3822	016110	001402				BEQ	DERROC	; IF NOT: BR
3823	016112	111103				MOVB	(R1), R3	; GET CHAR
3824	016114	000401				BR	DERROD	
3825	016116	114103			DERROC:	MOVB	-(R1), R3	; LOAD EXPECTED DATA
3826	016120	004737	024222		DERROD:	JSR	PC, DOUT	; GO PRINT CHAR
3827	016124	012704	024456			MOV	#MSG3, R4	

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

3884	016442	000402			BR	DEREX	
3885	016444	012205	177777		MOV	#-1,RS	:SET TO EOR
3886	016450	005777	162134		TST	#SWR	:BRANCH IF NOT HALT ON ERROR
3887	016454	100012			BPL	DEREX1	
3888	016456	000000			HALT		
3889	016460	005737	000672		TST	PFLG	:SEE IF PRINTED
3890	016464	001006			BNE	DEREX1	:IF SO: BR
3891	016466	032777	002000	162114	BIT	#2000,SWR	:SEE IF SHOULD PRINT
3892	016474	001002			BNE	DEREX1	:IF NOT: BR
3893	016476	000137	016010		JMP	DERRO	:ELSE PRINT
3894	016502	005037	000672		DEREX1: CLR	PFLG	:CLEAR FLAG
3895	016506	005237	000706		INC	DERFL	:BUMP DATA ERROR FLAG
3896	016512	000207			RTS	PC	:RETURN
3897							



```

3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916 016514 005037 000644
3917 016520 005037 000646
3918 016524 005037 000650
3919 016530 111137 000644
3920 016534 111237 000646
3921 016540 013704 000676
3922 016544 016437 000770 000722
3923 016552 016437 001010 000720
3924 016560 032737 010000 000562
3925 016566 001005
3926 016570 124142
3927 016572 112137 000644
3928 016576 112237 000646
3929 016602 004737 016614
3930 016606 004737 017034
3931 016612 000207
3932 016614 113703 000644
3933 016620 113704 000646
3934 016624 140403
3935 016626 001001
3936 016630 000207
3937 016632 012737 000010 000712
3938 016640 132703 000001
3939 016644 001455
3940 016646 105737 000650
3941 016652 001016
3942 016654 005277 162040
3943 016660 005777 162034
3944 016664 100045
3945 016666 032777 002000 161714
3946 016674 001402
3947 016676 004737 022570
3948 016702 004737 017100
3949 016706 000415
3950 016710 005277 162006
3951 016714 005777 162002
3952 016720 100027
3953 016722 032777 002000 161660

```

```

*****
DROPS AND PICKS SUBROUTINE:
THIS SUBROUTINE IS USED TO ACCUMULATE FROM
EACH BAD DATA CHARACTER FOUND THE NUMBER
OF BITS WHICH WERE EITHER DROPPED OR PICKED UP.
TWO COUNTERS PER SLAVE ARE USED TO ACCUMULATE THIS
INFORMATION AND CAN STORE UP TO 32K DROPS
OR PICKS BEFORE OVERFLOWING. IF OVERFLOW IS
ABOUT TO OCCUR, THESE ACCUMULATORS ARE
PRINTED IN OCTAL AND RESET TO ZERO.
THE CONTENTS OF THE ACCUMULATORS MAY BE
DISPLAYED AT ANY TIME BY SETTING CONSOLE
SWITCH FOURTEEN TO A ONE (1). THE PRINTOUT WILL OCCUR
AT THE END OF THE CURRENT BLOCK CYCLE.
*****
DRPKF: CLR TEMP1
CLR TEMP2
CLR TEMP3
MOVB (R1),TEMP1 ;LOAD GOOD CHAR
MOVB (R2),TEMP2 ;LOAD BAD CHAR
MOV UNP R4
MOV PIK1(R4),BPKP
MOV DRP1(R4),BDPP
BIT #10000,RDCMD ;SEE IF READ REVERSE
BNE DRPK ;IF SO: BR
CMPB -(R1),-(R2) ;POINT TO CHAR
MOVB (R1)+,TEMP1 ;LOAD GOOD CHAR
MOVB (R2)+,TEMP2 ;LOAD BAD CHAR
DRPK: JSR PC,DRDP ;GET DROPS
JSR PC,PICK ;GET PICKS
RTS PC ;EXIT
DROP: MOVB TEMP1,R3 ;R3 = GOOD CHAR
MOVB TEMP2,R4 ;R4 = BAD CHAR
DPC: BICB R4,R3 ;GET DROPS/PICKS
BNE DPCG ;IF SOME: BR
RTS PC ;RETURN
DPCG: MOV #10,BCNT ;SET NUMBER TO CHECK
DPC0: BITB #1,R3 ;SEE IF DROPPED OR PICKED THIS BIT
BEQ DPC2 ;IF NOT: BR
TSTB TEMP3 ;SEE IF ON PICKS
BNE DPC1 ;IF SO: BR
INC @BDPP ;BUMP DROP CNTR
TST @BDPP
BPL DPC2 ;IF NO OVERFLOW: BR
BIT #2000,@SWR ;SEE IF HAVE PRINTED DATA
BEQ DPC0A ;IF SO: BR
JSR PC,PAPRT ;PRINT CYCLE NUMBER
DPC0A: JSR PC,DPPRT ;PRINT DROPS AND PICKS
BR DPC2A
DPC1: INC @BPKP ;BUMP PICK CNTR
TST @BPKP ;SEE IF OVERFLOW
BPL DPC2 ;IF NOT: BR
BIT #2000,@SWR ;SEE IF HAVE PRINTED DATA

```

3954	016730	001402			BEQ	DPC1A		; IF SO: BR
3955	016732	004737	022570		JSR	PC, DPPRT		; PRINT CYCLE NUMBER
3956	016736	004737	017100		JSR	PC, DPPRT		; PRINT DROPS AND PICKS
3957	016742	013704	000676		DPC2A: MOV	UNP R4		
3958	016746	016403	001010		MOV	DRP1(R4), R3		; SET DROP POINTER
3959	016752	016404	000770		MOV	PIK1(R4), R4		; SET PICK POINTER
3960	016756	012737	000010	000712	MOV	#10, BCNT		; SET NUMBER OF BITS
3961	016764	005023			DPC2B: CLR	(R3)+		; CLEAR DROPS
3962	016766	005024			CLR	(R4)+		; CLEAR PICK
3963	016770	005337	000712		DEC	BCNT		; SEE IF DONE
3964	016774	001373			BNE	DPC2B		; IF NOT: BR
3965	016776	000207			RTS	PC		; EXIT
3966	017000	000241			DPC2: CLC			
3967	017002	106003			RORB	R3		; GET NEXT BIT
3968	017004	005337	000712		DEC	BCNT		; SEE IF DONE
3969	017010	001410			BEQ	DPC3		
3970	017012	062737	000002	000722	ADD	#2, BPKP		
3971	017020	062737	000002	000720	ADD	#2, BOPP		
3972	017026	000137	016640		JMP	DPC0		; CONTINUE
3973	017032	000207			DPC3: RTS	PC		; RETURN
3974	017034	013704	000676		PICK: MOV	UNP R4		; GET UNIT POINTER
3975	017040	016437	000770	000722	MOV	PIK1(R4), BPKP		; SET PICK POINTER
3976	017046	016437	001010	000720	MOV	DRP1(R4), BOPP		; SET DROP POINTER
3977	017054	113704	000644		MOVB	TEMP1, R4		; R4 = GOOD CHAR
3978	017060	113703	000546		MOVB	TEMP2, R3		; R3 = BAD CHAR
3979	017064	112737	000001	000650	MOVB	#1, TEMP3		; SET PICK FLAG
3980	017072	004737	016624		JSR	PC, DPC		; GO CHECK PICKS
3981	017076	000207			RTS	PC		; EXIT
3982	017100	012704	025075		DPPRT: MOV	#MSG26, R4		
3983	017104	000004			TYPE			; TYPE MSG
3984	017106	013704	000676		MOV	UNP R4		
3985	017112	016437	001010	000720	MOV	DRP1(R4), BOPP		; SET DROP POINTER
3986	017120	016437	000770	000722	MOV	PIK1(R4), BPKP		; SET PICK POINTER
3987	017126	062737	000016	000720	ADD	#16, BOPP		
3988	017134	062737	000016	000722	ADD	#16, BPKP		
3989	017142	012737	000010	000712	MOV	#10, BCNT		; SET NUMBER TO PRINT
3990	017150	017703	161544		DPPRT0: MOV	#BOPP, R3		
3991	017154	104400			TYPOCT			; PRINT DROPS
3992	017156	005337	000712		DEC	BCNT		; SEE IF DONE
3993	017162	001404			BEQ	DPPRT1		; IF NOT: BR
3994	017164	162737	000002	000720	SUB	#2, BOPP		; BUMP POINTER
3995	017172	000766			BR	DPPRT0		; CONTINUE FOR ALL 8 BITS
3996	017174	012737	000010	000712	DPPRT1: MOV	#10, BCNT		; SET NUMBER TO PRINT
3997	017202	012704	025106		MOV	#MSG27, R4		
3998	017206	000004			TYPE			; TYPE MSG
3999	017210	017703	161506		DPPRT2: MOV	#BPKP, R3		
4000	017214	104400			TYPOCT			; PRINT PICKS
4001	017216	005337	000712		DEC	BCNT		; SEE IF DONE
4002	017222	001404			BEQ	DPPRTX		; IF SO: BR
4003	017224	162737	000002	000722	SUB	#2, BPKP		; BUMP POINTER
4004	017232	000766			BR	DPPRT2		; CONTINUE FOR ALL 8 BITS
4005	017234	000207			DPPRTX: RTS	PC		; RETURN

4006  
4007  
4008  
4009  
4010  
4011  
4012  
4013  
4014  
4015  
4016  
4017  
4018  
4019  
4020  
4021  
4022  
4023  
4024  
4025  
4026  
4027  
4028  
4029  
4030  
4031  
4032  
4033  
4034  
4035  
4036  
4037  
4038  
4039  
4040  
4041  
4042  
4043  
4044  
4045  
4046  
4047  
4048  
4049  
4050  
4051  
4052  
4053  
4054  
4055  
4056  
4057  
4058  
4059  
4060  
4061

017236 013703 000556  
017242 032703 000001  
017246 001401  
017250 005303  
017252 005403  
017254 032737 000020 000552  
017262 001401  
017264 006203  
017266 032737 000010 000674  
017274 001414  
017276 032737 010000 000562  
017304 001405  
017306 012703 033334  
017312 162703 000002  
017316 000405  
017320 062703 033334  
017324 000402  
017326 062703 027326  
017332 010337 020774  
017336 012704 000007  
017342 012701 020776  
017346 005021  
017350 005304  
017352 001375  
017354 020377 161134  
017360 001402  
017362 005237 020776  
017366 032737 000010 000674  
017374 001006  
017376 005777 161114  
017402 001441

```

*****
STATUS CHECK SUBROUTINE:

THIS SUBROUTINE IS USED TO PERFORM A CHECK OF
BOTH THE MASSBUS CONTROLLER (RH11) AND THE TAPE
CONTROLLER (TMD2). THE RH11 IS CHECKED FOR ERRORS
AS REFLECTED IN REGISTERS CS1 AND CS2 AND ALSO THAT
THE BUS ADDRESS (BA) AND WORD COUNT (WC) ARE
CORRECT. THE TMD2 IS CHECKED FOR DRIVE STATUS (DS),
DRIVE ERRORS (ER), AND PROPER FRAME COUNT. THE SPECIAL
CHECK CHARACTERS (CRC+LRC) ARE ALSO CHECKED WHEN
APPROPRIATE (IE: NRZ READ OR WRITE). CERTAIN TYPES
OF DRIVE ERRORS IN PE OPERATION WILL BE ACCOMPANIED
BY THE DISPLAY OF THE DEAD TRACK REGISTER (CC). THESE
TYPES ARE ER BITS 15,10,7,6. THE PRINTOUTS OF BAD
CRC,LRC,FC, AND BA WILL SHOW BOTH THE EXPECTED AND
RECEIVED VALUES (IE: EXPT-RCVD). ONLY THOSE REGISTERS
WHICH ARE IN ERROR WILL BE PRINTED AND ALL PRINTOUTS
ARE IN OCTAL FORMAT WITH NO LEADING ZEROS. AS IN
DATA ERRORS, STATUS ERRORS ARE PRECEDED BY HEADER
DESCRIBING THE HARDWARE UNDER TEST, THE BLOCKING
INFORMATION, AND THE ERROR TYPE.
*****
ERCHK: MOV FMCNT,R3 ;GET FRAME COUNT
        BIT #1,R3 ;SEE IF 000
        BEQ ERO ;IF NOT: BR
        DEC R3 ;BUMP COUNT
        NEG R3
        BIT #20,UDES ;SEE IF CORE DUMP
        BEQ EROB ;IF NOT: BR
        ASR R3 ;SET TO FC/2
        BIT #10,MTC1 ;SEE IF WRITE OP
        BEQ ER1 ;IF SO: BR
        BIT #10000,RDCMD
        BEQ EROA
        MOV #RDATA,R3
        SUB #2,R3 ;SET POINTER
        BR ER2
        ADD #RDATA,R3 ;BUILD EXPT READ ADDRESS
        BR ER2
        ADD #RDATA,R3 ;BUILD EXPT WRITE ADDRESS
        MOV R3,CADR ;SAVE ADDRESS
        MOV #7,R4
        MOV #BAER,R1
        CLR (R1)+ ;CLEAR FLAGS
        DEC R4
        BNE ER2A0
        CMP R3,2BA ;SEE IF ADDRESS OK
        BEQ ER2A1 ;IF SO: BR
        INC BAER ;SET BUS ADDRESS ERROR
        BIT #10,MTC1 ;SEE IF WRITE OPER
        BNE ER2B ;IF NOT: BR
        TST @FC ;SEE IF FC=0
        BEQ ER3 ;IF SO: BR

```

4062	017404	005237	021004			INC	FCER		;SET FC ERROR
4063	017410	000436				BR	ER3		
4064	017412	032737	000040	000674	ER2B:	BIT	#40,MTC1		;SEE IF SPACE OPER
4065	017420	001766				BEQ	ER2A		;IF SO: BR
4066	017422	005737	000700			TST	TMFLG		;SEE IF TM TIME
4067	017426	001011				BNE	ER20		;IF SO: BR
4068	017430	013703	000556			MOV	FMCNT,R3		
4069	017434	005403				NEG	R3		;R3 = EXPT RECORD SIZE
4070	017436	020377	161054		ER2C:	CMP	R3,#FC		;SEE IF FC = EXPT
4071	017442	001421				BEQ	ER3		;IF SO: BR
4072	017444	005237	021004			INC	FCER		;SET FC ERROR FLAG
4073	017450	000416				BR	ER3		
4074	017452	032737	002000	000552	ER2D:	BIT	#2000,UDES		;SEE IF PE
4075	017460	001346				BNE	ER2A		;IF SO: BR
4076	017462	032737	010000	000562		BIT	#10000,RDCMD		;SEE IF READ REVERSE
4077	017470	001003				BNE	ER2E		;IF SO: BR
4078	017472	012703	000002			MOV	#2,R3		
4079	017476	000757				BR	ER2C		;LOOK FOR EXPT = 2
4080	017500	012703	000001		ER2E:	MOV	#1,R3		
4081	017504	000754				BR	ER2C		;GO CHECK FC FOR TM
4082	017506	032777	160000	160774	ER3:	BIT	#160000,#C1		;SEE IF COUNT ERROR
4083	017514	001437				BEQ	ER4		
4084	017516	017703	160776			MOV	#CS,R3		;GET CONT STATUS REG
4085	017522	042703	000307			BIC	#307,R3		;MASK OUT IR,OR,UNIT NO. & SEE IF OTHER ERRORS
4086	017526	001406				BEQ	ER3A		;IF NOT: BR
4087	017530	005737	000700			TST	TMFLG		;SEE IF TAPE MARK TIME
4088	017534	001425				BEQ	ER3B		;IF NOT: BR
4089	017536	042703	001000			BIC	#1000,R3		;MASK MISSED TRANS & BR IF OTHER ERRORS
4090	017542	001022				BNE	ER3B		
4091	017544	032777	060000	160736	ER3A:	BIT	#60000,#C1		;SEE IF EITHER TRE OR MCPE
4092	017552	001420				BEQ	ER4		;IF NOT: BR
4093	017554	005737	000700			TST	TMFLG		;SEE IF TM TIME
4094	017560	001413				BEQ	ER3B		;IF NOT: BR
4095	017562	017703	160736			MOV	#ER,R3		;GET ERROR REGISTER
4096	017566	032737	000010	000552		BIT	#10,UDES		;SEE IF EVEN PARITY
4097	017574	001402				BEQ	ER3A1		;IF NOT: BR
4098	017576	042703	000100			BIC	#100,R3		;MASK PAR
4099	017602	042703	001000		ER3A1:	BIC	#1000,R3		;MASK FCE
4100	017606	001402				BEQ	ER4		;IF NO ERRORS EXCEPT FCE: BR
4101	017610	005237	021000		ER3B:	INC	CONER		;SET CONT ERROR FLAG
4102	017614	032777	040000	160700	ER4:	BIT	#40000,#DS		;SEE IF DRIVE ERROR
4103	017622	001420				BEQ	ER6		;IF NOT: BR
4104	017624	005737	000700			TST	TMFLG		;SEE IF TAPE MARK TIME
4105	017630	001413				BEQ	ER4A		;IF NOT: BR
4106	017632	017703	160666			MOV	#ER,R3		;GET ER
4107	017636	032737	000010	000552		BIT	#10,UDES		;SEE IF EVEN PARITY
4108	017644	001402				BEQ	ER4A1		;IF NOT: BR
4109	017646	042703	000100			BIC	#100,R3		;MASK PAR
4110	017652	042703	001000		ER4A1:	BIC	#1000,R3		;MASK OUT FCE & BRANCH IF
4111	017656	001402				BEQ	ER6		;NO OTHER ERRORS
4112	017660	005237	021002		ER4A:	INC	DRVER		;SET DRIVER ERROR FLAG
4113	017664	032737	002000	000552	ER6:	BIT	#2000,UDES		
4114	017672	001071				BNE	ERPT		;IF IN PE MODE: BR
4115	017674	032777	020000	160706		BIT	#20000,#SWR		;SEE IF NO DATA CHECK
4116	017702	001065				BNE	ERPT		;IF NOT: BR (ALLOW READ OF UNKNOWN TAPES)
4117	017704	032737	000040	000674		BIT	#40,MTC1		;SEE IF WRITE OR READ OP

4118	017712	001461			BEQ	ERPT	: IF NOT: BR
4119	017714	005737	000700		TST	TMFLG	: SEE IF TAPE MARK TIME
4120	017720	001413			BEQ	ER6A	: IF NOT: BR
4121	017722	013737	015362	021016	MOV	EXCRC,CRCSV	: SAVE CRC
4122	017730	013737	015364	021014	MOV	EXLRC,LRCV	: SAVE LRC
4123	017736	001337	015362		CLR	EXCRC	
4124	017742	012737	00123	015364	MOV	#23,EXLRC	: SET CRC/LRC FOR TM
4125	017750	032737	000060	000552	BIT	#60,UDES	: SEE IF FORMAT 14
4126	017756	001037			BNE	ERPT	: IF NOT: BR
4127	017760	017703	160544		MOV	@CC,R3	: GET CRC CHARACTER
4128	017764	042703	177000		BIC	#177000,R3	
4129	017770	023703	015362		CMF	EXCRC,R3	
4130	017774	001402			BEQ	ER7	: IF CRC GOOD: BR
4131	017776	005237	021010		INC	CRCR	: SET ERROR FLAG
4132	020002	017703	160526		MOV	@MR,R3	: GET LRC
4133	020006	000303			SWAB	R3	
4134	020010	005703			TST	R3	
4135	020012	100002			BPL	ER10	
4136	020014	052703	000400		BIS	#400,R3	
4137	020020	042703	177000		BIC	#177000,R3	
4138	020024	023703	015364		CMF	EXLRC,R3	
4139	020030	001412			BEQ	ERPT	: IF LRC GOOD: BR
4140	020032	010337	021012		MOV	R3,ACTLRC	: SAVE ACTUAL LRC
4141	020036	005237	021006		INC	LRCR	: SET LRC ERROR FLAG
4142	020042	032737	010000	000562	BIT	#10000,ROCMD	: SEE IF READ REVERSE
4143	020050	001402			BEQ	ERPT	: IF NOT: BR
4144	020052	005037	021006		CLR	LRCR	: ELSE CLEAR LRC ERROR
4145	020056	012703	000006		MOV	#6,R3	
4146	020062	005037	000710		CLR	SERFL	: CLEAR ERROR FLAG
4147	020066	005037	000724		CLR	ERSAV	
4148	020072	012704	020776		MOV	#BAER,R4	
4149	020076	005724			TST	(R4)+	: SEE IF ANY ERROR
4150	020100	001004			BNE	ERPTG	: IF SO: BR
4151	020102	005303			DEC	R3	
4152	020104	001374			BNE	ERPTT	
4153	020106	000137	020740		JMP	ERPXI	
4154	020112	005237	000710		INC	SERFL	: SET ERROR FLAG
4155	020116	017737	160402	000724	MOV	@ER,ERSAV	: SAVE ERROR REGISTER
4156	020124	032777	002000	160456	BIT	#2000,@SHR	: SEE IF PRINT
4157	020132	001420			BEQ	ERPTO	: IF SO: BR
4158	020134	022737	000002	000714	CMF	#2,RTYFL	: SEE IF READ RETRY
4159	020142	001006			BNE	ERPTG1	: IF NOT: BR
4160	020144	013703	000704		MOV	RTCNT,R3	
4161	020150	005203			INC	R3	: BUMP RETRY COUNT
4162	020152	020337	000604		CMF	R3,RETRY	: SEE IF LAST RETRY
4163	020156	001406			BEQ	ERPTO	: IF SO: BR
4164	020160	022737	000002	021002	CMF	#2,DAVER	: SEE IF TM STATUS ERROR
4165	020166	001402			BEQ	ERPTO	: IF SO: BR
4166	020170	000137	020620		JMP	ERPXD	
4167	020174	005237	000672		INC	PFLG	
4168	020200	004737	022570		JSR	PC,PAPRT	: PRINT HEADER
4169	020204	013704	000654		MOV	EMADOR,R4	
4170	020210	000004			TYPE		: TYPE MSG
4171	020212	004737	021020		JSR	PC,FRPRT	: PRINT F OR R
4172	020216	005737	000700		TST	TMFLG	
4173	020222	001407			BEQ	ERPTI	

4174	020224	022737	026026	000654	CMP	#MSG54,EMADDR	
4175	020232	001403			BEQ	ERPT1	
4176	020234	012704	026044		MOV	#MSG56,R4	;PRINT TM
4177	020240	000004			TYPE		;TYPE MSG
4178	020242	005737	021000		ERPT1: TST	CONER	
4179	020246	001414			BEQ	ERPT2	;IF NO CONT ERROR: BR
4180	020250	012704	024725		MOV	#MSG23,R4	
4181	020254	000004			TYPE		;TYPE MSG
4182	020256	017703	160226		MOV	@C1,R3	
4183	020262	104400			TYPOCT		;PRINT CONTROL 1
4184	020264	012704	024752		MOV	#MSG23D,R4	;PRINT CS TAG
4185	020270	000004			TYPE		;TYPE MSG
4186	020272	017703	160222		MOV	@CS,R3	
4187	020276	104400			TYPOCT		;PRINT CONT STATUS
4188	020300	005737	021002		ERPT2: TST	DRVER	
4189	020304	001414			BEQ	ERPT3	;IF NO DRIVE ERROR: BR
4190	020306	012704	024760		MOV	#MSG23E,R4	
4191	020312	000004			TYPE		;TYPE MSG
4192	020314	017703	160202		MOV	@DS,R3	
4193	020320	104400			TYPOCT		;PRINT DRIVE STATUS
4194	020322	012704	024765		MOV	#MSG23F,R4	
4195	020326	000004			TYPE		;TYPE MSG
4196	020330	017703	160170		MOV	@ER,R3	
4197	020334	104400			TYPOCT		;PRINT DRIVE ERROR
4198	020336	005737	020776		ERPT3: TST	BAER	
4199	020342	001416			BEQ	FRPT4	;IF NO BA ERROR: BR
4200	020344	012704	024740		MOV	#MSG23B,R4	
4201	020350	000004			TYPE		;TYPE MSG
4202	020352	017703	160136		MOV	@BA,R3	
4203	020356	104400			TYPOCT		;PRINT BUS ADDRESS
4204	020360	012737	000255	000640	MOV	@255,T0B	
4205	020366	004737	023744		JSR	PC,T0G	;PRINT /
4206	020372	013703	020774		MOV	CADER,R3	
4207	020376	104400			TYPOCT		;PRINT EXPT BUS ADDRESS
4208	020400	005737	021004		ERPT4: TST	FCER	
4209	020404	001406			BEQ	ERPT5	;IF NO FC ERROR: BR
4210	020406	012704	024745		MOV	#MSG23C,R4	
4211	020412	000004			TYPE		;TYPE MSG
4212	020414	017703	160076		MOV	@FC,R3	
4213	020420	104400			TYPOCT		;PRINT FRAME COUNT
4214	020422	012704	024733		ERPT5: MOV	#MSG23A,R4	
4215	020426	000004			TYPE		;TYPE MSG
4216	020430	017703	160056		MOV	@WC,R3	
4217	020434	104400			TYPOCT		;PRINT WORD COUNT
4218	020436	005737	021010		TST	CRCR	
4219	020442	001420			BEQ	ERPT5A	;IF NO CRC ERROR: BR
4220	020444	012704	026071		MOV	#MSG58,R4	
4221	020450	000004			TYPE		;TYPE MSG
4222	020452	017703	160052		MOV	@CC,R3	
4223	020456	042703	177000		BIC	@177000,R3	
4224	020462	104400			TYPOCT		;PRINT ACTUAL CRC
4225	020464	012737	000255	000640	MOV	@255,T0B	
4226	020472	004737	023744		JSR	PC,T0G	
4227	020476	013703	015362		MOV	EXCRC,R3	
4228	020502	104400			TYPOCT		;PRINT EXPECTED CRC
4229	020504	005737	021006		ERPT5A: TST	LRCR	

4230	020510	001416				BEQ	ERPT6		; IF NO LRC ERROR: BR
4231	020512	012704	026077			MOV	MSG59,R4		
4232	020514	000004				TYPE			; TYPE MSG
4233	020516	013703	021012			MOV	ACTLRC,R3		
4234	020518	104400				TYPOCT			; PRINT ACTUAL LRC
4235	020520	012737	000255	000640		MOV	#255,T08		
4236	020522	004737	023744			JSR	PC,T08		
4237	020524	013703	015364			MOV	EXLRC,R3		
4238	020526	104400				TYPOCT			; PRINT EXPECTED LRC
4239	020528	005737	021002		ERPT6:	TST	DRVER		
4240	020530	001421				BEQ	ERPT7		; IF NO DRIVE ERROR: BR
4241	020532	032737	002000	000552		BIT	#2000,UDES		
4242	020534	001415				BEQ	ERPT7		; IF NO FE: BR
4243	020536	017704	157734			MOV	ER,R4		
4244	020538	042704	075477			BIC	#75477,R4		; MASK OUT ALL BUT BITS 15,10,7,6
4245	020540	001410				BEQ	ERPT7		; IF NO CONDITIONALS SET: BR
4246	020542	012704	024777			MOV	MSG23H,R4		
4247	020544	000004				TYPE			; TYPE MSG
4248	020546	017703	157720			MOV	ACC,R3		
4249	020548	042703	177000			BIC	#177000,R3		; MASK CC
4250	020550	104400				TYPOCT			; PRINT CHECK CHARACTERS
4251	020552	000240			ERPT7:	NOP			
4252	020554	005777	157764		ERPXD:	TST	ASW		; BRANCH IF NOT HALT ON ERROR
4253	020556	101012				BPL	ERPXD		
4254	020558	001000				HALT			
4255	020560	005737	000672			TST	PFLG		; SEE IF HAVE PRINTED
4256	020562	001006				BNE	ERPXD		; IF SO: BR
4257	020564	032777	002000	157744		BIT	#2000,ASW		; SEE IF SHOULD PRINT
4258	020566	001002				BNE	ERPXD		; IF NOT: BR
4259	020568	000137	020174			JMP	ERPT0		; PRINT ERROR
4260	020570	005037	000672		ERPXD:	CLR	PFLG		
4261	020572	005737	000566			TST	CRCC		; BRANCH IF CRC ERROR
4262	020574	001007				BNE	IS		; CORRECTION DESIRED
4263	020576	012777	000040	157626		MOV	#40,ACS		; ELSE INIT
4264	020578	013777	000550	157620		MOV	DVN,ACS		; RESET DRIVE NUMBER
4265	020580	000414				BR	2S		
4266	020582	012777	000011	157600	1S:	MOV	#11,AC1		; DRIVE CLEAR
4267	020584	017704	157612			MOV	AS,R4		
4268	020586	010477	157606			MOV	R4,AS		; CLEAR AS
4269	020588	013704	000510			MOV	C1,R4		
4270	020590	005204				INC	R4		
4271	020592	152714	000100			BISB	#100,(R4)		; RESET TRE
4272	020594	013777	000552	157602	2S:	MOV	UDES,ATC		; RESET TC
4273	020596	032737	000040	000674	ERPXD1:	BIT	#40,ATC1		
4274	020598	001411				BEQ	ERPXD2		; IF NOT READ/WRITE OP: BR
4275	020600	005737	000700			TST	TMFLG		
4276	020602	001406				BEQ	ERPXD2		; IF NOT TM TIME: BR
4277	020604	013737	021016	015362		MOV	CRCSV,EXCRC		; RESTORE CRC
4278	020606	013737	021014	015364		MOV	LRCNV,EXLRC		; RESTORE LRC
4279	020608	000207			ERPXD2:	RTS	PC		; EXIT
4280	020610	000000			CADER:	0			; EXPT ADDRESS SAVE
4281	020612	000000			BAER:	0			
4282	020614	000000			CONER:	0			
4283	020616	000000			DRVER:	0			
4284	020618	000000			FCER:	0			
4285	020620	000000			LRCER:	0			

4286 021010 000000  
4287 021012 000000  
4288 021014 000000  
4289 021016 000000

CRCER: 0  
ACTLRC: 0  
LRCSV: 0  
CRCSV: 0

4290  
4291  
4292  
4293  
4294  
4295  
4296  
4297  
4298  
4299

\*\*\*\*\*  
: F FOR FORWARD/R FOR REVERSE PRINT SUBROUTINE:  
: THIS SUBROUTINE IS USED TO PRINT OUT THE  
: TAPE DIRECTION USED WHEN ANY ERROR IS  
: DETECTED IN STATUS OF READ OR WRITE, DATA, OR  
: SPACING OPERATIONS.  
\*\*\*\*\*

4300 021020 032737 000010 000674  
4301 021026 001411  
4302 021030 012704 024633  
4303 021034 032737 000002 000674  
4304 021042 001002  
4305 021044 012704 024630  
4306 021050 000004  
4307 021052 000207  
4308

FRPRT: BIT #10,MTC1 ;SEE IF WRITE COMMAND  
BEQ 25 ;IF SO: BR  
MOV #MSG17,R4 ;SET TO TYPE REVERSE MSG  
BIT #2,MTC1 ;BRANCH IF REVERSE  
BNE 15  
MOV #MSG16,R4 ;SET FORWARD MESSAGE  
15: TYPE ;TYPE MSG  
25: RTS PC ;EXIT



4309  
4310  
4311  
4312  
4313  
4314  
4315  
4316  
4317  
4318  
4319  
4320  
4321  
4322  
4323  
4324  
4325  
4326  
4327  
4328  
4329  
4330  
4331  
4332  
4333  
4334  
4335  
4336  
4337  
4338  
4339  
4340  
4341  
4342  
4343  
4344  
4345  
4346  
4347  
4348  
4349  
4350  
4351  
4352  
4353  
4354  
4355  
4356  
4357  
4358  
4359  
4360  
4361  
4362  
4363  
4364

021054 005037 000644  
021060 013777 000550 157432  
021066 032777 010000 157426  
021074 001026  
021076 005237 000644  
021102 001371  
021104 004737 022570  
021110 032737 000010 000674  
021116 001004  
021120 012704 024470  
021124 000004  
021126 000405  
021130 012704 024475  
021134 000004  
021136 004737 021020  
021142 012704 025055  
021146 000004  
021150 000000  
021152 032777 020000 157342  
021160 001411  
021162 004737 022570  
021166 012704 027154  
021172 000004  
021174 032777 020000 157320  
021202 001374  
021204 022737 000026 000674  
021212 001003

```
*****  
TAPE COMMAND EXECUTE SUBROUTINE:  
  
THIS SUBROUTINE IS USED TO EXECUTE THE  
MAG TAPE COMMAND DESCRIBED BY THE READ  
OR WRITE ROUTINE. THE FINAL COMMAND IS  
SENT TO THE DEVICE REGISTER ALONG WITH THE  
INTERRUPT ENABLE AND GO BITS.  
ONCE THE COMMAND IS ISSUED, AN INTERRUPT  
TIMER IS STARTED AND IF NO INTERRUPT IS RETURNED  
BEFORE TIME OUT OCCURS, AN ERROR WILL BE  
PRINTED AND THE PROGRAM STOPPED. TESTING MAY  
BE RESUMED BY PRESSING THE CONTINUE SWITCH.  
TWO INTERRUPT HANDLERS ARE USED, ONE FOR MAG TAPE  
AND ANOTHER FOR TELETYPE (TTY).  
UPON RECEIPT OF A MAG TAPE INTERRUPT, HOUSEKEEPING  
IS PERFORMED AND CONTROL RETURNED TO THE CALLING  
ROUTINE (READ, WRITE, ETC).  
RECEIPT OF A TTY INTERRUPT WILL CAUSE THE  
PROGRAM TO CHECK FOR ENTRY OF A CNTRL C CHARACTER.  
IF NOT CNTRL C, THEN CONTINUATION OF WAIT FOR MAG  
TAPE INTERRUPT IS RETURNED. IF HOWEVER, THE TTY  
INTERRUPT WAS CAUSED BY ENTRY OF A CNTRL C,  
THEN AT THIS TIME REQUESTS FOR NEW STALL VALUES  
ARE PRINTED AND THE RESPONSES ENTERED. RESUMPTION  
OF TAPE INTERRUPT WAIT IS THEN RESUMED.  
*****  
  
CLR      TEMP1  
MOV      DYN, DCS      ; SET DRIVE NO.  
BIT      #10000, DCS   ; SEE IF HAVE MOL  
BNE     TAPG3         ; IF SO: BR  
INC      TEMP1        ; SEE IF TIMED OUT  
BNE     TAPG0         ; WAIT FOR READY  
JSR     PC, PAPRT     ; PRINT CYCLE NUMBER  
BIT      #10, MTC1    ; SEE IF WRITE OP  
BNE     TAPG1         ; IF NOT: BR  
MOV      #MSG5, R4  
TYPE                    ; TYPE MSG  
BR      TAPG2  
TAPG1:  MOV      #MSG6, R4  
TYPE                    ; TYPE MSG  
JSR     PC, FRPRT     ; PRINT F OR R  
TAPG2:  MOV      #MSG25, R4  
TYPE                    ; TYPE MSG  
HALT  
TAPG3:  BIT      #20000, DCS ; SEE IF PIP RESET  
BEQ     TAPG3F       ; IF SO: BR  
JSR     PC, PAPRT     ; PRINT HEADER  
MOV      #MSG116, R4  
TYPE                    ; TYPE MSG  
IS:     BIT      #20000, DCS  
BNE     IS           ; AWAIT PIP RESET  
TAPG3F: CMP      #26, MTC1 ; SEE IF WRITE TM  
BNE     TAPG3A       ; IF NOT: BR
```

4365	021214	012704	177777		MOV	#-1,R4	;ELSE SET FC FOR -1
4366	021220	000406			BR	TAPG3B	
4367	021222	013704	000556	TAPG3A:	MOV	FM CNT,R4	
4368	021226	032704	000001		BIT	#1,R4	
4369	021232	001401			BEQ	TAPG3B	
4370	021234	005304			DEC	R4	
4371	021236	000261		TAPG3B:	SEC		
4372	021240	006004			ROR	R4	;SET WC = FC/2 FOR NORMAL FORMAT
4373	021242	032737	000020 000552		BIT	#20,UDES	;SEE IF CORE DUMP FORMAT
4374	021250	001402			BEQ	TAPG3C	;IF NOT: BR
4375	021252	000261			SEC		
4376	021254	006004			ROR	R4	;SET WC = FC/4 FOR CORE DUMP
4377	021256	010477	157230	TAPG3C:	MOV	R4,WC	;SET WORD COUNT
4378	021262	012777	000011 157220		MOV	#11,JC1	;DRIVE CLEAR
4379	021270	017777	157222 157220		MOV	FC,FC	;RESET FC LOADED
4380	021276	005737	000570		TST	INTR	;SEE IF INTERCHANGE READ
4381	021302	001407			BEQ	TAPG3D	;IF NOT: BR
4382	021304	032737	000040 000674		BIT	#40,MTC1	;SEE IF READ OP
4383	021312	001403			BEQ	TAPG3D	;IF NOT: BR
4384	021314	012777	000003 157212		MOV	#3,IMR	;SET INTERCHANGE READ MAINT. MODE
4385	021322	013704	000674	TAPG3D:	MOV	MTC1,R4	;GET COMMAND
4386	021326	042704	177707		BIC	#177707,R4	;MASK OP CODE
4387	021332	022704	000030		CMF	#30,R4	;SEE IF SPACE OP CODE
4388	021336	001403			BEQ	TAPG3E	;IF SO: BR
4389	021340	012737	177740 000670		MOV	#-40,STAL	;SET INTERRUPT DELAY MULT TO 40
4390	021346	052737	000101 000674	TAPG3E:	BIS	#101,MTC1	;SET INTERRUPT ENABLE AND GO
4391	021354	000240			NOP		
4392	021356	013777	000674 157124		MOV	MTC1,JC1	;EXECUTE COMMAND
4393	021364	005077	157216		CLR	PSW	;CLEAR PRIORITY
4394	021370	005037	000644		CLR	TEMP1	
4395	021374	005237	000644	TAPG4:	INC	TEMP1	;SEE IF HAVE TIMED OUT
4396	021400	001375			BNE	TAPG4	;IF NOT: BR
4397	021402	005237	000670		INC	STAL	
4398	021406	001372			BNE	TAPG4	;DO TIME DELAY MULTIPLIER
4399	021410	012777	000340 157170	TAPG5:	MOV	#340,PSW	;RESET PRIORITY
4400	021416	032777	002000 157164		BIT	#2000,PSW	;SEE IF SHOULD PRINT ERRORS
4401	021424	001012			BNE	TAPG6	;IF NOT: BR
4402	021426	004737	022570		JSR	PC,PAPRT	;PRINT CYCLE NUMBER
4403	021432	013704	000654		MOV	EMADDR,R4	
4404	021436	000004			TYPE		;TYPE MSG
4405	021440	004737	021020		JSR	PC,FRPRT	;PRINT F OR R
4406	021444	012704	025035		MOV	#MSG24,R4	
4407	021450	000004			TYPE		;TYPE MSG
4408	021452	005777	157132	TAPG6:	TST	PSW	;BRANCH IF NOT HALT ON ERROR
4409	021456	100001			BPL	TAPG7	
4410	021460	000000			HALT		
4411	021462	000137	021720	TAPG7:	JMP	MTINTA	;RETURN TO CALLING ROUTINE
4412							



```

4464
4465
4466
4467
4468
4469
4470
4471
4472
4473 021734 012704 026576
4474 021740 000004
4475 021742 012705 000744
4476 021746 012701 000002
4477 021752 012702 000001
4478 021756 012703 000000
4479 021762 004737 023340
4480 021766 005037 000740
4481 021772 004737 022110
4482 021776 012704 026527
4483 022002 000004
4484 022004 012704 026556
4485 022010 000004
4486 022012 013703 000740
4487 022016 104400
4488 022020 012704 026565
4489 022024 000004
4490 022026 012700 000746
4491 022032 005710
4492 022034 100403
4493 022036 012003
4494 022040 104400
4495 022042 000773
4496 022044 004737 022274
4497 022050 004737 022434
4498 022054 022737 000007 000740
4499 022062 001403
4500 022064 005237 000740
4501 022070 000740
4502 022072 005737 000744
4503 022076 001003
4504 022100 000137 005004
4505 022104 000000
4506 022106 000727

```

```

*****
: 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.
*****

```

```

ASEQ:  MOV  #MSG104,R4
        TYPE
        MOV  #ASEQCF,R5      ;TYPE MSG
        MOV  #2,R1          ;SET ADDRESS OF ENTRY
        MOV  #1,R2          ;SET SIZE OF ENTRY
        MOV  #C,R3          ;SET UPPER LIMIT
        JSR  PC,TTR         ;SET LOWER LIMIT
        JSR  PC,TTR         ;GO GET INPUT
ASEQ0:  CLR  ADRVN          ;CLEAR DRV NUM
ASEQ1:  JSR  PC,HRDS        ;GO SELECT HARDWARE CONFIGURATION
        MOV  #MSG101,R4
        TYPE
        MOV  #MSG102,R4    ;TYPE MSG
        TYPE
        MOV  ADRVN,R3      ;TYPE MSG
        TYPOCT
        MOV  #MSG103,R4    ;PRINT TMO3
        TYPE
        MOV  #UN1,R0       ;TYPE MSG
        TST  (R0)          ;POINT TO START OF SLAVE TABLE
        BMI ASEQ3          ;SEE IF END
        MOV  (R0)+,R3      ;IF SO: BR
        TYPOCT
        BR   ASEQ2        ;PRINT SLAVE TABLE
ASEQ3:  JSR  PC,AMOD1      ;DO ALL
        JSR  PC,AMOD2      ;GO DO MODE 1(NRZ)
ASEQ4:  CMP  #7,ADRVN      ;GO DO MODE 2(PE)
        BEQ  ASEQX         ;SEE IF DONE ALL DRIVES
        INC  ADRVN         ;IF SO: BR
        BR   ASEQ1        ;BUMP DRIVE NUMBER
ASEQX:  TST  ASEQCF        ;CONTINUE
        BNE ASEQXX        ;SEE IF CONTINUOUS AUTO SEQ
        JMP  TEND         ;IF SO: BR
ASEQXX: BR   ASEQ0

```

```

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

```

```

4542
4543
4544
4545 022274 005037 000656
4546 022300 012701 000746
4547 022304 052721 001700
4548 022310 022711 177777
4549 022314 001373
4550 022316 004737 005070
4551 022322 012737 000006 000742
4552 022330 012737 174000 000556
4553 022336 012737 000100 000554
4554 022344 013737 000740 000550
4555 022352 012737 000001 000560
4556 022360 005037 000564
4557 022364 005037 000570
4558 022370 004737 003346
4559 022374 012737 000010 000560
4560 022402 004737 003346
4561 022406 012737 000014 000560
4562 022414 004737 003346
4563 022420 012737 177777 000560 3$:
4564 022426 004737 003346
4565 022432 000207

;SUBROUTINE TO SELECT NRZ AUTO TEST MODE*****
AMOD1: CLR BLCNTR ;ASSURE BLOCK COUNTER IS 0
MOV #UNI,R1 ;GET START OF SLAVE TABLE
1$: BIS #1700,(R1)+ ;SET ALL SLAVE TO NRZ,NORM,ODD
CAP #-1,(R1) ;LOOP UNTIL REACHED END OF TABLE
BNE 1$
JSR PC,RUNDA ;GO REWIND ALL AVAIL SLAVES
MOV #6,R1 ;SET NUMBER OF BLOCKS FOR MODE 1
MOV #-4000,FMCNT ;SET FC = 4000
MOV #100,RCNT ;SET REC CNTR = 100
MOV ADRVN,DVN ;SELECT DRIVE
MOV #1,PATRN ;SELECT PATTERN 1
CLR TMX ;ASSURE NO TMK
CLR INTRF ;ASSURE NORMAL READ
JSR PC,STAUTO ;GO DO AUTO MODE 1
MOV #10,PATRN ;SELECT PATTERN 10
JSR PC,STAUTO ;GO DO PATTERN 10
MOV #14,PATRN ;SELECT PATTERN 14
JSR PC,STAUTO
3$: MOV #-1,PATRN ;SELECT AUTO RANDOM DATA
JSR PC,STAUTO
RTS PC ;RETURN TO SEQ

```

```

4566
4567
4568
4569 022434 005037 000656
4570 022440 012701 000746
4571 022444 042711 001700
4572 022450 052721 002370
4573 022454 022711 177.77
4574 022460 001371
4575 022462 004737 005070
4576 022466 012737 000006 000742
4577 022474 012737 174000 000556
4578 022502 012737 000100 000554
4579 022510 012737 000010 000560
4580 022516 004737 003346
4581 022522 012737 000014 000560
4582 022530 004737 003346
4583 022534 012737 000015 000560
4584 022542 004737 003346
4585 022546 012737 177777 000742
4586 022554 012737 177777 000560
4587 022562 004737 003346
4588 022566 000207
4589
4590

```

;SUBROUTINE TO SELECT PE AUTO TEST MODE\*\*\*\*\*

```

AMOD2: CLR      BLCNTR      ;CLEAR BLOCK CNTR
        MOV      #UN1,R1    ;SET START OF SLAVE TABLE
1$:     BIC      #1700,(R1)  ;CLEAR NRZ
        BIS      #2300,(R1)+ ;SET TO PE NORM, 000
        CMP      #-1,(R1)   ;LOOP UNTIL END OF TABLE
        BNE     1$
        JSR     PC,RWINDA   ;REWIND ALL SLAVES
        MOV      #ABL CNT   ;SET AUTO BLOCK COUNT
        MOV      #-4000,FCNT ;SET FC = 4000
        MOV      #100,RCNT  ;SET REC CNTR TO 100
        MOV      #10,PATRN  ;SELECT PATTERN 10
        JSR     PC,STAUTO   ;GO DO AUTO SEQ
        MOV      #14,PATRN  ;SELECT PATTERN 14
        JSR     PC,STAUTO
        MOV      #15,PATRN  ;SELECT PATTERN 15
        JSR     PC,STAUTO
        MOV      #-1,ABL CNT ;FORCE TO END OF TAPE
        MOV      #-1,PATRN  ;SELECT AUTO RANDOM DATA
        JSR     PC,STAUTO
3$:     RTS      PC        ;RETURN TO SEQ

```

```

4591
4592
4593
4594
4595
4596
4597
4598
4599
4600
4601
4602
4603
4604
4605
4606
4607 022570 012704 024546
4608 022574 000004
4609 022576 013703 000550
4610 022602 104400
4611 022604 012704 024532
4612 022610 000004
4613 22612 013703 000552
4614 022616 042703 177770
4615 022622 104400
4616 022624 012704 026105
4617 022630 000004
4618 022632 013703 000552
4619 022636 000303
4620 022640 042703 177770
4621 022644 104400
4622 022646 012704 026111
4623 022652 000004
4624 022654 005003
4625 022656 032737 000010 000552
4626 022664 001402
4627 022666 012703 000001
4628 022672 104400
4629 022674 012704 026115
4630 022700 000004
4631 022702 013703 000552
4632 022706 000241
4633 022710 006003
4634 022712 006003
4635 022714 006003
4636 022716 006003
4637 022720 042703 177760
4638 022724 104400
4639 022726 012704 024507
4640 022732 000004
4641 022734 032777 000400 155646
4642 022742 001406
4643 022744 012737 000122 000640
4644 022752 004737 023744
4645 022756 000411
4646 022760 005737 000736

```

```

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

```

```

PAPRT:  MOV    #MSG12,R4
        TYPE                   ;TYPE MSG
        MOV    DVN,R3
        TYPOCT                  ;PRINT DRIVE NUMBER
        MOV    #MSG11,R4
        TYPE                   ;TYPE MSG
        MOV    UDES,R3
        BIC    #177770,R3
        TYPOCT                  ;PRINT UNIT NUMBER
        MOV    #MSG60,R4
        TYPE                   ;TYPE MSG
        MOV    UDES,R3
        SWAB   R3
        BIC    #177770,R3
        TYPOCT                  ;PRINT DENSITY
        MOV    #MSG61,R4
        TYPE                   ;TYPE MSG
        CLR    R3
        BIT    #10,UDES
        BEQ    PAPRT0
        MOV    #1,R3
        PAPRT0:  TYPOCT          ;PRINT PARITY
        MOV    #MSG62,R4
        TYPE                   ;TYPE MSG
        MOV    UDES,R3
        CLC
        ROR    R3
        ROR    R3
        ROR    R3
        ROR    R3
        BIC    #177760,R3
        TYPOCT                  ;PRINT FORMAT
        MOV    #MSG8,R4
        TYPE                   ;TYPE MSG
        BIT    #400,JSWR
        BEQ    PAPRTB
        BEQ    #122,TOB
        JSR    PC,TOG
        BR     PAPRTD
        TST   ASEQF
        ;SEE IF AUTO SEQ

```





4685  
4686  
4687  
4688  
4689  
4690  
4691  
4692  
4693  
4694  
4695  
4696  
4697  
4698  
4699  
4700  
4701  
4702  
4703  
4704  
4705  
4706  
4707  
4708  
4709  
4710  
4711  
4712  
4713  
4714  
4715  
4716  
4717  
4718  
4719  
4720  
4721  
4722  
4723  
4724  
4725  
4726  
4727  
4728  
4729  
4730  
4731  
4732  
4733  
4734  
4735  
4736  
4737  
4738  
4739  
4740

023152 063737 000630 000626  
023160 063737 000626 000630  
023166 023701 000630  
023172 101367  
023174 020237 000630  
023200 101364  
023202 000207  
  
023204 022737 000176 000610  
023212 001027  
023214 004737 023274  
023220 012704 024424  
023224 000004  
023226 017703 155356  
023232 104400  
023234 012704 024434  
023240 000004  
023242 013705 000610  
023246 012701 000007  
023252 012702 177777  
023256 012703 000000  
023262 004737 023340  
023266 004737 023316  
023272 000207  
  
010546  
010446  
010346  
010246  
010146  
010046  
016646 000014  
000207  
  
012666 000014  
012600  
012601  
012602  
012603  
012604  
012605  
000207

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

RANG: ADD RANSV, RANBAS ; GET NEW NUMBER  
ADD RANBAS, RANSV ; SEE IF NUMBER TOO BIG  
CMP RANSV, R1 ; IF SO: BR  
BHI RANG ; SEE IF NUMBER TOO SMALL  
CMP R2, RANSV ; IF SO: BR  
BHI RANG ; EXIT  
RTS PC

;SUBROUTINE TO GET NEW SOFTWARE SWR

GTSWR: CMP #SWREG, SWR ; BRANCH IF SOFTWARE SWR  
BNE IS ; NOT INVOKED  
JSR PC, .SAVE ; SAVE REGISTERS ON THE STACK  
MOV #SMSWR, R4 ; TYPE 'SWR = '  
TYPE ; TYPE MSG  
MOV #SWR, R3 ; GET CURRENT SWR  
TYPOCT  
MOV #SMNEW, R4 ; ASK FOR NEW SETTING  
TYPE ; TYPE MSG  
MOV SWR, R5 ; TTR ROUTINE RETURNS VALUE TO (R5)  
MOV #7, R1 ; LIMIT RESPONSE TO 7 CHARS  
MOV #177777, R2 ; BETWEEN 0 AND 177777  
MOV #0, R3  
JSR PC, TTR ; GET RESPONSE  
JSR PC, .RESTORE ; RESTORE REGISTERS  
IS: RTS PC ; RETURN

;ROUTINE TO SAVE REGISTERS ON THE STACK

.SAVE: MOV %5, -(SP) ; R5 IS SAVED AT 12(SP)  
MOV %4, -(SP) ; R4 IS SAVED AT 10(SP)  
MOV %3, -(SP) ; R3 IS SAVED AT 6(SP)  
MOV %2, -(SP) ; R2 IS SAVED AT 4(SP)  
MOV %1, -(SP) ; R1 IS SAVED AT 2(SP)  
MOV %0, -(SP) ; R0 IS SAVED AT (SP)  
MOV 14(SP), -(SP) ; PUSH RETURN PC ON THE STACK  
RTS PC ; RETURN TO CALLER

;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK

.RESTORE: MOV (SP)+, 14(SP) ; STORE RETURN PC ON STACK  
MOV (SP)+, %0  
MOV (SP)+, %1  
MOV (SP)+, %2  
MOV (SP)+, %3  
MOV (SP)+, %4  
MOV (SP)+, %5  
RTS PC ; RETURN

4741  
4742  
4743  
4744  
4745  
4746  
4747  
4748  
4749  
4750  
4751  
4752  
4753  
4754  
4755  
4756  
4757  
4758 023340 010146  
4759 023342 011601  
4760 023344 005037 000644  
4761 023350 005000  
4762 023352 004737 023572  
4763 023356 122737 000003 000642  
4764 023364 001003  
4765 023366 000005  
4766 023370 000137 000200  
4767 023374 122737 000015 000642 11\$:  
4768 023402 001004  
4769 023404 005737 000644  
4770 023410 001457  
4771 023412 000451  
4772 023414 122737 000025 000642 2\$:  
4773 023422 001004  
4774 023424 012704 025117  
4775 023430 000004  
4776 023432 000743  
4777 023434 122737 000177 000642 21\$:  
4778 023442 001011  
4779 023444 000241  
4780 023446 006000  
4781 023450 006200  
4782 023452 006200  
4783 023454 012704 027230  
4784 023460 000004  
4785 023462 005201  
4786 023464 000732  
4787 023466 122737 000060 000642 3\$:  
4788 023474 101027  
4789 023476 122737 000070 000642 4\$:  
4790 023504 101423  
4791 023506 005237 000644 5\$:  
4792 023512 006300  
4793 023514 006300  
4794 023516 006300  
4795 023520 042737 177770 000642  
4796 023526 053700 000642

```

*****
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
*****
TTR: MOV R1, -(SP) ;SAVE CHAR COUNT
10$: MOV (SP), R1 ;RESTORE CHAR COUNT (FOR ↑U)
CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
CLR RO
1$: JSR PC, TTIN ;GO READ CHARACTER
CMPB #3, TIB ;BRANCH IF NOT ↑C
BNE 11$
RESET
JMP #200 ;RESTART AT 200
4767: CMPB #15, TIB ;SEE IF CR
BNE 2$ ;IF NOT: BR
TST TEMP1 ;SEE IF FIRST CHARACTER
BEQ 9$ ;IF SO: BR
BR 6$ ;ELSE GO LOAD VALUE
4772: CMPB #25, TIB ;BRANCH IF NOT CONTROL U
BNE 21$
MOV #MSG28, R4 ;TYPE (CR)(LF)
TYPE ;TYPE MSG
BR 10$
4777: CMPB #177, TIB ;BRANCH IF NOT 'RUBOUT'
BNE 3$
CLC ;REMOVE LAST CHARACTER
ROR RO
ASR RO
ASR RO
4783: MOV #MSG118, R4 ;TYPE '\ '
TYPE ;TYPE MSG
INC R1 ;DEC CHAR RECEIVED COUNT
BR 1$ ;GET NEXT CHARACTER
4787: CMPB #60, TIB ;SEE IF CHAR IS LESS THAN 0
BHI TIBR
4789: CMPB #70, TIB ;SEE IF CHAR IS GREATER THAN 7
BLOS TIBR
4791: INC TEMP1 ;SET FIRST CHARACTER FLAG
ASL RO
ASL RO ;SHIFT 3 LEFT
ASL RO
BIC #177770, TIB ;STRIP ASCII
BIS TIB, RO ;LOAD CHARACTER

```

4797	023532	005301		DEC	R1	: SEE IE DONE
4798	023534	001306		BNE	18	: IF NOT: BR
4799	023536	020002		6S: CMP	R0, R2	: SEE IF EXCEEDED MAXIMUM LIMIT
4800	023540	101005		BHI	TIMER	
4801	023542	020300		7S: CMP	R3, R0	: SEE IF BELOW MINIMUM LIMIT
4802	023544	101003		BHI	TIMER	
4803	023546	010015		8S: MOV	R0, (R5)	: LOAD VALUE
4804	023550	005726		9S: TST	(SP)+	: POP CHAR COUNT OFF STACK
4805	023552	000207		RTS	PC	: EXIT
4806						
4807	023554	012704	025617	TIMER: MOV	#MSG43, R4	
4808	023560	000004		TYPE		: TYPE MSG
4809	023562	005726		TST	(SP)+	: POP CHAR COUNT OFF STACK
4810	023564	162716	000020	SUB	#20, (SP)	: RESET SP TO START OF VALUE ROUTINE
4811	023570	000207		RTS	PC	: REDO VALUE ENTRY

```

4812
4813
4814
4815 023572 005277 155014
4816 023576 105777 155010
4817 023602 100375
4818 023604 017737 155004 000642
4819 023612 042737 000200 000642
4820 023620 013737 000642 000640
4821 023626 004737 0' 44
4822 023632 000207
4823
4824
4825
4826 023634 112437 000640
4827 023640 105737 000640
4828 023644 001436
4829 023646 122737 000045 000640
4830 023654 001407
4831 023656 122737 000041 000640
4832 023664 001436
4833 023666 004737 023744
4834 023672 000760
4835 023674 112737 000015 000640
4836 023702 004737 023744
4837 023706 012703 000006
4838 023712 005037 000640
4839 023716 004737 023744
4840 023722 005303
4841 023724 001372
4842 023726 112737 000012 000640
4843 023734 004737 023744
4844 023740 000735
4845 023742 000002
4846
4847 023744 105777 154646
4848 023750 100375
4849 023752 113777 000640 154640
4850 023760 000207
4851
4852 023762 012703 000002
4853 023766 012737 000007 000640
4854 023774 004737 023744
4855 024000 005303
4856 024002 001371
4857 024004 000713
4858
4859

; TTY READ SUBROUTINE*****
TTIN: INC @TKS
1$: TSTB @TKS
BPL 1$
MOV @TKB,TIB
BIC #200,TIB ;STRIP PARITY BIT
MOV TIB,TOB ;MOVE CHAR TO TTY OUTPUT BFR
JSR PC,TOG ;ECHO CHARACTER
RTS PC

; TTY OUTPUT SUBROUTINE*****
TTOUT: MOVB (R4)+,TOB
TSTB TOB
BEQ 3$
CMPB #45,TOB
BEQ 1$
CMPB #41,TOB
BEQ TBELL ;DO BELL
JSR PC,TOG
BR TTOUT
1$: MOVB #15,TOB
JSR PC,TOG
MOV #6,R3
2$: CLR TOB
JSR PC,TOG
DEC R3
BNE 2$ ;DO FILLERS
MOVB #12,TOB
JSR PC,TOG
BR TTOUT
3$: RTI ;RETURN

TOG: TSTB @TPS
BPL TOG
MOVB TOB,@TPB
RTS PC ;RETURN

TBELL: MOV #2,R3
1$: MOV #7,TOB
JSR PC,TOG
DEC R3
BNE 1$
BR TTOUT

```

```

;OCTAL OUTPUT SUBROUTINE*****
4860
4861
4862 024006 005037 024220 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
4863 024012 010304 MOV R3,R4 ;SEE IF NUMBER IS ZERO
4864 024014 001003 BNE OCTPO ;IF NOT ZERO: BR
4865 024016 004737 024200 JSR PC,OCTPG1 ;ELSE PRINT ZERO
4866 024022 000447 BR OCTP3 ;SPACE AND EXIT
4867 024024 005704 OCTPO: TST R4 ;BRANCH IF MSD = 0
4868 024026 100006 BPL OCTP1
4869 024030 012704 000001 MOV #1,R4
4870 024034 004737 024156 JSR PC,OCTPG ;PRINT 1
4871 024040 000137 024052 JMP OCTP2
4872 024044 005004 OCTP1: CLR R4 ;PRINT 0
4873 024046 004737 024156 JSR PC,OCTPG
4874 024052 010304 OCTP2: MOV R3,R4
4875 024054 006004 ROR R4
4876 024056 006004 ROR R4
4877 024060 006004 ROR R4 ;POSITION DIGIT
4878 024062 006004 ROR R4
4879 024064 000304 SWAB R4
4880 024066 004737 024156 JSR PC,OCTPG ;PRINT DIGIT 2
4881 024072 010304 MOV R3,R4
4882 024074 006004 ROR R4
4883 024076 000304 SWAB R4
4884 024100 004737 024156 JSR PC,OCTPG ;PRINT DIGIT 3
4885 024104 010304 MOV R3,R4
4886 024106 006104 ROL R4
4887 024110 006104 ROL R4
4888 024112 000304 SWAB R4
4889 024114 004737 024156 JSR PC,OCTPG ;PRINT DIGIT 4
4890 024120 010304 MOV R3,R4
4891 024122 006004 ROR R4
4892 024124 006004 ROR R4
4893 024126 006004 ROR R4
4894 024130 004737 024156 JSR PC,OCTPG
4895 024134 010304 MOV R3,R4
4896 024136 004737 024156 JSR PC,OCTPG ;PRINT DIGIT 5
4897 024142 012737 000240 000640 OCTP3: MOV #240,T08
4898 024150 004737 023744 JSR PC,T08 ;PRINT SPACE
4899 024154 000002 RTI ;EXIT
4900 024156 042704 177770 OCTPG: BIC #177770,R4
4901 024162 001004 BNE OCTPG0
4902 024164 005737 024220 TST OFL
4903 024170 001001 BNE OCTPG0
4904 024172 000207 RTS PC
4905
4906 024174 005237 024220 OCTPG0: INC OFL
4907 024200 052704 000260 OCTPG1: BIS #260,R4
4908 024204 010437 000640 MOV R4,T08
4909 024210 004737 023744 JSR PC,T08
4910 024214 010304 MOV R3,R4
4911 024216 000207 RTS PC
4912 024220 000000 OFL: 0 ;FIRST CHAR FLAG
4913
  
```

```

4914
4915 ;DATA CHARACTER OUTPUT SUBROUTINE*****
4916
4917 024222 012704 000010 DOUT: MOV #10,R4 ;SET NUMBER TO PRINT
4918 024226 110337 000640 MOVB R3,T08
4919 024232 105777 154360 1$: TSTB #T08
4920 024236 100375 BPL 1$
4921 024240 105737 000640 TSTB T08
4922 024244 100004 BPL 2$
4923 024246 012777 000061 154344 MOV #061,#T08
4924 024254 000403 BR 3$
4925 024256 012777 000060 154334 2$: MOV #060,#T08
4926 024264 006337 000640 3$: ASL T08
4927 024270 005304 DEC R4
4928 024272 001357 BNE 1$
4929 024274 000207 RTS PC
4930
4931 024276 013703 000650 DOUTD: MOV TEMP3,R3
4932 024302 000303 SWAB R3
4933 024304 004737 024222 JSR PC,DOUT
4934 024310 013703 000650 MOV TEMP3,R3
4935 024314 004737 024222 JSR PC,DOUT
4936 024320 000207 RTS PC
4937
4938 ;TU16 SERIAL NUMBER PRINT SUBROUTINE*****
4939
4940 024322 010304 SNPT: MOV R3,R4
4941 024324 000304 SWAB R4
4942 024326 006004 ROR R4
4943 024330 006004 ROR R4
4944 024332 006004 ROR R4
4945 024334 006004 ROR R4
4946 024336 004737 024400 JSR PC,SNPG ;PRINT FIRST DIGIT
4947 024342 010304 MOV R3,R4
4948 024344 000304 SWAB R4
4949 024346 004737 024400 JSR PC,SNPG ;PRINT SECOND DIGIT
4950 024352 010304 MOV R3,R4
4951 024354 006004 ROR R4
4952 024356 006004 ROR R4
4953 024360 006004 ROR R4
4954 024362 006004 ROR R4
4955 024364 004737 024400 JSR PC,SNPG ;PRINT THIRD DIGIT
4956 024370 010304 MOV R3,R4
4957 024372 004737 024400 JSR PC,SNPG ;PRINT FOURTH DIGIT
4958 024376 000207 RTS PC ;EXIT
4959 024400 012737 0002E0 000640 SNPG: MOV #260,T08 ;SET NUMBER BASE
4960 024406 042704 177760 BIC #177760,R4 ;MASK NUMBER
4961 024412 050437 000640 BIS R4,T08 ;BUILD DIGIT
4962 024416 004737 023744 JSR PC,T08 ;GO TYPE
4963 024422 000207 RTS PC ;RETURN
4964

```

```

4965
4966 ;ERROR MESSAGES*****
4967
4968 024424 051445 051127 036440 SMSWR: .ASCIZ /%SWR = /
4969 024432 000040
4970 024434 047040 093505 036440 SMNEW: .ASCIZ / NEW = /
4971 024442 000040
4972 024444 042052 020105 000 MSG1: .ASCIZ /%DE /
4973 024451 045 035507 000040 MSG2: .ASCIZ /%G; /
4974 024456 041045 020073 000 MSG3: .ASCIZ /%B; /
4975 024463 045 047103 000040 MSG4: .ASCIZ /%CN /
4976 024470 053452 020105 000 MSG5: .ASCIZ /%HE /
4977 024475 052 042522 000040 MSG6: .ASCIZ /%RE /
4978 024502 051052 020123 000 MSG7: .ASCIZ /%RS /
4979 024507 052 040520 051124 MSG8: .ASCIZ /%PATRN /
4980 024514 020116 000
4981 024517 045 047123 020072 MSG9: .ASCIZ /%SN: /
4982 024524 000
4983 024525 052 042523 000040 MSG10: .ASCIZ /%SE /
4984 024532 051452 040514 042526 MSG11: .ASCIZ /%SLAVE NO. /
4985 024540 047040 027117 000040
4986 024546 022445 042045 044522 MSG12: .ASCIZ /%:DRIVE NO. /
4987 024554 042526 047040 027117
4988 024562 000040
4989 024564 025045 047102 000040 MSG13: .ASCIZ /%*BN /
4990 024572 051052 020116 000 MSG14: .ASCIZ /%*RN /
4991 024577 045 020041 020040 MSG15: .ASCIZ /%! BAD RECORD%/
4992 024604 020040 020040 020040
4993 024612 041040 042101 051040
4994 024620 041505 051117 022504
4995 024626 000045
4996 024630 043040 000 MSG16: .ASCIZ / F/
4997 024633 040 000122 MSG17: .ASCIZ / R/
4998 024636 020041 047505 020124 MSG20: .ASCIZ /! EOT NO: /
4999 024644 047516 020072 000
5000
5001 024651 045 047111 042524 MSG21: .ASCIZ /%INTERCHANGE READ = /
5002 024656 041522 040510 043516
5003 024664 020105 042522 042101
5004 024672 036440 000040
5005 024676 020445 046111 042514 MSG22: .ASCIZ /%!ILLEGAL BOT: HALT%%/
5006 024704 040507 020114 047502
5007 024712 035124 044040 046101
5008 024720 022524 022445 000
5009 024725 045 051503 020061 MSG23: .ASCIZ /%CS1 /
5010 024732 000
5011 024733 045 041527 000040 MSG23A: .ASCIZ /%*C /
5012 024740 041045 020101 000 MSG23B: .ASCIZ /%*BA /
5013 024745 045 041506 000040 MSG23C: .ASCIZ /%*FC /
5014 024752 041445 031123 000040 MSG23D: .ASCIZ /%*CS2 /
5015 024760 042045 020123 000 MSG23E: .ASCIZ /%*DS /
5016 024765 045 051105 000040 MSG23F: .ASCIZ /%*ER /
5017 024772 040445 020123 000 MSG23G: .ASCIZ /%*AS /
5018 024777 045 045503 000040 MSG23H: .ASCIZ /%*CK /
5019 025004 042045 020102 000 MSG23I: .ASCIZ /%*DB /
5020 025011 045 051115 000040 MSG23J: .ASCIZ /%*MR /

```



5021	025016	042045	020124	000	MSG3K: .ASCIZ	/%DT /
5022	025023	045	041524	000040	MSG23L: .ASCIZ	/%TC /
5023	025030	051445	020116	000	MSG23M: .ASCIZ	/%SN /
5024	025035	045	047041	020117	MSG24: .ASCIZ	/%!NO INTERRUPT%/
5025	025042	047111	042524	051122		
5026	025050	050125	022524	000		
5027	025055	045	047041	020117	MSG25: .ASCIZ	/%!NO MOL: HALT%/
5028	025062	047515	035114	044040		
5029	025070	046101	022524	000		
5030	025075	045	051104	050117	MSG26: .ASCIZ	/%DROPS: /
5031	025102	035123	000040			
5032	025106	050045	041511	051513	MSG27: .ASCIZ	/%PICKS: /
5033	025114	020072	000			
5034	025117	045	000		MSG28: .ASCIZ	/%/
5035	025121	045	052045	030505	MSG30: .ASCIZ	/%%TE16 AUTO SEQUENCE (DZTED-A)%/
5036	025126	020066	052501	047524		
5037	025134	051440	050505	042525		
5038	025142	041516	020105	042050		
5039	025150	052122	042105	040455		
5040	025156	022451	000			
5041	025161	045	052045	030115	MSG31: .ASCIZ	/%%TM03-TE16 DATA RELIABILITY TEST (DZTED-A)%/
5042	025166	026463	042524	033061		
5043	025174	042040	052101	020101		
5044	025202	042522	044514	041101		
5045	025210	046111	052111	020131		
5046	025216	042524	052123	024040		
5047	025224	055104	042524	026504		
5048	025232	024501	000045			
5049	025236	054524	042520	036040	MSG31A: .ASCIZ	/%TYPE <CR> TO TERMINATE ALL REQUESTS & tC TO RESTART%/
5050	025244	051103	020076	047524		
5051	025252	052040	051105	044515		
5052	025260	040516	042524	040440		
5053	025266	046114	051040	050505		
5054	025274	042525	052123	020123		
5055	025302	020046	041536	052740		
5056	025310	020117	042522	052123		
5057	025316	051101	022524	000		
5058	025323	045	046123	053101	MSG32: .ASCIZ	/%SLAVE NUMBER = /
5059	025330	020105	052516	041115		
5060	025336	051105	036440	000040		
5061	025344	042045	047105	044523	MSG33: .ASCIZ	/%DENSITY = /
5062	025352	054524	036440	000040		
5063	025360	050045	051101	052111	MSG34: .ASCIZ	/%PARITY = /
5064	025366	020131	020075	000		
5065	025373	045	042522	047503	MSG35: .ASCIZ	/%RECORD COUNT = /
5066	025400	042122	041440	052517		
5067	025406	052116	036440	000040		
5068	025414	041445	040510	020122	MSG36: .ASCIZ	/%CHAR COUNT = /
5069	025422	047503	047125	020124		
5070	025430	020075	000			
5071	025433	045	040520	052124	MSG37: .ASCIZ	/%PATTERN NUMBER = /
5072	025440	051105	020116	052516		
5073	025446	041115	051105	036440		
5074	025454	000040				
5075	025456	051445	047111	046107	MSG38: .ASCIZ	/%SINGLE PASS = /
5076	025464	020105	040520	051523		

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

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

5189	026542	025052	025052	025052	
5190	026550	025052	025052	000052	
5191	026556	052052	030115	020063	MSG102: .ASCIZ /*TMO3 /
5192	026564	000			
5193	026565	052	046123	053101	MSG103: .ASCIZ /*SLAVES /
5194	026572	051505	000040		
5195	026576	040445	052125	020117	MSG104: .ASCIZ /*AUTO CONT: /
5196	026604	047503	052116	020072	
5197	026612	000			
5198	026613	045	042522	047503	MSG105: .ASCIZ /*RECOVERED/
5199	026620	042526	042522	000104	
5200	026626	020452	041041	042101	MSG106: .ASCIZ /*!!BAD TAPE OVERFLOW/
5201	026634	052040	050101	020105	
5202	026642	053117	051105	046106	
5203	026650	053517	000		
5204	026653	045	042522	044527	MSG16A: .ASCIZ /*REWIND TAPE; RESTART AT BLOCK 1/
5205	026660	042116	052040	050101	
5206	026666	035505	051040	051505	
5207	026674	040524	052122	040440	
5208	026702	020124	046102	041517	
5209	026710	020113	000061		
5210	026714	020445	052441	051116	MSG107: .ASCIZ /*!!UNRECOVERABLE BAD SPOT/
5211	026722	041505	053117	051105	
5212	026730	041101	042514	041040	
5213	026736	042101	051440	047520	
5214	026744	020124			
5215	026746	041045	042101	051040	.ASCIZ /*BAD RECORD LEFT ON TAPE%/
5216	026754	041505	051117	020104	
5217	026762	042514	052106	047440	
5218	026770	020116	040524	042520	
5219	026776	000045			
5220	027000	020452	050041	051517	MSG109: .ASCIZ /*!!POSITION LOST IN RETRY/
5221	027006	052111	047511	020116	
5222	027014	047514	052123	044440	
5223	027022	020116	042522	051124	
5224	027030	000131			
5225	027032	051445	051525	042520	MSG110: .ASCIZ /*SUSPECT BAD TAPE/
5226	027040	052103	041040	042101	
5227	027046	052040	050101	000105	
5228	027054	051045	050105	040505	MSG111: .ASCIZ /*REPEAT: /
5229	027062	035124	000040		
5230	027066	041040	042101	052040	MSG112: .ASCIZ / BAD TAPE SPOTS%/
5231	027074	050101	020105	050123	
5232	027102	052117	022523	000	
5233					
5234	027107	045	051440	043117	MSG113: .ASCIZ /* SOFT: /
5235	027114	035124	000040		
5236					
5237	027120	020045	040510	042122	MSG114: .ASCIZ /* HARD: /
5238	027126	020072	000		
5239					
5240	027131	045	020441	040510	MSG115: .ASCIZ /*!!HARD READ ERROR/
5241	027136	042122	051040	040505	
5242	027144	020104	051105	047522	
5243	027152	000122			
5244	027154	020445	047125	052111	MSG116: .ASCIZ /*!UNIT IS REWINDING: TEST WILL START AT BOT/

5245	027162	044440	020123	042522
5246	027170	044527	042116	047111
5247	027176	035107	052040	051505
5248	027204	020124	044527	046114
5249	027212	051440	040524	052122
5250	027220	040440	020124	047502
5251	027226	000124		
5252	027230	000134		
5253	027232	051045	046505	053117
5254	027240	020105	046524	050104
5255	027246	043040	047522	020115
5256	027254	046123	053101	020105
5257	027262	047524	041040	020105
5258	027270	042524	052123	042105
5259	027276	00J045		
5260	027300	044045	051101	053504
5261	027306	051101	020105	053523
5262	027314	020122	047111	052440
5263	027322	042523	000045	
5264				
5265				
5266	027326	000000		
5267				
5268		033334		
5269	033334	000000		
5270				
5271		000001		

MSG118: .ASCIZ /\/  
MSG120: .ASCIZ /%REMOVE TMDP FROM SLAVE TO BE TESTED%/

MSG121: .ASCIZ /%HARDWARE SWR IN USE%/

WDATA: .EVEN ;WRITE BUFFER  
0  
RDATA: .=.+4004 ;READ BUFFER  
0  
.END

ABL CNT	000742	BT PRT1	007376	DAT0A	014320	DF0C0	015514	EOTREC	000662
ACTLRC	021012	BTPT	000732	DAT0B	014326	DF00	015470	ER	000524
ADRVN	000740	BTSTF	000730	DAT0C	014372	DF0E	015462	ERCHK	017236
AM001	022274	BTUR	007402	DAT0D	014400	DF0F	015454	ERPT	020056
AM002	022434	BT00	001610	DAT0E	014410	DF1	015702	ERPTG	020112
APATS	014740	BT01	001714	DAT0F	014424	DF2	015712	ERPTG1	020160
AS	000526	BT02	002020	DAT1	014434	DF3	015730	ERPTT	020076
ASEQ	021734	BT03	002124	DAT1A	014440	DF4	015772	ERPTO	020174
ASEQCF	000744	BT04	002230	DAT10	014556	DOUT	024222	ERPT1	020242
ASEQF	000736	BT05	002334	DAT11	014606	DOUTD	024276	ERPT2	020300
ASEQX	022072	BT06	002440	DAT12	014626	DPC	016624	ERPT3	020336
ASEQXX	022106	BT07	002544	DAT13	014650	DPCG	016632	ERPT4	020400
ASEQ0	021766	B0	012026	DAT14	014660	DPC0	016640	ERPT5	020422
ASEQ1	021772	B1	012072	DAT15	014710	DPC0A	016702	ERPT5A	020504
ASEQ2	022032	B2	012102	DAT2	014454	DPC1	016710	ERPT6	020546
ASEQ3	022044	CADER	020774	DAT3	014460	DPC1A	016736	ERPT7	020616
ASEQ4	022054	CC	000530	DAT3A	014466	DPC2	017000	ERPX	020652
BA	000514	CCNTR	012126	DAT4	014504	DPC2A	016742	ERPX0	020620
BAER	020776	CHNFLG	003040	DAT5	014514	DPC2B	016764	ERPX1	020740
BBC	000660	CLLAST	015122	DAT6	014522	DPC3	017032	ERPX2	020772
BCNT	000712	CLP	015232	DAT7	014530	DPPRT	017100	ERSAV	000724
30PP	000720	CLPE	015256	DB	000532	DPPRTX	017234	ERTFL	000734
B000	001410	CLP2	015314	DCHK	015366	DPPRT0	017150	ERO	017252
B010	001430	CLP3	015326	DCHKO	015414	DPPRT1	017174	EROA	017320
B020	001450	CLO	015042	DEREV1	001170	DPPRT2	017210	EROB	017266
B030	001470	CL1	015070	DEREX	016450	DROP	016614	ER1	017326
B040	001510	CL2	015112	DEREX1	016502	DRPK	016602	ER10	020020
B050	001530	CL3	015174	DERFL	000706	DRPKF	016514	ER2	017332
B060	001550	CONER	021000	DERR	016000	DRP1	001010	ER2A	017376
B070	001570	CRCC	000566	DERR0	016010	DRP2	001012	ER2A0	017346
BKRT	012046	CR CER	021010	DERR0A	016040	DRP3	001014	ER2A1	017366
BKSP	011672	CRCLRC	015024	DERR0B	016072	DRP4	001016	ER2B	017412
BKTM	011754	CRCSV	021016	DERR0C	016116	DRP5	001020	ER2C	017436
BKTH0	012016	CS	000520	DERR0D	016120	DRP6	001022	ER2D	017452
BLCNTR	000656	C1	000510	DERR1	016146	DRP7	001024	ER2E	017500
BPKP	000722	DAT0	002772	DERR2	016150	DRP8	001026	ER3	017506
BP00	001210	DAT01	002774	DERR3	016164	DRVER	021002	ER3A	017544
BP10	001230	DAT010	003012	DERR4	016166	DS	000522	ER3A1	017602
BP20	001250	DAT011	003014	DERR4A	016326	DSUP	014110	ER3B	017610
BP30	001270	DAT012	003016	DERR4B	016374	DS0	014116	ER4	017614
BP40	001310	DAT013	003020	DERR5	016432	DS0A	014200	ER4A	017650
BP50	001330	DAT014	003022	DERR6	016444	DS0B	014176	ER4A1	017652
BP60	001350	DAT015	003024	DFX	015776	DS0C	014140	ER6	017664
BP70	001370	DAT02	002776	DF0	015670	DS2A	014226	ER6A	017750
BTADDR	001030	DAT03	003000	DF0A	015564	DS3	014232	ER7	020002
BTFLG	000726	DAT04	003002	DF0A0	015606	DS4	014244	FXCRC	015362
BTOV	007166	DAT05	003004	DF0A1	015622	DT	000536	EXLRC	015364
BTOVX	007324	DAT06	003006	DF0A2	015636	DVN	000550	FC	000516
BTOV0	007204	DAT07	003010	DF0A3	015652	DOFL	014432	FCER	021004
BTOV1	007214	DATBL	002770	DF0A4	015656	EMADDR	000654	FCSAV	000634
BTOV2	007300	DATER1	001130	DF0B	015524	ENDFLG	000736	FMCNT	000556
BTJY3	007316	DATR	014760	DF0B0	015546	ENDTBL	002770	FRPRT	021020
BT PRT	007326	DAT0	014274	DF0C	015504	EOTCO	002650	GTSWR	023204

MOFL	020456	MSG21	025023	MSG73	026303	RANSAY	000630	ROS	010472
HE	01042	MSG22	025030	MSG74	026314	RANSET	004276	RO6	010514
HRUS	01110	MSG23	025035	MSG75	026325	RCNT	000554	RO7	010540
INIT	010436	MSG24	025055	MSG76	026330	RCNTR	012166	RO7A	010570
INTRF	014570	MSG25	025075	MSG77	026363	RCSAV	000632	READ	007730
LRCER	021006	MSG26	025106	MSG78	026374	ROA	007770	REGS	000544
LRCV	021014	MSG27	025117	MSG79	026405	RDATA	033334	REOT	004330
NR	000534	MSG28	024456	MSG8	026446	RDCMD	000562	REOTC	005054
MSG1	024444	MSG29	025121	MSG88	024507	REERR1	001150	REOTX	004760
MSG10	024525	MSG30	025161	MSG9	024517	REER1	001110	REOTXX	005050
MSG100	025506	MSG31	025236	MT	000674	REER2	001112	REOT1A	004376
MSG101	02557	MSG31A	025236	MT T	021716	REER3	001114	REOT1B	004424
MSG102	025576	MSG32	025323	MTINTA	021720	REER4	001116	REOT1C	004444
MSG103	02565	MSG33	025344	NOP	000240	REER5	001120	REOT1E	004470
MSG104	02576	MSG34	025360	NRTP	011236	REER6	001122	REOT1F	004440
MSG105	025613	MSG35	025373	NRZOF	000652	REER7	001124	REOT2	004512
MSG106	025626	MSG36	025414	OCTP	024036	REER8	001126	REOT2A	004550
MSG107	025714	MSG37	025433	OCTPG	024156	RDEX	010656	REOT3	004576
MSG109	027000	MSG38	025456	OCTPG0	024174	ROFL	015022	REOT4	004626
MSG11	024532	MSG39	025476	OCTPG1	024200	ROFG	010752	REOT5	004642
MSG110	027032	MSG4	024463	OCTP0	024024	ROTX	011234	REOT6	004706
MSG111	027054	MSG40	025536	OCTP1	024044	ROTY	010664	REOT7	004736
MSG112	027066	MSG41	025565	OCTP2	024052	RORT0	010676	RETRY	000604
MSG113	027107	MSG42	025577	OCTP3	024142	RORT1	010730	RFHARD	002730
MSG114	027120	MSG43	025617	OFL	024220	RORT1A	010726	RFSOFT	002670
MSG115	027131	MSG44	025623	PAPRT	022570	RORT1B	010746	RPCNT	000702
MSG116	027154	MSG45	025652	PAPRTA	022744	RORT2	011030	RRHARD	002750
MSG118	027230	MSG46	025664	PAPRTB	022760	RORT3	011052	RRSOFT	002710
MSG12	024546	MSG47	025671	PAPRTC	022774	RORT4	011056	RSEQ	007416
MSG120	027232	MSG48	025676	PAPRTD	023002	RORT5	011060	RSEX	007552
MSG121	027300	MSG49	025731	PAPRTY	023076	RORT5A	011124	RSF	007470
MSG13	024564	MSG5	024470	PAPRTD	022672	RORT5B	011134	RSFR	007566
MSG14	024572	MSG50	025745	PAPRTD	023046	RORT6	011164	RSFRX	007726
MSG15	024577	MSG52	025772	PAPRT1	023100	RORT7	011230	RSFR0	007620
MSG16	024630	MSG53	026013	PAPRT2	023100	ROX	010662	RSFR1	007632
MSG16A	026653	MSG54	026026	PAPRT3	023102	ROO	010026	RSFR2	007664
MSG17	024633	MSG55	026035	PARS	014272	RO1	010074	RSFO	007524
MSG2	024451	MSG56	026044	PATRN	000560	RO1A	010110	RSF1	007540
MSG20	024636	MSG57	026050	PATS	014270	RO1B	010116	RSR	007446
MSG21	024651	MSG58	026071	PFLG	000672	RO1D	010124	RSTAL	000574
MSG22	024676	MSG59	026077	PICK	017034	RO1D	010602	RTCNT	000704
MSG23	024725	MSG6	024475	PIK1	000770	RO11	010612	RTRN	000664
MSG23A	024733	MSG60	026105	PIK2	000772	RO2	010130	RTYFL	000714
MSG23B	024740	MSG61	026111	PIK3	000774	RO3	010170	RTY1	001050
MSG23C	024745	MSG62	026115	PIK4	000776	RO4	010220	RTY2	001052
MSG23D	024752	MSG64	026121	PIK5	001000	RO4A	010316	RTY3	001054
MSG23E	024760	MSG65	026143	PIK6	001002	RO4A0	010350	RTY4	001056
MSG23F	024765	MSG66	026154	PIK7	001004	RO4A1	010372	RTY5	001060
MSG23G	024772	MSG67	026167	PIK8	001006	RO4A2	010414	RTY6	001062
MSG23H	024777	MSG68	026177	PR8	000624	RO4B	010422	RTY7	001064
MSG23I	025004	MSG69	026210	PRS	000622	RO4C	010426	RTY8	001066
MSG23J	025011	MSG70	026226	PSM	000606	RO4D	010456	RWNO	005056
MSG23K	025016	MSG71	026262	RANBAS	007626	RO4E	010462	RWNOA	005070