

# TM02/TU16

DATA RELIABILITY PROGRAM  
MD-11-DZTUA-G

EP-DZTUA-G-DL-B  
COPYRIGHT © 74-77  
FICHE 1 OF 1

OCT 1977  
**digital**  
MADE IN USA

This microfiche card contains a grid of frames. The left side of the card features a vertical column of frames, each containing a small, illegible image or document snippet. The remaining frames in the grid contain various data, including text, tables, and diagrams, which are too small to be legible. The card is oriented vertically and is a standard format for data storage on microfiche.

B01

EDF:027408880411  
DZ\*JAG.P11

17-JUN-77 09:52

000280006/TE1670006RELIABILITY PROGRAM411

MACY110080008TUAASEJUN-77 08:44

00010800

771026  
SEG 0001

.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZ\*UA-G-D  
PRODUCT TITLE: TMO2-TU16/TE16 DATA RELIBILITY PROGRAM  
DATE CREATED: 21 JUNE 1977  
MAINTAINER: DIAGNOSTIC GROUP  
AUTHOR: R.B. BARNES  
REVISED: MAY 1977 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 DOCUMENT.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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

COPYRIGHT (C) 1974,1977 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

PARAGRAPH	SUBJECT	PAGE
1	ABSTRACT	3
2	REQUIREMENTS	3
3	LOADING PROCEDURE	4
4	STARTING PROCEDURE	4
5	AUTOMATIC MODE OPER.	10
6	DATA PATTERNS	10
7	RANDOMIZATION	11
8	DYNAMIC PARAMETERS	12
9	CONSOLE SWITCH	18
10	ERROR PRINTOUTS	17
11	STATISTICS PRINTOUT	26
12	AUTO SEQUENCE	27
13	TESTING PROCEDURES	29
14	LISTING	30

1. ABSTRACT  
-----

THIS PROGRAM IS DESIGNED TO BE USED BY AN EXPERIENCED ENGINEER /TECHNICIAN FOR EVALUATION AND DEBUGGING OF MAG TAPE DRIVES. THE PROGRAM IS CAPABLE OF EXERCISING ANY TAPE DRIVE THAT CAN BE OPERATED ON A MASSBUS THROUGH THE TMO2 MAG TAPE CONTROLLER. ANY TYPE OF TAPE DRIVE; NRZI, PE, 7 OR 9 TRACK MAY BE USED. ANY NUMBER OF DRIVES, SINGLE OR MULTIDRIVE SYSTEMS, UP TO EIGHT (8), MAY BE TESTED BY A SINGLE EXECUTION OF THE PROGRAM. THIS FLEXIBILITY IS POSSIBLE BECAUSE THE PROGRAM HAS NO FIXED PARAMETERS OR TESTING SEQUENCE. THE ENTIRE TEST PLAN, INCLUDING PARAMETERS AND OPERATING SEQUENCE, IS DETERMINED BY THE OPERATOR THROUGH RESPONSES TO TELETYPE REQUESTS AND SETTING OF CONSOLE SWITCHES.

THE PROGRAM PROVIDES FOR TESTING OF ALL TAPE DRIVE FUNCTIONS SUCH AS WRITING, READING, REWINDING, TAPE POSITIONING, EOT - BOT SENSING AND ASSUMES A GOOD RH AND TMO2.

HOWEVER, THE RH AND TMO2 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 TMO2.

2. REQUIREMENTS (HARDWARE)  
-----

- A. ANY PDP-11 PROCESSOR - WITH OR WITHOUT HARDWARE SWITCH REGISTER
- B. 8K OF CORE
- C. TELETYPE
- D. TMO2 TAPE CONTROLLER
- E. 1 TO 8 MAG TAPE DRIVES
- F. MASSBUS CONTROLLER

3. LOADING PROCEDURE  
-----

USE STANDARD PROCEDURE FOR LOADING BINARY TAPES

4. STARTING PROCEDURE

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

- A. 200(8): THIS ADDRESS MUST BE USED ON INITIAL START FROM LOAD AS ALL PARAMETERS ARE ENTERED FROM HERE. REQUESTS ARE PRINTED ON THE TELETYPE FOR ENTRY OF RH STARTING ADDRESS, VECTOR ADDRESS, DRIVE NUMBER(TM02 ADDRESS), SLAVE NUMBER, DENSITY, PARITY, FORMAT, RECORD COUNT, CHARACTER COUNT, PATTERN NUMBER, TAPE MARK AND STALL FOR READ, WRITE, AND TURNAROUND. ALL 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. SEE ALSO SECTION 11 FOR DETAILS.
- E. 300(8): THIS ADDRESS IS TO BE USED AS A RESTART ONLY AND WILL PERFORM JUST AS IN 200(8) EXCEPT THAT THE PARAMETER INPUT LIST IS SHORTENED. THE SHORT PARAMETER LIST CONSISTS OF DRIVE NUMBER, SLAVE NUMBER, DENSITY, PARITY, FORMAT, RECORD COUNT, CHARACTER COUNT, PATTERN, TAPE MARK, AND INTERCHANGE READ.

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

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

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

DRIVE NUMBER: THE DRIVE NUMBER (MASSBUS ADDRESS OF THE TMO2) IS ENTERED AS ONE (1) OCTAL CHARACTER AND MUST BE WITHIN THE LIMITS OF 0 THROUGH 7.

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

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

DENSITY: THE 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. 0 = 200BPI, NRZI
- B. 1 = 556BPI, NRZI
- C. 2 = 800BPI, NRZI
- D. 3 = 800BPI, NRZI
- E. 4 = 1600BPI, PE (9 CHANNEL ONLY)

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

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

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

- A. 14 = 9 CHANNEL NORMAL (TWO FRAMES PER WORD)
- B. 15 = CORE DUMP (FOUR FRAMES PER WORD)

RECORD COUNT: THIS REQUEST IS RESPONDED TO BY A SIX (6) CHARACTER OCTAL NUMBER FROM 1 TO 177777. REMEMBER LEADING ZEROS ARE NOT REQUIRED AND IF LESS THAN SIX CHARACTERS ARE ENTERED, A CARRIAGE RETURN WILL TERMINATE THE RESPONSE. THE RECORD COUNT IS USED IN 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.

H01

TM02-TJ16/TE16 DATA RELIABILITY PROGRAM MACY11 30(1046) 22-JUN-77 08:44 PAGE 7  
DZTJAG.P11 17-JUN-77 09:52

SEG 0007

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

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

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



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

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

READ: THE TIME DELAY BETWEEN EACH RECORD READ

WRITE: THE TIME DELAY BETWEEN EACH RECORD WRITTEN

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

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

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

SAMPLE START AT 200(8):

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

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

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

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

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

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

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

4.1 AUTOMATIC MODE OPERATION  
-----

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

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

5. DATA PATTERNS  
-----

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

THE FOLLOWING IS A LIST OF THE DATA PATTERNS AVAILABLE:

DATA0: EXTERNAL INPUT THRU HIGH SPEED READER (SEE DTC)  
 DATA1: ALL ONE BITS IN ALL CHARACTERS  
 DATA2: ALL ZERO BITS IN ALL CHARACTERS  
 DATA3: A ONE BIT WALKING FROM RIGHT TO LEFT IN A FIELD OF ZEROS  
 DATA4: A ZERO BIT WALKING FROM RIGHT TO LEFT IN A FIELD OF ONES.  
 DATA5: ALTERNATING ONE AND ZERO BITS IN EACH CHARACTER  
 DATA6: ALTERNATING ZERO AND ONE BITS IN EACH CHARACTER  
 DATA7: SAME AS DATA5 BUT WITH EVERY OTHER CHARACTER COMPLEMENTED  
 DATA10: WALKING ONE/ALL ONE IN ALTERNATING CHARACTERS  
 DATA11: INCREMENTING CHARACTERS (00-377)  
 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 C CHARACTER AT THE TELETYPE. AS SOON AS THE BUS IS RELEASED BY THE MAG TAPE OPERATION IN PROGRESS, THE PROGRAM WILL RESPOND TO THE CONTROL C INPUT BY TYPING A REQUEST FOR NEW STALL PARAMETERS. THE LAST VALUES THAT WERE ENTERED WILL BE PRINTED AS THE STORED VALUES AND MAY BE CHANGED BY ENTERING NEW VALUES OR LEFT UNCHANGED BY TYPING A CARRIAGE RETURN.

THE YOZZLE STALL IS ALSO DYNAMIC AND CAN BE CHANGED BY TYPING A CNTRL C WHILE DOING A YOZZLE. A YOZZLE STALL REQUEST WILL BE PRINTED AND SHOULD BE RESPONDED TO WITH THE DESIRED VALUE.

B. CONSOLE SWITCH SETTINGS

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

CONTROL:

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

- 1) TYPE CONTROL G (<↑G>); THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.
  - A) THIS PROGRAM WILL PROCESS THE <↑G> EITHER IN FLAG MODE OR INTERRUPT DEPENDING ON WHERE IN THE PROGRAM THE <↑G> IS EXCEPTED. THE PROGRAM WILL SERVICE THE INTERRUPT ONLY WHEN THE PRIORITY IS LOWERED TO ALLOW THE TTY TO INTERRUPT.
- 2) THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)
- 3) AFTER THE ''NEW=''' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE OF THE FOLLOWING AT THE TTY:
  - A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>. (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS WILL BE ALLOWED) IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.
  - B) IF A CONTROL U (<↑U>) IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 2.

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

SW15: 1=STOP ON ERROR  
(10000)0=CONTINUE ON ERROR

SW14: 1=PRINT READ/WRITE STATISTICS  
(04000)0=DO NOT PRINT STATS

SW13: 1=DO NOT CHECK DATA ERRORS

(020000)0=CHECK DATA ERRORS

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

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

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

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

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

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

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

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

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

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

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

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

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

## SWITCH EXPLANATION AND EXAMPLES:

SW0-3:

THESE SWITCHES ARE USED TO CONTROL THE SEQUENCE OF MAG TAPE OPERATIONS 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.

\*\*\*\*\*PROGRAM HALTS\*\*\*\*\*

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

9. ERROR PRINTOUTS

THERE ARE THREE TYPES OF ERROR PRINTOUTS MADE BY THE PROGRAM: OPERATION ERRORS, DATA ERRORS, AND CONDITION ERRORS. EACH ERROR MESSAGE PRINTED IS 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 TM02 ITSELF OR BY THE MASSBUS CONTROLLER. ALL STATUS ERRORS WILL BE REPORTED.
2. TAPE POSITION ERRORS: THESE ARE INDICATED BY AN INCORRECT SPACE OR REWIND OPERATION IN WHICH TAPE POSITION BECOMES UNRELIABLE.

## B. DATA ERRORS:

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

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

## C. CONDITION ERRORS: (CATASTROPHIC)

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

1. EOT: WHEN EOT (END OF TAPE) IS ENCOUNTERED DURING  
EITHER A READ OR WRITE, THE CYCLE IS COMPLETED  
ON THE SHORTENED BLOCK AFTER WHICH THE SLAVE  
WILL BE REWOUND AND FLAGGED AS UNAVAILABLE  
FOR TESTING UNTIL ALL SLAVES HAVE REACHED EOT AND  
ARE REWOUND. WHEN THE LAST AVAILABLE SLAVE  
HAS REACHED EOT AND BEEN REWOUND TO BOT,  
TESTING WILL BE RESUMED ON ALL SLAVES.
2. ILLEGAL BOT: WHEN A SLAVE ENCOUNTERS BOT DURING  
A READ, WRITE, OR SPACE OPERATION, AN ERROR  
IS PRINTED AND THE PROGRAM HALTED. THIS IS  
A CATASTROPHIC ERROR. TESTING MAY BE RESUMED  
BY PRESSING CONTINUE; BUT A RESTART IS  
SUGGESTED.
3. NO INTERRUPT RETURNED: EACH TAPE OPERATION SHOULD BE  
TERMINATED BY THE SETTING OF AN INTERRUPT IN  
THE CPU. IF NO INTERRUPT IS RETURNED WITHIN  
THE APPROPRIATE TIME, AN ERROR IS PRINTED.
4. NO MEDIUM ON-LINE: BEFORE AN OPERATION IS ATTEMPTED,  
THE TMO2 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 UNTIL ALL ARE  
RESTARTED AT BLOCK ONE.
8. HARD READ ERROR: IF ANY HARD READ ERROR IS ENCOUNTERED  
DURING A RETRY, A MESSAGE IS PRINTED  
REGARDLESS OF THE SETTING OF SW10.
9. NON-RETRYABLE: IF ANY NON-RETRYABLE ERROR IS ENCOUNTERED, A  
MESSAGE IS PRINTED REGARDLESS OF THE SETTING OF SW10.

## D. EXAMPLES:

## GLOSSARY:

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

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

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

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

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

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

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

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

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

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

DRIVE NO. 3 \*SLAVE NO. 1 \*D 4 \*P 0 \*F 14 \*PATRN R  
\*BN 100 \*RN 66-200 \*RS 2000 \*DE F

CN 0  
G 11111111  
B 00000000  
CN 1  
G 11111111  
B 00000000  
CN 2  
G 11111111  
B 00000000  
CN 3  
G 11111111  
B 00000000  
CN 4  
G 11111111  
B 00000000  
CN 5  
G 1111'11  
B 00000000  
CN 6  
G 11111111  
B 00000000  
CN 7  
G 11111111  
B 00000000

BAD RECORD

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

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



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

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

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

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

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

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

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

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

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

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

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

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

10. STATISTICS PRINTOUT

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

STATISTICS PRINT EXAMPLE (A HEADER WILL PRECEED THE STATS)

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

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

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

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

11. AUTO SEQUENCE  
-----

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

SAMPLE START AT 240(8): AUTO SEQUENCE.

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

TJ16 AUTO SEQUENCE TEST  
ENTER CONDITIONS IN OCTAL

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

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

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

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

## 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



# E03

TM02-TU16/TE16 DATA RELIABILITY PROGRAM MACY11 30(1046) 22-JUN-77 08:44 PAGE 30  
DZTUAG.P11 17-JUN-77 09:52

SEG 0030

:REVISED (++) J.G.ADAMS MAY 1977

:++G  
:++G  
:++G  
:++G  
:++G  
:++G  
:++G

1)INTERMITTENT PGM FAILURE  
ON BAD TAPE OVERFLOW  
2)TAPE RUNAWAY AT EOT  
3)ERRONEOUS ERROR TYPEOUT  
4)CHANGED MISC INST'S TO  
CONSERVE MEMORY USAGE.  
5)ADDED ACT11 HOOKS  
6)FIXED TTY INPUT

.MCALL .SACT11,.SEOP,SCCHAIN ;++G ACT11 HOOKS  
.NLIST MC ;++G DO NOT LIST MACRO CALLS  
.LIST ME ;++G LIST MACRO EXPANSIONS  
.ENABLE ABS,AMA ;++G ENABLE ABS AND MODE '37'

: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

:SWC: 1=DO NOT WRITE  
: 0=WRITE

1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400





```

1446 ;REGISTER EQUIVS*****
1447
1448 000000 R0=%0
1449 000001 R1=%1
1450 000002 R2=%2
1451 000003 R3=%3
1452 000004 R4=%4
1453 000005 R5=%5
1454 000006 SP=%6
1455 000007 PC=%7
1456 000240 NOP=240
1457
1458 ;TRAP CATCHERS*****
1459 =30
1460 000030 024622 TRAP30
1461 000032 000032 =32
1462 000032 000340 340
1463
1464 ;ACT11 HOOK *****
1465
1466 000034 $SVPC= ;SAVE CURRENT LOCATION CTR
1467 000046 =46
1468 000046 005072 .WORD SENDAD ;SET LOCATION 46
1469 000052 000052 =52
1470 000052 000000 .WORD 0 ;SET LOCATION 52 = 0
1471 000034 =$SVPC ;RESTORE LOCATION CTR
1472
1473
1474 ;TTY INTERRUPT VECTOR*****
1475 000060 =60
1476 000060 021602 TTINT ;TTY INTERRUPT HANDLER ADDRESS
1477 000062 000000 0
1478
1479
1480 ;SOFTWARE SWITCH REGISTER LOC. 176*****
1481
1482 000176 =176
1483 000176 000000 SWREG: 0 ;SOFTWARE SWITCH REGISTER
1484
1485 ;START ADDRESS*****
1486
1487 000200 =200
1488 000200 000137 003026 JMP START ;ENTER PARAMETERS VIA TTY
1489
1490 000204 =204
1491 000204 000137 003152 JMP STARTC ;USE FIXED PARAMETERS: HOLD DATA
1492
1493 000210 =210
1494 000210 005037 015122 CLR RDFL
1495 000214 000137 003160 JMP STARTA ;USE FIXED PARAMETERS: NEW DATA
1496
1497 ;MAG TAPE INTERRUPT VECTOR*****
1498
1499 000224 =224
1500 000224 021666 MTINT ;MAG TAPE INTERRUPT HANDLER ADDRESS
1501 000226 000340 340
    
```

H03

TMO2-TJ16/TE16 DATA RELIABILITY PROGRAM MACY11 30(1046) 22-JUN-77 08:44 PAGE 33  
CZTUAG.P11 17-JUN-77 09:52

SEQ 0033

```
1502  
1503  
1504  
1505  
1506 000240 000240  
1507 000244 000137 000734 003136  
;AUTO SEQUENCE START*****  
.=240  
INC ASEQF ;SET AUTO SEQUENCE FLAG  
JMP STAUT ;GO TO START OF AUTO SEQUENCE
```

```

1508 ;SHORT CONVERSATION RESTART*****
1509
1510 ;=300
1511 000300 005237 014122 INC SCVFL ;SET SHORT CONVERSATION FLAG
1512 000304 000137 003026 JMP START ;ENTER SHORT PARAMETER LIST
1513
1514 000510 ;=510
1515 ;TU16/TE16 REGISTER EQUIVS*****
1516
1517 000510 172440 C1: 172440
1518 000512 172442 WC: 172442
1519 000514 172444 BA: 172444
1520 000516 172446 FC: 172446
1521 000520 172450 CS: 172450
1522 000522 172452 DS: 172452
1523 000524 172454 ER: 172454
1524 000526 172456 AS: 172456
1525 000530 172460 CC: 172460
1526 000532 172462 DB: 172462
1527 000534 172464 MR: 172464
1528 000536 172466 DT: 172466
1529 000540 172470 SN: 172470
1530 000542 172472 C2: 172472
1531
1532 ;CONSTANTS*****
1533
1534 000544 172440 REGS: 172440 ;STARTING REGISTER ADDRESS (CS1)
1535 000546 000224 VECT: 224 ;VECTOR ADDRESS (RH INTERRUPT)
1536 000550 000000 DYN: 0 ;DRIVE NUMBER
1537 000552 000000 UDES: 0 ;UNIT DESCRIPTION (PARITY,DENSITY,UNIT,FORMAT)
1538 000554 000100 RCNT: 100 ;RECORD COUNTER
1539 000556 177600 FMCNT: 177600 ;NUMBER OF CHAR (4 - 4000) OCTAL IN TWOS COMPLEMENT
1540 000560 000001 PATRN: 1 ;DATA PATTERN SELECTOR (0 - 15) OCTAL
1541 000562 000002 RDCMD: 2 ;READ COMMAND
1542 000564 000000 TMEX: 0 ;TAPE MARK FLAG: 1=TM 0=NO TM
1543 000566 000000 INTRF: 0 ;INTERCHANGE READ 1=YES 0=NO
1544 000570 000000 SPFLG: 0 ;SINGLE PASS 1=YES 0=NO
1545 000572 000001 RSTAL: 1 ;READ STALL
1546 000574 000001 WSTAL: 1 ;WRITE STALL
1547 000576 000001 TSTAL: 1 ;TURN AROUND STAL
1548 000600 002000 YSTAL: 2000 ;YOZZLE STAL
1549 000602 000010 RETRY: 10 ;READ RETRY NUMBER
1550 000604 177776 PSW: 177776 ;PROCESSOR STATUS
1551 000606 177570 SWR: 177570 ;CONSOLE SWITCHES
1552 000610 177560 TKS: 177560 ;TTY READ STATUS REGISTER
1553 000612 177562 TKB: 177562 ;TTY READ BUFFER
1554 000614 177564 TPS: 177564 ;TTY PUNCH STATUS REGISTER
1555 000616 177566 TPB: 177566 ;TTY PUNCH OUTPUT REGISTER
1556 000620 177550 PRS: 177550 ;H/S READER STATUS REGISTER
1557 000622 177552 PRB: 177552 ;H/S READER BUFFER
1558 000624 153624 RANBAS: 153624 ;RANDOM NUMBER GENERATOR BASE
1559 000626 032561 RANSAB: 032561 ;RANDOM NUMBER BUFFER
1560 000630 000000 RCSAV: 0 ;RECORD COUNT SAVE
1561 000632 000000 FCSAV: 0 ;FRAME COUNT SAVE

```

1562					
1563					
1564					
1565	000634	000000	TINF:	0	: TTY ENTRY FLAG
1566	000636	000000	TOB:	0	: TTY OUTPUT BUFFER
1567	000640	000000	TIB:	0	: TTY INPUT BUFFER
1568	000642	000000	TEMP1:	0	: TEMP STORAGE
1569	000644	000000	TEMP2:	0	: TEMP STORAGE
1570	000646	000000	TEMP3:	0	: TEMP STORAGE
1571	000650	000000	NRZOF:	0	: NRZ ONLY FLAG
1572	000652	000000	EMADDR:	0	: ERROR MSG ADDRESS STORAGE
1573	000654	000000	BLCNTR:	0	: BLOCK COUNTER
1574	000656	000000	BBC:	0	: BAD RECORD COUNTER
1575	000660	000000	EOTREC:	0	: EOT FLAG
1576	000662	000000	RTRN:	0	: INTERRUPT RETURN STORAGE
1577	000664	000000	HDRFL:	0	: HEADER FLAG
1578	000666	000000	STAL:	0	: DELAY STORAGE
1579	000670	000000	PFLG:	0	: PRINT FLAG
1580	000672	000000	MTC1:	0	: MAG TAPE CONT REGISTER BUFFER
1581	000674	000000	UNP:	0	: UNIT TABLE POINTER
1582	000676	000000	TMFLG:	0	: TAPE MARK FLAG
1583	000700	000000	RPCNT:	0	: REPEAT COUNTER
1584	000702	000000	RTCNT:	0	: RETRY COUNTER
1585	000704	000000	DERFL:	0	: DATA ERROR FLAG
1586	000706	000000	SERFL:	0	: STATUS ERROR FLAG
1587	000710	000000	BCNT:	0	: BIT COUNTER
1588	000712	000000	RTYFL:	0	: RETRY FLAG
1589	000714	000000	UPS:	0	: UNIT POINTER SAVE
1590	000716	000000	BDPP:	0	: BITS DROPPED POINTER
1591	000720	000000	BPKP:	0	: BITS PICKED POINTER
1592	000722	000000	ERSAV:	0	: ERROR SAVE LOC
1593	000724	000000	BTFLG:	0	: BAD TAPE FLAG
1594	000726	000000	BTSTF:	0	: STATISTIC PRINT FLAG
1595	000730	000000	BTPT:	0	: BAD TAPE POINTER
1596	000732	000000	ERTFL:	0	: ERASE FLAG
1597	000734	000000	ASEQF:	0	: AUTO SEQ FLAG
1598	000736	000000	ADRVN:	0	: JTO SEQ DRIVE NUMBER
1599	000740	000000	ABL CNT:	0	: AUTO BLOCK COUNTER
1600	000742	000001	ASEQCF:	1	: AUTO SEQ CONTINUOUS FLAG
1601	000744	000000	EOPB1:	0	: EOP FLAG

K03

SEQ 0036

```

1602
1603
1604
1605 000746 000000
1606 000750 000000
1607 000752 000000
1608 000754 000000
1609 000756 000000
1610 000760 000000
1611 000762 000000
1612 000764 000000
1613 000766 177777
1614
1615
1616
1617 000770 001210
1618 000772 001230
1619 000774 001250
1620 000776 001270
1621 001000 001310
1622 001002 001330
1623 001004 001350
1624 001006 001370
1625 001010 001410
1626 001012 001430
1627 001014 001450
1628 001016 001470
1629 001020 001510
1630 001022 001530
1631 001024 001550
1632 001026 001570
1633
1634
1635
1636 001030 001610
1637 001032 001714
1638 001034 002020
1639 001036 002124
1640 001040 002230
1641 001042 002334
1642 001044 002440
1643 001046 002544
1644
1645
1646
1647 001050 000000
1648 001052 000000
1649 001054 000000
1650 001056 000000
1651 001060 000000
1652 001062 000000
1653 001064 000000
1654 001066 000000
1655
1656
1657

```

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

```

JN1: 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\*\*\*\*\*

```

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

```

;UNIT WRITE ERRORS\*\*\*\*\*

1658 001070 000000  
1659 001072 000000  
1660 001074 000000  
1661 001076 000000  
1662 001100 000000  
1663 001102 000000  
1664 001104 000000  
1665 001106 000000

WTER1: 0  
WTER2: 0  
WTER3: 0  
WTER4: 0  
WTER5: 0  
WTER6: 0  
WTER7: 0  
WTER8: 0

;UNIT READ FORWARD ERRORS\*\*\*\*\*

1668  
1669 001110 000000  
1670 001112 000000  
1671 001114 000000  
1672 001116 000000  
1673 001120 000000  
1674 001122 000000  
1675 001124 000000  
1676 001126 000000

RDER1: 0  
RDER2: 0  
RDER3: 0  
RDER4: 0  
RDER5: 0  
RDER6: 0  
RDER7: 0  
RDER8: 0

;UNIT DATA ERRORS FORWARD\*\*\*\*\*

1677  
1678  
1679  
1680 001130 000000  
1681 001132 000000  
1682 001134 000000  
1683 001136 000000  
1684 001140 000000  
1685 001142 000000  
1686 001144 000000  
1687 001146 000000

DATER1: 0  
0  
0  
0  
0  
0  
0  
0

;UNIT READ REVERSE ERRORS\*\*\*\*\*

1688  
1689  
1690  
1691 001150 000000  
1692 001152 000000  
1693 001154 000000  
1694 001156 000000  
1695 001160 000000  
1696 001162 000000  
1697 001164 000000  
1698 001166 000000

RDERR1: 0  
0  
0  
0  
0  
0  
0  
0

;UNIT DATA ERRORS REVERSE\*\*\*\*\*

1699  
1700  
1701  
1702 001170 000000  
1703 001172 000000  
1704 001174 000000  
1705 001176 000000  
1706 001200 000000  
1707 001202 000000  
1708 001204 000000  
1709 001206 000000

DEREV1: 0  
0  
0  
0  
0  
0  
0  
0

```

1710 ;DROPS + PICKS PER CHANNEL PER UNIT*****
1711
1712 001210 000000 BP00: 0
1713 001230 001230 .=.+16
1714 001230 000000 BP10: 0
1715 001250 001250 .=.+16
1716 001250 000000 BP20: 0
1717 001270 001270 .=.+16
1718 001270 000000 BP30: 0
1719 001310 001310 .=.+16
1720 001310 000000 BP40: 0
1721 001330 001330 .=.+16
1722 001330 000000 BP50: 0
1723 001350 001350 .=.+16
1724 001350 000000 BP60: 0
1725 001370 001370 .=.+16
1726 001370 000000 BP70: 0
1727 001410 001410 .=.+16
1728 001410 000000 B000: 0
1729 001430 001430 .=.+16
1730 001430 000000 B010: 0
1731 001450 001450 .=.+16
1732 001450 000000 B020: 0
1733 001470 001470 .=.+16
1734 001470 000000 B030: 0
1735 001510 001510 .=.+16
1736 001510 000000 B040: 0
1737 001530 001530 .=.+16
1738 001530 000000 B050: 0
1739 001550 001550 .=.+16
1740 001550 000000 B060: 0
1741 001570 001570 .=.+16
1742 001570 000000 B070: 0
1743 001610 001610 .=.+16
1744
1745

```

```

1746
1747 ;UNIT BAD TAPE COUNTER:16 PER SLAVE*****
1748
1749 001610 000000 BT00: 0
1750 001714 000000 .=.+102
1751 001714 000000 BT01: 0
1752 002020 000000 .=.+102
1753 002020 000000 BT02: 0
1754 002124 000000 .=.+102
1755 002124 000000 BT03: 0
1756 002230 000000 .=.+102
1757 002230 000000 BT04: 0
1758 002334 000000 .=.+102
1759 002334 000000 BT05: 0
1760 002440 000000 .=.+102
1761 002440 000000 BT06: 0
1762 002544 000000 .=.+102
1763 002544 000000 BT07: 0
1764 002650 000000 .=.+102
1765
1766 ;UNIT END OF TAPE COUNTERS 1 PER SLAVE*****
1767
1768 002650 000000 EOTCO: 0
1769 002652 000000 0
1770 002654 000000 0
1771 002656 000000 0
1772 002660 000000 0
1773 002662 000000 0
1774 002664 000000 0
1775 002666 000000 0
1776
1777 ;UNIT READ FORWARD SOFT ERROR*****
1778
1779 002670 000000 RFSOFT: 0
1780 002672 000000 0
1781 002674 000000 0
1782 002676 000000 0
1783 002700 000000 0
1784 002702 000000 0
1785 002704 000000 0
1786 002706 000000 0
1787
1788 ;UNIT READ REVERSE SOFT ERROR*****
1789
1790 002710 000000 RRSOFT: 0
1791 002712 000000 0
1792 002714 000000 0
1793 002716 000000 0
1794 002720 000000 0
1795 002722 000000 0
1796 002724 000000 0
1797 002726 000000 0
1798

```



```

1799
1800 ;UNIT READ FORWARD HARD ERROR*****
1801
1802 002730 000000 RFHARD: 0
1803 002732 000000 0
1804 002734 000000 0
1805 002736 000000 0
1806 002740 000000 0
1807 002742 000000 0
1808 002744 000000 0
1809 002746 000000 0

```

```

1810
1811 ;UNIT READ REVERSE HARD ERROR*****
1812
1813 002750 000000 RRHARD: 0
1814 002752 000000 0
1815 002754 000000 0
1816 002756 000000 0
1817 002760 000000 0
1818 002762 000000 0
1819 002764 000000 0
1820 002766 000000 0

```

```

1821
1822 ;DATA PATTERN GENERATORS*****
1823

```

```

1824 002770 002770 DATBL: . ;ENTRY TABLE
1825 002772 014364 DATA0: DAT0 ;EXTERNAL INPUT FROM H/S READER(SEE MAINDEC-11-DZTUF)
1826 002774 014530 DATA1: DAT1 ;ALL ONES
1827 002776 014550 DATA2: DAT2 ;ALL ZEROS
1828 003000 014554 DATA3: DAT3 ;WALKING ONE
1829 003002 014600 DATA4: DAT4 ;WALKING ZERO
1830 003004 014610 DATA5: DAT5 ;ALTERNATING ONE/ZERO
1831 003006 014616 DATA6: DAT6 ;ALTERNATING ZERO/ONE
1832 003010 014624 DATA7: DAT7 ;ALTERNATING ONE/ZERO IN ALTERNATING CHARACTERS
1833 003012 014652 DATA10: DAT10 ;WALKING ONE/ALL ONE IN ALTERNATING CHARACTERS
1834 003014 014702 DATA11: DAT11 ;ALL BITS 0-377
1835 003016 014722 DATA12: DAT12 ;ALL BITS 377-0
1836 003020 014744 DATA13: DAT13 ;ALTERNATING CHARACTERS 0 AND 377
1837 003022 014754 DATA14: DAT14 ;WALKING ZERO/ALL ZERO IN ALTERNATING CHARACTERS
1838 003024 015004 DATA15: DAT15 ;AUTO SEQUENCE PATTERN 0,0,-1,-1,-1,0,0

```

C04

```

1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859 003026 012706 000500
1860 003032 005037 000734
1861 003036 005027
1862 003040 000000
1863
1864 003042 022737 005072 000042
1865 003050 001404
1866 003052 005737 000042
1867 003056 001413
1868 003060 000406
1869 003062 012737 000176 000506 50$:
1870 003070 012777 100000 175510
1871 003076 005237 003040 51$:
1872 003102 000137 003126
1873 003106
1874 003106 122737 000006 000041 52$:
1875 003114 001010
1876 003116 012704 027340
1877 003122 104000
1878 003124 000404
1879 003126 005237 000734 3$:
1880 003132 000137 021770
1881
1882
1883 003136 012737 000001 000634
1884 003144 005037 015122
1885 003150 000405
1886
1887
1888 003152 005037 000634
1889 003156 000434
1890
1891
1892 003160 005037 000634
1893 003164 012700 000636
1894 003170 012700 000037
1895 003174 005020

```

```

.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      ;++G 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$:  MOV    #SWREG,SWR   ;INVOKE SOFTWARE SWR
      MOV    #100000,2$SWR ;WITH HALT ON ERROR SET
51$:  INC    CHNFLG      ;SET CHNFLG = CHAIN MODE
      JMP   3$         ;GO TO CHAIN ADDRESS
52$:  CMPB   #6,2#41     ;++G BRANCH IF NOT LOADED VIA TMDP
      BNE   STAUT
      MOV   #MSG120,R4
      TTOUTT           ;++G ADVISE USER TO REMOVE TMDP
      BR    STAUT      ;++G
3$:   INC    ASEQF      ;++G SET AUTO SEQUENCE FLAG
      JMP   ASEQ0      ;++G GO TO AUTO SEQUENCER

```

```

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

```

```

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

```

```

:START 210 *****
STARTA: CLR    TINF     ;CLEAR TTY ENTRY FLAG
STARTB: MOV    #TOB,R0
      MOV    #37,R1
STARTC: CLR    R0,+     ;CLEAR FLAGS AND COUNTERS

```

1896	003176	005301			DEC	R1	
1897	003200	001375			BNE	START0	
1898	003202	012706	000500		MOV	#500,SP	;SET STACK POINTER
1899	003206	004737	004350		JSR	PC,RANSET	;GO RESET RANDOM BASE
1900	003212	012700	001050		MOV	#RTY1,RO	
1901	003216	012701	000750		MOV	#750,R1	
1902	003222	005020			STARTF: CLR	(RO)+	;CLEAR STATISTIC COUNTERS
1903	003224	005301			DEC	R1	
1904	003226	001375			BNE	STARTF	
1905	003230	012737	177777	014360	MOV	#-1,PATS	;PRESET PATTERN
1906	003236	005037	000744		CLR	EOPB1	
1907	003242	012737	000001	000654	STARTE: MOV	#1,BLCNTR	;PRESET BLOCK COUNTER
1908	003250	012706	000500		STARTD: MOV	#500,SP	
1909	003254	012777	000340	175322	MOV	#340,@PSW	
1910	003262	013746	000006		SUSWR: MOV	@#6,-(SP)	;SAVE VECTORS
1911	003266	013746	000004		MOV	@#4,-(SP)	
1912	003272	012737	003312	000004	MOV	#15,@#4	;SET UP FOR TIMEOUT
1913	003300	022777	177777	175300	CMP	#-1,@SWR	;REFERENCE HARDWARE SWITCH REGISTER
1914	003306	001402			BEQ	2\$	
1915	003310	000404			BR	3\$	
1916	003312	022626			1\$: CMP	(SP)+,(SP)+	;ADJUST STACK
1917	003314	012737	000176	000606	2\$: MOV	#SWREG,SWR	;POINT TO SOFTWARE SWITCH REG
1918	003322	012637	000004		3\$: MOV	(SP)+,@#4	;RESTORE VECTORS
1919	003326	012637	000006		MOV	(SP)+,@#6	
1920	003332	022737	000176	000606	CMP	#SWREG,SWR	;IS SWREG SELECTED
1921	003340	001007			BNE	4\$	
1922	003342	005737	000744		TST	EOPB1	
1923	003346	001004			BNE	4\$	
1924	003350	005037	000744		CLR	EOPB1	
1925	003354	004737	024404		JSR	PC,CNTLU	;CHECK FOR CONTROL G
1926	003360	004737	012352		4\$: JSR	PC,TIMP	;GO GET PARAMETERS FROM TTY
1927	003364	012777	000040	175126	MOV	#40,@CS	;INITIALIZE
1928	003372	005000			STAUTO: CLR	RO	;POINT TO FIRST ENTRY
1929	003374	022760	177777	000746	1\$: CMP	#-1,UNI(RO)	;++G BRANCH IF LAST ENTRY
1930	003402	001406			BEQ	2\$	
1931	003404	042760	100000	000746	BIC	#100000,UNI(RO)	;CLEAR EOT FLAG
1932	003412	062700	000002		ADD	#2,RO	;POINT TO NEXT UNIT ENTRY
1933	003416	000766			BR	1\$	;++G CONTINUE CLEARING
1934	003420	013703	005130		2\$: MOV	REOTC,R3	
1935	003424	000303			SWAB	R3	
1936	003426	110337	005130		MOV	R3,REOTC	;RESTORE EOT CNTR
1937	003432	012777	000100	175150	START1: MOV	#100,@TKS	;SET TTY INTERRUPT ENABLE
1938	003440	013700	000674		MOV	UNP,RO	;RO = UNIT TABLE POINTER
1939	003444	022760	177777	000746	STAR1A: CMP	#-1,UNI(RO)	;++G BRANCH IF LAST ENTRY
1940	003452	001404			BEQ	STAR1B	;IF LAST UNIT IN STRING: BR
1941	003454	016037	000746	000552	MOV	UNI(RO),UDES	;LOAD NEXT UNIT DESCRIPTION
1942	003462	000446			BR	START4	;++G
1943	003464	005237	000654		STAR1B: INC	BLCNTR	;BUMP BLOCK COUNTER
1944	003470	005737	000734		TST	ASEQF	;SEE IF AUTO SEQ
1945	003474	001411			BEQ	STAR1C	;IF NOT: BR
1946	003476	023737	000654	000740	CMP	BLCNTR,ABL CNT	;SEE IF DONE SEQ
1947	003504	001005			BNE	STAR1C	;IF NOT: BR
1948	003506	005037	000654		CLR	BLCNTR	;RESET BLOCK CNTR
1949	003512	005037	000674		CLR	UNP	;RESET UNIT POINTER
1950	003516	000207			RTS	PC	;RETURN TO AUTO SEQ

# E04

TMO2-TU16/TE16 DATA RELIABILITY PROGRAM MACY11 30(1046) 22-JUN-77 08:44 PAGE 43  
 027JAG.P11 17-JUN-77 09:52

SEG 0043

1952	003520	005037	000674		STAR1C:	CLR	UNP		
1953	003524	005000				CLR	RO		
1954	003526	016037	000746	000552		MOV	UN1(RO), UDES	: LOAD FIRST UNIT DESCRIPTION	
1955	003534	105777	175046			TSTB	JSWR	: ++G BRANCH IF NOT RANDOM RECORD	
1956	003540	100003				BPL	START2	: ++G SIZE REQUESTED.	
1957	003542	001402				BEQ	START2	: IF NOT: BR	
1958	003544	004737	012266			JSR	PC, CNTR	: GO GENERATE RANDOM RECORD SIZE	
1959	003550	032777	000400	175030	START2:	BIT	#400 JSWR	: SEE IF RANDOM DATA	
1960	003556	001402				BEQ	START3	: IF NOT: BR	
1961	003560	004737	015054			JSR	PC, DATA	: GO GENERATE RANDOM DATA	
1962	003564	032777	000100	175014	START3:	BIT	#100 JSWR	: SEE IF RANDOM RECORD COUNT	
1963	003572	001402				BEQ	START4	: IF NOT: BR	
1964	003574	004737	012326			JSR	PC, RCNTR	: GO GENERATE RANDOM RECORD COUNT	
1965	003600	005760	000746		START4:	TST	UN1(RO)	: ++G BRANCH IF NOT AT EOT	
1966	003604	100002				BPL	START40	: IF NOT: BR	
1967	003606	000137	004336			JMP	START7	: ELSE GO TO NEXT UNIT	
1968	003612	013777	000550	174700	STAR40:	MOV	DVN, JCS	: SET DRIVE NUMBER	
1969	003620	013777	000552	174714		MOV	UDES, JCS	: SET UNIT NUMBER	
1970	003626	105777	174670			TSTB	JDS	: ++G BRANCH IF UNIT AVAIL	
1971	003632	100412				BMI	STAR4A		
1972	003634	005337	000666			DEC	STAL		
1973	003640	001357				BNE	START4	: AWAIT TUR	
1974	003642	004737	022662			JSR	PC, PAPRT	: PRINT HEADER	
1975	003646	012704	026041			MOV	#MSG49, R4		
1976	003652	104000				TTOUTT	: PRINT NOT AVAIL		
1977	003654	104006				STOPP		: STOP	
1978	003656	000750				BR	START4	: ++G RETRY	
1979	003660	013746	000552		STAR4A:	MOV	UDES, -(SP)	: GET UNIT DESCRIPTION	
1980	003664	042716	175400			BIC	#175400, (SP)	: ++G CLEAR ALL BUT FORMAT BITS	
1981	003670	022726	001700			CMP	#1700, (SP)+	: ++G BRANCH IF NRZ	
1982	003674	001406				BEQ	IS	: ++G	
1983	003676	032777	000040	174616		BIT	#40, JDS	: ++G BRANCH IF SLAVE IN PE FORMAT	
1984	003704	001002				BNE	IS	: ++G	
1985	003706	000137	004336			JMP	START7	: ++G GO TO NEXT UNIT	
1986	003712	004737	014152		IS:	JSR	PC, DSUP	: GO SET UP WRITE DATA	
1987	003716	004737	005132			JSR	PC, RWND	: REWIND	
1988	003722	004737	005474			JSR	PC, WRITE	: WRITE	
1989	003726	013737	000576	000666		MOV	TSTAL, STAL	: SET TURN AROUND DELAY	
1990	003734	004737	012256			JSR	PC, STALL	: DELAY	
1991	003740	004737	007366			JSR	PC, RSEQ	: GO TO READ SEQUENCER	
1992	003744	013737	000576	000666		MOV	TSTAL, STAL	: SET TURN AROUND DELAY	
1993	003752	004737	012256			JSR	PC, STALL	: DELAY	
1994	003756	032777	040000	174622		BIT	#40000, JSWR	: SEE IF SHOULD PRINT STATISTICS	
1995	003764	001541				BEQ	START5	: IF NOT: BR	
1996	003766	012700	000001			MOV	#1, RO	: SET RECORD COUNTER TO 1	
1997	003772	004737	022662			JSR	PC, PAPRT	: PRINT CYCLE NUMBER	
1998	003776	004737	004006			JSR	PC, STP	: GO PRINT STATS	
1999	004002	000137	004254			JMP	STPX		
2000	004006	004737	017222		STP:	JSR	PC, DPPRT	: PRINT DROPS AND PICKS	
2001	004012	012704	026254			MOV	#MSG65, R4		
2002	004016	104000				TTOUTT	: PRINT RETRY TOTAL		
2003	004020	013704	000674			MOV	UNP, R4		
2004	004024	016402	001050			MOV	RTY1(R4), R3		
2005	004030	104002				OCTPP		: PRINT RETRIES	
2006	004032	012704	026425			MOV	#MSG73, R4		
2007	004036	104000				TTOUTT		: PRINT WRITE ERROR TAG	

# F04

TMO2-TU16/TE16 DATA RELIABILITY PROGRAM MACY11 30(1046) 22-JUN-77 08:44 PAGE 44  
 02TUAG.P11 17-JUN-77 09:52

SEG 0044

2008	004040	013704	000674	MOV	UNP,R4	
2009	004044	016403	001070	MOV	WTER1(R4),R3	
2010	004050	104002		OCTPP		;PRINT WRITE ERRORS
2011	004052	012704	026414	MOV	#MSG72,R4	
2012	004056	104000		TTOUTT		;PRINT READ FORWARD ERROR TAG
2013	004060	013704	000674	MOV	UNP,R4	
2014	004064	016403	001110	MOV	RDER1(R4),R3	
2015	004070	104002		OCTPP		;PRINT READ FORWARD ERRORS
2016	004072	012704	027222	MOV	#MSG113,R4	
2017	004076	104000		TTOUTT		;PRINT SOFT TAG
2018	004100	013704	000674	MOV	UNP,R4	
2019	004104	016403	002670	MOV	RFSOFT(R4),R3	
2020	004110	104002		OCTPP		;PRINT FORWARD SOFT ERRORS
2021	004112	012704	027233	MOV	#MSG114,R4	
2022	004116	104000		TTOUTT		;PRINT HARD TAG
2023	004120	013704	000674	MOV	UNP,R4	
2024	004124	016403	002730	MOV	RFHARD(R4),R3	
2025	004130	104002		OCTPP		;PRINT HARD FORWARD ERRORS
2026	004132	012704	026505	MOV	#MSG77,R4	
2027	004136	104000		TTOUTT		;PRINT DATA ERROR FORWARD TAG
2028	004140	013704	000674	MOV	UNP,R4	
2029	004144	016403	001130	MOV	DATER1(R4),R3	
2030	004150	104002		OCTPP		;PRINT DATA ERROR FORWARD NUMBER
2031	004152	012704	026310	MOV	#MSG68,R4	
2032	004156	104000		TTOUTT		;PRINT READ ERROR REVERSE TAG
2033	004160	013704	000674	MOV	UNP,R4	
2034	004164	016403	001150	MOV	RDERR1(R4),R3	
2035	004170	104002		OCTPP		;PRINT REVERSE ERROR NUMBER
2036	004172	012704	027222	MOV	#MSG113,R4	
2037	004176	104000		TTOUTT		;PRINT SOFT TAG
2038	004200	013704	000674	MOV	UNP,R4	
2039	004204	016403	002710	MOV	RRSOFT(R4),R3	
2040	004210	104002		OCTPP		;PRINT REVERSE SOFT ERROR
2041	004212	012704	027233	MOV	#MSG114,R4	
2042	004216	104000		TTOUTT		;PRINT HARD TAG
2043	004220	013704	000674	MOV	UNP,R4	
2044	004224	016403	002750	MOV	RRHARD(R4),R3	
2045	004230	104002		OCTPP		
2046	004232	012704	026474	MOV	#MSG76,R4	
2047	004236	104000		TTOUTT		;PRINT DATA ERROR REVERSE TAG
2048	004240	013704	000674	MOV	UNP,R4	
2049	004244	016403	001170	MOV	GEREV1(R4),R3	
2050	004250	104002		OCTPP		;PRINT DATA REVERSE ERROR NUMBER
2051	004252	000207		RTS	PC	;RETURN
2052	004254	005237	000726	STPX: INC	BTSTF	;SET STAT ONLY PRINT
2053	004260	004737	007276	JSR	PC,BTPRT	;PRINT BAD TAPE STATS
2054	004264	005037	000726	CLR	BTSTF	;CLEAR FLAG
2055	004270	017700	174312	STARTS: MOV	#SWR,R0	;LOAD SWR
2056	004274	042700	177762	BIC	#177762,R0	;MASK READ/WRITE SWITCHES
2057	004300	022700	000015	CMP	#15,R0	;SEE IF HAVE READ OR WRITE
2058	004304	001417		BEG	START8	;IF NOT: BR
2059	004306	105777	174210	START6: TSTB	#DS	;++G BRANCH IF HAVE UNIT READ
2060	004312	004111		BMI	START7	;++G
2061	004314	005337	000666	DEC	STAL	
2062	004320	001372		BNE	START6	;DELAY FOR TUR
2063	004322	004737	022662	JSR	PC,PAPRT	;PRINT HEADER

G04

TM02-TU16/TE16 DATA RELIABILITY PROGRAM MACY11 30(1046) 22-JUN-77 08:44 PAGE 45  
DZTUAG.P11 17-JUN-77 09:52

SEG 0045

```

2064 004326 012704 026041      MOV      #MSG49,R4
2065 004332 104000      TTOUTT      ;PRINT NOT AVAIL
2066 004334 104006      STOPP      ;STOP
2067 004336 062737 000002 000674  START7: ADD      #2,UNP      ;POINT TO NEXT UNIT
2068 004344 000137 003432  START8: JMP      START1      ;CONTINUE
2069
2070      ;RANDOM BASE RESET*****
2071
2072 004350 012737 153624 000624  RANSET: MOV      #153624,RANBAS ;RESET BASE
2073 004351 012737 032561 000626      MOV      #32561,RANSAV ;RESET BUFFER
2074 004364 013737 000630 000554      MOV      RCSAV,RCNT ;RESET RECORD COUNT
2075 004372 013737 000632 000556      MOV      FCSAV,FMCNT ;RESET FRAME COUNT
2076 004400 000207      RTS      PC

```

2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133

004402 013777 000552 174132  
004410 012777 000011 174072  
004416 105777 174100  
004422 100375  
004424 012777 000007 174056  
004432 005737 000724  
004436 001004  
004440 013700 000660  
004444 042700 100000  
004450 005037 000660  
004454 004737 022662  
004460 022737 000002 000724  
004466 001003  
004470 012704 027113  
004474 000406  
004476 022737 000001 000724  
004504 001004  
004506 012704 026724  
004512 104000  
004514 000412  
004516 012704 025052  
004522 104000  
004524 013704 000674  
004530 005264 002650  
004534 016403 002650  
004540 104002  
004542 012704 026751  
004546 104000  
004550 005037 000724  
004554 004737 004006  
004560 004737 007276  
004564 105777 173732  
004570 100414  
004572 005337 000666  
004576 001372  
004600 012737 024713 000652  
004606 004737 022662  
004612 012704 026216  
004616 104000  
004620 104006  
004622 005337 005130  
004626 001410  
004630 013700 000674  
004634 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,RC2 ;LOAD COMMAND REGISTER  
MOV R11,RC1 ;DRIVE CLEAR  
IS: TSTB JDS ;++G WAIT FOR DRIVE READY  
BPL IS ;AWAIT DRY  
MOV R7,RC1 ;START REWIND  
TST BTFLG ;SEE IF BAD TAPE OVERFLOW REWIND  
BNE REOT1A ;IF SO: BR  
MOV EOTREC,R0  
BIC #100000,R0 ;SET RECORD NUMBER OF EOT  
REOT1A: CLR EOTREC ;++G CLEAR EOT IND & REC CTR  
JSR PC,PAPRT ;PRINT HEADER  
CMP R2,BTFLG ;SEE IF POSITION ERROR  
BNE REOT1B ;IF NOT: BR  
MOV #MSG109,R4 ;SET POSITION ERROR MSG  
BR REOT1F  
REOT1B: CMP R1,BTFLG ;SEE IF BAD TAPE OVERFLOW  
BNE REOT1C ;IF NOT: BR  
MOV #MSG106,R4 ;SET BAD TAPE OVERFLOW MSG  
REOT1F: TTOUTT ;PRINT REWIND REASON  
BR REOT1E  
REOT1C: MOV #MSG20,R4 ;SET EOT MSG  
REOT1D: TTOUTT ;PRINT MSG  
MOV UNP,R4  
INC EOTCO(R4) ;BUMP CNTR  
MOV EOTCO(R4),R3  
OCTPP ;PRINT EOT CNTR  
REOT1E: MOV #MSG16A,R4 ;PRINT RESTART MSG  
TTOUTT ;CLEAR BAD TAPE FLAG  
CLR BTFLG ;PRINT STATS  
JSR PC,STP ;PRINT BAD TAPE STATS  
JSR PC,BTPRT ;++G BRANCH IF UNIT IS READY  
REOT2: TSTB JDS  
BMI REOT2A  
DEC STAL  
BNE REOT2 ;WAIT DRY  
MOV #MSG6,EMADDR  
JSR PC,PAPRT ;PRINT HEADER  
MOV #MSG60,R4 ;PRINT NO DRIVE READY  
TTOUTT  
STOPP  
REOT2A: DECB REOTC ;SEE IF LAST UNIT TO REACH EOT  
BEQ REOT3 ;IF SO: BR  
MOV UNP,R0  
BIS #100000,UNP,R0 ;SET EOT FLAG
```

```

2134 004642 005726          TST      (SP)+          ;RESET STACK POINTER
2135 004644 000137 004336    JMP      START7        ;GO TO NEXT UNIT
2136 004650 000337 005130    REOT3: SWAB      REOTC
2137 004654 013700 005130    MOV      REOTC,R0
2138 004660 000337 005130    SWAB      REOTC
2139 004664 110037 005130    MOV      RD,REOTC      ;RESTORE EOT UNIT COUNTER
2140 004670 005037 000674    CLR      UNP
2141 004674 013700 000674    MOV      UNP,R0        ;POINT TO FIRST UNIT
2142 004700 016037 000746 000552 REOT4: MOV      UNI(R0),UDES ;LOAD UNIT DESCRIPTION
2143 004706 013777 000552 173626 MOV      UDES,R0      ;LOAD COMMAND REGISTER
2144 004714 032777 020000 173670 REOT5: BIT      #20000,R0S
2145 004722 001374          BNE      REOT5        ;AWAIT PIP RESET
2146 004724 032777 000002 173570 BIT      #2,R0S       ;SEE IF HAVE BOT
2147 004732 001012          BNE      REOT6        ;IF SO: BR
2148 004734 012700 000001    MOV      #1,R0
2149 004740 004737 022662    JSR      PC,PAPRT     ;PRINT HEADER
2150 004744 012704 026007    MOV      #MSG48,R4
2151 004750 104000          TTOUTT          ;PRINT BOT ERROR
2152 004752 104006          STOPP
2153 004754 013700 000674    MOV      UNP,R0
2154 004760 042760 100000 000746 REOT6: BIC      #100000,UNI(R0) ;CLEAR EOT FLAG
2155 004766 062737 000002 000674 ADD      #2,UNP
2156 004774 013700 000674    MOV      UNP,R0        ;POINT TO NEXT UNIT
2157 005000 022760 177777 000746 REOT7: CMP      #-1,UNI(R0) ;++G BRANCH IF NOT LAST UNIT
2158 005006 005037 000674    CLR      UNP          ;CLEAR UNIT POINTER
2159 005012 005037 000634    CLR      TINF        ;CLEAR TTY INPUT FLAG
2160 005016 005737 000734    TST      ASEQF       ;SEE IF AUTO SEQ
2161 005022 001402          BEQ      REOTX       ;IF NOT: BR
2162 005024 005726          TST      (SP)+      ;RESET STACK POINTER
2163 005026 000207          RTS               ;RETURN TO AUTO SEQ
2164
2165 005030 004737 004350 014360 REOTX: JSR      PC,RANSET ;GO RESET RANDOM BASE
2166 005034 012737 177777    MOV      #-1,PATS    ;PRESET PATTERN
2167 005042 005037 015122    CLR      RDFL        ;CLEAR RANDOM FLAG
2168 005046 005737 000570    TST      SPFLG       ;SEE IF SINGLE PASS
2169 005052 001422          BEQ      REOTXX     ;IF NOT: BR
2170 005054 012704 026617    TEND: MOV      #MSG100,R4
2171 005060 104000          TTOUTT          ;PRINT END OF PASS
2172 005062 013700 000042    MOV      #42,R0      ;GET ACT11 RETURN ADDRESS
2173 005066 001405          BEQ      HERE       ;BRANCH IF NOT ACT11
2174 005070 000005          RESET
2175 005072 004710    SENDAD: JSR      PC,(R0)
2176 005074 000240    NOP
2177 005076 000240    NOP
2178 005100 000240    NOP
2179 005102 000240    HERE: NOP
2180 005104 005737 003040    TST      CHNFLG     ;++G BRANCH IF NOT CHAIN MODE
2181 005110 001402          BEQ      IS
2182 005112 000137 021770    JMP      ASEQO      ;++G RETURN TO AUTO SEQUENCER
2183 005116 104006          IS: STOPP
2184 005120 005237 000744    REOTXX: INC      EOPB1
2185 005124 000137 003242    JMP      STARTE    ;RESTART AT BLOCK NUMBER ONE
2186 005130 000000    REOTC: D           ;EOT UNIT COUNTER
    
```



```

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

```

```

2187
2188
2189
2190
2191
2192
2193
2194
2195
2196 005132 032777 001000 173446 RWND: BIT #1000,JSWR ;SEE IF SHOULD REWIND
2197 005140 001001 BNE RWNDA ;IF SO: BR
2198 005142 000207 RTS PC ;ELSE EXIT
2199 005144 013737 000674 000714 RWNDA: MOV UNP,UPS ;SAVE UNIT POINTER
2200 005152 005037 000674 CLR UNP ;CLEAR POINTER
2201 005156 005037 000660 CLR EOTREC ;CLEAR EDT FLAG
2202 005162 000337 005130 SWAB REOTC
2203 005166 013700 005130 MOV REOTC,RO
2204 005172 000337 005130 SWAB REOTC
2205 005176 110037 005130 MOVB RO,REOTC ;RESTORE EOT UNIT COUNTER
2206 005202 013700 000674 RWND0: MOV UNP,RO ;POINT TO UNIT ENTRY
2207 005206 022760 177777 000746 CMP #-1,UNI(RO) ;++G BRANCH IF LAST ENTRY
2208 005214 001445 BEQ RWND2 ;IF SO: BR
2209 005216 005760 000746 TST UNI(RO) ;++G BRANCH IF ALREADY REWINDING
2210 005222 100433 BMI RWND1A ;++G
2211 005224 016037 000746 000552 MOV UNI(RO),UDES ;SET UNIT DESCRIPTION
2212 005232 013777 000552 173302 MOV UDES,JC2 ;LOAD COMMAND REGISTER
2213 005240 012777 000011 173242 MOV #11,JC1 ;DRIVE CLEAR
2214 005246 012777 000007 173234 MOV #7,JC1 ;START REWIND
2215 005254 105777 173242 RWND1: TSTB JDS ;++G WAIT FOR DRIVE READY
2216 005260 100414 BMI RWND1A ;IF DRY: BR
2217 005262 005337 000666 DEC STAL
2218 005266 001372 BNE RWND1 ;AWAIT DRY
2219 005270 012737 024713 000652 MOV #MSG6,EMADDR
2220 005276 004737 022662 JSR PC,PAPRT ;PRINT HEADER
2221 005302 012704 025337 MOV #MSG70,R4
2222 005306 104000 TTOUTT ;PRINT NO DRIVE READY
2223 005310 104006 STOPP
2224 005312 042760 100000 000746 RWND1A: BIC #100000,UNI(RO) ;CLEAR EOT FLAG
2225 005320 062737 000002 000674 ADD #2,UNP ;BUMP POINTER
2226 005326 000725 BR RWND0 ;++G DO NEXT UNIT
2227 005330 005037 000674 RWND2: CLR UNP ;CLEAR POINTER
2228 005334 013700 000674 RWND3: MOV UNP,RO ;POINT TO UNIT ENTRY
2229 005340 022760 177777 000746 CMP #-1,UNI(RO) ;++G BRANCH IF LAST ENTRY
2230 005346 001436 BEQ RWNDX ;IF SO: BR
2231 005350 016037 000746 000552 MOV UNI(RO),UDES ;SET UNIT DESCRIPTION
2232 005356 013777 000552 173156 MOV UDES,JC2 ;LOAD COMMAND REGISTER
2233 005364 032777 020000 173130 RWND4: BIT #20000,JDS
2234 005372 001374 BNE RWND4 ;AWAIT PIP RESET
2235 005374 032777 000002 173120 BIT #2,JDS ;SEE IF HAVE BOT
2236 005402 001407 BEQ RWND6 ;IF NOT: BR
2237 005404 062737 000002 000674 RWND5: ADD #2,UNP ;BUMP POINTER
2238 005412 012777 000011 173070 MOV #11,JC1 ;DRIVE CLEAR
2239 005420 000745 BR RWND3 ;++G DO NEXT UNIT
2240 005422 012700 000001 RWND6: MOV #1,RO
2241 005426 004737 022662 JSR PC,PAPRT ;PRINT HEADER
2242 005432 012704 026007 MC. #MSG48,R4

```

K04

TM02-TL16/TE16 DATA RELIABILITY PROGRAM MACY11 30(1046) 22-JUN-77 08:44 PAGE 49  
DZTJAG.P11 17-JUN-77 09:52

SEG 0049

```
2243 005436 104000          TTOUTT          ;PRINT NO BOT
2244 005440 104006          STOPP
2245 005442 000760          BR             RWND5          ;++G DO NEXT UNIT
2246 005444 013737 000714 000674 RWNDX: MOV          UPS,UNP          ;RESTORE UNIT POINTER
2247 005452 013700 000674          MOV          UNP,RO
2248 005456 016037 000746 000552          MOV          UNI(RO),UDES ;RESET UNIT DESCRIPTION
2249 005464 013777 000552 173050          MOV          UDES,PC2
2250 005472 000207          RTS           PC           ;RETURN TO TEST
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  
2281  
2282  
2283  
2284  
2285  
2286  
2287  
2288  
2289  
2290  
2291  
2292  
2293  
2294  
2295  
2296  
2297  
2298  
2299  
2300  
2301  
2302  
2303  
2304  
2305  
2306  
2307

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

```
005474 032777 000001 173104 WRITE: BIT #1,JSWR ;SEE IF SHOULD WRITE  
005502 001402 BEQ WRITE  
005504 000137 006272 JMP WEX ;IF NOT: BR  
005510 013700 000554 WRITE: MOV RCNT,RO ;RO=RECORD COUNT  
005514 012737 024706 000652 W0: MOV #MSG5,EMADDR ;SET ERROR MSG ADDRESS  
005522 013777 000556 172766 MOV FMCNT,JFC ;LOAD CHAR COUNT  
005530 012777 027430 172756 MOV #WDATA,JBA ;SET DATA ADDR  
005536 112737 000060 000672 MOV #60,MT01 ;SET WRITE OP COMMAND  
005544 012737 005556 000662 MOV #W1,RTRN ;SET RETURN ADDRESS  
005552 000137 021170 JMP TAPG ;GO EXECUTE COMMAND  
005556 032777 002000 172736 W1: BIT #2000,SDS ;SEE IF EOT  
005564 001414 BEQ W2 ;IF NOT AT EOT: BR  
005566 005737 000660 TST EOTREC ;++G BRANCH IF WRITTEN PAST EOT  
005572 100411 W2 ;++G  
005574 010037 000660 MOV RO,EOTREC ;SAVE EOT RECORD NUMBER  
005600 052737 100000 000660 BIS #100000,EOTREC ;++G SET EOT FLAG  
005606 005337 000660 DEC EOTREC ;++G ADJUST RECORD COUNT  
005612 012700 000002 MOV #2,RO ;++G SET RO TO WRITE 1 MORE RECORD  
005616 032777 010000 172762 W2: BIT #10000,JSWR ;SEE IF SHOULD CHECK ERRORS  
005624 001002 BNE W3 ;IF NOT: BR  
005626 004737 017360 JSR PC,ERCHK ;GO CHECK ERRORS  
005632 013737 000574 000666 W3: MOV WSTAL,STAL ;SET DELAY  
005640 004737 012256 JSR PC,STAL ;DELAY
```

2308	005644	005737	000712		TST	RTYFL		;SEE IF RETRY TIME
2309	005650	001401			BEQ	W3A		;IF NOT: BR
2310	005652	000207			RTS	PC		;ELSE RETURN
2311	005654	005737	000706	W3A:	TST	SERFL		;SEE IF WRITE ERROR
2312	005660	001450			BEQ	W5		;IF NOT: BR
2313	005662	013704	000674		MOV	UNP,R4		
2314	005666	005264	001070		INC	WTER1(R4)		;BUMP WRITE ERROR
2315	005672	005037	000706		CLR	SERFL		;CLEAR STATUS ERROR FLAG
2316	005676	032777	000020	172702	BIT	#20,@SWR		;SEE IF RETRY
2317	005704	001436			BEQ	W5		;IF NOT: BR
2318	005706	013703	000722		MOV	ERSAV,R3		
2319	005712	042703	10270C		BIC	#102700,R3		;MASK UNRECOVERABLE ERROR
2320	005716	001410			BEQ	W4		;IF SO: BR
2321	005720	004737	022662		JSR	PC,PAPRT		;PRINT HEADER
2322	005724	012704	026516		MOV	#MSG78,R4		
2323	005730	104000			TTOUTT			;PRINT NON-RETRYABLE ERROR TAG
2324	005732	004737	011232		JSR	PC,NRTP		;PRINT ER FOR NON-RETRYABLE
2325	005736	000421			BR	W5		
2326	005740	013704	000674	W4:	MOV	UNP,R4		
2327	005744	005264	001050		INC	RTY1(R4)		;BUMP RETRY CNTR
2328	005750	032777	002000	172630	BIT	#2000,@SWR		;SEE IF PRINT ERRORS
2329	005756	001003			BNE	W4A		;IF NOT: BR
2330	005760	012704	026232		MOV	#MSG64,R4		
2331	005764	104000			TTOUTT			;PRINT ORIGINAL ERROR TAG
2332	005766	005037	000702	W4A:	CLR	RTCNT		;CLEAR RETRY NUMBER
2333	005772	005037	000700		CLR	RPCNT		;CLEAR REPEAT COUNTER
2334	005776	004737	006334		JSR	PC,WRTY		;GO RETRY WRITE ERROR
2335	006002	005037	000712	W5:	CLR	RTYFL		;CLEAR RETRY COUNTER
2336	006006	005300			DEC	RD		;SEE IF DONE ALL
2337	006010	001241			BNE	W0		;IF NOT: BR
2338	006012	005737	000564	W6:	TST	TMEX		;SEE IF TM
2339	006016	001525			BEQ	WEX		;IF NOT: BR
2340	006020	005237	000676		INC	TMFLG		;SET TM FLAG
2341	006024	012737	026137	000652	MOV	#MSG54,EMADDR		;POINT TO TM ERROR MSG
2342	006032	012737	000026	000672	MOV	#26,MTC1		;SET TM OP CODE
2343	006040	012777	000000	172450	MOV	#0,@FC		;LOAD FRAME COUNTER
2344	006046	012777	027430	172440	MOV	#WDATA,@BA		;LOAD BUS ADDRESS
2345	006054	012737	006066	000662	MOV	#WTMO,@ATR		;SAVE RETURN ADDRESS
2346	006062	000137	021170		JMP	TAPG		;WRITE TM
2347	006066	032777	010000	172512	BIT	#10000,@SWR		;SEE IF SHOULD CHECK ERRORS
2348	006074	001076			BNE	WEX		
2349	006076	032777	000004	172416	BIT	#4,@DS		;SEE IF TM STATUS
2350	006104	001011			BNE	WTM1		;IF SO: BR
2351	006106	012737	027430	021104	MOV	#WDATA,CADER		;SET EXPT BUS ADDRESS
2352	006114	012737	000001	021112	MOV	#1,DRVER		;INDICATE ERROR
2353	006122	004737	020206		JSR	PC,ERPT		;PRINT TM ERROR
2354	006126	000404			BR	WTM2		
2355	006130	012703	027430		MOV	#WDATA,R3		;SET EXPT ADDRESS
2356	006134	004737	017456		JSR	PC,ER2		;GO CHECK FOR OTHER ERRORS
2357	006140	005737	000712		WTM2:	TST	RTYFL	;SEE IF RETRY
2358	006144	001401			BEQ	WTM3		;IF NOT: BR
2359	006146	000207			RTS	PC		;ELSE RETURN TO RETRY ROUTINE
2360	006150	005737	000706		WTM3:	TST	SERFL	;SEE IF WRITE ERROR
2361	006154	001446			BEQ	WEX		;IF NOT: BR
2362	006156	013704	000674		MOV	UNP,R4		
2363	006162	005264	001070		INC	WTER1(R4)		;BUMP WRITE ERROR

2364	006166	032777	000020	172412	BIT	#20,@SWR	;SEE IF SHOULD RETRY
2365	006174	001436			BEQ	WEX	;IF NOT: BR
2366	006176	013703	000722		MOV	ERSAV,R3	
2367	006202	042703	102700		BIC	#102700,R3	;MASK UNRECOVERABLE ERROR
2368	006206	001410			BEQ	WTM4	;IF SO: BR
2369	006210	004737	022662		JSR	PC,PAPRT	;PRINT HEADER
2370	006214	012704	026516		MOV	#MSG78,R4	
2371	006220	104000			TTOUTT		;PRINT UNRETRYABLE TAG
2372	006222	004737	011232		JSR	PC,NRTP	;PRINT ER FOR NON-RETRYABLE
2373	006226	000421			BR	WEX	
2374	006230	005037	000700	WTM4:	CLR	RPCNT	;CLEAR REPEAT CNTR
2375	006234	013704	000674		MOV	UNP,R4	
2376	006240	005264	001050		INC	RTY1(R4)	;BUMP RETRY CNTR
2377	006244	005037	000702		CLR	RTCNT	;CLEAR RETRY CNTR
2378	006250	032777	002000	172330	BIT	#2000,@SWR	;SEE IF PRINT ERRORS
2379	006256	001003			BNE	WTM4A	;IF NOT: BR
2380	006260	012704	026232		MOV	#MSG64,R4	
2381	006264	104000			TTOUTT		;PRINT ORIGINAL ERROR TAG
2382	006266	004737	006334	WTM4A:	JSR	PC,WRTY	;GO DO RETRY
2383	006272	005037	000712	WEX:	CLR	RTYFL	;CLEAR RETRY FLAG
2384	006276	005037	000676		CLR	TMFLG	;CLEAR TAPE MARK FLAG
2385	006302	005737	000660		TST	EOTREC	;+G BRANCH IF NOT AT EOT
2386	006306	100011			BPL	WRWX	;+G
2387	006310	017703	172272	WRW:	MOV	@SWR,R3	
2388	006314	042703	177763		BIC	#177763,R3	
2389	006320	022703	000014		CMP	#14,R3	;SEE IF WRITE ONLY
2390	006324	001002			BNE	WRWX	;IF NOT: BR
2391	006326	000137	004402		JMP	REOT	;ELSE REWIND
2392	006332	000207		WRWX:	RTS	PC	;EXIT

```

2393                                     :*****
2394                                     :WRITE ERROR RETRY
2395                                     :*****
2396
2397
2398 006334 012737 000001 000712 WRTY:  MOV #1,RTYFL ;SET RETRY FLAG
2399 006342 004737 006736 WRTY0: JSR PC,WRTSB ;GO SPACE REVERSE FOR REPEAT
2400 006346 005737 000676          TST TMFLG ;SEE IF TAPE MARK TIME
2401 006352 001003          BNE WRTYTM ;IF SO: BR
2402 006354 004737 005514          JSR PC,W0 ;REWRITE RECORD
2403 006360 000402          BR WRTYR ;GO ON
2404 006362 004737 006024 WRTYTM: JSR PC,WTM ;GO WRITE TAPE MARK AGAIN
2405 006366 005737 000706 WRTYR: TST SERFL ;REWRITE GOOD
2406 006372 001024          BNE WRTY2 ;IF NOT: BR
2407 006374 005237 000700          INC RPCNT ;BUMP REPEAT COUNTER
2408 006400 022737 000004 000700          CMP #4,RPCNT ;SEE IF FOUR GOOD REPEATS
2409 006406 001355          BNE WRTY0 ;IF NOT: REPEAT
2410 006410 032777 002000 172170          BIT #2000,@SWR ;SEE IF PRINT
2411 006416 001011          BNE WRTY1 ;IF NOT: BR
2412 006420 012704 026711          MOV #MSG105,R4
2413 006424 104000          TTOUTT ;PRINT RECOVERED MESSAGE
2414 006426 012704 026254          MOV #MSG65,R4
2415 006432 104000          TTOUTT ;PRINT RETRY TAG
2416 006434 013703 000702          MOV RTCNT,R3
2417 006440 104002          OCTPP ;PRINT RETRY NUMBER
2418 006442 000207 WRTY1: RTS PC ;RESUME TESTING
2419 006444 005037 000646 WRTY2: CLR TEMP3 ;++G CLEAR RECOVERABLE ERROR FLAG
2420 006450 013703 000722          MOV ERSAV,R3 ;GET ER
2421 006454 042703 102700          BIC #102700,R3 ;MASK RECOVERABLE BITS
2422 006460 001413          BEQ WRTY2A ;IF RECOVERABLE: BR
2423 006462 004737 022662          JSR PC,PAPRT ;PRINT HEADER
2424 006466 012704 026516          MOV #MSG78,R4
2425 006472 104000          TTOUTT ;PRINT NON-RECOVERABLE MSG
2426 006474 004737 011232          JSR PC,NRTP ;PRINT ER
2427 006500 012737 000001 000646          MOV #1,TEMP3 ;SET FLAG
2428 006506 000407          BR WRTY2B
2429 006510 032777 002000 172070 WRTY2A: BIT #2000,@SWR ;SEE IF PRINT
2430 006516 001025          BNE WRTY3 ;IF NOT: BR
2431 006520 012704 027145          MOV #MSG110,R4
2432 006524 104000          TTOUTT ;PRINT BAD TAPE SUSPECT
2433 006526 012704 026254 WRTY2B: MOV #MSG65,R4
2434 006532 104000          TTOUTT ;PRINT RETRY TAG
2435 006534 013703 000702          MOV RTCNT,R3
2436 006540 104002          OCTPP ;PRINT RETRY NUMBER
2437 006542 012704 027167          MOV #MSG111,R4
2438 006546 104000          TTOUTT ;PRINT REPEAT TAG
2439 006550 013703 000700          MOV RPCNT,R3
2440 006554 104002          OCTPP ;PRINT REPEAT NUMBER
2441 006556 005737 000646          TST TEMP3 ;SEE IF DID NON-RECOVERABLE
2442 006562 001403          BEQ WRTY3 ;IF NOT: BR
2443 006564 005037 000646          CLR TEMP3 ;CLEAR FLAG
2444 006570 000207          RTS PC ;EXIT
2445 006572 005737 000702 WRTY3: TST RTCNT ;SEE IF FIRST RETRY
2446 006576 001004          BNE WRTY3A ;IF NOT: BR
2447 006600 013704 000674          MOV UNP,R4
2448 006604 005364 001070          DEC WTER1(R4) ;DECREMENT WRITE ERROR CNTR

```

```

2449 006610 013704 000674          WRTY3A: MOV      UNP,R4          ;GET UNIT NUMBER
2450 006614 016437 001030 000730  MOV      BTADDR(R4),BTPT ;GET ADDRESS OF UNIT BAD TAPE CNTR
2451 006622 017704 172102          MOV      @BTPT,R4        ;GET COUNTER
2452 006626 005724          TST      (R4)+           ;SET POINTER OFFSET
2453 006630 010477 172074          MOV      R4,@BTPT
2454 006634 013703 000730          MOV      BTPT,R3
2455 006640 060304          ADD      R3,R4           ;SET ABSOLUTE POINTER
2456 006642 013714 000654          MOV      BLCNTR,(R4)    ;SET BLOCK NUMBER
2457 006646 062704 000040          ADD      @40,R4         ;ADD RCNT OFFSET
2458 006652 013714 000554          MOV      RCNT,(R4)
2459 006656 160014          SUB      R0,(R4)        ;SET RECORD NUMBER
2460 006660 005214          INC      (R4)           ;CORRECT RECORD NUMBER
2461 006662 022777 000040 172040          CMP      @40,@BTPT     ;SEE IF TOO MANY BAD SPOTS
2462 006670 001002          BNE     WRTY4           ;IF NOT: BR
2463 006672 000137 007132          JMP      BTOV           ;ELSE GO TO BAD TAPE OVERFLOW
2464 006676 005237 000702          WRTY4: INC      RTCNT     ;BUMP RETRY COUNTER
2465 006702 022737 000004 000702          CMP      @4,RTCNT     ;SEE IF DONE 4 RETRIES
2466 006710 001410          BEQ     WRTY5           ;IF 50: BR
2467 006712 013704 000674          MOV      UNP,R4
2468 006716 005264 001050          INC      RTY1(R4)      ;BUMP RETRY COUNTER
2469 006722 005237 000732          INC      ERTFL         ;SET ERASE FLAG
2470 006726 000137 006342          JMP      WRTY0         ;DO NEXT RETRY
2471 006732 000137 007352          WRTY5: JMP      BTUR         ;ELSE GO TO BAD TAPE UNRECOVERABLE
2472
2473          ;WRITE RETRY BACKSPACE-ERASE SUBROUTINE*****
2474
2475 006736 005037 000706          WRTSB: CLR      SERFL        ;CLEAR FLAG
2476 006742 013737 000576 000666          MOV      TSTAL,STAL
2477 006750 004737 012256          JSR     PC,STALL       ;DO TURN AROUND DELAY
2478 006754 012737 026265 000652          MOV      @MSG66,EMADOR ;SET ERROR CODE
2479 006762 012777 177777 171526          MOV      @-1,@FC      ;SET TO BACKSPACE 1 RECORD
2480 006770 012777 033436 171516          MOV      @RDATA,@BA   ;SET BA
2481 006776 004737 012206          JSR     PC,BKRT       ;GO BACKSPACE
2482 007002 005737 000706          TST      SERFL        ;SEE IF ERROR
2483 007006 001406          BEQ     WRTSB1        ;IF NOT: BR
2484 007010 012737 000002 000724          WRTSB0: MOV      @2,BTFLG   ;SET FLAG
2485 007016 022626          CMP      (SP)+,(SP)+  ;RESET STACK
2486 007020 000137 004402          JMP      REOT         ;GO REWIND AND REMOVE FROM TESTING
2487 007024 005737 000732          WRTSB1: TST      ERTFL   ;SEE IF SHOULD ERASE
2488 007030 001001          BNE     WRTSB2        ;IF 50: BR
2489 007032 000207          RTS     PC            ;RETURN
2490 007034 005037 000732          WRTSB2: CLR      ERTFL   ;CLEAR ERASE FLAG
2491 007040 005037 000700          CLR      RPNCT       ;CLEAR REPEAT CNTR
2492 007044 005037 000706          CLR      SERFL        ;CLEAR FLAG
2493 007050 012737 026300 000652          MOV      @MSG67,EMADOR ;SET ERROR CODE
2494 007056 005077 171434          CLR      @FC         ;CLEAR FRAME COUNT
2495 007062 012737 000024 000672          MOV      @24,HTC1     ;SET ERASE OP-CODE
2496 007070 012777 027430 171416          MOV      @RDATA,@BA   ;SET BA
2497 007076 012737 007110 000662          MOV      @WRTSB3,RTRN ;SET RETURN ADDRESS
2498 007104 000137 021170          JMP      TAPG         ;GO ERASE
2499 007110 012703 027430          WRTSB3: MOV      @RDATA,R3 ;SET EXPT BA
2500 007114 004737 017456          JSR     PC,ER2       ;GO CHECK ERRORS
2501 007120 005737 000706          TST      SERFL        ;SEE IF ERROR
2502 007124 001737          BEQ     WRTSB1        ;IF NOT: BR
2503 007126 000137 007010          JMP      WRTSBC
    
```

;BAD TAPE OVERFLOW SUBROUTINE\*\*\*\*\*

```

2505
2506
2507 007132 005037 000712      BTOV:  CLR      RTYFL      ;CLEAR RETRY FLAG
2508 007136 012737 000001 000724  MOV      #1,BTFLG     ;SET BAD TAPE OVERFLOW FLAG
2509 007144 005726                TST      (SP)+        ;++G ADJUST STACK
2510 007146 000137 004402                JMP      REOT         ;GO REWIND AND REMOVE FROM TESTING
2511 007152 013701 000730      BTOV0:  MOV      BTPT,R1   ;SET TABLE POINTER
2512 007156 005721                TST      (R1)↓
2513 007160 005000                CLR      R0
2514 007162 010003      BTOV1:  MOV      R0,R3
2515 007164 000241                CLC
2516 007166 006003                ROR      R3           ;R3=R3/2 FOR CORRECT NUMBER
2517 007170 104002                OCTPP                    ;PRINT ENTRY NUMBER
2518 007172 012704 025000      MOV      #MSG13,R4
2519 007176 105724                TSTB     (R4)+        ;SKIP CR/LF
2520 007200 104000                TTOUTT                    ;PRINT BLOCK NUMBER TAG
2521 007202 011103                MOV      (R1),R3
2522 007204 104002                OCTPP                    ;PRINT BLOCK NUMBER
2523 007206 012704 025006      MOV      #MSG14,R4
2524 007212 104000                TTOUTT                    ;PRINT RECORD NUMBER TAG
2525 007214 062701 000040      ADD      #40,R1       ;SET POINTER OFFSET FOR RECOED NUMBER
2526 007220 012103                MOV      (R1)+,R3
2527 007222 104002                OCTPP                    ;PRINT RECORD NUMBER
2528 007224 162701 000040      SUB      #40,R1       ;RESET POINTER FOR BLOCK NUMBER
2529 007230 005720                TST      (R0)+
2530 007232 020077 171472      CMP      R0,BTPT     ;SEE IF DONE
2531 007236 001404                BEQ      BTOV2        ;IF SO: BR
2532 007240 012704 025331      MOV      #MSG28,R4
2533 007244 104000                TTOUTT                    ;DO CR/LF
2534 007246 000745                BR
2535 007250 005737 000726      BTOV2:  TST      BTSTF     ;CONTINUE
2536 007254 001007                BNE     BTOVX         ;SEE IF STAT ONLY PRINT
2537 007256 012703 000041      MOV      #41,R3       ;IF SO: BR
2538 007262 013704 000730      MOV      BTPT,R4      ;SET SIZE OF TABLE
2539 007266 005024                BTOV3:  CLR      (R4)↓    ;SET POINTER
2540 007270 005303                DEC      R3           ;CLEAR TABLE
2541 007272 001375                BNE     BTOV3        ;SEE IF DONE
2542 007274 000207                BTOVX:  RTS      BTOV3 ;IF NOT: BR
2543                PC
    
```



E05

```

2544
2545
2546
2547 007276 012704 025331
2548 007302 104000
2549 007304 013704 000674
2550 007310 016437 001030 000730
2551 007316 017703 171406
2552 007322 000241
2553 007324 006003
2554 007326 104002
2555 007330 012704 027201
2556 007334 104000
2557 007336 005777 171366
2558 007342 001001
2559 007344 000207
2560 007346 000137 007152
2561
2562
2563
2564 007352 004737 022662
2565 007356 012704 027014
2566 007362 104000
2567 007364 000207
2568

```

```

;BAD TAPE STATISTIC PRINT*****
BTPRT: MOV     #MSG28,R4
        TTOUTT
        ;DO CR/LF
        MOV     UNP,R4
        MOV     BTADDR(R4),BTPT ;SET TABLE POINTER
        MOV     @BTPT,R3
        CLC
        ROR     R3
        ;CORRECT NUMBER
        OCTPP
        ;PRINT NUMBER OF BAD SPOTS
        MOV     #MSG112,R4
        TTOUTT
        ;PRINT BAD TAPE TAG
        TST     @BTPT
        ;SEE IF ANY BAD SPCTS
        BNE     BTPRT1
        ;IF SO: BR
        RTS     PC
        ;ELSE RETURN
        ;PRINT STATS
BTPRT1: JMP     BTOVO

;BAD TAPE UNRECOVERABLE SUBROUTINE*****
BTUR:  JSR     PC,PAPRT
        ;PRINT HEADER
        MOV     #MSG107,R4
        TTOUTT
        ;PRINT UNRECOVERABLE BAD SPOT MSG
        RTS     PC
        ;RESUME TESTING

```

# F05

```

2569          ;*****
2570          ;READ SEQUENCER:
2571          ;
2572          ;THIS ROUTINE IS USED TO DETERMINE THE SEQUENCE
2573          ;IN WHICH READ TAPE OPERATIONS ARE TO BE PERFORMED.
2574          ;THIS IS NECESSARY WHEN THE UNIT BEING TESTED IS
2575          ;CAPABLE OF READING DATA IN BOTH THE FORWARD AND
2576          ;REVERSE DIRECTIONS.  CONSOLE SWITCHES ONE (1), TWO (2),
2577          ;AND THREE (3) ARE USED TO DETERMINE THE READ SEQUENCE.
2578          ;CONSOLE SWITCH ONE (1) DETERMINES WHETHER TO READ
2579          ;THE BLOCK OF DATA FORWARD FIRST OR REVERSE FIRST.
2580          ;SWITCH TWO (2) DISALLOWS READING IN THE REVERSE
2581          ;DIRECTION AND SWITCH THREE (3) DISALLOWS READING IN
2582          ;THE FORWARD DIRECTION.
2583          ;*****
2584
2585 007366 012737 000002 000562 RSEQ:  MOV    #2,RDCMD
2586 007374 017704 171206          MOV    #SWR,R4          ;READ SWITCHES
2587 007400 042704 177763          BIC    #177763,R4      ;MASK READ BITS
2588 007404 005704          TST    R4              ;SEE IF BOTH READS
2589 007406 001004          BNE    RSR              ;IF NOT: BR
2590 007410 032777 000002 171170  BIT    #2,SWR          ;SEE IF READ REVERSE FIRST
2591 007416 001051          BNE    RSFR             ;IF NOT: BR
2592 007420 032777 000004 171160  RSR:   BIT    #4,SWR          ;SEE IF SHOULD READ REVERSE
2593 007426 001005          BNE    RSF              ;IF NOT: BR
2594 007430 012737 010000 000562  MOV    #10000,RDCMD    ;LOAD READ REVERSE COMMAND
2595 007436 004737 007704          JSR    PC,READ         ;GO READ REVERSE
2596 007442 032777 000010 171136  RSF:   BIT    #10,SWR         ;SEE IF SHOULD READ FORWARD
2597 007450 001026          BNE    RSEX             ;IF NOT: BR
2598 007452 032737 010000 000562  BIT    #10000,RDCMD    ;SEE IF HAVE READ REVERSE
2599 007460 001407          BEQ    RSFO             ;IF NOT: BR
2600 007462 013737 000576 000666  MOV    TSTAL,STAL      ;DO READ STALL
2601 007470 004737 012256          JSR    PC,STALL
2602 007474 000137 007514          JMP    RSF1
2603 007500 032777 000001 171100  RSFO:  BIT    #1,SWR          ;SEE IF WRITE
2604 007506 001002          BNE    RSF1             ;IF NOT: BR
2605 007510 004737 012030          JSR    PC,BKSP         ;GO BACKSPACE
2606 007514 012737 000002 000562  RSF1:  MOV    #2,RDCMD         ;LOAD READ FORWARD COMMAND
2607 007522 004737 007704          JSR    PC,READ         ;GO READ
2608 007526 005737 000660          RSEX:  TST    EOTREC      ;++G BRANCH IF NOT AT EOT
2609 007532 100002          BPL    IS              ;++G
2610 007534 000137 004402          JMP    REOT            ;++G ELSE GO REWIND
2611 007540 000207          IS:   RTS    PC           ;++G EXIT
2612
2613 007542 012737 010000 000562  RSFR:  MOV    #10000,RDCMD
2614 007550 032777 000010 171030  BIT    #10,SWR         ;SEE IF SHOULD READ FORWARD
2615 007556 001013          BNE    RSFR1           ;IF NOT: BR
2616 007560 032777 000001 171020  BIT    #1,SWR          ;SEE IF WRITE
2617 007566 001002          BNE    RSFR0           ;IF NOT: BR
2618 007570 004737 012030          JSR    PC,BKSP         ;GO BACKSPACE TO START
2619 007574 012737 000002 000562  RSFR0: MOV    #2,RDCMD         ;LOAD READ FORWARD COMMAND
2620 007602 004737 007704          JSR    PC,READ         ;GO READ FORWARD
2621 007606 032777 000004 170772  RSFR1: BIT    #4,SWR          ;SEE IF SHOULD READ REVERSE
2622 007614 001344          BNE    RSEX            ;IF NOT: BR
2623 007616 032737 010000 000562  BIT    #10000,RDCMD
2624 007624 001005          BNE    RSFR2           ;IF READ REVERSE: BR
    
```

G05

TM02-TU16/TE16 DATA RELIABILITY PROGRAM MACY11 30(1046) 22-JUN-77 08:44 PAGE 58  
 DZTUAG.P11 17-JUN-77 09:52

SEG 0058

2625	007626	013737	000576	000666		MOV	TSTAL,STAL	;DO READ STALL
2626	007634	004737	012256			JSR	PC,STALL	
2627	007640	012737	010000	000562	RSFR2:	MOV	#10000,RDCMD	:LOAD READ REVERSE
2628	007646	004737	007704			JSR	PC,READ	:GO READ REVERSE
2629	007652	005737	000660			TST	EOTREC	:SEE IF AT END OF TAPE
2630	007656	100011				BPL	RSFRX	:++G IF NOT: BR
2631	007660	163737	000554	000660		SUB	RCNT,EOTREC	
2632	007666	005437	000660			NEG	EOTREC	:SET TO PROPER RECORD NUMBER
2633	007672	005237	000660			INC	EOTREC	
2634	007676	000137	004402			JMP	REOT	:ELSE GO TO REWIND
2635	007702	000207			RSFRX:	RTS	PC	:EXIT
2636								

```

*****
: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)
*****

```

```

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
2663 007704 013700 000554 READ:
2664 007710 005737 000660
2665 007714 100013
2666 007716 032737 010000 000562
2667 007724 001407
2668 007726 042737 100000 000660
2669 007734 013703 000660
2670 007740 160300
2671 007742 005200
2672 007744 012737 024713 000652 RDA:
2673 007752 005037 000676
2674 007756 032737 010000 000562
2675 007764 001406
2676 007766 005737 000564
2677 007772 001403
2678 007774 005237 000676
2679 010000 005200
2680 010002 013777 000556 170506 RDC:
2681 010010 012777 033436 170476
2682 010016 032737 010000 000562
2683 010024 001417
2684 010026 013703 000556
2685 010032 005103
2686 010034 032737 000020 000552
2687 010042 001402
2688 010044 000241
2689 010046 006003
2690 010050 060377 170440 RDI:
2691 010054 012737 000076 000672
2692 010062 000403

```

```

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 #100000,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 R00 ;IF READ FORWARD: BR
TST TMEX ;SEE IF TM
BEQ R00 ;IF NOT: BR
INC TMFLG ;SET TM FLAG
INC R0
RDC: MOV FMCNT,RFC ;LOAD CHAR CNTR
MOV #RDATA,RBA ;LOAD DATA ADDR
BIT #10000,RDCMD ;SEE IF READ REVERSE
BEQ R01A ;IF NOT: BR
MOV FMCNT,R3
COM R3
BIT #20,UDES ;SEE IF CORE DUMP
BEQ R01 ;IF NOT: BR
CLC
ROR R3 ;R3 = FC/2
RDI: ADD R3,RBA ;SET REVERSE BUS ADDRESS
MOV #76,MTC1 ;SET READ REVERSE
BR RDI

```

2693	010064	012737	000070	000672	RD1A:	MOV	#70, MTC1	; SET READ FORWARD
2694	010072	012737	010104	000662	RD1B:	MOV	#RD2, RTRN	; SET INTERRUPT RETURN ADDRESS
2695	010100	000137	021170		RD1D:	JMP	TAPG	; GO EXECUTE TAPE COMMAND
2696	010104	032737	010000	000562	RD2:	BIT	#10000, RDCMD	; SEE IF READ REVERSE
2697	010112	001024				BNE	RD3	; IF S0: BR
2698	010114	032777	000020	170400		BIT	#20, JDS	
2699	010122	001404				BEQ	RD2B	; AWAIT SWDN
2700	010124	032777	000020	170370	RD2A:	BIT	#20, JDS	
2701	010132	001374				BNE	RD2A	; AWAIT TUR
2702	010134	032777	002000	170360	RD2B:	BIT	#2000, JDS	; SEE IF EOT
2703	010142	001410				BEQ	RD3	; IF NOT: BR
2704	010144	005737	000676			TST	TMFLG	; SEE IF TM
2705	010150	001005				BNE	RD3	; IF S0: BR
2706	010152	010037	000660			MOV	RD, EOTREC	
2707	010156	052737	100000	000660		BIS	#100000, EOTREC	; SET EOT FLAG
2708	010164	032777	000002	170330	RD3:	BIT	#2, JDS	; SEE IF AT LOAD POINT
2709	010172	001410				BEQ	RD4	; IF NOT: BR
2710	010174	004737	022662			JSR	PC, PAPRT	; PRINT CYCLE NUMBER
2711	010200	012704	025112			MOV	#MSG22, R4	
2712	010204	104000				TTOUTT		; PRINT BOT ERROR
2713	010206	104006				STOPP		
2714	010210	000137	003160			JMP	STARTA	; RESTART
2715	010214	032777	004000	170364	RD4:	BIT	#4000, JSWR	; SEE IF SHOULD CHECK ERRORS
2716	010222	001121				BNE	RD5	; IF NOT: BR
2717	010224	005737	000676			TST	TMFLG	
2718	010230	001472				BEQ	RD4B	; IF NO TM EXPT: BR
2719	010232	032777	000004	170262		BIT	#4, JDS	
2720	010240	001024				BNE	RD4A	; IF TM RECVD: BR
2721	010242	012737	033436	021104		MOV	#RDATA, CADER	; SAVE EXPT BUS ADDRESS
2722	010250	012737	000002	021112		MOV	#2, DRIVER	; SET TM STATUS ERROR FLAG
2723	010256	004737	020206			JSR	PC, ERPT	; GO PRINT TM ERROR
2724	010262	013704	000674			MOV	UMP, R4	
2725	010266	032737	010000	000562		BIT	#10000, RDCMD	; SEE IF READ REVERSE
2726	010274	001403				BEQ	1\$	; IF NOT: BR
2727	010276	005264	001150			INC	RDERR1(R4)	; BUMP READ REVERSE ERROR
2728	010302	000502				BR	RD6	
2729	010304	005264	001110		1\$:	INC	RDERR1(R4)	; BUMP READ FORWARD ERROR
2730	010310	000477				BR	RD6	
2731	010312	012703	033436		RD4A:	MOV	#RDATA, R3	
2732	010316	032737	010000	000562		BIT	#10000, RDCMD	; SEE IF READ REVERSE
2733	010324	001007				BNE	RD4A0	; IF S0: BR
2734	010326	032737	002000	000552		BIT	#2000, UDES	; SEE IF IN PE
2735	010334	001025				BNE	RD4A2	; IF S0: BR
2736	010336	062703	000002			ADD	#2, R3	
2737	010342	000422				BR	RD4A2	
2738	010344	013704	000556		RD4A0:	MOV	FMCNT, R4	
2739	010350	005104				COM	R4	
2740	010352	032737	000020	000552		BIT	#20, UDES	; SEE IF CORE DUMP
2741	010360	001402				BEQ	RD4A1	; IF NOT: BR
2742	010362	000241				CLC		
2743	010364	006004				ROR	R4	; SET TO FC/2
2744	010366	060403			RD4A1:	ADD	R4, R3	; SET EXPT BUS ADDRESS
2745	010370	042703	000001			BIC	#1, R3	; MAKE EXPT ADDRESS EVEN
2746	010374	032737	002000	000552		BIT	#2000, UDES	; SEE IF IN PE
2747	010402	001002				BNE	RD4A2	; IF S0: BR
2748	010404	162703	000002			SUB	#2, R3	

# J05

TM02-TU16/TE16 DATA RELIABILITY PROGRAM MACY11 30(1046) 22-JUN-77 08:44 PAGE 61  
 DZTUAG.P11 17-JUN-77 09:52

SEQ 0061

2749	010410	004737	017456		RD4A2:	JSR	PC,ER2	
2750	010414	000402				BR	RD4C	
2751	010416	004737	017360		RD4B:	JSR	PC,ERCHK	;GO CHECK ERRORS
2752	010422	005737	000706		RD4C:	TST	SEFL	
2753	010426	001417				BEQ	R05	;IF NO ERROR: BR
2754	010430	013704	000674			MOV	UNP,R4	
2755	010434	032737	010000	000562		BIT	#10000,RDCMD	;SEE IF READ REVERSE
2756	010442	001003				BNE	R04D	;IF SO: BR
2757	010444	005264	001110			INC	RDER1(R4)	;BUMP READ FORWARD ERROR
2758	010450	000402				BR	R04E	
2759	010452	005264	001150		RD4C:	INC	RDER1(R4)	;BUMP READ REVERSE ERROR
2760	010456	004737	010660		RD4E:	JSR	PC,RDRTY	;GO RETRY
2761	010462	005037	000712			CLR	RTYFL	;CLEAR RETRY FLAG
2762	010466	032777	020000	170112	R05:	BIT	#20000,ASWR	;SEE IF SHOULD DO DATA CHECK
2763	010474	001005				BNE	R06	;IF NOT; BR
2764	010476	005737	000676			TST	TMFLG	
2765	010502	001002				BNE	R06	
2766	010504	004737	015516			JSR	PC,DCHK	;GO CHECK DATA
2767	010510	005037	000706		R06:	CLR	SEFL	;CLEAR STATUS ERROR FLAG
2768	010514	004737	014324			JSR	PC,D53	;CLEAR BUFFER
2769	010520	032777	000040	170060		BIT	#40,ASWR	;SEE IF SHOULD YOZZLE
2770	010526	001402				BEQ	R07	;IF NOT: BR
2771	010530	004737	011246			JSR	PC,YOZ	;ELSE GO YOZZLE
2772	010534	013737	000572	000666	R07:	MOV	RSTAL,STAL	;SET DELAY
2773	010542	004737	012256			JSR	PC,STALL	;STALL
2774	010546	032737	010000	000562		BIT	#10000,RDCMD	;SEE IF READ REVERSE
2775	010554	001403				BEQ	R07A	;IF NOT: BR
2776	010556	005037	000676			CLR	TMFLG	;CLEAR TAPE MARK FLAG
2777	010562	000405				BR	R010	
2778	010564	005737	000660		RD7A:	TST	EOTREC	;SEE IF EOT FOUND
2779	010570	100002				BPL	R010	;IF NOT: BR
2780	010572	012700	000001			MOV	#1,R0	;SET TO EOT
2781	010576	005300			RD10:	DEC	R0	
2782	010600	001402				BEQ	R011	;IF DONE ALL: BR
2783	010602	000137	010002			JMP	R00	
2784	010606	032737	010000	000562	RD11:	BIT	#10000,RDCMD	;SEE IF READ REVERSE
2785	010614	001016				BNE	RDEX	;IF SO: BR
2786	010616	005737	000660			TST	EOTREC	;SEE IF FOUND EOT
2787	010622	100413				BMI	RDEX	;IF SO: BR
2788	010624	005737	000564			TST	TMEX	;SEE IF TM EXPECTED
2789	010630	001410				BEQ	RDEX	;IF NOT: BR
2790	010632	005737	000676			TST	TMFLG	;SEE IF TM FOUND
2791	010636	001005				BNE	RDEX	;IF SO: BR
2792	010640	005237	000676			INC	TMFLG	;ELSE SET FLAG
2793	010644	005200				INC	R0	;SET RECORD COUNT TO ONE
2794	010646	000137	010002			JMP	R00	;GO READ TM
2795	010652	005037	000676		RDEX:	CLR	TMFLG	
2796	010656	000207			R0X:	RTS	PC	;EXIT

```

2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808 010660 032777 000020 167720 RDRTY: BIT #20,JSWR ;SEE IF RETRY INHIBITED
2809 010666 001001 BNE RDRT0 ;IF NOT: BR
2810 010670 000207 RTS PC ;ELSE RETURN
2811 010672 013703 000722 RDRT0: MOV ERSAV,R3
2812 010676 042703 102700 BIC #102700,R3 ;MARK NON-RECOVERABLE ERROR BITS
2813 010702 001410 BEQ RDRT1 ;IF NOT: BR
2814 010704 004737 022662 JSR PC,PAPRT ;PRINT HEADER
2815 010710 012704 026557 MOV #MSG79,R4
2816 010714 104000 TTOUTT ;PRINT NON-RECOVERABLE MESSAGE
2817 010716 004737 011232 JSR PC,NRTP ;PRINT ER FOR NON-RETRYABLE ERROR
2818 010722 000207 RTS PC ;RETURN
2819 010724 032777 002000 167654 RDRT1A: BIT #2000,JSWR ;SEE IF PRINT INHIBITED
2820 010732 001003 BNE RDRT1B ;IF SO: BR
2821 010734 012704 026232 MOV #MSG64,R4
2822 010740 104000 TTOUTT ;PRINT ORIGINAL ERROR TAG
2823 010742 005037 000702 RDRT1B: CLR RTCNT ;CLEAR RETRY COUNTER
2824 010746 005037 000706 RDRTG: CLR SERFL ;CLEAR STATUS ERROR FLAG
2825 010752 012737 000002 000712 MOV #2,RTYFL ;SET READ RETRY FLAG
2826 010760 004737 011246 JSR PC,YOZ ;GO TO YOZZLE TO RETRY READ
2827 010764 005737 000706 TST SERFL ;SEE IF RETRY ERROR
2828 010770 001031 BNE RDRT5 ;IF SO: BR
2829 010772 032777 002000 167606 BIT #2000,JSWR
2830 011000 001011 BNE RDRT2
2831 011002 012704 026711 MOV #MSG105,R4
2832 011006 104000 TTOUTT ;PRINT RECOVERED MESSAGE
2833 011010 012704 026254 MOV #MSG65,R4
2834 011014 104000 TTOLTT ;PRINT RETRY TAG
2835 011016 013703 000702 MOV RTCNT,R3
2836 011022 104002 OCTPP ;PRINT RETRY NUMBER
2837 011024 013704 000674 RDRT2: MOV UNP,R4
2838 011030 032737 010000 000562 BIT #10000,RDCMD ;SEE IF READ REVERSE
2839 011036 001003 BNE RDRT3 ;IF SO: BR
2840 011040 005264 002670 INC RFSOFT(R4) ;ELSO BUMP FORWARD SOFT ERROR COUNTER
2841 011044 000402 BR RDRT4
2842 011046 005264 002710 RDRT3: INC RRSOFT(R4) ;BUMP ERRORS SOFT CNTR
2843 011052 000207 RDRT4: RTS PC ;RETURN
2844 011054 005037 000646 RDRT5: CLR TEMP3 ;++G CLEAR RECOVERABLE ERROR INDICATOR
2845 011060 013703 000722 MOV ERSAV,R3 ;GET ER
2846 011064 042703 102700 BIC #102700,R3 ;MASK RECOVERABLE BITS
2847 011070 001413 BEQ RDRT5A ;IF RECOVERABLE: BR
2848 011072 004737 022662 JSR PC,PAPRT ;PRINT HEADER
2849 011076 012704 026557 MOV #MSG79,R4
2850 011102 104000 TTOLTT ;PRINT NON-RECOVERABLE MSG
2851 011104 004737 011232 JSR PC,NRTP ;PRINT ER
2852 011110 012737 000001 000646 MOV #1,TEMP3 ;SET FLAG

```

2853	011116	000404			BR	RDRT5B	
2854	011120	032777	002000	167460	RDRT5A: BIT	#2000,JSWR	;SEE IF PRINT INHIBITED
2855	011126	001014			BNE	RDRT6	;IF SO: BR
2856	011130	012704	026254		RDRT5B: MOV	#MSG65,R4	
2857	011134	104000			TTOUTT		;PRINT RETRY TAG
2858	011136	013703	000702		MOV	RTCNT,R3	
2859	011142	104002			OCTPP		;PRINT RETRY NUMBER
2860	011144	005737	000646		TST	TEMP3	;SEE IF DID NON-RECOVERABLE
2861	011150	001403			BEQ	RDRT6	;IF NOT: BR
2862	011152	005037	000646		CLR	TEMP3	;CLEAR FLAG
2863	011156	000207			RTS	PC	;EXIT
2864	011160	005237	000702		RDRT6: INC	RTCNT	
2865	011164	023737	000702	000602	CMP	RTCNT,RETRY	;SEE IF DONE 8 RETRIES
2866	011172	001265			BNE	RDRTG	;IF NOT: BR
2867	011174	012704	027244		MOV	#MSG115,R4	
2868	011200	104000			TTOUTT		;PRINT HARD ERROR MESSAGE
2869	011202	013704	000674		MOV	UNP,R4	
2870	011206	032737	010000	000562	BIT	#10000,RDCMD	;SEE IF READ REVERSE
2871	011214	001003			BNE	RDRT7	;IF SO: BR
2872	011216	005264	002730		INC	RFHARD(R4)	;BUMP FORWARD HARD ERROR CNTR
2873	011222	000402			BR	RDRTX	
2874	011224	005264	002750		RDRT7: INC	RRHARD(R4)	;BUMP REVERSE HARD ERROR CNTR
2875	011230	000207			RDRTX: RTS	PC	;RETURN
2876							
2877	011232	013703	000722		NRTP: MOV	ERSAV,R3	;GET ER REGISTER
2878	011236	104002			OCTPP		;PRINT ER
2879	011240	004737	021130		JSR	PC,FRPRT	;PRINT F OR R
2880	011244	000207			RTS	PC	;RETURN



2881  
2882  
2883  
2884  
2885  
2886  
2887  
2888  
2889  
2890  
2891  
2892  
2893  
2894  
2895  
2896  
2897  
2898  
2899  
2900  
2901  
2902  
2903  
2904  
2905  
2906  
2907  
2908  
2909  
2910  
2911  
2912  
2913  
2914  
2915  
2916  
2917  
2918  
2919  
2920  
2921  
2922  
2923  
2924  
2925  
2926  
2927  
2928  
2929  
2930  
2931  
2932  
2933  
2934  
2935  
2936

011246 012777 000001 167334 YOZ:  
011254 013737 000600 000666  
011262 004737 012256  
011266 012777 177777 167222 YOZO:  
011274 032737 010000 000562  
011302 001404  
011304 112737 000030 000672  
011312 000403  
011314 112737 000032 000672 YOZA:  
011322 012737 011342 000662 YOZB:  
011330 012737 177775 000666  
011336 000137 021170  
011342 005737 000676 YOZC:  
011346 001404  
011350 012737 040000 000666  
011356 000403  
011360 013737 000600 000666 15:  
011366 004737 012256 25:  
011372 012777 033436 167114  
011400 032737 010000 000562  
011406 001417  
011410 013703 000556  
011414 005103  
011416 032737 000020 000552  
011424 001402  
011426 000241  
011430 006003  
011432 060377 167056 YOZCO:  
011436 012737 000076 000672  
011444 000403  
011446 012737 000070 000672 YOZC1:  
011454 013777 000556 167034 YOZC2:  
011462 012737 011474 000662  
011470 000137 021170  
011474 032777 004000 167104 YOZD:  
011502 001051  
011504 005737 000676  
011510 001444  
011512 032737 010000 000562  
011520 001426  
011522 012703 033436  
011526 013704 000556  
011532 005104

```
*****  
;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.  
*****  
MOV #1, JTKS ;SET TTY ENABLE  
MOV YSTAL, STAL  
JSR PC, STALL ;DO YOZZLE STALL  
MOV #-1, JFC ;SET TO 1 RECORD SPACING  
BIT #10000, RDCMD ;SEE IF READ REVERSE  
BEQ YOZA ;IF NOT: BR  
MOVB #30, MTC1 ;SET TO SPACE FORWARD  
BR YOZB  
MOVB #32, MTC1 ;SET TO SPACE REVERSE  
MOV #YOZC, RTRN ;SET RETURN ADDRESS  
MOV #177775, STAL ;SET TIME MULTIPLIER  
JMP TAPG ;GO YOZZLE  
TST TMFLG ;SEE IF TM  
BEQ 15 ;IF NOT: BR  
MOV #40000, STAL ;SET TM STALL  
BR 25  
MOV YSTAL, STAL  
JSR PC, STALL ;DO YOZZLE STALL  
MOV #RDATA, JBA ;SET BUS ADDRESS  
BIT #10000, RDCMD ;SEE IF READ REVERSE  
BEQ YOZC1 ;IF NOT: BR  
MOV FMCNT, R3  
COM R3  
BIT #20, UDES ;SEE IF CORE DUMP  
BEQ YOZCO ;IF NOT: BR  
CLC  
ROR R3 ;R3 = FC/2  
ADD R3, JBA ;SET REVERSE BUS ADDRESS  
MOV #76, MTC1 ;SET READ REVERSE  
BR YOZC2  
MOV #70, MTC1 ;SET READ FORWARD  
MOV FMCNT, JFC ;SET CHARACTER COUNT  
MOV #YOZD, RTRN ;SET RETURN ADDRESS  
JMP TAPG ;GO READ  
BIT #4000, JSWR ;SEE IF SHOULD CHECK ERRORS  
BNE YOZE ;IF NOT: BR  
TST TMFLG ;SEE IF TAPE MARK TIME  
BEQ YOZD1 ;IF NOT: BR  
BIT #10000, RDCMD ;SEE IF READ REVERSE  
BEQ YOZD0 ;IF NOT: BR  
MOV #RDATA, R3  
MOV FMCNT, R4  
COM R4
```

2937	011534	032737	000020	000552	BIT	#20, UDES	; SEE IF CORE DUMP
2938	011542	001402			BEG	YOZD4	; IF NOT: BR
2939	011544	000241			CLC		
2940	011546	006004			ROR	R4	; SET TO FC/2
2941	011550	060403			YOZD4: ADD	R4, R3	; SET EXPT BUS ADDRESS
2942	011552	042703	000001		BIC	#1, R3	; MAKE EXPT ADDRESS EVEN
2943	011556	032737	002000	000552	BIT	#2000, UDES	; SEE IF PE
2944	011564	001001			BNE	YOZD2	; IF SO: BR
2945	011566	005743			TST	-(R3)	; SET EXPT BA
2946	011570	004737	017456		YOZD2: JSR	PC, ER2	; GO CHECK ERRORS
2947	011574	000430			BR	YOZF	
2948	011576	012703	033436		YOZD0: MOV	#RDATA, R3	
2949	011602	032737	002000	000552	BIT	#2000, UDES	; SEE IF PE
2950	011610	001001			BNE	YOZD3	; IF SO: BR
2951	011612	005723			TST	(R3)+	; SET EXPT BA
2952	011614	004737	017456		YOZD3: JSR	PC, ER2	; GO CHECK ERRORS
2953	011620	000416			BR	YOZF	
2954	011622	004737	017360		YOZD1: JSR	PC, ERCHK	; ELSE GO CHECK ERRORS
2955	011626	032777	020000	166752	YOZE: BIT	#20000, @SWR	; SEE IF SHOULD CHECK DATA
2956	011634	001010			BNE	YOZF	; IF NOT: BR
2957	011636	005737	000676		TST	TMFLG	; SEE IF TAPE MARK
2958	011642	001005			BNE	YOZF	; IF SO: BR
2959	011644	005737	000712		TST	RTYFL	; SEE IF RETRY
2960	011650	001004			BNE	YOZFO	; IF SO: BR
2961	011652	004737	015516		JSR	PC, DCHK	; ELSE GO CHECK DATA
2962	011656	004737	014324		YOZF: JSR	PC, DS3	; GO CLEAR DATA AREA
2963	011662	105777	166722		YOZFO: TSTB	@TKS	; SEE IF HAVE NEW STALL VALUE
2964	011666	100032			BPL	YOZG	; IF NOT: BR
2965	011670	122777	000203	166714	CMPB	#203, @TKB	; SEE IF CONT C
2966	011676	001026			BNE	YOZG	; IF NOT: BR
2967	011700	012704	025734		MOV	#MSG44, R4	
2968	011704	104000			TTOUTT		; PRINT YSTALL REQUEST
2969	011706	013703	000600		MOV	YSTAL, R3	
2970	011712	104002			OCTPP		; PRINT PRESENT STALL
2971	011714	010037	000646		MOV	RO, TEMP3	; SAVE RO(REC CNT)
2972	011720	012705	000600		MOV	#YSTAL, R5	; SET ADDRESS OF YSTL
2973	011724	012701	000006		MOV	#6, R1	; SET NUMBER OF CHAR TO INPUT
2974	011730	012702	177777		MOV	#-1, R2	; SET MAXIMUM LIMIT
2975	011734	012703	002000		MOV	#2000, R3	; SET MINIMUM LIMIT
2976	011740	004737	023300		JSR	PC, TTR	; GO GET VALUE
2977	011744	013700	000646		MOV	TEMP3, RO	; RESTORE RO(REC CNTR)
2978	011750	000137	011246		JMP	YOZ	; RESTART YOZZLE
2979	011754	122777	000207	166630	YOZG: CMPB	#207, @TKB	; CHECK FOR CNTL G
2980	011762	001010			BNE	YOZI	
2981	011764	022737	000176	000606	CMP	#SWREG, SWR	; IS SWREG SELECTED
2982	011772	001004			BNE	YOZI	
2983	011774	005077	166612		CLR	@TKB	; CLEAR CNTL G OUT OF BUFFER
2984	012000	004737	024376		JSR	PC, CNTG	; GO CHANGE SWREG
2985	012004	032777	000040	166574	YOZI: BIT	#40, @SWR	; SEE IF SHOULD CONTINUE YOZZLE
2986	012012	001402			BEG	YOZH	; IF NOT: BR
2987	012014	000137	011266		JMP	YOZO	
2988	012020	012777	000100	166562	YOZH: MOV	#100, @TKS	; SET TTY INTERRUPT ENABLE
2989	012026	000207			RTS	PC	; EXIT
2990							

2991  
2992  
2993  
2994  
2995  
2996  
2997  
2998  
2999  
3000  
3001  
3002  
3003  
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  
3042

012030 013737 000576 000666 BKSP:  
012036 004737 012256  
012042 012737 024743 000652  
012050 012777 033436 166436  
012056 005737 000564  
012062 001440  
012064 012777 177777 166424  
012072 012737 000032 000672  
012100 012737 012112 000662  
012106 000137 021170  
012112 032777 010000 166466 BKTM:  
012120 001021  
012122 012737 026146 000652  
012130 032777 000004 166364  
012136 001006  
012140 012737 033436 021104  
012146 004737 020206  
012152 000404  
012154 012703 033436 BKTMO:  
012160 004737 017456  
012164 013700 000554 BO:  
012170 005100  
012172 005200  
012174 012737 024743 000652  
012202 010077 166310  
012206 012737 000032 000672 BKRT:  
012214 012737 012232 000662  
012222 010037 000666  
012226 000137 021170  
012232 012703 033436 B1:  
012236 004737 017456  
012242 013737 000576 000666 B2:  
012250 004737 012256  
012254 000207

```
*****  
:BACKSPACE SUBROUTINE:  
:THIS SUBROUTINE IS USED TO PERFORM THE  
:BACKSPACE OPERATION REQUIRED BY THE READ  
:ROUTINE FOR REAC 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.  
*****  
:TSTAL STAL  
:PC,STALL ;DO TURN AROUND STALL  
:MSG10,EMADDR  
:RDATA,2BA  
:TST TMEX ;SEE IF TM  
:BEQ BO ;IF NOT: BR  
:MOV #-1,2FC  
:MOV #32,MTC1  
:MOV #BKTM,RTRN  
:JMP TAPG ;SPACE TO TM  
:BIT #10000,2SWR ;SEE IF SHOULD CHECK ERROR  
:BNE BO ;IF NOT: BR  
:MOV #MSG55,EMADDR  
:BIT #4,2OS ;SEE IF TM  
:BNE BKTMO ;IF SO: BR  
:MOV #RDATA,CADER  
:JSR PC,ERPT ;PRINT ERROR  
:BR BO  
:MOV #RDATA,R3  
:JSR PC,ER2  
:MOV RCNT,RO  
:COM RO ;BUILD SPACE AMOUNT  
:INC RO  
:MOV #MSG10,EMADDR ;SET ERROR MESSG ADDRESS  
:MOV RO,2FC  
:MOV #32,MTC1 ;SET SPACE REVERSE  
:MOV #B1,RTRN ;SET RETURN ADDRESS  
:MOV RO,STAL ;SET INTERRUPT TIME MULTIPLIER  
:JMP TAPG ;GO DO SPACE  
:MOV #RDATA,R3  
:JSR PC,ER2  
:MOV TSTAL,STAL ;DO STALL  
:JSR PC,STALL ;STALL  
:RTS PC ;EXIT
```

3053  
3054  
3055  
3056  
3057  
3058  
3059  
3060  
3061  
3062  
3063  
3064

```
*****
: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.
*****
```

012256 005337 000666  
012262 00:375  
012264 00020

STALL: DEC STAL  
BNE STALL ;DELAY  
RTS PC ;EXIT

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  
3098  
3099  
0123  
0124  
0125

```

012266 012701 177760
012272 012702 174000
012276 004737 023246
012302 042737 000001 000626
012310 013737 000626 000556
012316 012737 177777 014360
012324 000207

012326 012702 000001
012332 012701 000500
012336 004737 023246
012342 013737 000626 000554
012350 000207
    
```

```

CNTF:
RCNTR:
    
```

```

*****
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.
*****

      MOV      #20,R1        ;SET HIGH LIMIT
      MOV      #-4000,R2    ;SET LOW LIMIT
      JSR      PC,RANG      ;GO GENERATE NUMBER
      BIC      #!,RANSAV
      MOV      RANSAV,FMCT   ;SET CHAR COUNT
      MOV      #-1,PATS     ;PRESET DATA PATTERN
      RTS      PC           ;EXIT

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

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

0103  
0104  
0105  
0106  
0107  
0108  
0109  
0110  
0111  
0112  
0113  
0114  
0115  
0116  
0117  
0118  
0119  
0120  
0121  
0122  
0123  
0124  
0125  
0126  
0127  
0128  
0129  
0130  
0131  
0132  
0133  
0134  
0135  
0136  
0137  
0138  
0139  
0140  
0141  
0142  
0143  
0144  
0145  
0146  
0147  
0148  
0149  
0150  
0151  
0152  
0153  
0154  
0155  
0156  
0157  
0158

```

*****
TEST CONDITION ENTRY ROUTINE:

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

```

```

012352 005737 000634
012356 001001
012360 000207
012362 005037 000674
012366 005037 005130
012372 012700 000010
012376 012701 00074E
012402 005021
012404 005300
012406 001375

```

```

TINP: TST TINF ;SEE IF SHOULD INPUT FROM TTY
      BNE TINPA ;IF SO: BR
      RTS PC ;EXIT
TINPA: CLR UNP ;CLEAR TABLE POINTER
      CLR REOTC ;CLEAR EOT UNIT COUNTER
      MOV #10,RO ;SET SIZE OF TABLE
      MOV #UN1,R1 ;SET START OF TABLE
TINPB: CLR (R1)+ ;CLEAR TABLE
      DEC RO ;SEE IF DONE
      BNE TINPB ;IF NOT: BR

```

3159	012410	012704	025405		MOV	#MSG31,R4	
3160	012414	005737	000734		TST	ASEQF	: SEE IF AUTO SEQ
3161	012420	001402			BEQ	TINPB1	: IF NOT: BR
3162	012422	012704	025333		MOV	#MSG30,R4	: SET AUTO SEQ HDR
3163	012426	104000		TINPB1:	TTOUTT		: PRINT PROGRAM NAME
3164	012430	005737	014122		TST	SCVFL	: SEE IF SHORT CONVERSATION
3165	012434	001067			BNE	TINPC	: IF SO: BR
3166	012436	012704	026436		MOV	#MSG74,R4	
3167	012442	104000			TTOUTT		: REQUEST STARTING REGISTER ADDRESS
3168	012444	013703	000544		MOV	REGS,R3	
3169	012450	104002			OCTPP		: PRINT CURRENT REG START
3170	012452	012705	000544		MOV	#REGS,R5	: SAVE ADDRESS LOCATION
3171	012456	012701	000006		MOV	#6,R1	: SET SIZE OF ENTRY
3172	012462	012702	176400		MOV	#176400,R2	: SET UPPER LIMIT
3173	012466	012703	172300		MOV	#172300,R3	: SET LOWER LIMIT
3174	012472	004737	023300		JSR	PC,TTR	: GO GET RESPONSE
3175	012476	012704	026461		MOV	#MSG75,R4	
3176	012502	104000			TTOUTT		: GO REQUEST VECTOR ADDRESS
3177	012504	013703	000546		MOV	VECT,R3	
3178	012510	104002			OCTPP		: PRINT CURRENT VECTOR
3179	012512	012705	000546		MOV	#VECT,R5	: SET SAVE LOCATION
3180	012516	012701	000003		MOV	#3,R1	: SET SIZE OF ENTRY
3181	012522	012702	000224		MOV	#224,R2	: SET UPPER LIMIT
3182	012526	012703	000150		MOV	#150,R3	: SET LOWER LIMIT
3183	012532	004737	023300		JSR	PC,TTR	: GO GET RESPONSE
3184	012536	013700	000546		MOV	VECT,R0	: GET VECTOR ADDRESS
3185	012542	012720	021666		MOV	#MTINT,(R0)+	: LOAD VECTOR WITH HANDLER ADDRESS
3186	012546	012710	000340		MOV	#340,(R0)	: LOAD PRIORITY LEVEL
3187	012552	013700	000544		MOV	REGS,R0	: GET STARTING REGISTER ADDRESS
3188	012556	012701	000016		MOV	#16,R1	: SET NUMBER OF REGISTERS
3189	012562	012702	000510		MOV	#C1,R2	: GET FIRST ADDRESS LOCATION
3190	012566	010022		TINPBO:	MOV	R0,(R2)+	: BUILD TABLE OF ADDRESSES
3191	012570	062700	000002		ADD	#2,R0	: BUMP ADDRESS
3192	012574	005301			DEC	R1	: SEE IF DONE
3193	012576	001373			BNE	TINPBO	: IF NOT: BR
3194	012600	005737	000734		TST	ASEQF	: SEE IF AUTO SEQ
3195	012604	001403			BEQ	TINPC	: IF NOT: BR
3196	012606	005726			TST	(SP)+	: RESET STACK POINTER
3197	012610	000137	021704		JMP	ASEQ	: GO TO AUTO SEQUENCE
3198	012614	012777	000040	165676	TINPC:	MOV	#10,@CS
3199	012622	012704	026102		MOV	#MSG52,R4	
3200	012626	104000			TTOLTT		: REQUEST DRIVE NUMBER
3201	012630	012705	000550		MOV	#DYN,R5	: GET ADDRESS
3202	012634	012701	000001		MOV	#1,R1	: SET SIZE OF RESPONSE
3203	012640	012702	000007		MOV	#7,R2	: SET UPPER LIMIT
3204	012644	012703	000000		MOV	#0,R3	: SET LOWER LIMIT
3205	012650	004737	023300		JSR	PC,TTR	: GO GET DRIVE NUMBER
3206	012654	013777	000550	165636	MOV	DYN,@CS	
3207	012662	005777	165622		TST	@C1	: ACCESS DRIVE
3208	012666	032777	010000	165624	BIT	#10000,@CS	: SEE IF NEG
3209	012674	001411			BEQ	TINPO	: IF NOT: BR
3210	012676	012704	026373		MOV	#MSG71,R4	
3211	012702	104000			TTOUTT		: PRINT NEG
3212	012704	013704	000510		MOV	C1,R4	
3213	012710	005204			INC	R4	
3214	012712	152714	000100		BISB	#100,R4	: CLEAR TPE

3215	012716	000736			BR	TINPC		;++G RETRY DVN
3216	012720	012704	025470		TINPC: MOV	#MSG32,R4		
3217	012724	104000			TTOUTT			:PRINT UNIT NUMBER REQUEST
3218	012726	005037	000644		CLR	TEMP2		:CLEAR BUFFER
3219	012732	012705	000644		MOV	#TEMP2,R5		:SET UNIT DESCRIPTION BUFFER ADDRESS
3220	012736	012701	000001		MOV	#1,R1		:SET NUMBER OF CHARACTERS TO INPUT
3221	012742	012702	000007		MOV	#7,R2		:SET MAXIMUM LIMIT
3222	012746	012703	000000		MOV	#0,R3		:SET MINIMUM LIMIT
3223	012752	004737	023300		JSR	PC,TTR		:GO GET UNIT NUMBER
3224	012756	005737	000642		TST	TEMP1		:SEE IF HAVE NEW PARAMETER
3225	012762	001013			BNE	TINPOB		:IF SO: BR
3226	012764	005737	000674		TST	UNP		:SEE IF FIRST ENTRY
3227	012770	001001			BNE	TINPOA		:IF NOT: BR
3228	012772	000752			BR	TINPO		:++G ELSE RETRY
3229	012774	013700	000674		TINPOA: MOV	UNP,R0		
3230	013000	012760	177777	000746	MOV	#-1,UNI(R0)		:SET END UNIT TABLE
3231	013006	000137	013376		JMP	TINP2C		:GO GET RECORD COUNT
3232	013012	013700	000674		TINPOB: MOV	UNP,R0		
3233	013016	042760	000007	000746	BIC	#7,UNI(R0)		:CLEAR UNIT NUMBER
3234	013024	004737	014136		JSR	PC,TPOS1		:GO LOAD UNIT NUMBER TO PROPER POSITION
3235	013030	012777	000040	165462	MOV	#40,ACS		
3236	013036	013777	000550	165454	MOV	DVN,ACS		
3237	013044	016077	000746	165470	MOV	UNI(R0),AC2		:LOAD UNIT NUMBER
3238	013052	032777	002000	165456	TINPOC: BIT	#2000,ADT		:SEE IF SLAVE PRESENT
3239	013060	001005			BNE	TINPOD		:IF SO: BR
3240	013062	012704	026161		MOV	#MSG57,R4		
3241	013066	104000			TTOUTT			:PRINT NON-EXIST SLAVE
3242	013070	000137	012720		JMP	TINPO		:REDO
3243	013074	022777	142011	165434	TINPOD: CMP	#142011,ADT		:++G SEE IF 9TRK TMO2,TU16/TE16
3244	013102	001406			BEQ	TINPOE		:IF SO: BR
3245	013104	012704	026055		MOV	#MSG50,R4		:ILLEGAL DRIVE TYPE
3246	013110	104000			TTOUTT			:GO PRINT
3247	013112	017703	165420		MOV	ADT,R3		
3248	013116	104002			OCTPP			:PRINT DRIVE TYPE REGISTER
3249	013120	012704	024735		TINPOE: MOV	#MSG9,R4		
3250	013124	104000			TTOUTT			:PRINT SERIAL NUMBER TAG
3251	013126	017703	165406		MOV	ASN,R3		
3252	013132	004737	024224		JSR	PC,SNPT		:PRINT SERIAL NUMBER
3253	013136	012704	025511		TINP1: MOV	#MSG33,R4		
3254	013142	104000			TTOUTT			:PRINT DENSITY REQUEST
3255	013144	005037	000644		CLR	TEMP2		:CLEAR BUFFER
3256	013150	012701	000001		MOV	#1,R1		:SET NUMBER OF CHARACTERS TO INPUT
3257	013154	012702	000007		MOV	#7,R2		:SET MAXIMUM LIMIT
3258	013160	012703	000000		MOV	#0,R3		:SET MINIMUM LIMIT
3259	013164	004737	023300		JSR	PC,TTR		:GO GET DENSITY
3260	013170	005737	000642		TST	TEMP1		:SEE IF HAVE NEW PARAMETER
3261	013174	001407			BEQ	TINP2		:IF NOT: BR
3262	013176	042737	003400	000552	BIC	#3400,UDES		:ELSE CLEAR OLD PARAMETER
3263	013204	012703	000010		MOV	#10,R3		:SET POSITION FACTOR
3264	013210	004737	014124		JSR	PC,TPOS		:GO LOAD DENSITY INTO PROPER POSITION
3265	013214	012704	025525		TINP2: MOV	#MSG34,R4		
3266	013220	104000			TTOLTT			:PRINT PARITY REQUEST
3267	013222	005037	000644		CLR	TEMP2		:CLR BUFFER
3268	013226	012701	000001		MOV	#1,R1		:SET NUMBER OF CHARACTERS TO INPUT
3269	013232	012702	000007		MOV	#7,R2		:SET MAXIMUM LIMIT
3270	013236	012703	000000		MOV	#0,R3		:SET MINIMUM LIMIT



3271	013242	004737	023300		JSR	PC,TTR	:GO INPUT PARITY
3272	013246	005737	000642		TST	TEMP1	:SEE IF HAVE NEW PARAMETER
3273	013252	001407			BEQ	TINP2A	:IF NOT: BR
3274	013254	042737	000010	000552	BIC	#10,UDES	:ELSE CLEAR OLD PARAMETER
3275	013262	012703	000003		MOV	#3,R3	:SET POSITION FACTOR
3276	013266	004737	014124		JSR	PC,TPOS	:GO LOAD PARITY TO PROPER POSITION
3277	013272	012704	026124		TINP2A: MOV	#MSG53,R4	
3278	013276	104000			TTOUTT		:REQUEST FORMAT
3279	013300	005037	000644		CLR	TEMP2	
3280	013304	012701	000002		MOV	#2,R1	
3281	013310	012702	000016		MOV	#16,R2	
3282	013314	012703	000014		MOV	#14,R3	
3283	013320	004737	023300		JSR	PC,TTR	:GO GET FORMAT
3284	013324	005737	000642		TST	TEMP1	:SEE IF NEW PARAMETER
3285	013330	001407			BEQ	TINP2B	:IF NOT: BR
3286	013332	042737	000170	000552	BIC	#170,UDES	
3287	013340	012703	000004		MOV	#4,R3	
3288	013344	004737	014124		JSR	PC,TPOS	
3289	013350	005237	005130		TINP2B: INC	REOTC	:BUMP EOT UNIT COUNTER
3290	013354	022737	000016	000674	CMP	#16,UNP	:SEE IF DONE UNITS
3291	013362	001405			BEQ	TINP2C	:IF 50: BR
3292	013364	062737	000002	000674	ADD	#2,UNP	:POINT TO NEXT JNIT
3293	013372	000137	012720		JMP	TINP0	:ELSE LOOK FOR NEXT JNIT
3294	013376	005037	000674		TINP2C: CLR	UNP	:CLEAR UNIT POINTER
3295	013402	013700	005130		MOV	REOTC,R0	
3296	013406	000337	005130		SWAB	REOTC	
3297	013412	110037	005130		MOV	R0,REOTC	:SET UNIT EOT COUNTER
3298	013416	012704	025540		TINP3: MOV	#MSG35,R4	
3299	013422	104000			TTOUTT		:PRINT RECORD COUNT REQUEST
3300	013424	013703	000554		MOV	RCNT,R3	
3301	013430	104002			OCTPP		:PRINT RECORD COUNT
3302	013432	012705	000554		MOV	#RCNT,R5	:SET RECORD COUNT ADDRESS
3303	013436	012701	000006		MOV	#6,R1	:SET NUMBER OF CHARACTERS TO INPL*
3304	013442	012702	177777		MOV	#-1,R2	:SET MAXIMUM LIMIT
3305	013446	012703	000001		MOV	#1,R3	:SET MINIMUM LIMIT
3306	013452	004737	023300		JSR	PC,TTR	:GO GET RECORD COUNT
3307	013456	013737	000554	000630	MOV	RCNT,RCSAV	:SAVE RECORD COUNT
3308	013464	012704	025561		MOV	#MSG36,R4	
3309	013470	104000			TTOUTT		:PRINT CHARACTER COUNT REQUEST
3310	013472	005437	000556		NEG	FCMNT	
3311	013476	013703	000556		MOV	FCMNT,R3	
3312	013502	104002			OCTPP		:PRINT CHAR COUNT
3313	013504	012705	000556		MOV	#FCMNT,R5	:SET CHARACTER COUNT ADDRESS
3314	013510	012701	000006		MOV	#6,R1	:SET NUMBER OF CHARACTERS TO INPL*
3315	013514	012702	004000		MOV	#4000,R2	:SET MAXIMUM LIMIT
3316	013520	012703	000004		MOV	#4,R3	:SET MINIMUM LIMIT
3317	013524	004737	023300		JSR	PC,TTR	:GO GET CHARACTER COUNT
3318	013530	005437	000556		NEG	FCMNT	:SET TO TWO'S COMPLIMENT
3319	013534	013737	000556	000632	MOV	FCMNT,FCSAV	:SAVE FRAME COUNT
3320	013542	012704	025605		MOV	#MSG37,R4	:PRINT PATTERN NUMBER REQUEST
3321	013546	104000			TTOUTT		
3322	013550	013703	000560		MOV	PATRN,R3	
3323	013554	104002			OCTPP		:PRINT PATTERN
3324	013556	005037	014526		CLR	DOFL	:CLEAR EXTERNAL DATA FLAG
3325	013562	012705	000560		MOV	#PATRN,R5	:SET PATTERN NUMBER ADDRESS
3326	013566	012701	000002		MOV	#2,R1	:SET NUMBER OF CHARACTERS TO INPL*

3327	013572	012702	000015	MOV	#15,R2	;SET MAXIMUM LIMIT
3328	013576	012703	000000	MOV	#0,R3	;SET MINIMUM LIMIT
3329	013602	004737	023300	JSR	PC,TTR	;GO GET PATTERN NUMBER
3330	013606	012704	026321	MOV	#MSG69,R4	
3331	013612	104000		TTOUTT		;REQUEST TM
3332	013614	013703	000564	MOV	TMEX,R3	
3333	013620	104002		OCTPP		;PRINT CURRENT TM FLAG SETTING
3334	013622	012705	000564	MOV	#TMEX,R5	;GET TM FLAG ADDRESS
3335	013626	012701	000001	MOV	#1,R1	;SET SIZE OF RESPONSE
3336	013632	012702	000001	MOV	#1,R2	;SET UPPER LIMIT
3337	013636	012703	000000	MOV	#0,R3	;SET LOWER LIMIT
3338	013642	004737	023300	JSR	PC,TTR	;TM 1=YES
3339	013646	012704	025065	MOV	#MSG21,R4	
3340	013652	104000		TTOUTT		;REQUEST INTERCHANGE REAC
3341	013654	013703	000566	MOV	INTRF,R3	
3342	013660	104002		OCTPP		;PRINT CURRENT SETTING
3343	013662	012705	000566	MOV	#INTRF,R5	;GET FLAG ADDRESS
3344	013666	012701	000001	MOV	#1,R1	;SET SIZE OF RESPONSE
3345	013672	012702	000001	MOV	#1,R2	;SET UPPER LIMIT
3346	013676	012703	000000	MOV	#0,R3	;SET LOWER LIMIT
3347	013702	004737	023300	JSR	PC,TTR	;GO GET RESPONSE
3348	013706	012704	025630	MOV	#MSG38,R4	
3349	013712	104000		TTOUTT		;REQUEST SINGLE PASS
3350	013714	013703	000570	MOV	SPFLG,R3	
3351	013720	104002		OCTPP		;PRINT CURRENT SETTING
3352	013722	012705	000570	MOV	#SPFLG,R5	;SET ADDRESS OF FLAG
3353	013726	012701	000001	MOV	#1,R1	;SET SIZE OF RESPONSE
3354	013732	012702	000001	MOV	#1,R2	;SET UPPER LIMIT
3355	013736	012703	000000	MOV	#0,R3	;SET LOWER LIMIT
3356	013742	004737	023300	JSR	PC,TTR	;GO GET RESPONSE
3357	013746	005737	014122	TST	SCVFL	;SEE IF SHORT CONVERSATION
3358	013752	001060		BNE	TINPX	;IF SO: BR
3359	013754	012704	025650	MOV	#MSG40,R4	
3360	013760	104000		TTOUTT		;PRINT READ STALL REQUEST
3361	013762	013703	000572	MOV	RSTAL,R3	
3362	013766	104002		OCTPP		;PRINT READ STALL
3363	013770	012705	000572	MOV	#RSTAL,R5	;SET READ STALL ADDRESS
3364	013774	012701	000006	MOV	#6,R1	;SET NUMBER OF CHARACTERS TO INPUT
3365	014000	012702	177777	MOV	#-1,R2	;SET MAXIMUM LIMIT
3366	014004	012703	000001	MOV	#1,R3	;SET MINIMUM LIMIT
3367	014010	004737	023300	JSR	PC,TTR	;GO GET READ STALL
3368	014014	012704	025676	MOV	#MSG41,R4	
3369	014020	104000		TTOUTT		;PRINT WRITE STALL REQUEST
3370	014022	013703	000574	MOV	WSTAL,R3	
3371	014026	104002		OCTPP		;PRINT READ STALL
3372	014030	012705	000574	MOV	#WSTAL,R5	;SET WRITE STALL ADDRESS
3373	014034	012701	000006	MOV	#6,R1	;SET NUMBER OF CHARACTERS TO INPUT
3374	014040	012702	177777	MOV	#-1,R2	;SET MAXIMUM LIMIT
3375	014044	012703	000001	MOV	#1,R3	;SET MINIMUM LIMIT
3376	014050	004737	023300	JSR	PC,TTR	;GO GET WRITE STALL
3377	014054	012704	025710	MOV	#MSG42,R4	
3378	014060	104000		TTOUTT		;PRINT TURN AROUND STALL REQUEST
3379	014062	013703	000576	MOV	TSTAL,R3	
3380	014066	104002		OCTPP		;PRINT TA STALL
3381	014070	012705	000576	MOV	#TSTAL,R5	;SET TURN AROUND STALL ADDRESS
3382	014074	012701	000006	MOV	#6,R1	;SET NUMBER OF CHARACTERS TO INPUT

\*INP4:

J06

TM02-TU16/TE16 DATA RELIABILITY PROGRAM MACY11 30(1046) 22-JUN-77 08:44 PAGE 74  
 DZUAG.P11 17-JUN-77 09:52

SEG 0074

```

3383 014100 012702 177777      MOV      #-1,R2      ;SET MAXIMUM LIMIT
3384 014104 012703 000001      MOV      #1,R3      ;SET MINIMUM LIMIT
3385 014110 004737 023300      JSR      PC,TTR     ;GO GET TURN AROUND STALL
3386 014114 005037 014122      TINPX:  CLR      SCVFL ;CLEAR SHORT CONVERSATION FLAG
3387 014120 000207              RTS      PC         ;EXIT
3388 014122 000000      SCVFL:  0          ;SHORT CONVERSATION FLAG
3389
3390                          ;UNIT DESCRIPTION POSITIONING SUBROUTINE*****
3391
3392 014124 000241      TPOS:  CLC              ;POSITION CHARACTER
3393 014126 006137 000644      ROL      TEMP2      ;SEE IF DONE
3394 014132 005303              DEC      R3         ;IF NOT: BR
3395 014134 001373              BNE     TPOS        ;LOAD UNIT POINTER
3396 014136 013700 000674      TPOS1: MOV      UNP,RO  ;LOAD CHARACTER INTO UN1(RO)
3397 014142 053760 000644 000746  BIS      TEMP2,UN1(RO)
3398 014150 000207              RTS      PC         ;EXIT
3399
  
```

0000  
0001  
0002  
0003  
0004  
0005  
0006  
0007  
0008  
0009  
0010  
0011  
0012  
0013  
0014  
0015  
0016  
0017  
0018  
0019  
0020  
0021  
0022  
0023  
0024  
0025  
0026  
0027  
0028  
0029  
0030  
0031  
0032  
0033  
0034  
0035  
0036  
0037  
0038  
0039  
0040  
0041  
0042  
0043  
0044  
0045  
0046  
0047  
0048  
0049  
0050  
0051  
0052  
0053  
0054  
0055  
0056

```

014152 005737 015122
014156 001044
014160 005737 000734
014164 001406
014166 005737 000560
014172 100003
014174 004737 015054
014200 000207
014202 023737 000560 014360
014210 001014
014212 013703 000552
014216 042703 177767
014222 023703 014362
014226 001404
014230 010337 014362
014234 004737 015124
014240 000207
014242 012703 027430
014246 013701 000560
014252 010137 014360
014256 062701 000001
014262 006301
014264 004771 002770
014270 032777 010000 164240
014276 001410
014300 012702 002002
014304 012701 027430
014310 042721 140300
014314 005302
014316 001374
014320 004737 015124
014324 012702 002000
014330 012701 033436
014334 005021
014336 005302
014340 001375
014342 013737 000552 014362
014350 042737 177767 014362
  
```

```

*****
: 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 C
: 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.
*****

DSUP: TST      R0FL      ;SEE IF DID RANDOM DATA
      BNE     DS1      ;++G F SO BRANCH
DSO:   TST     ASEOF    ;SEE IF AUTO SEQ
      BEQ     DSOC     ;IF NOT: BR
      TST     PATRN    ;SEE IF AUTO RANDOM
      BPL     DSOC     ;IF NOT: BR
      JSR     PC, DATR ;ELSE GO GENERATE RANDOM DATA
      RTS     PC       ;RETURN
DSOC:  CMP     PATRN,PATS ;SEE IF NEW PATTERN
      BNE     DSOR     ;IF SO: BR
      MOV     UDES,R3  ;GET UNIT DESCRIPTION
      BIC     #177767,R3 ;MASK EVEN PARITY
      CMP     PARS,R3  ;SEE IF SAME AS LAST TIME
      BEQ     DSOB     ;IF SO: BR
      MOV     R3,PARS  ;SAVE PARITY
      JSR     PC,CRC/LRC ;GO GENERATE EXPT CRC/LRC
DSOB:  RTS     PC
DSOR:  MOV     #WDATA,R3 ;R3 = ADDR OF WRITE BUFFER
      MOV     PATRN,R1 ;R1 = PATTERN SELECTOR
      MOV     R1,PATS
      ADD     #1,R1
      ASL     R1
      JSR     PC, DATBL(R1) ;BUMP POINTER
      BIT     #10000, RDT ;++G MAKE PATTERN SELECTOR EVEN
      BEQ     DS2A     ;GO GENERATE PATTERN
      MOV     #2002,R2  ;SEE IF 7 CH
      MOV     #WDATA,R1 ;IF NOT: BR
      BIC     #140300,(R1)+ ;SET BUFFER SIZE
      DEC     R2        ;SET START OF BUFFER
      BNE     DS2      ;MASK FOR 7 CH
      JSR     PC,CRC/LRC ;SEE IF DONE
      MOV     #2000,R2  ;IF NOT: BR
      MOV     #RDATA,R1 ;GO GENERATE EXPT CRC/LRC
      CLR     (R1)+    ;R2=BUFFER SIZE
      DEC     R2        ;R1=READ DATA START
      BNE     DS4      ;CLEAR BUFFER
      JSR     PC,CRC/LRC ;SEE IF DONE ALL
      MOV     #2000,R2  ;IF NOT: BR
      MOV     #RDATA,R1 ;GET UNIT DESCRIPTION
      CLR     (R1)+    ;MASK PARITY
      DEC     R2
      BNE     DS4
      MOV     UDES,PARS
      BIC     #177767,PARS
  
```

L06

```

3456 014356 000207          RTS      PC          ;EXIT
3457 014360 177777          PATS:   -1          ;PATTERN NUMBER SAVE
3458 014362 000000          PARS:   0
3459
3460          ;EXTERNAL DATA INPUT FROM H/S READER (256 CHARACTER MAXIMUM)
3461
3462 014364 005737 014526          DATO:   TST      DOFL          ;++G BRANCH IF EXTERNAL INPUT
3463 014370 001401          BEQ      IS          ;++G
3464 014372 000207          RTS      PC          ;++G RETURN
3465 014374 012737 000001 014526  IS:   MOV      #1,DOFL        ;SET EXTERNAL FLAG
3466 014402 005077 164214          CLR      @PRB        ;CLEAR READER BUFFER
3467 014406 005077 164206          CLR      @PRS        ;CLEAR READER STATUS
3468 014412 005037 000642          CLR      TEMP1       ;CLEAR FOR USE AS CHARACTER FLAG
3469 014416 052777 000001 164174  DATOA:  BIS      #1,@PRS        ;START READER
3470 014424 105777 164170          DATOB:  TSTB     @PRS        ;++G SEE IF DONE
3471 014430 100375          BPL      DATOB        ;++B
3472 014432 005001          CLR      R1          ;CLEAR SAVE LOCATION
3473 014434 117701 164162          MOVB    @PRB,R1       ;SAVE CHARACTER
3474 014440 005737 000642          TST     TEMP1        ;SEE IF HAVE FOUND START CHARACTER
3475 014444 001011          BNE     DATOC        ;IF SO : BR
3476 014446 105701          TSTB    R1          ;SEE IF CHARACTER IS 0
3477 014450 001762          BEQ     DATOA        ;IF SO : BR
3478 014452 012737 000001 000642  MOV     #1,TEMP1      ;ELSE SET CHARACTER FOUND FLAG
3479 014460 010137 000644          MOV     R1,TEMP2     ;SAVE DATA SIZE
3480 014464 010102          MOV     R1,R2        ;SAVE DATA SIZE
3481 014466 000753          BR      DATOA        ;++G GO GET FIRST DATA CHAR
3482 014470 110123          DATOC:  MOVB    R1,(R3)+ ;LOAD BUFFER
3483 014472 005302          DEC     R2          ;SEE IF READ ALL
3484 014474 001350          BNE     DATOA        ;IF NOT : BR
3485 014476 012701 027430          DATOD:  MOV     @WDATA,R1 ;R1 = START OF WRITE BUFFER
3486 014502 013702 000644          MOV     TEMP2,R2     ;R2 = SIZE OF DATA FIELD
3487 014506 112123          DATOE:  MOVB    (R1)+,(R3)+ ;REPEAT LOAD OF DATA FIELD
3488 014510 022703 033436          CMP     @RDATA,R3    ;SEE IF DONE
3489 014514 003001          BGT     DATOF        ;IF NOT: BR
3490 014516 000207          RTS     PC          ;++G RETURN
3491 014520 005302          DATOF:  DEC     R2          ;SEE IF AT END OF DATA FIELD
3492 014522 001371          BNE     DATOE        ;IF NOT : BR
3493 014524 000764          BR      DATOD        ;++G ELSE RESTART FILL
3494 014526 000000          COFL:   0          ;EXTERNAL DATA FLAG=1 IF ALREADY DONE
3495
    
```

```

3496                                     ;ALL ONES*****
3497
3498 014530 012701 177777  DAT1:  MOV    #-1,R1          ;R1=DATA
3499 014534 012702 002002  DAT1A: MOV    #2002,R2        ;R2=WORD COUNT +2
3500 014540 010123          DAT1B: MOV    R1,(R3)+      ;LOAD BUFFER
3501 014542 005302          DEC    R2                ;SEE IF DONE
3502 014544 001375          BNE    DAT1B            ;IF NOT: BR
3503 014546 000207          RTS    PC                ;++G RETURN
3504
3505                                     ;ALL ZEROS*****
3506
3507 014550 005001  DAT2:  CLR    R1          ;R1=DATA
3508 014552 000770  BR     DAT1A        ;++G LOAD BUFFER
3509
3510                                     ;WALKING ONE*****
3511
3512 014554 012701 000001  DAT3:  MOV    #1,R1          ;R1=DATA
3513 014560 000241          CLC
3514 014562 012702 004004  DAT3A: MOV    #4004,R2        ;R2=CHARACTER COUNT+4
3515 014566 110123          DAT3B: MOV    R1,(R3)+      ;LOAD BUFFER
3516 014570 106101          ROLB   R1                ;SET NEXT CHARACTER
3517 014572 005302          DEC    R2                ;SEE IF DONE
3518 014574 001374          BNE    DAT3B            ;IF NOT: BR
3519 014576 000207          RTS    PC                ;++G RETURN
3520
3521                                     ;WALKING ZERO*****
3522
3523 014600 012701 000376  DAT4:  MOV    #376,R1        ;R1=START OF DATA
3524 014604 000261          SEC
3525 014606 000765          BR     DAT3A            ;++G LOAD BUFFER
3526
3527                                     ;ALTERNATING ONE/ZERO*****
3528
3529
3530 014610 012701 052525  DAT5:  MOV    #52525,R1      ;R1=DATA
3531 014614 000747          BR     DAT1A            ;++G LOAD BUFFER
3532
3533                                     ;ALTERNATING ZERO/ONE*****
3534
3535 014616 012701 125252  DAT6:  MOV    #125252,R1    ;R1=DATA
3536 014622 000744          BR     DAT1A            ;++G LOAD BUFFER
3537
3538                                     ;ONE/ZERO IN ALTERNATING WORDS*****
3539
3540 014624 012701 125252  DAT7:  MOV    #125252,R1    ;SET WORD 1
3541 014630 012702 052525  MOV    #52525,R2        ;SET WORD 2
3542 014634 012704 001002  MOV    #1002,R4         ;SET NUMBER OF ENTRIES
3543 014640 010123          DAT7A: MOV    R1,(R3)+      ;LOAD WORD 1
3544 014642 010223          MOV    R2,(R3)+      ;LOAD WORD 2
3545 014644 005304          DEC    R4                ;SEE IF DONE
3546 014646 001374          BNE    DAT7A            ;IF NOT: BR
3547 014650 000207          RTS    PC                ;++G RETURN
3548

```

```

3549                                     ;WALKING ONE/ALL ONE IN ALTERNATING CHARS****
3550
3551 014652 012702 002002      DAT10: MOV      #2002,R2      ;SET BUFFER SIZE
3552 014656 012701 000001      MOV      #1,R1        ;SET WALK BASE
3553 014662 000241
3554 014664 012713 177400      DAT10A: MOV     #177400,(R3) ;LOAD ALL ONE BYTE
3555 014670 050123              BIS      R1,(R3)+    ;LOAD WALK BYTE
3556 014672 106101              ROLB    R1           ;WALK ONE
3557 014674 005302              DEC     R2
3558 014676 001372              BNE     DAT10A      ;DO FULL BUFFER
3559 014700 000207              RTS     PC          ;++G RETURN
3560
3561                                     ;ALL BITS 0-377*****
3562
3563 014702 005001      DAT11: CLR     R1        ;R1=STARTING DATA
3564 014704 012702 004004      MOV     #4004,R2     ;R2=CHARACTER COUNT+4
3565 014710 110123      DAT11A: MOVB   R1,(R3)+ ;LOAD BUFFER
3566 014712 105201      INCB   R1           ;BUMP DATA
3567 014714 005302      DEC    R2          ;SEE IF DONE
3568 014716 001374      BNE    DAT11A     ;IF NOT: BR
3569 014720 000207      RTS    PC          ;++G RETURN
3570
3571                                     ;ALL BITS 377-0*****
3572
3573 014722 012701 000377      DAT12: MOV     #377,R1 ;R1=STARTING DATA
3574 014726 012702 004004      MOV     #4004,R2     ;R2=CHARACTER COUNT+4
3575 014732 110123      DAT12A: MOVB   R1,(R3)+ ;LOAD BUFFER
3576 014734 105301      DECB   R1           ;BUMP DATA
3577 014736 005302      DEC    R2          ;SEE IF DONE
3578 014740 001374      BNE    DAT12A     ;IF NOT: BR
3579 014742 000207      RTS    PC          ;++G RETURN
3580
3581                                     ;ALTERNATING CHARACTERS 0 AND 377*****
3582
3583 014744 012701 000377      DAT13: MOV     #377,R1 ;R1 = DATA
3584 014750 000137 014534      JMP     DAT1A      ;LOAD BUFFER
3585
3586                                     ;WALKING ZERO/ALL ZERO IN ALTERNATING CHARS*****
3587
3588 014754 012702 002002      DAT14: MOV     #2002,R2 ;SET BUFFER SIZE
3589 014760 012701 000376      MOV     #376,R1     ;SET WALK BASE
3590 014764 000261      SEC
3591 014766 010113      DAT14A: MOV     R1,(R3) ;LOAD WALK BYTE
3592 014770 042723 177400      BIC    #177400,(R3)+ ;CLEAR HIGH BYTE
3593 014774 106101      ROLB   R1           ;WALK ZERO BIT
3594 014776 005302      DEC    R2
3595 015000 001372      BNE    DAT14A     ;FILL BUFFER
3596 015002 000207      RTS    PC          ;++G RETURN
3597
    
```

```

;AUTO SEQUENCE PATTERN*****
3598
3599
3600 015004 012702 000200 DAT15: MOV #200,R2 ;SET NUMBER OF ENTRIES
3601 015010 012701 015034 DAT15A: MOV #APATS,R1 ;SET START OF PATTERN
3602 015014 012704 000010 MOV #10,R4 ;SET SIZE OF PATTERN
3603 015020 012123 DAT15B: MOV (R1)+,(R3)+ ;FILL BUFFER
3604 015022 005304 DEC R4 ;SEE IF DONE PATTERN
3605 015024 001375 BNE DAT15B ;IF NOT: BR
3606 015026 005302 DEC R2 ;SEE IF DONE BUFER
3607 015030 001367 BNE DAT15A ;IF NOT: BR
3608 015032 000207 RTS PC ;++G RETURN
3609 015034 000000 APATS: 0
3610 015036 177400 177400
3611 015040 000377 377
3612 015042 000000 0
3613 015044 177777 -1
3614 015046 000377 377
3615 015050 177400 177400
3616 015052 177777 -1
3617
3618 ;RANDOM DATA GENERATOR SUBROUTINE*****
3619
3620 015054 013704 000556 DATR: MOV FMCNT,R4 ;SET NUMBER OF FRAMES
3621 015060 012703 027430 MOV #WDATA,R3 ;SET ADDRESS OF START OF BUFFER
3622 015064 012701 177777 MOV #-1,R1 ;SET HIGH LIMIT
3623 015070 005002 CLR R2 ;SET LOW LIMIT
3624 015072 004737 023246 DATRO: JSR PC,RANG ;GO GENERATE NUMBER
3625 015076 013723 000626 MOV RANSV,(R3)+ ;LOAD BUFFER
3626 015102 005204 INC R4 ;SEE IF DONE WHOLE BUFFER
3627 015104 001372 BNE DATRO ;IF NOT: BR
3628 015106 004737 014270 JSR PC,DS1 ;GO CHECK FOR 7 CH
3629 015112 012737 000001 015122 MOV #1,RDFL ;SET RANDOM DATA FLAG
3630 015120 000207 RTS PC ;EXIT
3631 015122 000000 RDFL: 0 ;RANDOM DATA SELECT FLAG
    
```



```

3632
3633
3634
3635
3636
3637
3638
3639
3640
3641 015124 013700 000556 CRCLRC: MOV FMCNT,R0 ;SET RECORD SIZE
3642 015130 005400 NEG RO
3643 015132 012701 027430 MOV #WDATA,R1 ;SET START OF BUFFER
3644 015136 005037 015506 CLR XORS
3645 015142 111104 CLD: MOV#B (R1),R4 ;GET CHARACTER
3646 015144 004737 015334 JSR PC,CLP ;GO GET PARITY OF CHARACTER
3647 015150 004737 015462 JSR PC,XOR ;XOR CHARACTER
3648 015154 000241 CLC
3649 015156 006004 ROR R4 ;ROTATE 1 RIGHT
3650 015160 103014 BCC CL2 ;IF NO CARRY: BR
3651 015162 052704 000400 BIS #400,R4 ;SET BIT NINE
3652 015166 000241 CLC
3653 015170 010405 CL1: MOV R4,R5 ;SAVE CHARACTER
3654 015172 042705 177703 BIC #177703,R5
3655 015176 005105 COM R5
3656 015200 042705 177703 BIC #177703,R5
3657 015204 042704 000074 BIC #74,R4
3658 015210 050504 BIS R5,R4 ;COMPLIMENT BITS 2,3,4,5
3659 015212 010437 015506 CL2: MOV R4,XORS
3660 015216 005300 DEC RO
3661 015220 001401 BEQ CLLAST ;IF LAST CHARACTER: BR
3662 015222 000747 BR CLLAST ;++G GET NEXT
3663 015224 013704 015506 CLLAST: MOV XORS,R4
3664 015230 005137 015506 COM XORS
3665 015234 042737 177050 015506 BIC #177050,XORS
3666 015242 042704 177727 BIC #177727,R4 ;COMPLIMENT ALL BUT BITS 3&5
3667 015246 050437 015506 BIS R4,XORS
3668 015252 013737 015506 015510 MOV XORS,EXCRC ;SAVE EXPECTED CRC
3669 015260 013700 000556 MOV FMCNT,R0
3670 015264 005400 NEG RO
3671 015266 012701 027430 MOV #WDATA,R1 ;DO EXPT LRC
3672 015272 005037 015506 CLR XORS
3673 015276 111104 CL3: MOV#B (R1),R4
3674 015300 004737 015334 JSR PC,CLP ;GET PARITY
3675 015304 004737 015462 JSR PC,XOR ;XOR CHARACTER
3676 015310 005300 DEC RO
3677 015312 001371 BNE CL3 ;DO ALL FOR LRC
3678 015314 013704 015510 MOV EXCRC,R4
3679 015320 004737 015462 JSR PC,XOR ;XOR CRC TO DATA
3680 015324 013737 015506 015512 MOV XORS,EXLRC ;SAVE EXPT LRC
3681 015332 000207 RTS PC ;RETURN
3682 015334 005704 CLP: TST R4 ;SEE IF 0 CHAR
3683 015336 001010 BNE CLPE ;IF NOT: BR
3684 015340 032737 000010 000552 BIT #10,UDES ;SEE IF EVEN PARITY
3685 015346 001404 BEQ CLPE ;IF NOT: BR
3686 015350 012704 000420 MOV #420,R4 ;SET 0 CHAR EVEN PARITY
3687 015354 00520: INC R1 ;BUMP PCINTER

```

3688	015356	000207			RTS	PC		; RETURN
3689	015360	005037	015514		CLPE: CLR	PARCNT		; CLEAR BIT COUNTER
3690	015364	012703	000010		MOV	#10,R3		; SET NUMBER OF BITS
3691	015370	032704	000001		CLPO: BIT	#1,R4		; SEE IF ONE BIT
3692	015374	001402			BEO	CLP1		; IF NOT: BR
3693	015376	005237	015514		INC	PARCNT		; BUMP COUNTER
3694	015402	000241			CLP1: CLC			
3695	015404	006004			ROR	R4		; ROTATE TO NEXT BIT
3696	015406	005303			DEC	R3		
3697	015410	001367			BNE	CLPO		; CONTINUE FOR ALL BITS
3698	015412	112104			MOV8	(R1)+,R4		
3699	015414	042704	177400		BIC	#177400,R4		
3700	015420	032737	000001	015514	BIT	#1,PARCNT		; SEE IF ODD NUMBER OF ONE BITS
3701	015426	001005			BNE	CLP2		; IF SO: BR
3702	015430	032737	000010	000552	BIT	#10,UDES		; SEE IF SHOULD BE EVEN PARITY
3703	015436	001406			BEO	CLP3		; IF NOT: BR
3704	015440	000207			RTS	PC		; ELSE EXIT
3705	015442	032737	000010	000552	CLP2: BIT	#10,UDES		; SEE IF SHOULD BE ODD PARITY
3706	015450	001001			BNE	CLP3		; IF NOT: BR
3707	015452	000207			RTS	PC		; ELSE EXIT
3708	015454	052704	000400		CLP3: BIS	#400,R4		; SET PARITY BIT
3709	015460	000207			RTS	PC		
3710	015462	010446			XOR: MOV	R4, -(SP)		
3711	015464	043716	015506		BIC	XORS, (SP)		
3712	015470	040437	015506		BIC	R4, XORS		; XOR SUBROUTINE: R4 WITH XORS
3713	015474	052637	015506		BIS	(SP)+, XORS		
3714	015500	013704	015506		MOV	XORS, R4		
3715	015504	000207			RTS	PC		
3716								
3717	015506	000000			XORS: 0			; XOR SAVE
3718	015510	000000			EXCRC: 0			; EXPECTED CRC
3719	015512	000000			EXLRC: 0			; EXPECTED LRC
3720	015514	000000			PARCNT: 0			; PARITY COUNTER
3721								

```

3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737 015516 005037 000656          DCHK: CLR      BBC          ; CLEAR BAD RECORD CNTR
3738 015522 005037 000704          CLR      DERFL        ; CLEAR DATA ERROR FLAG
3739 015526 013705 000556          MOV      FMCNT,R5     ; LOAD CHAR COUNT
3740 015532 032737 000020 000552  BIT      #20,UDES     ; SEE IF CORE DUMP
3741 015540 001402          BEQ      DCHK0        ; IF NOT: BR
3742 015542 000261          SEC
3743 015544 006005          ROR
3744 015546 012701 027430          DCHK0: MOV      #WDATA,R1   ; R5 = FC/2
3745 015552 012702 033436          MOV      #RDATA,R2   ; SET WRITE DATA ADDR
3746 015556 032737 000010 000552  BIT      #10,UDES     ; SET READ DATA ADDR
3747 015564 001430          BEQ      DFOC0        ; SEE IF EVEN PARITY
3748 015566 032737 000020 000552  BIT      #20,UDES     ; IF NOT: BR
3749 015574 001024          BNE      DFOC0        ; SEE IF CORE DUMP PARITY
3750 015576 032737 002000 000552  BIT      #2000,UDES   ; IF SO: BR
3751 015604 001020          BNE      DFOC0        ; SEE IF PE MODE
3752 015606 105711          DFOF: TSTB      (R1)   ; IF SO: BR
3753 015610 001404          BEQ      DFOD         ; SEE IF 0 CHAR
3754 015612 005201          INC      R1           ; IF SO: BR
3755 015614 005205          DFOE: INC      R5           ; BUMP POINTER
3756 015616 001373          BNE      DFOF         ; SEE IF DONE
3757 015620 000406          BR       DFOC         ; IF NOT: BR
3758 015622 112721 000020          DFOD: MOVB     #20,(R1)+ ; ELSE CONTINUE
3759 015626 012737 177777 014360  MOV      #-1,PATS    ; SET 20 IN PLACE OF 0
3760 015634 000767          BR       DFOE         ; SET PATTERN GENERATE FLAG
3761 015636 013705 000556          DFOC: MOV      FMCNT,R5 ; RESET CHAR CNT
3762 015642 012701 027430          MOV      #WDATA,R1   ; RESET DATA ADDRESS
3763 015646 032737 010000 000562  DFOC0: BIT      #10000,RDCMD ; SEE IF READ REVERSE
3764 015654 001462          BEQ      DFO         ; IF NOT: BR
3765 015656 013704 000556          DFOB: MOV      FMCNT,R4 ; GET FRAME COUNT
3766 015662 005404          NEG      R4           ; SET TO WHOLE NUMBER
3767 015664 032737 000020 000552  BIT      #20,UDES     ; SEE IF CORE DUMP
3768 015672 001402          BEQ      DFOB0        ; IF NOT: BR
3769 015674 000241          CLC
3770 015676 006004          ROR      R4           ; SET TO FC/2
3771 015700 060401          DFOB0: ADD      R4,R1      ; POINT TO START OF WRITE DATA
3772 015702 060402          ADD      R4,R2      ; POINT TO START OF READ DATA
3773 015704 032737 000001 000556  BIT      #1,FMCNT     ; SEE IF ODD FRAME COUNT
3774 015712 001401          BEQ      DFOA        ; IF NOT: BR
3775 015714 105722          TSTB    (R2)+        ; BUMP POINTER
3776 015716 032737 000020 000552  DFOA: BIT      #20,UDES     ; SEE IF CORE DUMP
3777 015724 001431          BEQ      DFOA4        ; IF NOT: BR

```

# F07

```

3778 015726 000241          CLC
3779 015730 132742 000001  BITB    #1, -(R2)      ;SEE IF BIT 0 = 1
3780 015734 001401          BEQ     DFOA0          ;IF NOT: BR
3781 015736 000261          SEC
3782 015740 106012          DFOA0: RORB    (R2)
3783 015742 000241          CLC
3784 015744 132712 000001  BITB    #1, (R2)
3785 015750 001401          BEQ     DFOA1
3786 015752 000261          SEC
3787 015754 106012          DFOA1: RORB    (R2)      ;POSITION BITS FOR REVERSE CORE DUMP
3788 015756 000241          CLC
3789 015760 132712 000001  BITB    #1, (R2)
3790 015764 001401          BEQ     DFOA2
3791 015766 000261          SEC
3792 015770 106012          DFOA2: RORB    (R2)
3793 015772 000241          CLC
3794 015774 132712 000001  BITB    #1, (R2)
3795 016000 001401          BEQ     DFOA3
3796 016002 000261          SEC
3797 016004 106012          DFOA3: RORB    (R2)
3798 016006 005202          INC     R2            ;RESET POINTER
3799 016010 124142          DFOA4: CMPB    -(R1), -(R2) ;TEST DATA CHARACTER
3800 016012 001010          BNE     DF1            ;IF NOT GOOD: BR
3801 016014 105037 000656  CLRB    BBC            ;CLEAR BAD RECORD COUNTER
3802 016020 000411          BR      DF2
3803 016022 122122          DFO:   CMPB    (R1)+, (R2)+ ;CHECK DATA
3804 016024 001003          BNE     DF1            ;IF BAD: BR
3805 016026 105037 000656  CLRB    BBC            ;CLEAR BAD RECORD CNTR
3806 016032 000404          BR      DF2
3807 016034 004737 016640  JSR     PC, DRPKF      ;GO GET DROPS AND PICKS
3808 016040 004737 016132  JSR     PC, DERR      ;GO DO PRINT
3809 016044 005205          DFO2: INC     R5            ;BUMP CHAR CNTR
3810 016046 001405          BEQ     DF3            ;IF DONE ALL: BR
3811 016050 032737 010000 000562 BIT     #10000, R0CMD ;SEE IF READ REVERSE
3812 016056 001761          BEQ     DFO            ;IF NOT: BR
3813 016060 000716          BR      DFOA          ;ELSE CONTINUE READ REV
3814 016062 005037 000664  DFO3: CLR     HDRFL      ;CLEAR HEADER FLAG
3815 016066 005737 000704  TST     DERFL          ;SEE IF HAD DATA ERROR
3816 016072 001416          BEQ     DFX            ;IF NOT: BR
3817 016074 005737 000706  TST     SERFL
3818 016100 001013          BNE     DFX            ;IF NOT DATA ERROR ONLY: BR
3819 016102 013704 000674  MOV     UNP, R4
3820 016106 032737 010000 000562 BIT     #10000, R0CMD ;SEE IF READ REVERSE
3821 016114 001003          BNE     DF4            ;IF SO: BR
3822 016116 005264 001130  INC     DATER1(R4)    ;BUMP DATA ERROR FORWARD COUNTER
3823 016122 000402          BR      DFX
3824 016124 005264 001170  DFO4: INC     DEREV1(R4) ;BUMP REVERSE DATA ERROR
3825 016130 000207          DFX:   RTS            ;EXIT
3826
    
```

3827  
3828  
3829  
3830  
3831  
3832  
3833  
3834  
3835  
3836  
3837  
3838  
3839  
3840  
3841  
3842  
3843  
3844  
3845  
3846  
3847  
3848  
3849  
3850  
3851  
3852  
3853  
3854  
3855  
3856  
3857  
3858  
3859  
3860  
3861  
3862  
3863  
3864  
3865  
3866  
3867  
3868  
3869  
3870  
3871  
3872  
3873  
3874  
3875  
3876  
3877  
3878  
3879  
3880  
3881  
3882

016132 032777 002000 162446  
016140 001067  
016142 005237 000670  
016146 005737 000664  
016152 001007  
016154 004737 022662  
016160 012704 024662  
016164 104000  
016166 004737 021130  
016172 012704 024701  
016176 104000  
016200 010203  
016202 162703 033436  
016206 005303  
016210 032737 010000 000562  
016216 001402  
016220 010503  
016222 005103  
016224 104002  
016226 012704 024667  
016232 104000  
016234 032737 010000 000562  
016242 001402  
016244 111103  
016246 000401  
016250 114103  
016252 004737 024112  
016256 012704 024674

DEPR: BIT #2000,JSWR  
BNE DERR4  
DERR0: INC PFLG  
TST HDRFL  
BNE DERR0A  
JSR PC,PAPRT  
MOV #MSG1,R4  
TTOUTT  
JSR PC,FRPRT  
DERR0A: MOV #MSG4,R4  
TTOUTT  
MOV R2,R3  
SUB #RDATA,R3  
DEC R3  
BIT #10000,RDCMD  
BEQ DERR0B  
MOV R5,R3  
COM R3  
DERR0B: OCTPP  
MOV #MSG2,R4  
TTOUTT  
BIT #10000,RDCMD  
BEQ DERR0C  
MOVB (R1),R3  
BR DERR0C  
DERR0C: MOVB -(R1),R3  
DERR0D: JSR PC,DOUT  
MOV #MSG3,R4

\*\*\*\*\*  
: 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.  
\*\*\*\*\*  
: SEE IF SHOULD PRINT ERRORS  
: ++G BRANCH IF NOT  
: SET PRINT FLAG  
: SEE IF HAVE PRINTED HEADER  
: IF SO: BR  
: PRINT CYCLE NUMBER  
: LOAD ERROR MSG ADDR  
: PRINT ERROR  
: PRINT F OR R  
: PRINT CHAR NO. HEADER  
: POINT TO CHAR  
: SEE IF READ REVERSE  
: IF NOT: BR  
: GET CHAR NUMBER  
: PRINT CHAR NUMBER  
: PRINT EXPECTED DATA  
: SEE IF READ REVERSE  
: IF NOT: BR  
: GET CHAR  
: LOAD EXPECTED DATA  
: GO PRINT CHAR

3883	016262	104000				TTOUTT		: PRINT RECIEVED DATA
3884	016264	032737	010000	000562		BIT	#10000,RDCMD	: SEE IF READ REVERSE
3885	016272	001402				BEQ	DERR1	: IF NOT: BR
3886	016274	111203				MCVB	(R2),R3	: GET CHAR
3887	016276	000401				BR	DERR2	
3888	016300	114203			DERR1:	MOVB	-(R2),R3	
3889	016302	004737	024112		DERR2:	JSR	PC,DOUT	: PRINT BAD CHAR
3890	016306	032737	010000	000562		BIT	#10000,RDCMD	: SEE IF READ REVERSE
3891	016314	001001				BNE	DERR4	: ++G BRANCH IF NOT
3892	016316	122122			DERR3:	CMPB	(R1)+,(R2)+	: RESET POINTERS
3893	016320	105237	000656		DERR4:	INCB	BBC	: BUMP BAD RECORD CNTR
3894	016324	122737	000010	000656		CMPB	#10,BBC	: SEE IF BLD BTH
3895	016332	001120				BNE	DEREX	: IF NOT: BR
3896	016334	032777	002000	162244		BIT	#2000,ASWR	: SEE IF PRINT INHIBIT
3897	016342	001003				BNE	IS	: IF SO: BR
3898	016344	012704	025013			MOV	#MSG15,R4	
3899	016350	104000				TTOUTT		: PRINT BLD BTH
3900	016352	105037	000656		IS:	CLRB	BBC	: RESET BAD RECORD CNTR
3901	016356	000337	000656			SWAB	BBC	: POSITION BLD BTH AMOUNT
3902	016362	105237	000656			INCB	BBC	: BUMP AMOUNT
3903	016366	122737	000003	000656		CMPB	#3,BBC	: SEE IF HAD 3 BLD BTHS
3904	016374	101052				BHI	DERR4B	: IF NOT: BR
3905	016376	000337	000656			SWAB	BBC	: REPOSITION BBC
3906	016402	022705	177767			CMP	#177767,R5	: SEE IF ON LAST EIGHT CHARS
3907	016406	101470				BLOS	DERR6	: IF SO: BR
3908	016410	012705	177767			MOV	#177767,R5	: SET CHAR CNTR TO 8
3909	016414	032737	010000	000562		BIT	#10000,RDCMD	: SEE IF READ REVERSE
3910	016422	001416				BEQ	DERR4A	: IF NOT: BR
3911	016424	012701	027430			MOV	#WDATA,R1	: GET START OF BUFFER
3912	016430	012702	033436			MOV	#RDATA,R2	: GET START OF BUFFER
3913	016434	062701	000010			ADD	#10,R1	
3914	016440	062702	000010			ADD	#10,R2	: POINT TO START +10
3915	016444	032737	000001	000556		BIT	#1,FMCNT	: SEE IF ODD FRAME COUNT
3916	016452	001450				BEQ	DEREX	: IF NOT: BR
3917	016454	105722				TSTB	(R2)+	: BUMP POINTER
3918	016456	000446				BR	DEREX	
3919	016460	013737	000556	000642	DERR4A:	MOV	FMCNT,TEMP1	: LOAD CHAR COUNT
3920	016466	005437	000642			NEG	TEMP1	: ++G FORM TWO'S COMPLEMENT
3921	016472	162737	000010	000642		SUB	#10,TEMP1	: POINT TO BUFFER -8
3922	016500	013701	000642			MOV	TEMP1,R1	: POINT TO NEXT CHAR
3923	016504	062701	027430			ADD	#WDATA,R1	: POINT TO NEXT WRITE CHAR
3924	016510	013702	000642			MOV	TEMP1,R2	: POINT TO END OF READ DATA -8 FORWARD
3925	016514	062702	033436			ADD	#RDATA,R2	: POINT TO NEXT CHAR
3926	016520	000425				BR	DEREX	: EXIT
3927	016522	000337	000656		DERR4B:	SWAB	BBC	: REPOSITION BBC
3928	016526	062705	000024			ADD	#24,R5	: SKIP 20 CHARS
3929	016532	103416				BCS	DERR6	: IF EXCEED RECORD SIZE: BR
3930	016534	032737	010000	000562		BIT	#10000,RDCMD	: SEE IF READ REVERSE
3931	016542	001405				BEQ	DERR5	: IF NOT: BR
3932	016544	162701	000024			SUB	#24,R1	
3933	016550	162702	000024			SUB	#24,R2	: RESET POINTERS
3934	016554	000407				BR	DEREX	
3935	016556	062701	000024		DERR5:	ADD	#24,R1	: SKIP 20 CHARS
3936	016562	062702	000024			ADD	#24,R2	: SKIP FORWARD 20 CHARS
3937	016566	000402				BR	DEREX	
3938	016570	012705	177777		DERR6:	MOV	#-1,R5	: SET TO EOR



```

3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969 016640 005037 000642 DRPKF: CLR TEMP1
3970 016644 005037 000644 CLR TEMP2
3971 016650 005037 000646 CLR TEMP3
3972 016654 111137 000642 MOV (R1),TEMP1 ;LOAD GOOD CHAR
3973 016660 111237 000644 MOV (R2),TEMP2 ;LOAD BAD CHAR
3974 016664 013704 000674 MOV UNP,R4
3975 016670 016437 000770 000720 MOV PIK1(R4),BPKP
3976 016676 016437 001010 000716 MOV DRP1(R4),BDPP
3977 016704 032737 010000 000562 BIT #10000,R0CMD ;SEE IF READ REVERSE
3978 016712 001005 BNE DRPK ;IF SO: BR
3979 016714 124142 CMPB -(R1),-(R2) ;POINT TO CHAR
3980 016716 112137 000642 MOV (R1)+,TEMP1 ;LOAD GOOD CHAR
3981 016722 112237 000644 MOV (R2)+,TEMP2 ;LOAD BAD CHAR
3982 016726 004737 016740 DRPK: JSR PC,DROP ;GET DROPS
3983 016732 004737 017156 JSR PC,PICK ;GET PICKS
3984 016736 000207 RTS PC ;EXIT
3985 016740 113703 000642 DROP: MOV TEMP1,R3 ;R3 = GOOD CHAR
3986 016744 113704 000644 MOV TEMP2,R4 ;R4 = BAD CHAR
3987 016750 140403 DPC: BICB R4,R3 ;GET DROPS/PICKS
3988 016752 001001 BNE DPCG ;IF SOME: BR
3989 016754 000207 RTS PC ;RETURN
3990 016756 012737 000010 000710 DPCG: MOV #10,BCNT ;SET NUMBER TO CHECK
3991 016764 132703 000001 DPCD: BITB #1,R3 ;SEE IF DROPPED OR PICKED THIS BIT
3992 016770 001455 BEQ DPC2 ;IF NOT: BR
3993 016772 105737 000646 TSTB TEMP3 ;SEE IF ON PICKS
3994 016776 001016 BNE DPC1 ;IF SO: BR
3995 017000 005277 161712 INC #BDPP ;BUMP DROP CNTR
3996 017004 005777 161706 TST #BDPP
3997 017010 100045 BPL DPC2 ;IF NO OVERFLOW: BR
3998 017012 032777 002000 161566 BIT #2000,#SWR ;SEE IF HAVE PRINTED DATA
3999 017020 001402 BEQ DPC0A ;IF SO: BR
4000 017022 004737 022662 JSR PC,PAPRT ;PRINT CYCLE NUMBER
4001 017026 004737 017222 DPCOA: JSR PC,DPPRT ;PRINT DROPS AND PICKS
4002 017032 000415 BR DPC2A
4003 017034 005277 161660 DPC1: INC #BPKP ;BUMP PICK CNTR
4004 017040 005777 161654 TST #BPKP ;SEE IF OVERFLOW
4005 017044 100027 BPL DPC2 ;IF NOT: BR
4006 017046 032777 002000 161532 BIT #2000,#SWF ;SEE IF HAVE PRINTED DATA

```



K07

4007	017054	001402			BEG	DPC1A		; IF SO: BR
4008	017056	004737	022662		JSR	PC,PAPRT		; PRINT CYCLE NUMBER
4009	017062	004737	017222		DPC1A: JSR	PC,DPPRT		; PRINT DROPS AND PICKS
4010	017066	013704	000674		DPC2A: MOV	UNP,R4		
4011	017072	016403	001010		MOV	DRP1(R4),R3		; SET DROP POINTER
4012	017076	016404	000770		MOV	PIK1(R4),R4		; SET PICK POINTER
4013	017102	012737	000010	000710	MOV	#10,BCNT		; SET NUMBER OF BITS
4014	017110	005023			DPC2B: CLR	(R3)+		; CLEAR DROPS
4015	017112	005024			CLR	(R4)+		; CLEAR PICK
4016	017114	005337	000710		DEC	BCNT		; SEE IF DONE
4017	017120	001373			BNE	DPC2B		; IF NOT: BR
4018	017122	000207			RTS	PC		; EXIT
4019	017124	000241			DPC2: CLC			
4020	017126	106003			RORB	R3		; GET NEXT BIT
4021	017130	005337	000710		DEC	BCNT		; SEE IF DONE
4022	017134	001407			BEG	DPC3		
4023	017136	062737	000002	000720	ADD	#2,BPKP		
4024	017144	062737	000002	000716	ADD	#2,BOPP		
4025	017152	000704			BR	DPC0		; ++G CONTINUE
4026	017154	000207			DPC3: RTS	PC		; RETURN
4027	017156	013704	000674		PICK: MOV	UNP,R4		; GET UNIT POINTER
4028	017162	016437	000770	000720	MOV	FIK1(R4),BPKP		; SET PICK POINTER
4029	017170	016437	001010	000716	MOV	DRP1(R4),BOPP		; SET DROP POINTER
4030	017176	113704	000642		MOVB	TEMP1,R4		; R4 = GOOD CHAR
4031	017202	113703	000644		MOVB	TEMP2,R3		; R3 = BAD CHAR
4032	017206	112737	000001	000646	MOVB	#1,TEMP3		; SET PICK FLAG
4033	017214	004737	016750		JSR	PC,DPC		; GO CHECK PICKS
4034	017220	000207			RTS	PC		; EXIT
4035	017222	012704	025307		DPPRT: MOV	#MSG26,R4		
4036	017226	104000			TTOUTT			; PRINT DROP HEADER
4037	017230	013704	000674		MOV	UNP,R4		
4038	017234	016437	001010	000716	MOV	DRP1(R4),BOPP		; SET DROP POINTER
4039	017242	016437	000770	000720	MOV	PIK1(R4),BPKP		; SET PICK POINTER
4040	017250	062737	000016	000716	ADD	#16,BOPP		
4041	017256	062737	000016	000720	ADD	#16,BPKP		
4042	017264	012737	000010	000710	MOV	#10,BCNT		; SET NUMBER TO PRINT
4043	017272	017703	161420		DPPRT0: MOV	#BOPP,R3		
4044	017276	104002			OCTPP			; PRINT DROPS
4045	017300	005337	000710		DEC	BCNT		; SEE IF DONE
4046	017304	001404			BEG	DPPRT1		; IF NOT: BR
4047	017306	162737	000002	000716	SUB	#2,BOPP		; BUMP POINTER
4048	017314	000726			BR	DPPRT0		; CONTINUE FOR ALL 8 BITS
4049	017316	012737	000010	000710	DPPRT1: MOV	#10,BCNT		; SET NUMBER TO PRINT
4050	017324	012704	025320		MOV	#MSG27,R4		
4051	017330	104000			TTOUTT			; PRINT PICK HEADER
4052	017332	017703	161362		DPPRT2: MOV	#BPKP,R3		
4053	017336	104002			OCTPP			; PRINT PICKS
4054	017340	005337	000710		DEC	BCNT		; SEE IF DONE
4055	017344	001404			BEG	DPPRTX		; IF SO: BR
4056	017346	162737	000002	000720	SUB	#2,BPKP		; BUMP POINTER
4057	017354	000766			BR	DPPRT2		; CONTINUE FOR ALL 8 BITS
4058	017356	000207			DPPRTX: RTS	PC		; RETURN

4059  
4060  
4061  
4062  
4063  
4064  
4065  
4066  
4067  
4068  
4069  
4070  
4071  
4072  
4073  
4074  
4075  
4076  
4077  
4078  
4079  
4080  
4081  
4082  
4083  
4084  
4085  
4086  
4087  
4088  
4089  
4090  
4091  
4092  
4093  
4094  
4095  
4096  
4097  
4098  
4099  
4100  
4101  
4102  
4103  
4104  
4105  
4106  
4107  
4108  
4109  
4110  
4111  
4112  
4113  
4114

017360 013703 000556  
017364 032703 000001  
017370 001401  
017372 005303  
017374 005403  
017376 032737 000020 000552  
017404 001402  
017406 000241  
017410 006003  
017412 032737 000010 000672  
017420 001414  
017422 032737 010000 000562  
017430 001405  
017432 012703 033436  
017436 162703 000002  
017442 000405  
017444 062703 033436  
017450 000402  
017452 062703 027430  
017456 010337 021104  
017462 012704 000007  
017466 012701 021106  
017472 005021  
017474 005304  
017476 001375  
017500 020377 161010  
017504 001402  
017506 005237 021106  
017512 032737 000010 000672  
017520 001006  
017522 005777 160770

```
*****  
: STATUS CHECK SUBROUTINE:  
: THIS SUBROUTINE IS USED TO PERFORM A CHECK OF  
: BOTH THE MASSBUS CONTROLLER (RH11) AND THE TAPE  
: CONTROLLER (TM02). THE RH11 IS CHECKED FOR ERRORS  
: AS REFLECTED IN REGISTERS CS1 AND CS2 AND ALSO THAT  
: THE BUS ADDRESS (BA) AND WORD COUNT (WC) ARE  
: CORRECT. THE TM02 IS CHECKED FOR DRIVE 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 ODD  
BEQ ERO ;IF NOT: BR  
DEC R3 ;BUMP COUNT  
ERO: NEG R3  
BIT #20,UDES ;SEE IF CORE DUMP  
BEQ EROB ;IF NOT: BR  
CLC  
ROR R3 ;SET TO FC/2  
EROB: 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  
EROA: ADD #RDATA,R3 ;BUILD EXPT READ ADDRESS  
BR ER2  
ER1: ADD #WDATA,R3 ;BUILD EXPT WRITE ADDRESS  
ER2: MOV R3,CADR ;SAVE ADDRESS  
MOV #7,R4  
MOV #BAER,R1  
ER2A0: CLR (R1)+ ;CLEAR FLAGS  
DEC R4  
BNE ER2A0  
CMP R3,#BA ;SEE IF ADDRESS OK  
BEQ ER2A1 ;IF SO: BR  
INC BAER ;SET BUS ADDRESS ERROR  
ER2A1: BIT #10,MTC1 ;SEE IF WRITE OPER  
BNE ER2B ;IF NOT: BR  
ER2A: TST #FC ;SEE IF FC=0
```

```

4115 017526 001441 BEQ ER3 ;IF SO: BR
4116 017530 005237 021114 INC FCER ;SET FC ERROR
4117 017534 000436 BR ER3 ;++G
4118 017536 032737 000040 000672 ER2B: BIT #40,MTC1 ;SEE IF SPACE OPER
4119 017544 001766 BEQ ER2A ;IF SO: BR
4120 017546 005737 000676 TST TMFLG ;SEE IF TM TIME
4121 017552 001011 BNE ER2D ;IF SO: BR
4122 017554 013703 000556 MOV FMCNT,R3
4123 017560 005403 NEG R3 ;R3 = EXPT RECORD SIZE
4124 017562 020377 160730 ER2C: CMP R3,FC ;SEE IF FC = EXPT
4125 017566 001421 BEQ ER3 ;IF SO: BR
4126 017570 005237 021114 INC FCER ;SET FC ERROR FLAG
4127 017574 000416 BR ER3 ;++G
4128 017576 032737 002000 000552 ER2D: BIT #2000,UDES ;SEE IF PE
4129 017604 001346 BNE ER2A ;IF SO: BR
4130 017606 032737 010000 000562 BIT #10000,RDCMD ;SEE IF READ REVERSE
4131 017614 001003 BNE ER2E ;IF SO: BR
4132 017616 012703 000002 MOV #2,R3
4133 017622 000757 BR ER2C ;LOOK FOR EXPT = 2
4134 017624 012703 000001 ER2E: MOV #1,R3
4135 017630 000754 BR ER2C ;GO CHECK FC FOR TM
4136 017632 032777 160000 160650 ER3: BIT #160000,AC1 ;SEE IF COUNT ERROR
4137 017640 001441 BEQ ER4
4138 017642 017703 160652 MOV ACS,R3 ;GET CONT STATUS REG
4139 017646 042703 000307 BIC #307,R3 ;MASK OUT IR,OR,UNIT NO.
4140 017652 005703 TST R3 ;SEE IF ANY OTHER ERRORS
4141 017654 001407 BEQ ER3A ;IF NOT: BR
4142 017656 005737 000676 TST TMFLG ;SEE IF TAPE MARK TIME
4143 017662 001426 BEQ ER3B ;IF NOT: BR
4144 017664 042703 001000 BIC #1000,R3 ;MASK MISSED TRANS
4145 017670 005703 TST R3 ;SEE IF ANY OTHER ERRORS
4146 017672 001022 BNE ER3B ;IF SO: BR
4147 017674 032777 060000 160606 ER3A: BIT #60000,AC1 ;SEE IF EITHER TRE OR MCPE
4148 017702 001420 BEQ ER4 ;IF NOT: BR
4149 017704 005737 000676 TST TMFLG ;SEE IF TM TIME
4150 017710 001413 BEQ ER3B ;IF NOT: BR
4151 017712 017703 160606 MOV DER,R3 ;GET ERROR REGISTER
4152 017716 032737 000010 000552 BIT #10,UDES ;SEE IF EVEN PARITY
4153 017724 001402 BEQ ER3A1 ;IF NOT: BR
4154 017726 042703 000100 BIC #100,R3 ;MASK PAR
4155 017732 042703 001000 ER3A1: BIC #1000,R3 ;MASK FCE
4156 017736 001402 BEQ ER4 ;IF NO ERRORS EXCEPT FCE: BR
4157 017740 005237 021110 ER3B: INC CONER ;SET CONT ERROR FLAG
4158 017744 032777 040000 160550 ER4: BIT #40000,ADS ;SEE IF DRIVE ERROR
4159 017752 001420 BEQ ER6 ;IF NOT: BR
4160 017754 005737 000676 TST TMFLG ;SEE IF TAPE MARK TIME
4161 017760 001413 BEQ ER4A ;IF NOT: BR
4162 017762 017703 160536 MOV DER,R3 ;GET ER
4163 017766 032737 000010 000552 BIT #10,UDES ;SEE IF EVEN PARITY
4164 017774 001402 BEQ ER4A1 ;IF NOT: BR
4165 017776 042703 000100 BIC #100,R3 ;MASK PAR
4166 020002 042703 001000 ER4A1: BIC #1000,R3 ;MASK OUT FCE
4167 020006 001402 BEQ ER6 ;++G & BR IF NO OTHER ERR BITS ARE SET
4168 020010 005237 021112 ER4A: INC DRIVER ;SET DRIVER ERROR FLAG
4169 020014 032737 002000 000552 ER6: BIT #2000,UDES
4170 020022 001407 BNE ERPT ;IF IN PE MODE: BR

```

```

4171 020024 032777 020000 160554          BIT      #20000, @SWR      ;SEE IF NO DATA CHECK
4172 020032 001065                      BNE      ERPT          ;IF NOT: BR (ALLOW READ OF UNKNOWN TAPES)
4173 020034 032737 000040 000672          BIT      #40, MTC1     ;SEE IF WRITE OR READ OP
4174 020042 001461                      BEQ      ERPT          ;IF NOT: BR
4175 020044 005737 000676                      TST      TMFLG        ;SEE IF TAPE MARK TIME
4176 020050 001413                      BEQ      ER6A         ;IF NOT: BR
4177 020052 013737 015510 021126          MOV      EXCRC, CRCSV  ;SAVE CRC
4178 020060 013737 015512 021124          MOV      EXLRC, LRCSV ;SAVE LRC
4179 020066 005037 015510                      CLR      EXCRC
4180 020072 012737 000023 015512          MOV      #23, EXLRC   ;SET CRC/LRC FOR TM
4181 020100 032737 000060 000552  ER6A:    BIT      #60, UDES     ;SEE IF FORMAT 14
4182 020106 001037                      BNE      ERPT          ;IF NOT: BR
4183 020110 017703 160414                      MOV      @CC, R3      ;GET CRC CHARACTER
4184 020114 042703 177000                      BIC      #177000, R3
4185 020120 023703 015510                      CMP      EXCRC, R3
4186 020124 00140?                      BEQ      ER7
4187 020126 005237 021120                      INC      CRCER        ;IF CRC GOOD: BR
4188 020132 017703 160376  ER7:      MOV      @MR, R3      ;SET ERROR FLAG
4189 020136 000303                      SWAB
4190 020140 005703                      R3
4191 020142 100002                      TST      R3
4192 020144 052703 000400                      BPL      ER10
4193 020150 042703 177000  ER10:    BIC      #177000, R3
4194 020154 023703 015512                      CMP      EXLRC, R3
4195 020160 001412                      BEQ      ERPT          ;IF LRC GOOD: BR
4196 020162 010337 021122                      MOV      R3, ACTLRC   ;SAVE ACTUAL LRC
4197 020166 005237 021116                      INC      LRCER        ;SET LRC ERROR FLAG
4198 020172 032737 010000 000562          BIT      #10000, RDCMD ;SEE IF READ REVERSE
4199 020200 001402                      BEQ      ERPT          ;IF NOT: BR
4200 020202 005037 021116                      CLR      LRCER        ;ELSE CLEAR LRC ERROR
4201 020206 012703 000006  ERPT:    MOV      #6, R3
4202 020212 005037 000706                      CLR      SERFL        ;CLEAR ERROR FLAG
4203 020216 005037 000722                      CLR      ERSAV
4204 020222 012704 021106  ERPTT:   MOV      #BAER, R4
4205 020226 005724                      TST      (R4)+        ;SEE IF ANY ERROR
4206 020230 001004                      BNE      ERPTG        ;IF SO: BR
4207 020232 005303                      DEC      R3
4208 020234 001374                      BNE      ERPTT
4209 020236 000137 021050                      JMP      ERPX1
4210 020242 005237 000706  ERPTG:   INC      SERFL        ;SET ERROR FLAG
4211 020246 017737 160252 000722          MOV      @ER, ERSAV   ;SAVE ERROR REGISTER
4212 020254 032777 002000 160324          BIT      #2000, @SWR  ;SEE IF PRINT
4213 020262 001420                      BEQ      ERPT0        ;IF SO: BR
4214 020264 022737 000002 000712          CMP      #2, RTYFL    ;SEE IF READ RETRY
4215 020272 001006                      BNE      ERPTG1       ;IF NOT: BR
4216 020274 013703 000702                      MOV      RTCNT, R3
4217 020300 005203                      INC      R3           ;BUMP RETRY COUNT
4218 020302 020337 000602                      CMP      R3, RETRY    ;SEE IF LAST RETRY
4219 020306 001406                      BEQ      ERPT0        ;IF SO: BR
4220 020310 022737 000002 021112  ERPTG1:  CMP      #2, DRVER    ;SEE IF TM STATUS ERROR
4221 020316 001402                      BEQ      ERPT0        ;IF SO: BR
4222 020320 000137 020752                      JMP      ERPX0
4223 020324 005237 000670  ERPT0:   INC      PFLG
4224 020330 004737 022662                      JSR      PC, PAPRT    ;PRINT HEADER
4225 020334 013704 000652                      MOV      EMADDR, R4
4226 020340 104000                      TTOUTT                ;PRINT ERROR TYPE
    
```

4227	020342	004737	021130		JSR	PC,FRPRT	;PRINT F OR R
4228	020346	005737	000676		TST	TMFLG	
4229	020352	001407			BEQ	ERPT1	
4230	020354	022737	026137	000652	CMP	#MSG54,EMADDR	
4231	020362	001403			BEQ	ERPT1	
4232	020364	012704	026155		MOV	#MSG56,R4	;PRINT TM
4233	020370	104000			TTOUTT		
4234	020372	005737	021110		ERPT1: TST	CONER	
4235	020376	001414			BEQ	ERPT2	;IF NO CONT ERROR: BR
4236	020400	012704	025137		MOV	#MSG23,R4	
4237	020404	104000			TTOUTT		;PRINT C1 TAG
4238	020406	017703	160076		MOV	@C1,R3	
4239	020412	104002			OCTPP		;PRINT CONTROL 1
4240	020414	012704	025164		MOV	#MSG230,R4	;PRINT CS TAG
4241	020420	104000			TTOUTT		
4242	020422	017703	160072		MOV	@CS,R3	
4243	020426	104002			OCTPP		;PRINT CONT STATUS
4244	020430	005737	021112		ERPT2: TST	DRVER	
4245	020434	001414			BEQ	ERPT3	;IF SO DRIVE ERROR: BR
4246	020436	012704	025172		MOV	#MSG23E,R4	
4247	020442	104000			TTOUTT		;PRINT DS TAG
4248	020444	017703	160052		MOV	@DS,R3	
4249	020450	104002			OCTPP		;PRINT DRIVE STATUS
4250	020452	012704	025177		MOV	#MSG23F,R4	
4251	020456	104000			TTOUTT		;PRINT ER TAG
4252	020460	017703	160040		MOV	@ER,R3	
4253	020464	104002			OCTPP		;PRINT DRIVE ERROR
4254	020466	005737	021106		ERPT3: TST	BAER	
4255	020472	001416			BEQ	ERPT4	;IF NO BA ERROR: BR
4256	020474	012704	025152		MOV	#MSG23B,R4	
4257	020500	104000			TTOUTT		;PRINT BA TAG
4258	020502	017703	160006		MOV	@BA,R3	
4259	020506	104002			OCTPP		;PRINT BUS ADDRESS
4260	020510	012737	000255	000636	MOV	#255,TOB	
4261	020516	004737	023614		JSR	PC,TOG	;PRINT /
4262	020522	013703	021104		MOV	CADER,R3	
4263	020526	104002			OCTPP		;PRINT EXPT BUS ADDRESS
4264	020530	005737	021114		ERPT4: TST	FCER	
4265	020534	001406			BEQ	ERPT5	;IF NO FC ERROR: BR
4266	020536	012704	025157		MOV	#MSG23C,R4	
4267	020542	104000			TTOUTT		;PRINT FC TAG
4268	020544	017703	157746		MOV	@FC,R3	
4269	020550	104002			OCTPP		;PRINT FRAME COUNT
4270	020552	012704	025145		ERPT5: MOV	#MSG23A,R4	
4271	020556	104000			TTOUTT		;PRINT WC TAG
4272	020560	017703	157726		MOV	@WC,R3	
4273	020564	104002			OCTPP		;PRINT WORD COUNT
4274	020566	005737	021120		TST	CRCER	
4275	020572	001420			BEQ	ERPT5A	;IF NO CRC ERROR: BR
4276	020574	012704	026202		MOV	#MSG58,R4	
4277	020600	104000			TTOUTT		;PRINT CRC TAG
4278	020602	017703	157722		MOV	@CC,R3	
4279	020606	042703	177000		BIC	#177000,R3	
4280	020612	104002			OCTPP		;PRINT ACTUAL CRC
4281	020614	012737	000255	000636	MOV	#255,TOB	
4282	020622	004737	023614		JSR	PC,TOG	

```

4283 020626 013703 015510          MOV      EXCRC,R3
4284 020632 104002          OCTPP
4285 020634 005737 021116          ERPT5A: TST      LRCER          ;PRINT EXPECTED CRC
4286 020640 001416          BEQ      ERPT6          ;IF NO LRC ERROR: BR
4287 020642 012704 026210          MOV      #MSG59,R4
4288 020646 104000          TTOUTT          ;PRINT LRC TAG
4289 020650 013703 021122          MOV      ACTLRC,R3
4290 020654 104002          OCTPP          ;PRINT ACTUAL LRC
4291 020656 012737 000255 000636          MOV      #255,TOB
4292 020664 004737 023614          JSR      PC,TOG
4293 020670 013703 015512          MOV      EXLRC,R3
4294 020674 104002          OCTPP
4295 020676 005737 021112          ERPT6: TST      DRIVER          ;PRINT EXPECTED LRC
4296 020702 001422          BEQ      ERPT7          ;IF NO DRIVE ERROR: BR
4297 020704 032737 002000 000552          BIT      #2000,UDES
4298 020712 001416          BEQ      ERPT7          ;IF NO PE: BR
4299 020714 017704 157604          MOV      #ER,R4
4300 020720 042704 075477          BIC      #75477,R4          ;MASK OUT ALL BUT BITS 15,10,7,6
4301 020724 005704          TST      R4
4302 020726 001410          BEQ      ERPT7          ;IF NO CONDITIONALS SET: BR
4303 020730 012704 025211          MOV      #MSG23H,R4
4304 020734 104000          TTOUTT          ;PRINT CC TAG
4305 020736 017703 157566          MOV      #CC,R3
4306 020742 042703 177000          BIC      #177000,R3          ;MASK CC
4307 020746 104002          OCTPP          ;PRINT CHECK CHARACTERS
4308 020750 000240          ERPT7: NOP
4309 020752 032777 100000 157626          ERPX0: BIT      #100000,#SWR          ;SEE IF STOP ON ERROR
4310 020760 001412          BEQ      ERPX          ;IF NOT: BR
4311 020762 104006          STOPP
4312 020764 005737 000670          TST      PFLG          ;SEE IF HAVE PRINTED
4313 020770 001006          BNE      ERPX          ;IF SO: BR
4314 020772 032777 002000 157606          BIT      #2000,#SWR          ;SEE IF SHOULD PRINT
4315 021000 001002          BNE      ERPX          ;IF NOT: BR
4316 021002 000137 020324          JMP      ERPT0          ;PRINT ERROR
4317 021006 005037 000670          ERPX: CLR      PFLG
4318 021012 012777 000011 157470          MOV      #11,#C1          ;DRIVE CLEAR
4319 021020 017704 157502          MOV      #AS,R4
4320 021024 010477 157476          MOV      R4,#AS          ;CLEAR AS
4321 021030 013704 000510          MOV      C1,R4
4322 021034 005204          INC      R4
4323 021036 152714 000100          BISB     #100,(R4)          ;RESET TRE
4324 021042 013777 000552 157472          MOV      UDES,#C2          ;RESET TC
4325 021050 032737 000040 000672          ERPX1: BIT      #40,#TC1
4326 021056 001411          BEQ      ERPX2          ;IF NOT READ/WRITE OP: BR
4327 021060 005737 000676          TST      TMFLG
4328 021064 001406          BEQ      ERPX2          ;IF NOT TM TIME: BR
4329 021066 013737 021126 015510          MOV      CRCV,EXCRC          ;RESTORE CRC
4330 021074 013737 021124 015512          MOV      LRCV,EXLRC          ;RESTORE LRC
4331 021102 000207          ERPX2: RTS      PC          ;EXIT
4332 021104 000000          CADER: 0          ;EXPT ADDRESS SAVE
4333 021106 000000          BAER: 0
4334 021110 000000          CONER: 0
4335 021112 000000          DRIVER: 0
4336 021114 000000          FCER: 0
4337 021116 000000          LRCER: 0
4338 021120 000000          CRCER: 0
    
```

4339 021122 000000 ACTLRC: 0  
4340 021124 000000 LRCSV: 0  
4341 021126 000000 CRCSV: 0

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

4342 021130 032737 000010 000672 FRPRT: BIT #10,MTC1 ;SEE IF WRITE COMMAND  
4343 021136 001413 BEQ FREX ;IF SO: BR  
4344 021140 032737 000002 000672 BIT #2,MTC1 ;SEE IF REVERSE  
4345 021146 001404 BEQ FRO ;IF NOT: BR  
4346 021150 012704 025047 MOV #MSG17,R4  
4347 021154 104000 TTOUTT ;PRINT R  
4348 021156 000403 BR FREX  
4349 021160 012704 025044 FRO: MOV #MSG16,R4  
4350 021164 104000 TTOUTT ;PRINT F  
4351 021166 000207 FREX: RTS PC ;EXIT  
4352

4363  
4364  
4365  
4366  
4367  
4368  
4369  
4370  
4371  
4372  
4373  
4374  
4375  
4376  
4377  
4378  
4379  
4380  
4381  
4382  
4383  
4384  
4385  
4386  
4387  
4388  
4389  
4390  
4391  
4392  
4393  
4394  
4395  
4396  
4397  
4398  
4399  
4400  
4401  
4402  
4403  
4404  
4405  
4406  
4407  
4408  
4409  
4410  
4411  
4412  
4413  
4414  
4415  
4416  
4417  
4418

021170 005037 000642 TAPG:  
021174 013777 000550 157316  
021202 032777 010000 157312 TAPG0:  
021210 001026  
021212 005237 000642  
021216 001371  
021220 004737 022662  
021224 032737 000010 000672  
021232 001004  
021234 012704 024706  
021240 104000  
021242 000405  
021244 012704 024713 TAPG1:  
021250 104000  
021252 004737 021130  
021256 012704 025267 TAPG2:  
021262 104000  
021264 104006  
021266 032777 020000 157226 TAPG3:  
021274 001411  
021276 004737 022662  
021302 012704 027267  
021306 104000  
021310 032777 020000 157204 15:  
021316 001374  
021320 022737 000026 000672 TAPG3F:  
021326 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      DVN,ACS      ;SET DRIVE NO.
BIT      #10000,ADS   ;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
TTOUTT
;PRINT WRITE ERR
BR      TAPG2
MOV      #MSG6,R4
TTOUTT
;PRINT READ ERR
;PRINT F OR R
JSR      PC,FRPRT
MOV      #MSG25,R4
TTOUTT
;PRINT NO MOL ERR
STOPP
BIT      #20000,ADS   ;SEE IF PIP RESET
BEQ      TAPG3F      ;IF SO: BR
JSR      PC,PAPRT    ;PRINT HEADER
MOV      #MSG116,R4
TTOUTT
;PRINT REWINDING MESSAGE
BIT      #20000,ADS
15
;AWAIT PIP RESET
BNE      15
;SEE IF WRITE TM
CMP      #26, MTC1
BNE      TAPG3A
;IF NOT: BR
```



4419	021330	012704	177777		MOV	#-1,R4	;ELSE SET FC FOR -1	
4420	021334	000406			BR	TAPG3B		
4421	021336	013704	000556	TAPG3A:	MOV	FMCNT,R4		
4422	021342	032704	000001		BIT	#1,R4		
4423	021346	001401			BEQ	TAPG3B		
4424	021350	005304			DEC	R4		
4425	021352	000261		TAPG3B:	SEC			
4426	021354	006004			ROR	R4	;SET WC = FC/2 FOR NORMAL FORMAT	
4427	021356	032737	000020	000552	BIT	#20,UDES	;SEE IF CORE DUMP FORMAT	
4428	021364	001402			BEQ	TAPG3C	;IF NOT: BR	
4429	021366	000261			SEC			
4430	021370	006004			ROR	R4	;SET WC = FC/4 FOR CORE DUMP	
4431	021372	010477	157114	TAPG3C:	MOV	R4,@WC	;SET WORD COUNT	
4432	021376	012777	000011	157104	MOV	#11,@C1	;DRIVE CLEAR	
4433	021404	017777	157106	157104	MOV	@FC,@FC	;RESET FC LOADED	
4434	021412	005737	000566		TST	INTRF	;SEE IF INTERCHANGE READ	
4435	021416	001407			BEQ	TAPG3D	;IF NOT: BR	
4436	021420	032737	000040	000672	BIT	#40,MTC1	;SEE IF READ OP	
4437	021426	001403			BEQ	TAPG3D	;IF NOT: BR	
4438	021430	012777	000003	157076	MOV	#3,@MR	;SET INTERCHANGE READ MAINT. MODE	
4439	021436	013704	000672	TAPG3D:	MOV	MTC1,R4	;GET COMMAND	
4440	021442	042704	177707		BIC	#177707,R4	;MASK OP CODE	
4441	021446	022704	000030		CMP	#30,R4	;SEE IF SPACE OP CODE	
4442	021452	001403			BEQ	TAPG3E	;IF 50: BR	
4443	021454	012737	177740	000666	MOV	#-40,STAL	;SET INTERRUPT DELAY MULT TO 40	
4444	021462	052737	000101	000672	TAPG3E:	BIS	#101,MTC1	;SET INTERRUPT ENABLE AND GO
4445	021470	000240			NOP			
4446	021472	013777	000672	157010	MOV	MTC1,@C1	;EXECUTE COMMAND	
4447	021500	005077	157100		CLR	@PSW	;CLEAR PRIORITY	
4448	021504	005037	000642		CLR	TEMP1		
4449	021510	005237	000642	TAPG4:	INC	TEMP1	;SEE IF HAVE TIMED OUT	
4450	021514	001375			BNE	TAPG4	;IF NOT: BR	
4451	021516	005237	000666		INC	STAL		
4452	021522	001372			BNE	TAPG4	;DO TIME DELAY MULTIPLIER	
4453	021524	012777	000340	157052	TAPG5:	MOV	#340,@PSW	;RESET PRIORITY
4454	021532	032777	002000	157046	BIT	#2000,@SWR	;SEE IF SHOULD PRINT ERRORS	
4455	021540	001012			BNE	TAPG6	;IF NOT: BR	
4456	021542	004737	022662		JSR	PC,PAPRT	;PRINT CYCLE NUMBER	
4457	021546	013704	000652		MOV	EMADDR,R4		
4458	021552	104000			TTOUTT		;PRINT ERROR OP	
4459	021554	004737	021130		JSR	PC,FRPRT	;PRINT F OR R	
4460	021560	012704	025247		MOV	#MSG24,R4		
4461	021564	104000			TTOUTT		;PRINT NO INTERRUPT	
4462	021566	005777	157014	TAPG6:	TST	@SWR	;++G BRANCH IF CONTINUE ON ERROR	
4463	021572	100001			BPL	TAPG7		
4464	021574	104006			STOPP			
4465	021576	000137	021670	TAPG7:	JMP	MTINTA	;RETURN TO CALLING ROUTINE	

```

4467
4468 ;TTY INTERRUPT HANDLER*****
4469
4470 021602 012777 000340 156774 TTINT: MOV #340, @PSW ;RESET PSW
4471 021610 017746 156776 MOV @TKB, -(SP) ;++G GET CHARACTER
4472 021614 042716 000200 BIC #200, (SP) ;++G STRIP PARITY BIT
4473 021620 122716 000003 CMPB #3, (SP) ;++G SEE IF CONT C
4474 021624 001412 BEQ TTINTO ;IF SO: BR
4475 021626 122716 000007 CMPB #7, (SP) ;++G CHECK FOR CNTL G
4476 021632 001013 BNE RETURN
4477 021634 022737 000176 000606 CMP #SWREG, SWR ;IS SOFTWARE SWITCH REGISTER USED
4478 021642 001007 BNE RETURN ;NO, GET OUT
4479 021644 004737 024376 JSR PC, CNTG ;GO CHANGE SWREG
4480 021650 000404 BR RETURN ;+G GO TO EXIT
4481
4482 021652 010046 TTINTO: MOV RO, -(SP) ;++G SAVE RO(REC CNTR)
4483 021654 004737 013746 JSR PC, TINT4 ;GO GET STALL VALUES
4484 021660 012600 MOV (SP)+, RO ;++G RESTORE RO(REC CNTR)
4485 021662 005726 RETURN: TST (SP)+ ;++G POP CHAR OFF STACK
4486 021664 000002 RTI ;RETURN
4487
4488 ;MAG TAPE INTERRUPT HANDLER*****
4489
4490 021666 000240 MTINT: NOP
4491 021670 042777 000037 156636 M*INTA: BIC #37, @MR ;CLEAR MAINT MODE
4492 021676 013716 000662 MOV RTRN, (SP) ;++G GET RETURN ADDRESS
4493 021702 000002 RTI ;++G RETURN

```

```

4494
4495
4496
4497
4498
4499
4500
4501
4502
4503 021704 012704 027077 ASEQ: MOV #MSG108,R4
4504 021710 104000 TTOUTT ;PRINT NRZ ONLY REQUEST
4505 021712 012705 000650 MOV #NRZOF,R5 ;SET ADDRESS OF FLAG
4506 021716 012701 000001 MOV #1,R1 ;SET SIZE OF ENTRY
4507 021722 012702 000001 MOV #1,R2 ;SET UPPER LIMIT
4508 021726 012703 000000 MOV #0,R3 ;SET LOWER LIMIT
4509 021732 004737 023300 JSR PC,TTR ;GO GET RESPONSE
4510 021736 012704 026674 MOV #MSG104,R4
4511 021742 104000 TTOUTT ;REQUEST CONT OR NOT
4512 021744 012705 000742 MOV #ASEQCF,R5 ;SET ADDRESS OF ENTRY
4513 021750 012701 000001 MOV #1,R1 ;SET SIZE OF ENTRY
4514 021754 012702 000001 MOV #1,R2 ;SET UPPER LIMIT
4515 021760 012703 000000 MOV #0,R3 ;SET LOWER LIMIT
4516 021764 004737 023300 JSR PC,TTR ;GO GET INPUT
4517 021770 005037 000736 ASEQ0: CLR ADRVN ;CLEAR DRV NUM
4518 021774 004737 022106 ASEQ1: JSR PC,HRDS ;GO SELECT HARDWARE CONFIGURATION
4519 022000 012704 026640 MOV #MSG101,R4
4520 022004 104000 TTOUTT ;PRINT DIVIDER
4521 022006 012704 026654 MOV #MSG102,R4
4522 022012 104000 TTOUTT ;PRINT TMO2 NUMBER
4523 022014 013703 000736 MOV ADRVN,R3
4524 022020 104002 OCTPP ;PRINT TMO2
4525 022022 012704 026663 MOV #MSG103,R4
4526 022026 104000 TTOUTT ;PRINT SLAVE HDR
4527 022030 012700 000746 MOV #UNI,R0 ;POINT TO START OF SLAVE TABLE
4528 022034 005710 ASEQ2: TST (R0) ;SEE IF END
4529 022036 100403 BMI ASEQ3 ;IF SO: BR
4530 022040 012003 MOV (R0)+,R3
4531 022042 104002 OCTPP ;PRINT SLAVE TABLE
4532 022044 000773 BR ASEQ2 ;DO ALL
4533 022046 004737 022312 ASEQ3: JSR PC,AMOD1 ;GO DO MODE 1(NRZ)
4534 022052 004737 022506 JSR PC,AMOD2 ;GO DO MODE 2(PE)
4535 022056 022737 000007 000736 ASEQ4: CMP #7,ADRVN ;SEE IF DONE ALL DRIVES
4536 022064 001403 BEQ ASEQX ;IF SO: BR
4537 022066 005237 000736 INC ADRVN ;BUMP DRIVE NUMBER
4538 022072 000740 BR ASEQ1 ;CONTINUE
4539 022074 005737 000742 ASEQX: TST ASEQCF ;BRANCH IF CONTINUOUS ALTO SEG
4540 022100 001733 BEQ ASEQ0 ;++G CONTINUE TESTING
4541 022102 000137 005054 JMP TEND ;++G GO TO END OF PASS

```

```

4542
4543
4544 ;SUBROUTINE TO SELECT AUTO SEQUENCE HARDWARE*****
4545 022106 005037 005130 HRDS: CLR REOTC ;CLEAR EOT UNIT CNTR
4546 022112 005037 000642 CLR TEMP1
4547 022116 012777 000040 156374 MOV #40,ACS ;INIT
4548 022124 013777 000736 156366 MOV ADRVN,ACS ;SET DRIVE
4549 022132 017701 156400 MOV JDT,R1 ;READ DRIVE TYPE
4550 022136 032777 010000 156354 BIT #10000,ACS ;TEST FOR NON-EXISTANT DRIVE
4551 022144 001403 BEQ HRDS1 ;IF DRIVE AVAIL: BR
4552 022146 005726 HRDS0: TST (SP)+ ;RESET STACK POINTER
4553 022150 000137 022056 JMP ASEQ4 ;GO SEE IF TRIED ALL DRIVES
4554 022154 042701 002007 HRDS1: BIC #2007,R1 ;MASK SLAVE TYPE
4555 022160 022701 140010 CMP #140010,R1 ;++G SEE IF TU16/TE16 TAPE
4556 022164 001370 BNE HRDS0 ;IF NOT: BR
4557 022166 005000 CLR RO
4558 022170 012701 000746 MOV #UN1,R1 ;SET START OF SLAVE TABLE
4559 022174 005737 003040 TST CHNFLG ;++G BRANCH IF NOT IN CHAIN MODE
4560 022200 001410 BEQ HRDS2
4561 022202 122737 000006 000041 CMPB #6,#41 ;++G BRANCH IF NOT LOADED VIA TMOP
4562 022210 001004 BNE HRDS2
4563 022212 005737 000736 TST ADRVN ;++G BRANCH IF NOT DRIVE 0
4564 022216 001001 BNE HRDS2 ;++G
4565 022220 005200 INC RO ;++G DO NOT TEST DRIVE 0 SLAVE 0
4566 ;++G IF TMOP CHAIN
4567 022222 010077 156314 HRCS2: MOV RO,ACS ;SELECT SLAVE
4568 022226 032777 010000 156266 BIT #10000,ACS ;SEE IF SLAVE AVAIL FOR TEST(MOL)
4569 022234 001403 BEQ HRDS3 ;IF NOT: BR
4570 022236 005237 000642 INC TEMP1 ;SET SLAVE FOUND FLAG
4571 022242 010021 MOV RO,(R1)+ ;LOAD SLAVE TABLE
4572 022244 022700 000007 HRDS3: CMP #7,RO ;SEE IF DONE ALL SLAVES
4573 022250 001402 BEQ HRDS4 ;IF SO: BR
4574 022252 005200 INC RO ;ELSE BUMP SLAVE NUMBER
4575 022254 000762 BR HRDS2 ;CONTINUE SELECTION
4576 022256 005737 000642 HRCS4: TST TEMP1 ;SEE IF FOUND ANY SLAVES
4577 022262 001731 BEQ HRDS0 ;IF NOT: BR
4578 022264 013737 000642 005130 MOV TEMP1,REOTC ;SET NUMBER OF UNITS
4579 022272 000337 000642 SWAB TEMP1
4580 022276 053737 000642 005130 BIS TEMP1,REOTC ;SET EOT CNTR
4581 022304 012711 177777 MOV #-1,(R1) ;TERMINATE SLAVE TABLE
4582 022310 000207 RTS PC ;RETURN TO SEC
    
```

```

4583
4584
4585 ;SUBROUTINE TO SELECT NRZ AUTO TEST MODE*****
4586 022312 005037 000654 AMOD1: CLR BLCNTR ;ASSURE BLOCK COUNTER IS 0
4587 022316 012701 000746 MOV #UNI,R1 ;GET START OF SLAVE TABLE
4588 022322 052721 001700 AMOD1A: BIS #1700,(R1)+ ;SET ALL SLAVE TO NRZ,NORM,000
4589 022326 005111 COM (R1)
4590 022330 001402 BEQ AMOD1B ;IF FILLED ALL SLAVES: BR
4591 022332 005111 COM (R1)
4592 022334 000772 BR AMOD1A ;ELSE DO ALL
4593 022336 005111 AMOD1B: COM (R1)
4594 022340 004737 005144 JSR PC,RWINDA ;GO REWIND ALL AVAIL SLAVES
4595 022344 012737 000006 000740 MOV #6,ABL CNT ;SET NUMBER OF BLOCKS FOR MODE 1
4596 022352 012737 174000 000556 MOV #-4000,FMCNT ;SET FC = 4000
4597 022360 012737 000100 000554 MOV #100,RCNT ;SET REC CNTR = 100
4598 022366 013737 000736 000550 MOV ADRVN,DVN ;SELECT DRIVE
4599 022374 012737 000001 000560 MOV #1,PATRN ;SELECT PATTERN 1
4600 022402 005037 000564 CLR TMEX ;ASSURE NO TMK
4601 022406 005037 000566 CLR INTRF ;ASSURE NORMAL READ
4602 022412 004737 003372 JSR PC,STAUTO ;GO DO AUTO MODE 1
4603 022416 012737 000010 000560 MOV #10,PATRN ;SELECT PATTERN 10
4604 022424 004737 003372 JSR PC,STAUTO ;GO DO PATTERN 10
4605 022430 012737 000014 000560 MOV #14,PATRN ;SELECT PATTERN 14
4606 022436 004737 003372 JSR PC,STAUTO
4607 022442 005737 000650 TST NRZOF ;SEE IF NRZ ONLY
4608 022446 001411 BEQ AMOD1C ;IF NOT: BR
4609 022450 012737 177777 000740 MOV #-1,ABL CNT ;FORCE TO EOT
4610 022456 012737 153624 000624 MOV #153624,RANBAS
4611 022464 012737 032561 000626 MOV #32561,RANSAV ;RESET RANDOM DATA BASE
4612 022472 012737 177777 000560 AMOD1C: MOV #-1,PATRN ;SELECT AUTO RANDOM DATA
4613 022500 004737 003372 JSR PC,STAUTO
4614 022504 000207 RTS PC ;RETURN TO SEG
    
```

```

4615
4616
4617
4618 022506 005737 003040
4619 022512 001003
4620 022514 005737 000650
4621 022520 001057
4622 022522 005037 000654
4623 022526 012701 000746
4624 022532 042711 001700
4625 022536 052721 002300
4626 022542 005111
4627 022544 001402
4628 022546 005111
4629 022550 000770
4630 022552 005111
4631 022554 004737 005144
4632 022560 012737 000006 000740
4633 022566 012737 174000 000556
4634 022574 012737 000100 000554
4635 022602 012737 000010 000560
4636 022610 004737 003372
4637 022614 012737 000014 000560
4638 022622 004737 003372
4639 022626 012737 000015 000560
4640 022634 004737 003372
4641 022640 012737 177777 000740
4642 022646 012737 177777 000560
4643 022654 004737 003372
4644 022660 000207
4645
4646

```

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

```

AMOD2: TST     CHNFLG      ;++G BRANCH IF IN CHAIN MODE
        BNE     IS        ;++G
        TST     NRZOF     ;SEE IF NRZ ONLY
        BNE     AMOD2X    ;IF SO: BR
IS:     CLR     BLCNTR    ;CLEAR BLOCK CNTR
        MOV     #UN1,R1   ;SET START OF SLAVE TABLE
AMOD2A: BIC     #1700,(R1) ;CLEAR NRZ
        BIS     #2300,(R1)+ ;SET TO PE NORM, ODD
        COM     (R1)      ;SEE IF END OF TABLE
        BEQ     AMOD2B    ;IF SO: BR
        COM     (R1)
        BR     AMOD2A     ;CONTINUE
AMOD2B: COM     (R1)
        JSR     PC,RWMDA  ;REWIND ALL SLAVES
        MOV     #6,ABLCNT ;SET AUTO BLOCK COUNT
        MOV     #-4000,FMCNT ;SET FC = 4000
        MOV     #100,RCNT ;SET REC CNTR TO 100
        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,ABLCNT ;FORCE TO END OF TAPE
        MOV     #-1,PATRN ;SELECT ALTO RANDOM DATA
        JSR     PC,STAJTO
AMOD2X: RTS     PC       ;RETURN TO SEQ

```

4647  
4648  
4649  
4650  
4651  
4652  
4653  
4654  
4655  
4656  
4657  
4658  
4659  
4660  
4661  
4662  
4663  
4664  
4665  
4666  
4667  
4668  
4669  
4670  
4671  
4672  
4673  
4674  
4675  
4676  
4677  
4678  
4679  
4680  
4681  
4682  
4683  
4684  
4685  
4686  
4687  
4688  
4689  
4690  
4691  
4692  
4693  
4694  
4695  
4696  
4697  
4698  
4699  
4700  
4701  
4702

022662 012704 024764  
022666 104000  
022670 013703 000550  
022674 104002  
022676 012704 024750  
022702 104000  
022704 013703 000552  
022710 042703 177770  
022714 104002  
022716 012704 026216  
022722 104000  
022724 013703 000552  
022730 000303  
022732 042703 177770  
022736 104002  
022740 012704 026222  
022744 104000  
022746 005003  
022750 032737 000010 000552  
022756 001402  
022760 012703 000001  
022764 104002  
022766 012704 026226  
022772 104000  
022774 013703 000552  
023000 000241  
023002 006003  
023004 006003  
023006 006003  
023010 006003  
023012 042703 177760  
023016 104002  
023020 012704 024725  
023024 104000  
023026 032777 000400 155552  
023034 001406  
023036 012737 000122 0J0636  
023044 004737 023614  
023050 000411  
023052 005737 000734

```
*****  
: 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  
TTOUTT ;PRINT DRIVE HEADER  
MOV DVN,R3  
OCTPP ;PRINT DRIVE NUMBER  
MOV #MSG11,R4  
TTOUTT ;PRINT UNIT HEADER  
MOV UDES,R3  
BIC #177770,R3  
OCTPP ;PRINT UNIT NUMBER  
MOV #MSG60,R4  
TTOUTT ;PRINT DENSITY TAG  
MOV UDES,R3  
SWAB R3  
BIC #177770,R3  
OCTPP ;PRINT DENSITY  
MOV #MSG61,R4  
TTOUTT ;PRINT PARITY TAG  
CLR R3  
BIT #10,UDES  
BEQ PAPRT0  
MOV #1,R3  
PAPRT0: OCTPP ;PRINT PARITY  
MOV #MSG62,R4  
TTOUTT ;PRINT FORMAT TAG  
MOV UDES,R3  
CLC  
ROR R3  
ROR R3  
ROR R3 ;PONTION FORMAT  
ROR R3  
BIC #177760,R3  
OCTPP ;PRINT FORMAT  
MOV #MSG8,R4  
TTOUTT ;PRINT PATRN TAG  
BIT #400,JSWR ;SEE IF RANDOM DATA  
BEQ PAPRT8 ;IF NOT: BR  
PAPRTA: MOV #122,TOB  
JSP PC,TOG ;PRINT R  
BR PAPRT0  
PAPRTB: TST ASEQF ;SEE IF AUTO SEQ
```





4742  
4743  
4744  
4745  
4746  
4747  
4748  
4749  
4750  
4751  
4752  
4753  
4754  
4755  
4756  
4757  
4758

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

```
023246 063737 000626 000624 RANG: ADD RANSV,RANBAS  
023254 063737 000624 000626 ADD RANBAS,RANSV ;GET NEW NUMBER  
023262 023701 000626 CMP RANSV,R1 ;SEE IF NUMBER TOO BIG  
023266 101367 BHI RANG ;IF SO: BR  
023270 020237 000626 CMP R2,RANSV ;SEE IF NUMBER TOO SMALL  
023274 101364 BHI RANG ;IF SO: BR  
023276 000207 RTS PC ;EXIT
```

```

4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776 023300 005037 000642 TTR: CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
4777 023304 005000 CLR RO
4778 023306 104010 TTR0: TTINN ;GO READ CHARACTER
4779 023310 122737 000015 000640 CMPB #15,TIB ;++G SEE IF CR
4780 023316 001004 BNE TTR1 ;IF NOT: BR
4781 023320 005737 000642 TST TEMP1 ;SEE IF FIRST CHARACTER
4782 023324 001436 BEQ TTR5 ;IF SO: BR
4783 023326 000426 BR TTR2 ;++G ELSE GO LOAD VALUE
4784 023330 122737 000060 000640 TTR1: CMPB #60,TIB ;++G SEE IF CHAR IS LESS THAN 0
4785 023336 101401 BLOS TTR1A ;IF NOT: BR
4786 023340 000431 BR TINER ;++G ELSE GO TO ERROR
4787 023342 122737 000070 000640 TTR1A: CMPB #70,TIB ;++G SEE IF CHAR IS GREATER THAN 7
4788 023350 101001 BHI TTR1B ;IF NOT: BR
4789 023352 000424 BR TINER ;++G ELSE GO TO ERROR
4790 023354 005237 000642 TTR1B: INC TEMP1 ;SET FIRST CHARACTER FLAG
4791 023360 006300 ASL RO
4792 023362 006300 ASL RO ;SHIFT 3 LEFT
4793 023364 006300 ASL RO
4794 023366 042737 177770 000640 BIC #177770,TIB ;STRIP ASCII
4795 023374 053700 000640 BIS TIB,RO ;LOAD CHARACTER
4796 023400 005301 DEC R1 ;SEE IF DONE
4797 023402 001341 BNE TTR0 ;IF NOT: BR
4798 023404 020002 TTR2: CMP RO,R2 ;SEE IF EXCEEDED MAXIMUM LIMIT
4799 023406 101401 BLOS TTR3 ;IF NOT: BR
4800 023410 000405 BR TINER ;++G ELSE GO TO ERROR
4801 023412 020300 TTR3: CMP R3,RO ;SEE IF BELOW MINIMUM LIMIT
4802 023414 101401 BLOS TTR4 ;IF NOT: BR
4803 023416 000402 BR TINER ;++G ELSE GO TO ERROR
4804 023420 010015 TTR4: MOV RO,(R5) ;LOAD VALUE
4805 023422 000207 TTR5: RTS PC ;EXIT
4806 023424 012704 025730 TINER: MOV #MSG43,R4
4807 023430 104000 TTOUTT ;PRINT?
4808 023432 162716 000020 SUB #20,(SP) ;RESET SP TO START OF VALUE ROUTINE
4809 023436 000207 RTS PC ;RECO VALUE ENTRY

```

```

4810
4811 ;TTY READ SUBROUTINE*****
4812
4813 023440 005277 155144 TTIN: INC @TKS
4814 023444 105777 155140 TTINI: TSTB @TKS
4815 023450 100375 BPL TTINI
4816 023452 017737 155134 000640 MOV @TKB,TIB
4817 023460 042737 000200 000640 BIC @200,TIB ;++G STRIP PARITY BIT
4818 023466 105777 155122 TTIN2: TSTB @TPS
4819 023472 100375 BPL TTIN2
4820 023474 113777 000640 155114 MOVB T,@TPB
4821 023502 000207 RTS PC
4822
4823 ;TTY OUTPUT SUBROUTINE*****
4824
4825 023504 112437 000636 TTOUT: MOVB (R4)+,TOB
4826 023510 122737 000043 000636 CMPB #43,TOB
4827 023516 001444 BEQ TEX
4828 023520 122737 000045 000636 CMPB #45,TOB
4829 023526 001407 BEQ TCRLF
4830 023530 122737 000041 000636 CMPB #41,TOB
4831 023536 001435 BEQ TBELL ;DO BELL
4832 023540 004737 023614 JSR PC,TOG
4833 023544 000757 BR TTOUT
4834 023546 112737 000015 000636 TCRLF: MOVB #15,TOB
4835 023554 004737 023614 JSR PC,TOG
4836 023560 012703 000006 MOV #6,R3
4837 023564 005037 000636 TCRLFA: CLR TOB
4838 023570 004737 023614 JSR PC,TOG
4839 023574 005303 DEC R3
4840 023576 001372 BNE TCRLFA ;DO FILLERS
4841 023600 112737 000012 000636 MOVB #12,TOB
4842 023606 004737 023614 JSR PC,TOG
4843 023612 000734 BR TTOUT
4844
4845 023614 105777 154774 TOG: TSTB @TPS
4846 023620 100375 BPL TOG
4847 023622 113777 000636 154766 MOVB TOB,@TPB
4848 023630 000207 RTS PC
4849 023632 012703 000002 TEX: MOV #2,R3
4850 023636 012737 000007 000636 TBELL: MOV #7,TOB
4851 023644 004737 023614 TBELA: JSR PC,TOG
4852 023650 005303 DEC R3
4853 023652 001371 BNE TBELA
4854 023654 000713 BR TTOUT
4855
4856

```

```

;OCTAL OUTPUT SUBROUTINE*****
4857
4858
4859 023656 005037 024110 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
4860 023662 000403 BR OCTPE1
4861 023664 012737 000001 024110 OCTPE: MOV #1,OFL
4862 023672 010304 OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
4863 023674 001006 BNE OCTPO ;IF NOT ZERO: BR
4864 023676 005737 024110 TST OFL
4865 023702 001003 BNE OCTPO
4866 023704 004737 024070 JSR PC,OCTPG1 ;ELSE PRINT ZERO
4867 023710 000450 BR OCTP3 ;++G SPACE AND EXIT
4868 023712 032704 100000 OCTPO: BIT #100000,R4 ;SEE IF MSD = 1
4869 023716 001406 BEQ OCTP1 ;IF NOT: BR
4870 023720 012704 000001 MOV #1,R4
4871 023724 004737 024046 JSR PC,OCTPG ;PRINT 1
4872 023730 000137 023742 JMP OCTP2
4873 023734 005004 OCTP1: CLR R4
4874 023736 004737 024046 JSR PC,OCTPG ;PRINT 0
4875 023742 010304 OCTP2: MOV R3,R4
4876 023744 006004 ROR R4
4877 023746 006004 ROR R4
4878 023750 006004 ROR R4 ;POSITION DIGIT
4879 023752 006004 ROR R4
4880 023754 000304 SWAB R4
4881 023756 004737 024046 JSR PC,OCTPG ;PRINT DIGIT 2
4882 023762 010304 MOV R3,R4
4883 023764 006004 ROR R4
4884 023766 000304 SWAB R4
4885 023770 004737 024046 JSR PC,OCTPG ;PRINT DIGIT 3
4886 023774 010304 MOV R3,R4
4887 023776 006104 ROL R4
4888 024000 006104 ROL R4
4889 024002 000304 SWAB R4
4890 024004 004737 024046 JSR PC,OCTPG ;PRINT DIGIT 4
4891 024010 010304 MOV R3,R4
4892 024012 006004 ROR R4
4893 024014 006004 ROR R4
4894 024016 006004 ROR R4
4895 024020 004737 024046 JSR PC,OCTPG
4896 024024 010304 MOV R3,R4
4897 024026 004737 024046 JSR PC,OCTPG ;PRINT DIGIT 5
4898 024032 012737 000240 000636 OCTP3: MOV #240,T0B
4899 024040 004737 023614 JSR PC,T0B ;PRINT SPACE
4900 024044 000207 RTS PC ;EXIT
4901 024046 042704 177770 OCTPG: BIC #177770,R4
4902 024052 001004 BNE OCTPG0
4903 024054 005737 024110 TST OFL
4904 024060 001001 BNE OCTPG0
4905 024062 000207 RTS PC
4906 024064 005237 024110 OCTPG0: INC OFL
4907 024070 052704 000260 OCTPG1: BIS #260,R4
4908 024074 010437 000636 MOV R4,T0B
4909 024100 004737 023614 JSR PC,T0B
4910 024104 010304 MOV R3,R4
4911 024106 000207 RTS PC
4912 024110 000000 OFL: 0 ;FIRST CHAR FLAG

```

```

4913
4914
4915 ;DATA CHARACTER OUTPUT SUBROUTINE*****
4916 024112 005037 000636 DOUT: CLR TOB
4917 024116 012704 000010 MOV #10,R4 ;SET NUMBER TO PRINT
4918 024122 110337 000636 MOVB R3,TOB
4919 024126 105777 154462 DOUT1: TSTB #TPS
4920 024132 100375 BPL DOUT1
4921 024134 132737 000200 000636 BITB #200,TOB
4922 024142 105737 000636 TSTB TOB ;++G
4923 024146 100004 BPL DOUT2 ;++G
4924 024150 012777 000061 154440 MOV #061,#TPB
4925 024156 000403 BR DOUT3
4926 024160 012777 000060 154430 DOUT2: MOV #060,#TPB
4927 024166 006137 000636 DOUT3: ROL TOB
4928 024172 005304 DEC R4
4929 024174 001354 BNE DOUT1
4930 024176 000207 RTS PC
4931 024200 013703 000646 DOUTD: MOV TEMP3,R3
4932 024204 000303 SWAB R3
4933 024206 004737 024112 JSR PC,DOUT
4934 024212 013703 000646 MOV TEMP3,R3
4935 024216 004737 024112 JSR PC,DOUT
4936 024222 000207 RTS PC

```

```

4937 ;++G TU16/TE16 SERIAL NUMBER PRINT SUBROUTINE*****
4938
4939
4940 024224 010304 SNPT: MOV R3,R4
4941 024226 000304 SWAB R4
4942 024230 006004 ROR R4
4943 024232 006004 ROR R4
4944 024234 006004 ROR R4
4945 024236 006004 ROR R4
4946 024240 004737 024302 JSR PC,SNPG ;PRINT FIRST DIGIT
4947 024244 010304 MOV R3,R4
4948 024246 000304 SWAB R4
4949 024250 004737 024302 JSR PC,SNPG ;PRINT SECOND DIGIT
4950 024254 010304 MOV R3,R4
4951 024256 006004 ROR R4
4952 024260 006004 ROR R4
4953 024262 006004 ROR R4
4954 024264 006004 ROR R4
4955 024266 004737 024302 JSR PC,SNPG ;PRINT THIRD DIGIT
4956 024272 010304 MOV R3,R4
4957 024274 004737 024302 JSR PC,SNPG ;PRINT FOURTH DIGIT
4958 024300 000207 RTS PC ;EXIT
4959 024302 012737 000260 000636 SNPG: MOV #260,TOB ;SET NUMBER BASE
4960 024310 042704 177760 BIC #177760,R4 ;MASK NUMBER
4961 024314 050437 000636 BIS R4,TOB ;BUILD DIGIT
4962 024320 004737 023614 JSR PC,TOG ;GO TYPE
4963 024324 000207 RTS PC ;RETURN

```

```

4964
4965 ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR #3 TO ALLOW CHANGING
4966 ;OF LOC.176.
4967 ;CALL IS BY WAY OF CKSWRP

```

```

4969                                     :LOCATIONS USED:
4970 024326 000000 TEMPST: .WORD 0
4971 024330 000000 COUNT: .WORD 0
4972 024332 000000 RDSW: .WORD 0
4973 024334 022737 000176 000606 CKSWR: CMP #SWREG,SWR ;SOFTWARE SWITCH REG PRESENT
4974 024342 001123 BNE OUT ;NO GET OUT
4975 024344 105777 154240 TSTB #TKS ;YES WAIT FOR
4976 024350 100120 BPL OUT ;READY GET CHARACTER
4977 024352 017737 154234 000640 MOV #TKB,TIB ;AND STRIP OFF
4978 024350 042737 177600 000640 BIC #177600,TIB ;THE GARBAGE
4979 024366 022737 000007 000640 CMP #7,TIB ;IS IT A '<IG>'
4980 024374 001106 BNE OUT
4981 024376 012704 027403 CNTG: MOV #SCNTG,R4
4982 024402 104000 TTOUTT
4983 024404 012704 027407 CNTLU: MOV #SMSWR,R4
4984 024410 104000 TTOUTT
4985 024412 017703 154170 MOV #SWR,R3
4986 024416 004737 023664 JSR PC,OCPE
4987 024422 012704 027416 MOV #SNEW,R4
4988 024426 104000 TTOUTT
4989 024430 005037 024326 $READ: CLR TEMPST
4990 024434 012737 000007 024330 MOV #7,COUNT
4991 024442 104010 1$: TTINN ;GO READ A CHARACTER
4992 024444 122737 000025 000640 CMPB #25,TIB ;IS IT A 'U'?
4993 024452 001001 BNE 2$ ;BRANCH IF NOT
4994 024454 000753 3$: BR CNTLU ;START OVER
4995 024456 122737 000015 000640 2$: CMPB #15,TIB ;IS IT A '<CR>'?
4996 024464 001013 BNE 4$ ;BRANCH IF NOT
4997 024466 012737 000200 024332 MOV #200,RDSW
4998 024474 012704 027426 MOV #MCRLF,R4
4999 024500 104000 TTOUTT
5000 024502 022737 000007 024330 CMP #7,COUNT ;WAS IT FIRST CHARACTER
5001 024510 001035 BNE 7$ ;CHANGE SWR IF NOT FIRST ONE
5002 024512 000437 BR OUT ;GET OUT
5003 024514 122737 000060 000640 4$: CMPB #60,TIB
5004 024522 003004 BGT 5$
5005 024524 122737 000067 000640 CMPB #67,TIB
5006 024532 002004 BGE 6$
5007 024534 012704 025730 5$: MOV #MSG43,R4
5008 024540 104000 TTOUTT
5009 024542 000744 BR 3$ ;START OVER IF NOT LEGAL CHARACTER
5010 024544 006337 024326 6$: ASL TEMPST
5011 024550 006337 024326 ASL TEMPST
5012 024554 006337 024326 ASL TEMPST
5013 024560 142737 000060 000640 BICB #60,TIB ;GET NITTY-GRITTY
5014 024566 153737 000640 024326 BISB TIB,TEMPST
5015 024574 005337 024330 DEC COUNT ;ONLY WANT 6 DIGITS
5016 024600 001755 BEQ 5$
5017 024602 000717 BR 1$
5018 024604 013777 024326 153774 7$: MOV TEMPST,#SWR ;CHANGE SWITCH REGISTER CONTENTS
5019 024612 000207 OLT: RTS PC ;RETURN TO BODY OF PROGRAM
5020                                     ;HALT HANDLER*****
5021
5022 024614 000000 STOP: HALT
5023 024616 024304 CKSWR ;CHECK FOR CONTROL G
5024 024622 000207 RTS PC

```

```

5025
5026
5027
5028
5029 024622 016677 000002 153754 TRAP30: MOV 2(6) @PSW ;ADJUST PSW
5030 024630 011666 000002 MOV @SP, 2(6) ;PLACE RETURN ADDRESS OVER PSW
5031 024634 162716 000002 SUB @2, @SP ;SUB. 2 FROM RETURN ADDRESS
5032 024640 013646 MOV @2(6)+, -(6)
5033 024642 062716 120650 ADD @TABLE-104000, @SP ;GET SUBROUTINE STARTING ADDRESS
5034 024646 013607 MOV @2(SP)+, PC ;GO TO SUBROUTINE
5035 024650 023504 TABLE: TTOUT
5036 024652 023656 OCTP
5037 024654 024334 CKSWR
5038 024656 024614 STOP
5039 024660 023440 TTIN
5040 104000 TTOUTT= 104000
5041 104002 OCTPP= 104002
5042 104004 CKSWRR= 104004
5043 104006 STOPP= 104006
5044 104010 TTINN= 104010

```

```

5046
5047 ;ERROR MESSAGES*****
5048
5049 024662 042052 020105 043 MSG1: .ASCII /*DE */
5050
5051 024667 045 035507 021440 MSG2: .ASCII /*G; */
5052
5053 024674 041045 020073 043 MSG3: .ASCII /*B; */
5054
5055 024701 045 047103 021440 MSG4: .ASCII /*CN */
5056
5057 024706 053452 020105 043 MSG5: .ASCII /*WE */
5058
5059 024713 052 042522 021440 MSG6: .ASCII /*RE */
5060
5061 024720 051052 020123 043 MSG7: .ASCII /*RS */
5062
5063 024725 052 040520 051124 MSG8: .ASCII /*PATRN */
5064 024732 020116 043
5065 024735 040 047123 020072 MSG9: .ASCII / SN: */
5066 024742 043
5067 024743 052 042523 021440 MSG10: .ASCII /*SE */
5068
5069 024750 051452 040514 042526 MSG11: .ASCII /*SLAVE NO. */
5070 024756 047040 027117 021440
5071
5072 024764 042045 044522 042526 MSG12: .ASCII /*DRIVE NO. */
5073 024772 047040 027117 021440
5074
5075 025000 025045 047102 021440 MSG13: .ASCII /**BN */
5076
5077 025006 051052 020116 043 MSG14: .ASCII /*RN */
5078
5079 025013 045 020041 020040 MSG15: .ASCII /*! BAD RECORD%*/
5080 025020 020040 020040 020040
5081 025026 041040 042101 051040
5082 025034 041505 051117 022504
5083 025042 021445
5084
5085 025044 043040 043 MSG16: .ASCII / F*/
5086
5087 025047 040 021522 MSG17: .ASCII / R*/
5088
5089 025052 020041 047505 020124 MSG20: .ASCII /*! EOT NO: */
5090 025060 047516 020072 043
5091
5092
5093 025065 045 047111 042524 MSG21: .ASCII /*INTERCHANGE READ = */
5094 025072 041522 040510 043516
5095 025100 020105 042522 042101
5096 025106 036440 021440
5097
5098 025112 020445 046111 042514 MSG22: .ASCII /*!ILLEGAL BOT: HALT%*/
5099 025120 040507 020114 047502
5100 025126 035124 044040 046101
5101 025134 022524 043

```



5102									
5103	025137	045	051503	020061	MSG23:	.ASCII	/%CSI #/		
5104	025144	043							
5105									
5106	025145	045	041527	021440	MSG23A:	.ASCII	/%WC #/		
5107									
5108	025152	041045	020101	043	MSG23B:	.ASCII	/%BA #/		
5109									
5110	025157	045	041506	021440	MSG23C:	.ASCII	/%FC #/		
5111									
5112	025164	041445	031123	021440	MSG23D:	.ASCII	/%CS2 #/		
5113									
5114	025172	042045	020123	043	MSG23E:	.ASCII	/%DS #/		
5115									
5116	025177	045	051105	021440	MSG23F:	.ASCII	/%ER #/		
5117									
5118	025204	040445	020123	043	MSG23G:	.ASCII	/%AS #/		
5119									
5120	025211	045	045503	021440	MSG23H:	.ASCII	/%CK #/		
5121									
5122	025216	042045	020102	043	MSG23I:	.ASCII	/%DB #/		
5123									
5124	025223	045	051115	021440	MSG23J:	.ASCII	/%MR #/		
5125									
5126	025230	042045	020124	043	MSG23K:	.ASCII	/%DT #/		
5127									
5128	025235	045	041524	021440	MSG23L:	.ASCII	/%TC #/		
5129									
5130	025242	051445	020116	043	MSG23M:	.ASCII	/%SN #/		
5131									
5132	025247	045	047041	020117	MSG24:	.ASCII	/%!NO INTERRUPT%#/		
5133	025254	047111	042524	051122					
5134	025262	050125	022524	043					
5135									
5136	025267	045	047041	020117	MSG25:	.ASCII	/%!NO MOL: HALT%#/		
5137	025274	047515	035114	044040					
5138	025302	046101	022524	043					
5139									
5140	025307	045	051104	050117	MSG26:	.ASCII	/%DROPS: #/		
5141	025314	035123	021440						
5142									
5143	025320	050045	041511	051513	MSG27:	.ASCII	/%PICKS: #/		
5144	025326	020072	043						
5145									
5146	025331	045	043		MSG28:	.ASCII	/%#/		
5147	025333	045	052045	030115	MSG30:	.ASCII	'%%TMO2-TU16/TE16 AUTO SEQUENCE (DZTUA-G)%#'	;++G	
5148	025340	026462	052524	033061					
5149	025346	052057	030505	020066					
5150	025354	052501	047524	051440					
5151	025362	050505	042526	041516					
5152	025370	020105	042050	052132					
5153	025376	040525	043455	022451					
5154	025401	043							
5155	025405	045	052045	030115	MSG31:	.ASCII	'%%TMO2-TU16/TE16 DATA RELIABILITY TEST (DZTUA-G)%#'	;++G	
5156	025411	026462	052524	033061					
5157	025417	052057	030505	020066					
5158	025423	052057	030505	020066					

5158	025426	040504	040524	051040		
5159	025434	046105	040511	044502		
5160	025442	044514	054524	052040		
5161	025450	051505	020124	042050		
5162	025456	052132	040525	043455		
5163	025464	022451	021445			
5164						
5165	025470	051445	040514	042526	MSG32:	.ASCII /%SLAVE NUMBER = #/
5166	025476	047040	046525	042502		
5167	025504	020122	020075	043		
5168						
5169	025511	045	042504	051516	MSG33:	.ASCII /%DENSITY = #/
5170	025516	052111	020131	020075		
5171	025524	043				
5172						
5173	025525	045	040520	044522	MSG34:	.ASCII /%PARITY = #/
5174	025532	054524	036440	021440		
5175						
5176	025540	051045	041505	051117	MSG35:	.ASCII /%RECORD COUNT = #/
5177	025546	020104	047503	047125		
5178	025554	020124	020075	043		
5179						
5180	025561	045	044103	051101	MSG36:	.ASCII /%CHARACTER COUNT = #/
5181	025566	041501	042524	020122		
5182	025574	047503	047125	020124		
5183	025602	020075	043			
5184						
5185	025605	045	040520	052124	MSG37:	.ASCII /%PATTERN NUMBER = #/
5186	025612	051105	020116	052516		
5187	025620	041115	051105	036440		
5188	025626	021440				
5189	025630	051445	047111	046107	MSG38:	.ASCII /%SINGLE PASS = #/
5190	025636	020105	040520	051523		
5191	025644	036440	021440			
5192	025650	042445	052116	051105	MSG40:	.ASCII /%ENTER STALLS%READ = #/
5193	025656	051440	040524	046114		
5194	025664	022523	042522	042101		
5195	025672	036440	021440			
5196						
5197	025676	053445	044522	042524	MSG41:	.ASCII /%WRITE = #/
5198	025704	036440	021440			
5199						
5200	025710	052045	051125	020116	MSG42:	.ASCII /%TURN AROUND = #/
5201	025716	051101	052517	042116		
5202	025724	036440	021440			
5203						
5204	025730	037445	021445		MSG43:	.ASCII /%?%#/
5205						
5206	025734	042445	052116	051105	MSG44:	.ASCII /%ENTER YOZZLE STALL = #/
5207	025742	054440	055117	046132		
5208	025750	020105	052123	046101		
5209	025756	020114	020075	043		
5210						
5211	025763	045	051105	020122	MSG45:	.ASCII /%ERR AMT #/
5212	025770	046501	020124	043		
5213						

K09

SEQ 0114

5214	025775	045	041506	021440	MSG46:	.ASCII	/%FC #/
5215							
5216	026002	041445	020101	043	MSG47:	.ASCII	/%CA #/
5217							
5218	026007	045	047041	020117	MSG48:	.ASCII	/%!NO BOT ON REWIND: HALT%#/
5219	026014	047502	020124	047117			
5220	026022	051040	053505	047111			
5221	026030	035104	044040	046101			
5222	026036	022524	043				
5223							
5224	026041	040	047516	020124	MSG49:	.ASCII	/ NOT AVAIL #/
5225	026046	053101	044501	020114			
5226	026054	043					
5227	026055	040	046111	042514	MSG50:	.ASCII	/ ILLEGAL DRIVE TYPE #/
5228	026062	040507	020114	051104			
5229	026070	053111	020105	054524			
5230	026076	042520	021440				
5231	026102	022445	051104	053111	MSG52:	.ASCII	/%DRIVE NUMBER = #/
5232	026110	020105	052516	041115			
5233	026116	051105	036440	021440			
5234							
5235	026124	043045	051117	040515	MSG53:	.ASCII	/%FORMAT = #/
5236	026132	020124	020075	043			
5237							
5238	026137	052	042527	052040	MSG54:	.ASCII	/%WE TM#/
5239	026144	021515					
5240							
5241	026146	051452	020105	046524	MSG55:	.ASCII	/%SE TM#/
5242	026154	043					
5243							
5244	026155	040	046524	043	MSG56:	.ASCII	/ TM#/
5245							
5246	026161	040	047516	026516	MSG57:	.ASCII	/ NON-EXIST SLAVE#/
5247	026166	054105	051511	020124			
5248	026174	046123	053101	021505			
5249	026202	041445	041522	021440	MSG58:	.ASCII	/%CRC #/
5250	026210	046045	041522	021440	MSG59:	.ASCII	/%LRC #/
5251	026216	042052	021440		MSG60:	.ASCII	/%D #/
5252	026222	050052	021440		MSG61:	.ASCII	/%P #/
5253	026226	043052	021440		MSG62:	.ASCII	/%F #/
5254							
5255	026232	025045	051117	043511	MSG64:	.ASCII	/%*ORIGINAL ERROR*#/
5256	026240	047111	046101	042440			
5257	026246	051122	051117	021452			
5258							
5259	026254	051045	052105	054522	MSG65:	.ASCII	/%RETRY: #/
5260	026262	020072	043				
5261							
5262	026265	052	051441	020105	MSG66:	.ASCII	/%!SE RTRY #/
5263	026272	052122	054522	021440			
5264							
5265	026300	020452	051105	051501	MSG67:	.ASCII	/%!ERASE#/
5266	026306	021505					
5267							
5268	026310	051045	051105	053105	MSG68:	.ASCII	/%REREV: #/
5269	026316	020072	043				

5270	026321	045	040524	042520	MSG69:	.ASCII	/%TAPE MARK = #/
5271	026326	046440	051101	020113			
5272	026334	020075	043				
5273							
5274	026337	045	047041	020117	MSG70:	.ASCII	/%!NO DRY FROM REWIND: HALT%#/
5275	026344	051104	020131	051106			
5276	026352	046517	051040	053505			
5277	026360	047111	035104	044040			
5278	026366	046101	022524	043			
5279	026373	040	047516	026516	MSG71:	.ASCII	/ NON-EXIST DRIVE#/
5280	026400	054105	051511	020124			
5281	026406	051104	053111	021505			
5282	026414	051045	043105	042127	MSG72:	.ASCII	/%REFWD: #/
5283	026422	020072	043				
5284	026425	045	052127	051105	MSG73:	.ASCII	/%WTEPR: #/
5285	026432	035122	021440				
5286	026436	051045	043505	051511	MSG74:	.ASCII	/%REGISTER START = #/
5287	026444	042524	020122	052123			
5288	026452	051101	020124	020075			
5289	026460	043					
5290	026461	045	042526	052103	MSG75:	.ASCII	/%VECTOR = #/
5291	026466	051117	036440	021440			
5292	026474	042045	051105	053105	MSG76:	.ASCII	/%DEREV: #/
5293	026502	020072	043				
5294	026505	045	042504	053506	MSG77:	.ASCII	/%DEFWD: #/
5295	026512	035104	021440				
5296	026516	020445	047516	026516	MSG78:	.ASCII	/%!NON-RETRYABLE WRITE ERROR: ER #/
5297	026524	042522	051124	040531			
5298	026532	046102	020105	051127			
5299	026540	052111	020105	051105			
5300	026546	047522	035122	042440			
5301	026554	020122	043				
5302	026557	045	047041	047117	MSG79:	.ASCII	/%!NON-RETRYABLE READ ERROR: ER #/
5303	026564	051055	052105	054522			
5304	026572	041101	042514	051040			
5305	026600	040505	020104	051105			
5306	026606	047522	035122	042440			
5307	026614	020122	043				
5308	026617	045	020441	047105	MSG100:	.ASCII	/%!!END OF PASS %#/
5309	026624	020104	043117	050040			
5310	026632	051501	020123	021445			
5311	026640	025045	025052	025052	MSG101:	.ASCII	/%*****%#/
5312	026646	025052	025052	021445			
5313	026654	052052	030115	020062	MSG102:	.ASCII	/*TMO2 #/
5314	026662	043					
5315	026663	052	046123	053101	MSG103:	.ASCII	/*SLAVES #/
5316	026670	051505	021440				
5317	026674	040445	052125	020117	MSG104:	.ASCII	/%AUTO CONT: #/
5318	026702	047503	052116	020072			
5319	026710	043					
5320	026711	045	042522	047503	MSG105:	.ASCII	/%RECOVERED#/
5321	026716	042526	042522	021504			
5322	026724	020452	041041	042101	MSG106:	.ASCII	/%!!BAD TAPE OVERFLOW#/
5323	026732	052040	050101	020105			
5324	026740	053117	051105	046106			
5325	026746	053517	043				

5326	026751	045	042522	044527	MSG16A: .ASCII	/%REWIND TAPE; RESTART AT BLOCK ONE#/ .
5327	026756	042116	052040	050101		
5328	026764	035505	051040	051505		
5329	026772	040524	052122	040440		
5330	027000	020124	046102	041517		
5331	027006	020113	047117	021505		
5332	027014	020445	052441	051116	MSG107: .ASCII	/%!!UNRECOVERABLE BAD SPOT/ .
5333	027022	041505	053117	051105		
5334	027030	041101	042514	041040		
5335	027036	042101	051440	047520		
5336	027044	124				
5337	027045	045	040502	020104	.ASCII	/%BAD RECORD LEFT ON TAPE%#/ .
5338	027052	042522	047503	042122		
5339	027060	046040	043105	020124		
5340	027066	047117	052040	050101		
5341	027074	022505	043			
5342	027077	045	051116	020132	MSG108: .ASCII	/%NRZ ONLY: #/ .
5343	027104	047117	054514	020072		
5344	027112	043				
5345	027113	052	020441	047520	MSG109: .ASCII	/%!!POSITION LOST IN RETRY#/ .
5346	027120	044523	044524	047117		
5347	027126	046040	051517	020124		
5348	027134	047111	051040	052105		
5349	027142	054522	043			
5350	027145	045	052523	050123	MSG110: .ASCII	/%SUSPECT BAD TAPE#/ .
5351	027152	041505	020124	040502		
5352	027160	020104	040524	042520		
5353	027166	043				
5354	027167	045	042522	042520	MSG111: .ASCII	/%REPEAT: #/ .
5355	027174	052101	020072	043		
5356	027201	040	040502	020104	MSG112: .ASCII	/ BAD TAPE SPOTS%#/ .
5357	027206	040524	042520	051440		
5358	027214	047520	051524	021445		
5359						
5360	027222	020045	047523	052106	MSG113: .ASCII	/% SOFT: #/ .
5361	027230	020072	043			
5362						
5363	027233	045	044040	051101	MSG114: .ASCII	/% HARD: #/ .
5364	027240	035164	021440			
5365						
5366	027244	020445	044041	051101	MSG115: .ASCII	/%!!HARD READ ERROR#/ .
5367	027252	020104	042522	042101		
5368	027260	042440	051122	051117		
5369	027266	043				
5370	027267	045	052441	044516	MSG116: .ASCII	/%!!UNIT REWINDING: TEST WILL START AT BOT#/ .
5371	027274	020124	042522	044527		
5372	027302	042116	047111	035107		
5373	027310	052040	051505	020124		
5374	027316	044527	046114	051440		
5375	027324	040524	052122	040440		
5376	027332	020124	047502	021524		
5377	027340	051045	046505	053117	MSG120: .ASCII	/%REMOVE TMOP FROM UNIT UNDER TEST%#/ .
5378	027346	020105	046524	050104		
5379	027354	043040	047522	020115		
5380	027362	047125	052111	052440		
5381	027370	042116	051105	052040		

```

5382 027376 051505 022524 043
5383 027403 045 043536 043 $CNTG: .ASCII /%TG#/
5384 027407 045 053523 036522 $MSWR: .ASCII /%SWR= #/
5385 027414 021440
5386 027416 020040 042516 036527 $MNEW: .ASCII / NEW= #/
5387 027424 021440
5388 027426 021445 MCRLF: .ASCII /%#/
5389
5390
5391 027430 000000 WDATA: 0 ;WRITE BUFFER
5392
5393 033436 033436
5394 033436 000000 RDATA: 0 ;READ BUFFER
5395
5396 000001 .END

```

ABL CNT	000740	B*OVO	007152	DATA14	003022	DERR08	016224	DRP1	001010
ACTLRC	021122	BTOV1	007162	DATA15	003024	DERR0C	016250	DRP2	001012
AORVN	000736	BTOV2	007250	DATA2	002776	DEKRO0	016252	DRP3	001014
AM001	022312	BTOV3	007266	DATA3	003000	DERR1	016300	DRP4	001016
AM001A	022322	BTPRT	007276	DATA4	003002	DERR2	016302	DRP5	001020
AM001B	022336	BTPRT1	007346	DATA5	003004	DERR3	016316	DRP6	001022
AM001C	022472	BTPT	000730	DATA6	003006	DERR4	016320	DRP7	001024
AM002	022506	BTSTF	000726	DATA7	003010	DERR4A	016460	DRP8	001026
AM002A	022532	BTUR	007352	DATBL	002770	DERR4B	016522	DRVER	021112
AM002B	022552	BT00	001610	DATER1	001130	DERR5	016556	DS	000522
AM002X	022660	BT01	001714	DATR	015054	DERR6	016570	DSUP	014152
APATS	015034	BT02	002020	DATRO	015072	DFX	016130	DSO	014160
AS	000526	BT03	002124	DATO	014364	DF0	016022	DS0A	014242
ASE0	021704	BT04	002230	DAT0A	014416	DF0A	015716	DS0B	014240
ASE0CF	000742	BT05	002334	DAT0B	014424	DF0A0	015740	DS0C	014202
ASE0F	000734	BT06	002440	DAT0C	014470	DF0A1	015754	FS1	014270
ASE0X	022074	BT07	002544	DAT0D	014476	DF0A2	015770	FS2	014310
ASE00	021770	B0	012164	DAT0E	014506	DF0A3	016004	DS2A	014320
ASE01	021774	B1	012232	DAT0F	014520	DF0A4	016010	DS3	014324
ASE02	022034	B2	012242	DAT1	014530	DF0B	015656	DS4	014334
ASE03	022046	CADER	021104	DAT1A	014534	DF0B0	015700	DT	000536
ASE04	022056	CC	000530	DAT1B	014540	DF0C	015636	DVN	000550
BA	000514	CCNTR	012266	DAT10	014652	DF0C0	015646	DOFL	014526
BAER	021106	CHNFLG	003040	DAT10A	014664	DF0D	015622	EMADOR	000652
BBC	000656	CKSWR	024334	DAT11	014702	DF0E	015614	EOPB1	000744
BCNT	000710	CKSWRR=	104004	DAT11A	014710	DF0F	015606	EOTCO	002650
BDPP	000716	CLLAST	015224	DAT12	014722	DF1	016034	EOTREC	000660
BD00	001410	CLP	015334	DAT12A	014732	DF2	016044	ER	000524
BD10	001430	CLPE	015360	DAT13	014744	DF3	016062	ERCHK	017360
BD20	001450	CLP0	015370	DAT14	014754	DF4	016124	ERPT	020206
BD30	001470	CLP1	015402	DAT14A	014766	DOUT	024112	ERPTG	020242
BD40	001510	CLP2	015442	DAT15	015004	DOUT0	024200	ERPTG1	020310
BD50	001530	CLP3	015454	DAT15A	015010	DOUT1	024126	ERPTT	020226
BD60	001550	CLD	015142	DAT15B	015020	DOUT2	024160	ERPTO	020324
BD70	001570	CL1	015170	DAT2	014550	DOUT3	024166	ERPT1	020372
BKRT	012206	CL2	015212	DAT3	014554	DPC	016750	ERPT2	020430
BKSP	012030	CL3	015276	DAT3A	014562	DPCG	016756	ERPT3	020466
BKTM	012112	CNTG	024376	DAT3B	014566	DPC0	016764	ERPT4	020530
BKTHO	012154	CNTLU	024404	DAT4	014600	DPC0A	017026	ERPT5	020552
BLCNTR	000654	CONER	021110	DAT5	014610	DPC1	017034	ERPT5A	020634
BPKP	000720	COUNT	024330	DAT6	014616	DPC1A	017062	ERPT6	020676
BP00	001210	CRCER	021120	DAT7	014624	DPC2	017124	ERPT7	020750
BP10	001230	CRCLRC	015124	DAT7A	014640	DPC2A	017066	ERPX	021006
BP20	001250	CRCSV	021126	DB	000532	DPC2B	017110	ERPX0	020752
BP30	001270	CS	000520	DCHK	015516	DPC3	017154	ERPX1	021050
BP40	001310	C1	000510	DCHKO	015546	DPPRT	017222	ERPX2	021102
BP50	001330	C2	000542	DEREV1	001170	DPPRTX	017356	ERSAV	000722
BP60	001350	DATA0	002772	DEREX	016574	DPPRT0	017272	ERTFL	000732
BP70	001370	DAT01	002774	DEREX1	016626	DPPRT1	017316	ERO	017374
B*ROOF	001030	DAT010	003012	DERFL	000704	DPPRT2	017332	EROA	017444
ETFLG	000724	DAT011	003014	DERR	016132	DROP	016740	ERCB	017412
ET01	007132	DAT012	003016	DERR0	016142	DRPK	016726	ERI	017452
ET011	007274	DAT013	003020	DERR0A	016172	DRPKF	016640	ER10	020152

ER2	017456	MSG110	027145	MSG44	025734	OCTP1	023734	RORTG	010746
ER2A	017522	MSG111	027167	MSG45	025763	OCTP2	023742	RORTX	011230
ER2A0	017472	MSG112	027201	MSG46	025775	OCTP3	024032	RORTY	010660
ER2A1	017512	MSG113	027222	MSG47	026002	OFL	024110	RORTO	010672
ER2B	017536	MSG114	027233	MSG48	026007	OUT	024612	RORTI	010724
ER2C	017562	MSG115	027244	MSG49	026041	PAPRT	022662	RORTIA	010722
ER2D	017576	MSG116	027267	MSG5	024706	PAPRTA	023036	RORTIB	010742
ER2E	017624	MSG12	024764	MSG50	026055	PAPRTB	023052	RORT2	011024
ER3	017632	MSG120	027340	MSG52	026102	PAPRTC	023066	RORT3	011046
ER3A	017674	MSG13	025000	MSG53	026124	PAPRTD	023074	RORT4	011052
ER3A1	017732	MSG14	025006	MSG54	026137	PAPRTY	023170	RORT5	011054
ER3B	017740	MSG15	025013	MSG55	026146	PAPRTD	022764	RORT5A	011120
ER4	017744	MSG16	025044	MSG56	026155	PAPRT1	023140	RORT5B	011130
ER4A	020010	MSG16A	026751	MSG57	026151	PAPRT2	023172	RORT6	011160
ER4A1	020002	MSG17	025047	MSG58	026202	PAPRT3	023174	RORT7	011224
ER6	020014	MSG2	024667	MSG59	026210	PARCNT	015514	ROSM	024332
ER6A	020100	MSG20	025052	MSG6	024713	PARS	014362	ROX	010656
ER7	020132	MSG21	025065	MSG60	026216	PATRN	000560	ROO	010002
EXCRC	015510	MSG22	025112	MSG61	026222	PATS	014360	RO1	010050
EXLRC	015512	MSG23	025137	MSG62	026226	PFLG	000670	RO1A	010064
FC	000516	MSG23A	025145	MSG64	026232	PICK	017156	RO1B	010072
FCER	021114	MSG23B	025152	MSG65	026254	PIK1	000770	RO1D	010100
FCSAV	000632	MSG23C	025157	MSG66	026265	PIK2	000772	RO1E	010570
FCNT	000556	MSG23D	025164	MSG67	026300	PIK3	000774	RO1F	010606
FREX	021166	MSG23E	025172	MSG68	026310	PIK4	000776	RO2	010104
FRPRT	021130	MSG23F	025177	MSG69	026321	PIK5	001000	RO2A	010124
FRO	021160	MSG23G	025204	MSG7	024720	PIK6	001002	RO2B	010134
HORFL	000664	MSG23H	025211	MSG70	026337	PIK7	001004	RO3	010164
HERE	005102	MSG23I	025216	MSG71	026373	PIK8	001006	RO4	010214
HROS	022106	MSG23J	025223	MSG72	026414	PRB	000622	RO4A	010312
HROS0	022146	MSG23K	025230	MSG73	026425	PRS	000620	RO4A0	010344
HROS1	022154	MSG23L	025235	MSG74	026436	PSW	000604	RO4A1	010366
HROS2	022222	MSG23M	025242	MSG75	026461	RANBAS	000624	RO4A2	010410
HROS3	022244	MSG24	025247	MSG76	026474	RANG	023246	RO4B	010416
HROS4	022256	MSG25	025267	MSG77	026505	RANSAV	000626	RO4C	010422
INTRF	000566	MSG26	025307	MSG78	026516	RANSET	004350	RO4D	010452
LRCER	021116	MSG27	025320	MSG79	026557	RCNT	000554	RO4E	010456
LRCV	021124	MSG28	025331	MSG8	024725	RCNTR	012326	RO5	010466
MCRLF	027426	MSG3	024674	MSG9	024735	RCSAV	000630	RO6	010510
MP	000534	MSG30	025333	MTC1	000672	RDA	007744	RO7	010534
MSG1	024662	MSG31	025405	MTINT	021666	ROATA	033436	RO7A	010564
MSG10	024743	MSG32	025470	MTINTA	021670	RDCMD	000562	READ	007704
MSG100	026617	MSG33	025511	NOP =	000240	RDERR1	001150	REGS	000544
MSG101	026640	MSG34	025525	NRTP	011232	RDERR	001110	REOT	004402
MSG102	026654	MSG35	025540	NRZOF	000650	RDERR2	001112	REOTC	005130
MSG103	026663	MSG36	025561	OCTP	023656	RDERR3	001114	REOTX	005030
MSG104	026674	MSG37	025605	OCTPE	023664	RDERR4	001116	REOTXX	005120
MSG105	026711	MSG38	025630	OCTPE1	023672	RDERR5	001120	REOT1A	004450
MSG106	026724	MSG4	024701	OCTPG	024046	RDERR6	001122	REOT1B	004476
MSG107	027014	MSG40	025650	OCTPG0	024064	RDERR7	001124	REOT1C	004516
MSG108	027077	MSG41	025676	OCTPG1	024070	RDERR8	001126	REOT1D	004522
MSG109	027113	MSG42	025710	OCTPP =	104002	RDEX	010652	RECT1E	004542
MSG11	024750	MSG43	025730	OCTPO	023712	ROFL	015122	RECT1F	004512



REOT2	004564	SNPT	024224	TCRLF	023546	TTR0	023306	WTER7	001104
REOT2A	004622	SPFLG	000570	TCRLF A	023564	TTR1	023330	WTER8	001106
REOT3	004650	STAL	000666	TEMPST	024326	TTR1A	023342	WTM	006024
REOT4	004700	STALL	012256	TEMP1	000642	TTR1B	023354	WTMO	006066
REOT5	004714	START	003026	TEMP2	000644	TTR2	023404	WTM1	006130
REOT6	004760	STARTA	003160	TEMP3	000646	TTR3	023412	WTM2	006140
REOT7	005006	STARTB	003164	TENO	005054	TTR4	023420	WTM3	006150
RETRY	000602	STARTC	003152	TEX	023630	TTR5	023422	WTM4	006230
RETURN	021662	STARTD	003250	TIB	000640	UDES	000552	WTM4A	006266
RFHARD	002730	STARTE	003242	TIMER	023424	UNP	000674	WO	005514
RF SOFT	002670	STARTF	003222	TINF	000634	UNX	000766	W1	005556
RPCNT	000700	STARTO	003174	TIMP	012352	UN1	000746	W2	005616
RRHARD	002750	START1	003432	TIMPA	012362	UN2	000750	W3	005632
RRSOFT	002710	START2	003550	TIMP8	012402	UN3	000752	W3A	005654
RSEQ	007366	START3	003564	TIMP80	012566	UN4	000754	W4	005740
RSEX	007526	START4	003600	TIMP81	012426	UN5	000756	W4A	005766
RSF	007442	START5	004270	TIMPC	012614	UN6	000760	W5	006002
RSFR	007542	START6	004306	TIMPX	014114	UN7	000762	W6	006012
RSFRX	007702	START7	004336	TIMPO	012720	UN8	000764	XOR	015462
RSFRO	007574	START8	004344	TIMPOA	012774	UPS	000714	XORS	015506
RSFRI	007606	STAR1A	003444	TIMPOB	013012	VECT	000546	YOZ	011246
RSFR2	007640	STAR1B	003464	TIMPOC	013052	WC	000512	YOZA	011314
RSFO	007500	STAR1C	003520	TIMPOD	013074	WDATA	027430	YOZB	011322
RSF1	007514	STAR4A	003660	TIMPOE	013120	WEX	006272	YOZC	011342
RSR	007420	STAR40	003612	TIMP1	013136	WRITE	005474	YOZCO	011432
RSTAL	000572	STAUT	003136	TIMP2	013214	WRITE	005510	YOZC1	011446
RTCNT	000702	STAUTO	003372	TIMP2A	013272	WRTS8	006736	YOZC2	011454
RTRN	000662	STOP	024614	TIMP2B	013350	WRTS80	007010	YOZD	011474
RTYFL	000712	STOPP =	104006	TIMP2C	013376	WRTS81	007024	YOZD0	011576
RTY1	001050	STP	004006	TIMP3	013416	WRTS82	007034	YOZD1	011622
RTY2	001052	STPX	004254	TIMP4	013746	WRTS83	007110	YOZD2	011570
RTY3	001054	SUSWR	003262	TKB	000612	WRTY	006334	YOZD3	011614
RTY4	001056	SWR	000606	TKS	000610	WRTYR	006366	YOZD4	011550
RTY5	001060	SWREG	000176	TMEX	000564	WRTYTM	006362	YOZE	011626
RTY6	001062	TABLE	024650	TMFLG	000676	WRTY0	006342	YOZF	011656
RTY7	001064	TAPG	021170	TOB	000636	WRTY1	006442	YOZFO	011662
RTY8	001066	TAPG0	021202	TOG	023614	WRTY2	006444	YOZG	011754
RUNC	005132	TAPG1	021244	TPB	000616	WRTY2A	006510	YOZH	012020
RUNDA	005144	TAPG2	021256	TP05	014124	WRTY2B	006526	YOZI	012004
RUNDX	005444	TAPG3	021266	TP051	014136	WRTY3	006572	YOZO	011266
RUNDO	005202	TAPG3A	021336	TPS	000614	WRTY3A	006610	YSTAL	000600
RUNO1	005254	TAPG3B	021352	TRAP30	024622	WRTY4	006676	SCNTG	027403
RUNO1A	005312	TAPG3C	021372	TSTAL	000576	WRTY5	006732	SENDAD	005072
RUNO2	005330	TAPG3D	021436	TTIN	023440	WRW	006310	SMNEW	027416
RUNO3	005334	TAPG3E	021462	TTINN =	104010	WRWX	006332	SMWR	027407
RUNO4	005364	TAPG3F	021320	TTINT	021602	WSTAL	000574	SREAD	024430
RUNO5	005404	TAPG4	021510	TTINTO	021652	WTER1	001070	SSVPC =	000034
RUNO6	005422	TAPG5	021524	TTIN1	023444	WTER2	001072	=	033440
SCYFL	014122	TAPG6	021566	TTIN2	023466	WTER3	001074		
SERFL	000706	TAPG7	021576	TTOUT	023504	WTER4	001076		
SN	000540	TBELA	023636	TTOUTT =	104000	WTER5	001100		
SNP	024302	TBELL	023632	TTR	023300	WTER6	001102		

E10

TMD2-TU16/TE16 DATA RELIABILITY PROGRAM MACY11 30(1046) 22-JUN-77 08:44 PAGE 122  
DZTUAG.P11 17-JUN-77 09:52 SYMBOL TABLE

SEG 0121

ERRORS DETECTED: 0

DZTUAG DZTUAG/SOL+DZTUAG  
RUN-TIME: 5 10 .8 SECONDS  
RUN-TIME RATIO: 180/16=10.8  
CORE USED: 6K (11 PAGES)

F10