

This microfiche card contains a grid of frames, each containing technical data. The data is organized into columns and rows, with some frames containing diagrams or tables. The text is small and difficult to read, but appears to be technical specifications or test results. The card is oriented vertically, with the grid of frames occupying most of the area.

.REM 2

IDENTIFICATION

PRODUCT CODE: AC-9447H-MC  
PRODUCT TITLE: CZTUANO TM02-TU16/TE16 RELIAB  
DATE CREATED: JUNE 1978  
MAINTAINER: DIAGNOSTIC GROUP  
AUTHOR: R.B. BARNES

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, 1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

TABLE OF CONTENTS

PARAGRAPH	SUBJECT	PAGE
1.	ABSTRACT	3
2.	REQUIREMENTS	3
3.	LOADING PROCEDURE	3
4.	STARTING PROCEDURE	4
4.1	AUTOMATIC MODE OPER.	10
5.	DATA PATTERNS	10
6.	RANDOMIZATION	11
7.	DYNAMIC PARAMETERS	12
8.	CONSOLE SWITCH	18
9.	ERROR PRINTOUTS	17
10.	STATISTICS PRINTOUT	26
11.	AUTO SEQUENCE	27
12.	TESTING PROCEDURES	29
13.	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(TMO2 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, HOW-  
EVER, AT LEAST ONE MUST BE USED.  
THE SLAVE NUMBERS AND THEIR RESPECTIVE  
DENSITY, PARITY AND FORMAT MAY BE ENTERED  
IN ANY ORDER. THE INFORMATION FOR  
EACH SLAVE ENTERED IS LOADED INTO A  
TABLE FOR REFERENCE IN TESTING.  
IF LESS THAN EIGHT(8) SLAVES ARE  
REQUIRED, THEN RESPONDING TO THE  
SLAVE NUMBER REQUEST WITH A CARRIAGE  
RETURN WILL TERMINATE THE SLAVE  
ENTRIES AND CONTINUE TO THE NEXT  
PARAMETER. IT SHOULD BE REMEMBERED  
THAT AT LEAST ONE SLAVE NUMBER REQUEST

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

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.

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 PATTERN ZERO (0) EXTERNAL PAPER TAPE NEED ONLY BE READ ONCE AT INITIAL START OF 200(8), AND NEED NOT BE READ AGAIN UNLESS OVERWRITTEN BY RANDOM DATA. BE SURE TO LOAD THE READER BEFORE PRESSING START.

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

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



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

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

READ: THE TIME DELAY BETWEEN EACH RECORD READ

WRITE: THE TIME DELAY BETWEEN EACH RECORD WRITTEN

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

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

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

SAMPLE START AT 200(8):

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

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

\*\*\*SWR=XXXXXX NEW= WILL BE TYPED FIRST IF THE SOFTWARE  
REGISTER IS SELECTED(REFER TO SECTION 8 FOR OPERATOR OPTIONS).

TU16 TAPE DRIVE TEST  
ENTER CONDITIONS IN OCTAL

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

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

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

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

#### 4.1 AUTOMATIC MODE OPERATION

-----

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

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

#### 5. DATA PATTERNS

-----

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

THE FOLLOWING IS A LIST OF THE DATA PATTERNS AVAILABLE:

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

6. RANDOMIZATION

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

8. CONSOLE SWITCH SETTINGS

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

CONTROL:

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

- 1) TYPE CONTROL G < G>; THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.
  - A) THIS PROGRAM WILL PROCESS THE < G> EITHER IN FLAG MODE OR INTERRUPT DEPENDING ON WHERE IN THE PROGRAM THE < G> IS EXCEPTED. THE PROGRAM WILL SERVICE THE INTERRUPT ONLY WHEN THE 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 E. CH SPACE/READ OF THE RECORD AND IT MAY BE VARIED BY TYPING CONTROL C ON THE TELETYPE DURING THE EXECUTION OF THE YOZZLE AND RESPONDING TO THE PRINTED REQUEST WITH A SIX (6) DIGIT VALUE. THE YOZZLE STALL IS PRESET TO A VALUE OF 3000 IN THE PROGRAM TO PREVENT EXCESSIVE TAPE WEAR, BUT MAY BE SET TO ANY VALUE THROUGH THE TELETYPE.
- SW6-8: THESE THREE (3) SWITCHES CONTROL THE RANDOMIZATION OF DATA AND BLOCK SIZE AND MAY BE SET AND RESET AT ANY TIME. THE ACTUAL CHANGE WILL TAKE PLACE BETWEEN BLOCK CYCLES.
- SW9: SWITCH NINE (9) WHEN SET WILL CAUSE ALL AVAILABLE TAPE UNITS TO BE REWOUND AT THE END OF THE CURRENT BLOCK CYCLE. TESTING WILL BE RESUMED AT A BLOCK COUNT OF ONE (1) WHEN ALL UNITS HAVE REACHED BOT.

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

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

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

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

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

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

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

9. ERROR PRINTOUTS

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

A. OPERATION ERRORS:

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

1. READ/WRITE STATUS ERRORS: THESE ARE DETECTED BY EITHER THE TMO2 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 MAINTENANCE 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 TMO2 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 TMO2 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 5: THIS EXAMPLE SHOWS A READ DATA ERROR WHICH OCCURRED, WITHOUT AN ACCOMPANING STATUS ERROR, WHICH RESULTED IN A BAD RECORD.

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

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

BAD RECORD

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

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

CZTU  
CZTU  
15  
15  
15  
15  
15  
15  
15  
15  
15  
15  
15  
15  
15  
15  
15  
15  
15  
15  
15  
15  
15  
15  
15  
16  
16  
16  
16  
16  
16  
16









11. AUTO SEQUENCE  
-----

THE AUTO SEQUENCE (START AT ADDRESS 240) WILL EXECUTE A  
PREDETERMINED TEST PLAN ON ALL AVAILABLE SLAVES ON EACH  
AVAILABLE TMO2. THE ONLY OPERATOR RESPONSE IS TO THE TYPED  
REQUESTS FOR THE RH ADDRESS, VECTOR, CONTINUOUS OR SINGLE  
CYCLE, AND NRZ ONLY. ALL SWITCHES REMAIN ACTIVE AND MAY BE  
USED NORMALLY; HOWEVER THE IDEA IS TO LEAVE ALL SWITCHES  
DOWN AND ALLOW FULL EXECUTION OF THE TEST PLAN FOR  
SYSTEM CHECKOUT.

SAMPLE START AT 240(8): AUTO SEQUENCE.

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

TU16 AUTO SEQUENCE TEST  
ENTER CONDITIONS IN OCTAL

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

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

AS EACH TMO2 AND ITS SLAVES ARE FOUND, A DIVIDER LINE OF  
ASTERICKS WILL BE PRINTED FOLLOWED BY A PRINTOUT OF THE  
TMO2 AND ITS SLAVES BEING TESTED. AS EACH TMO2 AND  
ITS SLAVES ARE FINISHED, ANOTHER DIVIDER IS PRINTED  
BEFORE TESTING IS RESUMED ON THE NEXT AVAILABLE DRIVE.

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

AUTO SEQUENCE TEST PLAN:

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

1. NRZ CYCLE:

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

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

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

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

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

12. TESTING PROCEDURES  
-----

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

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

TRY IT, YOU'LL LIKE IT.

13. LISTING  
-----

x

```
1347 .LIST BIN,LOC,SEQ
1348 .TITLE CZTUAWO TMO2-TU16/TE16 RELIAB
1349 :ZZ - CZTUAWO
1350 :21 APRIL 76
1351 :R. BARNES
1352
1353 :REVISED (++) J.G.ADAMS MAY 1977
1354 :++G
1355 :++G
1356 :++G
1357 :++G
1358 :++G
1359 :++G
1360 :++G
1361
1362 .MCALL .SACT11,.$EOP,$CHAIN ;++G ACT11 HOOKS
1363 .NLIST MC ;++G DO NOT LIST MACRO CALLS
1364 .LIST ME ;++G LIST MACRO EXPANSIONS
1365 .ENABLE ABS,AMA ;++G ENABLE ABS AND MODE '37'
1366
1367
1368 :CONSOLE SWITCHES*****
1369
1370 :SW15: 1=STOP ON ERROR
1371 : 0=CONTINUE ON ERROR
1372
1373 :SW14: 1=PRINT READ/WRITE STATS
1374 : 0=DO NOT PRINT STATS
1375
1376 :SW13: 1=DO NOT CHECK DATA
1377 : 0=CHECK DATA
1378 :SW12: 1=DO NOT CHECK WRITE ERRORS
1379 : 0=CHECK WRITE ERRORS
1380 :SW11: 1=DO NOT CHECK READ ERRORS
1381 : 0=CHECK READ ERRORS
1382 :SW10: 1=DO NOT PRINT ERRORS
1383 : 0=PRINT ERRORS
1384
1385 :SW9: 1=REWIND TAPE
1386 : 0=DO NOT REWIND
1387
1388 :SW8: 1=USE RANDOM DATA
1389 : 0=USE FIXED DATA PATTERN
1390 :SW7: 1=USE RANDOM CHARACTER COUNT
1391 : 0=USE FIXED CHAR COUNT
1392 :SW6: 1=USE RANDOM RECORD COUNT
1393 : 0=USE FIXED RECORD COUNT
1394
1395 :SW5: 1=YOZZLE ON CURRENT RECORD
1396 : 0=DO NOT YOZZLE
1397
1398 :SW4: 1=DO BOTH READ AND WRITE RETRIES
1399 : 0=INHIBIT RETRIES
1400
1401 :SW3: 1=DO NOT READ FORWARD
```

1402  
1403  
1404  
1405  
1406  
1407  
1408

: 0=READ FORWARD  
:SW2: 1=DO NOT READ REVERSE  
: 0=READ REVERSE  
:SW1: 1=READ FORWARD FIRST  
: 0=READ REVERSE FIRST  
:SW0: 1=DO NOT WRITE  
: 0=WRITE

CZ  
CZ





```
1455 ;REGISTER EQUIVS*****
1456
1457 000000 R0=%0
1458 000001 R1=%1
1459 000002 R2=%2
1460 000003 R3=%3
1461 000004 R4=%4
1462 000005 R5=%5
1463 000006 SP=%6
1464 000007 PC=%7
1465 000240 NOP=240
1466
1467 ;TRAP CATCHERS*****
1468 000030 .=30
1469 000030 024670 TRAP30
1470 000032 000032 .=32
1471 000032 000340 340
1472
1473
1474 ;ACT11 HOOK *****
1475 000034 $SVPC=. ;SAVE CURRENT LOCATION CTR
1476 000046 .=46
1477 000046 005116 .WORD SENDAD ;SET LOCATION 46
1478 000052 000052 .=52
1479 000052 000000 .WORD 0 ;SET LOCATION 52 = 0
1480 000034 .=$SVPC ;RESTORE LOCATION CTR
1481
1482
1483 ;TTY INTERRUPT VECTOR*****
1484 000060 .=60
1485 000060 021630 TTINT ;TTY INTERRUPT HANDLER ADDRESS
1486 000062 000000 0
1487
1488
1489 ;SOFTWARE SWITCH REGISTER LOC. 176*****
1490
1491 000176 .=176
1492 000176 000000 SWREG: 0 ;SOFTWARE SWITCH REGISTER
1493
1494 ;START ADDRESS*****
1495
1496 000200 .=200
1497 000200 000137 003026 JMP START ;ENTER PARAMETERS VIA TTY
1498
1499 000204 .=204
1500 000204 000137 003152 JMP STARTC ;USE FIXED PARAMETLRS; HOLD DATA
1501
1502 000210 .=210
1503 000210 005037 015150 CLR RDFL
1504 000214 000137 003160 JMP STARTA ;USE FIXED PARAMETERS: NEW DATA
1505
1506 ;MAG TAPE INTERRUPT VECTOR*****
1507
1508 000224 .=224
1509 000224 021714 MTINT ;MAG TAPE INTERRUPT HANDLER ADDRESS
1510 000226 000340 340
```





```
1571
1572
1573
1574 000634 000000
1575 000636 000000
1576 000640 000000
1577 000642 000000
1578 000644 000000
1579 000646 000000
1580 000650 000000
1581 000652 000000
1582 000654 000000
1583 000656 000000
1584 000660 000000
1585 000662 000000
1586 000664 000000
1587 000666 000000
1588 000670 000000
1589 000672 000000
1590 000674 000000
1591 000676 000000
1592 000700 000000
1593 000702 000000
1594 000704 000000
1595 000706 000000
1596 000710 000000
1597 000712 000000
1598 000714 000000
1599 000716 000000
1600 000720 000000
1601 000722 000000
1602 000724 000000
1603 000726 000000
1604 000730 000000
1605 000732 000000
1606 000734 000000
1607 000736 000000
1608 000740 000000
1609 000742 000000
1610 000744 000000
```

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

TINF: 0	; TTY ENTRY FLAG
TOB: 0	; TTY OUTPUT BUFFER
TIB: 0	; TTY INPUT BUFFER
TEMP1: 0	; TEMP STORAGE
TEMP2: 0	; TEMP STORAGE
TEMP3: 0	; TEMP STORAGE
NRZOF: 0	; NRZ ONLY FLAG
EMADDR: 0	; ERROR MSG ADDRESS STORAGE
BLCNTR: 0	; BLOCK COUNTER
BBC: 0	; BAD RECORD COUNTER
EOTREC: 0	; EOT FLAG
RTRN: 0	; INTERRUPT RETURN STORAGE
HDRFL: 0	; HEADER FLAG
STAL: 0	; DELAY STORAGE
PFLG: 0	; PRINT FLAG
MTC1: 0	; MAG TAPE CNT REGISTER BUFFER
UNP: 0	; UNIT TABLE POINTER
TMFLG: 0	; TAPE MARK FLAG
RPCNT: 0	; REPEAT COUNTER
RTCNT: 0	; RETRY COUNTER
DERFL: 0	; DATA ERROR FLAG
SERFL: 0	; STATUS ERROR FLAG
BCNT: 0	; BIT COUNTER
RTYFL: 0	; RETRY FLAG
UPS: 0	; UNIT POINTER SAVE
BDPP: 0	; BITS DROPPED POINTER
BPKP: 0	; BITS PICKED POINTER
ERSAV: 0	; ERROR SAVE LOC
BTFLG: 0	; BAD TAPE FLAG
BTSTF: 0	; STATISTIC PRINT FLAG
BTPT: 0	; BAD TAPE POINTER
ERTFL: 0	; ERASE FLAG
ASEQF: 0	; AUTO SEQ FLAG
ADRVN: 0	; UTO SEQ DRIVE NUMBER
ABLCNT: 0	; AUTO BLOCK COUNTER
ASEQCF: 0	; AUTO SEQ CONTINUOUS FLAG
EOPB1: 0	; EOP FLAG

1611  
1612 ;UNIT ORDER AND DESCRIPTION TABLE \*\*\*\*\*  
1613  
1614 000746 000000 UN1: 0 ;THIS TABLE IS LOADED  
1615 000750 000000 UN2: 0 ;WITH UNIT NUMBERS AND  
1616 000752 000000 UN3: 0 ;THEIR DESCRIPTIONS IN  
1617 000754 000000 UN4: 0 ;THE ORDER THAT THEY  
1618 000756 000000 UN5: 0 ;WILL BE TESTED  
1619 000760 000000 UN6: 0  
1620 000762 000000 UN7: 0  
1621 000764 000000 UN8: 0  
1622 000766 177777 UNX: -1

1623  
1624 ;UNIT DROPS AND PICKS POINTERS\*\*\*\*\*  
1625  
1626 000770 001210 PIK1: BP00  
1627 000772 001230 PIK2: BP10  
1628 000774 001250 PIK3: BP20  
1629 000776 001270 PIK4: BP30  
1630 001000 001310 PIK5: BP40  
1631 001002 001330 PIK6: BP50  
1632 001004 001350 PIK7: BP60  
1633 001006 001370 PIK8: BP70  
1634 001010 001410 DRP1: BD00  
1635 001012 001430 DRP2: BD10  
1636 001014 001450 DRP3: BD20  
1637 001016 001470 DRP4: BD30  
1638 001020 001510 DRP5: BD40  
1639 001022 001530 DRP6: BD50  
1640 001024 001550 DRP7: BD60  
1641 001026 001570 DRP8: BD70

1642  
1643 ;UNIT BAD TAPE POINTERS\*\*\*\*\*  
1644  
1645 001030 001610 BTADDR: BT00  
1646 001032 001714 BT01  
1647 001034 002020 BT02  
1648 001036 002124 BT03  
1649 001040 002230 BT04  
1650 001042 002334 BT05  
1651 001044 002440 BT06  
1652 001046 002544 BT07

1653  
1654 ;UNIT WRITE RETRY COUNTER\*\*\*\*\*  
1655  
1656 001050 000000 RTY1: 0  
1657 001052 000000 RTY2: 0  
1658 001054 000000 RTY3: 0  
1659 001056 000000 RTY4: 0  
1660 001060 000000 RTY5: 0  
1661 001062 000000 RTY6: 0  
1662 001064 000000 RTY7: 0  
1663 001066 000000 RTY8: 0

1664  
1665 ;UNIT WRITE ERRORS\*\*\*\*\*  
1666

22  
22  
22  
22  
22  
22  
22  
22  
22







```
1755
1756                ;UNIT BAD TAPE COUNTER:16 PER SLAVE*****
1757
1758 001610 000000    BT00: 0
1759                .=.+102
1760 001714 000000    BT01: 0
1761                .=.+102
1762 002020 000000    BT02: 0
1763                .=.+102
1764 002124 000000    BT03: 0
1765                .=.+102
1766 002230 000000    BT04: 0
1767                .=.+102
1768 002334 000000    BT05: 0
1769                .=.+102
1770 002440 000000    BT06: 0
1771                .=.+102
1772 002544 000000    BT07: 0
1773                .=.+102
1774
1775                ;UNIT END OF TAPE COUNTERS 1 PER SLAVE*****
1776
1777 002650 000000    EOTCO: 0
1778 002652 000000    0
1779 002654 000000    0
1780 002656 000000    0
1781 002660 000000    0
1782 002662 000000    0
1783 002664 000000    0
1784 002666 000000    0
1785
1786                ;UNIT READ FORWARD SOFT ERROR*****
1787
1788 002670 000000    RFSOFT: 0
1789 002672 000000    0
1790 002674 000000    0
1791 002676 000000    0
1792 002700 000000    0
1793 002702 000000    0
1794 002704 000000    0
1795 002706 000000    0
1796
1797                ;UNIT READ REVERSE SOFT ERROR*****
1798
1799 002710 000000    RRSOFT: 0
1800 002712 000000    0
1801 002714 000000    0
1802 002716 000000    0
1803 002720 000000    0
1804 002722 000000    0
1805 002724 000000    0
1806 002726 000000    0
1807
```

```
1808
1809 ;UNIT READ FORWARD HARD ERROR*****
1810
1811 002730 000000 RFHARD: 0
1812 002732 000000 0
1813 002734 000000 0
1814 002736 000000 0
1815 002740 000000 0
1816 002742 000000 0
1817 002744 000000 0
1818 002746 000000 0
1819
1820 ;UNIT READ REVERSE HARD ERROR*****
1821
1822 002750 000000 RRHARD: 0
1823 002752 000000 0
1824 002754 000000 0
1825 002756 000000 0
1826 002760 000000 0
1827 002762 000000 0
1828 002764 000000 0
1829 002766 000000 0
1830
1831 ;DATA PATTERN GENERATORS*****
1832
1833 002770 002770 DATBL: . ;ENTRY TABLE
1834 002772 014412 DATA0: DAT0 ;EXTERNAL INPUT FROM H/S READER(SEE MAINDEC-11-DZTUF)
1835 002774 014556 DATA1: DAT1 ;ALL ONES
1836 002776 014576 DATA2: DAT2 ;ALL ZEROS
1837 003000 014602 DATA3: DAT3 ;WALKING ONE
1838 003002 014626 DATA4: DAT4 ;WALKING ZERO
1839 003004 014636 DATA5: DAT5 ;ALTERNATING ONE/ZERO
1840 003006 014644 DATA6: DAT6 ;ALTERNATING ZERO/ONE
1841 003010 014652 DATA7: DAT7 ;ALTERNATING ONE/ZERO IN ALTERNATING CHARACTERS
1842 003012 014700 DATA10: DAT10 ;WALKING ONE/ALL ONE IN ALTERNATING CHARACTERS
1843 003014 014730 DATA11: DAT11 ;ALL BITS 0-377
1844 003016 014750 DATA12: DAT12 ;ALL BITS 377-0
1845 003020 014772 DATA13: DAT13 ;ALTERNATING CHARACTERS 0 AND 377
1846 003022 015002 DATA14: DAT14 ;WALKING ZERO/ALL ZERO IN ALTERNATING CHARACTERS
1847 003024 015032 DATA15: DAT15 ;AUTO SEQUENCE PATTERN 0,0,-1,-1,-1,0,0
1848
```

```
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
```

```

.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
      CMP    #SENDAD,@#42 ;:BRANCH IF LOADED VIA ACT11 CHAIN MODE
      BEQ    50$
      TST   @#42         ;:BRANCH IF IN DUMP MODE
      BEQ    52$
      BR    51$
50$:  MOV    #SWREG,SWR   ;:INVOKE SOFTWARE SWR
      MOV    #100000,@SWR ;:HALT ON ERROR
51$:  INC    CHNFLG      ;:SET CHNFLG = CHAIN MODE
      JMP    3$         ;:GO TO CHAIN ADDRESS
52$:  CMPB   #6,@#41     ;++G BRANCH IF NOT LOADED VIA TMDP
      BNE   STAUT
      MOV   #MSG120,R4
      TTOUTT           ;++G ADVISE USER TO REMOVE TMDP
      BR    STAUT      ;++G
3$:   INC    ASEQF      ;++G SET AUTO SEQUENCE FLAG
      JMP   SUSWR      ;CHECK AND SET UP HRD/SOFT SWITCH REG ++ C.W

:START 240 *****
STAUT: MOV    #1,TINF    ;SET TTY ENTRY FLAG
      CLR    RDFL       ;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,RO
      MOV    #37,R1
STARTO: CLR    (RO)+   ;CLEAR FLAGS AND COUNTERS
```

1905	003176	005301			DEC	R1	
1906	003200	001375			BNE	START0	
1907	003202	012706	000500		MOV	#500,SP	;SET STACK POINTER
1908	003206	004737	004372		JSR	PC,RANSET	;GO RESET RANDOM BASE
1909	003212	012700	001050		MOV	#RTY1,R0	
1910	003216	012701	000750		MOV	#750,R1	
1911	003222	005020		STARTF:	CLR	(R0)+	;CLEAR STATISTIC COUNTERS
1912	003224	005301			DEC	R1	
1913	003226	001375			BNE	STARTF	
1914	003230	012737	177777	014406	MOV	#-1,PATS	;PRESET PATTERN
1915	003236	005037	000744		CLR	EOPB1	
1916	003242	012737	000001	000654	STARTE:	MOV	#1,BLCNTR ;PRESET BLOCK COUNTER
1917	003250	012706	000500		STARTD:	MOV	#500,SP
1918	003254	012777	000340	175322	MOV	#340,@PSW	
1919	003262	013746	000006		SUSWR:	MOV	@#6,-(SP) ;SAVE VECTORS
1920	003266	013746	000004		MOV	@#4,-(SP)	
1921	003272	012737	003312	000004	MOV	#1\$,@#4	;SET UP FOR TIMEOUT
1922	003300	022777	177777	175300	CMP	#-1,@SWR	;REFERENCE HARDWARE SWITCH REGISTER
1923	003306	001402			BEQ	2\$	
1924	003310	000404			BR	3\$	
1925	003312	022626		1\$:	CMP	(SP)+,(SP)+	;ADJUST STACK
1926	003314	012737	000176	000606	2\$:	MOV	#SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
1927	003322	012637	000004		3\$:	MOV	(SP)+,@#4 ;RESTORE VECTORS
1928	003326	012637	000006		MOV	(SP)+,@#6	
1929	003332	022737	000176	000606	CMP	#SWREG,SWR	;IS SWREG SELECTED
1930	003340	001020			BNE	4\$	
1931	003342	005737	000744		TST	EOPB1	
1932	003346	001015			BNE	4\$	
1933	003350	005037	000744		CLR	EOPB1	
1934	003354	022737	005116	000042	CMP	#SENDAD,@#42	;ACT MODE? ++ C.W
1935	003362	001402			BEQ	6\$	;BRANCH - IF YES ++ C.W
1936	003364	004737	024452		JSR	PC,CNTLU	;CHECK FOR CONTROL G
1937	003370	005737	000734	6\$:	TST	ASEQF	;AUTO SEQ MODE? ++ C.W
1938	003374	001402			BEQ	4\$	;BRANCH - IF NO ++ C.W
1939	003376	000137	022016		JMP	ASEQ0	;GO DO AUTO SEQ ++ C.W
1940	003402	004737	012400	4\$:	JSR	PC,TINP	;GO GET PARAMETERS FROM TTY
1941	003406	012777	000040	175104	MOV	#40,@CS	;INITIALIZE
1942	003414	005000		STAUTO:	CLR	R0	;POINT TO FIRST ENTRY
1943	003416	022760	177777	000746	1\$:	CMP	#-1,UN1(R0) ;++G BRANCH IF LAST ENTRY
1944	003424	001406			BEQ	2\$	
1945	003426	042760	100000	000746	BIC	#100000,UN1(R0)	;CLEAR EOT FLAG
1946	003434	062700	000002		ADD	#2,R0	;POINT TO NEXT UNIT ENTRY
1947	003440	000766			BR	1\$	;++G CONTINUE CLEARING
1948	003442	013703	005156	2\$:	MOV	REOTC,R3	
1949	003446	000303			SWAB	R3	
1950	003450	110337	005156		MOVB	R3,REOTC	;RESTORE EOT CNTR
1951	003454	012777	000100	175126	START1:	MOV	#100,@TKS ;SET TTY INTERRUPT ENABLE
1952	003462	013700	000674		MOV	UNP,R0	;R0 = UNIT TABLE POINTER
1953	003466	022760	177777	000746	STAR1A:	CMP	#-1,UN1(R0) ;++G BRANCH IF LAST ENTRY
1954	003474	001404			BEQ	STAR1B	;IF LAST UNIT IN STRING: BR
1955	003476	016037	000746	000552	MOV	UN1(R0),UDES	;LOAD NEXT UNIT DESCRIPTION
1956	003504	000446			BR	START4	;++G
1957	003506	005237	000654		STAR1B:	INC	BLCNTR ;BUMP BLOCK COUNTER
1958	003512	005737	000734		TST	ASEQF	;SEE IF AUTO SEQ
1959	003516	001411			BEQ	STAR1C	;IF NOT: BR
1960	003520	023737	000654	000740	CMP	BLCNTR,ABL CNT	;SEE IF DONE SEQ

1961	003526	001005			BNE	STAR1C		:IF NOT: BR
1962	003530	005037	000654		CLR	BLCNTR		:RESET BLOCK CNTR
1963	003534	005037	000674		CLR	UNP		:RESET UNIT POINTER
1964	003540	000207			RTS	PC		:RETURN TO AUTO SEQ
1965								
1966	003542	005037	000674		STAR1C:	CLR	UNP	
1967	003546	005000				CLR	RO	
1968	003550	016037	000746	000552	MOV	UN1(RO), UDES		:LOAD FIRST UNIT DESCRIPTION
1969	003556	105777	175024		TSTB	@SWR		:++G BRANCH IF NOT RANDOM RECORD
1970	003562	100003			BPL	START2		:++G SIZE REQUESTED.
1971	003564	001402			BEQ	START2		:IF NOT: BR
1972	003566	004737	012314		JSR	PC, CCNTR		:GO GENERATE RANDOM RECORD SIZE
1973	003572	032777	000400	175006	STAR2:	BIT	#400, @SWR	:SEE IF RANDOM DATA
1974	003600	001402			BEQ	START3		:IF NOT: BR
1975	003602	004737	015102		JSR	PC, DATR		:GO GENERATE RANDOM DATA
1976	003606	032777	000100	174772	STAR3:	BIT	#100, @SWR	:SEE IF RANDOM RECORD COUNT
1977	003614	001402			BEQ	START4		:IF NOT: BR
1978	003616	004737	012354		JSR	PC, RCNTR		:GO GENERATE RANDOM RECORD COUNT
1979	003622	005760	000746		STAR4:	TST	UN1(RO)	:++G BRANCH IF NOT AT EOT
1980	003626	100002			BPL	STAR40		:IF NOT: BR
1981	003630	000137	004360		JMP	START7		:ELSE GO TO NEXT UNIT
1982	003634	013777	000550	174656	STAR40:	MOV	DVN, @CS	:SET DRIVE NUMBER
1983	003642	013777	000552	174672	MOV	UDES, @C2		:SET UNIT NUMBER
1984	003650	105777	174646		TSTB	@DS		:++G BRANCH IF UNIT AVAIL
1985	003654	100412			BMI	STAR4A		
1986	003656	005337	000666		DEC	STAL		
1987	003662	001357			BNE	START4		:AWAIT TUR
1988	003664	004737	022730		JSR	PC, PAPRT		:PRINT HEADER
1989	003670	012704	026066		MOV	#MSG49, R4		
1990	003674	104000			TTOUTT			:PRINT NOT AVAIL
1991	003676	104006			STOPP			:STOP
1992	003700	000750			BR	START4		:++G RETRY
1993	003702	013746	000552		STAR4A:	MOV	UDES, -(SP)	:GET UNIT DESCRIPTION
1994	003706	042716	175400		BIC	#175400, (SP)		:++G CLEAR ALL BUT FORMAT BITS
1995					;CMP	#1700, (SP)+		:++G BRANCH IF NRZ
1996	003712	032726	002000		BIT	#2000, (SP)+		:++H BRANCH IF NZR
1997	003716	001406			BEQ	1\$		:++G
1998	003720	032777	000040	174574	BIT	#40, @DS		:++G BRANCH IF SLAVE IN PE FORMAT
1999	003726	001002			BNE	1\$		:++G
2000	003730	000137	004360		JMP	START7		:++G GO TO NEXT UNIT
2001	003734	004737	014200		1\$:	JSR	PC, DSUP	:GO SET UP WRITE DATA
2002	003740	004737	005160		JSR	PC, RWND		:REWIND
2003	003744	004737	005522		JSR	PC, WRITE		:WRITE
2004	003750	013737	000576	000666	MOV	TSTAL, STAL		:SET TURN AROUND DELAY
2005	003756	004737	012304		JSR	PC, STALL		:DELAY
2006	003762	004737	007414		JSR	PC, RSEQ		:GO TO READ SEQUENCER
2007	003766	013737	000576	000666	MOV	TSTAL, STAL		:SET TURN AROUND DELAY
2008	003774	004737	012304		JSR	PC, STALL		:DELAY
2009	004000	032777	040000	174600	BIT	#40000, @SWR		:SEE IF SHOULD PRINT STATISTICS
2010	004006	001541			BEQ	START5		:IF NOT: BR
2011	004010	012700	000001		MOV	#1, RO		:SET RECORD COUNTER TO 1
2012	004014	004737	022730		JSR	PC, PAPRT		:PRINT CYCLE NUMBER
2013	004020	004737	004030		JSR	PC, STP		:GO PRINT STATS
2014	004024	000137	004276		JMP	STPX		
2015	004030	004737	017250		STP:	JSR	PC, DPPRT	:PRINT DROPS AND PICKS
2016	004034	012704	026301		MOV	#MSG65, R4		

2017	004040	104000		TTOUTT	;PRINT RETRY TOTAL
2018	004042	013704	000674	MOV	UNP,R4
2019	004046	016403	001050	MOV	RTY1(R4),R3
2020	004052	104002		OCTPP	;PRINT RETRIES
2021	004054	012704	026452	MOV	#MSG73,R4
2022	004060	104000		TTOUTT	;PRINT WRITE ERROR TAG
2023	004062	013704	000674	MOV	UNP,R4
2024	004066	016403	001070	MOV	WTER1(R4),R3
2025	G04072	104002		OCTPP	;PRINT WRITE ERRORS
2026	004074	012704	026441	MOV	#MSG72,R4
2027	004100	104000		TTOUTT	;PRINT READ FORWARD ERROR TAG
2028	004102	013704	000674	MOV	UNP,R4
2029	004106	016403	001110	MOV	RDER1(R4),R3
2030	004112	104002		OCTPP	;PRINT READ FORWARD ERRORS
2031	004114	012704	027247	MOV	#MSG113,R4
2032	004120	104000		TTOUTT	;PRINT SOFT TAG
2033	004122	013704	000674	MOV	UNP,R4
2034	004126	016403	002670	MOV	RFSOFT(R4),R3
2035	004132	104002		OCTPP	;PRINT FORWARD SOFT ERRORS
2036	004134	012704	027260	MOV	#MSG114,R4
2037	004140	104000		TTOUTT	;PRINT HARD TAG
2038	004142	013704	000674	MOV	UNP,R4
2039	004146	016403	002730	MOV	RFHARD(R4),R3
2040	004152	104002		OCTPP	;PRINT HARD FORWARE ERRORS
2041	004154	012704	026532	MOV	#MSG77,R4
2042	004160	104000		TTOUTT	;PRINT DATA ERROR FORWARD TAG
2043	004162	013704	000674	MOV	UNP,R4
2044	004166	016403	001130	MOV	DATER1(R4),R3
2045	004172	104002		OCTPP	;PRINT DATA ERROR FORWARD NUMBER
2046	004174	012704	026335	MOV	#MSG68,R4
2047	004200	104000		TTOUTT	;PRINT READ ERROR REVERSE TAG
2048	004202	013704	000674	MOV	UNP,R4
2049	004206	016403	001150	MOV	RDERR1(R4),R3
2050	004212	104002		OCTPP	;PRINT REVESE ERROR NUMBER
2051	004214	012704	027247	MOV	#MSG113,R4
2052	004220	104000		TTOUTT	;PRINT SOFT TAG
2053	004222	013704	000674	MOV	UNP,R4
2054	004226	016403	002710	MOV	RRSOFT(R4),R3
2055	004232	104002		OCTPP	;PRINT REVERSE SOFT ERROR
2056	004234	012704	027260	MOV	#MSG114,R4
2057	004240	104000		TTOUTT	;PRINT HARD TAG
2058	004242	013704	000674	MOV	UNP,R4
2059	004246	016403	002750	MOV	RRHARD(R4),R3
2060	004252	104002		OCTPP	
2061	004254	012704	026521	MOV	#MSG76,R4
2062	004260	104000		TTOUTT	;PRINT DATA ERROR REVERSE TAG
2063	004262	013704	000674	MOV	UNP,R4
2064	004266	016403	001170	MOV	DEREV1(R4),R3
2065	004272	104002		OCTPP	;PRINT DATA REVERSE ERROR NUMBER
2066	004274	000207		RTS	PC
2067	004276	005237	000726	STPX: INC	BTSTF
2068	004302	004737	007324	JSR	PC,BTPRT
2069	004306	005037	000726	CLR	BTSTF
2070	004312	017700	174270	START5: MOV	@SWR,R0
2071	004316	042700	177762	BIC	#177762,R0
2072	004322	022700	000015	CMP	#15,R0

```
2073 004326 001417          BEQ      START8          ;IF NOT: BR
2074 004330 105777 174166   START6: TSTB     @DS          ;++G BRANCH IF HAVE UNIT READY
2075 004334 100411          BMI      START7          ;++G
2076 004336 005337 000666   DEC      STAL
2077 004342 001372          BNE      START6          ;DELAY FOR TUR
2078 004344 004737 022730   JSR      PC,PAPRT        ;PRINT HEADER
2079 004350 012704 026066   MOV      #MSG49,R4
2080 004354 104000          TTOUTT
2081 004356 104006          STOPP
2082 004360 062737 000002 000674 START7: ADD      #2,UNP        ;POINT TO NEXT UNIT
2083 004366 000137 003454   START8: JMP      START1      ;CONTINUE
2084
2085          ;RANDOM BASE RESET*****
2086
2087 004372 012737 153624 000624 RANSET: MOV      #153624,RANBAS ;RESET BASE
2088 004400 012737 032561 000626   MOV      #32561,RANSAV    ;RESET BUFFER
2089 004406 013737 000630 000554   MOV      RCSAV,RCNT       ;RESET RECORD COUNT
2090 004414 013737 000632 000556   MOV      FCSAV,FMCNT     ;RESET FRAME COUNT
2091 004422 000207          RTS      PC
2092
```







```

2204 ;*****
2205 ;REWIND ALL AVAIL TAPES:
2206 ;
2207 ;THIS ROUTINE; ENTERED VIA CONSOLE SWITCH NINE (9),
2208 ;WILL REWIND ALL AVAILABLE TAPES TO BOT NO MATTER
2209 ;WHERE THEY ARE CURRENTLY POSITIONED AND RESUME TESTING
2210 ;ON THE CURRENTLY SELECTED UNIT.
2211 ;*****
2212
2213 005160 032777 001000 173420 RWND: BIT #1000,@SWR ;SEE IF SHOULD REWIND
2214 005166 001001 BNE RWNDA ;IF SO: BR
2215 005170 000207 RTS PC ;ELSE EXIT
2216 005172 013737 000674 000714 RWNDA: MOV UNP,UPS ;SAVE UNIT POINTER
2217 005200 005037 000674 CLR UNP ;CLEAR POINTER
2218 005204 005037 000660 CLR EOTREC ;CLEAR EDT FLAG
2219 005210 000337 005156 SWAB REOTC
2220 005214 013700 005156 MOV REOTC,R0
2221 005220 000337 005156 SWAB REOTC
2222 005224 110037 005156 MOV#B R0,REOTC ;RESTORE EOT UNIT COUNTER
2223 005230 013700 000674 RWND0: MOV UNP,R0 ;POINT TO UNIT ENTRY
2224 005234 022760 177777 000746 CMP #-1,UN1(R0) ;++G BRANCH IF LAST ENTRY
2225 005242 001445 BEQ RWND2 ;IF SO: BR
2226 005244 005760 000746 TST UN1(R0) ;++G BRANCH IF ALREADY REWINDING
2227 005250 100433 BMI RWND1A ;++G
2228 005252 016037 000746 000552 MOV UN1(R0),UDES ;SET UNIT DESCRIPTION
2229 005260 013777 000552 173254 MOV UDES,@C2 ;LOAD COMMAND REGISTER
2230 005266 012777 000011 173214 MOV #11,@C1 ;DRIVE CLEAR
2231 005274 012777 000007 173206 MOV #7,@C1 ;START REWIND
2232 005302 105777 173214 RWND1: TSTB @DS ;++G WAIT FOR DRIVE READY
2233 005306 100414 BMI RWND1A ;IF DRY: BR
2234 005310 005337 000666 DEC STAL
2235 005314 001372 BNE RWND1 ;AWAIT DRY
2236 005316 012737 024761 000652 MOV #MSG6,EMADDR
2237 005324 004737 022730 JSR PC,PAPRT ;PRINT HEADER
2238 005330 012704 026364 MOV #MSG70,R4
2239 005334 104000 TTOUTT ;PRINT NO DRIVE READY
2240 005336 104006 STOPP
2241 005340 042760 100000 000746 RWND1A: BIC #100000,UN1(R0) ;CLEAR EOT FLAG
2242 005346 062737 000002 000674 ADD #2,UNP ;BUMP POINTER
2243 005354 000725 BR RWND0 ;++G DO NEXT UNIT
2244 005356 005037 000674 RWND2: CLR UNP ;CLEAR POINTER
2245 005362 013700 000674 RWND3: MOV UNP,R0 ;POINT TO UNIT ENTRY
2246 005366 022760 177777 000746 CMP #-1,UN1(R0) ;++G BRANCH IF LAST ENTRY
2247 005374 001436 BEQ RWNDX ;IF SO: BR
2248 005376 016037 000746 000552 MOV UN1(R0),UDES ;SET UNIT DESCRIPTION
2249 005404 013777 000552 173130 MOV UDES,@C2 ;LOAD COMMAND REGISTER
2250 005412 032777 020000 173102 RWND4: BIT #20000,@DS
2251 005420 001374 BNE RWND4 ;AWAIT PIP RESET
2252 005422 032777 000002 173072 BIT #2,@DS ;SEE IF HAVE BOT
2253 005430 001407 BEQ RWND6 ;IF NOT: BR
2254 005432 062737 000002 000674 RWND5: ADD #2,UNP ;BUMP POINTER
2255 005440 012777 000011 173042 MOV #11,@C1 ;DRIVE CLEAR
2256 005446 000745 BR RWND3 ;++G DO NEXT UNIT
2257 005450 012700 000001 RWND6: MOV #1,R0
2258 005454 004737 022730 JSR PC,PAPRT ;PRINT HEADER
2259 005460 012704 026034 MOV #MSG48,R4
  
```







2381	006214	032777	000020	172364	BIT	#20,@SWR	:SEE IF SHOULD RETRY	
2382	006222	001436			BEQ	WEX	:IF NOT: BR	
2383	006224	013703	000722		MOV	ERSAV,R3		
2384	006230	042703	102700		BIC	#102700,R3	:MASK UNRECOVERABLE ERROR	
2385	006234	001410			BEQ	WTM4	:IF SO: BR	
2386	006236	004737	022730		JSR	PC,PAPRT	:PRINT HEADER	
2387	006242	012704	026543		MOV	#MSG78,R4		
2388	006246	104000			TTOUTT		:PRINT UNRETRYABLE TAG	
2389	006250	004737	011260		JSR	PC,NRTP	:PRINT ER FOR NON-RETRYABLE	
2390	006254	000421			BR	WEX		
2391	006256	005037	000700		WTM4:	CLR	RPCNT	:CLEAR REPEAT CNTR
2392	006262	013704	000674		MOV	UNP,R4		
2393	006266	005264	001050		INC	RTY1(R4)	:BUMP RETRY CNTR	
2394	006272	005037	000702		CLR	RTCNT	:CLEAR RETRY CNTR	
2395	006276	032777	002000	172302	BIT	#2000,@SWR	:SEE IF PRINT ERRORS	
2396	006304	001003			BNE	WTM4A	:IF NOT: BR	
2397	006306	012704	026257		MOV	#MSG64,R4		
2398	006312	104000			TTOUTT		:PRINT ORIGINAL ERROR TAG	
2399	006314	004737	006362		WTM4A:	JSR	PC,WRTY	:GO DO RETRY
2400	006320	005037	000712		WEX:	CLR	RTYFL	:CLEAR RETRY FLAG
2401	006324	005037	000676			CLR	TMFLG	:CLEAR TAPE MARK FLAG
2402	006330	005737	000660			TST	EOTREC	:++G BRANCH IF NOT AT EOT
2403	006334	100011				BPL	WRWX	:++G
2404	006336	017703	172244		WRW:	MOV	@SWR,R3	
2405	006342	042703	177763			BIC	#177763,R3	
2406	006346	022703	000014			CMP	#14,R3	:SEE IF WRITE ONLY
2407	006352	001002				BNE	WRWX	:IF NOT: BR
2408	006354	000137	004424			JMP	REOT	:ELSE REWIND
2409	006360	000207			WRWX:	RTS	PC	:EXIT

```
2410 ;*****
2411 ;WRITE ERROR RETRY
2412 ;
2413 ;*****
2414
2415 006362 012737 000001 000712 WRTY: MOV #1,RTYFL ;SET RETRY FLAG
2416 006370 004737 006764 WRTY0: JSR PC,WRTSB ;GO SPACE REVERSE FOR REPEAT
2417 006374 005737 000676 TST TMFLG ;SEE IF TAPE MARK TIME
2418 006400 001003 BNE WRTYTM ;IF SO: BR
2419 006402 004737 005542 JSR PC,W0 ;REWRITE RECORD
2420 006406 000402 BR WRTYR ;GO ON
2421 006410 004737 006052 WRTYTM: JSR PC,WTM ;GO WRITE TAPE MARK AGAIN
2422 006414 005737 000706 WRTYR: TST SERFL ;REWRITE GOOD
2423 006420 001024 BNE WRTY2 ;IF NOT: BR
2424 006422 005237 000700 INC RPCNT ;BUMP REPEAT COUNTER
2425 006426 022737 000004 000700 CMP #4,RPCNT ;SEE IF FOUR GOOD REPEATS
2426 006434 001355 BNE WRTY0 ;IF NOT: REPEAT
2427 006436 032777 002000 172142 BIT #2000,@SWR ;SEE IF PRINT
2428 006444 001011 BNE WRTY1 ;IF NOT: BR
2429 006446 012704 026736 MOV #MSG105,R4
2430 006452 104000 TTOUTT ;PRINT RECOVERED MESSAGE
2431 006454 012704 026301 MOV #MSG65,R4
2432 006460 104000 TTOUTT ;PRINT RETRY TAG
2433 006462 013703 000702 MOV RTCNT,R3
2434 006466 104002 OCTPP ;PRINT RETRY NUMBER
2435 006470 000207 WRTY1: RTS PC ;RESUME TESTING
2436 006472 005037 000646 WRTY2: CLR TEMP3 ;++G CLEAR RECOVERABLE ERROR FLAG
2437 006476 013703 000722 MOV ERSAV,R3 ;GET ER
2438 006502 042703 102700 BIC #102700,R3 ;MASK RECOVERABLE BITS
2439 006506 001413 BEQ WRTY2A ;IF RECOVERABLE: BR
2440 006510 004737 022730 JSR PC,PAPRT ;PRINT HEADER
2441 006514 012704 026543 MOV #MSG78,R4
2442 006520 104000 TTOUTT ;PRINT NON-RECOVERABLE MSG
2443 006522 004737 011260 JSR PC,NRTP ;PRINT ER
2444 006526 012737 000001 000646 MOV #1,TEMP3 ;SET FLAG
2445 006534 000407 BR WRTY2B
2446 006536 032777 002000 172042 WRTY2A: BIT #2000,@SWR ;SEE IF PRINT
2447 006544 001025 BNE WRTY3 ;IF NOT: BR
2448 006546 012704 027172 MOV #MSG110,R4
2449 006552 104000 TTOUTT ;PRINT BAD TAPE SUSPECT
2450 006554 012704 026301 WRTY2B: MOV #MSG65,R4
2451 006560 104000 TTOUTT ;PRINT RETRY TAG
2452 006562 013703 000702 MOV RTCNT,R3
2453 006566 104002 OCTPP ;PRINT RETRY NUMBER
2454 006570 012704 027214 MOV #MSG111,R4
2455 006574 104000 TTOUTT ;PRINT REPEAT TAG
2456 006576 013703 000700 MOV RPCNT,R3
2457 006602 104002 OCTPP ;PRINT REPEAT NUMBER
2458 006604 005737 000646 TST TEMP3 ;SEE IF DID NON-RECOVERABLE
2459 006610 001403 BEQ WRTY3 ;IF NOT: BR
2460 006612 005037 000646 CLR TEMP3 ;CLEAR FLAG
2461 006616 000207 RTS PC ;EXIT
2462 006620 005737 000702 WRTY3: TST RTCNT ;SEE IF FIRST RETRY
2463 006624 001004 BNE WRTY3A ;IF NOT: BR
2464 006626 013704 000674 MOV UNP,R4
2465 006632 005364 001070 DEC WTER1(R4) ;DECREMENT WRITE ERROR CNTR
```

```

2466 006636 013704 000674          WRTY3A: MOV      UNP,R4          ;GET UNIT NUMBER
2467 006642 016437 001030 000730  MOV      BTADDR(R4),BTPT ;GET ADDRESS OF UNIT BAD TAPE CNTR
2468 006650 017704 172054          MOV      @BTPT,R4        ;GET COUNTER
2469 006654 005724          TST      (R4)+          ;SET POINTER OFFSET
2470 006656 010477 172046          MOV      R4,@BTPT
2471 006662 013703 000730          MOV      BTPT,R3
2472 006666 060304          ADD      R3,R4          ;SET ABSOLUTE POINTER
2473 006670 013714 000654          MOV      BLCNTR,(R4)    ;SET BLOCK NUMBER
2474 006674 062704 000040          ADD      #40,R4        ;ADD RCNT OFFSET
2475 006700 013714 000554          MOV      RCNT,(R4)
2476 006704 160014          SUB      R0,(R4)        ;SET RECORD NUMBER
2477 006706 005214          INC      (R4)          ;CORRECT RECORD NUMBER
2478 006710 022777 000040 172012  CMP      #40,@BTPT     ;SEE IF TOO MANY BAD SPOTS
2479 006716 001002          BNE      WRTY4         ;IF NOT: BR
2480 006720 000137 007160          JMP      BTOV          ;ELSE GO TO BAD TAPE OVERFLOW
2481 006724 005237 000702          WRTY4: INC      RTCNT     ;BUMP RETRY COUNTER
2482 006730 022737 000004 000702  CMP      #4,RTCNT     ;SEE IF DONE 4 RETRIES
2483 006736 001410          BEQ      WRTY5         ;IF SO: BR
2484 006740 013704 000674          MOV      UNP,R4
2485 006744 005264 001050          INC      RTY1(R4)      ;BUMP RETRY COUNTER
2486 006750 005237 000732          INC      ERTFL         ;SET ERASE FLAG
2487 006754 000137 006370          JMP      WRTY0         ;DO NEXT RETRY
2488 006760 000137 007400          WRTY5: JMP      BTUR          ;ELSE GO TO BAD TAPE UNRECOVERABLE
2489
2490          ;WRITE RETRY BACKSPACE-ERASE SUBROUTINE*****
2491
2492 006764 005037 000706          WRTSB: CLR      SERFL      ;CLEAR FLAG
2493 006770 013737 000576 000666  MOV      TSTAL,STAL
2494 006776 004737 012304          JSR      PC,STALL     ;DO TURN AROUND DELAY
2495 007002 012737 026312 000652  MOV      #MSG66,EMADDR ;SET ERROR CODE
2496 007010 012777 177777 171500  MOV      #-1,@FC      ;SET TO BACKSPACE 1 RECORD
2497 007016 012777 033464 171470  MOV      #RDATA,@BA   ;SET BA
2498 007024 004737 012234          JSR      PC,BKRT      ;GO BACKSPACE
2499 007030 005737 000706          TST      SERFL        ;SEE IF ERROR
2500 007034 001406          BEQ      WRTSB1       ;IF NOT: BR
2501 007036 012737 000002 000724  WRTSB0: MOV      #2,BTFLG   ;SET FLAG
2502 007044 022626          CMP      (SP)+,(SP)+  ;RESET STACK
2503 007046 000137 004424          JMP      REOT         ;GO REWIND AND REMOVE FROM TESTING
2504 007052 005737 000732          WRTSB1: TST      ERTFL   ;SEE IF SHOULD ERASE
2505 007056 001001          BNE      WRTSB2       ;IF SO: BR
2506 007060 000207          RTS      PC           ;RETURN
2507 007062 005037 000732          WRTSB2: CLR      ERTFL   ;CLEAR ERASE FLAG
2508 007066 005037 000700          CLR      RPCNT       ;CLEAR REPEAT CNTR
2509 007072 005037 000706          CLR      SERFL       ;CLEAR FLAG
2510 007076 012737 026325 000652  MOV      #MSG67,EMADDR ;SET ERROR CODE
2511 007104 005077 171406          CLR      @FC         ;CLEAR FRAME COUNT
2512 007110 012737 000024 000672  MOV      #24,MTC1     ;SET ERASE OP-CODE
2513 007116 012777 027456 171370  MOV      #WDATA,@BA   ;SET BA
2514 007124 012737 007136 000662  MOV      #WRTSB3,RTRN ;SET RETURN ADDRESS
2515 007132 000137 021216          JMP      TAPG         ;GO ERASE
2516 007136 012703 027456          WRTSB3: MOV      #WDATA,R3 ;SET EXPT BA
2517 007142 004737 017504          JSR      PC,ER2      ;GO CHECK ERRORS
2518 007146 005737 000706          TST      SERFL       ;SEE IF ERROR
2519 007152 001737          BEQ      WRTSB1       ;IF NOT: BR
2520 007154 000137 007036          JMP      WRTSB0
2521
    
```



:BAD TAPE OVERFLOW SUBROUTINE\*\*\*\*\*

```
2522  
2523  
2524 007160 005037 000712      BTOV: CLR      RTYFL      ;CLEAR RETRY FLAG  
2525 007164 012737 000001 000724  MOV      #1,BTFLG      ;SET BAD TAPE OVERFLOW FLAG  
2526 007172 005726          TST      (SP)+        ;++G ADJUST STACK  
2527 007174 000137 004424          JMP      REOT         ;GO REWIND AND REMOVE FROM TESTING  
2528 007200 013701 000730      BTOV0: MOV      BTPT,R1      ;SET TABLE POINTER  
2529 007204 005721          TST      (R1)+  
2530 007206 005000          CLR      R0  
2531 007210 010003      BTOV1: MOV      R0,R3  
2532 007212 000241          CLC  
2533 007214 006003          ROR      R3           ;R3=R3/2 FOR CORRECT NUMBER  
2534 007216 104002          OCTPP          ;PRINT ENTRY NUMBER  
2535 007220 012704 025046      MOV      #MSG13,R4  
2536 007224 105724          TSTB     (R4)+        ;SKIP CR/LF  
2537 007226 104000          TTOUTT        ;PRINT BLOCK NUMBER TAG  
2538 007230 011103          MOV      (R1),R3  
2539 007232 104002          OCTPP          ;PRINT BLOCK NUMBER  
2540 007234 012704 025054      MOV      #MSG14,R4  
2541 007240 104000          TTOUTT        ;PRINT RECORD NUMBER TAG  
2542 007242 062701 000040      ADD      #40,R1       ;SET POINTER OFFSET FOR RECOED NUMBER  
2543 007246 012103          MOV      (R1)+,R3  
2544 007250 104002          OCTPP          ;PRINT RECORD NUMBER  
2545 007252 162701 000040      SUB      #40,R1       ;RESET POINTER FOR BLOCK NUMBER  
2546 007256 005720          TST      (R0)+  
2547 007260 020077 171444      CMP      R0,@BTPT     ;SEE IF DONE  
2548 007264 001404          BEQ      BTOV2        ;IF SO: BR  
2549 007266 012704 025377      MOV      #MSG28,R4  
2550 007272 104000          TTOUTT        ;DO CR/LF  
2551 007274 000745          BR       BTOV1        ;CONTINUE  
2552 007276 005737 000726      BTOV2: TST      BTSTF     ;SEE IF STAT ONLY PRINT  
2553 007302 001007          BNE      BTOVX        ;IF SO: BR  
2554 007304 012703 000041      MOV      #41,R3       ;SET SIZE OF TABLE  
2555 007310 013704 000730      MOV      BTPT,R4  
2556 007314 005024      BTOV3: CLR      (R4)+    ;CLEAR TABLE  
2557 007316 005303          DEC      R3           ;SEE IF DONE  
2558 007320 001375          BNE      BTOV3        ;IF NOT: BR  
2559 007322 000207      BTOVX: RTS      PC     ;RETURN  
2560
```

```
2561
2562
2563
2564 007324 012704 025377
2565 007330 104000
2566 007332 013704 000674
2567 007336 016437 001030 000730
2568 007344 017703 171360
2569 007350 000241
2570 007352 006003
2571 007354 104002
2572 007356 012704 027226
2573 007362 104000
2574 007364 005777 171340
2575 007370 001001
2576 007372 000207
2577 007374 000137 007200
2578
2579
2580
2581 007400 004737 022730
2582 007404 012704 027041
2583 007410 104000
2584 007412 000207
2585
```

```

;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 SPOTS
        BNE BTPRT1 ;IF SO: BR
        RTS PC ;ELSE RETURN
BTPRT1: JMP BTOVO ;PRINT STATS

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





2654 ;\*\*\*\*\*  
2655 ;READ ROUTINE:  
2656 ;  
2657 ;THIS ROUTINE PERFORMS THE READ OPERATION DETERMINED  
2658 ;BY THE READ SEQUENCE ROUTINE ONE RECORD AT A TIME.  
2659 ;AT THE END OF EACH READ OPERATION THE STATUS REGISTER  
2660 ;IS SCANNED FOR EITHER END OF TAPE OR BEGINNING OF TAPE.  
2661 ;IF EOT WAS REACHED, CONTROL WILL BE PASSED TO  
2662 ;THE EOT SUBROUTINE TO REWIND THE UNIT AND FLAG IT  
2663 ;UNAVAILABLE UNTIL ALL UNITS HAVE REACHED EOT.  
2664 ;IF BOT WAS REACHED AN ERROR IS PRINTED AND THE  
2665 ;PROGRAM WILL HALT. TESTING MAY BE RESUMED BY PRESSING  
2666 ;THE CONTINUE SWITCH.  
2667 ;IF A TAPE MARK IS EXPECTED (TM=1) THEN THE  
2668 ;READ ROUTINE EXPECTS THE FIRST RECORD OF A  
2669 ;READ REVERSE TO BE A TM, AND THE LAST RECORD  
2670 ;OF A READ FORWARD TO BE A TM. REMEMBER  
2671 ;THAT THE TM ADDS ONE (1) TO THE TOTAL NUMBER  
2672 ;OF RECORDS IN A BLOCK.  
2673 ;CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13) DETERMINE WHETHER  
2674 ;OR NOT TO CHECK FOR STATUS ERRORS (11) OR DATA ERRORS (13),  
2675 ;CONSOLE SWITCH FIVE (5) IS USED TO CAUSE A CONTINUOUS  
2676 ;READ AND SPACE (FORWARD OR REVERSE) OF THE CURRENT  
2677 ;RECORD ON TAPE (YOZZLE).  
2678 ;\*\*\*\*\*  
2679

2680	007732	013700	000554		READ:	MOV	RCNT,R0		;LOAD REC CNTR
2681	007736	005737	000660			TST	EOTREC		;SEE IF EOT
2682	007742	100013				BPL	RDA		;IF NOT: BR
2683	007744	032737	010000	000562		BIT	#10000,RDCMD		;SEE IF READ FORWARD
2684	007752	001407				BEQ	RDA		;IF SO: BR
2685	007754	042737	100000	000660		BIC	#100000,EOTREC		;CLEAR FLAG
2686	007762	013703	000660			MOV	EOTREC,R3		;GET MODIFIED RECORD COUNT
2687	007766	160300				SUB	R3,R0		;SET RECORD AT
2688	007770	005200				INC	R0		;SET TO PROPER NUMBER OF RECORDS
2689	007772	012737	024761	000652	RDA:	MOV	#MSG6,EMADDR		;SET ERROR MSG ADDRESS
2690	010000	005037	000676			CLR	TMFLG		
2691	010004	032737	010000	000562		BIT	#10000,RDCMD		
2692	010012	001406				BEQ	RDO		;IF READ FORWARD: BR
2693	010014	005737	000564			TST	TMEX		;SEE IF TM
2694	010020	001403				BEQ	RDO		;IF NOT: BR
2695	010022	005237	000676			INC	TMFLG		;SET TM FLAG
2696	010026	005200				INC	R0		
2697	010030	013777	000556	170460	RDO:	MOV	FMCNT,@FC		;LOAD CHAR CNTR
2698	010036	012777	033464	170450		MOV	#RDATA,@BA		;LOAD DATA ADDR
2699	010044	032737	010000	000562		BIT	#10000,RDCMD		;SEE IF READ REVERSE
2700	010052	001417				BEQ	RD1A		;IF NOT: BR
2701	010054	013703	000556			MOV	FMCNT,R3		
2702	010060	005103				COM	R3		
2703	010062	032737	000020	000552		BIT	#20,UDES		;SEE IF CORE DUMP
2704	010070	001402				BEQ	RD1		;IF NOT: BR
2705	010072	000241				CLC			
2706	010074	006003				ROR	R3		;R3 = FC/2
2707	010076	060377	170412		RD1:	ADD	R3,@BA		;SET REVERSE BUS ADDRESS
2708	010102	012737	000076	000672		MOV	#76,MTC1		;SET READ REVERSE
2709	010110	000403				BR	RD1B		













```

2954 011562 032737 000020 000552 BIT #20,UDES ;SEE IF CORE DUMP
2955 011570 001402 BEQ YOZD4 ;IF NOT: BR
2956 011572 000241 CLC
2957 011574 006004 ROR R4 ;SET TO FC/2
2958 011576 060403 YOZD4: ADD R4,R3 ;SET EXPT BUS ADDRESS
2959 011600 042703 000001 BIC #1,R3 ;MAKE EXPT ADDRESS EVEN
2960 011604 032737 002000 000552 BIT #2000,UDES ;SEE IF PE
2961 011612 001001 BNE YOZD2 ;IF SO: BR
2962 011614 005743 TST -(R3) ;SET EXPT BA
2963 011616 004737 017504 YOZD2: JSR PC,ER2 ;GO CHECK ERRORS
2964 011622 000430 BR YOZF
2965 011624 012703 033464 YOZD0: MOV #RDATA,R3
2966 011630 032737 002000 000552 BIT #2000,UDES ;SEE IF PE
2967 011636 001001 BNE YOZD3 ;IF SO: BR
2968 011640 005723 TST (R3)+ ;SET EXPT BA
2969 011642 004737 017504 YOZD3: JSR PC,ER2 ;GO CHECK ERRORS
2970 011646 000416 BR YOZF
2971 011650 004737 017406 YOZD1: JSR PC,ERCHK ;ELSE GO CHECK ERRORS
2972 011654 032777 020000 166724 YOZE: BIT #20000,@SWR ;SEE IF SHOULD CHECK DATA
2973 011662 001010 BNE YOZF ;IF NOT: BR
2974 011664 005737 000676 TST TMFLG ;SEE IF TAPE MARK
2975 011670 001005 BNE YOZF ;IF SO: BR
2976 011672 005737 000712 TST RTYFL ;SEE IF RETRY
2977 011676 001004 BNE YOZFO ;IF SO: BR
2978 011700 004737 015544 JSR PC,DCHK ;ELSE GO CHECK DATA
2979 011704 004737 014352 YOZF: JSR PC,DS3 ;GO CLEAR DATA AREA
2980 011710 105777 166674 YOZFO: TSTB @TKS ;SEE IF HAVE NEW STALL VALUE
2981 011714 100032 BPL YOZG ;IF NOT: BR
2982 011716 122777 000203 166666 CMPB #203,@TKB ;SEE IF CONT C
2983 011724 001026 BNE YOZG ;IF NOT: BR
2984 011726 012704 025761 MOV #MSG44,R4
2985 011732 104000 TTOUTT ;PRINT YSTALL REQUEST
2986 011734 013703 000600 MOV YSTAL,R3
2987 011740 104002 OCTPP ;PRINT PRESENT STALL
2988 011742 010037 000646 MOV RO,TEMP3 ;SAVE RO(REC CNT)
2989 011746 012705 000600 MOV #YSTAL,R5 ;SET ADDRESS OF YSTL
2990 011752 012701 000006 MOV #6,R1 ;SET NUMBER OF CHAR TO INPUT
2991 011756 012702 177777 MOV #-1,R2 ;SET MAXIMUM LIMIT
2992 011762 012703 002000 MOV #2000,R3 ;SET MINIMUM LIMIT
2993 011766 004737 023346 JSR PC,TTR ;GO GET VALUE
2994 011772 013700 000646 MOV TEMP3,RO ;RESTORE RO(REC CNTR)
2995 011776 000137 011274 JMP YOZ ;RESTART YOZZLE
2996 012002 122777 000207 166602 YOZG: CMPB #207,@TKB ;CHECK FOR CNTL G
2997 012010 001010 BNE YOZI
2998 012012 022737 000176 000606 CMP #SWREG,SWR ;IS SWREG SELECTED
2999 012020 001004 BNE YOZI
3000 012022 005077 166564 CLR @TKB ;CLEAR CNTL G OUT OF BUFFER
3001 012026 004737 024444 JSR PC,CNTG ;GO CHANGE SWREG
3002 012032 032777 000040 166546 YOZI: BIT #40,@SWR ;SEE IF SHOULD CONTINUE YOZZLE
3003 012040 001402 BEQ YOZH ;IF NOT: BR
3004 012042 000137 011314 JMP YOZO
3005 012046 012777 000100 166534 YOZH: MOV #100,@TKS ;SET TTY INTERRUPT ENABLE
3006 012054 000207 RTS PC ;EXIT
3007
    
```

```
3008
3009
3010 ;*****
3011 ;BACKSPACE SUBROUTINE:
3012 ;
3013 ;THIS SUBROUTINE IS USED TO PERFORM THE
3014 ;BACKSPACE OPERATION REQUIRED BY THE READ
3015 ;ROUTINE FOR READ FORWARD AFTER WRITING.
3016 ;IF A TAPE MARK IS EXPECTED (TM=1) THEN THE SPACE
3017 ;ROUTINE ASSUMES THAT THE TM WILL BE FIRST WHEN
3018 ;BACKSPACING. THEREFORE TWO OPERATIONS ARE REQUIRED
3019 ;TO SPACE OVER A BLOCK. FIRST SPACE OVER THE TM, THEN
3020 ;SPACE OVER THE DATA RECORDS.
3021 ;A CHECK FOR RECORD COUNT ZERO IS MADE AT THE
3022 ;END OF THE SPACE OPERATION TO ASSURE THAT PROPER
3023 ;TAPE POSITIONING WAS DONE.
3024 ;*****
3025 012056 013737 000576 000666 BKSP: MOV TSTAL,STAL
3026 012064 004737 012304 JSR PC,STALL ;DO TURN AROUND STALL
3027 012070 012737 025011 000652 MOV #MSG10,EMADDR
3028 012076 012777 033464 166410 MOV #RDATA,@BA
3029 012104 005737 000564 TST TMEX ;SEE IF TM
3030 012110 001440 BEQ BO ;IF NOT: BR
3031 012112 012777 177777 166376 MOV #-1,@FC
3032 012120 012737 000032 000672 MOV #32,MTC1
3033 012126 012737 012140 000662 MOV #BKTM,RTRN
3034 012134 000137 021216 JMP TAPG ;SPACE TO TM
3035 012140 032777 010000 166440 BKTM: BIT #10000,@SWR ;SEE IF SHOULD CHECK ERROR
3036 012146 001021 BNE BO ;IF NOT: BR
3037 012150 012737 026173 000652 MOV #MSG55,EMADDR
3038 012156 032777 000004 166336 BIT #4,@DS ;SEE IF TM
3039 012164 001006 BNE BKTM0 ;IF SO: BR
3040 012166 012737 033464 021132 MOV #RDATA,CADER
3041 012174 004737 020234 JSR PC,ERPT ;PRINT ERROR
3042 012200 000404 BR BO
3043 012202 012703 033464 BKTM0: MOV #RDATA,R3
3044 012206 004737 017504 JSR PC,ER2
3045 012212 013700 000554 BO: MOV RCNT,R0
3046 012216 005100 COM R0 ;BUILD SPACE AMOUNT
3047 012220 005200 INC R0
3048 012222 012737 025011 000652 MOV #MSG10,EMADDR ;SET ERROR MESSG ADDRESS
3049 012230 010077 166262 MOV R0,@FC
3050 012234 012737 000032 000672 BKRT: MOV #32,MTC1 ;SET SPACE REVERSE
3051 012242 012737 012260 000662 MOV #B1,RTRN ;SET RETURN ADDRESS
3052 012250 010037 000666 MOV R0,STAL ;SET INTERRUPT TIME MULTIPLIER
3053 012254 000137 021216 JMP TAPG ;GO DO SPACE
3054 012260 012703 033464 B1: MOV #RDATA,R3
3055 012264 004737 017504 JSR PC,ER2
3056 012270 013737 000576 000666 B2: MOV TSTAL,STAL ;DO STALL
3057 012276 004737 012304 JSR PC,STALL ;STALL
3058 012302 000207 RTS PC ;EXIT
3059
```

3060  
3061  
3062  
3063  
3064  
3065  
3066  
3067  
3068  
3069  
3070  
3071  
3072  
3073  
3074  
3075  
3076  
3077  
3078

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

3079 012304 005337 000666  
3080 012310 001375  
3081 012312 000207

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

3082  
3083  
3084  
3085  
3086  
3087  
3088  
3089  
3090  
3091  
3092  
3093  
3094  
3095  
3096  
3097  
3098  
3099  
3100  
3101  
3102  
3103  
3104  
3105  
3106  
3107  
3108  
3109  
3110  
3111  
3112  
3113  
3114  
3115  
3116  
3117  
3118  
3119

012314 012701 177760  
012320 012702 174000  
012324 004737 023314  
012330 042737 000001 000626  
012336 013737 000626 000556  
012344 012737 177777 014406  
012352 000207  
  
012354 012702 000001  
012360 012701 000500  
012364 004737 023314  
012370 013737 000626 000554  
012376 000207

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

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

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

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

3120  
3121  
3122  
3123  
3124  
3125  
3126  
3127  
3128  
3129  
3130  
3131  
3132  
3133  
3134  
3135  
3136  
3137  
3138  
3139  
3140  
3141  
3142  
3143  
3144  
3145  
3146  
3147  
3148  
3149  
3150  
3151  
3152  
3153  
3154  
3155  
3156  
3157  
3158  
3159  
3160  
3161  
3162  
3163  
3164  
3165

```

:*****
: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 WRITING OF THE TM AND CAUSES THE READ
:AND SPACE ROUTINES TO EXPECT NO TM TO BE PRESENT.
:THE LAST REQUESTS ARE FOR ENTRY OF THE DESIRED
:WRITE, READ, AND TURN AROUND STALLS.
:*****

```

3166	012400	005737	000634	TINP:	TST	TINF	:SEE IF SHOULD INPUT FROM TTY
3167	012404	001001		-	BNE	TINPA	:IF SO: BR
3168	012406	000207			RTS	PC	:EXIT
3169	012410	005037	000674	TINPA:	CLR	UNP	:CLEAR TABLE POINTER
3170	012414	005037	005156		CLR	REOTC	:CLEAR EOT UNIT COUNTER
3171	012420	012700	000010		MOV	#10,R0	:SET SIZE OF TABLE
3172	012424	012701	000746		MOV	#UN1,R1	:SET START OF TABLE
3173	012430	005021		TINPB:	CLR	(R1)+	:CLEAR TABLE
3174	012432	005300			DEC	R0	:SEE IF DONE
3175	012434	001375			BNE	TINPB	:IF NOT: BR





3232	012744	000736				BR	TINPC		;++G RETRY DVN
3233	012746	012704	025515			TINPO: MOV	#MSG32,R4		
3234	012752	104000				TTOUTT			;PRINT UNIT NUMBER REQUEST
3235	012754	005037	000644			CLR	TEMP2		;CLEAR BUFFER
3236	012760	012705	000644			MOV	#TEMP2,R5		;SET UNIT DESCRIPTION BUFFER ADDRESS
3237	012764	012701	000001			MOV	#1,R1		;SET NUMBER OF CHARACTERS TO INPUT
3238	012770	012702	000007			MOV	#7,R2		;SET MAXIMUM LIMIT
3239	012774	012703	000000			MOV	#0,R3		;SET MINIMUM LIMIT
3240	013000	004737	023346			JSR	PC,TTR		;GO GET UNIT NUMBER
3241	013004	005737	000642			TST	TEMP1		;SEE IF HAVE NEW PARAMETER
3242	013010	001013				BNE	TINPOB		;IF SO: BR
3243	013012	005737	000674			TST	UNP		;SEE IF FIRST ENTRY
3244	013016	001001				BNE	TINPOA		;IF NOT: BR
3245	013020	000752				BR	TINPO		;++G ELSE RETRY
3246	013022	013700	000674			TINPOA: MOV	UNP,R0		
3247	013026	012760	177777	000746		MOV	#-1,UN1(R0)		;SET END UNIT TABLE
3248	013034	000137	013424			JMP	TINP2C		;GO GET RECORD COUNT
3249	013040	013700	000674			TINPOB: MOV	UNP,R0		
3250	013044	042760	000007	000746		BIC	#7,UN1(R0)		;CLEAR UNIT NUMBER
3251	013052	004737	014164			JSR	PC,TPOS1		;GO LOAD UNIT NUMBER TO PROPER POSITION
3252	013056	012777	000040	165434		MOV	#40,@CS		
3253	013064	013777	000550	165426		MOV	DVN,@CS		
3254	013072	016077	000746	165442		MOV	UN1(R0),@C2		;LOAD UNIT NUMBER
3255	013100	032777	002000	165430		TINPOC: BIT	#2000,@DT		;SEE IF SLAVE PRESENT
3256	013106	001005				BNE	TINPOD		;IF SO: BR
3257	013110	012704	026206			MOV	#MSG57,R4		
3258	013114	104000				TTOUTT			;PRINT NON-EXIST SLAVE
3259	013116	000137	012746			JMP	TINPO		;REDO
3260	013122	022777	142011	165406		TINPOD: CMP	#142011,@DT		;++G SEE IF 9TRK TM02,TU16/TE16
3261	013130	001406				BEQ	TINPOE		;IF SO: BR
3262	013132	012704	026102			MOV	#MSG50,R4		;ILLEGAL DRIVE TYPE
3263	013136	104000				TTOUTT			;GO PRINT
3264	013140	017703	165372			MOV	@DT,R3		
3265	013144	104002				OCTPP			;PRINT DRIVE TYPE REGISTER
3266	013146	012704	025003			TINPOE: MOV	#MSG9,R4		
3267	013152	104000				TTOUTT			;PRINT SERIAL NUMBER TAG
3268	013154	017703	165360			MOV	@SN,R3		
3269	013160	004737	024272			JSR	PC,SNPT		;PRINT SERIAL NUMBER
3270	013164	012704	025536			TINP1: MOV	#MSG33,R4		
3271	013170	104000				TTOUTT			;PRINT DENSITY REQUEST
3272	013172	005037	000644			CLR	TEMP2		;CLEAR BUFFER
3273	013176	012701	000001			MOV	#1,R1		;SET NUMBER OF CHARACTERS TO INPUT
3274	013202	012702	000007			MOV	#7,R2		;SET MAXIMUM LIMIT
3275	013206	012703	000000			MOV	#0,R3		;SET MINIMUM LIMIT
3276	013212	004737	023346			JSR	PC,TTR		;GO GET DENSITY
3277	013216	005737	000642			TST	TEMP1		;SEE IF HAVE NEW PARAMETER
3278	013222	001407				BEQ	TINP2		;IF NOT: BR
3279	013224	042737	003400	000552		BIC	#3400,UDES		;ELSE CLEAR OLD PARAMETER
3280	013232	012703	000010			MOV	#10,R3		;SET POSITION FACTOR
3281	013236	004737	014152			JSR	PC,TPOS		;GO LOAD DENSITY INTO PROPER POSITION
3282	013242	012704	025552			TINP2: MOV	#MSG34,R4		
3283	013246	104000				TTOUTT			;PRINT PARITY REQUEST
3284	013250	005037	000644			CLR	TEMP2		;CLR BUFFER
3285	013254	012701	000001			MOV	#1,R1		;SET NUMBER OF CHARACTERS TO INPUT
3286	013260	012702	000001			MOV	#1,R2		;SET MAXIMUM LIMIT
3287	013264	012703	000000			MOV	#0,R3		;SET MINIMUM LIMIT











```
3513 ;ALL ONES*****
3514
3515 014556 012701 177777 DAT1: MOV #-1,R1 ;R1=DATA
3516 014562 012702 002002 DAT1A: MOV #2002,R2 ;R2=WORD COUNT +2
3517 014566 010123 DAT1B: MOV R1,(R3)+ ;LOAD BUFFER
3518 014570 005302 DEC R2 ;SEE IF DONE
3519 014572 001375 BNE DAT1B ;IF NOT: BR
3520 014574 000207 RTS PC ;++G RETURN
3521
3522 ;ALL ZEROS*****
3523
3524 014576 005001 DAT2: CLR R1 ;R1=DATA
3525 014600 000770 BR DAT1A ;++G LOAD BUFFER
3526
3527 ;WALKING ONE*****
3528
3529 014602 012701 000001 DAT3: MOV #1,R1 ;R1=DATA
3530 014606 000241 CLC
3531 014610 012702 004004 DAT3A: MOV #4004,R2 ;R2=CHARACTER COUNT+4
3532 014614 110123 DAT3B: MOV R1,(R3)+ ;LOAD BUFFER
3533 014616 106101 ROLB R1 ;SET NEXT CHARACTER
3534 014620 005302 DEC R2 ;SEE IF DONE
3535 014622 001374 BNE DAT3B ;IF NOT: BR
3536 014624 000207 RTS PC ;++G RETURN
3537
3538 ;WALKING ZERO*****
3539
3540 014626 012701 000376 DAT4: MOV #376,R1 ;R1=START OF DATA
3541 014632 000261 SEC
3542 014634 000765 BR DAT3A ;++G LOAD BUFFER
3543
3544 ;ALTERNATING ONE/ZERO*****
3545
3546
3547 014636 012701 052525 DAT5: MOV #52525,R1 ;R1=DATA
3548 014642 000747 BR DAT1A ;++G LOAD BUFFER
3549
3550 ;ALTERNATING ZERO/ONE*****
3551
3552 014644 012701 125252 DAT6: MOV #125252,R1 ;R1=DATA
3553 014650 000744 BR DAT1A ;++G LOAD BUFFER
3554
3555 ;ONE/ZERO IN ALTERNATING WORDS*****
3556
3557 014652 012701 125252 DAT7: MOV #25252,R1 ;SET WORD 1
3558 014656 012702 052525 MOV #52525,R2 ;SET WORD 2
3559 014662 012704 001002 MOV #1002,R4 ;SET NUMBER OF ENTRIES
3560 014666 010123 DAT7A: MOV R1,(R3)+ ;LOAD WORD 1
3561 014670 010223 MOV R2,(R3)+ ;LOAD WORD 2
3562 014672 005304 DEC R4 ;SEE IF DONE
3563 014674 001374 BNE DAT7A ;IF NOT: BR
3564 014676 000207 RTS PC ;++G RETURN
3565
```

CZTUAM  
CZTUAM  
4132  
4133  
4134  
4135  
4136  
4137  
4138  
4139  
4140  
4141  
4142  
4143  
4144  
4145  
4146  
4147  
4148  
4149  
4150  
4151  
4152  
4153  
4154  
4155  
4156  
4157  
4158  
4159  
4160  
4161  
4162  
4163  
4164  
4165  
4166  
4167  
4168  
4169  
4170  
4171  
4172  
4173  
4174  
4175  
4176  
4177  
4178  
4179  
4180  
4181  
4182  
4183  
4184  
4185  
4186  
4187  
4188

```
3566 ;WALKING ONE/ALL ONE IN ALTERNATING CHARS****
3567
3568 014700 012702 002002 DAT10: MOV #2002,R2 ;SET BUFFER SIZE
3569 014704 012701 000001 MOV #1,R1 ;SET WALK BASE
3570 014710 000241 CLC
3571 014712 012713 177400 DAT10A: MOV #177400,(R3) ;LOAD ALL ONE BYTE
3572 014716 050123 BIS R1,(R3)+ ;LOAD WALK BYTE
3573 014720 106101 ROLB R1 ;WALK ONE
3574 014722 005302 DEC R2
3575 014724 001372 BNE DAT10A ;DO FULL BUFFER
3576 014726 000207 RTS PC ;++G RETURN
3577
3578 ;ALL BITS 0-377*****
3579
3580 014730 005001 DAT11: CLR R1 ;R1=STARTING DATA
3581 014732 012702 004004 MOV #4004,R2 ;R2=CHARACTER COUNT+4
3582 014736 110123 DAT11A: MOV R1,(R3)+ ;LOAD BUFFER
3583 014740 105201 INCB R1 ;BUMP DATA
3584 014742 005302 DEC R2 ;SEE IF DONE
3585 014744 001374 BNE DAT11A ;IF NOT: BR
3586 014746 000207 RTS PC ;++G RETURN
3587
3588 ;ALL BITS 377-0*****
3589
3590 014750 012701 000377 DAT12: MOV #377,R1 ;R1=STARTING DATA
3591 014754 012702 004004 MOV #4004,R2 ;R2=CHARACTER COUNT+4
3592 014760 110123 DAT12A: MOV R1,(R3)+ ;LOAD BUFFER
3593 014762 105301 DECB R1 ;BUMP DATA
3594 014764 005302 DEC R2 ;SEE IF DONE
3595 014766 001374 BNE DAT12A ;IF NOT: BR
3596 014770 000207 RTS PC ;++G RETURN
3597
3598 ;ALTERNATING CHARACTERS 0 AND 377*****
3599
3600 014772 012701 000377 DAT13: MOV #377,R1 ;R1 = DATA
3601 014776 000137 014562 JMP DAT1A ;LOAD BUFFER
3602
3603 ;WALKING ZERO/ALL ZERO IN ALTERNATING CHARS*****
3604
3605 015002 012702 002002 DAT14: MOV #2002,R2 ;SET BUFFER SIZE
3606 015006 012701 000376 MOV #376,R1 ;SET WALK BASE
3607 015012 000261 SEC
3608 015014 010113 DAT14A: MOV R1,(R3) ;LOAD WALK BYTE
3609 015016 042723 177400 BIC #177400,(R3)+ ;CLEAR HIGH BYTE
3610 015022 106101 ROLB R1 ;WALK ZERO BIT
3611 015024 005302 DEC R2
3612 015026 001372 BNE DAT14A ;FILL BUFFER
3613 015030 000207 RTS PC ;++G RETURN
3614
```



```
3615                                     ;AUTO SEQUENCE PATTERN*****
3616
3617 015032 012702 000200      DAT15:  MOV    #200,R2          ;SET NUMBER OF ENTRIES
3618 015036 012701 015062      DAT15A: MOV    #APATS,R1       ;SET START OF PATTERN
3619 015042 012704 000010      MOV    #10,R4                ;SET SIZE OF PATTERN
3620 015046 012123              DAT15B: MOV    (R1)+,(R3)+     ;FILL BUFFER
3621 015050 005304              DEC    R4                    ;SEE IF DONE PATTERN
3622 015052 001375              BNE    DAT15B                ;IF NOT: BR
3623 015054 005302              DEC    R2                    ;SEE IF DONE BUFER
3624 015056 001367              BNE    DAT15A                ;IF NOT: BR
3625 015060 000207              RTS    PC                    ;++G RETURN
3626 015062 000000      APATS:  0
3627 015064 177400              177400
3628 015066 000377              377
3629 015070 000000              0
3630 015072 177777              -1
3631 015074 000377              377
3632 015076 177400              177400
3633 015100 177777              -1
3634
3635                                     ;RANDOM DATA GENERATOR SUBROUTINE*****
3636
3637 015102 013704 000556      DATR:   MOV    FMCNT,R4        ;SET NUMBER OF FRAMES
3638 015106 012703 027456      MOV    #WDATA,R3            ;SET ADDRESS OF START OF BUFFER
3639 015112 012701 177777      MOV    #-1,R1               ;SET HIGH LIMIT
3640 015116 005002              CLR    R2                    ;SET LOW LIMIT
3641 015120 004737 023314      DATRO:  JSR    PC,RANG         ;GO GENERATE NUMBER
3642 015124 013723 000626      MOV    RANSV,(R3)+          ;LOAD BUFFER
3643 015130 005204              INC    R4                    ;SEE IF DONE WHOLE BUFFER
3644 015132 001372              BNE    DATRO                 ;IF NOT: BR
3645 015134 004737 014316      JSR    PC,DS1                ;GO CHECK FOR 7 CH
3646 015140 012737 000001 015150  MOV    #1,RDFL              ;SET RANDOM DATA FLAG
3647 015146 000207              RTS    PC                    ;EXIT
3648 015150 000000      RDFL:   0                    ;RANDOM DATA SELECT FLAG
```

```
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658 015152 013700 000556
3659 015156 005400
3660 015160 012701 027456
3661 015164 005037 015534
3662 015170 111104
3663 015172 004737 015362
3664 015176 004737 015510
3665 015202 000241
3666 015204 006004
3667 015206 103014
3668 015210 052704 000400
3669 015214 000241
3670 015216 010405
3671 015220 042705 177703
3672 015224 005105
3673 015226 042705 177703
3674 015232 042704 000074
3675 015236 050504
3676 015240 010437 015534
3677 015244 005300
3678 015246 001401
3679 015250 000747
3680 015252 013704 015534
3681 015256 005137 015534
3682 015262 042737 177050 015534
3683 015270 042704 177727
3684 015274 050437 015534
3685 015300 013737 015534 015536
3686 015306 013700 000556
3687 015312 005400
3688 015314 012701 027456
3689 015320 005037 015534
3690 015324 111104
3691 015326 004737 015362
3692 015332 004737 015510
3693 015336 005300
3694 015340 001371
3695 015342 013704 015536
3696 015346 004737 015510
3697 015352 013737 015534 015540
3698 015360 000207
3699 015362 005704
3700 015364 001010
3701 015366 032737 000010 000552
3702 015374 001404
3703 015376 012704 000420
3704 015402 005201

;*****
;CRC/LRC CHARACTER BUILD;
;
;THIS ROUTINE WILL CONSTRUCT AND SAVE THE EXPECTED
;CRC AND LRC CHARACTERS ACCORDING TO DATA AND
;RECORD SIZE IF OPERATING IN NRZ MODE
;*****

CRCLRC: MOV FMCNT,R0 ;SET RECORD SIZE
        NEG R0
        MOV #WDATA,R1 ;SET START OF BUFFER
        CLR XORS
CLO:    MOVB (R1),R4 ;GET CHARACTER
        JSR PC,CLP ;GO GET PARITY OF CHARACTER
        JSR PC,XOR ;XOR CHARACTER
        CLC
        ROR R4 ;ROTATE 1 RIGHT
        BCC CL2 ;IF NO CARRY: BR
        BIS #400,R4 ;SET BIT NINE
        CLC
CL1:    MOV R4,R5 ;SAVE CHARACTER
        BIC #177703,R5
        COM R5
        BIC #177703,R5
        BIC #74,R4
        BIS R5,R4 ;COMPLIMENT BITS 2,3,4,5
CL2:    MOV R4,XORS
        DEC R0
        BEQ CLLAST ;IF LAST CHARACTER: BR
        BR CLO ;++G GET NEXT
CLLAST: MOV XORS,R4
        COM XORS
        BIC #177050,XORS
        BIC #177727,R4 ;COMPLIMENT ALL BUT BITS 3&5
        BIS R4,XORS
        MOV XORS,EXCRC ;SAVE EXPECTED CRC
        MOV FMCNT,R0
        NEG R0
        MOV #WDATA,R1 ;DO EXPT LRC
        CLR XORS
CL3:    MOVB (R1),R4
        JSR PC,CLP ;GET PARITY
        JSR PC,XOR ;XOR CHARACTER
        DEC R0
        BNE CL3 ;DO ALL FOR LRC
        MOV EXCRC,R4
        JSR PC,XOR ;XOR CRC TO DATA
        MOV XORS,EXLRC ;SAVE EXPT LRC
        RTS PC ;RETURN
CLP:    TST R4 ;SEE IF 0 CHAR
        BNE CLPE ;IF NOT: BR
        BIT #10,UDES ;SEE IF EVEN PARITY
        BEQ CLPE ;IF NOT: BR
        MOV #420,R4 ;SET 0 CHAR EVEN PARITY
        INC R1 ;BUMP POINTER
```

```
3705 015404 000207          RTS      PC      ;RETURN
3706 015406 005037 015542    CLPE:    CLR      PARCNT ;CLEAR BIT COUNTER
3707 015412 012703 000010          MOV      #10,R3  ;SET NUMBER OF BITS
3708 015416 032704 000001    CLP0:    BIT      #1,R4  ;SEE IF ONE BIT
3709 015422 001402          BEQ      CLP1    ;IF NOT: BR
3710 015424 005237 015542          INC      PARCNT  ;BUMP COUNTER
3711 015430 000241          CLP1:    CLC          ;
3712 015432 006004          ROR      R4      ;ROTATE TO NEXT BIT
3713 015434 005303          DEC      R3      ;
3714 015436 001367          BNE      CLP0    ;CONTINUE FOR ALL BITS
3715 015440 112104          MOVB    (R1)+,R4 ;
3716 015442 042704 177400          BIC     #177400,R4 ;
3717 015446 032737 000001 015542    BIT     #1,PARCNT ;SEE IF ODD NUMBER OF ONE BITS
3718 015454 001005          BNE     CLP2    ;IF SO: BR
3719 015456 032737 000010 000552    BIT     #10,UDES ;SEE IF SHOULD BE EVEN PARITY
3720 015464 001406          BEQ     CLP3    ;IF NOT: BR
3721 015466 000207          RTS     PC      ;ELSE EXIT
3722 015470 032737 000010 000552    CLP2:   BIT     #10,UDES ;SEE IF SHOULD BE ODD PARITY
3723 015476 001001          BNE     CLP3    ;IF NOT: BR
3724 015500 000207          RTS     PC      ;ELSE EXIT
3725 015502 052704 000400    CLP3:   BIS     #400,R4 ;SET PARITY BIT
3726 015506 000207          RTS     PC      ;
3727 015510 010446          XOR:    MOV     R4,-(SP) ;
3728 015512 043716 015534          BIC     XORS,(SP) ;
3729 015516 040437 015534          BIC     R4,XORS  ;XOR SUBROUTINE: R4 WITH XORS
3730 015522 052637 015534          BIS     (SP)+,XORS ;
3731 015526 013704 015534          MOV     XORS,R4 ;
3732 015532 000207          RTS     PC      ;
3733
3734 015534 000000          XORS:   0      ;XOR SAVE
3735 015536 000000          EXCRC:  0      ;EXPECTED CRC
3736 015540 000000          EXLRC:  0      ;EXPECTED LRC
3737 015542 000000          PARCNT: 0      ;PARITY COUNTER
3738
```





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

```

:*****
:DATA ERROR SUBROUTINE:
:
:THIS SUBROUTINE IS USED TO PRINT OUT ANY
:ERRORS FOUND DURING THE DATA CHECK.
:
:EVERY 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.
:*****

```

3872	016160	032777	002000	162420	DERR:	BIT	#2000,@SWR	:SEE IF SHOULD PRINT ERRORS
3873	016166	001067				BNE	DERR4	:++G BRANCH IF NOT
3874	016170	005237	000670		DERRO:	INC	PFLG	:SET PRINT FLAG
3875	016174	005737	000664			TST	HDRFL	:SEE IF HAVE PRINTED HEADER
3876	016200	001007				BNE	DERROA	:IF SO: BR
3877	016202	004737	022730			JSR	PC,PAPRT	:PRINT CYCLE NUMBER
3878	016206	012704	024730			MOV	#MSG1,R4	:LOAD ERROR MSG ADDR
3879	016212	104000				TTOUTT		:PRINT ERROR
3880	016214	004737	021156			JSR	PC,FRPRT	:PRINT F OR R
3881	016220	012704	024747		DERROA:	MOV	#MSG4,R4	
3882	016224	104000				TTOUTT		:PRINT CHAR NO. HEADER
3883	016226	010203				MOV	R2,R3	
3884	016230	162703	033464			SUB	#RDATA,R3	:POINT TO CHAR
3885	016234	005303				DEC	R3	
3886	016236	032737	010000	000562		BIT	#10000,RDCMD	:SEE IF READ REVERSE
3887	016244	001402				BEQ	DERROB	:IF NOT: BR
3888	016246	010503				MOV	R5,R3	:GET CHAR NUMBER
3889	016250	005103				COM	R3	
3890	016252	104002			DERROB:	OCTPP		:PRINT CHAR NUMBER
3891	016254	012704	024735			MOV	#MSG2,R4	
3892	016260	104000				TTOUTT		:PRINT EXPECTED DATA
3893	016262	032737	010000	000562		BIT	#10000,RDCMD	:SEE IF READ REVERSE
3894	016270	001402				BEQ	DERROC	:IF NOT: BR
3895	016272	111103				MOVB	(R1),R3	:GET CHAR
3896	016274	000401				BR	DERROD	
3897	016276	114103			DERROC:	MOVB	-(R1),R3	:LOAD EXPECTED DATA
3898	016300	004737	024160		DERROD:	JSR	PC,DOUT	:GO PRINT CHAR
3899	016304	012704	024742			MOV	#MSG3,R4	



```

3956 016622 005777 161760         DEREX:  TST      @SWR           ;++G BRANCH IF CONTINUE ON ERROR
3957 016626 100012                  BPL      DEREX1           ;++G
3958 016630 104006                  STOPP
3959 016632 005737 000670         TST      PFLG            ;SEE IF PRINTED
3960 016636 001006                  BNE      DEREX1          ;IF SO: BR
3961 016640 032777 002000 161740  BIT      #2000,@SWR      ;SEE IF SHOULD PRINT
3962 016646 001002                  BNE      DEREX1          ;IF NOT: BR
3963 016650 000137 016170         JMP      DERRO           ;ELSE PRINT
3964 016654 005037 000670         DEREX1: CLR      PFLG            ;CLEAR FLAG
3965 016660 005237 000704         INC      DERFL           ;BUMP DATA ERROR FLAG
3966 016664 000207                  RTS      PC              ;RETURN
3967

```

CZTU  
 CZTU  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 45  
 46  
 46  
 46  
 46  
 46











```

4188 020052 032777 020000 160526      BIT      #20000,@SWR      ;SEE IF NO DATA CHECK
4189 020060 001065                      BNE      ERPT        ;IF NOT: BR (ALLOW READ OF UNKNOWN TAPES)
4190 020062 032737 000040 000672      BIT      #40,MTC1    ;SEE IF WRITE OR READ OP
4191 020070 001461                      BEQ      ERPT        ;IF NOT: BR
4192 020072 005737 000676                      TST      TMFLG      ;SEE IF TAPE MARK TIME
4193 020076 001413                      BEQ      ER6A       ;IF NOT: BR
4194 020100 013737 015536 021154      MOV      EXCRC,CRCSV ;SAVE CRC
4195 020106 013737 015540 021152      MOV      EXLRC,LRCV  ;SAVE LRC
4196 020114 005037 015536                      CLR      EXCRC
4197 020120 012737 000023 015540      MOV      #23,EXLRC   ;SET CRC/LRC FOR TM
4198 020126 032737 000060 000552  ER6A:  BIT      #60,UDES    ;SEE IF FORMAT 14
4199 020134 001037                      BNE      ERPT        ;IF NOT: BR
4200 020136 017703 160366                      MOV      @CC,R3      ;GET CRC CHARACTER
4201 020142 042703 177000                      BIC      #177000,R3
4202 020146 023703 015536                      CMP      EXCRC,R3
4203 020152 001402                      BEQ      ER7        ;IF CRC GOOD: BR
4204 020154 005237 021146                      INC      CRCER       ;SET ERROR FLAG
4205 020160 017703 160350                      ER7:  MOV      @MR,R3  ;GET LRC
4206 020164 000303                      SWAB    R3
4207 020166 005703                      TST      R3
4208 020170 100002                      BPL      ER10
4209 020172 052703 000400                      BIS      #400,R3
4210 020176 042703 177000                      ER10:  BIC      #177000,R3
4211 020202 023703 015540                      CMP      EXLRC,R3
4212 020206 001412                      BEQ      ERPT        ;IF LRC GOOD: BR
4213 020210 010337 021150                      MOV      R3,ACTLRC   ;SAVE ACTUAL LRC
4214 020214 005237 021144                      INC      LRCER       ;SET LRC ERROR FLAG
4215 020220 032737 010000 000562      BIT      #10000,RDCMD ;SEE IF READ REVERSE
4216 020226 001402                      BEQ      ERPT        ;IF NOT: BR
4217 020230 005037 021144                      CLR      LRCER       ;ELSE CLEAR LRC ERROR
4218 020234 012703 000006                      ERPT:  MOV      #6,R3
4219 020240 005037 000706                      CLR      SERFL
4220 020244 005037 000722                      CLR      ERSAV      ;CLEAR ERROR FLAG
4221 020250 012704 021134                      ERPTT: MOV      #BAER,R4
4222 020254 005724                      TST      (R4)+      ;SEE IF ANY ERROR
4223 020256 001004                      BNE      ERPTG      ;IF SO: BR
4224 020260 005303                      DEC      R3
4225 020262 001374                      BNE      ERPTT
4226 020264 000137 021076                      JMP      ERPX1
4227 020270 005237 000706                      ERPTG: INC      SERFL ;SET ERRGR FLAG
4228 020274 017737 160224 000722      MOV      @ER,ERSAV  ;SAVE ERROR REGISTER
4229 020302 032777 002000 160276      BIT      #2000,@SWR ;SEE IF PRINT
4230 020310 001420                      BEQ      ERPT0      ;IF SO: BR
4231 020312 022737 000002 000712      CMP      #2,RTYFL   ;SEE IF READ RETRY
4232 020320 001006                      BNE      ERPTG1     ;IF NOT: BR
4233 020322 013703 000702                      MOV      RTCNT,R3
4234 020326 005203                      INC      R3          ;BUMP RETRY COUNT
4235 020330 020337 000602                      CMP      R3,RETRY   ;SEE IF LAST RETRY
4236 020334 001406                      BEQ      ERPT0      ;IF SO: BR
4237 020336 022737 000002 021140  ERPTG1: CMP      #2,DRVER  ;SEE IF TM STATUS ERROR
4238 020344 001402                      BEQ      ERPT0      ;IF SO: BR
4239 020346 000137 021000                      JMP      ERPX0
4240 020352 005237 000670                      ERPT0: INC      PFLG
4241 020356 004737 022730                      JSR      PC,PAPRT   ;PRINT HEADER
4242 020362 013704 000652                      MOV      EMADDR,R4
4243 020366 104000                      TTOUTT             ;PRINT ERROR TYPE
    
```

4244	020370	004737	021156		JSR	PC,FRPRT	:PRINT F OR R
4245	020374	005737	000676		TST	TMFLG	
4246	020400	001407			BEQ	ERPT1	
4247	020402	022737	026164	000652	CMP	#MSG54,EMADDR	
4248	020410	001403			BEQ	ERPT1	
4249	020412	012704	026202		MOV	#MSG56,R4	:PRINT TM
4250	020416	104000			TTOUTT		
4251	020420	005737	021136	ERPT1:	TST	CONER	
4252	020424	001414			BEQ	ERPT2	:IF NO CONT ERROR: BR
4253	020426	012704	025205		MOV	#MSG23,R4	
4254	020432	104000			TTOUTT		:PRINT C1 TAG
4255	020434	017703	160050		MOV	@C1,R3	
4256	020440	104002			OCTPP		:PRINT CONTROL 1
4257	020442	012704	025232		MOV	#MSG23D,R4	:PRINT CS TAG
4258	020446	104000			TTOUTT		
4259	020450	017703	160044		MOV	@CS,R3	
4260	020454	104002			OCTPP		:PRINT CONT STATUS
4261	020456	005737	021140	ERPT2:	TST	DRVER	
4262	020462	001414			BEQ	ERPT3	:IF SO DRIVE ERROR: BR
4263	020464	012704	025240		MOV	#MSG23E,R4	
4264	020470	104000			TTOUTT		:PRINT DS TAG
4265	020472	017703	160024		MOV	@DS,R3	
4266	020476	104002			OCTPP		:PRINT DRIVE STATUS
4267	020500	012704	025245		MOV	#MSG23F,R4	
4268	020504	104000			TTOUTT		:PRINT ER TAG
4269	020506	017703	160012		MOV	@ER,R3	
4270	020512	104002			OCTPP		:PRINT DRIVE ERROR
4271	020514	005737	021134	ERPT3:	TST	BAER	
4272	020520	001416			BEQ	ERPT4	:IF NO BA ERROR: BR
4273	020522	012704	025220		MOV	#MSG23B,R4	
4274	020526	104000			TTOUTT		:PRINT BA TAG
4275	020530	017703	157760		MOV	@BA,R3	
4276	020534	104002			OCTPP		:PRINT BUS ADDRESS
4277	020536	012737	000255	000636	MOV	#255,TOB	
4278	020544	004737	023662		JSR	PC,TOG	:PRINT /
4279	020550	013703	021132		MOV	CADER,R3	
4280	020554	104002			OCTPP		:PRINT EXPT BUS ADDRESS
4281	020556	005737	021142	ERPT4:	TST	FCER	
4282	020562	001406			BEQ	ERPT5	:IF NO FC ERROR: BR
4283	020564	012704	025225		MOV	#MSG23C,R4	
4284	020570	104000			TTOUTT		:PRINT FC TAG
4285	020572	017703	157720		MOV	@FC,R3	
4286	020576	104002			OCTPP		:PRINT FRAME COUNT
4287	020600	012704	025213	ERPT5:	MOV	#MSG23A,R4	
4288	020604	104000			TTOUTT		:PRINT WC TAG
4289	020606	017703	157700		MOV	@WC,R3	
4290	020612	104002			OCTPP		:PRINT WORD COUNT
4291	020614	005737	021146		TST	CRCER	
4292	020620	001420			BEQ	ERPT5A	:IF NO CRC ERROR: BR
4293	020622	012704	026227		MOV	#MSG58,R4	
4294	020626	104000			TTOUTT		:PRINT CRC TAG
4295	020630	017703	157674		MOV	@CC,R3	
4296	020634	042703	177000		BIC	#177000,R3	
4297	020640	104002			OCTPP		:PRINT ACTUAL CRC
4298	020642	012737	000255	000636	MOV	#255,TOB	
4299	020650	004737	023662		JSR	PC,TOG	

4300	020654	013703	015536		MOV	EXCRC,R3	
4301	020660	104002			OCTPP		;PRINT EXPECTED CRC
4302	020662	005737	021144	ERPT5A:	TST	LRCER	
4303	020666	001416			BEQ	ERPT6	;IF NO LRC ERROR: BR
4304	020670	012704	026235		MOV	#MSG59,R4	
4305	020674	104000			TTOUTT		;PRINT LRC TAG
4306	020676	013703	021150		MOV	ACTLRC,R3	
4307	020702	104002			OCTPP		;PRINT ACTUAL LRC
4308	020704	012737	000255	000636	MOV	#255,TOB	
4309	020712	004737	023662		JSR	PC,TOG	
4310	020716	013703	015540		MOV	EXLRC,R3	
4311	020722	104002			OCTPP		;PRINT EXPECTED LRC
4312	020724	005737	021140	ERPT6:	TST	DRVER	
4313	020730	001422			BEQ	ERPT7	;IF NO DRIVE ERROR: BR
4314	020732	032737	002000	000552	BIT	#2000,UDES	
4315	020740	001416			BEQ	ERPT7	;IF NO PE: BR
4316	020742	017704	157556		MOV	@ER,R4	
4317	020746	042704	075477		BIC	#75477,R4	;MASK OUT ALL BUT BITS 15,10,7,6
4318	020752	005704			TST	R4	
4319	020754	001410			BEQ	ERPT7	;IF NO CONDITIONALS SET: BR
4320	020756	012704	025257		MOV	#MSG23H,R4	
4321	020762	104000			TTOUTT		;PRINT CC TAG
4322	020764	017703	157540		MOV	@CC,R3	
4323	020770	042703	177000		BIC	#177000,R3	;MASK CC
4324	020774	104002			OCTPP		;PRINT CHECK CHARACTERS
4325	020776	000240		ERPT7:	NOP		
4326	021000	032777	100000	157600	ERPX0:	BIT	#100000,@SWR
4327	021006	001412			BEQ	ERPX	;SEE IF STOP ON ERROR
4328	021010	104006			STOPP		;IF NOT: BR
4329	021012	005737	000670		TST	PFLG	;SEE IF HAVE PRINTED
4330	021016	001006			BNE	ERPX	;IF SO: BR
4331	021020	032777	002000	157560	BIT	#2000,@SWR	;SEE IF SHOULD PRINT
4332	021026	001002			BNE	ERPX	;IF NOT: BR
4333	021030	000137	020352		JMP	ERPT0	;PRINT ERROR
4334	021034	005037	000670	ERPX:	CLR	PFLG	
4335	021040	012777	000011	157442	MOV	#11,@C1	;DRIVE CLEAR
4336	021046	017704	157454		MOV	@AS,R4	
4337	021052	010477	157450		MOV	R4,@AS	;CLEAR AS
4338	021056	013704	000510		MOV	C1,R4	
4339	021062	005204			INC	R4	
4340	021064	152714	000100		BISB	#100,(R4)	;RESET TRE
4341	021070	013777	000552	157444	MOV	UDES,@C2	;RESET TC
4342	021076	032737	000040	000672	ERPX1:	BIT	#40,MTC1
4343	021104	001411			BEQ	ERPX2	;IF NOT READ/WRITE OP: BR
4344	021106	005737	000676		TST	TMFLG	
4345	021112	001406			BEQ	FRPX2	;IF NOT TM TIME: BR
4346	021114	013737	021154	015536	MOV	CRCSV,EXCRC	;RESTORE CRC
4347	021122	013737	021152	015540	MOV	LRCSV,EXLRC	;RESTORE LRC
4348	021130	000207		ERPX2:	RTS	PC	;EXIT
4349	021132	000000		CADER:	0		;EXPT ADDRESS SAVE
4350	021134	000000		BAER:	0		
4351	021136	000000		CONER:	0		
4352	021140	000000		DRVER:	0		
4353	021142	000000		FCER:	0		
4354	021144	000000		LRCER:	0		
4355	021146	000000		CRCER:	0		

4356 021150 000000 ACTLRC: 0  
4357 021152 000000 LRCSV: 0  
4358 021154 000000 CRCSV: 0  
4359

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

4368  
4369 021156 032737 000010 000672 FRPRT: BIT #10,MTC1 ;SEE IF WRITE COMMAND  
4370 021164 001413 BEQ FREX ;IF SO: BR  
4371 021166 032737 000002 000672 BIT #2,MTC1 ;SEE IF REVERSE  
4372 021174 001404 BEQ FRO ;IF NOT: BR  
4373 021176 012704 025115 MOV #MSG17,R4  
4374 021202 104000 TTOUTT ;PRINT R  
4375 021204 000403 BR FREX  
4376 021206 012704 025112 FRO: MOV #MSG16,R4  
4377 021212 104000 TTOUTT ;PRINT F  
4378 021214 000207 FREX: RTS PC ;EXIT  
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

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

```
4409 021216 005037 000642 TAPG: CLR TEMP1  
4410 021222 013777 000550 157270 MOV DVN,@CS ;SET DRIVE NO.  
4411 021230 032777 010000 157264 TAPG0: BIT #10000,@DS ;SEE IF HAVE MOL  
4412 021236 001026 BNE TAPG3 ;IF SO: BR  
4413 021240 005237 000642 INC TEMP1 ;SEE IF TIMED OUT  
4414 021244 001371 BNE TAPG0 ;WAIT FOR READY  
4415 021246 004737 022730 JSR PC,PAPRT ;PRINT CYCLE NUMBER  
4416 021252 032737 000010 000672 BIT #10,MTC1 ;SEE IF WRITE OP  
4417 021260 001004 BNE TAPG1 ;IF NOT: BR  
4418 021262 012704 024754 MOV #MSG5,R4  
4419 021266 104000 TTOUTT ;PRINT WRITE ERR  
4420 021270 000405 BR TAPG2  
4421 021272 012704 024761 TAPG1: MOV #MSG6,R4  
4422 021276 104000 TTOUTT ;PRINT READ ERR  
4423 021300 004737 021156 JSR PC,FRPRT ;PRINT F OR R  
4424 021304 012704 025335 TAPG2: MOV #MSG25,R4  
4425 021310 104000 TTOUTT ;PRINT NO MOL ERR  
4426 021312 104006 STOPP  
4427 021314 032777 020000 157200 TAPG3: BIT #20000,@DS ;SEE IF PIP RESET  
4428 021322 001411 BEQ TAPG3F ;IF SO: BR  
4429 021324 004737 022730 JSR PC,PAPRT ;PRINT HEADER  
4430 021330 012704 027314 MOV #MSG116,R4  
4431 021334 104000 TTOUTT ;PRINT REWINDING MESSAGE  
4432 021336 032777 020000 157156 1$: BIT #20000,@DS  
4433 021344 001374 BNE 1$ ;AWAIT PIP RESET  
4434 021346 022737 000026 000672 TAPG3F: CMP #26,MTC1 ;SEE IF WRITE TM  
4435 021354 001003 BNE TAPG3A ;IF NOT: BR
```

```
4436 021356 012704 177777          MOV    #-1,R4          ;ELSE SET FC FOR -1
4437 021362 000406          BR     TAPG3B
4438 021364 013704 000556    TAPG3A: MOV   FMCNT,R4
4439 021370 032704 000001          BIT    #1,R4
4440 021374 001401          BEQ   TAPG3B
4441 021376 005304          DEC   R4
4442 021400 000261    TAPG3B: SEC
4443 021402 006004          ROR   R4          ;SET WC = FC/2 FOR NORMAL FORMAT
4444 021404 032737 000020 000552    BIT    #20,UDES      ;SEE IF CORE DUMP FORMAT
4445 021412 001402          BEQ   TAPG3C      ;IF NOT: BR
4446 021414 000261          SEC
4447 021416 006004          ROR   R4          ;SET WC = FC/4 FOR CORE DUMP
4448 021420 010477 157066    TAPG3C: MOV   R4,@WC  ;SET WORD COUNT
4449 021424 012777 000011 157056    MOV   #11,@C1      ;DRIVE CLEAR
4450 021432 017777 157060 157056    MOV   @FC,@FC      ;RESET FC LOADED
4451 021440 005737 000566          TST   INTRF        ;SEE IF INTERCHANGE READ
4452 021444 001407          BEQ   TAPG3D      ;IF NOT: BR
4453 021446 032737 000040 000672    BIT    #40,MTC1     ;SEE IF READ OP
4454 021454 001403          BEQ   TAPG3D      ;IF NOT: BR
4455 021456 012777 000003 157050    MOV   #3,@MR       ;SET INTERCHANGE READ MAINT. MODE
4456 021464 013704 000672    TAPG3D: MOV   MTC1,R4 ;GET COMMAND
4457 021470 042704 177707          BIC   #177707,R4   ;MASK OP CODE
4458 021474 022704 000030          CMP   #30,R4       ;SEE IF SPACE OP CODE
4459 021500 001403          BEQ   TAPG3E      ;IF SO: BR
4460 021502 012737 177740 000666    MOV   #-40,STAL    ;SET INTERRUPT DELAY MULT TO 40
4461 021510 052737 000101 000672    TAPG3E: BIS   #101,MTC1 ;SET INTERRUPT ENABLE AND GO
4462 021516 000240          NOP
4463 021520 013777 000672 156762    MOV   MTC1,@C1     ;EXECUTE COMMAND
4464 021526 005077 157052          CLR   @PSW         ;CLEAR PRIORITY
4465 021532 005037 000642          CLR   TEMP1
4466 021536 005237 000642    TAPG4: INC   TEMP1   ;SEE IF HAVE TIMED OUT
4467 021542 001375          BNE   TAPG4       ;IF NOT: BR
4468 021544 005237 000666          INC   STAL
4469 021550 001372          BNE   TAPG4       ;DO TIME DELAY MULTIPLIER
4470 021552 012777 000340 157024    TAPG5: MOV   #340,@PSW ;RESET PRIORITY
4471 021560 032777 002000 157020    BIT    #2000,@SWR  ;SEE IF SHOULD PRINT ERRORS
4472 021566 001012          BNE   TAPG6       ;IF NOT: BR
4473 021570 004737 022730          JSR   PC,PAPRT    ;PRINT CYCLE NUMBER
4474 021574 013704 000652          MOV   EMADDR,R4
4475 021600 104000          TTOUTT           ;PRINT ERROR OP
4476 021602 004737 021156          JSR   PC,FRPRT    ;PRINT F OR R
4477 021606 012704 025315          MOV   #MSG24,R4
4478 021612 104000          TTOUTT           ;PRINT NO INTERRUPT
4479 021614 005777 156766    TAPG6: TST   @SWR   ;++G BRANCH IF CONTINUE ON ERROR
4480 021620 100001          BPL   TAPG7
4481 021622 104006          STOPP
4482 021624 000137 021716    TAPG7: JMP   MTINTA  ;RETURN TO CALLING ROUTINE
4483
```



```
4511 ;*****  
4512 ;AUTO SEQUENCE  
4513 ;  
4514 ;THIS ROUTINE ,ENTERED VIA STARTING ADDRESS 240  
4515 ;WILL EXERCISE ALL AVAILABLE SLAVES ON ALL AVAILABLE  
4516 ;DRIVES IN BOTH PE AND NRZ ACCORDING TO THE PRESELECTED  
4517 ;TEST PLAN. IF NRZ ONLY, PE TESTING WILL NOT BE ATTEMPTED.  
4518 ;*****  
4519 ;  
4520 021732 012704 027124 ASEQ: MOV #MSG108,R4  
4521 021736 104000 TTOUTT ;PRINT NRZ ONLY REQUEST  
4522 021740 012705 000650 MOV #NRZOF,R5 ;SET ADDRESS OF FLAG  
4523 021744 012701 000001 MOV #1,R1 ;SET SIZE OF ENTRY  
4524 021750 012702 000001 MOV #1,R2 ;SET UPPER LIMIT  
4525 021754 012703 000000 MOV #0,R3 ;SET LOWER LIMIT  
4526 021760 004737 023346 JSR PC,TTR ;GO GET RESPONSE  
4527 021764 012704 026721 MOV #MSG104,R4  
4528 021770 104000 TTOUTT ;REQUEST CONT OR NOT  
4529 021772 012705 000742 MOV #ASEQCF,R5 ;SET ADDRESS OF ENTRY  
4530 021776 012701 000001 MOV #1,R1 ;SET SIZE OF ENTRY  
4531 022002 012702 000001 MOV #1,R2 ;SET UPPER LIMIT  
4532 022006 012703 000000 MOV #0,R3 ;SET LOWER LIMIT  
4533 022012 004737 023346 JSR PC,TTR ;GO GET INPUT  
4534 022016 005037 000736 ASEQ0: CLR ADRVN ;CLEAR DRV NUM  
4535 022022 004737 022154 ASEQ1: JSR PC,HRDS ;GO SELECT HARDWARE CONFIGURATION  
4536 022026 005737 000042 TST @#42 ;AUTO MODE? ++ C.W  
4537 022032 001404 BEQ 1$ ;BRANCH - IF NO ++ C.W  
4538 022034 012737 000001 000742 MOV #1,ASEQCF ;SET AUTO SEQ FLAG ++ C.W  
4539 022042 000414 BR 2$ ;DO AUTO SEQ TESTS ++ C.W  
4540 022044 012704 026665 1$: MOV #MSG101,R4  
4541 022050 104000 TTOUTT ;PRINT DIVIDER  
4542 022052 012704 026701 MOV #MSG102,R4  
4543 022056 104000 TTOUTT ;PRINT TMO2 NUMBER  
4544 022060 013703 000736 MOV ADRVN,R3  
4545 022064 104002 OCTPP ;PRINT TMO2  
4546 022066 012704 026710 MOV #MSG103,R4  
4547 022072 104000 TTOUTT ;PRINT SLAVE HDR  
4548 022074 012700 000746 2$: MOV #UN1,R0 ;POINT TO START OF SLAVE TABLE  
4549 022100 005710 ASEQ2: TST (R0) ;SEE IF END  
4550 022102 100403 BMI ASEQ3 ;IF SO: BR  
4551 022104 012003 MOV (R0)+,R3  
4552 022106 104002 OCTPP ;PRINT SLAVE TABLE  
4553 022110 000773 BR ASEQ2 ;DO ALL  
4554 022112 004737 022360 ASEQ3: JSR PC,AMOD1 ;GO DO MODE 1(NRZ)  
4555 022116 004737 022554 JSR PC,AMOD2 ;GO DO MODE 2(PE)  
4556 022122 022737 000007 000736 ASEQ4: CMP #7,ADRVN ;SEE IF DONE ALL DRIVES  
4557 022130 001403 BEQ ASEQX ;IF SO: BR  
4558 022132 005237 000736 INC ADRVN ;BUMP DRIVE NUMBER  
4559 022136 000731 BR ASEQ1 ;CONTINUE  
4560 022140 005737 000742 ASEQX: TST ASEQCF ;CONTINUOUS AUTO SEQUENCE? ++ C.W  
4561 022144 001402 BEQ 1$ ;BRANCH - IF NO ++ C.W  
4562 022146 000137 005100 JMP TEND ;DO ACT END OF PASS ++ C.W  
4563 022152 000000 1$: HALT
```

```
4564  
4565 ;SUBROUTINE TO SELECT AUTO SEQUENCE HARDWARE*****  
4566  
4567 022154 005037 005156 HRDS: CLR REOTC ;CLEAR EOT UNIT CNTR  
4568 022160 005037 000642 CLR TEMP1  
4569 022164 012777 000040 156326 MOV #40,@CS ;INIT  
4570 022172 013777 000736 156320 MOV ADRVN,@CS ;SET DRIVE  
4571 022200 017701 156332 MOV @DT,R1 ;READ DRIVE TYPE  
4572 022204 032777 010000 156306 BIT #10000,@CS ;TEST FOR NON-EXISTANT DRIVE  
4573 022212 001403 BEQ HRDS1 ;IF DRIVE AVAIL: BR  
4574 022214 005726 HRDS0: TST (SP)+ ;RESET STACK POINTER  
4575 022216 000137 022122 JMP ASEQ4 ;GO SEE IF TRIED ALL DRIVES  
4576 022222 042701 002007 HRDS1: BIC #2007,R1 ;MASK SLAVE TYPE  
4577 022226 022701 140010 CMP #140010,R1 ;++G SEE IF TU16/TE16 TAPE  
4578 022232 001370 BNE HRDS0 ;IF NOT: BR  
4579 022234 005000 CLR RO  
4580 022236 012701 000746 MOV #UN1,R1 ;SET START OF SLAVE TABLE  
4581 022242 005737 003040 TST CHNFLG ;++G BRANCH IF NOT IN CHAIN MODE  
4582 022246 001410 BEQ HRDS2  
4583 022250 122737 000006 000041 CMPB #6,@#41 ;++G BRANCH IF NOT LOADED VIA TMDP  
4584 022256 001004 BNE HRDS2  
4585 022260 005737 000736 TST ADRVN ;++G BRANCH IF NOT DRIVE 0  
4586 022264 001001 BNE HRDS2 ;++G  
4587 022266 005200 INC RO ;++G DO NOT TEST DRIVE 0 SLAVE 0  
4588 ;++G IF TMDP CHAIN  
4589 022270 010077 156246 HRDS2: MOV RO,@C2 ;SELECT SLAVE  
4590 022274 032777 010000 156220 BIT #10000,@DS ;SEE IF SLAVE AVAIL FOR TEST(MOL)  
4591 022302 001403 BEQ HRDS3 ;IF NOT: BR  
4592 022304 005237 000642 INC TEMP1 ;SET SLAVE FOUND FLAG  
4593 022310 010021 MOV RO,(R1)+ ;LOAD SLAVE TABLE  
4594 022312 022700 000007 HRDS3: CMP #7,RO ;SEE IF DONE ALL SLAVES  
4595 022316 001402 BEQ HRDS4 ;IF SO: BR  
4596 022320 005200 INC RO ;ELSE BUMP SLAVE NUMBER  
4597 022322 000762 BR HRDS2 ;CONTINUE SELECTION  
4598 022324 005737 000642 HRDS4: TST TEMP1 ;SEE IF FOUND ANY SLAVES  
4599 022330 001731 BEQ HRDS0 ;IF NOT: BR  
4600 022332 013737 000642 005156 MOV TEMP1,REOTC ;SET NUMBER OF UNITS  
4601 022340 000337 000642 SWAB TEMP1  
4602 022344 053737 000642 005156 BIS TEMP1,REOTC ;SET EOT CNTR  
4603 022352 012711 177777 MOV #-1,(R1) ;TERMINATE SLAVE TABLE  
4604 022356 000207 RTS PC ;RETURN TO SEQ
```











4764  
4765  
4766  
4767  
4768  
4769  
4770  
4771  
4772  
4773  
4774  
4775  
4776  
4777  
4778  
4779  
4780

```
*****  
:RANDOM NUMBER GENERATOR SUBROUTINE:  
:  
:THIS SUBROUTINE IS USED TO GENERATE THE RANDOM  
:NUMBERS REQUIRED FOR USE AS RANDOM DATA,  
:RECORD COUNT, AND CHARACTER COUNT.  
:*****  
ADD RANSAV,RANBAS  
ADD RANBAS,RANSAV ;GET NEW NUMBER  
CMP RANSAV,R1 ;SEE IF NUMBER TOO BIG  
BHI RANG ;IF SO: BR  
CMP R2,RANSAV ;SEE IF NUMBER TOO SMALL  
BHI RANG ;IF SO: BR  
RTS PC ;EXIT
```

023314 063737 000626 000624 RANG:  
023322 063737 000624 000626  
023330 023701 000626  
023334 101367  
023336 020237 000626  
023342 101364  
023344 000207

```
4781 ;*****  
4782 ;TTY ENTRY SUBROUTINE:  
4783 ;  
4784 ;THIS SUBROUTINE IS USED BY THE TEST CONDITION  
4785 ;ENTRY ROUTINE TO READ THE RESPONSE ENTERED  
4786 ;AT THE TTY AND CHECK THEM FOR LEGALITY AND  
4787 ;LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL  
4788 ;(0-7) AND MUST FALL WITHIN THE LIMITS SET BY  
4789 ;THE CALLING ROUTINE.  
4790 ;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,  
4791 ;A QUESTION MARK IS TYPED (?) AND THE RESPONSE  
4792 ;MAY BE REENTERED.  
4793 ;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND  
4794 ;MAY BE TERMINATED AT LESS THAN SIX BY TYPING A  
4795 ;CARRIAGE RETURN  
4796 ;*****  
4797  
4798 023346 005037 000642 TTR: CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG  
4799 023352 005000 CLR RO  
4800 023354 104010 TTR0: TTINN ;GO READ CHARACTER  
4801 023356 122737 000015 000640 CMPB #15,TIB ;++G SEE IF CR  
4802 023364 001004 BNE TTR1 ;IF NOT: BR  
4803 023366 005737 000642 TST TEMP1 ;SEE IF FIRST CHARACTER  
4804 023372 001436 BEQ TTR5 ;IF SO: BR  
4805 023374 000426 BR TTR2 ;++G ELSE GO LOAD VALUE  
4806 023376 122737 000060 000640 TTR1: CMPB #60,TIB ;++G SEE IF CHAR IS LESS THAN 0  
4807 023404 101401 BLOS TTR1A ;IF NOT: BR  
4808 023406 000431 BR TINER ;++G ELSE GO TO ERROR  
4809 023410 122737 000070 000640 TTR1A: CMPB #70,TIB ;++G SEE IF CHAR IS GREATER THAN 7  
4810 023416 101001 BHI TTR1B ;IF NOT: BR  
4811 023420 000424 BR TINER ;++G ELSE GO TO ERROR  
4812 023422 005237 000642 TTR1B: INC TEMP1 ;SET FIRST CHARACTER FLAG  
4813 023426 006300 ASL RO  
4814 023430 006300 ASL RO ;SHIFT 3 LEFT  
4815 023432 006300 ASL RO  
4816 023434 042737 177770 000640 BIC #177770,TIB ;STRIP ASCII  
4817 023442 053700 000640 BIS TIB,RO ;LOAD CHARACTER  
4818 023446 005301 DEC R1 ;SEE IF DONE  
4819 023450 001341 BNE TTR0 ;IF NOT: BR  
4820 023452 020002 TTR2: CMP RO,R2 ;SEE IF EXCEEDED MAXIMUM LIMIT  
4821 023454 101401 BLOS TTR3 ;IF NOT: BR  
4822 023456 000405 BR TINER ;++G ELSE GO TO ERROR  
4823 023460 020300 TTR3: CMP R3,RO ;SEE IF BELOW MINIMUM LIMIT  
4824 023462 101401 BLOS TTR4 ;IF NOT: BR  
4825 023464 000402 BR TINER ;++G ELSE GO TO ERROR  
4826 023466 010015 TTR4: MOV RO,(R5) ;LOAD VALUE  
4827 023470 000207 TTR5: RTS PC ;EXIT  
4828 023472 012704 025755 TINER: MOV #MSG43,R4  
4829 023476 104000 TTOUTT ;PRINT?  
4830 023500 162716 000020 SUB #20,(SP) ;RESET SP TO START OF VALUE ROUTINE  
4831 023504 000207 RTS PC ;REDO VALUE ENTRY
```

```
4832
4833 ;TTY READ SUBROUTINE*****
4834
4835 023506 005277 155076 TFIN: INC @TKS
4836 023512 105777 155072 TFIN1: TSTB @TKS
4837 023516 100375 BPL TFIN1
4838 023520 017737 155066 000640 MOV @TKB,TIB
4839 023526 042737 000200 000640 BIC #200,TIB ;++G STRIP PARITY BIT
4840 023534 105777 155054 TFIN2: TSTB @TPS
4841 023540 100375 BPL TFIN2
4842 023542 113777 000640 155046 MOVB TIB,@TPB
4843 023550 000207 RTS PC
4844
4845 ;TTY OUTPUT SUBROUTINE*****
4846
4847 023552 112437 000636 TTOUT: MOVB (R4)+,TOB
4848 023556 122737 000043 000636 CMPB #43,TOB
4849 023564 001444 BEQ TEX
4850 023566 122737 000045 000636 CMPB #45,TOB
4851 023574 001407 BEQ TCRLF
4852 023576 122737 000041 000636 CMPB #41,TOB
4853 023604 001435 BEQ TBELL ;DO BELL
4854 023606 004737 023662 JSR PC,TOG
4855 023612 000757 BR TTOUT
4856 023614 112737 000015 000636 TCRLF: MOVB #15,TOB
4857 023622 004737 023662 JSR PC,TOG
4858 023626 012703 000006 MOV #6,R3
4859 023632 005037 000636 TCRLFA: CLR TOB
4860 023636 004737 023662 JSR PC,TOG
4861 023642 005303 DEC R3
4862 023644 001372 BNE TCRLFA ;DO FILLERS
4863 023646 112737 000012 000636 MOVB #12,TOB
4864 023654 004737 023662 JSR PC,TOG
4865 023660 000734 BR TTOUT
4866
4867 023662 105777 154726 TOG: TSTB @TPS
4868 023666 100375 BPL TOG
4869 023670 113777 000636 154720 MOVB TOB,@TPB
4870 023676 000207 TEX: RTS PC
4871 023700 012703 000002 TBELL: MOV #2,R3
4872 023704 012737 000007 000636 TBELA: MOV #7,TOB
4873 023712 004737 023662 JSR PC,TOG
4874 023716 005303 DEC R3
4875 023720 001371 BNE TBELA
4876 023722 000713 BR TTOUT
4877
4878
```

```
4879                                     :OCTAL OUTPUT SUBROUTINE*****
4880
4881 023724 005037 024156      OCTP:  CLR   OFL           ;CLEAR FLAG FOR LEADING ZERO
4882 023730 000403           BR     OCTPE1
4883 023732 012737 000001 024156 OCTPE: MOV   #1,OFL
4884 023740 010304           OCTPE1: MOV  R3,R4           ;SEE IF NUMBER IS ZERO
4885 023742 001006           BNE   OCTP0           ;IF NOT ZERO: BR
4886 023744 005737 024156      TST   OFL
4887 023750 001003           BNE   OCTP0
4888 023752 004737 024136      JSR   PC,OCTPG1       ;ELSE PRINT ZERO
4889 023756 000450           BR     OCTP3           ;++G SPACE AND EXIT
4890 023760 032704 100000      OCTP0: BIT  #100000,R4   ;SEE IF MSD = 1
4891 023764 001406           BEQ   OCTP1           ;IF NOT: BR
4892 023766 012704 000001      MOV   #1,R4
4893 023772 004737 024114      JSR   PC,OCTPG       ;PRINT 1
4894 023776 000137 024010      JMP   OCTP2
4895 024002 005004           OCTP1: CLR   R4
4896 024004 004737 024114      JSR   PC,OCTPG       ;PRINT 0
4897 024010 010304           OCTP2: MOV  R3,R4
4898 024012 006004           ROR   R4
4899 024014 006004           ROR   R4
4900 024016 006004           ROR   R4           ;POSITION DIGIT
4901 024020 006004           ROR   R4
4902 024022 000304           SWAB  R4
4903 024024 004737 024114      JSR   PC,OCTPG       ;PRINT DIGIT 2
4904 024030 010304           MOV  R3,R4
4905 024032 006004           ROR   R4
4906 024034 000304           SWAB  R4
4907 024036 004737 024114      JSR   PC,OCTPG       ;PRINT DIGIT 3
4908 024042 010304           MOV  R3,R4
4909 024044 006104           ROL   R4
4910 024046 006104           ROL   R4
4911 024050 000304           SWAB  R4
4912 024052 004737 024114      JSR   PC,OCTPG       ;PRINT DIGIT 4
4913 024056 010304           MOV  R3,R4
4914 024060 006004           ROR   R4
4915 024062 006004           ROR   R4
4916 024064 006004           ROR   R4
4917 024066 004737 024114      JSR   PC,OCTPG
4918 024072 010304           MOV  R3,R4
4919 024074 004737 024114      JSR   PC,OCTPG       ;PRINT DIGIT 5
4920 024100 012737 000240 000636 OCTP3: MOV  #240,TOB
4921 024106 004737 023662      JSR   PC,TOG         ;PRINT SPACE
4922 024112 000207           RTS   PC             ;EXIT
4923 024114 042704 177770      OCTPG: BIC  #177770,R4
4924 024120 001004           BNE  OCTPG0
4925 024122 005737 024156      TST  OFL
4926 024126 001001           BNE  OCTPG0
4927 024130 000207           RTS   PC
4928 024132 005237 024156      OCTPG0: INC  OFL
4929 024136 052704 000260      OCTPG1: BIS  #260,R4
4930 024142 010437 000636      MOV  R4,TOB
4931 024146 004737 023662      JSR  PC,TOG
4932 024152 010304           MOV  R3,R4
4933 024154 000207           RTS  PC
4934 024156 000000      OFL:  0             ;FIRST CHAR FLAG
```

4935  
4936  
4937  
4938  
4939  
4940  
4941  
4942  
4943  
4944  
4945  
4946  
4947  
4948  
4949  
4950  
4951  
4952  
4953  
4954  
4955  
4956  
4957  
4958  
4959  
4960  
4961  
4962  
4963  
4964  
4965  
4966  
4967  
4968  
4969  
4970  
4971  
4972  
4973  
4974  
4975  
4976  
4977  
4978  
4979  
4980  
4981  
4982  
4983  
4984  
4985  
4986  
4987  
4988  
4989  
4990

024160 005037 000636  
024164 012704 000010  
024170 110337 000636  
024174 105777 154414  
024200 100375  
024202 132737 000200 000636  
024210 105737 000636  
024214 100004  
024216 012777 000061 154372  
024224 000403  
024226 012777 000060 154362  
024234 006137 000636  
024240 005304  
024242 001354  
024244 000207  
024246 013703 000646  
024252 000303  
024254 004737 024160  
024260 013703 000646  
024264 004737 024160  
024270 000207  
  
010304  
000304  
006004  
006004  
006004  
006004  
006004  
024350  
010304  
000304  
024350  
010304  
006004  
006004  
006004  
006004  
006004  
024350  
010304  
024350  
006004  
006004  
006004  
024350  
000207  
000260 000636  
042704 177760  
050437 000636  
004737 023662  
000207

```
;DATA CHARACTER OUTPUT SUBROUTINE*****
DOUT: CLR TOB
      MOV #10,R4 ;SET NUMBER TO PRINT
      MOV R3,TOB
DOUT1: TSTB @TPS
      BPL DOUT1
      BITB #200,TOB
      TSTB TOB ;++G
      BPL DOUT2 ;++G
      MOV #061,@TPB
      BR DOUT3
DOUT2: MOV #060,@TPB
DOUT3: ROL TOB
      DEC R4
      BNE DOUT1
      RTS PC
DOUTD: MOV TEMP3,R3
      SWAB R3
      JSR PC,DOUT
      MOV TEMP3,R3
      JSR PC,DOUT
      RTS PC

; ++G TU16/TE16 SERIAL NUMBER PRINT SUBROUTINE*****
SNPT: MOV R3,R4
      SWAB R4
      ROR R4
      ROR R4
      ROR R4
      ROR R4
      ROR R4
      JSR PC,SNPG ;PRINT FIRST DIGIT
      MOV R3,R4
      SWAB R4
      JSR PC,SNPG ;PRINT SECOND DIGIT
      MOV R3,R4
      ROR R4
      ROR R4
      ROR R4
      ROR R4
      JSR PC,SNPG ;PRINT THIRD DIGIT
      MOV R3,R4
      JSR PC,SNPG ;PRINT FOURTH DIGIT
      RTS PC ;EXIT
SNPG: MOV #260,TOB ;SET NUMBER BASE
      BIC #177760,R4 ;MASK NUMBER
      BIS R4,TOB ;BUILD DIGIT
      JSR PC,TOG ;GO TYPE
      RTS PC ;RETURN
```

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











5180	025474	033061	052057	030505				
5181	025502	020066	042522	044514				
5182	025510	041101	022445	043				
5183								
5184	025515	045	046123	053101	MSG32:	.ASCII	/%SLAVE NUMBER = #/	
5185	025522	020105	052516	041115				
5186	025530	051105	036440	021440				
5187								
5188	025536	042045	047105	044523	MSG33:	.ASCII	/%DENSITY = #/	
5189	025544	054524	036440	021440				
5190								
5191	025552	050045	051101	052111	MSG34:	.ASCII	/%PARITY - #/	
5192	025560	020131	020075	043				
5193								
5194	025565	045	042522	047503	MSG35:	.ASCII	/%RECORD COUNT - #/	
5195	025572	042122	041440	052517				
5196	025600	052116	036440	021440				
5197								
5198	025606	041445	040510	040522	MSG36:	.ASCII	/%CHARACTER COUNT = #/	
5199	025614	052103	051105	041440				
5200	025622	052517	052116	036440				
5201	025630	021440						
5202								
5203	025632	050045	052101	042524	MSG37:	.ASCII	/%PATTERN NUMBER = #/	
5204	025640	047122	047040	046525				
5205	025646	042502	020122	020075				
5206	025654	043						
5207	025655	045	044523	043516	MSG38:	.ASCII	/%SINGLE PASS = #/	
5208	025662	042514	050040	051501				
5209	025670	020123	020075	043				
5210	025675	045	047105	042524	MSG40:	.ASCII	/%ENTER STALLS%READ = #/	
5211	025702	020122	052123	046101				
5212	025710	051514	051045	040505				
5213	025716	020104	020075	043				
5214								
5215	025723	045	051127	052111	MSG41:	.ASCII	/%WRITE = #/	
5216	025730	020105	020075	043				
5217								
5218	025735	045	052524	047122	MSG42:	.ASCII	/%TURN AROUND = #/	
5219	025742	040440	047522	047125				
5220	025750	020104	020075	043				
5221								
5222	025755	045	022477	043	MSG43:	.ASCII	/%?%#/	
5223								
5224	025761	045	047105	042524	MSG44:	.ASCII	/%ENTER YOZZLE STALL = #/	
5225	025766	020122	047531	055132				
5226	025774	042514	051440	040524				
5227	026002	046114	036440	021440				
5228								
5229	026010	042445	051122	040440	MSG45:	.ASCII	/%ERR AMT #/	
5230	026016	052115	021440					
5231								
5232	026022	043045	020103	043	MSG46:	.ASCII	/%FC #/	
5233								
5234	026027	045	040503	021440	MSG47:	.ASCII	/%CA #/	
5235								

CZTU  
CZTU  
PAPF  
PAPF  
PAPF  
PAPF  
PAPF  
PARC  
PARS  
PATF  
PATF  
PFLC  
PICK  
PIK  
PIK  
PIK  
PIK  
PIK  
PIK  
PRB  
PRS  
PSW  
RANF  
RANC  
RANS  
RAN  
RCM  
RCM  
RCSA  
RDA  
RDA  
RDCI  
RDEI  
RDEI  
RDEI  
RDEI  
RDEI  
RDEI  
RDEI  
RDFL  
RDR  
RDR  
RDR  
RDR  
RDR  
RDR







```
5404 027434 051445 051127 020075 SMSWR: .ASCII /%SWR= #/  
5405 027442      043  
5406 027443      040 047040 SMNEW: .ASCII / NEW= #/  
5407 027450 020075      043  
5408 027453      045      043 MCRLF: .ASCII /%#/  
5409  
5410      027456  
5411 027456 000000 WDATA: 0 ;WRITE BUFFER  
5412  
5413      033464  
5414 033464 000000 RDATA: 0 ;READ BUFFER  
5415  
5416      000001 .END
```







DATA4	003002	1838#							
DATA5	003004	1839#							
DATA6	003006	1840#							
DATA7	003010	1841#							
DATBL	002770	1833#	3457						
DATER1	001130	1689#	2044	3839*					
DATR	015102	1975	3441	3637#					
DATRO	015120	3641#	3644						
DATO	014412	1834	3479#						
DATOA	014444	3486#	3494	3498	3501				
DATOB	014452	3487#	3488						
DATOC	014516	3492	3499#						
DATOD	014524	3502#	3510						
DATOE	014534	3504#	3509						
DATOF	014546	3506	3508#						
DAT1	014556	1835	3515#						
DAT1A	014562	3516#	3525	3548	3553	3601			
DAT1B	014566	3517#	3519						
DAT10	014700	1842	3568#						
DAT10A	014712	3571#	3575						
DAT11	014730	1843	3580#						
DAT11A	014736	3582#	3585						
DAT12	014750	1844	3590#						
DAT12A	014760	3592#	3595						
DAT13	014772	1845	3600#						
DAT14	015002	1846	3605#						
DAT14A	015014	3608#	3612						
DAT15	015032	1847	3617#						
DAT15A	015036	3618#	3624						
DAT15B	015046	3620#	3622						
DAT2	014576	1836	3524#						
DAT3	014602	1837	3529#						
DAT3A	014610	3531#	3542						
DAT3B	014614	3532#	3535						
DAT4	014626	1838	3540#						
DAT5	014636	1839	3547#						
DAT6	014644	1840	3552#						
DAT7	014652	1841	3557#						
DAT7A	014666	3560#	3563						
DB	000532	1535#							
DCHK	015544	2783	2978	3754#					
DCHKO	015574	3758	3761#						
DEREV1	001170	1711#	2064	3841*					
DEREX	016622	3912	3933	3935	3943	3951	3954	3956#	
DEREX1	016654	3957	3960	3962	3964#				
DERFL	000704	1594#	3755*	3832	3965*				
DERR	016160	3825	3872#						
DERRO	016170	3874#	3963						
DERROA	016220	3876	3881#						
DERROB	016252	3887	3890#						
DERROC	016276	3894	3897#						
DERROD	016300	3896	3898#						
DERR1	016326	3902	3905#						
DERR2	016330	3904	3906#						
DERR3	016344	3909#							
DERR4	016346	3873	3908	3910#					







MSG14	025054	2540	4734	5099#
MSG15	025061	3915	5101#	
MSG16	025112	4376	5107#	
MSG16A	026776	2131	5348#	
MSG17	025115	4373	5109#	
MSG2	024735	3891	5073#	
MSG20	025120	2125	5111#	
MSG21	025133	3356	5115#	
MSG22	025160	2728	5120#	
MSG23	025205	4253	5125#	
MSG23A	025213	4287	5128#	
MSG23B	025220	4273	5130#	
MSG23C	025225	4283	5132#	
MSG23D	025232	4257	5134#	
MSG23E	025240	4263	5136#	
MSG23F	025245	4267	5138#	
MSG23G	025252	5140#		
MSG23H	025257	4320	5142#	
MSG23I	025264	5144#		
MSG23J	025271	5146#		
MSG23K	025276	5148#		
MSG23L	025303	5150#		
MSG23M	025310	5152#		
MSG24	025315	4477	5154#	
MSG25	025335	4424	5158#	
MSG26	025355	4052	5162#	
MSG27	025366	4067	5165#	
MSG28	025377	2549	2564	5168#
MSG3	024742	3899	5075#	
MSG30	025401	3179	5169#	
MSG31	025453	3176	5177#	
MSG32	025515	3233	5184#	
MSG33	025536	3270	5188#	
MSG34	025552	3282	5191#	
MSG35	025565	3315	5194#	
MSG36	025606	3325	5198#	
MSG37	025632	3337	5203#	
MSG38	025655	3365	5207#	
MSG4	024747	3881	5077#	
MSG40	025675	3376	5210#	
MSG41	025723	3385	5215#	
MSG42	025735	3394	5218#	
MSG43	025755	4828	5029	5222#
MSG44	025761	2984	5224#	
MSG45	026010	5229#		
MSG46	026022	5232#		
MSG47	026027	5234#		
MSG48	026034	2165	2259	5236#
MSG49	026066	1989	2079	5242#
MSG5	024754	2306	4418	5079#
MSG50	026102	3262	5244#	
MSG52	026127	3216	5248#	
MSG53	026151	3294	5253#	
MSG54	026164	2358	4247	5256#
MSG55	026173	3037	5259#	
MSG56	026202	4249	5262#	













TEX	023676	4849	4870#											
TIB	000640	1576#	4801	4806	4809	4816*	4817	4838*	4839*	4842	4999*	5000*	5001	5014
		5017	5025	5027	5035*	5036								
TINER	023472	4808	4811	4822	4825	4828#								
TINF	000634	1574#	1892*	1897*	1901*	2175*	3166							
TINP	012400	1940	3166#											
TINPA	012410	3167	3169#											
TINPB	012430	3173#	3175											
TINPBO	012614	3207#	3210											
TINPB1	012454	3178	3180#											
TINPC	012642	3182	3212	3215#	3232									
TINPX	014142	3375	3403#											
TINPO	012746	3226	3233#	3245	3259	3310								
TINPOA	013022	3244	3246#											
TINPOB	013040	3242	3249#											
TINPOC	013100	3255#												
TINPOD	013122	3256	3260#											
TINPOE	013146	3261	3266#											
TINP1	013164	3270#												
TINP2	013242	3278	3282#											
TINP2A	013320	3290	3294#											
TINP2B	013376	3302	3306#											
TINP2C	013424	3248	3308	3311#										
TINP3	013444	3315#												
TINP4	013774	3374#	4500											
TKB	000612	1562#	2982	2996	3000*	4488	4838	4999						
TKS	000610	1561#	1951*	2911*	2980	3005*	4835*	4836	4997					
TMEX	000564	1551#	2355	2693	2805	3029	3349	3351	4622*					
TMFLG	000676	1591#	2357*	2401*	2417	2690*	2695*	2721	2734	2781	2793*	2807	2809*	2812*
		2923	2947	2974	4137	4159	4166	4177	4192	4245	4344	4742		
TOB	000636	1575#	1902	4277*	4298*	4308*	4721*	4751*	4847*	4848	4850	4852	4856*	4859*
		4863*	4869	4872*	4920*	4930*	4938*	4940*	4943	4944	4949*	4981*	4983*	
TOG	023662	4278	4299	4309	4722	4752	4854	4857	4860	4864	4867#	4868	4873	4921
		4931	4984											
TPB	000616	1564#	4842*	4869*	4946*	4948*								
TPOS	014152	3281	3293	3305	3409#	3412								
TPOS1	014164	3251	3413#											
TPS	000614	1563#	4840	4867	4941									
TRAP30	024670	1469	5051#											
TSTAL	000576	1556#	2004	2007	2493	2617	2642	3025	3056	3396	3398			
TTIN	023506	4835#	5061											
TTINN =	104010	4800	5013	5066#										
TTINT	021630	1485	4487#											
TTINTO	021700	4491	4499#											
TTIN1	023512	4836#	4837											
TTIN2	023534	4840#	4841											
TTOUT	023552	4847#	4855	4865	4876	5057								
TTOUTT=	104000	1886	1990	2017	2022	2027	2032	2037	2042	2047	2052	2057	2062	2080
		2123	2126	2132	2143	2166	2187	2200	2239	2260	2340	2348	2388	2398
		2430	2432	2442	2449	2451	2455	2537	2541	2550	2565	2573	2583	2729
		2833	2839	2849	2851	2867	2874	2885	2985	3180	3184	3193	3217	3228
		3234	3258	3263	3267	3271	3283	3295	3316	3326	3338	3348	3357	3366
		3377	3386	3395	3879	3882	3892	3900	3916	4053	4068	4243	4250	4254
		4258	4264	4268	4274	4284	4288	4294	4305	4321	4374	4377	4419	4422
		4425	4431	4475	4478	4521	4528	4541	4543	4547	4686	4690	4695	4701
		4708	4718	4731	4735	4756	4829	5004	5006	5010	5021	5030	5062#	





.	= 033466	1468#	1470#	1475	1476#	1478#	1480#	1484#	1491#	1496#	1499#	1502#	1508#	1514#
		1519#	1523#	1722#	1724#	1726#	1728#	1730#	1732#	1734#	1736#	1738#	1740#	1742#
		1744#	1746#	1748#	1750#	1752#	1759#	1761#	1763#	1765#	1767#	1769#	1771#	1773#
		1833	5410#	5413#										

. ABS. 033466 000

ERRORS DETECTED: 0

DSKW:CZTUAM,DSKW:CZTUAM.SEQ/SOL/NL:TOC=CZTUAM.SML/ML,CZTUAM.P11

RUN-TIME: 6 12 2 SECONDS

RUN-TIME RATIO: 131/21=6.1

CORE USED: 13K (25 PAGES)

DOCUMENT PAGES: 134