

.REM *

IDENTIFICATION

PRODUCT CODE: AC-9447J-MC
PRODUCT TITLE: CZTUAJO TM02-TU16/TE16 RELIAB
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: R.B. BARNES
PRODUCT DATE: 25 MAY 1984

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) 1974,1984 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORORATON

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(TM02 ADDRESS), SLAVE NUMBER, DENSITY, PARITY, FORMAT, RECORD COUNT, CHARACTER COUNT, PATTERN NUMBER, TAPE MARK AND STALL FOR READ, WRITE, AND TURNAROUND. ALL REPOSSES SHOULD BE MADE IN OCTAL AND WITHIN THE LIMITS OF THE PARAMETER. A QUESTION MARK (?) WILL BE TYPED IF ANY CHARACTER ENTERED IS NOT BETWEEN 0 THRU 7 (OCTAL). THE CHARACTER MAY BE RETYPED FOLLOWING THE QUESTION MARK. IF THE RESPONSE IS NOT WITHIN ITS LIMITS. A QUESTION MARK (?) IS TYPED AND THE ENTIRE RESPONSE MAY BE RENTERED. SOME RESPONSES REQUIRE MORE THAN ONE (1) CHARACTER, BUT NONE REQUIRES MORE THAN SIX (6). RESPONSES OF MORE THAN ONE CHARACTER NEED NOT HAVE LEADING ZEROS AND SHOULD BE TERMINATED BY A CARRIAGE RETURN IF LESS THAN THE MAXIMUM NUMBER OF CHARACTERS IS INPUT.
- B. 204(8): THIS ADDRESS SHOULD BE USED ANYTIME A RESTART OF THE PROGRAM IS NECESSARY AND THE PARAMETERS ENTERED AT THE INITIAL START OF 200(8) NEED NOT BE CHANGED. ALSO NOTE THAT ANY DATA PATTERN WHICH HAD BEEN GENERATED BY SETTING THE RANDOM DATA SWITCH (CONSOLE SWITCH EIGHT) WILL NOT BE OVERWRITTEN AND 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 RM
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 RM 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 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-DZTLF-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
(100000)0=CONTINUE ON ERROR

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

SW13: 1=DO NOT CHECK DATA ERRORS

(020000)0-CHECK DATA ERRORS

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

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

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

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

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

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

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

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

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

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

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

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

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

SWITCH EXPLANATION AND EXAMPL :

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

FXAMPLES: 0-3

- A. SWO=0,SW1=0,SW2=1,SW3=1
WRITE ONLY X RECORDS OF Y CHARACTERS
- B. SWO=0,SW1=0,SW2=1,SW3=0
WRITE THEN BACKSPACE AND READ FORWARD X RECORDS
- C. SWO=0,SW1=0,SW2=0,SW3=1
WRITE THEN READ REVERSE X RECORDS.
- D. SWO=0,SW1=0,SW2=0,SW3=0
WRITE THEN READ REVERSE AND READ FORWARD X RECORDS
- E. SWO=0,SW1=1,SW2=0,SW3=0
WRITE THEN BACKSPACE AND READ FORWARD THEN REVERSE
- F. SWO=1,SW1=0,SW2=1,SW3=0
READ TAPE FORWARD X RECORDS
- G. SWO=1,SW1=0,SW2=0,SW3=1
READ TAPE REVERSE X RECORDS
- H. SWO=1,SW1=0,SW2=0,SW3=0
READ TAPE REVERSE THEN FORWARD
- I. SWO=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 SKJP ERASE IS DONE, A SUPECTED BAD TAPE SPOT IS LOGGED AT THIS BLOCK AND RECORD NUMBER, AND A SECOND RETRY OF FOUR REPEATS IS DONE. IF AFTER FOUR (4) RETRIES, THE RECORD CANNOT BE RECOVERED A NOTIFICATION IS PRINTED, AND TESTING IS RESUMED ON THE NEXT RECORD. IF 20(8) BAD TAPE SPOTS ARE FOUND, THE SLAVE WILL BE REWOUND AND REMOVED FROM TESTING WITH AN APPROPRIATE MESSAGE PRINTED. THE READ RETRY SCHEME CONSISTS OF REREADING THE RECORD UP TO EIGHT TIMES. IF ALL EIGHT REREADS ARE BAD, IT IS A HARD ERROR. IF ANY REREAD IS SUCCESSFUL, THIS IS A SOFT ERROR. IF THE ORIGINAL ERROR IS OF THE NON-RETRYABLE TYPE (IE: ILF,RMR,ILR,NEF,CBUSPE), THE RETRY SCHEME IS NOT ENTERED AND A MESSAGE IS PRINTED.
- SW5: SWITCH FIVE (5) WHEN SET DURING A READ FORWARD OR REVERSE WILL CAUSE THE TAPE TO CONTINUOUSLY READ THE CURRENT RECORD BY SPACING EITHER FORWARD OR REVERSE AND REREADING THAT RECORD. THIS TAPE MOVEMENT IS CALLED YOZZLING. THERE IS A SOFTWARE DELAY EXECUTED BETWEEN EACH SPACE/READ OF THE RECORD AND IT MAY BE VARIED BY TYPING CONTROL C ON THE TELETYPE DURING THE EXECUTION OF THE YOZZLE AND RESPONDING TO THE PRINTED REQUEST WITH A SIX (6) DIGIT VALUE. THE YOZZLE STALL IS PRESET TO A VALUE OF 3000 IN THE PROGRAM TO PREVENT EXCESSIVE TAPE WEAR, BUT MAY BE SET TO ANY VALUE THROUGH THE TELETYPE.
- SW6-8: THESE THREE (3) SWITCHES CONTROL THE RANDOMIZATION OF DATA AND BLOCK SIZE AND MAY BE SET AND RESET AT ANY TIME. THE ACTUAL CHANGE WILL TAKE PLACE BETWEEN BLOCK CYCLES.
- SW9: SWITCH NINE (9) WHEN SET WILL CAUSE ALL AVAILABLE TAPE UNITS TO BE REWOUND AT THE END OF THE CURRENT BLOCK CYCLE. TESTING WILL BE RESUMED AT A BLOCK COUNT OF ONE (1) WHEN ALL UNITS HAVE REACHED BOT.

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

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

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

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

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

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

9. ERROR PRINTOUTS

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

A. OPERATION ERRORS:

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

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

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

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

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

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

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

BAD RECORD

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

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

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

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

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

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

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

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

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

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

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

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

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

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

10. STATISTICS PRINTOUT

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

STATISTICS PRINT EXAMPLE (A HEADER WILL PRECEED THE STATS)

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

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

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

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

11. AUTO SEQUENCE

THE AUTO SEQUENCE (START AT ADDRESS 240) WILL EXECUTE A PREDETERMINED TEST PLAN ON ALL AVAILABLE SLAVES ON EACH AVAILABLE 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 TO ALLOW SCOPING OF THIS PARTICULAR RECORD.

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

TRY IT, YOU'LL LIKE IT.

13. LISTING

4

```

1343 .LIST BIN,LOC,SEQ
1344 .TITLE CZTUAJO TMO2-TU16/TE16 RELIAB
1345 ;ZZ - CZTUAJO
1346 ;21 APRIL 76
1347 ;R. BARNES
1348
1349 ;REVISED (..G) J.G.ADAMS MAY 1977
1350 ;..G
1351 ;..G
1352 ;..G
1353 ;..G
1354 ;..G
1355 ;..G
1356 ;..G
1357
1358 ;REVISED (..JH) J. HITT MAY 1984
1359 ;
1360
1361 .MCALL .%ACT11,.$EOP,$CHAIN ;..G ACT11 HOOKS
1362 .NLIST MC ;..G DO NOT LIST MACRO CALLS
1363 .LIST ME ;..G LIST MACRO EXPANSIONS
1364 .ENABLE ABS,AMA ;..G ENABLE ABS AND MODE '37'
1365
1366
1367 ;CONSOLE SWITCHES*****
1368
1369 ;SW15: 1=STOP ON ERROR
1370 ; 0=CONTINUE ON ERROR
1371
1372 ;SW14: 1=PRINT READ/WRITE STATS
1373 ; 0=DO NOT PRINT STATS
1374
1375 ;SW13: 1=DO NOT CHECK DATA
1376 ; 0=CHECK DATA
1377 ;SW12: 1=DO NOT CHECK WRITE ERRORS
1378 ; 0=CHECK WRITE ERRORS
1379 ;SW11: 1=DO NOT CHECK READ ERRORS
1380 ; 0=CHECK READ ERRORS
1381 ;SW10: 1=DO NOT PRINT ERRORS
1382 ; 0=PRINT ERRORS
1383
1384 ;SW9: 1=REWIND TAPE
1385 ; 0=DO NOT REWIND
1386
1387 ;SW8: 1=USE RANDOM DATA
1388 ; 0=USE FIXED DATA PATTERN
1389 ;SW7: 1=USE RANDOM CHARACTER COUNT
1390 ; 0=USE FIXED CHAR COUNT
1391 ;SW6: 1=USE RANDOM RECORD COUNT
1392 ; 0=USE FIXED RECORD COUNT
1393
1394 ;SW5: 1=YOZZLE ON CURRENT RECORD
1395 ; 0=DO NOT YOZZLE
1396
1397 ;SW4: 1=DO BOTH READ AND WRITE RETRIES

```

```

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

```

```

ADDED XON/XOFF FUNCTIONALITY
FOR REMOTE DIAGNOSIS

```

1398
1399
1400
1401
1402
1403
1404
1405
1406
1407

; 0=INHIBIT RETRIES
;SW3: 1=DO NOT READ FORWARD
; 0=READ FORWARD
;SW2: 1=DO NOT READ REVERSE
; 0=READ REVERSE
;SW1: 1=READ FORWARD FIRST
; 0=READ REVERSE FIRST
;SW0: 1=DO NOT WRITE
; 0=WRITE


```

1454                                ;REGISTER EQUIVS*****
1455
1456                                R0=#0
1457                                R1=#1
1458                                R2=#2
1459                                R3=#3
1460                                R4=#4
1461                                R5=#5
1462                                SP=#6
1463                                PC=#7
1464                                NOP=240
1465
1466                                ;TRAP CATCHERS*****
1467                                . =30
1468 000030 025016                TRAP30
1469                                . =32
1470 000032 000340                340
1471
1472
1473                                ;ACT11 HOOK *****
1474                                $SVPC= .                                ;SAVE CURRENT LOCATION CTR
1475                                . =46
1476 000046 005120                .WORD $ENDAD                                ;SET LOCATION 46
1477                                . =52
1478 000052 000000                .WORD 0                                ;SET LOCATION 52 = 0
1479                                . = $SVPC                                ;RESTORE LOCATION CTR
1480
1481
1482                                ;TTY INTERRUPT VECTOR*****
1483                                . =60
1484 000060 021632                TTINT                                ;TTY INTERRUPT HANDLER ADDRESS
1485 000062 000000                0
1486
1487
1488                                ;SOFTWARE SWITCH REGISTER LOC. 176*****
1489
1490                                . =176
1491 000176 000000                SWREG: 0                                ;SOFTWARE SWITCH REGISTER
1492
1493                                ;START ADDRESS*****
1494
1495                                . =200
1496 000200 000137 003030        JMP START                                ;ENTER PARAMETERS VIA TTY
1497
1498                                . =204
1499 000204 000137 003154        JMP STARTC                                ;USE FIXED PARAMETERS; HOLD DATA
1500
1501                                . =210
1502 000210 005037 015152        CLR RDFL                                ;USE FIXED PARAMETERS; NEW DATA
1503 000214 000137 003162        JMP STARTA
1504
1505                                ;MAG TAPE INTERRUPT VECTOR*****
1506
1507                                . =224
1508 000224 021750                MTINT                                ;MAG TAPE INTERRUPT HANDLER ADDRESS
1509 000226 000340                340

```

```
1510  
1511  
1512  
1513 000240 000240  
1514 000240 005237 000734  
1515 000244 000137 003140  
;AUTO SEQUENCE START*****  
.=240  
INC ASEQF ;SET AUTO SEQUENCE FLAG  
JMP STAUT ;GO TO START OF AUTO SEQUENCE
```

```

1516                                     ;SHORT CONVERSATION RESTART*****
1517
1518                                     .-300
1519 000300 005237 014152             INC     SCVFL           ;SET SHORT CONVERSATION FLAG
1520 000304 000137 003030             JMP     START         ;ENTER SHORT PARAMETER LIST
1521
1522                                     .-510
1523                                     ;TU16/TE16 REGISTER EQUIVS*****
1524
1525 000510 172440                     C1:     172440
1526 000512 172442                     WC:     172442
1527 000514 172444                     BA:     172444
1528 000516 172446                     FC:     172446
1529 000520 172450                     CS:     172450
1530 000522 172452                     DS:     172452
1531 000524 172454                     ER:     172454
1532 000526 172456                     AS:     172456
1533 000530 172460                     CC:     172460
1534 000532 172462                     DB:     172462
1535 000534 172464                     MR:     172464
1536 000536 172466                     DT:     172466
1537 000540 172470                     SN:     172470
1538 000542 172472                     C2:     172472
1539
1540                                     ;CONSTANTS*****
1541
1542 000544 172440                     REGS:   172440           ;STARTING REGISTER ADDRESS (CS1)
1543 000546 000224                     VECT:   224            ;VECTOR ADDRESS (RM INTERRUPT)
1544 000550 000000                     DVN:    0              ;DRIVE NUMBER
1545 000552 000000                     UDES:   0              ;UNIT DESCRIPTION (PARITY,DENSITY,UNIT,FORMAT)
1546 000554 000100                     RCNT:   100           ;RECORD COUNTER
1547 000556 177600                     FMCNT:  177600        ;NUMBER OF CHAR (4 - 4000) OCTAL IN TWOS COMPLEMENT
1548 000560 000001                     PATRN:  1              ;DATA PATTERN SELECTOR (0 - 15) OCTAL
1549 000562 000002                     RDCHD:  2              ;READ COMMAND
1550 000564 000000                     TMEX:   0              ;TAPE MARK FLAG: 1=TM 0=NO TM
1551 000566 000000                     INTRF:  0              ;INTERCHANGE READ 1=YES 0=NO
1552 000570 000000                     SPFLG:  0              ;SINGLE PASS 1=YES 0=NO
1553 000572 000001                     RSTAL:  1              ;READ STALL
1554 000574 000001                     WSTAL:  1              ;WRITE STALL
1555 000576 000001                     TSTAL:  1              ;TURN AROUND STAL
1556 000600 002000                     YSTAL:  2000          ;YOZZLE STAL
1557 000602 000010                     RETRY:  10             ;READ RETRY NUMBER
1558 000604 177776                     PSW:    177776        ;PROCESSOR STATUS
1559 000606 177570                     SWR:    177570        ;CONSOLE SWITCHES
1560 000610 177560                     TKS:    177560        ;TTY READ STATUS REGISTER
1561 000612 177562                     TKB:    177562        ;TTY READ BUFFER
1562 000614 177564                     TPS:    177564        ;TTY PUNCH STATUS REGISTER
1563 000616 177566                     TPB:    177566        ;TTY PUNCH OUTPUT REGISTER
1564 000620 177550                     PRS:    177550        ;H/S READER STATUS REGISTER
1565 000622 177552                     PRB:    177552        ;H/S READER BUFFER
1566 000624 153624                     RANBAS: 153624        ;RANDOM NUMBER GENERATOR BASE
1567 000626 032561                     RANSAV: 032561       ;RANDOM NUMBER BUFFER
1568 000630 000000                     RCSAV:  0              ;RECORD COUNT SAVE
1569 000632 000000                     FCSAV:  0              ;FRAME COUNT SAVE

```

1570				
1571				
1572				; FLAGS AND COUNTERS*****
1573	000634	000000	TINF: 0	;TTY ENTRY FLAG
1574	000636	000000	TOB: 0	;TTY OUTPUT BUFFER
1575	000640	000000	TIB: 0	;TTY INPUT BUFFER
1576	000642	000000	TEMP1: 0	;TEMP STORAGE
1577	000644	000000	TEMP2: 0	;TEMP STORAGE
1578	000646	000000	TEMP3: 0	;TEMP STORAGE
1579	000650	000000	NRZOF: 0	;NRZ ONLY FLAG
1580	000652	000000	EMADDR: 0	;ERROR MSG ADDRESS STORAGE
1581	000654	000000	BLCNTR: 0	;BLOCK COUNTER
1582	000656	000000	BBC: 0	;BAD RECORD COUNTER
1583	000660	000000	EOTREC: 0	;EOT FLAG
1584	000662	000000	RTRN: 0	;INTERRUPT RETURN STORAGE
1585	000664	000000	HDRFL: 0	;HEADER FLAG
1586	000666	000000	STAL: 0	;DELAY STORAGE
1587	000670	000000	PFLG: 0	;PRINT FLAG
1588	000672	000000	MTC1: 0	;MAG TAPE CONT REGISTER BUFFER
1589	000674	000000	UNP: 0	;UNIT TABLE POINTER
1590	000676	000000	TMFLG: 0	;TAPE MARK FLAG
1591	000700	000000	RPCNT: 0	;REPEAT COUNTER
1592	000702	000000	RTCNT: 0	;RETRY COUNTER
1593	000704	000000	DERFL: 0	;DATA ERROR FLAG
1594	000706	000000	SERFL: 0	;STATUS ERROR FLAG
1595	000710	000000	BCNT: 0	;BIT COUNTER
1596	000712	000000	RTYFL: 0	;RETRY FLAG
1597	000714	000000	UPS: 0	;UNIT POINTER SAVE
1598	000716	000000	BDPP: 0	;BITS DROPPED POINTER
1599	000720	000000	BPKP: 0	;BITS PICKED POINTER
1600	000722	000000	ERSAV: 0	;ERROR SAVE LOC
1601	000724	000000	BTFLG: 0	;BAD TAPE FLAG
1602	000726	000000	BTSTF: 0	;STATISTIC PRINT FLAG
1603	000730	000000	BTPT: 0	;BAD TAPE POINTER
1604	000732	000000	ERTFL: 0	;ERASE FLAG
1605	000734	000000	ASEQF: 0	;AUTO SEQ FLAG
1606	000736	000000	ADRVN: 0	;AUTO SEQ DRIVE NUMBER
1607	000740	000000	ABLCNT: 0	;AUTO BLOCK COUNTER
1608	000742	000000	ASEQCF: 0	;AUTO SEQ CONTINUOUS FLAG
1609	000744	000000	EOPB1: 0	;EOP FLAG
1610	000746	000000	CTRLS: 0	;JH XON/XOFF FLAG

1667 001072 000000
 1668 001074 000000
 1669 001076 000000
 1670 001100 000000
 1671 001102 000000
 1672 001104 000000
 1673 001106 000000
 1674 001110 000000

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

;UNIT READ FORWARD ERRORS*****

1675
 1676
 1677
 1678 001112 000000
 1679 001114 000000
 1680 001116 000000
 1681 001120 000000
 1682 001122 000000
 1683 001124 000000
 1684 001126 000000
 1685 001130 000000

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

;UNIT DATA ERRORS FORWARD*****

1686
 1687
 1688
 1689 001132 000000
 1690 001134 000000
 1691 001136 000000
 1692 001140 000000
 1693 001142 000000
 1694 001144 000000
 1695 001146 000000
 1696 001150 000000

DATER1: 0
 0
 0
 0
 0
 0
 0
 0

;UNIT READ REVERSE ERRORS*****

1697
 1698
 1699
 1700 001152 000000
 1701 001154 000000
 1702 001156 000000
 1703 001160 000000
 1704 001162 000000
 1705 001164 000000
 1706 001166 000000
 1707 001170 000000

RDERR1: 0
 0
 0
 0
 0
 0
 0
 0

;UNIT DATA ERRORS REVERSE*****

1708
 1709
 1710
 1711 001172 000000
 1712 001174 000000
 1713 001176 000000
 1714 001200 000000
 1715 001202 000000
 1716 001204 000000
 1717 001206 000000
 1718 001210 000000

DEREV1: 0
 0
 0
 0
 0
 0
 0
 0

```

1719                                ;DROPS + PICKS PER CHANNEL PER UNIT*****
1720
1721 001212 000000          BP00: 0
1722                                .=. +16
1723 001232 000000          BP10: 0
1724                                .=. +16
1725 001252 000000          BP20: 0
1726                                .=. +16
1727 001272 000000          BP30: 0
1728                                .=. +16
1729 001312 000000          BP40: 0
1730                                .=. +16
1731 001332 000000          BP50: 0
1732                                .=. +16
1733 001352 000000          BP60: 0
1734                                .=. +16
1735 001372 000000          BP70: 0
1736                                .=. +16
1737 001412 000000          BD00: 0
1738                                .=. +16
1739 001432 000000          BD10: 0
1740                                .=. +16
1741 001452 000000          BD20: 0
1742                                .=. +16
1743 001472 000000          BD30: 0
1744                                .=. +16
1745 001512 000000          BD40: 0
1746                                .=. +16
1747 001532 000000          BD50: 0
1748                                .=. +16
1749 001552 000000          BD60: 0
1750                                .=. +16
1751 001572 000000          BD70: 0
1752                                .=. +16
1753                                .=. +16
1754                                .=. +16

```

```

1755
1756                                     ;UNIT BAD TAPE COUNTER:16 PER SLAVE*****
1757
1758 001612 000000          BT00: 0
1759          001716          .*.102
1760 001716 000000          BT01: 0
1761          002022          .*.102
1762 002022 000000          BT02: 0
1763          002126          .*.102
1764 002126 000000          BT03: 0
1765          002232          .*.102
1766 002232 000000          BT04: 0
1767          002336          .*.102
1768 002336 000000          BT05: 0
1769          002442          .*.102
1770 002442 000000          BT06: 0
1771          002546          .*.102
1772 002546 000000          BT07: 0
1773          002652          .*.102
1774
1775                                     ;UNIT END OF TAPE COUNTERS 1 PER SLAVE*****
1776
1777 002652 000000          EOTCO: 0
1778 002654 000000          0
1779 002656 000000          0
1780 002660 000000          0
1781 002662 000000          0
1782 002664 000000          0
1783 002666 000000          0
1784 002670 000000          0
1785
1786                                     ;UNIT READ FORWARD SOFT ERROR*****
1787
1788 002672 000000          RFSOFT: 0
1789 002674 000000          0
1790 002676 000000          0
1791 002700 000000          0
1792 002702 000000          0
1793 002704 000000          0
1794 002706 000000          0
1795 002710 000000          0
1796
1797                                     ;UNIT READ REVERSE SOFT ERPR*****
1798
1799 002712 000000          RRSOFT: 0
1800 002714 000000          0
1801 002716 000000          0
1802 002720 000000          0
1803 002722 000000          0
1804 002724 000000          0
1805 002726 000000          0
1806 002730 000000          0
1807

```



```

1808
1809
1810
1811 002732 000000 RFHARD: 0
1812 002734 000000 0
1813 002736 000000 0
1814 002740 000000 0
1815 002742 000000 0
1816 002744 000000 0
1817 002746 000000 0
1818 002750 000000 0
1819
1820 ;UNIT READ REVERSE HARD ERROR*****
1821
1822 002752 000000 RRHARD: 0
1823 002754 000000 0
1824 002756 000000 0
1825 002760 000000 0
1826 002762 000000 0
1827 002764 000000 0
1828 002766 000000 0
1829 002770 000000 0
1830
1831 ;DATA PATTERN GENERATORS*****
1832
1833 002772 002772 DATBL: . ;ENTRY TABLE
1834 002774 014414 DATA0: DAT0 ;EXTERNAL INPUT FROM W/S READER(SEE MAINDEC-11-DZTUF)
1835 002776 014560 DATA1: DAT1 ;ALL ONES
1836 003000 014600 DATA2: DAT2 ;ALL ZEROS
1837 003002 014604 DATA3: DAT3 ;WALKING ONE
1838 003004 014630 DATA4: DAT4 ;WALKING ZERO
1839 003006 014640 DATA5: DAT5 ;ALTERNATING ONE/ZERO
1840 003010 014646 DATA6: DAT6 ;ALTERNATING ZERO/ONE
1841 003012 014654 DATA7: DAT7 ;ALTERNATING ONE/ZERO IN ALTERNATING CHARACTERS
1842 003014 014702 DATA10: DAT10 ;WALKING ONE/ALL ONE IN ALTERNATING CHARACTERS
1843 003016 014732 DATA11: DAT11 ;ALL BITS 0-377
1844 003020 014752 DATA12: DAT12 ;ALL BITS 377-0
1845 003022 014774 DATA13: DAT13 ;ALTERNATING CHARACTERS 0 AND 377
1846 003024 015004 DATA14: DAT14 ;WALKING ZERO/ALL ZERO IN ALTERNATING CHARACTERS
1847 003026 015034 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
                                ;BRANCH IF LOADED VIA ACT11 CHAIN MODE
        CMP    #ENDAD,#42    ;BRANCH IF IN DUMP MODE
        BEQ   50#
        TST   #42
        BEQ   52#
        BR    51#
50#:    MOV    #SMREG,SMR     ;INVOKE SOFTWARE SMR
        MOV   #10000,BSMR    ;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
3#:    INC    ASEQF          ;..G
        JMP   SUSWR         ;..G SET AUTO SEQUENCE FLAG
                                ;CHECK AND SET UP HRD/SOFT SWITCH REG .. C.W

;START 240 *****
1892:  STAUT: MOV    #1,TINF   ;SET TTY ENTRY FLAG
1893:    CLR    RDFL        ;CLEAR RANDOM DATA FLAG
1894:    BR    STARTB      ;..G

;START 204 *****
1897:  STARTC: CLR    TINF   ;CLEAR TTY INPUT FLAG
1898:    BR    STARTD      ;..G

;START 210 *****
1901:  STARTA: CLR    TINF   ;CLEAR TTY ENTRY FLAG
1902:  STARTB: MOV    #TOB,RO
1903:    MOV    #37,R1
1904:  STARTO: CLR    (RO).  ;CLEAR FLAGS AND COUNTERS

```

1905	003200	005301			DEC	R1	
1906	003202	001375			BNE	STARTO	
1907	003204	012706	000500		MOV	#500,SP	;SET STACK POINTER
1908	003210	004737	004374		JSR	PC,RANSET	;GO RESET RANDOM BASE
1909	003214	012700	001052		MOV	#RTY1,RO	
1910	003220	012701	000750		MOV	#750,R1	
1911	003224	005020			STARTF: CLR	(RO)+	;CLEAR STATISTIC COUNTERS
1912	003226	005301			DEC	R1	
1913	003230	001375			BNE	STARTF	
1914	003232	012737	177777	014410	MOV	#-1,PATS	;PRESET PATTERN
1915	003240	005037	000744		CLR	EOPB1	
1916	003244	012737	000001	000654	STARTE: MOV	#1,BLCNTR	;PRESET BLOCK COUNTER
1917	003252	012706	000500		STARTD: MOV	#500,SP	
1918	003256	012777	000340	175320	MOV	#340,#PSW	
1919	003264	013746	000006		SUSWR: MOV	#06,-(SP)	;SAVE VECTORS
1920	003270	013746	000004		MOV	#04,-(SP)	
1921	003274	012737	003314	000004	MOV	#1,#04	;SET UP FOR TIMEOUT
1922	003302	022777	177777	175276	CMP	#-1,#SWR	;REFERENCE HARDWARE SWITCH REGISTER
1923	003310	001402			BEQ	2#	
1924	003312	000404			BR	3#	
1925	003314	022626			1#: CMP	(SP)+,(SP)+	;ADJUST STACK
1926	003316	012737	000176	000606	2#: MOV	#SWREG,SWR	;POINT TO SOFTWARE SWITCH REG
1927	003324	012637	000004		3#: MVL	(SP)+,#04	;RESTORE VECTORS
1928	003330	012637	000006		MOV	(SP)+,#06	
1929	003334	022737	000176	000606	CMP	#SWREG,SWR	;IS SWREG SELECTED
1930	003342	001020			BNE	4#	
1931	003344	005737	000744		TST	EOPB1	
1932	003350	001015			BNE	4#	
1933	003352	005037	000744		CLR	EOPB1	
1934	003356	022737	005120	000042	CMP	#ENDAD,#042	;ACT MODE? .. C.W
1935	003364	001402			BEQ	6#	;BRANCH - IF YES .. C.W
1936	003366	004737	024600		JSR	PC,CNTLU	;CHECK FOR CONTROL G
1937	003372	005737	000734		6#: TST	ASEQF	;AUTO SEQ MODE? .. C.W
1938	003376	001402			BEQ	4#	;BRANCH - IF NO .. C.W
1939	003400	000137	022052		JMP	ASEQO	;GO DO AUTO SEQ .. C.W
1940	003404	004737	012402		4#: JSR	PC,TIMP	;GO GET PARAMETERS FROM TTY
1941	003410	012777	000040	175102	MOV	#40,#CS	;INITIALIZE
1942	003416	005000			STAUTO: CLR	RO	;POINT TO FIRST ENTRY
1943	003420	022760	177777	000750	1#: CMP	#-1,UN1(RO)	;..G BRANCH IF LAST ENTRY
1944	003426	001406			BEQ	2#	
1945	003430	042760	100000	000750	BIC	#100000,UN1(RO)	;CLEAR EOT FLAG
1946	003436	062700	000002		ADD	#2,RO	;POINT TO NEXT UNIT ENTRY
1947	003442	000766			BR	1#	;..G CONTINUE CLEARING
1948	003444	013703	005160		2#: MOV	REOTC,R3	
1949	003450	000303			SWAB	R3	
1950	003452	110337	005160		MOV	R3,REOTC	;RESTORE EOT CNTR
1951	003456	012777	000100	175124	START1: MOV	#100,#TKS	;SET TTY INTERRUPT ENABLE
1952	003464	013700	000674		MOV	UNP,RO	;RO = UNIT TABLE POINTER
1953	003470	022760	177777	000750	STAR1A: CMP	#-1,UN1(RO)	;..G BRANCH IF LAST ENTRY
1954	003476	001404			BEQ	STAR1B	;IF LAST UNIT IN STRING: BR
1955	003500	016037	000750	000552	MOV	UN1(RO),UDES	;LOAD NEXT UNIT DESCRIPTION
1956	003506	000446			BR	STAR1A	;..G
1957	003510	005237	000654		STAR1B: INC	BLCNTR	;BUMP BLOCK COUNTER
1958	003514	005737	000734		TST	ASEQF	;SEE IF AUTO SEQ
1959	003520	001411			BEQ	STAR1C	;IF NOT: BR
1960	003522	023737	000654	000740	CMP	BLCNTR,ABL CNT	;SEE IF DONE SEQ

1961	003530	001005			BNE	STAR1C		;IF NOT: BR
1962	003532	005037	000654		CLR	BLCNTR		;RESET BLOCK CNTR
1963	003536	005037	000674		CLR	UNP		;RESET UNIT POINTER
1964	003542	000207			RTS	PC		;RETURN TO AUTO SEQ
1965								
1966	003544	005037	000674		STAR1C:	CLR	UNP	
1967	003550	005000				CLR	RO	
1968	003552	016037	000750	000552		MOV	UNI(RO),UDES	;LOAD FIRST UNIT DESCRIPTION
1969	003560	105777	175022			TSTB	@SWR	;+G BRANCH IF NOT RANDOM RECORD
1970	003564	100003				BPL	START2	;+G SIZE REQUESTED.
1971	003566	001402				BEQ	START2	;IF NOT: BR
1972	003570	004737	012316			JSR	PC,CCNTR	;GO GENERATE RANDOM RECORD SIZE
1973	003574	032777	000400	175004	START2:	BIT	@400,@SWR	;SEE IF RANDOM DATA
1974	003602	001402				BEQ	START3	;IF NOT: BR
1975	003604	004737	015104			JSR	PC,DATR	;GO GENERATE RANDOM DATA
1976	003610	032777	000100	174770	START3:	BIT	@100,@SWR	;SEE IF RANDOM RECORD COUNT
1977	003616	001402				BEQ	START4	;IF NOT: BR
1978	003620	004737	012356			JSR	PC,RCNTR	;GO GENERATE RANDOM RECORD COUNT
1979	003624	005760	000750		START4:	TST	UNI(RO)	;+G BRANCH IF NOT AT EOT
1980	003630	100002				BPL	STAR40	;IF NOT: BR
1981	003632	000137	004362			JMP	START7	;ELSE GO TO NEXT UNIT
1982	003636	013777	000550	174654	STAR40:	MOV	DVN,@CS	;SET DRIVE NUMBER
1983	003644	013777	000552	174670		MOV	UDES,@C2	;SET UNIT NUMBER
1984	003652	105777	174644			TSTB	@DS	;+G BRANCH IF UNIT AVAIL
1985	003656	100412				BMI	STAR4A	
1986	003660	005337	000666			DEC	STAL	
1987	003664	001357				BNE	START4	;AWAIT TUR
1988	003666	004737	022772			JSR	PC,PAPRT	;PRINT HEADER
1989	003672	012704	026214			MOV	@MSG49,R4	
1990	003676	104000				TTOUTT		;PRINT NOT AVAIL
1991	003700	104006				STOPP		
1992	003702	000750				BR	START4	;STOP
1993	003704	013746	000552		STAR4A:	MOV	UDES,-(SP)	;+G RETRY
1994	003710	042716	175400			BIC	@175400,(SP)	;GET UNIT DESCRIPTION
1995						;CMP	@1700,(SP)	;+G CLEAR ALL BUT FORMAT BITS
1996	003714	032726	002000			BIT	@2000,(SP)	;+G BRANCH IF NRZ
1997	003720	001406				BEQ	1@	;+G BRANCH IF NZR
1998	003722	032777	000040	174572		BIT	@40,@DS	;+G BRANCH IF SLAVE IN PE FORMAT
1999	003730	001002				BNE	1@	;+G
2000	003732	000137	004362			JMP	START7	;+G GO TO NEXT UNIT
2001	003736	004737	014202		1@:	JSR	PC,DSUP	;GO SET UP WRITE DATA
2002	003742	004737	005162			JSR	PC,RIND	;REWIND
2003	003746	004737	005524			JSR	PC,WRITE	;WRITE
2004	003752	013737	000576	000666		MOV	TSTAL,STAL	;SET TURN AROUND DELAY
2005	003760	004737	012306			JSR	PC,STALL	;DELAY
2006	003764	004737	007416			JSR	PC,RSEQ	;GO TO READ SEQUENCER
2007	003770	013737	000576	000666		MOV	TSTAL,STAL	;SET TURN AROUND DELAY
2008	003776	004737	012306			JSR	PC,STALL	;DELAY
2009	004002	032777	040000	174576		BIT	@40000,@SWR	;SEE IF SHOULD PRINT STATISTICS
2010	004010	001541				BEQ	START5	;IF NOT: BR
2011	004012	012700	000001			MOV	@1,RO	;SET RECORD COUNTER TO 1
2012	004016	004737	022772			JSR	PC,PAPRT	;PRINT CYCLE NUMBER
2013	004022	004737	004032			JSR	PC,STP	;GO PRINT STATS
2014	004026	000137	004300			JMP	STPX	
2015	004032	004737	017252		STP:	JSR	PC,DPPRT	;PRINT DROPS AND PICKS
2016	004036	012704	026427			MOV	@MSG65,R4	

2017	004042	104000		TTOUTT	;PRINT RETRY TOTAL
2018	004044	013704	000674	MOV	UNP,R4
2019	004050	016403	001052	MOV	RTY1(R4),P3
2020	004054	104002		OCTPP	;PRINT RETRIES
2021	004056	012704	026600	MOV	#MSG73,R4
2022	004062	104000		TTOUTT	;PRINT WRITE ERROR TAG
2023	004064	013704	000674	MOV	UNP,R4
2024	004070	016403	001072	MOV	WTER1(R4),R3
2025	004074	104002		OCTPP	;PRINT WRITE ERRORS
2026	004076	012704	026567	MOV	#MSG72,R4
2027	004102	104000		TTOUTT	;PRINT READ FORWARD ERROR TAG
2028	004104	013704	000674	MOV	UNP,R4
2029	004110	016403	001112	MOV	RDER1(R4),R3
2030	004114	104002		OCTPP	;PRINT READ FORWARD ERRORS
2031	004116	012704	027375	MOV	#MSG113,R4
2032	004122	104000		TTOUTT	;PRINT SOFT TAG
2033	004124	013704	000674	MOV	UNP,R4
2034	004130	016403	002672	MOV	RFSOFT(R4),R3
2035	004134	104002		OCTPP	;PRINT FORWARD SOFT ERRORS
2036	004136	012704	027406	MOV	#MSG114,R4
2037	004142	104000		TTOUTT	;PRINT HARD TAG
2038	004144	013704	000674	MOV	UNP,R4
2039	004150	016403	002732	MOV	RFHARD(R4),R3
2040	004154	104002		OCTPP	;PRINT HARD FORWARE ERRORS
2041	004156	012704	026660	MOV	#MSG77,R4
2042	004162	104000		TTOUTT	;PRINT DATA ERROR FORWARD TAG
2043	004164	013704	000674	MOV	UNP,R4
2044	004170	016403	001132	MOV	DATER1(R4),R3
2045	004174	104002		OCTPP	;PRINT DATA ERROR FORWARD NUMBER
2046	004176	012704	026463	MOV	#MSG68,R4
2047	004202	104000		TTOUTT	;PRINT READ ERROR REVERSE TAG
2048	004204	013704	000674	MOV	UNP,R4
2049	004210	016403	001152	MOV	RDERR1(R4),R3
2050	004214	104002		OCTPP	;PRINT REVESE ERROR NUMBER
2051	004216	012704	027375	MOV	#MSG113,R4
2052	004222	104000		TTOUTT	;PRINT SOFT TAG
2053	004224	013704	000674	MOV	UNP,R4
2054	004230	016403	002712	MOV	RRSOFI(R4),R3
2055	004234	104002		OCTPP	;PRINT REVERSE SOFT ERROR
2056	004236	012704	027406	MOV	#MSG114,R4
2057	004242	104000		TTOUTT	;PRINT HARD TAG
2058	004244	013704	000674	MOV	UNP,R4
2059	004250	016403	002752	MOV	RRHARD(R4),R3
2060	004254	104002		OCTPP	
2061	004256	012704	026647	MOV	#MSG76,R4
2062	004262	104000		TTOUTT	;PRINT DATA ERROR REVERSE TAG
2063	004264	013704	000674	MOV	UNP,R4
2064	004270	016403	001172	MOV	DEREV1(R4),R3
2065	004274	104002		OCTPP	;PRINT DATA REVERSE ERROR NUMBER
2066	004276	000207		RTS	PC
2067	004300	005237	000726	STPX: INC	BTSTF
2068	004304	004737	007326	JSR	PC,BTPRT
2069	004310	005037	000726	CLR	BTSTF
2070	004314	017700	174266	START5: MOV	BSWR,RO
2071	004320	042700	177762	BIC	#177762,RO
2072	004324	022700	000015	CMP	#15,RO

```

;RETURN
;SET STAT ONLY PRINT
;PRINT BAD TAPE STATS
;CLEAR FLAG
;LOAD SWR
;MASK READ/WRITE SWITCHES
;SEE IF HAVE READ OR WRITE

```

2073	004330	001417			BEQ	START8		;IF NOT: BR
2074	004332	105777	174164		START6: TSTB	BDS		;+G BRANCH IF HAVE UNIT READY
2075	004336	100411			BMI	START7		;+G
2076	004340	005337	000666		DEC	STAL		
2077	004344	001372			BNE	START6		;DELAY FOR TUR
2078	004346	004737	022772		JSR	PC,PAPRT		;PRINT HEADER
2079	004352	012704	026214		MOV	#MSG49,R4		
2080	004356	104000			TTOUTT			;PRINT NOT AVAIL
2081	004360	104006			STOPP			;STOP
2082	004362	062737	000002	000674	START7: ADD	#2,UNP		;POINT TO NEXT UNIT
2083	004370	000137	003456		START8: JMP	START1		;CONTINUE
2084								
2085								;RANDOM BASE RESET*****
2086								
2087	004374	012737	153624	000624	RANSET: MOV	#153624,RANBAS		;RESET BASE
2088	004402	012737	032561	000626	MOV	#32561,RANSAV		;RESET BUFFER
2089	004410	013737	000630	000554	MOV	RCSAV,RCNT		;RESET RECORD COUNT
2090	004416	013737	000632	000556	MOV	FCSAV,FMCNT		;RESET FRAME COUNT
2091	004424	000207			RTS	PC		
2092								

```

2093 ;*****
2094 ;REWIND FROM EOT:
2095 ;
2096 ;WHEN ANY TRANSPORT BEING TESTED REACHES END OF TAPE
2097 ;DURING A READ OR WRITE OPERATION, IT WILL BE REWOUND
2098 ;AND FLAGGED AS UNAVAILABLE UNTIL ALL AVAILABLE UNITS
2099 ;HAVE REACHED EOT AT WHICH TIME ALL TESTING WILL BE RESUMED
2100 ;AT A BLOCK COUNT OF ONE (1). A MESSAGE WILL BE
2101 ;PRINTED ON THE SUPERVISORS CONSOLE AS EACH UNIT REACHES
2102 ;EOT AND IS REWOUND.
2103 ;*****
2104
2105 004426 013777 000552 174106 REOT: MOV UDES,BC2 ;LOAD COMMAND REGISTER
2106 004434 012777 000011 174046 MOV #11,BC1 ;DRIVE CLEAR
2107 004442 105777 174054 1+: TSTB SDS ;++G WAIT FOR DRIVE READY
2108 004446 100375 BPL 1: ;AWAIT DRY
2109 004450 012777 000007 174032 MOV #7,BC1 ;START REWIND
2110 004456 005737 000724 TST BTFLG ;SEE IF BAD TAPE OVERFLOW REWIND
2111 004462 001004 BNE REOT1A ;IF SO: BR
2112 004464 013700 000660 MOV EOTREC,R0
2113 004470 042700 100000 BIC #100000,R0 ;SET RECORD NUMBER OF EOT
2114 004474 005037 000660 REOT1A: CLR EOTREC ;++G CLEAR EOT IND & REC CTR
2115 004500 004737 022772 JSR PC,PAPRT ;PRINT HEADER
2116 004504 022737 000002 000724 JSR #2,BTFLG ;SEE IF POSITION ERROR
2117 004512 001003 BNE REOT1B ;IF NOT: BR
2118 004514 012704 027266 MOV #MSG109,R4 ;SET POSITION ERROR MSG
2119 004520 000406 BR REOT1F
2120 004522 022737 000001 000724 REOT1B: CMP #1,BTFLG ;SEE IF BAD TAPE OVERFLOW
2121 004530 001004 BNE REOT1C ;IF NOT: BR
2122 004532 012704 027077 MOV #MSG106,R4 ;SET BAD TAPE OVERFLOW MSG
2123 004536 104000 REOT1F: TTOUTT ;PRINT REWIND REASON
2124 004540 000412 BR REOT1E
2125 004542 012704 025246 REOT1C: MOV #MSG20,R4 ;SET EOT MSG
2126 004546 104000 REOT1D: TTOUTT ;PRINT MSG
2127 004550 013704 000674 MOV UNP,R4
2128 004554 005264 002652 JNC EOTCO(R4) ;BUMP CNTR
2129 004560 016403 002652 MOV EOTCO(R4),R3
2130 004564 104002 OCTPP ;PRINT EOT CNTR
2131 004566 012704 027124 REOT1E: MOV #MSG16A,R4
2132 004572 104000 TTOUTT ;PRINT RESTART MSG
2133 004574 005037 000724 CLR BTFLG ;CLEAR BAD TAPE FLAG
2134 004600 004737 004032 JSR PC,STP ;PRINT STATS
2135 004604 004737 007326 JSR PC,BTPRT ;PRINT BAD TAPE STATS
2136 004610 105777 173706 REOT2: TSTB SDS ;++G BRANCH IF UNIT IS READY
2137 004614 100414 BHI REOT2A
2138 004616 005337 000666 DEC STAL
2139 004622 001372 BNE REOT2 ;WAIT DRY
2140 004624 012737 025107 000652 MOV #MSG6,EMADDR
2141 004632 004737 022772 JSR PC,PAPRT ;PRINT HEADER
2142 004636 012704 026371 MOV #MSG60,R4
2143 004642 104000 TTOUTT ;PRINT NO DRIVE READY
2144 004644 104006 STOPP
2145 004646 105337 005160 REOT2A: DECB REOTC ;SEE IF LAST UNIT TO REACH EOT
2146 004652 001410 BEQ REOT3 ;IF SO: BR
2147 004654 013700 000674 MOV UNP,R0
2148 004660 052760 100000 000750 BIS #100000,UN1(R0) ;SET EOT FLAG

```

```

2149 004666 005726          TST      (SP)+      ;RESET STACK POINTER
2150 004670 000137 004362    JMP      START7     ;GO TO NEXT UNIT
2151 004674 000337 005160    REOT3:  SWAB      REOTC
2152 004700 013700 005160    MOV      REOTC,R0
2153 004704 000337 005160    SWAB      REOTC
2154 004710 110037 005160    MOV      RO,REOTC   ;RESTORE EOT UNIT COUNTER
2155 004714 005037 000674    CLR      UNP
2156 004720 013700 000674    MOV      UNP,R0     ;POINT TO FIRST UNIT
2157 004724 016037 000750 000552 REOT4:  MOV      UN1(R0),UDES ;LOAD UNIT DESCRIPTION
2158 004732 013777 000552 173602 MOV      UDES,R02   ;LOAD COMMAND REGISTER
2159 004740 032777 020000 173554 REOT5:  BIT      #20000,R0S
2160 004746 001374          BNE      REOT5      ;AWAIT PIP RESET
2161 004750 032777 000002 173544 BIT      #2,R0S     ;SEE IF HAVE BOT
2162 004756 001012          BNE      REOT6      ;IF SO: BR
2163 004760 012700 000001    MOV      #1,R0
2164 004764 004737 022772    JSR      PC,PAPRT   ;PRINT HEADER
2165 004770 012704 026162    MOV      #MSG48,R4
2166 004774 104000          TTOUTT           ;PRINT BOT ERROR
2167 004776 104006          STOPP
2168 005000 013700 000674    MOV      UNP,R0
2169 005004 042760 100000 000750 REOT6:  BIC      #100000,UN1(R0) ;CLEAR EOT FLAG
2170 005012 062737 000002 000674 ADD      #2,UNP
2171 005020 013700 000674    MOV      UNP,R0     ;POINT TO NEXT UNIT
2172 005024 022760 177777 000750 CMP      #-1,UN1(R0) ;G BRANCH IF NOT LAST UNIT
2173 005032 001334          BNE      REOT4      ;**
2174 005034 005037 000674    REOT7:  CLR      UNP   ;CLEAR UNIT POINTER
2175 005040 005037 000634    CLR      TINF      ;CLEAR TTY INPUT FLAG
2176 005044 005737 000734    TST      ASEQF     ;SEE IF AUTO SEQ
2177 005050 001402          BEQ      REOTX     ;IF NOT: BR
2178 005052 005726          TST      (SP)+     ;RESET STACK POINTER
2179 005054 000412          BR       TEND      ;GO DO END OF PASS ** C.W
2180
2181 005056 004737 004374    REOTX:  JSR      PC,RANSET ;GO RESET RANDOM BASE
2182 005062 012737 177777 014410 MOV      #-1,PATS   ;PRESET PATTERN
2183 005070 005037 015152    CLR      R0FL      ;CLEAR RANDOM FLAG
2184 005074 005737 000570    TST      SPFLG     ;SEE IF SINGLE PASS
2185 005100 001420          BEQ      REOTXX    ;IF NOT: BR
2186 005102 012704 026772    TEND:   MOV      #MSG100,R4
2187 005106 104000          TTOUTT           ;PRINT END OF PASS
2188 005110 013700 000042    MOV      #42,R0    ;GET ACT11 RETURN ADDRESS
2189 005114 001405          BEQ      HERE      ;BRANCH IF NOT ACT11
2190 005116 000005          RESET
2191 005120 004710          #ENDAD: JSR      PC,(R0)
2192 005122 000240          NOP
2193 005124 000240          NOP
2194 005126 000240          NOP
2195 005130 005737 000734    HERE:   TST      ASEQF   ;AUTO MODE? ** C.W
2196 005134 001401          BEQ      1#        ;BRANCH - IF NO ** C.W
2197 005136 000207          RTS      PC        ;RETURN TO AUTO SEQ ** C.W
2198 005140 104006          1#:   STOPP
2199 005142 012704 026772    REOTXX: MOV      #MSG100,R4 ;GET END OF PASS MESSAGE ** C.W
2200 005146 104000          TTOUTT           ;PRINT MESSAGE ** C.W
2201 005150 005237 000744    INC      EOPB1
2202 005154 000137 003244    JMP      STARTE    ;RESTART AT BLOCK NUMBER ONE
2203 005160 000060    REOTC:  0           ;EOT UNIT COUNTER

```



```

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 005162 032777 001000 173416 RWND: BIT #1000,BSWR ;SEE IF SHOULD REWIND
2214 005170 001001 BNE RWNDA ;IF SO: BR
2215 005172 000207 RTS PC ;ELSE EXIT
2216 005174 013737 C00674 000714 RWNDA: MOV UNP,UPS ;SAVE UNIT POINTER
2217 005202 005037 000674 CLR UNP ;CLEAR POINTER
2218 005206 005037 000660 CLR EOTREC ;CLEAR EDT FLAG
2219 005212 000337 005160 SWAB REOTC
2220 005216 013700 005160 MOV REOTC,RO
2221 005222 000337 005160 SWAB REOTC
2222 005226 110037 005160 MOVB RO,REOTC ;RESTORE EOT UNIT COUNTER
2223 005232 013700 000674 RWND0: MOV UNP,RO ;POINT TO UNIT ENTRY
2224 005236 022760 177777 000750 CMP #-1,UN1(RO) ;G BRANCH IF LAST ENTRY
2225 005244 001445 BEQ RWND2 ;IF SO: BR
2226 005246 005760 000750 TST UN1(RO) ;G BRANCH IF ALREADY REWINDING
2227 005252 100433 BHI RWND1A ;G
2228 005254 016037 000750 000552 MOV UN1(RO),UDES ;SET UNIT DESCRIPTION
2229 005262 013777 000552 173252 MOV UDES,BC2 ;LOAD COMMAND REGISTER
2230 005270 012777 000011 173212 MOV #11,BC1 ;DRIVE CLEAR
2231 005276 012777 000007 173204 MOV #7,BC1 ;START REWIND
2232 005304 105777 173212 RWND1: TSTB BDS ;G WAIT FOR DRIVE READY
2233 005310 100414 BHI RWND1A ;IF DRY: BR
2234 005312 005337 000666 DEC STAL
2235 005316 001372 BNE RWND1 ;AWAIT DRY
2236 005320 012737 025107 000652 MOV #MSG6,EMADDR
2237 005326 004737 022772 JSR PC,PAPRT ;PRINT HEADER
2238 005332 012704 026512 MOV #MSG70,R4
2239 005336 104000 TROUTT
2240 005340 104006 STOPP ;PRINT NO DRIVE READY
2241 005342 042760 100000 000750 RWND1A: BIC #100000,UN1(RO) ;CLEAR EOT FLAG
2242 005350 062737 000002 000674 ADD #2,UNP ;BUMP POINTER
2243 005356 000725 BR RWND0 ;G DO NEXT UNIT
2244 005360 005037 000674 RWND2: CLR UNP ;CLEAR POINTER
2245 005364 013700 000674 RWND3: MOV UNP,RO ;POINT TO UNIT ENTRY
2246 005370 022760 177777 000750 CMP #-1,UN1(RO) ;G BRANCH IF LAST ENTRY
2247 005376 001436 BEQ RWNDX ;IF SO: BR
2248 005400 016037 000750 000552 MOV UN1(RO),UDES ;SET UNIT DESCRIPTION
2249 005406 013777 000552 173126 MOV UDES,BC2 ;LOAD COMMAND REGISTER
2250 005414 032777 020000 173100 RWND4: BIT #20000,BDS
2251 005422 001374 BNE RWND4 ;AWAIT PIP RESET
2252 005424 032777 000002 173070 BIT #2,BDS ;SEE IF HAVE BOT
2253 005432 001407 PEQ RWND6 ;IF NOT: BR
2254 005434 062737 000002 000674 RWND5: ADD #2,UNP ;BUMP POINTER
2255 005442 012777 000011 173040 MOV #11,BC1 ;DRIVE CLEAR
2256 005450 000745 BR RWND3 ;G DO NEXT UNIT
2257 005452 012700 000001 RWND6: MOV #1,RO
2258 005456 004737 022772 JSR PC,PAPRT ;PRINT HEADER
2259 005462 012704 026162 MOV #MSG48,R4

```



```

2269 ;*****
2270 ;WRITE ROUTINE:
2271 ;
2272 ;THIS ROUTINE IS USED TO WRITE ONTO TAPE THE BLOCK
2273 ;OF DATA DESCRIBED BY THE OPERATOR AND SET UP
2274 ;IN THE SEQUENCE FORMATTER. THE TAPE UNIT TO BE USED
2275 ;HAS BEEN ASSIGNED BY THE SEQUENCE FORMATTER AND
2276 ;ITS PARAMETERS SET IN A UNIT DESCRIPTION WORD.
2277 ;AS EACH RECORD OF THE BLOCK IS WRITTEN, IT IS CHECKED
2278 ;FOR STATUS ERRORS, WORD COUNT ZERO, AND CORRECT CURRENT
2279 ;MEMORY ADDRESS. IF THE WRITE OPERATION RESULTS IN
2280 ;ANY ERROR CONDITION, A WRITE RETRY OF THAT OPERATION
2281 ;MAY BE DONE BY SETTING SWITCH FOUR (4) TO A ONE (1).
2282 ;THE RETRY CONSISTS OF A BACKSPACE, ERASE FORWARD, AND
2283 ;REWRITE OF THE RECORD. (SEE WRITE RETRY SUBROUTINE)
2284 ;AFTER ALL DATA RECORDS IN THE BLOCK HAVE BEEN
2285 ;WRITTEN, THE WRITE ROUTINE WILL EXECUTE A WRITE
2286 ;TAPE MARK COMMAND IF THE TTY RESPONSE TM=1 WAS
2287 ;MADE AT INITIAL START. THE TM IS COUNTED AS TOTAL
2288 ;DATA RECORDS PLUS ONE (IE: IF 100 DATA RECORDS, TM=RECORD 101)
2289 ;IF THE WRITE OPERATION (DATA OR TM) CAUSES THE SELECTED SLAVE
2290 ;TO REACH END OF TAPE (EOT) AND THERE IS TO BE NO READING DONE,
2291 ;(SW2 AND SW3 SET TO A 1) THEN THE SLAVE IS REWOUND AND
2292 ;FLAGGED AS UNAVAILABLE FOR TESTING UNTIL ALL SLAVES HAVE
2293 ;REACHED EOT AND BEEN REWOUND AT WHICH TIME TESTING IS
2294 ;RESUMED ON ALL AVAILABLE SLAVES.
2295 ;WRITE RETRY MAY BE ALLOWED VIA CONSOLE SWITCH FOUR (4).
2296 ;ERROR CHECKING MAY BE DISALLOWED VIA CONSOLE SWITCH
2297 ;TWELVE (12).
2298 ;WRITING TO TAPE MAY BE DISALLOWED VIA CONSOLE SWITCH
2299 ;ZERO (0).
2300 ;*****

```

```

2301
2302 005524 032777 000001 173054 WRITE: BIT #1,BSWR ;SEE IF SHOULD WRITE
2303 005532 001402 BEQ WRITE
2304 005534 000137 006322 JMP WEX ;IF NOT: BR
2305 005540 013700 000554 WRTE: MOV RCNT,RO ;RO=RECORD COUNT
2306 005544 012737 025102 000652 W0: MOV #MSG5,EMADDP ;SET ERROR MSG ADDRESS
2307 005552 013777 000556 172736 MOV FMCNT,BFC ;LOAD CHAR COUNT
2308 005560 012777 027604 172726 MOV #MDATA,SBA ;SET DATA ADDR
2309 005566 112737 000060 000672 MOVB #60,MTC1 ;SET WRITE OP COMMAND
2310 005574 012737 005606 000662 MOV #W1,RTRN ;SET RETURN ADDRESS
2311 005602 000137 021220 JMP TAPG ;GO EXECUTE COMMAND
2312 005606 032777 002000 172706 W1: BIT #2000,BDS ;SEE IF EOT
2313 005614 001414 BEQ W2 ;IF NOT AT EOT: BR
2314 005616 005737 000660 TST EOTREC ;++G BRANCH IF WRITTEN PAST EOT
2315 005622 100411 BMI W2 ;++G
2316 005624 010037 000660 MOV RO,EOTREC ;SAVE EOT RECORD NUMBER
2317 005630 052737 100000 000660 BIS #100000,EOTREC ;++G SET EOT FLAG
2318 005636 005337 000660 DEC EOTREC ;++G ADJUST RECORD COUNT
2319 005642 012700 000002 MOV #2,RO ;++G SET RO TO WRITE 1 MORE RECORD
2320 005646 032777 010000 172732 W2: BIT #10000,BSWR ;SEE IF SHOULD CHECK ERRORS
2321 005654 001002 BNE W3 ;IF NOT: BR
2322 005656 004737 017410 JSR PC,ERCHK ;GO CHECK ERRORS
2323 005662 013737 000574 000666 W3: MOV WSTAL,STAL ;SET DELAY
2324 005670 004737 012306 JSR PC,STALL ;DELAY

```

2325	005674	005737	000712		TST	RTYFL		;SEE IF RETRY TIME
2326	005700	001401			BEQ	W3A		;IF NOT: BR
2327	005702	000207			RTS	PC		;ELSE RETURN
2328	005704	005737	000706	W3A:	TST	SERFL		;SEE IF WRITE ERROR
2329	005710	001450			BEQ	W5		;IF NOT: BR
2330	005712	013704	000674		MOV	UNP,R4		
2331	005716	005264	001072		INC	WTER1(R4)		;BUMP WRITE ERROR
2332	005722	005037	000706		CLR	SERFL		;CLEAR STATUS ERROR FLAG
2333	005726	032777	000020	172652	BIT	#20,BSWR		;SEE IF RETRY
2334	005734	001436			BEQ	W5		;IF NOT: BR
2335	005736	013703	000722		MOV	ERSAV,R3		
2336	005742	042703	102700		BIC	#102700,R3		;MASK UNRECOVERABLE ERROR
2337	005746	001410			BEQ	W4		;IF SO: BR
2338	005750	004737	022772		JSR	PC,PAPRT		;PRINT HEADER
2339	005754	012704	026671		MOV	#MSG78,R4		
2340	005760	104000			TTOUTT			;PRINT NON-RETRYABLE ERROR TAG
2341	005762	004737	011262		JSR	PC,NRTP		;PRINT ER FOR NON-RETRYABLE
2342	005766	000421			BR	W5		
2343	005770	013704	000674	W4:	MOV	UNP,R4		
2344	005774	005264	001052		INC	RTY1(R4)		;BUMP RETRY CNTR
2345	006000	032777	002000	172600	BIT	#2000,BSWR		;SEE IF PRINT ERRORS
2346	006006	001003			BNE	W4A		;IF NOT: BR
2347	006010	012704	026405		MOV	#MSG64,R4		
2348	006014	104000			TTOUTT			;PRINT ORIGINAL ERROR TAG
2349	006016	005037	000702	W4A:	CLR	RTCNT		;CLEAR RETRY NUMBER
2350	006022	005037	000700		CLR	RPCNT		;CLEAR REPEAT COUNTER
2351	006026	004737	006364		JSR	PC,WRTY		;GO RETRY WRITE ERROR
2352	006032	005037	000712	W5:	CLR	RTYFL		;CLEAR RETRY COUNTER
2353	006036	005300			DEC	RO		;SEE IF DONE ALL
2354	006040	001241			BNE	W0		;IF NOT: BR
2355	006042	005737	000564	W6:	TST	TMEX		;SEE IF TM
2356	006046	001525			BEQ	WEX		;IF NOT: BR
2357	006050	005237	000676		INC	TMFLG		;SET TM FLAG
2358	006054	012737	026312	000652	MOV	#MSG54,EMADDR		;POINT TO TM ERROR MSG
2359	006062	012737	000026	000672	MOV	#26,MTCL		;SET TM OP CODE
2360	006070	012777	000000	172420	MOV	#0,SFC		;LOAD FRAME COUNTER
2361	006076	012777	027604	172410	MOV	#WDATA,SBA		;LOAD BUS ADDRESS
2362	006104	012737	006116	000662	MOV	#WTMO,RTRN		;SAVE RETURN ADDRESS
2363	006112	000137	021220		JMP	TAPG		;WRITE TM
2364	006116	032777	010000	172462	WTMO:	BIT	#10000,BSWR	;SEE IF SHOULD CHECK ERRORS
2365	006124	001076			BNE	WEX		
2366	006126	032777	000004	172366	BIT	#4,SDS		;SEE IF TM STATUS
2367	006134	001011			BNE	WTM1		;IF SO: BR
2368	006136	012737	027604	021134	MOV	#WDATA,CADER		;SET EXPT BUS ADDRESS
2369	006144	012737	000001	021142	MOV	#1,DRVER		;INDICATE ERROR
2370	006152	004737	020236		JSR	PC,ERPT		;PRINT TM ERROR
2371	006156	000404			BR	WTM2		
2372	006160	012703	027604	WTM1:	MOV	#WDATA,R3		;SET EXPT ADDRESS
2373	006164	004737	017506		JSR	PC,ER2		;GO CHECK FOR OTHER ERRORS
2374	006170	005737	000712	WTM2:	TST	RTYFL		;SEE IF RETRY
2375	006174	001401			BEQ	WTM3		;IF NOT: BR
2376	006176	000207			RTS	PC		;ELSE RETURN TO RETRY ROUTINE
2377	006200	005737	000706	WTM3:	TST	SERFL		;SEE IF WRITE ERROR
2378	006204	001446			BEQ	WEX		;IF NOT: BR
2379	006206	013704	000674		MOV	UNP,R4		
2380	006212	005264	001072		INC	WTER1(R4)		;BUMP WRITE ERROR

2381	006216	032777	000020	172362	BIT	020,BSWR	;SEE IF SHOULD RETRY
2382	006224	001436			BEQ	WEX	;IF NOT: BR
2383	006226	013703	000722		MOV	ERSAV,R3	
2384	006232	042703	102700		BIC	0102700,R3	;MASK UNRECOVERABLE ERROR
2385	006236	001410			BEQ	WTM4	;IF SO: BR
2386	006240	004737	022772		JSR	PC,PAPRT	;PRINT HEADER
2387	006244	012704	026671		MOV	0MSG78,R4	
2388	006250	104000			TTOUTT		;PRINT UNRETRYABLE TAG
2389	006252	004737	011262		JSR	PC,NRTP	;PRINT ER FOR NON-RETRYABLE
2390	006256	000421			BR	WEX	
2391	006260	005037	000700	WTM4:	CLR	RPCNT	;CLEAR REPEAT CNTR
2392	006264	013704	000674		MOV	UNP,R4	
2393	006270	005264	001052		INC	RTY1(R4)	;BUMP RETRY CNTR
2394	006274	005037	000702		CLR	RTCNT	;CLEAR RETRY CNTR
2395	006300	032777	002000	172300	BIT	02000,BSWR	;SEE IF PRINT ERRORS
2396	006306	001003			BNE	WTM4A	;IF NOT: BR
2397	006310	012704	026405		MOV	0MSG64,R4	
2398	006314	104000			TTOUTT		;PRINT ORIGINAL ERROR TAG
2399	006316	004737	006364	WTM4A:	JSR	PC,WRTY	;GO DO RETRY
2400	006322	005037	000712	WEX:	CLR	RTYFL	;CLEAR RETRY FLAG
2401	006326	005037	000676		CLR	TMFLG	;CLEAR TAPE MARK FLAG
2402	006332	005737	000660		TST	EOTREC	;**G BRANCH IF NOT AT E01
2403	006336	100011			BPL	WRWX	;**G
2404	006340	017703	172242	WRW:	MOV	BSWR,R3	
2405	006344	042703	177763		BIC	0177763,R3	
2406	006350	022703	000014		CMF	014,R3	;SEE IF WRITE ONLY
2407	006354	001002			BNE	WRWX	;IF NOT: BR
2408	006356	000137	004426		JMP	REOT	;ELSE REWIND
2409	006362	000207		WRWX:	RTS	PC	;EXIT

```

2410 ;*****
2411 ;WRITE ERROR RETRY
2412 ;
2413 ;*****
2414
2415 006364 012737 000001 000712 WRTY: MOV @1,RTYFL ;SET RETRY FLAG
2416 006372 004737 006766 WRTYO: JSR PC,WRTSB ;GO SPACE REVERSE FOR REPEAT
2417 006376 005737 000676 TST TMLG ;SEE IF TAPE MARK TIME
2418 006402 001003 BNE WRTYTH ;IF SO: BR
2419 006404 004737 005544 JSR PC,W0 ;REWRITE RECORD
2420 006410 000402 BR WRTYR ;GO ON
2421 006412 004737 006054 WRTYTH: JSR PC,WTH ;GO WRITE TAPE MARK AGAIN
2422 006416 005737 000706 WRTYR: TST SERFL ;REWRITE GOOD
2423 006422 001024 BNE WRTY2 ;IF NOT: BR
2424 006424 005237 000700 INC RPCNT ;BUMP REPEAT COUNTER
2425 006430 022737 000004 000700 CMP @4,RPCNT ;SEE IF FOUR GOOD REPEATS
2426 006436 001355 BNE WRTYO ;IF NOT: REPEAT
2427 006440 032777 002000 172140 BIT @2000,@SMR ;SEE IF PRINT
2428 006446 001011 BNE WRTY1 ;IF NOT: BR
2429 006450 012704 027064 MOV @MSG105,R4
2430 006454 104000 TTOUTT ;PRINT RECOVERED MESSAGE
2431 006456 012704 026427 MOV @MSG65,R4
2432 006462 104000 TTOUTT ;PRINT RETRY TAG
2433 006464 013703 000702 MOV RTCNT,R3
2434 006470 104002 OCTPP ;PRINT RETRY NUMBER
2435 006472 000207 WRTY1: RTS PC ;RESUME TESTING
2436 006474 005037 000646 WRTY2: CLR TEMP3 ;+G CLEAR RECOVERABLE ERROR FLAG
2437 006500 013703 000722 MOV ERSAV,R3 ;GET ER
2438 006504 042703 102700 BIC @102700,R3 ;MASK RECOVERABLE BITS
2439 006510 001413 BEQ WRTY2A ;IF RECOVERABLE: BR
2440 006512 004737 022772 JSR PC,PAPRT ;PRINT HEADER
2441 006516 012704 026671 MOV @MSG78,R4
2442 006522 104000 TTOUTT ;PRINT NON-RECOVERABLE MSG
2443 006524 004737 011262 JSR PC,NRTP ;PRINT ER
2444 006530 012737 000001 000646 MOV @1,TEMP3 ;SET FLAG
2445 006536 000407 BR WRTY2B
2446 006540 032777 002000 172040 WRTY2A: BIT @2000,@SMR ;SEE IF PRINT
2447 006546 001025 BNE WRTY3 ;IF NOT: BR
2448 006550 012704 027320 MOV @MSG110,R4
2449 006554 104000 TTOUTT ;PRINT BAD TAPE SUSPECT
2450 006556 012704 026427 WRTY2B: MOV @MSG65,R4
2451 006562 104000 TTOUTT ;PRINT RETRY TAG
2452 006564 013703 000702 MOV RTCNT,R3
2453 006570 104002 OCTPP ;PRINT RETRY NUMBER
2454 006572 012704 027342 MOV @MSG111,R4
2455 006576 104000 TTOUTT ;PRINT REPEAT TAG
2456 006600 013703 000700 MOV RPCNT,R3
2457 006604 104002 OCTPP ;PRINT REPEAT NUMBER
2458 006606 005737 000646 TST TEMP3 ;SEE IF DID NON-RECOVERABLE
2459 006612 001403 BEQ WRTY3 ;IF NOT: BR
2460 006614 005037 000646 CLR TEMP3 ;CLEAR FLAG
2461 006620 000207 RTS PC ;EXIT
2462 006622 005737 000702 WRTY3: TST RTCNT ;SEE IF FIRST RETRY
2463 006626 001004 BNE WRTY3A ;IF NOT: BR
2464 006630 013704 000674 MOV UNP,R4
2465 006634 005364 001072 DEC WTER1(R4) ;DECREMENT WRITE ERROR CNTR

```

```

2466 006640 013704 000674          WRTY3A: MOV      UNP,R4          ;GET UNIT NUMBER
2467 006644 016437 001032 000730  MOV      BTADDR(R4),BTPT ;GET ADDRESS OF UNIT BAD TAPE CNTR
2468 006652 017704 172052          MOV      @BTPT,R4          ;GET COUNTER
2469 006656 005724                TST      (R4)              ;SET POINTER OFFSET
2470 006660 010477 172044          MOV      R4,@BTPT
2471 006664 013703 000730          MOV      BTPT,R3
2472 006670 060304                ADD      R3,R4              ;SET ABSOLUTE POINTER
2473 006672 013714 000654          MOV      BLCNTR,(R4)      ;SET BLOCK NUMBER
2474 006676 062704 000040          ADD      @40,R4            ;ADD RCNT OFFSET
2475 006702 013714 000554          MOV      RCNT,(R4)
2476 006706 160014                SUB      R0,(R4)          ;SET RECORD NUMBER
2477 006710 005214                INC      (R4)              ;CORRECT RECORD NUMBER
2478 006712 022777 000040 172010  CMP      @40,@BTPT        ;SEE IF TOO MANY BAD SPOTS
2479 006720 001002                BNE      WRTY4             ;IF NOT: BR
2480 006722 000137 007162          JMP      BTOV              ;ELSE GO TO BAD TAPE OVERFLOW
2481 006726 005237 000702          WRTY4:  INC      RTCNT      ;BUMP RETRY COUNTER
2482 006732 022737 000004 000702  CMP      @4,RTCNT         ;SEE IF DONE 4 RETRIES
2483 006740 001410                BEQ      WRTY5             ;IF SO: BR
2484 006742 013704 000674          MOV      UNP,R4
2485 006746 005264 001052          INC      RTY1(R4)         ;BUMP RETRY COUNTER
2486 006752 005237 000732          INC      ERTFL            ;SET ERASE FLAG
2487 006756 000137 006372          JMP      WRTY0            ;DO NEXT RETRY
2488 006762 000137 007402          WRTY5:  JMP      BTUR              ;ELSE GO TO BAD TAPE UNRECOVERABLE
2489
2490          ;WRITE RETRY BACKSPACE-ERASE SUBROUTINE*****
2491
2492 006766 005037 000706          WRTSB:  CLR      SERFL          ;CLEAR FLAG
2493 006772 013737 000576 000666  MOV      TSTAL,STAL
2494 007000 004737 012306          JSR      PC,STALL          ;DO TURN AROUND DELAY
2495 007004 012737 026440 000652  MOV      @MSG66,EMADDR    ;SET ERROR CODE
2496 007012 012777 177777 171476  MOV      @-1,@FC          ;SET TO BACKSPACE 1 RECORD
2497 007020 012777 033612 171466  MOV      @RDATA,@BA       ;SET BA
2498 007026 004737 012236          JSR      PC,BKRT          ;GO BACKSPACE
2499 007032 005737 000706          TST      SERFL            ;SEE IF ERROR
2500 007036 001406                BEQ      WRTSB1           ;IF NOT: BR
2501 007040 012737 000002 000724  WRTSB0: MOV      @2,BTFLG      ;SET FLAG
2502 007046 022626                CMP      (SP),,(SP)       ;RESET STACK
2503 007050 000137 004426          JMP      REOT              ;GO REWIND AND REMOVE FROM TESTING
2504 007054 005737 000732          WRTSB1: TST      ERTFL      ;SEE IF SHOULD ERASE
2505 007060 001001                BNE      WRTSB2           ;IF SO: BR
2506 007062 000207                RTS                        ;RETURN
2507 007064 005037 000732          WRTSB2: CLR      ERTFL      ;CLEAR ERASE FLAG
2508 007070 005037 000700          CLR      RPCNT            ;CLEAR REPEAT CNTR
2509 007074 005037 000706          CLR      SERFL            ;CLEAR FLAG
2510 007100 012737 026453 000652  MOV      @MSG67,EMADDR    ;SET ERROR CODE
2511 007106 005077 171404          CLR      @FC              ;CLEAR FRAME COUNT
2512 007112 012737 000024 000672  MOV      @24,HTC1         ;SET ERASE OP-CODE
2513 007120 012777 027604 171366  MOV      @WDATA,@BA       ;SET BA
2514 007126 012737 007140 000662  MOV      @WRTSB3,RTRN     ;SET RETURN ADDRESS
2515 007134 000137 021220          JMP      TAPG              ;GO ERASE
2516 007140 012703 027604          WRTSB3: MOV      @WDATA,R3     ;SET EXPT BA
2517 007144 004737 017506          JSR      PC,ER2           ;GO CHECK ERRORS
2518 007150 005737 000706          TST      SERFL            ;SEE IF ERROR
2519 007154 001737                BEQ      WRTSB1           ;IF NOT: BR
2520 007156 000137 007040          JMP      WRTSB0
2521

```

```

2522                                     ;BAD TAPE OVERFLOW SUBROUTINE*****
2523
2524 007162 005037 000712          BTOV: CLR      RTYFL          ;CLEAR RETRY FLAG
2525 007166 012737 000001 000724  MOV     #1,BTFLG        ;SET BAD TAPE OVERFLOW FLAG
2526 007174 005726                TST     (SP)+          ;++G ADJUST STACK
2527 007176 000137 004426          JMP     REOT           ;GO REWIND AND REMOVE FROM TESTING
2528 007202 013701 000730          BTOV0: MOV     BTPT,R1    ;SET TABLE POINTER
2529 007206 005721                TST     (R1)+
2530 007210 005000                CLR     RO
2531 007212 010003          BTOV1: MOV     RO,R3
2532 007214 000241                CLC
2533 007216 006003                ROR     R3             ;R3=R3/2 FOR CORRECT NUMBER
2534 007220 104002                OCTPP   ;PRINT ENTRY NUMBER
2535 007222 012704 025174          MOV     #MSG13,R4
2536 007226 105724                TSTB   (R4)+          ;SKIP CR/LF
2537 007230 104000                TTOUTT ;PRINT BLOCK NUMBER TAG
2538 007232 011103                MOV     (R1),R3
2539 007234 104002                OCTPP   ;PRINT BLOCK NUMBER
2540 007236 012704 025202          MOV     #MSG14,R4
2541 007242 104000                TTOUTT ;PRINT RECORD NUMBER TAG
2542 007244 062701 000040          ADD     #40,R1        ;SET POINTER OFFSET FOR RECOED NUMBER
2543 007250 012103                MOV     (R1)+,R3
2544 007252 104002                OCTPP   ;PRINT RECORD NUMBER
2545 007254 162701 000040          SUB     #40,R1        ;RESET POINTER FOR BLOCK NUMBER
2546 007260 005720                TST     (R0)+
2547 007262 020077 171442          CMP     RO,GBTPT      ;SEE IF DONE
2548 007266 001404                BEQ    BTOV2          ;IF SO: BR
2549 007270 012704 025525          MOV     #MSG28,R4
2550 007274 104000                TTOUTT ;DO CR/LF
2551 007276 000745                BR     BTOV1          ;CONTINUE
2552 007300 005737 000726          BTOV2: TST     BTSTF    ;SEE IF STAT ONLY PRINT
2553 007304 001007                BNE    BTOVX          ;IF SO: BR
2554 007306 012703 000041          MOV     #41,R3        ;SET SIZE OF TABLE
2555 007312 013704 000730          MOV     BTPT,R4      ;SET POINTER
2556 007316 005024          BTOV3: CLR     (R4)+   ;CLEAR TABLE
2557 007320 005303                DEC     R3            ;SEE IF DONE
2558 007322 001375                BNE    BTOV3          ;IF NOT: BR
2559 007324 000207          BTOVX: RTS     PC      ;RETURN
2560

```



```

2561
2562
2563
2564 007326 012704 025525
2565 007332 104000
2566 007334 015704 000674
2567 007340 016437 001032 000730
2568 007346 017703 171356
2569 007352 000241
2570 007354 006003
2571 007356 104002
2572 007360 012704 027354
2573 007364 104000
2574 007366 005777 171336
2575 007372 001001
2576 007374 000207
2577 007376 000137 007202
2578
2579
2580
2581 007402 004737 022772
2582 007406 012704 027167
2583 007412 104000
2584 007414 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 BTPT1 ;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

```

```

2586 ;*****
2587 ;READ SEQUENCER:
2588 ;
2589 ;THIS ROUTINE IS USED TO DETERMINE THE SEQUENCE
2590 ;IN WHICH READ TAPE OPERATIONS ARE TO BE PERFORMED.
2591 ;THIS IS NECESSARY WHEN THE UNIT BEING TESTED IS
2592 ;CAPABLE OF READING DATA IN BOTH THE FORWARD AND
2593 ;REVERSE DIRECTIONS. CONSOLE SWITCHES ONE (1), TWO (2),
2594 ;AND THREE (3) ARE USED TO DETERMINE THE READ SEQUENCE.
2595 ;CONSOLE SWITCH ONE (1) DETERMINES WHETHER TO READ
2596 ;THE BLOCK OF DATA FORWARD FIRST OR REVERSE FIRST.
2597 ;SWITCH TWO (2) DISALLOWS READING IN THE REVERSE
2598 ;DIRECTION AND SWITCH THREE (3) DISALLOWS READING IN
2599 ;THE FORWARD DIRECTION.
2600 ;*****
2601
2602 007416 012737 000002 000562 RSEQ: MOV #2,RDCMD
2603 007424 017704 171156 MOV BSWR,R4 ;READ SWITCHES
2604 007430 042704 177763 BIC #177763,R4 ;MASK READ BITS
2605 007434 005704 TST R4 ;SEE IF BOTH READS
2606 007436 001004 BNE RSR ;IF NOT: BR
2607 007440 032777 000002 171140 BIT #2,BSWR ;SEE IF READ REVERSE FIRST
2608 007446 001051 BNE RSFR ;IF NOT: BR
2609 007450 032777 000004 171130 RSR: BIT #4,BSWR ;SEE IF SHOULD READ REVERSE
2610 007456 001005 BNE RSF ;IF NOT: BR
2611 007460 012737 010000 000562 MOV #10000,RDCMD ;LOAD READ REVERSE COMMAND
2612 007466 004737 007734 JSR PC,READ ;GO READ REVERSE
2613 007472 032777 000010 171106 RSF: BIT #10,BSWR ;SEE IF SHOULD READ FORWARD
2614 007500 001026 BNE RSEX ;IF NOT: BR
2615 007502 032737 010000 000562 BIT #10000,RDCMD ;SEE IF HAVE READ REVERSE
2616 007510 001407 BEQ RSFO ;IF NOT: BR
2617 007512 013737 000576 000666 MOV TSTAL,STAL
2618 007520 004737 012306 JSR PC,STALL ;DO READ STALL
2619 007524 000137 007544 JMP RSF1
2620 007530 032777 000001 171050 RSFO: BIT #1,BSWR ;SEE IF WRITE
2621 007536 001002 BNE RSF1 ;IF NOT: BR
2622 007540 004737 012060 JSR PC,BKSP ;GO BACKSPACE
2623 007544 012737 000002 000562 RSF1: MOV #2,RDCMD ;LOAD READ FORWARD COMMAND
2624 007552 004737 007734 JSR PC,READ ;GO READ
2625 007556 005737 000660 RSEX: TST EOTREC ;++G BRANCH IF NOT AT EOT
2626 007562 100002 BPL 1# ;++G
2627 007564 000137 004426 JMP REOT ;++G ELSE GO REWIND
2628 007570 000207 1#: RTS ;++G EXIT
2629
2630 007572 012737 010000 000562 RSFR: MOV #10000,RDCMD
2631 007600 032777 000010 171000 BIT #10,BSWR ;SEE IF SHOULD READ FORWARD
2632 007606 001013 BNE RSFR1 ;IF NOT: BR
2633 007610 032777 000001 170770 BIT #1,BSWR ;SEE IF WRITE
2634 007616 001002 BNE RSFR0 ;IF NOT: BR
2635 007620 004737 012060 JSR PC,BKSP ;GO BACKSPACE TO START
2636 007624 012737 000002 000562 RSFR0: MOV #2,RDCMD ;LOAD READ FORWARD COMMAND
2637 007632 004737 007734 JSR PC,READ ;GO READ FORWARD
2638 007636 032777 000004 170742 RSFR1: BIT #4,BSWR ;SEE IF SHOULD READ REVERSE
2639 007644 001344 BNE RSEX ;IF NOT: BR
2640 007646 032737 010000 000562 BIT #10000,RDCMD
2641 007654 001005 BNE RSFR2 ;IF READ REVERSE: BR

```

2642	007656	013737	000576	000666	MOV	TSTAL,STAL	;DO READ STALL
2643	007664	004737	012306		JSR	PC,STALL	
2644	007670	012737	010000	000562	RSFR2: MOV	@10000,RDCMD	;LOAD READ REVERSE
2645	007676	004737	007734		JSR	PC,READ	;GO READ REVERSE
2646	007702	005737	000660		TST	EOTREC	;SEE IF AT END OF TAPE
2647	007706	100011			BPL	RSFRX	;**G IF NOT: BR
2648	007710	163737	000554	000660	SUB	RCNT,EOTREC	
2649	007716	005437	000660		NEG	EOTREC	;SET TO PROPER RECORD NUMBER
2650	007722	005237	000660		INC	EOTREC	
2651	007726	000137	004426		JMP	REOT	;ELSE GO TO REWIND
2652	007732	000207			RSFRX: RTS	PC	;EXIT
2653							

```

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 007734 013700 000554 READ: MOV RCNT,R0 ;LOAD REC CNTR
2681 007740 005737 000660 TST EOTREC ;SEE IF EOT
2682 007744 100013 BPL RDA ;IF NOT: BR
2683 007746 032737 010000 000562 BIT #10000,RDCMD ;SEE IF READ FORWARD
2684 007754 001407 BEQ RDA ;IF SO: BR
2685 007756 042737 100000 000660 BIC #100000,EOTREC ;CLEAR FLAG
2686 007764 013703 000660 MOV EOTREC,R3 ;GET MODIFIED RECORD COUNT
2687 007770 160300 SUB R3,R0 ;SET RECORD AT
2688 007772 005200 INC R0 ;SET TO PROPER NUMBER OF RECORDS
2689 007774 012737 025107 000652 RDA: MOV #MSG6,EMADDR ;SET ERROR MSG ADDRESS
2690 010002 005037 000676 CLR TMFLG
2691 010006 032737 010000 000562 BIT #10000,RDCMD
2692 010014 001406 BEQ RDO ;IF READ FORWARD: BR
2693 010016 005737 000564 TST TMEX ;SEE IF TM
2694 010022 001403 BEQ RDO ;IF NOT: BR
2695 010024 005237 000676 INC TMFLG ;SET TM FLAG
2696 010030 005200 INC R0
2697 010032 013777 000556 170456 RDO: MOV FMCNT,#FC ;LOAD CHAR CNTR
2698 010040 012777 033612 170446 MOV #RDATA,#BA ;LOAD DATA ADDR
2699 010046 032737 010000 000562 BIT #10000,RDCMD ;SEE IF READ REVERSE
2700 010054 001417 BEQ RD1A ;IF NOT: BR
2701 010056 013703 000556 MOV FMCNT,R3
2702 010062 005103 COM R3
2703 010064 032737 000020 000552 BIT #20,UDES ;SEE IF CORE DUMP
2704 010072 001402 BEQ RD1 ;IF NOT: BR
2705 010074 000241 CLC
2706 010076 006003 ROR R3 ;R3 = FC/2
2707 010100 060377 170410 RD1: ADD R3,#BA ;SET REVERSE BUS ADDRESS
2708 010104 012737 000076 000672 MOV #76,MTC1 ;SET READ REVERSE
2709 010112 000403 BR RD1B

```

2710	010114	012737	000070	000672	RD1A:	MOV	#70,MTC1	;SET READ FORWARD
2711	010122	012737	010134	000662	RD1B:	MOV	#RD2,RTRN	;SET INTERRUPT RETURN ADDRESS
2712	010130	000137	021220		RD1D:	JMP	TAPG	;GO EXECUTE TAPE COMMAND
2713	010134	032737	010000	000562	RD2:	BIT	#10000,RDCMD	;SEE IF READ REVERSE
2714	010142	001024				BNE	RD3	;IF SO: BR
2715	010144	032777	000020	170350		BIT	#20,SOS	
2716	010152	001404				BEQ	RD2B	;AWAIT SWDN
2717	010154	032777	000020	170340	RD2A:	BIT	#20,SOS	
2718	010162	001374				BNE	RD2A	;AWAIT TUR
2719	010164	032777	002000	170330	RD2B:	BIT	#2000,SOS	;SEE IF EOT
2720	010172	001410				BEQ	RD3	;IF NOT: BR
2721	010174	005737	000676			TST	TMFLG	;SEE IF TM
2722	010200	001005				BNE	RD3	;IF SO: BR
2723	010202	010037	000660			MOV	R0,EOTPEC	
2724	010206	052737	100000	000660		BIS	#100000,EOTREC	;SET EOT FLAG
2725	010214	032777	000002	170300	RD3:	BIT	#2,SOS	;SEE IF AT LOAD POINT
2726	010222	001410				BEQ	RD4	;IF NOT: BR
2727	010224	004737	022772			JSR	PC,PAPRT	;PRINT CYCLE NUMBER
2728	010230	012704	025306			MOV	#MSG22,R4	
2729	010234	104000				TTOUTT		;PRINT BOT ERROR
2730	010236	104006				STOPP		
2731	010240	000137	003162			JMP	STARTA	;RESTART
2732	010244	032777	004000	170334	RD4:	BIT	#4000,BSWR	;SEE IF SHOULD CHECK ERRORS
2733	010252	001121				BNE	RD5	;IF NOT: BR
2734	010254	005737	000676			TST	TMFLG	
2735	010260	001472				BEQ	RD4B	;IF NO TM EXPT: BR
2736	010262	032777	000004	170232		BIT	#4,SOS	
2737	010270	001024				BNE	RD4A	;IF TM RECVD: BR
2738	010272	012737	033612	021134		MOV	#RDATA,CADER	;SAVE EXPT BUS ADDRESS
2739	010300	012737	000002	021142		MOV	#2,DRVER	;SET TM STATUS ERROR FLAG
2740	010306	004737	020236			JSR	PC,ERPT	;GO PRINT TM ERROR
2741	010312	013704	000674			MOV	UNP,R4	
2742	010316	032737	010000	000562		BIT	#10000,RDCMD	;SEE IF READ REVERSE
2743	010324	001403				BEQ	18	;IF NOT: BR
2744	010326	005264	001152			INC	RDERR1(R4)	;BUMP READ REVERSE ERROR
2745	010332	000502				BR	RD6	
2746	010334	005264	001112		18:	INC	RDER1(R4)	;BUMP READ FORWARD ERROR
2747	010340	000477				BR	RD6	
2748	010342	012703	033612		RD4A:	MOV	#RDATA,R3	
2749	010346	032737	010000	000562		BIT	#10000,RDCMD	;SEE IF READ REVERSE
2750	010354	001007				BNE	RD4A0	;IF SO: BR
2751	010356	032737	002000	000552		BIT	#2000,UDES	;SEE IF IN PE
2752	010364	001025				BNE	RD4A2	;IF SO: BR
2753	010366	062703	000002			ADD	#2,R3	
2754	010372	000422				BR	RD4A2	
2755	010374	013704	000556		RD4A0:	MOV	FMCNT,R4	
2756	010400	005104				COM	R4	
2757	010402	032737	000020	000552		BIT	#20,UDES	;SEE IF CORE DUMP
2758	010410	001402				BEQ	RD4A1	;IF NOT: BR
2759	010412	000241				CLC		
2760	010414	006004				ROR	R4	;SET TO FC/2
2761	010416	060403			RD4A1:	ADD	R4,R3	;SET EXPT BUS ADDRESS
2762	010420	042703	000001			BIC	#1,R3	;MAKE EXPT ADDRESS EVEN
2763	010424	032737	002000	000552		BIT	#2000,UDES	;SEE IF IN PE
2764	010432	001002				BNE	RD4A2	;IF SO: BR
2765	010434	162703	000002			SUB	#2,R3	

2766	010440	004737	017506		RD4A2:	JSR	PC,ER2	
2767	010444	000402				BR	RD4C	
2768	010446	004737	017410		RD4B:	JSR	PC,ERCHK	;GO CHECK ERRORS
2769	010452	005737	000706		RD4C:	TST	SERFL	
2770	010456	001417				BEQ	RD5	;IF NO ERROR: BR
2771	010460	013704	000674			MOV	UNP,R4	
2772	010464	032737	010000	000562		BIT	#10000,RDCMD	;SEE IF READ REVERSE
2773	010472	001003				BNE	RD4D	;IF SO: BR
2774	010474	005264	001112			INC	RDER1(R4)	;BUMP READ FORWARD ERROR
2775	010500	000402				BR	RD4E	
2776	010502	005264	001152		RD4D:	INC	RDERR1(R4)	;BUMP READ REVERSE ERROR
2777	010506	004737	010710		RD4E:	JSR	PC,RDR1Y	;GO RETRY
2778	010512	005037	000712			CLR	RTYFL	;CLEAR RETRY FLAG
2779	010516	032777	020000	170062	RD5:	BIT	#20000,@SWR	;SEE IF SHOULD DO DATA CHECK
2780	010524	001005				BNE	RD6	;IF NOT: BR
2781	010526	005737	000676			TST	TMFLG	
2782	010532	001002				BNE	RD6	
2783	010534	004737	015546			JSR	PC,DCHK	;GO CHECK DATA
2784	010540	005037	000706		RD6:	CLR	SERFL	;CLEAR STATUS ERROR FLAG
2785	010544	004737	014354			JSR	PC,DS3	;CLEAR BUFFER
2786	010550	032777	000040	170030		BIT	#40,@SWR	;SEE IF SHOULD YOZZLE
2787	010556	001402				BEQ	RD7	;IF NOT: BR
2788	010560	004737	011276			JSR	PC,YOZ	;ELSE GO YOZZLE
2789	010564	013737	000572	000666	RD7:	MOV	RSTAL,STAL	;SET DELAY
2790	010572	004737	012306			JSR	PC,STALL	;STALL
2791	010576	032737	010000	000562		BIT	#10000,RDCMD	;SEE IF READ REVERSE
2792	010604	001403				BEQ	RD7A	;IF NOT: BR
2793	010606	005037	000676			CLR	TMFLG	;CLEAR TAPE MARK FLAG
2794	010612	009405				BR	RD10	
2795	010614	005737	000660		RD7A:	TST	EOTREC	;SEE IF EOT FOUND
2796	010620	100002				BPL	RD10	;IF NOT: BR
2797	010622	012700	000001			MOV	#1,R0	;SET TO EOT
2798	010626	005300			RD10:	DEC	R0	
2799	010630	001402				BEQ	RD11	;IF DONE ALL: BR
2800	010632	000137	010032			JMP	R00	
2801	010636	032737	010000	000562	RD11:	BIT	#10000,RDCMD	;SEE IF READ REVERSE
2802	010644	001016				BNE	RDEX	;IF SO: BR
2803	010646	005737	000660			TST	EOTREC	;SEE IF FOUND EOT
2804	010652	100413				BMI	RDEX	;IF SO: BR
2805	010654	005737	000564			TST	TMEX	;SEE IF TM EXPECTED
2806	010660	001410				BEQ	RDEX	;IF NOT: BR
2807	010662	005737	000676			TST	TMFLG	;SEE IF TM FOUND
2808	010666	001005				BNE	RDEX	;IF SO: BR
2809	010670	005237	000676			INC	TMFLG	;ELSE SET FLAG
2810	010674	005200				INC	R0	;SET RECORD COUNT TO ONE
2811	010676	000137	010032			JMP	R00	;GO READ TM
2812	010702	005037	000676		RDEX:	CLR	TMFLG	
2813	010706	000207			RDX:	RTS	PC	;EXIT

```
2814 ;*****
2815 ;READ ERROR RETRY SUBROUTINE:
2816 ;
2817 ;THIS SUBROUTINE WILL RETRY ALL DATA RELATED
2818 ;READ ERRORS UP TO EIGHT (8) TIMES. IF ALL
2819 ;FOUR RETRIES ARE BAD, IT IS CONSIDERED
2820 ;A HARD ERROR. IF ANY ARE GOOD, IT IS A
2821 ;SOFT ERROR. RETRIES MAY BE INHIBITED
2822 ;VIA SWITCH FOUR (SW4=0: INHIBIT RETRIES)
2823 ;*****
2824
2825 010710 032777 000020 167670 RDRTY: BIT #20,BSWR ;SEE IF RETRY INHIBITED
2826 010716 001001 BNE RDRT0 ;IF NOT: BR
2827 010720 000207 RTS PC ;ELSE RETURN
2828 010722 013703 000722 RDRT0: MOV ERSAV,R3
2829 010726 042703 102700 BIC #102700,R3 ;MARK NON-RECOVERABLE ERROR BITS
2830 010732 001410 BEQ RDRT1 ;IF NOT: BR
2831 010734 004737 022772 JSR PC,PAPRT ;PRINT HEADER
2832 010740 012704 026732 MOV #MSG79,R4
2833 010744 104000 TTOUTT ;PRINT NON-RECOVERABLE MESSAGE
2834 010746 004737 011262 JSR PC,NRTP ;PRINT ER FOR NON-RETRYABLE ERROR
2835 010752 000207 RDRT1A: RTS PC ;RETURN
2836 010754 032777 002000 167624 RDRT1: BIT #2000,BSWR ;SEE IF PRINT INHIBITED
2837 010762 001003 BNE RDRT1B ;IF SO: BR
2838 010764 012704 026405 MOV #MSG64,R4
2839 010770 104000 TTOUTT ;PRINT ORIGINAL ERROR TAG
2840 010772 005037 000702 RDRT1B: CLR RTCNT ;CLEAR RETRY COUNTER
2841 010776 005037 000706 RDRTG: CLR SERFL ;CLEAR STATUS ERROR FLAG
2842 011002 012737 000002 000712 MOV #2,RTYFL ;SET READ RETRY FLAG
2843 011010 004737 011276 JSR PC,YOZ ;GO TO YOZZLE TO RETRY READ
2844 011014 005737 000706 TST SERFL ;SEE IF RETRY ERROR
2845 011020 001031 BNE RDRT5 ;IF SO: BR
2846 011022 032777 002000 167556 BIT #2000,BSWR
2847 011030 001011 BNE RDRT2
2848 011032 012704 027064 MOV #MSG105,R4
2849 011036 104000 TTOUTT ;PRINT RECOVERED MESSAGE
2850 011040 012704 026427 MOV #MSG65,R4
2851 011044 104000 TTOUTT ;PRINT RETRY TAG
2852 011046 013703 000702 MOV RTCNT,R3
2853 011052 104002 GCTPP ;PRINT RETRY NUMBER
2854 011054 013704 000674 RDRT2: MOV UNP,R4
2855 011060 032737 010000 000562 BIT #10000,RDCHD ;SEE IF READ REVERSE
2856 011066 001003 BNE RDRT3 ;IF SO: BR
2857 011070 005264 002672 INC RFSOFT(R4) ;ELSD BUMP FORWARD SOFT ERROR COUNTER
2858 011071 000402 BR RDRT4
2859 011076 005264 002712 RDRT3: INC RRSOFT(R4) ;BUMP ERRORS SOFT CNTR
2860 011102 000207 RDRT4: RTS PC ;RETURN
2861 011104 005037 000646 RDRT5: CLR TEMP3 ;++G CLEAR RECOVERABLE ERROR INDICATOR
2862 011110 013703 000722 MOV ERSAV,R3 ;GET ER
2863 011114 042703 102700 BIC #102700,R3 ;MASK RECOVERABLE BITS
2864 011120 001413 BEQ RDRT5A ;IF RECOVERABLE: BR
2865 011122 004737 022772 JSR PC,PAPRT ;PRINT HEADER
2866 011126 012704 026732 MOV #MSG79,R4
2867 011132 104000 TTOUTT ;PRINT NON-RECOVERABLE MSG
2868 011134 004737 011262 JSR PC,NRTP ;PRINT ER
2869 011140 012737 000001 000646 MOV #1,TEMP3 ;SET FLAG
```

2870	011146	000404			BR	RDR15B	
2871	011150	032777	002000	167430	RDRT5A: BIT	#2000,BSWR	;SEE IF PRINT INHIBITED
2872	011156	001014			BNE	RDRT6	;IF SO: BR
2873	011160	012704	026427		RDRT5B: MOV	#MSG65,R4	
2874	011164	104000			TTOUTT		;PRINT RETRY TAG
2875	011166	013703	000702		MOV	RTCNT,R3	
2876	011172	104002			OCTPP		;PRINT RETRY NUMBER
2877	011174	005737	000646		TST	TEMP3	;SEE IF DID NON-RECOVERABLE
2878	011200	001403			BEQ	RDRT6	;IF NOT: BR
2879	011202	005037	000646		CLR	TEMP3	;CLEAR FLAG
2880	011206	000207			RTS	PC	;EXIT
2881	011210	005237	000702		RDRT6: INC	RTCNT	
2882	011214	023737	000702	000602	CMP	RTCNT,RETRY	;SEE IF DONE 8 RETRIES
2883	011222	001265			BNE	RDRTG	;IF NOT: BR
2884	011224	012704	027417		MOV	#MSG115,R4	
2885	011230	104000			TTOUTT		;PRINT HARD ERROR MESSAGE
2886	011232	013704	000674		MOV	UNP,R4	
2887	011236	032737	010000	000562	BIT	#10000,RDCMD	;SEE IF READ REVERSE
2888	011244	001003			BNE	RDRT7	;IF SO: BR
2889	011246	005264	002732		INC	RFHARD(R4)	;BUMP FORWARD HARD ERROR CNTR
2890	011252	000402			BR	RDRTX	
2891	011254	005264	002752		RDRT7: INC	RFHARD(R4)	;BUMP REVERSE HARD ERROR CNTR
2892	011260	000207			RDRTX: RTS	PC	;RETURN
2893							
2894	011262	013703	000722		NRTP: MOV	ERSAV,R3	;GET ER REGISTER
2895	011266	104002			OCTPP		;PRINT ER
2896	011270	004737	021160		JSR	PC,FRPRT	;PRINT F CR R
2897	011274	000207			RTS	PC	;RETURN


```

2898 ;*****
2899 ;YOZZLE SUBROUTINE:
2900 ;
2901 ;THIS SUBROUTINE, ENTERED VIA SWITCH FIVE (5), IS USED TO PERFORM
2902 ;A CONTINUOUS READ AND SPACE OVER OF THE CURPENT RECORD ON TAPE.
2903 ;FULL STATUS AND DATA CHECKING MAY BE PERFORMED
2904 ;OR NOT VIA CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13).
2905 ;A SOFTWARE DELAY IS PERFORMED BETWEEN EACH READ
2906 ;AND SPACE OPERATION AND MAY BE VARIED BY TYPING
2907 ;CNTRL C ON THE TTY AND ENTERING A VALUE IN RESPONSE
2908 ;TO THE PRINTED REQUEST.
2909 ;*****
2910
2911 011276 012777 000001 167304 YOZ:  MOV  #1,BTKS      ;SET TTY ENABLE
2912 011304 013737 000600 000666      MOV  YSTAL,STAL
2913 011312 004737 012306      JSR  PC,STALL      ;DO YOZZLE STALL
2914 011316 012777 177777 167172 YOZO:  MOV  #-1,BFC      ;SET TO 1 RECORD SPACING
2915 011324 032737 010000 000562      BIT  #10000,RDCMD  ;SEE IF READ REVERSE
2916 011332 001404      BEQ  YOZA          ;IF NOT: BR
2917 011334 112737 000030 000672      MOVB #30,MTC1     ;SET TO SPACE FORWARD
2918 011342 000403      BR   YOZB
2919 011344 112737 000032 000672 YOZA:  MOVB #32,MTC1     ;SET TO SPACE REVERSE
2920 011352 012737 011372 000662 YOZB:  MOV  #YOZC,RTRN    ;SET RETURN ADDRESS
2921 011360 012737 177775 000666      MOV  #177775,STAL ;SET TIME MULTIPLIER
2922 011366 000137 021220      JMP  TAPG         ;GO YOZZLE
2923 011372 005737 000676      YOZC: TST  TMFLG      ;SEE IF TH
2924 011376 001404      BEQ  1#          ;IF NOT: BR
2925 011400 012737 040000 000666      MOV  #40000,STAL ;SET TH STALL
2926 011406 000403      BR   2#
2927 011410 013737 000600 000666 1#:  MOV  YSTAL,STAL
2928 011416 004737 012306      2#:  JSR  PC,STALL      ;DO YOZZLE STALL
2929 011422 012777 033612 167064      MOV  #RDATA,BBA   ;SET BUS ADDRESS
2930 011430 032737 010000 000562      BIT  #10000,RDCMD ;SEE IF READ REVERSE
2931 011436 001417      BEQ  YOZC1        ;IF NOT: BR
2932 011440 013703 000556      MOV  FMCNT,R3
2933 011444 005103      COM  R3
2934 011446 032737 000020 000552      BIT  #20,UDES     ;SEE IF CORE DUMP
2935 011454 001402      BEQ  YOZCO        ;IF NOT: BR
2936 011456 000241      CLC
2937 011460 006003      ROR  R3           ;R3 = FC/2
2938 011462 060377 167026      YOZCO: ADD  R3,BBA      ;SET REVERSE BUS ADDRESS
2939 011466 012737 000076 000672      MOV  #76,MTC1     ;SET READ REVERSE
2940 011474 000403      BR   YOZC2
2941 011476 012737 000070 000672 YOZC1: MOV  #70,MTC1     ;SET READ FORWARD
2942 011504 013777 000556 167004 YOZC2: MOV  FMCNT,BFC    ;SET CHARACTER COUNT
2943 011512 012737 011524 000662      MOV  #YOZD,RTRN   ;SET RETURN ADDRESS
2944 011520 000137 021220      JMP  TAPG         ;GO READ
2945 011524 032777 004000 167054 YOZD:  BIT  #4000,BSWR   ;SEE IF SHOULD CHECK ERRORS
2946 011532 001051      BNE  YOZE        ;IF NOT: BR
2947 011534 005737 000676      TST  TMFLG      ;SEE IF TAPE MARK TIME
2948 011540 001444      BEQ  YOZD1        ;IF NOT: BR
2949 011542 032737 010000 000562      BIT  #10000,RDCMD ;SEE IF READ REVERSE
2950 011550 001426      BEQ  YOZDO        ;IF NOT: BR
2951 011552 012703 033612      MOV  #RDATA,R3
2952 011556 013704 000556      MOV  FMCNT,R4
2953 011562 005104      COM  R4

```

2954	011564	032737	000020	000552	BIT	#20,UDES	;SEE IF CORE DUMP
2955	011572	001402			BEQ	YOZD4	;IF NOT: BR
2956	011574	000241			CLC		
2957	011576	006004			ROR	R4	;SET TO FC/2
2958	011600	060403			YOZD4: ADD	R4,R3	;SET EXPT BUS ADDRESS
2959	011602	042703	000001		BIC	#1,R3	;MAKE EXPT ADDRESS EVEN
2960	011606	032737	002000	000552	BIT	#2000,UDES	;SEE IF PE
2961	011614	001001			BNE	YOZD2	;IF SO: BR
2962	011616	005743			TST	-(R3)	;SET EXPT BA
2963	011620	004737	017506		YOZD2: JSR	PC,ER2	;GO CHECK ERRORS
2964	011624	000430			BR	YOZF	
2965	011626	012703	033612		YOZD0: MOV	#RDATA,R3	
2966	011632	032737	002000	000552	BIT	#2000,UDES	;SEE IF PE
2967	011640	001001			BNE	YOZD3	;IF SO: BR
2968	011642	005723			TST	(R3)	;SET EXPT BA
2969	011644	004737	017506		YOZD3: JSR	PC,ER2	;GO CHECK ERRORS
2970	011650	000416			BR	YOZF	
2971	011652	004737	017410		YOZD1: JSR	PC,ERCHK	;ELSE GO CHECK ERRORS
2972	011656	032777	020000	166722	YOZE: BIT	#20000,BSWR	;SEE IF SHOULD CHECK DATA
2973	011664	001010			BNE	YOZF	;IF NOT: BR
2974	011666	005737	000676		TST	TMFLG	;SEE IF TAPE MARK
2975	011672	001005			BNE	YOZF	;IF SO: BR
2976	011674	005737	000712		TST	RTYFL	;SEE IF RETRY
2977	011700	001004			BNE	YOZFO	;IF SO: BR
2978	011702	004737	015546		JSR	PC,DCHK	;ELSE GO CHECK DATA
2979	011706	004737	014354		YOZF: JSR	PC,DS3	;GO CLEAR DATA AREA
2980	011712	105777	166672		YOZFO: TSTB	#TKS	;SEE IF HAVE NEW STALL VALUE
2981	011716	100032			BPL	YOZG	;IF NOT: BR
2982	011720	122777	000203	166664	CMPB	#203,#TKB	;SEE IF CONT C
2983	011726	001026			BNE	YOZG	;IF NOT: BR
2984	011730	012704	026107		MOV	#MSG44,R4	
2985	011734	104000			TTOUTT		;PRINT YSTALL REQUEST
2986	011736	013703	000600		MOV	YSTAL,R3	
2987	011742	104002			OCTPP		;PRINT PRESENT STALL
2988	011744	010037	000646		MOV	RO,TEMP3	;SAVE RO(REC CNT)
2989	011750	012705	000600		MOV	#YSTAL,R5	;SET ADDRESS OF YSTL
2990	011754	012701	000006		MOV	#6,R1	;SET NUMBER OF CHAR TO INPUT
2991	011760	012702	177777		MOV	#-1,R2	;SET MAXIMUM LIMIT
2992	011764	012703	002000		MOV	#2000,R3	;SET MINIMUM LIMIT
2993	011770	004737	023410		JSR	PC,TTR	;GO GET VALUE
2994	011774	013700	000646		MOV	TEMP3,RO	;RESTORE RO(REC CNTR)
2995	012000	000137	011276		JMP	YOZ	;RESTART YOZZLE
2996	012004	122777	000207	166600	YOZG: CMPB	#207,#TKB	;CHECK FOR CNTL G
2997	012012	001010			BNE	YOZI	
2998	012014	022737	000176	000606	CMP	#SWREG,SWR	;IS SWREG SELECTED
2999	012022	001004			BNE	YOZI	
3000	012024	005077	166562		CLR	#TKB	;CLEAR CNTL G OUT OF BUFFER
3001	012030	004737	024572		JSR	PC,CNTG	;GO CHANGE SWREG
3002	012034	032777	000040	166544	YOZI: BIT	#40,BSWR	;SEE IF SHOULD CONTINUE YOZZLE
3003	012042	001402			BEQ	YOZH	;IF NOT: BR
3004	012044	000137	011316		JMP	YOZO	
3005	012050	012777	000100	166532	YOZH: MOV	#100,#TKS	;SET TTY INTERRUPT ENABLE
3006	012056	000207			RTS	PC	;EXIT
3007							

3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059

012060 013737 000576 000666
012066 004737 012306
012072 012737 025137 000652
012100 012777 033612 166406
012106 005737 000564
012112 001440
012114 012777 177777 166374
012122 012737 000032 000672
012130 012737 012142 000662
012136 000137 021220
012142 032777 010000 166436
012150 001021
012152 012737 026321 000652
012160 032777 000004 166334
012166 001006
012170 012737 033612 021134
012176 004737 020236
012202 000404
012204 012703 033612
012210 004737 017506
012214 013700 000554
012220 005100
012222 005200
012224 012737 025137 000652
012232 010077 166260
012236 012737 000032 000672
012244 012737 012262 000662
012252 010037 000666
012256 000137 021220
012262 012703 033612
012266 004737 017506
012272 013737 000576 000666
012300 004737 012306
012304 000207

```
*****  
;BACKSPACE SUBROUTINE;  
;  
;THIS SUBROUTINE IS USED TO PERFORM THE  
;BACKSPACE OPERATION REQUIRED BY THE READ  
;ROUTINE FOR READ FORWARD AFTER WRITING.  
;IF A TAPE MARK IS EXPECTED (TM=1) THEN THE SPACE  
;ROUTINE ASSUMES THAT THE TM WILL BE FIRST WHEN  
;BACKSPACING. THEREFORE TWO OPERATIONS ARE REQUIRED  
;TO SPACE OVER A BLOCK. FIRST SPACE OVER THE TM, THEN  
;SPACE OVER THE DATA RECORDS.  
;A CHECK FOR RECORD COUNT ZERO IS MADE AT THE  
;END OF THE SPACE OPERATION TO ASSURE THAT PROPER  
;TAPE POSITIONING WAS DONE.  
*****  
BKSP:  MOV    TSTAL,STAL  
      JSR    PC,STALL      ;DO TURN AROUND STALL  
      MOV    #MSG10,EMADDR  
      MOV    #RDATA,8BA  
      TST    TMEX          ;SEE IF TM  
      BEQ    BO            ;IF NOT: BR  
      MOV    #-1,BFC  
      MOV    #32,MTC1  
      MOV    #BKTM,RTRN  
      JMP    TAPG          ;SPACE TO TM  
      BIT    #10000,8SWR   ;SEE IF SHOULD CHECK ERROR  
      BNE    BO            ;IF NOT: BR  
      MOV    #MSG55,EMADDR  
      BIT    #4,8DS        ;SEE IF TM  
      BNE    BKTM0        ;IF SO: BR  
      MOV    #RDATA,CADER  
      JSR    PC,ERPT      ;PRINT ERROR  
      BR     BO  
      BKTM0: MOV    #RDATA,R3  
      JSR    PC,ER2  
      BO:   MOV    RCNT,RO  
      COM    RO            ;BUILD SPACE AMOUNT  
      INC    RO  
      MOV    #MSG10,EMADDR ;SET ERROR MESH ADDRESS  
      MOV    RO,BFC  
      BKRT: MOV    #32,MTC1 ;SET SPACE REVERSE  
      MOV    #81,RTRN     ;SET RETURN ADDRESS  
      MOV    RO,STAL      ;SET INTERRUPT TIME MULTIPLIER  
      JMP    TAPG        ;GO DO SPACE  
      B1:   MOV    #RDATA,R3  
      JSR    PC,ER2  
      B2:   MOV    TSTAL,STAL ;DO STALL  
      JSR    PC,STALL    ;STALL  
      RTS    PC          ;EXIT
```

3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081

012306 005337 000666
012312 001375
012314 000207

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

```
*****
;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) 0. 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.
*****
```

```

3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095 012316 012701 177760
3096 012322 012702 174000
3097 012326 004737 023356
3098 012332 042737 000001 000626
3099 012340 013737 000626 000556
3100 012346 012737 177777 014410
3101 012354 000207
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113 012356 012702 000001
3114 012362 012701 000500
3115 012366 004737 023356
3116 012372 013737 000626 000554
3117 012400 000207
3118
3119

```

```

*****
;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.
*****
CCNTR: MOV      #20,R1          ;SET HIGH LIMIT
        MOV      #4000,R2      ;SET LOW LIMIT
        JSR      PC,RANG      ;GO GENERATE NUMBER
        BIC      #1,RANSAV
        MOV      RANSAV,FM CNT  ;SET CHAR COUNT
        MOV      #-1,PATS      ;PRESET DATA PATTERN
        RTS      PC           ;EXIT
*****
;RANDOM RECORD COUNT GENERATOR;
;
;THIS ROUTINE ENTERED VIA CONSOLE SWITCH SIX (6)
;IS USED TO GENERATE A RANDOM NUMBER OF RECORDS
;FOR EACH BLOCK OF DATA.
;THE LIMITS ARE ONE (1) TO FIVE HUNDRED (500) OCTAL
;RECORDS PER BLOCK.
*****
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 012402 005737 000634
3167 012406 001001
3168 012410 000207
3169 012412 005037 000674
3170 012416 005037 005160
3171 012422 012700 000010
3172 012426 012701 000750
3173 012432 005021
3174 012434 005300
3175 012436 001375

```

TINP:  TST      TINF      ;SEE IF SHOULD INPUT FROM TTY
       BNE     TINPA     ;IF SO: BR
       RTS     PC        ;EXIT
1INPA: CLR     UNP        ;CLEAR TABLE POINTER
       CLR     REOTC     ;CLEAR EOT UNIT COUNTER
       MOV     #10,R0    ;SET SIZE OF TABLE
       MOV     #UN1,R1   ;SET START OF TABLE
TINPB: CLR     (R1)+     ;CLEAR TABLE
       DEC     R0        ;SEE IF DONE
       BNE     TINPB     ;IF NOT: BR

```

3176	012440	012704	025601		MOV	#MSG31,R4		
3177	012444	005737	000734		TST	ASEQF	;SEE IF AUTO SEQ	
3178	012450	001402			BEQ	TINPB1	;IF NOT: BR	
3179	012452	012704	025527		MOV	#MSG30,R4	;SET AUTO SEQ HOR	
3180	012456	104000		TINPB1:	TTOUTT		;PRINT PROGRAM NAME	
3181	012460	005737	014152		TST	SCVFL	;SEE IF SHORT CONVERSATION	
3182	012464	001067			BNE	TINPC	;IF SO: BR	
3183	012466	012704	026611		MOV	#MSG74,R4		
3184	012472	104000			TTOUTT		;REQUEST STARTING REGISTER ADDRESS	
3185	012474	013703	000544		MOV	REGS,R3		
3186	012500	104002			OCTPP		;PRINT CURRENT REG START	
3187	012502	012705	000544		MOV	#REGS,R5	;SAVE ADDRESS LOCATION	
3188	012506	012701	000006		MOV	#6,R1	;SET SIZE OF ENTRY	
3189	012512	012702	176400		MOV	#176400,R2	;SET UPPER LIMIT	
3190	012516	012703	172300		MOV	#172300,R3	;SET LOWER LIMIT	
3191	012522	004737	023410		JSR	PC,TTR	;GO GET RESPONSE	
3192	012526	012704	026634		MOV	#MSG75,R4		
3193	012532	104000			TTOUTT		;GO REQUEST VECTOR ADDRESS	
3194	012534	013703	000546		MOV	VECT,R3		
3195	012540	104002			OCTPP		;PRINT CURRENT VECTOR	
3196	012542	012705	000546		MOV	#VECT,R5	;SET SAVE LOCATION	
3197	012546	012701	000003		MOV	#3,R1	;SET SIZE OF ENTRY	
3198	012552	012702	000224		MOV	#224,R2	;SET UPPER LIMIT	
3199	012556	012703	000150		MOV	#150,R3	;SET LOWER LIMIT	
3200	012562	004737	023410		JSR	PC,TTR	;GO GET RESPONSE	
3201	012566	013700	000546		MOV	VECT,R0	;GET VECTOR ADDRESS	
3202	012572	012720	021750		MOV	#MTINT,(R0)	;LOAD VECTOR WITH HANDLER ADDRESS	
3203	012576	012710	000340		MOV	#340,(R0)	;LOAD PRIORITY LEVEL	
3204	012602	013700	000544		MOV	REGS,R0	;GET STARTING REGISTER ADDRESS	
3205	012606	012701	000016		MOV	#16,R1	;SET NUMBER OF REGISTERS	
3206	012612	012702	000510		MOV	#C1,R2	;GET FIRST ADDRESS LOCATION	
3207	012616	010022		TINPB0:	MOV	R0,(R2)	;BUILD TABLE OF ADDRESSES	
3208	012620	062700	000002		ADD	#2,R0	;BUMP ADDRESS	
3209	012624	005301			DEC	R1	;SEE IF DONE	
3210	012626	001373			BNE	TINPB0	;IF NOT: BR	
3211	012630	005737	000734		TST	ASEQF	;SEE IF AUTO SEQ	
3212	012634	001403			BEQ	TINPC	;IF NOT: BR	
3213	012636	005726			TST	(SP)	;RESET STACK POINTER	
3214	012640	000137	021766		JMP	ASEQ	;GO TO AUTO SEQUENCE	
3215	012644	012777	000040	165646	TINPC:	MOV	#40,BCS	;INITIALIZE
3216	012652	012704	026255		MOV	#MSG52,R4		
3217	012656	104000			TTOUTT		;REQUEST DRIVE NUMBER	
3218	012660	012705	000550		MOV	#DVN,R5	;GET ADDRESS	
3219	012664	012701	000001		MOV	#1,R1	;SET SIZE OF RESPONSE	
3220	012670	012702	000007		MOV	#7,R2	;SET UPPER LIMIT	
3221	012674	012703	000000		MOV	#0,R3	;SET LOWER LIMIT	
3222	012700	004737	023410		JSR	PC,TTR	;GO GET DRIVE NUMBER	
3223	012704	013777	000550	165606	MOV	DVN,BCS		
3224	012712	005777	165572		TST	BC1	;ACCESS DRIVE	
3225	012716	032777	010000	165574	BIT	#10000,BCS	;SEE IF NED	
3226	012724	001411			BEQ	TINP0	;IF NOT: BR	
3227	012726	012704	026546		MOV	#MSG71,R4		
3228	012732	104000			TTOUTT		;PRINT NED	
3229	012734	013704	000510		MOV	C1,R4		
3230	012740	005204			INC	R4		
3231	012742	152714	000100		BISB	#100,(R4)	;CLEAR TRE	

3232	012746	000736			BR	TINPC	;++G RETRY DVN
3233	012750	012704	025643		TINPO: MOV	#MSG32,R4	
3234	012754	104000			TTOUTT		;PRINT UNIT NUMBER REQUEST
3235	012756	005037	000644		CLR	TEMP2	;CLEAR BUFFER
3236	012762	012705	000644		MOV	#TEMP2,R5	;SET UNIT DESCRIPTION BUFFER ADDRESS
3237	012766	012701	000001		MOV	#1,R1	;SET NUMBER OF CHARACTERS TO INPUT
3238	012772	012702	000007		MOV	#7,R2	;SET MAXIMUM LIMIT
3239	012776	012703	000000		MOV	#0,R3	;SET MINIMUM LIMIT
3240	013002	004737	023410		JSR	PC,TTR	;GO GET UNIT NUMBER
3241	013006	005737	000642		TST	TEMP1	;SEE IF HAVE NEW PARAMETER
3242	013012	001013			BNE	TINPOB	;IF SO: BR
3243	013014	005737	000674		TST	UNP	;SEE IF FIRST ENTRY
3244	013020	001001			BNE	TINPOA	;IF NOT: BR
3245	013022	000752			BR	TINPO	;++G ELSE RETRY
3246	013024	013700	000674		TINPOA: MOV	UNP,R0	
3247	013030	012760	177777	000750	MOV	#-1,UN1(R0)	;SET END UNIT TABLE
3248	013036	000137	013426		JMP	TINP2C	;GO GET RECORD COUNT
3249	013042	013700	000674		TINPOB: MOV	UNP,R0	
3250	013046	042760	000007	000750	BIC	#7,UN1(R0)	;CLEAR UNIT NUMBER
3251	013054	004737	014166		JSR	PC,TPOS1	;GO LOAD UNIT NUMBER TO PROPER POSITION
3252	013060	012777	000040	165432	MOV	#40,BCS	
3253	013066	013777	000550	165424	MOV	DVN,BCS	
3254	013074	016077	000750	165440	MOV	UN1(R0),BC2	;LOAD UNIT NUMBER
3255	013102	032777	002000	165426	TINPOC: BIT	#2000,BDT	;SEE IF SLAVE PRESENT
3256	013110	001005			BNE	TINPOD	;IF SO: BR
3257	013112	012704	026334		MOV	#MSG57,R4	
3258	013116	104000			TTOUTT		;PRINT NON-EXIST SLAVE
3259	013120	000137	012750		JMP	TINPO	;REDO
3260	013124	022777	142011	165404	TINPOD: CMP	#142011,BDT	;++G SEE IF 9TRK TMO2,TU16/TE16
3261	013132	001406			BEQ	TINPOE	;IF SO: BR
3262	013134	012704	026230		MOV	#MSG50,R4	;ILLEGAL DRIVE TYPE
3263	013140	104000			TTOUTT		;GO PRINT
3264	013142	017703	165370		MOV	BDT,R3	
3265	013146	104002			OCTPP		;PRINT DRIVE TYPE REGISTER
3266	013150	012704	025131		TINPOE: MOV	#MSG9,R4	
3267	013154	104000			TTOUTT		;PRINT SERIAL NUMBER TAG
3268	013156	017703	165356		MOV	BSN,R3	
3269	013162	004737	024420		JSR	PC,SNPT	;PRINT SERIAL NUMBER
3270	013166	012704	025664		TINP1: MOV	#MSG33,R4	
3271	013172	104000			TTOUTT		;PRINT DENSITY REQUEST
3272	013174	005037	000644		CLR	TEMP2	;CLEAR BUFFER
3273	013200	012701	000001		MOV	#1,R1	;SET NUMBER OF CHARACTERS TO INPUT
3274	013204	012702	000007		MOV	#7,R2	;SET MAXIMUM LIMIT
3275	013210	012703	000000		MOV	#0,R3	;SET MINIMUM LIMIT
3276	013214	004737	023410		JSR	PC,TTR	;GO GET DENSITY
3277	013220	005737	000642		TST	TEMP1	;SEE IF HAVE NEW PARAMETER
3278	013224	001407			BEQ	TINP2	;IF NOT: BR
3279	013226	042737	003400	000552	BIC	#3400,UDES	;ELSE CLEAR OLD PARAMETER
3280	013234	012703	000010		MOV	#10,R3	;SET POSITION FACTOR
3281	013240	004737	014154		JSR	PC,TPOS	;GO LOAD DENSITY INTO PROPER POSITION
3282	013244	012704	025700		TINP2: MOV	#MSG34,R4	
3283	013250	104000			TTOUTT		;PRINT PARITY REQUEST
3284	013252	005037	000644		CLR	TEMP2	;CLR BUFFER
3285	013256	012701	000001		MOV	#1,R1	;SET NUMBER OF CHARACTERS TO INPUT
3286	013262	012702	000001		MOV	#1,R2	;SET MAXIMUM LIMIT
3287	013266	012703	000000		MOV	#0,R3	;SET MINIMUM LIMIT

3288	013272	004737	023410		JSR	PC, ITR		;GO INPUT PARITY
3289	013276	005737	000642		TST	TEMP1		;SEE IF HAVE NEW PARAMETER
3290	013302	001407			BEQ	TINP2A		;IF NOT: BR
3291	013304	042737	000010	000552	BIC	#10,UDES		;ELSE CLEAR OLD PARAMETER
3292	013312	012703	000003		MOV	#3,R3		;SET POSITION FACTOR
3293	013316	004737	014154		JSR	PC,TPOS		;GO LOAD PARITY TO PROPER POSITION
3294	013322	012704	026277		TINP2A: MOV	#MSG53,R4		
3295	013326	104000			TTOUTT			;REQUEST FORMAT
3296	013330	005037	000644		CLR	TEMP2		
3297	013334	012701	000002		MOV	#2,R1		
3298	013340	012702	000016		MOV	#16,R2		
3299	013344	012703	000014		MOV	#14,R3		
3300	013350	004737	023410		JSR	PC,TTR		;GO GET FORMAT
3301	013354	005737	000642		TST	TEMP1		;SEE IF NEW PARAMETER
3302	013360	001407			BEQ	TINP2B		;IF NOT: BR
3303	013362	042737	000170	000552	BIC	#170,UDES		
3304	013370	012703	000004		MOV	#4,R3		
3305	013374	004737	014154		JSR	PC,TPOS		
3306	013400	005237	005160		TINP2B: INC	REOTC		;BUMP EOT UNIT COUNTER
3307	013404	022737	000016	000674	CMP	#16,UNP		;SEE IF DONE UNITS
3308	013412	001405			BEQ	TINP2C		;IF SO: BR
3309	013414	062737	000002	000674	ADD	#2,UNP		;POINT TO NEXT UNIT
3310	013422	000137	012750		JMP	TINP0		;ELSE LOOK FOR NEXT UNIT
3311	013426	005037	000674		TINP2C: CLR	UNP		;CLEAR UNIT POINTER
3312	013432	013700	005160		MOV	REOTC,R0		
3313	013436	000337	005160		SWAB	REOTC		
3314	013442	110037	005160		MOVB	R0,REOTC		;SET UNIT EOT COUNTER
3315	013446	012704	025713		TINP3: MOV	#MSG35,R4		
3316	013452	104000			TTOUTT			;PRINT RECORD COUNT REQUEST
3317	013454	013703	000554		MOV	RCNT,R3		
3318	013460	104002			OCTPP			;PRINT RECORD COUNT
3319	013462	012705	000554		MOV	#RCNT,R5		;SET RECORD COUNT ADDRESS
3320	013466	012701	000006		MOV	#6,R1		;SET NUMBER OF CHARACTERS TO INPUT
3321	013472	012702	177777		MOV	#-1,R2		;SET MAXIMUM LIMIT
3322	013476	012703	000001		MOV	#1,R3		;SET MINIMUM LIMIT
3323	013502	004737	023410		JSR	PC,TTR		;GO GET RECORD COUNT
3324	013506	013737	000554	000630	MOV	RCNT,RCSAV		;SAVE RECORD COUNT
3325	013514	012704	025734		MOV	#MSG36,R4		
3326	013520	104000			TTOUTT			;PRINT CHARACTER COUNT REQUEST
3327	013522	005437	000556		NEG	FMCNT		
3328	013526	013703	000556		MOV	FMCNT,R3		
3329	013532	104002			OCTPP			;PRINT CHAR COUNT
3330	013534	012705	000556		MOV	#FMCNT,R5		;SET CHARACTER COUNT ADDRESS
3331	013540	012701	000006		MOV	#6,R1		;SET NUMBER OF CHARACTERS TO INPUT
3332	013544	012702	004000		MOV	#4000,R2		;SET MAXIMUM LIMIT
3333	013550	012703	000004		MOV	#4,R3		;SET MINIMUM LIMIT
3334	013554	004737	023410		JSR	PC,TTR		;GO GET CHARACTER COUNT
3335	013560	005437	000556		NEG	FMCNT		;SET TO TWO'S COMPLIMENT
3336	013564	013737	000556	000632	MOV	FMCNT,FCSAV		;SAVE FRAME COUNT
3337	013572	012704	025760		MOV	#MSG37,R4		;PRINT PATTERN NUMBER REQUEST
3338	013576	104000			TTOUTT			
3339	013600	013703	000560		MOV	PATRN,R3		
3340	013604	104002			OCTPP			;PRINT PATTERN
3341	013606	005037	014556		CLR	DOFL		;CLEAR EXTERNAL DATA FLAG
3342	013612	012705	000560		MOV	#PATRN,R5		;SET PATTERN NUMBER ADDRESS
3343	013616	012701	000002		MOV	#2,R1		;SET NUMBER OF CHARACTERS TO INPUT

3344	013622	012702	000015	MOV	#15,R2	;SET MAXIMUM LIMIT
3345	013626	012703	000000	MOV	#0,R3	;SET MINIMUM LIMIT
3346	013632	004737	023410	JSR	PC,TTR	;GO GET PATTERN NUMBER
3347	013636	012704	026474	MOV	#MSG69,R4	
3348	013642	104000		TTOUTT		;REQUEST TM
3349	013644	013703	000564	MOV	TMEX,R3	
3350	013650	104002		OCTPP		;PRINT CURRENT TM FLAG SETTING
3351	013652	012705	000564	MOV	#TMEX,R5	;GET TM FLAG ADDRESS
3352	013656	012701	000001	MOV	#1,R1	;SET SIZE OF RESPONSE
3353	013662	012702	000001	MOV	#1,R2	;SET UPPER LIMIT
3354	013666	012703	000000	MOV	#0,R3	;SET LOWER LIMIT
3355	013672	004737	023410	JSR	PC,TTR	;TM 1=YES
3356	013676	012704	025261	MOV	#MSG21,R4	
3357	013702	104000		TTOUTT		;REQUEST INTERCHANGE READ
3358	013704	013703	000566	MOV	INTRF,R3	
3359	013710	104002		OCTPP		;PRINT CURRENT SETTING
3360	013712	012705	000566	MOV	#INTRF,R5	;GET FLAG ADDRESS
3361	013716	012701	000001	MOV	#1,R1	;SET SIZE OF RESPONSE
3362	013722	012702	000001	MOV	#1,R2	;SET UPPER LIMIT
3363	013726	012703	000000	MOV	#0,R3	;SET LOWER LIMIT
3364	013732	004737	023410	JSR	PC,TTR	;GO GET RESPONSE
3365	013736	012704	026003	MOV	#MSG38,R4	
3366	013742	104000		TTOUTT		;REQUEST SINGLE PASS
3367	013744	013703	000570	MOV	SPFLG,R3	
3368	013750	104002		OCTPP		;PRINT CURRENT SETTING
3369	013752	012705	000570	MOV	#SPFLG,R5	;SET ADDRESS OF FLAG
3370	013756	012701	000001	MOV	#1,R1	;SET SIZE OF RESPONSE
3371	013762	012702	000001	MOV	#1,R2	;SET UPPER LIMIT
3372	013766	012703	000000	MOV	#0,R3	;SET LOWER LIMIT
3373	013772	004737	023410	JSR	PC,TTR	;GO GET RESPONSE
3374	013776	005737	014152	TINP4: TST	SCVFL	;SEE IF SHORT CONVERSATION
3375	014002	001060		BNE	TINPX	;IF SO; BR
3376	014004	012704	026023	MOV	#MSG40,R4	
3377	014010	104000		TTOUTT		;PRINT READ STALL REQUEST
3378	014012	013703	000572	MOV	RSTAL,R3	
3379	014016	104002		OCTPP		;PRINT READ STALL
3380	014020	012705	000572	MOV	#RSTAL,R5	;SET READ STALL ADDRESS
3381	014024	012701	000006	MOV	#6,R1	;SET NUMBER OF CHARACTERS TO INPUT
3382	014030	012702	177777	MOV	#-1,R2	;SET MAXIMUM LIMIT
3383	014034	012703	000001	MOV	#1,R3	;SET MINIMUM LIMIT
3384	014040	004737	023410	JSR	PC,TTR	;GO GET READ STALL
3385	014044	012704	026051	MOV	#MSG41,R4	
3386	014050	104000		TTOUTT		;PRINT WRITE STALL REQUEST
3387	014052	013703	000574	MOV	WSTAL,R3	
3388	014056	104002		OCTPP		;PRINT READ STALL
3389	014060	012705	000574	MOV	#WSTAL,R5	;SET WRITE STALL ADDRESS
3390	014064	012701	000006	MOV	#6,R1	;SET NUMBER OF CHARACTERS TO INPUT
3391	014070	012702	177777	MOV	#-1,R2	;SET MAXIMUM LIMIT
3392	014074	012703	000001	MOV	#1,R3	;SET MINIMUM LIMIT
3393	014100	004737	023410	JSR	PC,TTR	;GO GET WRITE STALL
3394	014104	012704	026063	MOV	#MSG42,R4	
3395	014110	104000		TTOUTT		;PRINT TURN AROUND STALL REQUEST
3396	014112	013703	000576	MOV	TSTAL,R3	
3397	014116	104002		OCTPP		;PRINT TA STALL
3398	014120	012705	000576	MOV	#TSTAL,R5	;SET TURN AROUND STALL ADDRESS
3399	014124	012701	000006	MOV	#6,R1	;SET NUMBER OF CHARACTERS TO INPUT

3400	014130	012702	177777		MOV	2-1,R2		;SET MAXIMUM LIMIT
3401	014134	012703	000001		MOV	01,R3		;SET MINIMUM LIMIT
3402	014140	004737	023410		JSR	PC,TTR		;GO GET TURN AROUND STALL
3403	014144	005037	014152	TINPX:	CLR	SCVFL		;CLEAR SHORT CONVERSATION FLAG
3404	014150	000207			RTS	PC		;EXIT
3405	014152	000000		SCVFL:	0			;SHORT CONVERSATION FLAG
3406								
3407								;UNIT DESCRIPTION POSITIONING SUBROUTINE*****
3408								
3409	014154	000241		TPOS:	CLC			
3410	014156	006137	000644		ROL	TEMP2		;POSITION CHARACTER
3411	014162	005303			DEC	R3		;SEE IF DONE
3412	014164	001373			BNE	TPOS		;IF NOT: BR
3413	014166	013700	000674	TPOS1:	MOV	UNP,R0		;LOAD UNIT POINTER
3414	014172	053760	000644		BIS	TEMP2,UN1(R0)		;LOAD CHARACTER INTO UN1(R0)
3415	014200	000207			RTS	PC		;EXIT
3416								

3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472

014202 005737 015152
014206 001044
014210 005737 000734
014214 001406
014216 005737 000560
014222 100003
014224 004737 015104
014230 000207
014232 023737 000560 014410
014240 001014
014242 013703 000552
014246 042703 177767
014252 023703 014412
014256 001404
014260 010337 014412
014264 004737 015154
014270 000207
014272 012703 027604
014276 013701 000560
014302 010137 014410
014306 062701 000001
014312 006301
014314 004771 002772
014320 032777 010000 164210
014326 001410
014330 012702 002002
014334 012701 027604
014340 042721 140300
014344 005302
014346 001374
014350 004737 015154
014354 012702 002000
014360 012701 033612
014364 005021
014366 005302
014370 001375
014372 013737 000552 014412
014400 042737 177767 014412

DSUP:
DSO:
DSOC:
DSOB:
DSOA:
DS1:
DS2:
DS2A:
DS3:
DS4:

```
*****  
;DATA SETUP ROUTINE:  
;  
;THIS ROUTINE IS USED TO GENERATE INTO THE ENTIRE  
;WRITE BUFFER (4000 OCTAL CHARACTERS) THE DATA PATTERN  
;SELECTED BY THE OPERATOR. THERE ARE 15 (8) FIXED  
;DATA PATTERNS AVAILABLE AND ONE SELECTION (DATA PATTERN 0)  
;WHICH WILL READ ANY PATTERN PRESENTED AT THE  
;HIGH SPEED PAPER TAPE READER. THIS TAPE MUST BE PREPARED  
;BY USING THE PROGRAM CALLED DTC. (MAINDEC-11-DZTUF-A-D)  
;RANDOM DATA MAY ALSO BE USED VIA CONSOLE  
;SWITCH EIGHT (8).  
;THIS ROUTINE IS ALSO USED TO CLEAR OUT THE  
;READ BUFFER (4000 OCTAL CHARACTERS) BEFORE EACH  
;RECORD IS READ.  
*****  
TST RDFL ;SEE IF DID RANDOM DATA  
BNE DS1 ;G F SO BRANCH  
TST ASEQF ;SEE IF AUTO SEQ  
BEQ DSOC ;IF NOT; BR  
TST PATRN ;SEE IF AUTO RANDOM  
BPL DSOC ;IF NOT; BR  
JSR PC,DATR ;ELSE GO GENERATE RANDOM DATA  
RTS PC ;RETURN  
CMP PATRN,PAT5 ;SEE IF NEW PATTERN  
BNE DSOA ;IF SO; BR  
MOV UDES,R3 ;GET UNIT DESCRIPTION  
BIC #177767,R3 ;MASK EVEN PARITY  
CMP PARS,R3 ;SEE IF SAME AS LAST TIME  
BEQ DSOB ;IF SO; BR  
MOV R3,PAR5 ;SAVE PARITY  
JSR PC,CRC/LRC ;GO GENERATE EXPT CRC/LRC  
RTS PC  
DSOB: RTS PC  
DSOA: MOV #WDATA,R3 ;R3 = ADDRS OF WRITE BUFFER  
MOV PATRN,R1 ;R1 = PATTERN SELECTOR  
MOV R1,PAT5  
ADD #1,R1 ;BUMP POINTER  
ASL R1 ;G MAKE PATTERN SELECTOR EVEN  
JSR PC,@DATBL(R1) ;GO GENERATE PATTERN  
DS1: BIT #10000,BDT ;SEE IF 7 CH  
BEQ DS2A ;IF NOT; BR  
MOV #2002,R2 ;SET BUFFER SIZE  
MOV #WDATA,R1 ;SET START OF BUFFER  
DS2: BIC #140300,(R1) ;MASK FOR 7 CH  
DEC R2 ;SEE IF DONE  
BNE DS2 ;IF NOT; BR  
DS2A: JSR PC,CRC/LRC ;GO GENERATE EXPT CRC/LRC  
DS3: MOV #2000,R2 ;R2=BUFFER SIZE  
MOV #RDATA,R1 ;R1=READ DATA START  
DS4: CLR (R1) ;CLEAR BUFFER  
DEC R2 ;SEE IF DONE ALL  
BNE DS4 ;IF NOT; BR  
MOV UDES,PAR5 ;GET UNIT DESCRIPTION  
BIC #177767,PAR5 ;MASK PARITY
```

3473	014406	000207			RTS	PC		;EXIT
3474	014410	177777			PATS:	-1		;PATTERN NUMBER SAVE
3475	014412	000000			PARS:	0		
3476								
3477								;EXTERNAL DATA INPUT FROM H/S READER (256 CHARACTER MAXIMUM)
3478								
3479	014414	005737	014556		DATO:	TST	DOFL	;++G BRANCH IF EXTERNAL INPUT
3480	014420	001401				BEQ	1	;++G
3481	014422	000207				RTS	PC	;++G RETURN
3482	014424	012737	000001	014556	1:	MOV	#1,DOFL	;SET EXTERNAL FLAG
3483	014432	005077	164164			CLR	SPRB	;CLEAR READER BUFFER
3484	014436	005077	164156			CLR	SPRS	;CLEAR READER STATUS
3485	014442	005037	000642			CLR	TEMP1	;CLEAR FOR USE AS CHARACTER FLAG
3486	014446	052777	000001	164144	DATO:	BIS	#1,SPRS	;START READER
3487	014454	105777	164140		DATOB:	TSTB	SPRS	;++G SEE IF DONE
3488	014460	100375				BPL	DATOB	;++B
3489	014462	005001				CLR	R1	;CLEAR SAVE LOCATION
3490	014464	117701	164132			MOVB	SPRB,R1	;SAVE CHARACTER
3491	014470	005737	000642			TST	TEMP1	;SEE IF HAVE FOUND START CHARACTER
3492	014474	001011				BNE	DATOC	;IF SO : BR
3493	014476	105701				TSTB	R1	;SEE IF CHARACTER IS 0
3494	014500	001762				BEQ	DATO	;IF SO : BR
3495	014502	012737	000001	000642		MOV	#1,TEMP1	;ELSE SET CHARACTER FOUND FLAG
3496	014510	010137	000644			MOV	R1,TEMP2	;SAVE DATA SIZE
3497	014514	010102				MOV	R1,R2	;SAVE DATA SIZE
3498	014516	000753				BR	DATO	;++G GO GET FIRST DATA CHAR
3499	014520	110123			DATOC:	MOVB	R1,(R3)+	;LOAD BUFFER
3500	014522	005302				DEC	R2	;SEE IF READ ALL
3501	014524	001350				BNE	DATO	;IF NOT : BR
3502	014526	012701	027604		DATOD:	MOV	#DATA,R1	;R1 = START OF WRITE BUFFER
3503	014532	013702	000644			MOV	TEMP2,R2	;R2 = SIZE OF DATA FIELD
3504	014536	112123			DATOE:	MOVB	(R1)+,(R3)+	;REPEAT LOAD OF DATA FIELD
3505	014540	022703	033612			CMP	#RDATA,R3	;SEE IF DONE
3506	014544	003001				BGT	DATOF	;IF NOT: BR
3507	014546	000207				RTS	PC	;++G RETURN
3508	014550	005302			DATOF:	DEC	R2	;SEE IF AT END OF DATA FIELD
3509	014552	001371				BNE	DATOE	;IF NOT : BR
3510	014554	000764				BR	DATOD	;++G ELSE RESTART FILL
3511	014556	000000			DOFL:	0		;EXTERNAL DATA FLAG=1 IF ALREADY DONE
3512								

```

3513                                     ;ALL ONES*****
3514
3515 014560 012701 177777          DAT1:  MOV    # -1,R1          ;R1=DATA
3516 014564 012702 002002          DAT1A: MOV    #2002,R2        ;R2=WORD COUNT +2
3517 014570 010123                   DAT1B: MOV    R1,(R3)+       ;LOAD BUFFER
3518 014572 005302                   DEC     R2                  ;SEE IF DONE
3519 014574 001375                   BNE    DAT1B               ;IF NOT: BR
3520 014576 000207                   RTS     PC                  ;++G RETURN
3521
3522                                     ;ALL ZEROS*****
3523
3524 014600 005001          DAT2:  CLR     R1            ;R1=DATA
3525 014602 000770          BR     DAT1A            ;++G LOAD BUFFER
3526
3527                                     ;WALKING ONE*****
3528
3529 014604 012701 000001          DAT3:  MOV    #1,R1        ;R1=DATA
3530 014610 000241                   CLC
3531 014612 012702 004004          DAT3A: MOV    #4004,R2     ;R2=CHARACTER COUNT+4
3532 014616 110123                   DAT3B: MOV    R1,(R3)+     ;LOAD BUFFER
3533 014620 106101                   ROLB   R1                 ;SET NEXT CHARACTER
3534 014622 005302                   DEC     R2                 ;SEE IF DONE
3535 014624 001374                   BNE    DAT3B              ;IF NOT: BR
3536 014626 000207                   RTS     PC                  ;++G RETURN
3537
3538                                     ;WALKING ZERO*****
3539
3540 014630 012701 000376          DAT4:  MOV    #376,R1      ;R1=START OF DATA
3541 014634 000261                   SEC
3542 014636 000765                   BR     DAT3A              ;++G LOAD BUFFER
3543
3544                                     ;ALTERNATING ONE/ZERO*****
3545
3546
3547 014640 012701 052525          DAT5:  MOV    #52525,R1   ;R1=DATA
3548 014644 000747                   BR     DAT1A              ;++G LOAD BUFFER
3549
3550                                     ;ALTERNATING ZERO/ONE*****
3551
3552 014646 012701 125252          DAT6:  MOV    #125252,R1  ;R1=DATA
3553 014652 000744                   BR     DAT1A              ;++G LOAD BUFFER
3554
3555                                     ;ONE/ZERO IN ALTERNATING WORDS*****
3556
3557 014654 012701 125252          DAT7:  MOV    #125252,R1  ;SET WORD 1
3558 014660 012702 052525          MOV    #52525,R2        ;SET WORD 2
3559 014664 012704 001002          MOV    #1002,R4         ;SET NUMBER OF ENTRIES
3560 014670 010123          DAT7A: MOV    R1,(R3)+     ;LOAD WORD 1
3561 014672 010223                   MOV    R2,(R3)+         ;LOAD WORD 2
3562 014674 005304                   DEC     R4                ;SEE IF DONE
3563 014676 001374                   BNE    DAT7A             ;IF NOT: BR
3564 014700 000207                   RTS     PC                  ;++G RETURN
3565

```

```

3566                                     ;WALKING ONE/ALL ONE IN ALTERNATING CHARS****
3567
3568 014702 012702 002002      DAT10:  MOV    #2002,R2      ;SET BUFFER SIZE
3569 014706 012701 000001      MOV    #1,R1              ;SET WALK BASE
3570 014712 000241
3571 014714 012713 177400      DAT10A: MOV    #177400,(R3) ;LOAD ALL ONE BYTE
3572 014720 050123              BIS    R1,(R3)           ;LOAD WALK BYTE
3573 014722 106101              ROLB  R1                 ;WALK ONE
3574 014724 005302              DEC   R2
3575 014726 001372              BNE   DAT10A             ;DO FULL BUFFER
3576 014730 000207              RTS    PC                 ;..G RETURN
3577
3578                                     ;ALL BITS 0-377*****
3579
3580 014732 005001      DAT11:  CLR    R1              ;R1=STARTING DATA
3581 014734 012702 004004      MOV    #4004,R2          ;R2=CHARACTER COUNT.4
3582 014740 110123      DAT11A: MOVB   R1,(R3)       ;LOAD BUFFER
3583 014742 105201      INCB  R1                 ;BUMP DATA
3584 014744 005302      DEC   R2                 ;SEE IF DONE
3585 014746 001374      BNE   DAT11A             ;IF NOT: BR
3586 014750 000207      RTS    PC                 ;..G RETURN
3587
3588                                     ;ALL BITS 377-0*****
3589
3590 014752 012701 000377      DAT12:  MOV    #377,R1     ;R1=STARTING DATA
3591 014756 012702 004004      MOV    #4004,R2          ;R2=CHARACTER COUNT.4
3592 014762 110123      DAT12A: MOVB   R1,(R3)       ;LOAD BUFFER
3593 014764 105301      DECB  R1                 ;BUMP DATA
3594 014766 005302      DEC   R2                 ;SEE IF DONE
3595 014770 001374      BNE   DAT12A             ;IF NOT: BR
3596 014772 000207      RTS    PC                 ;..G RETURN
3597
3598                                     ;ALTERNATING CHARACTERS 0 AND 377*****
3599
3600 014774 012701 000377      DAT13:  MOV    #377,R1     ;R1 = DATA
3601 015000 000137 014564      JMP    DAT1A             ;LOAD BUFFER
3602
3603                                     ;WALKING ZERO/ALL ZERO IN ALTERNATING CHARS*****
3604
3605 015004 012702 002002      DAT14:  MOV    #2002,R2     ;SET BUFFER SIZE
3606 015010 012701 000376      MOV    #376,R1          ;SET WALK BASE
3607 015014 000261      SEC
3608 015016 010113      DAT14A: MOV    R1,(R3)       ;LOAD WALK BYTE
3609 015020 042723 177400      BIC   #177400,(R3)       ;CLEAR HIGH BYTE
3610 015024 106101      ROLB  R1                 ;WALK ZERO BIT
3611 015026 005302      DEC   R2
3612 015030 001372      BNE   DAT14A             ;FILL BUFFER
3613 015032 000207      RTS    PC                 ;..G RETURN
3614

```

```

3615                                     ;AUTO SEQUENCE PATTERN*****
3616
3617 015034 012702 000200      DAT15:  MOV    @200,R2      ;SET NUMBER OF ENTRIES
3618 015040 012701 015064      DAT15A: MOV    @APATS,R1    ;SET START OF PATTERN
3619 015044 012704 000010      MOV    @10,R4             ;SET SIZE OF PATTERN
3620 015050 012123              DAT15B: MOV    (R1),R3     ;FILL BUFFER
3621 015052 005304              DEC    R4                 ;SEE IF DONE PATTERN
3622 015054 001375              BNE    DAT15B            ;IF NOT: BR
3623 015056 005302              DEC    R2                 ;SEE IF DONE BUFER
3624 015060 001367              BNE    DAT15A            ;IF NOT: BR
3625 015062 000207              RTS    PC                 ;*G RETURN
3626 015064 000000
3627 015066 177400
3628 015070 000377
3629 015072 000000
3630 015074 177777
3631 015076 000377
3632 015100 177400
3633 015102 177777
3634
3635                                     ;RANDOM DATA GENERATOR SUBROUTINE*****
3636
3637 015104 013704 000556      DATR:  MOV    FMCNT,R4     ;SET NUMBER OF FRAMES
3638 015110 012703 027604      MOV    @NDATA,R3        ;SET ADDRESS OF START OF BUFFER
3639 015114 012701 177777      MOV    @-1,R1           ;SET HIGH LIMIT
3640 015120 005002              CLR    R2                ;SET LOW LIMIT
3641 015122 004737 023356      DATRO: JSR    PC,RANG     ;GO GENERATE NUMBER
3642 015126 013723 000626      MOV    RANSAR,(R3)     ;LOAD BUFFER
3643 015132 005204              INC    R4                 ;SEE IF DONE WHOLE BUFFER
3644 015134 001372              BNE    DATRO            ;IF NOT: BR
3645 015136 004737 014320      JSR    PC,DS1          ;GO CHECK FOR 7 CH
3646 015142 012737 000001 015152  MOV    @1,RDFL         ;SET RANDOM DATA FLAG
3647 015150 000207              RTS    PC                 ;EXIT
3648 015152 000000      RDFL:  0                 ;RANDOM DATA SELECT FLAG

```



```

3649
3650 ;*****
3651 ;CRC/LRC CHARACTER BUILD;
3652 ;
3653 ;THIS ROUTINE WILL CONSTRUCT AND SAVE THE EXPECTED
3654 ;CRC AND LRC CHARACTERS ACCORDING TO DATA AND
3655 ;RECORD SIZE IF OPERATING IN NRZ MODE
3656 ;*****
3657
3658 015154 013700 000556 CRCLRC: MOV FMCNT,RO ;SET RECORD SIZE
3659 015160 005400 NEG RO
3660 015162 012701 027604 MOV #WDATA,R1 ;SET START OF BUFFER
3661 015166 005037 015536 CLR XORS
3662 015172 111104 CLO: MOVB (R1),R4 ;GET CHARACTER
3663 015174 004737 015364 JSR PC,CLP ;GO GET PARITY OF CHARACTER
3664 015200 004737 015512 JSR PC,XOR ;XOR CHARACTER
3665 015204 000241 CLC
3666 015206 006004 ROR R4 ;ROTATE 1 RIGHT
3667 015210 103014 BCC CL2 ;IF NO CARRY: BR
3668 015212 052704 000400 BIS #400,R4 ;SET BIT NINE
3669 015216 000241 CLC
3670 015220 010405 CL1: MOV R4,R5 ;SAVE CHARACTER
3671 015222 042705 177703 BIC #177703,R5
3672 015226 005105 COM R5
3673 015230 042705 177703 BIC #177703,R5
3674 015234 042704 000074 BIC #74,R4
3675 015240 050504 BIS R5,R4 ;COMPLIMENT BITS 2,3,4,5
3676 015242 010437 015536 CL2: MOV R4,XORS
3677 015246 005300 DEC RO
3678 015250 001401 BEQ CLLAST ;IF LAST CHARACTER: BR
3679 015252 000747 BR CLO ;G GET NEXT
3680 015254 013704 015536 CLLAST: MOV XORS,R4
3681 015260 005137 015536 COM XORS
3682 015264 042737 177050 015536 BIC #177050,XORS
3683 015272 042704 177727 BIC #177727,R4 ;COMPLIMENT ALL BUT BITS 3&5
3684 015276 050437 015536 BIS R4,XORS
3685 015302 013737 015536 015540 MOV XORS,EXCRC ;SAVE EXPECTED CRC
3686 015310 013700 000556 MOV FMCNT,RO
3687 015314 005400 NEG RO
3688 015316 012701 027604 MOV #WDATA,R1 ;DO EXPT LRC
3689 015322 005037 015536 CLR XORS
3690 015326 111104 CL3: MOVB (R1),R4
3691 015330 004737 015364 JSR PC,CLP ;GET PARITY
3692 015334 004737 015512 JSR PC,XOR ;XOR CHARACTER
3693 015340 005300 DEC RO
3694 015342 001371 BNE CL3 ;DO ALL FOR LRC
3695 015344 013704 015540 MOV EXCRC,R4
3696 015350 004737 015512 JSR PC,XOR ;XOR CRC TO DATA
3697 015354 013737 015536 015542 MOV XORS,EXLRC ;SAVE EXPT LRC
3698 015362 000207 RTS PC ;RETURN
3699 015364 005704 CLP: TST R4 ;SEE IF 0 CHAR
3700 015366 001010 BNE CLPE ;IF NOT: BR
3701 015370 032737 000010 000552 BIT #10,UDES ;SEE IF EVEN PARITY
3702 015376 001404 BEQ CLPE ;IF NOT: BR
3703 015400 012704 000420 MOV #420,R4 ;SET 0 CHAR EVEN PARITY
3704 015404 005201 INC R1 ;BUMP POINTER

```



```

3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754 015546 005037 000656          DCHK: CLR      B8C          ;CLEAR BAD RECORD CNTR
3755 015552 005037 000704          CLR      DERFL        ;CLEAR DATA ERROR FLAG
3756 015556 013705 000556          MOV      FMCNT,R5     ;LOAD CHAR COUNT
3757 015562 032737 000020 000552  BIT      #20,UDES     ;SEE IF CORE DUMP
3758 015570 001402                    BEQ      DCHK0        ;IF NOT: BR
3759 015572 000261                    SEC
3760 015574 006005                    ROR
3761 015576 012701 027604          DCHK0: MOV      #WDATA,R1   ;R5 = FC/2
3762 015602 012702 033612          MOV      #RDATA,R2   ;SET WRITE DATA ADDR
3763 015606 032737 000010 000552  BIT      #10,UDES     ;SET READ DATA ADDR
3764 015614 001430                    BEQ      DFOC0        ;SEE IF EVEN PARITY
3765 015616 032737 000020 000552  BIT      #20,UDES     ;SEE IF CORE DUMP PARITY
3766 015624 001024                    BNE
3767 015626 032737 002000 000552  BIT      #2000,UDES   ;IF SO: BR
3768 015634 001020                    BNE      DFOC0        ;SEE IF PE MODE
3769 015636 105711                    DFOF: TSTB     (R1)    ;IF SO: BR
3770 015640 001404                    BEQ      DFO0        ;SEE IF 0 CHAR
3771 015642 005201                    INC      R1          ;IF SO: BR
3772 015644 005205                    DFOE: INC      R5          ;BUMP POINTER
3773 015646 001373                    BNE      DFOF        ;SEE IF DONE
3774 015650 000406                    BR       DFOC        ;IF NOT: BR
3775 015652 112721 000020          DFO0: MOV      #20,(R1)+ ;ELSE CONTINUE
3776 015656 012737 177777 014410  MOV      #-1,PATS    ;SET 20 IN PLACE OF 0
3777 015664 000767                    BR       DFOE        ;SET PATTERN GENERATE FLAG
3778 015666 013705 000556          DFOC: MOV      FMCNT,R5 ;RESET CHAR CNT
3779 015672 012701 027604          MOV      #WDATA,R1   ;RESET DATA ADDRESS
3780 015676 032737 010000 000562  DFOC0: BIT      #10000,RDCMD ;SEE IF READ REVERSE
3781 015704 001462                    BEQ
3782 015706 013704 000556          DFOB: MOV      FMCNT,R4 ;IF NOT: BR
3783 015712 005404                    NEG      R4          ;GET FRAME COUNT
3784 015714 032737 000020 000552  BIT      #20,UDES     ;SET TO WHOLE NUMBER
3785 015722 001402                    BEQ      DFOB0       ;SEE IF CORE DUMP
3786 015724 000241                    CLC
3787 015726 006004                    ROR      R4          ;IF NOT: BR
3788 015730 060401                    DFOB0: ADD      R4,R1   ;SET TO FC/2
3789 015732 060402                    ADD      R4,R2       ;POINT TO START OF WRITE DATA
3790 015734 032737 000001 000556  BIT      #1,FMCNT    ;POINT TO START OF READ DATA
3791 015742 001401                    BEQ      DFOA        ;SEE IF ODD FRAME COUNT
3792 015744 105722                    TSTB     (R2)+       ;IF NOT: BR
3793 015746 032737 000020 000552  DFOA: BIT      #20,UDES ;BUMP POINTER
3794 015754 001431                    BEQ      DFOA4       ;SEE IF CORE DUMP
                                     ;IF NOT: BR

```

```

3795 015756 000241          CLC
3796 015760 132742 000001  BITB    #1,.(R2)      ;SEE IF BIT 0 = 1
3797 015764 001401          BEQ     DFOA0         ;IF NOT: BR
3798 015766 000261          SEC
3799 015770 106012          DFOA0: RORB    (R2)
3800 015772 000241          CLC
3801 015774 132712 000001  BITB    #1,.(R2)
3802 016000 001401          BEQ     DFOA1
3803 016002 000261          SEC
3804 016004 106012          DFOA1: RORB    (R2)      ;POSITION BITS FOR REVERSE CORE DUMP
3805 016006 000241          CLC
3806 016010 132712 000001  BITB    #1,.(R2)
3807 016014 001401          BEQ     DFOA2
3808 016016 000261          SEC
3809 016020 106012          DFOA2: RORB    (R2)
3810 016022 000241          CLC
3811 016024 132712 000001  BITB    #1,.(R2)
3812 016030 001401          BEQ     DFOA3
3813 016032 000261          SEC
3814 016034 106012          DFOA3: RORB    (R2)
3815 016036 005202          INC     R2           ;RESET POINTER
3816 016040 124142          DFOA4: CMPB    -(R1),-(R2) ;TEST DATA CHARACTER
3817 016042 001010          BNE    DF1           ;IF NOT GOOD: BR
3818 016044 105037 000656  CLRB    BBC         ;CLEAR BAD RECORD COUNTER
3819 016050 000411          BR     DF2
3820 016052 122122          DFO:   CMPB    (R1),.(R2). ;CHECK DATA
3821 016054 001003          BNE    DF1           ;IF BAD: BR
3822 016056 105037 000656  CLRB    BBC         ;CLEAR BAD RECORD CNTR
3823 016062 000404          BR     DF2
3824 016064 004737 016670  JSR     PC,DRPKF    ;GO GET DROPS AND PICKS
3825 016070 004737 016162  JSR     PC,DERR     ;GO DO PRINT
3826 016074 005205          DFO2: INC     R5           ;BUMP CHAR CNTR
3827 016076 001405          BEQ     DF3           ;IF DONE ALL: BR
3828 016100 032737 010000 000562 BIT     #10000,RDCMD ;SEE IF READ REVERSE
3829 016106 001761          BEQ     DFO         ;IF NOT: BR
3830 016110 000716          BR     DFOA         ;ELSE CONTINUE READ REV
3831 016112 005037 000664  DFO3: CLR     HDRFL    ;CLEAR HEADER FLAG
3832 016116 005737 000704  TST     DERFL       ;SEE IF HAD DATA ERROR
3833 016122 001416          BEQ     DFX         ;IF NOT: BR
3834 016124 005737 000706  TST     SERFL
3835 016130 001013          BNE    DFX         ;IF NOT DATA ERROR ONLY: BR
3836 016132 013704 000674  MOV     UNP,R4
3837 016136 032737 010000 000562 BIT     #10000,RDCMD ;SEE IF READ REVERSE
3838 016144 001003          BNE    DF4           ;IF SO: BR
3839 016146 005264 001132  INC     DATER1(R4)  ;BUMP DATA ERROR FORWARD COUNTER
3840 016152 000402          BR     DFX
3841 016154 005264 001172  DFO4: INC     DEREV1(R4) ;BUMP REVERSE DATA ERROR
3842 016160 000207          DFX:  RTS     PC     ;EXIT
3843

```

```

3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872 016162 032777 002000 162416 DERR: BIT #2000,BSMR ;SEE IF SHOULD PRINT ERRORS
3873 016170 001067 BNE DERR4 ;++G BRANCH IF NOT
3874 016172 005237 000670 DERRO: INC PFLG ;SET PRINT FLAG
3875 016176 005737 000664 TST HDRFL ;SEE IF HAVE PRINTED HEADER
3876 016202 001007 BNE DERROA ;IF SO: BR
3877 016204 004737 022772 JSR PC,PAPRT ;PRINT CYCLE NUMBER
3878 016210 012704 025056 MOV #MSG1,R4 ;LOAD ERROR MSG ADDR
3879 016214 104000 TTOUTT ;PRINT ERROR
3880 016216 004737 021160 JSR PC,FRPRT ;PRINT F OR R
3881 016222 012704 025075 DERROA: MOV #MSG4,R4
3882 016226 104000 TTOUTT ;PRINT CHAR NO. HEADER
3883 016230 010203 MOV R2,R3
3884 016232 162703 033612 SUB #RDATA,R3 ;POINT TO CHAR
3885 016236 005303 DEC R3
3886 016240 032737 010000 000562 BIT #10000,RDCMD ;SEE IF READ REVERSE
3887 016246 001402 BEQ DERROB ;IF NOT: BR
3888 016250 010503 MOV R5,R3 ;GET CHAR NUMBER
3889 016252 005103 COM R3
3890 016254 104002 DERROB: OCTPP ;PRINT CHAR NUMBER
3891 016256 012704 025063 MOV #MSG2,R4
3892 016262 104000 TTOUTT ;PRINT EXPECTED DATA
3893 016264 032737 010000 000562 BIT #10000,RDCMD ;SEE IF READ REVERSE
3894 016272 001402 BEQ DERROC ;IF NOT: BR
3895 016274 111103 MOVB (R1),R3 ;GET CHAR
3896 016276 000401 BR DERROD
3897 016300 114103 DERROC: MOVB -(R1),R3 ;LOAD EXPECTED DATA
3898 016302 004737 024306 DERROD: JSR PC,DOUT ;GO PRINT CHAR
3899 016306 012704 025070 MOV #MSG3,R4

```

Line	Address	Offset	Control	OpCode	Register	Comment	
3900	016312	104000		TTOUTT		;PRINT RECIEVED DATA	
3901	016314	032737	010000	BIT	#10000,RDCMD	;SEE IF READ REVERSE	
3902	016322	001402		BEQ	DERR1	;IF NOT: BR	
3903	016324	111203		MOVB	(R2),R3	;GET CHAR	
3904	016326	000401		BR	DERR2		
3905	016330	114203		DERR1: MOVB	-(R2),R3		
3906	016332	004737	024306	DERR2: JSR	PC,DOUT	;PRINT BAD CHAR	
3907	016336	032737	010000	BIT	#10000,RDCMD	;SEE IF READ REVERSE	
3908	016344	001001		BNE	DERR4	;...G BRANCH IF NOT	
3909	016346	122122		DERR3: CMPB	(R1),.(R2),	;RESET POINTERS	
3910	016350	105237	000656	DERR4: INCB	BBC	;BUMP BAD RECORD CNTR	
3911	016354	122737	000010	CMPB	#10,BBC	;SEE IF BLD BTH	
3912	016362	001120		BNE	DEREX	;IF NOT: BR	
3913	016364	032777	002000	BIT	#2000,BSWR	;SEE IF PRINT INHIBIT	
3914	016372	001003		BNE	1#	;IF SO: BR	
3915	016374	012704	025207	MOV	#MSG15,R4		
3916	016400	104000		TTOUTT		;PRINT BLD BTH	
3917	016402	105037	000656	1#: CLRB	BBC	;RESET BAD RECORD CNTR	
3918	016406	000337	000656	SWAB	BBC	;POSITION BLD BTH AMOUNT	
3919	016412	105237	000656	INCB	BBC	;BUMP AMOUNT	
3920	016416	122737	000003	000656	CMPB	#3,BBC	;SEE IF HAD 3 BLD BTHS
3921	016424	101052		BHI	DERR4B	;IF NOT: BR	
3922	016426	000337	000656	SWAB	BBC	;REPOSITION BBC	
3923	016432	022705	177767	CMP	#177767,R5	;SEE IF ON LAST EIGHT CHARS	
3924	016436	101470		BLOS	DERR6	;IF SO: BR	
3925	016440	012705	177767	MOV	#177767,R5	;SET CHAR CNTR TO 8	
3926	016444	032737	010000	000562	BIT	#10000,RDCMD	;SEE IF READ REVERSE
3927	016452	001416		BEQ	DERR4A	;IF NOT: BR	
3928	016454	012701	027604	MOV	#DATA,R1	;GET START OF BUFFER	
3929	016460	012702	033612	MOV	#RDATA,R2	;GET START OF BUFFER	
3930	016464	062701	000010	ADD	#10,R1		
3931	016470	062702	000010	ADD	#10,R2	;POINT TO START +10	
3932	016474	032737	000001	000556	BIT	#1,FMCNT	;SEE IF ODD FRAME COUNT
3933	016502	001450		BEQ	DEREX	;IF NOT: BR	
3934	016504	105722		TSTB	(R2),	;BUMP POINTER	
3935	016506	000446		BR	DEREX		
3936	016510	013737	000556	000642	DERR4A: MOV	FMCNT,TEMP1	;LOAD CHAR COUNT
3937	016516	005437	000642	NEG	TEMP1	;...G FORM TWO'S COMPLEMENT	
3938	016522	162737	000010	000642	SUB	#10,TEMP1	;POINT TO BUFFER -8
3939	016530	013701	000642	MOV	TEMP1,R1	;POINT TO NEXT CHAR	
3940	016534	062701	027604	ADD	#DATA,R1	;POINT TO NEXT WRITE CHAR	
3941	016540	013702	000642	MOV	TEMP1,R2	;POINT TO END OF READ DATA -8 FORWARD	
3942	016544	062702	033612	ADD	#RDATA,R2	;POINT TO NEXT CHAR	
3943	016550	000425		BR	DEREX	;EXIT	
3944	016552	000337	000656	DERR4B: SWAB	BBC	;REPOSITION BBC	
3945	016556	062705	000024	ADD	#24,R5	;SKIP 20 CHARS	
3946	016562	103416		BCS	DERR6	;IF EXCEED RECORD SIZE: BR	
3947	016564	032737	010000	000562	BIT	#10000,RDCMD	;SEE IF READ REVERSE
3948	016572	001405		BEQ	DERR5	;IF NOT: BR	
3949	016574	162701	000024	SUB	#24,R1		
3950	016600	162702	000024	SUB	#24,R2	;RESET POINTERS	
3951	016604	000407		BR	DEREX		
3952	016606	062701	000024	DERR5: ADD	#24,R1	;SKIP 20 CHARS	
3953	016612	062702	000024	ADD	#24,R2	;SKIP FORWARD 20 CHARS	
3954	016616	000402		BR	DEREX		
3955	016620	012705	177777	DERR6: MOV	#-1,R5	;SET TO EOR	

3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023

016670 005037 000642
016674 005037 000644
016700 005037 000646
016704 111137 000642
016710 111237 000644
016714 013704 000674
016720 016437 000772 000720
016726 016437 001012 000716
016734 032737 010000 000562
016742 001005
016744 124142
016746 112137 000642
016752 112237 000644
016756 004737 016770
016762 004737 017206
016766 000207
016770 113703 000642
016774 113704 000644
017000 140403
017002 001001
017004 000207
017006 012737 000010 000710
017014 132703 000001
017020 001455
017022 105737 000646
017026 001016
017030 005277 161662
017034 005777 161656
017040 100045
017042 032777 002000 161536
017050 001402
017052 004737 022772
017056 004737 017252
017062 000415
017064 005277 161630
017070 005777 161624
017074 100027
017076 032777 002000 161502

```
DRPKF: CLR TEMP1
        CLR TEMP2
        CLR TEMP3
        MOVB (R1),TEMP1 ;LOAD GOOD CHAR
        MOVB (R2),TEMP2 ;LOAD BAD CHAR
        MOV UNP,R4
        MOV PIK1(R4),BPKP
        MOV DRP1(R4),BDPP
        BIT #10000,RDCMD ;SEE IF READ REVERSE
        BNE DRPK ;IF SO: BR
        CNPB -(R1),-(R2) ;POINT TO CHAR
        MOVB (R1)+,TEMP1 ;LOAD GOOD CHAR
        MOVB (R2)+,TEMP2 ;LOAD BAD CHAR
DRPK: JSR PC,DROP ;GET DROPS
      JSR PC,PICK ;GET PICKS
      RTS PC ;EXIT
DROP: MOVB TEMP1,R3 ;R3 = GOOD CHAR
      MOVB TEMP2,R4 ;R4 = BAD CHAR
DPC: BICB R4,R3 ;GET DROPS/PICKS
     BNE DPCG ;IF SOME: BR
     RTS PC ;RETURN
DPCG: MOV #10,BCNT ;SET NUMBER TO CHECK
      BIT #1,R3 ;SEE IF DROPPED OR PICKED THIS BIT
      BEQ DPC2 ;IF NOT: BR
      TSTB TEMP3 ;SEE IF ON PICKS
      BNE DPC1 ;IF SO: BR
      INC #BDPP ;BUMP DROP CNTR
      BPL DPC2 ;IF NO OVERFLOW: BR
      BIT #2000,#SWR ;SEE IF HAVE PRINTED DATA
      BEQ DPC0A ;IF SO: BR
      JSR PC,PAPRT ;PRINT CYCLE NUMBER
DPC0A: JSR PC,DPPRT ;PRINT DROPS AND PICKS
       BR DPC2A
DPC1: INC #BPKP ;BUMP PICK CNTR
      TST #BPKP ;SEE IF OVERFLOW
      BPL DPC2 ;IF NOT: BR
      BIT #2000,#SWR ;SEE IF HAVE PRINTED DATA
```

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


4024	017104	001402			BEQ	DPC1A		;IF SO: BR
4025	017106	004737	022772		JSR	PC,PAPRT		;PRINT CYCLE NUMBER
4026	017112	004737	017252		DPC1A: JSR	PC,DPPRT		;PRINT DROPS AND PICKS
4027	017116	013704	000674		DPC2A: MOV	UNP,R4		
4026	017122	016403	001012		MOV	DRP1(R4),R3		;SET DROP POINTER
4029	017126	016404	000772		MOV	PIK1(R4),R4		;SET PICK POINTER
4030	017132	012737	000010	000710	MOV	#10,BCNT		;SET NUMBER OF BITS
4031	017140	005023			DPC2B: CLR	(R3),		;CLEAR DROPS
4032	017142	005024			CLR	(R4),		;CLEAR PICK
4033	017144	005337	000710		DEC	BCNT		;SEE IF DONE
4034	017150	001373			BNE	DPC2B		;IF NOT: BR
4035	017152	000207			RTS	PC		;EXIT
4036	017154	000241			DPC2: CLC			
4037	017156	106003			RORB	R3		;GET NEXT BIT
4038	017160	005337	000710		DEC	BCNT		;SEE IF DONE
4039	017164	001407			BEQ	DPC3		
4040	017166	062737	000002	000720	ADD	#2,BPKP		
4041	017174	062737	000002	000716	ADD	#2,BDPP		
4042	017202	000704			BR	DPC0		;+G CONTINUE
4043	017204	000207			DPC3: RTS	PC		;RETURN
4044	017206	013704	000674		PICK: MOV	UNP,R4		;GET UNIT POINTER
4045	017212	016437	000772	000720	MOV	PIK1(R4),BPKP		;SET PICK POINTER
4046	017220	016437	001012	000716	MOV	DRP1(R4),BDPP		;SET DROP POINTER
4047	017226	113704	000642		MOVB	TEMP1,R4		;R4 = GOOD CHAR
4048	017232	113703	000644		MOVB	TEMP2,R3		;R3 = BAD CHAR
4049	017236	112737	000001	000646	MOVB	#1,TEMP3		;SET PICK FLAG
4050	017244	004737	017000		JSR	PC,DPC		;GO CHECK PICKS
4051	017250	000207			RTS	PC		;EXIT
4052	017252	012704	025503		DPPRT: MOV	#MSG26,R4		
4053	017256	104000			TTOUTT			;PRINT DROP HEADER
4054	017260	013704	000674		MOV	UNP,R4		
4055	017264	016437	001012	000716	MOV	DRP1(R4),BDPP		;SET DROP POINTER
4056	017272	016437	000772	000720	MOV	PIK1(R4),BPKP		;SET PICK POINTER
4057	017300	062737	000016	000716	ADD	#16,BDPP		
4058	017306	062737	000016	000720	ADD	#16,BPKP		
4059	017314	012737	000010	000710	MOV	#10,BCNT		;SET NUMBER TO PRINT
4060	017322	017703	161370		DPPRT0: MOV	#BDPP,R3		
4061	017326	104002			OCTPP			;PRINT DROPS
4062	017330	005337	000710		DEC	BCNT		;SEE IF DONE
4063	017334	001404			BEQ	DPPRT1		;IF NOT: BR
4064	017336	162737	000002	000716	SUB	#2,BDPP		;BUMP POINTER
4065	017344	000766			BR	DPPRT0		;CONTINUE FOR ALL 8 BITS
4066	017346	012737	000010	000710	DPPRT1: MOV	#10,BCNT		;SET NUMBER TO PRINT
4067	017354	012704	025514		MOV	#MSG27,R4		
4068	017360	104000			TTOUTT			;PRINT PICK HEADER
4069	017362	017703	161332		DPPRT2: MOV	#BPKP,R3		
4070	017366	104002			OCTPP			;PRINT PICKS
4071	017370	005337	000710		DEC	BCNT		;SEE IF DONE
4072	017374	001404			BEQ	DPPRTX		;IF SO: BR
4073	017376	162737	000002	000720	SUB	#2,BPKP		;BUMP POINTER
4074	017404	000766			BR	DPPRT2		;CONTINUE FOR ALL 8 BITS
4075	017406	000207			DPPRTX: RTS	PC		;RETURN

```

4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101 017410 013703 000556 ERCHK: MOV FMCNT,R3 ;GET FRAME COUNT
4102 017414 032703 000001 BIT #1,R3 ;SEE IF ODD
4103 017420 001401 BEQ ERO ;IF NOT: BR
4104 017422 005303 DEC R3 ;BUMP COUNT
4105 017424 005403 ERO: NEG R3
4106 017426 032737 000020 000552 BIT #20,UDES ;SEE IF CORE DUMP
4107 017434 001402 BEQ EROB ;IF NOT: BR
4108 017436 000241 CLC
4109 017440 006003 ROR R3 ;SET TO FC/2
4110 017442 032737 000010 000672 EROB: BIT #10,MTC1 ;SEE IF WRITE OP
4111 017450 001414 BEQ ER1 ;IF SO: BR
4112 017452 032737 010000 000562 BIT #10000,RDCMD
4113 017460 001405 BEQ EROA
4114 017462 012703 033612 MOV #RDATA,R3
4115 017466 162703 000002 SUB #2,R3 ;SET POINTER
4116 017472 000405 BR ER2
4117 017474 062703 033612 EROA: ADD #RDATA,R3 ;BUILD EXPT READ ADDRESS
4118 017500 000402 BR ER2
4119 017502 062703 027604 ER1: ADD #WDATA,R3 ;BUILD EXPT WRITE ADDRESS
4120 017506 010337 021134 ER2: MOV R3,CADER ;SAVE ADDRESS
4121 017512 012704 000007 MOV #7,R4
4122 017516 012701 021136 MOV #BAER,R1
4123 017522 005021 EROA0: CLR (R1)+ ;CLEAR FLAGS
4124 017524 005304 DEC R4
4125 017526 001375 BNE ER2A0
4126 017530 020377 160760 CMP R3,#BA ;SEE IF ADDRESS OK
4127 017534 001402 BEQ ER2A1 ;IF SO: BR
4128 017536 005237 021136 INC BAER ;SET BUS ADDRESS ERROR
4129 017542 032737 000010 000672 ER2A1: BIT #10,MTC1 ;SEE IF WRITE OPER
4130 017550 001006 BNE ER2B ;IF NOT: BR
4131 017552 005777 160740 ER2A: TST #FC ;SEE IF FC=0

```

4132	017556	001441				BEQ	ER3		;IF SO: BR
4133	017560	005237	021144			INC	FCER		;SET FC ERROR
4134	017564	000436				BR	ER3		;++G
4135	017566	032737	000040	000672	ER2B:	BIT	#40,MTC1		;SEE IF SPACE OPER
4136	017574	001766				BEQ	ER2A		;IF SO: BR
4137	017576	005737	000676			TST	TMFLG		;SEE IF TM TIME
4138	017602	001011				BNE	ER2D		;IF SO: BR
4139	017604	013703	000556			MOV	FMCNT,R3		
4140	017610	005403				NEG	R3		;R3 = EXPT RECORD SIZE
4141	017612	020377	160700		ER2C:	CMP	R3,#FC		;SEE IF FC = EXPT
4142	01761E	001421				BEQ	ER3		;IF SO: BR
4143	017620	005237	021144			INC	FCER		;SET FC ERROR FLAG
4144	017624	000416				BR	ER3		;++G
4145	017626	032737	002000	000552	ER2D:	BIT	#2000,UDES		;SEE IF PE
4146	017634	001346				BNE	ER2A		;IF SO: BR
4147	017636	032737	010000	000562		BIT	#10000,RDCMD		;SEE IF READ REVERSE
4148	017644	001003				BNE	ER2E		;IF SO: BR
4149	017646	012703	000002			MOV	#2,R3		
4150	017652	000757				BR	ER2C		;LOOK FOR EXPT = 2
4151	017654	012703	000001		ER2E:	MOV	#1,R3		
4152	017660	000754				BR	ER2C		;GO CHECK FC FOR TM
4153	017662	032777	160000	160620	ER3:	BIT	#160000,#C1		;SEE IF COUNT ERROR
4154	017670	001441				BEQ	ER4		
4155	017672	017703	160622			MOV	#CS,R3		;GET CONT STATUS REG
4156	017676	042703	000307			BIC	#307,R3		;MASK OUT IR,OR,UNIT NO.
4157	017702	005703				TST	R3		;SEE IF ANY OTHER ERRORS
4158	017704	001407				BEQ	ER3A		;IF NOT: BR
4159	017706	005737	000676			TST	TMFLG		;SEE IF TAPE MARK TIME
4160	017712	001426				BEQ	ER3B		;IF NOT: BR
4161	017714	042703	001000			BIC	#1000,R3		;MASK MISSED TRANS
4162	017720	005703				TST	R3		;SEE IF ANY OTHER ERRORS
4163	017722	001022				BNE	ER3B		;IF SO: BR
4164	017724	032777	060000	160556	ER3A:	BIT	#60000,#C1		;SEE IF EITHER TRE OR MCPE
4165	017732	001420				BEQ	ER4		;IF NOT: BR
4166	017734	005737	000676			TST	TMFLG		;SEE IF TM TIME
4167	017740	001413				BEQ	ER3B		;IF NOT: BR
4168	017742	017703	160556			MOV	#ER,R3		;GET ERROR REGISTER
4169	017746	032737	000010	000552		BIT	#10,UDES		;SEE IF EVEN PARITY
4170	017754	001402				BEQ	ER3A1		;IF NOT: BR
4171	017756	042703	000100			BIC	#100,R3		;MASK PAR
4172	017762	042703	001000		ER3A1:	BIC	#1000,R3		;MASK FCE
4173	017766	001402				BEQ	ER4		;IF NO ERRORS EXCEPT FCE: BR
4174	017770	005237	021140		ER3B:	INC	CONER		;SET CONT ERROR FLAG
4175	017774	032777	040000	160520	ER4:	BIT	#40000,#DS		;SEE IF DRIVE ERROR
4176	020002	001420				BEQ	ER6		;IF NOT: BR
4177	020004	005737	000676			TST	TMFLG		;SEE IF TAPE MARK TIME
4178	020010	001413				BEQ	EP4A		;IF NOT: BR
4179	020012	017703	160506			MOV	#ER,R3		;GET ER
4180	020016	032737	000010	000552		BIT	#10,UDES		;SEE IF EVEN PARITY
4181	020024	001402				BEQ	ER4A1		;IF NOT: BR
4182	020026	042703	000100			BIC	#100,R3		;MASK PAR
4183	020032	042703	001000		ER4A1:	BIC	#1000,R3		;MASK OUT FCE
4184	020036	001402				BEQ	ER6		;++G & BR IF NO OTHER ERR BITS ARE SET
4185	020040	005237	021142		ER4A:	INC	DRVER		;SET DRIVER ERROR FLAG
4186	020044	032737	002000	000552	ER6:	BIT	#2000,UDES		
4187	020052	001071				BNE	ERPT		;IF IN PE MODE: BR

4188	020054	032777	020000	160524	BIT	@20000,@SWR	;SEE IF NO DATA CHECK
4189	020062	001065			BNE	ERPT	;IF NOT: BR (ALLOW READ OF UNKNOWN TAPES)
4190	020064	032737	000040	000672	BIT	@40,MTC1	;SEE IF WRITE OR READ OP
4191	020072	001461			BEQ	ERPT	;IF NOT: BR
4192	020074	005737	000676		TST	TMFLG	;SEE IF TAPE MARK TIME
4193	020100	001413			BEQ	ER6A	;IF NOT: BR
4194	020102	013737	015540	021156	MOV	EXCRC,CRCV	;SAVE CRC
4195	020110	013737	015542	021154	MOV	EXLRC,LRCV	;SAVE LRC
4196	020116	005037	015540		CLR	EXCRC	
4197	020122	012737	000023	015542	MOV	@23,EXLRC	;SET CRC/LRC FOR TM
4198	020130	032737	000060	000552	ER6A: BIT	@60,UDES	;SEE IF FORMAT 14
4199	020136	001037			BNE	ERPT	;IF NOT: BR
4200	020140	017703	160364		MOV	@CC,R3	;GET CRC CHARACTER
4201	020144	042703	177000		BIC	@177000,R3	
4202	020150	023703	015540		CMP	EXCRC,R3	
4203	020154	001402			BEQ	ER7	;IF CRC GOOD: BR
4204	020156	005237	021150		INC	CRCER	;SET ERROR FLAG
4205	020162	017703	160346		ER7: MOV	@MR,R3	;GET LRC
4206	020166	000303			SWAB	R3	
4207	020170	005703			TST	R3	
4208	020172	100002			BPL	ER10	
4209	020174	052703	000400		BIS	@400,R3	
4210	020200	042703	177000		ER10: BIC	@177000,R3	
4211	020204	023703	015542		CMP	EXLRC,R3	
4212	020210	001412			BEQ	ERPT	;IF LRC GOOD: BR
4213	020212	010337	021152		MOV	R3,ACTLRC	;SAVE ACTUAL LRC
4214	020216	005237	021146		INC	LRCER	;SET LRC ERROR FLAG
4215	020222	032737	010000	000562	BIT	@10000,RDCMD	;SEE IF READ REVERSE
4216	020230	001402			BEQ	ERPT	;IF NOT: BR
4217	020232	005037	021146		CLR	LRCER	;ELSE CLEAR LRC ERROR
4218	020236	012703	000006		ERPT: MOV	@6,R3	
4219	020242	005037	000706		CLR	SERFL	;CLEAR ERROR FLAG
4220	020246	005037	000722		CLR	ERSAV	
4221	020252	012704	021136		MOV	@BAER,R4	
4222	020256	005724			ERPTT: TST	(R4)	;SEE IF ANY ERROR
4223	020260	001004			BNE	ERPTG	;IF SO: BR
4224	020262	005303			DEC	R3	
4225	020264	001374			BNE	ERPTT	
4226	020266	000137	021100		JMP	ERPX1	
4227	020272	005237	000706		ERPTG: INC	SERFL	;SET ERROR FLAG
4228	020276	017737	160222	000722	MOV	@ER,ERSAV	;SAVE ERROR REGISTER
4229	020304	032777	002000	160274	BIT	@2000,@SWR	;SEE IF PRINT
4230	020312	001420			BEQ	ERPTO	;IF SO: BR
4231	020314	022737	000002	000712	CMP	@2,RTYFL	;SEE IF READ RETRY
4232	020322	001006			BNE	ERPTG1	;IF NOT: BR
4233	020324	013703	000702		MOV	RTCNT,R3	
4234	020330	005203			INC	R3	;BUMP RETRY COUNT
4235	020332	020337	000602		CMP	R3,RETRY	;SEE IF LAST RETRY
4236	020336	001406			BEQ	ERPTO	;IF SO: BR
4237	020340	022737	000002	021142	ERPTG1: CMP	@2,DRVER	;SEE IF TM STATUS ERROR
4238	020346	001402			BEQ	ERPTO	;IF SO: BR
4239	020350	000137	021002		JMP	ERPX0	
4240	020354	005237	000670		ERPTO: INC	PFLG	
4241	020360	004737	022772		JSR	PC,PAPRT	;PRINT HEADER
4242	020364	013704	000652		MOV	EMADDR,R4	
4243	020370	104000			TTOUIT		;PRINT ERROR TYPE

4244	020372	004737	021160		JSR	PC,↑RPRT	;PRINT F OR R
4245	020376	005737	000676		TST	TMFLG	
4246	020402	001407			BEQ	ERPT1	
4247	020404	022737	026312	000652	CMP	#MSG54,EMADDR	
4248	020412	001403			BEQ	ERPT1	
4249	020414	012704	026330		MOV	#MSG56,R4	;PRINT TM
4250	020420	104000			TTOUTT		
4251	020422	005737	021140		ERPT1: TST	CONER	
4252	020426	001414			BEQ	ERPT2	;IF NO CONT ERROR: BR
4253	020430	012704	025333		MOV	#MSG23,R4	
4254	020434	104000			TTOUTT		;PRINT C1 TAG
4255	020436	017703	160046		MOV	@C1,R3	
4256	020442	104002			OCTPP		;PRINT CONTROL 1
4257	020444	012704	025360		MOV	#MSG23D,R4	;PRINT CS TAG
4258	020450	104000			TTOUTT		
4259	020452	017703	160042		MOV	@CS,R3	
4260	020456	104002			OCTPP		;PRINT CONT STATUS
4261	020460	005737	021142		ERPT2: TST	DRVER	
4262	020464	001414			BEQ	ERPT3	;IF SO DRIVE ERROR: BR
4263	020466	012704	025366		MOV	#MSG23E,R4	
4264	020472	104000			TTOUTT		;PRINT DS TAG
4265	020474	017703	160022		MOV	@DS,R3	
4266	020500	104002			OCTPP		;PRINT DRIVE STATUS
4267	020502	012704	025373		MOV	#MSG23F,R4	
4268	020506	104000			TTOUTT		;PRINT ER TAG
4269	020510	017703	160010		MOV	@ER,R3	
4270	020514	104002			OCTPP		;PRINT DRIVE ERROR
4271	020516	005737	021136		ERPT3: TST	BAER	
4272	020522	001416			BEQ	ERPT4	;IF NO BA ERROR: BR
4273	020524	012704	025346		MOV	#MSG23B,R4	
4274	020530	104000			TTOUTT		;PRINT BA TAG
4275	020532	017703	157756		MOV	@BA,R3	
4276	020536	104002			OCTPP		;PRINT BUS ADDRESS
4277	020540	012737	000255	000636	MOV	@255,TOB	
4278	020546	004737	023724		JSR	PC,TOG	;PRINT /
4279	020552	013703	021134		MOV	CADER,R3	
4280	020556	104002			OCTPP		;PRINT EXPT BUS ADDRESS
4281	020560	005737	021144		ERPT4: TST	FCER	
4282	020564	001406			BEQ	ERPT5	;IF NO FC ERROR: BR
4283	020566	012704	025353		MOV	#MSG23C,R4	
4284	020572	104000			TTOUTT		;PRINT FC TAG
4285	020574	017703	157716		MOV	@FC,R3	
4286	020600	104002			OCTPP		;PRINT FRAME COUNT
4287	020602	012704	025341		ERPT5: MOV	#MSG23A,R4	
4288	020606	104000			TTOUTT		;PRINT WC TAG
4289	020610	017703	157676		MOV	@WC,R3	
4290	020614	104002			OCTPP		;PRINT WORD COUNT
4291	020616	005737	021150		TST	CRCER	
4292	020622	001420			BEQ	ERPT5A	;IF NO CRC ERROR: BR
4293	020624	012704	026355		MOV	#MSG58,R4	
4294	020630	104000			TTOUTT		;PRINT CRC TAG
4295	020632	017703	157672		MOV	@CC,R3	
4296	020636	042703	177000		BIC	@177000,R3	
4297	020642	104002			OCTPP		;PRINT ACTUAL CRC
4298	020644	012737	000255	000636	MOV	@255,TOB	
4299	020652	004737	023724		JSR	PC,TOG	

4300	020656	013703	015540		MOV	EXCHC,R3	
4301	020662	104002			OCTPP		;PRINT EXPECTED CRC
4302	020664	005737	021146		ERPT5A: TST	LRCER	
4303	020670	001416			BEQ	ERPT6	;IF NO LRC ERROR: BR
4304	020672	012704	026363		MOV	#MSG59,R4	
4305	020676	104000			TTOUTT		;PRINT LRC TAG
4306	020700	013703	021152		MOV	ACTLRC,R3	
4307	020704	104002			OCTPP		;PRINT ACTUAL LRC
4308	020706	012737	000255	000636	MOV	#255,TOB	
4309	020714	004737	023724		JSR	PC,TOG	
4310	020720	013703	015542		MOV	EXLRC,R3	
4311	020724	104002			OCTPP		;PRINT EXPECTED LRC
4312	020726	005737	021142		ERPT6: TST	DRVER	
4313	020732	001422			BEQ	ERPT7	;IF NO DRIVE ERROR: BR
4314	020734	032737	002000	000552	BIT	#2000,UDES	
4315	020742	001416			BEQ	ERPT7	;IF NO PE: BR
4316	020744	017704	157554		MOV	BER,R4	
4317	020750	042704	075477		BIC	#75477,R4	;MASK OUT ALL BUT BITS 15,10,7,6
4318	020754	005704			TST	R4	
4319	020756	001410			BEQ	ERPT7	;IF NO CONDITIONALS SET: BR
4320	020760	012704	025405		MOV	#MSG23H,R4	
4321	020764	104000			TTOUTT		;PRINT CC TAG
4322	020766	017703	157536		MOV	#CC,R3	
4323	020772	042703	177000		BIC	#177000,R3	;MASK CC
4324	020776	104002			OCTPP		;PRINT CHECK CHARACTERS
4325	021000	000240			ERPT7: NOP		
4326	021002	032777	100000	157576	ERPX0: BIT	#100000,BSWR	;SEE IF STOP ON ERROR
4327	021010	001412			BEQ	ERPX	;IF NOT: BR
4328	021012	104006			STOPP		
4329	021014	005737	000670		TST	PFLG	;SEE IF HAVE PRINTED
4330	021020	001006			BNE	ERPX	;IF SO: BR
4331	021022	032777	002000	157556	BIT	#2000,BSWR	;SEE IF SHOULD PRINT
4332	021030	001002			BNE	ERPX	;IF NOT: BR
4333	021032	000137	020354		JMP	ERPT0	;PRINT ERROR
4334	021036	005037	000670		ERPX: CLR	PFLG	
4335	021042	012777	000011	157440	MOV	#11,BC1	;DRIVE CLEAR
4336	021050	017704	157452		MOV	BAS,R4	
4337	021054	010477	157446		MOV	R4,BAS	;CLEAR AS
4338	021060	013704	000510		MOV	C1,R4	
4339	021064	005204			INC	R4	
4340	021066	152714	000100		BISB	#100,(R4)	;RESET TRE
4341	021072	013777	000552	157442	MOV	UDES,BC2	;RESET TC
4342	021100	032737	000040	000672	ERPX1: BIT	#40,MTC1	
4343	021106	001411			BEQ	ERPX2	;IF NOT READ/WRITE OP: BR
4344	021110	005737	000676		TST	TMFLG	
4345	021114	001406			BEQ	ERPX2	;IF NOT TM TIME: BR
4346	021116	013737	021156	015540	MOV	CRCV,EXCRC	;RESTORE CRC
4347	021124	013737	021154	015542	MOV	LRCV,EXLRC	;RESTORE LRC
4348	021132	000207			ERPX2: RTS	PC	;EXIT
4349	021134	000000			CADER: 0		;EXPT ADDRESS SAVE
4350	021136	000000			BAER: 0		
4351	021140	000000			CONER: 0		
4352	021142	000000			DRVER: 0		
4353	021144	000000			FCER: 0		
4354	021146	000000			LRCER: 0		
4355	021150	000000			CRCER: 0		

4356 021152 000000
4357 021154 000000
4358 021156 000000

ACTLRC: 0
LRCSV: 0
CRCSV: 0

4359
4360
4361
4362
4363
4364
4365
4366
4367

;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 021160 032737 000010 000672 FRPRT:
4370 021166 001413
4371 021170 032737 000002 000672
4372 021176 001404
4373 021200 012704 025243
4374 021204 104000
4375 021206 000403
4376 021210 012704 025240
4377 021214 104000
4378 021216 000207
4379

BIT #10,MTC1 ;SEE IF WRITE COMMAND
BEQ FREX ;IF SO: BR
BIT #2,MTC1 ;SEE IF REVERSE
BEQ FRO ;IF NOT: BR
MOV #MSG17,R4
TTOUTT ;PRINT R
BR FREX
FRO: MOV #MSG16,R4
TTOUTT ;PRINT F
FREX: RTS PC ;EXIT

```

4380
4381 ;*****
4382 ;TAPE COMMAND EXECUTE SUBROUTINE:
4383 ;
4384 ;THIS SUBROUTINE IS USED TO EXECUTE THE
4385 ;MAG TAPE COMMAND DESCRIBED BY THE READ
4386 ;OR WRITE ROUTINE. THE FINAL COMMAND IS
4387 ;SENT TO THE DEVICE REGISTER ALONG WITH THE
4388 ;INTERRUPT ENABLE AND GO BITS.
4389 ;ONCE THE COMMAND IS ISSUED, AN INTERRUPT
4390 ;TIMER IS STARTED AND IF NO INTERRUPT IS RETURNED
4391 ;BEFORE TIME OUT OCCURS, AN ERROR WILL BE
4392 ;PRINTED AND THE PROGRAM STOPPED. TESTING MAY
4393 ;BE RESUMED BY PRESSING THE CONTINUE SWITCH.
4394 ;TWO INTERRUPT HANDLERS ARE USED, ONE FOR MAG TAPE
4395 ;AND ANOTHER FOR TELETYPE (TTY).
4396 ;UPON RECEIPT OF A MAG TAPE INTERRUPT, HOUSEKEEPING
4397 ;IS PERFORMED AND CONTROL RETURNED TO THE CALLING
4398 ;ROUTINE (READ,WRITE,ETC).
4399 ;RECEIPT OF A TTY INTERRUPT WILL CAUSE THE
4400 ;PROGRAM TO CHECK FOR ENTRY OF A CNTRL C CHARACTER.
4401 ;IF NOT CNTRL C, THEN CONTINUATION OF WAIT FOR MAG
4402 ;TAPE INTERRUPT IS RETURNED. IF, HOWEVER, THE TTY
4403 ;INTERRUPT WAS CAUSED BY ENTRY OF A CNTRL C,
4404 ;THEN AT THIS TIME REQUESTS FOR NEW STALL VALUES
4405 ;ARE PRINTED AND THE RESPONSES ENTERED. RESUMPTION
4406 ;OF TAPE INTERRUPT WAIT IS THEN RESUMED.
4407 ;*****
4408
4409 021220 005037 000642 TAPG: CLR TEMP1
4410 021224 013777 000550 157266 MOV DVN,BCS ;SET DRIVE NO.
4411 021232 032777 010000 157262 TAPG0: BIT #10000,BDS ;SEE IF HAVE MOL
4412 021240 001026 BNE TAPG3 ;IF SO: BR
4413 021242 005237 000642 INC TEMP1 ;SEE IF TIMED OUT
4414 021246 001371 BNE TAPG0 ;WAIT FOR READY
4415 021250 004737 022772 JSR PC,PAPRT ;PRINT CYCLE NUMBER
4416 021254 032737 000010 000672 BIT #10,MTC1 ;SEE IF WRITE OP
4417 021262 001004 BNE TAPG1 ;IF NOT: BR
4418 021264 012704 025102 MOV #MSG5,R4
4419 021270 104000 TTOUTT ;PRINT WRITE ERR
4420 021272 000405 BR TAPG2
4421 021274 012704 025107 TAPG1: MOV #MSG6,R4
4422 021300 104000 TTOUTT ;PRINT READ ERR
4423 021302 004737 021160 JSR PC,FRPRT ;PRINT F OR R
4424 021306 012704 025463 TAPG2: MOV #MSG25,R4
4425 021312 104000 TTOUTT ;PRINT NO MOL ERR
4426 021314 104006 STOPP
4427 021316 032777 020000 157176 TAPG3: BIT #20000,BDS ;SEE IF PIP RESET
4428 021324 001411 BEQ TAPG3F ;IF SO: BR
4429 021326 004737 022772 JSR PC,PAPRT ;PRINT HEADER
4430 021332 012704 027442 MOV #MSG116,R4
4431 021336 104000 TTOUTT ;PRINT REWINDING MESSAGE
4432 021340 032777 020000 157154 11: BIT #20000,BDS
4433 021346 001374 BNE 11 ;AWAIT PIP RESET
4434 021350 022737 000026 000672 TAPG3F: CMP #26,MTC1 ;SEE IF WRITE TM
4435 021356 001003 BNE TAPG3A ;IF NOT: BR

```



```

4436 021360 012704 177777      MOV      @-1,R4      ;ELSE SET FC FOR -1
4437 021364 000406              BR      TAPG3B
4438 021366 013704 000556      TAPG3A: MOV      FMCNT,R4
4439 021372 032704 000001      BIT      @1,R4
4440 021376 001401              BEQ     TAPG3B
4441 021400 005304              DEC     R4
4442 021402 000261      TAPG3B: SEC
4443 021404 006004              ROR     R4      ;SET WC = FC/2 FOR NORMAL FORMAT
4444 021406 032737 000020 000552  BIT      @20,UDES  ;SEE IF CORE DUMP FORMAT
4445 021414 001402              BEQ     TAPG3C  ;IF NOT: BR
4446 021416 000261              SEC
4447 021420 006004              ROR     R4      ;SET WC = FC/4 FOR CORE DUMP
4448 021422 010477 157064      TAPG3C: MOV      R4,SMC  ;SET WORD COUNT
4449 021426 012777 000011 157054  MOV      @11,BC1  ;DRIVE CLEAR
4450 021434 017777 157056 157054  MOV      @FC,@FC  ;RESET FC LOADED
4451 021442 005737 000566      TST     INTRF    ;SEE IF INTERCHANGE READ
4452 021446 001407              BEQ     TAPG3D  ;IF NOT: BR
4453 021450 032737 000040 000672  BIT      @40,MT1  ;SEE IF READ OP
4454 021456 001403              BEQ     TAPG3D  ;IF NOT: BR
4455 021460 012777 000003 157046  MOV      @3,SMR   ;SET INTERCHANGE READ MAINT. MODE
4456 021466 013704 000672      TAPG3D: MOV      MTC1,R4  ;GET COMMAND
4457 021472 042704 177707      BIC     @177707,F:4 ;MASK OP CODE
4458 021476 022704 000030      CMP     @30,R4   ;SEE IF SPACE OP CODE
4459 021502 001403              BEQ     TAPG3E  ;IF SO: BR
4460 021504 012737 177740 000666  MOV      @-4C,STAL ;SET INTERRUPT DELAY MULT TO 40
4461 021512 052737 000101 000672  TAPG3E: BIS      @101,MT1 ;SET INTERRUPT ENABLE AND GO
4462 021520 000240              NOP
4463 021522 013777 000672 156760  MOV      MTC1,BC1 ;EXECUTE COMMAND
4464 021530 005077 157050      CLR     @PSW    ;CLEAR PRIORITY
4465 021534 005037 000642      CLR     TEMP1
4466 021540 005237 000642      TAPG4: INC     TEMP1  ;SEE IF HAVE TIMED OUT
4467 021544 001375              BNE     TAPG4   ;IF NOT: BR
4468 021546 005237 000666      INC     STAL
4469 021552 001372              BNE     TAPG4   ;DO TIME DELAY MULTIPLIER
4470 021554 012777 000340 157022  TAPG5: MOV      @340,@PSW ;RESET PRIORITY
4471 021562 032777 002000 157016  BIT      @2000,@SMR ;SEE IF SHOULD PRINT ERRORS
4472 021570 001012              BNE     TAPG6   ;IF NOT: BR
4473 021572 004737 022772      JSR     PC,PAPRT ;PRINT CYCLE NUMBER
4474 021576 013704 000652      MOV     EMADDR,R4
4475 021602 104000              TTOUTT  ;PRINT ERROR OP
4476 021604 004737 021160      JSR     PC,FRPRT ;PRINT F OR R
4477 021610 012704 025443      MOV     @MSG24,R4
4478 021614 104000              TTOUTT  ;PRINT NO INTERRUPT
4479 021616 005777 156764      TAPG6: TST     @SMR  ;...G BRANCH IF CONTINUE ON ERROR
4480 021622 100001              BPL     TAPG7
4481 021624 104006              STOPP
4482 021626 000137 021752      TAPG7: JMP     MTINTA ;RETURN TO CALLING ROUTINE
4483

```

```

4484
4485
4486
4487 021632 012777 000340 156744 TTINT: MOV #340,SPSW ;RESET PSW
4488 021640 017746 156746 MOV #TKB,-(SP) ;..G GET CHARACTER
4489 021644 042716 000200 BIC #200,(SP) ;..G STRIP PARITY BIT
4490 021650 122716 000003 CMPB #3,(SP) ;..G SEE IF CONT C
4491 021654 001427 BEQ TTINTO ;IF SO; BR
4492 021656 122716 000007 CMPB #7,(SP) ;..G CHECK FOR CNTL G
4493 021662 001007 BNE 1# ;JH
4494 021664 022737 000176 000606 CMP #SWREG,SWR ;IS SOFTWARE SWITCH REGISTER USED
4495 021672 001024 BNE RETURN ;NO, GET OUT
4496 021674 004737 024572 JSR PC,CNTG ;GO CHANGE SWREG
4497 021700 000421 BR RETURN ;..G GO TO EXIT
4498
4499 021702 122716 000023 1#: CMPB #23,(SP) ;..JH SEE IF +S WAS TYPED
4500 021706 001004 BNE 2# ;..JH BRANCH IF NOT
4501 021710 112737 000377 000746 MOVB #377,#CTRLS ;..JH SET XON FLAG
4502 021716 000412 BR RETURN ;..JH AND RETURN
4503 021720 122716 000021 2#: CMPB #21,(SP) ;..JH SEE IF +Q WAS TYPED
4504 021724 001007 BNE RETURN ;..JH BRANCH IF NOT
4505 021726 105037 000746 CLRB #CTRLS ;..JH CLEAR XON FLAG
4506 021732 000404 BR RETURN ;..JH AND RETURN
4507
4508 021734 010046 TTINTO: MOV RO,-(SP) ;..G SAVE RO(REC CNTR)
4509 021736 004737 013776 JSR PC,TINP4 ;GO GET STALL VALUES
4510 021742 012600 MOV (SP)+,RO ;..G RESTORE RO(REC CNTR)
4511 021744 005726 RETURN: TST (SP)+ ;..G POP CHAR OFF STACK
4512 021746 000002 RTI ;RETURN
4513
4514 ;MAG TAPE INTERRUPT HANDLER*****
4515
4516 021750 000240 MTINT: NOP
4517 021752 042777 000037 156554 MTINTA: BIC #37,BMR ;CLEAR MAINT MODE
4518 021760 013716 000662 MOV RTRN,(SP) ;..G GET RETURN ADDRESS
4519 021764 000002 RTI ;..G RETURN

```

4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574

021766 012704 027252
021772 104000
021774 012705 000650
022000 012701 000001
022004 012702 000001
022010 012703 000000
022014 004737 023410
022020 012704 027047
022024 104000
022026 012705 000742
022032 012701 000001
022036 012702 000001
022042 012703 000000
022046 004737 023410
022052 005037 000736
022056 004737 022216
022062 005737 000042
022066 001404
022070 012737 000001 000742
022076 000414
022100 012704 027013
022104 104000
022106 012704 027027
022112 104000
022114 013703 000736
022120 104002
022122 012704 027036
022126 104000
022130 012700 000750
022134 005710
022136 100403
022140 012003
022142 104002
022144 000773
022146 004737 022422
022152 004737 022616
022156 022737 000007 000736
022164 001403
022166 005237 000736
022172 000731
022174 005737 000742
022200 001405
022202 004737 005102
022206 005737 000734
022212 001317
022214 000000

```
*****  
;AUTO SEQUENCE  
;  
;THIS ROUTINE ,ENTERED VIA STARTING ADDRESS 240  
;WILL EXERCISE ALL AVAILABLE SLAVES ON ALL AVAILABLE  
;DRIVES IN BOTH PE AND NRZ ACCORDING TO THE PRESELECTED  
;TEST PLAN. IF NRZ ONLY, PE TESTING WILL NOT BE ATTEMPTED.  
;*****  
ASEQ:  MOV    #MSG108,R4  
      TTOUTT                ;PRINT NRZ ONLY REQUEST  
      MOV    #NRZOF,R5     ;SET ADDRESS OF FLAG  
      MOV    #1,R1          ;SET SIZE OF ENTRY  
      MOV    #1,R2          ;SET UPPER LIMIT  
      MOV    #0,R3          ;SET LOWER LIMIT  
      JSR    PC,TTR         ;GO GET RESPONSE  
      MOV    #MSG104,R4  
      TTOUTT                ;REQUEST CONT OR NOT  
      MOV    #ASEQCF,R5    ;SET ADDRESS OF ENTRY  
      MOV    #1,R1          ;SET SIZE OF ENTRY  
      MOV    #1,R2          ;SET UPPER LIMIT  
      MOV    #0,R3          ;SET LOWER LIMIT  
      JSR    PC,TTR         ;GO GET INPUT  
ASEQ0:  CLR    ADRVN        ;CLEAR DRV NUM  
ASEQ1:  JSR    PC,HRDS      ;GO SELECT HARDWARE CONFIGURATION  
      TST    #M42          ;AUTO MODE? .. C.W  
      BEQ    1$           ;BRANCH - IF NO .. C.W  
      MOV    #1,ASEQCF     ;SET AUTO SEQ FLAG .. C.W  
      BR    2$           ;DO AUTO SEQ TESTS .. C.W  
1$:    MOV    #MSG101,R4  
      TTOUTT                ;PRINT DIVIDER  
      MOV    #MSG102,R4  
      TTOUTT                ;PRINT TMO2 NUMBER  
      MOV    ADRVN,R3      ;PRINT TMO2  
      OCTPP  
      MOV    #MSG103,R4  
      TTOUTT                ;PRINT SLAVE HDR  
2$:    MOV    #UN1,R0      ;POINT TO START OF SLAVE TABLE  
ASEQ2:  TST    (R0)        ;SEE IF END  
      BMI    ASEQ3        ;IF SO: BR  
      MOV    (R0)+,R3  
      OCTPP  
      BR    ASEQ2        ;PRINT SLAVE TABLE  
      ;DO ALL  
ASEQ3:  JSR    PC,AMOD1    ;GO DO MODE 1(NRZ)  
      JSR    PC,AMOD2    ;GO DO MODE 2(PE)  
ASEQ4:  CMP    #7,ADRVN   ;SEE IF DONE ALL DRIVES  
      BEQ    ASEQX        ;IF SO: BR  
      INC    ADRVN        ;BUMP DRIVE NUMBER  
      BR    ASEQ1        ;CONTINUE  
ASEQX:  TST    ASEQCF     ;CONTINUOUS AUTO SEQUENCE? .. C.W  
      BEQ    1$           ;BRANCH - IF NO .. C.W  
      JSR    PC,TEND      ;GO DO ACT END OF PASS  
      TST    ASEQF        ;CONTINUE  
      BNE    ASEQO        ;GO START AGAIN  
1$:    HALT
```

```

4575
4576
4577
4578 022216 005037 005160
4579 022222 005037 000642
4580 022226 012777 000040 156264
4581 022234 013777 000736 156256
4582 022242 017701 156270
4583 022246 032777 010000 156244
4584 022254 001403
4585 022256 005726
4586 022260 000137 022156
4587 022264 042701 002007
4588 022270 022701 140010
4589 022274 001370
4590 022276 005000
4591 022300 012701 000750
4592 022304 005737 003042
4593 022310 001410
4594 022312 122737 000006 000041
4595 022320 001004
4596 022322 005737 000736
4597 022326 001001
4598 022330 005200
4599
4600 022332 010077 156204
4601 022336 032777 010000 156156
4602 022344 001403
4603 022346 005237 000642
4604 022352 010021
4605 022354 022700 000007
4606 022360 001402
4607 022362 005200
4608 022364 000762
4609 022366 005737 000642
4610 022372 001731
4611 022374 013737 000642 005160
4612 022402 000337 000642
4613 022406 053737 000642 005160
4614 022414 012711 177777
4615 022420 000207

```

```

;SUBROUTINE TO SELECT AUTO SEQUENCE HARDWARE*****
;CLEAR EOT UNIT CNTR
;INIT
;SET DRIVE
;READ DRIVE TYPE
;TEST FOR NON-EXISTANT DRIVE
;IF DRIVE AVAIL: BR
;RESET STACK POINTER
;GO SEE IF TRIED ALL DRIVES
;MASK SLAVE TYPE
;..G SEE IF TU16/TE16 TAPE
;IF NOT: BR
;SET START OF SLAVE TABLE
;..G BRANCH IF NOT IN CHAIN MODE
;..G BRANCH IF NOT LOADED VIA TMOP
;..G BRANCH IF NOT DRIVE 0
;..G
;..G DO NOT TEST DRIVE 0 SLAVE 0
;..G IF TMOP CHAIN
;SELECT SLAVE
;SEE IF SLAVE AVAIL FOR TEST(MOL)
;IF NOT: BR
;SET SLAVE FOUND FLAG
;LOAD SLAVE TABLE
;SEE IF DONE ALL SLAVES
;IF SO: BR
;ELSE BUMP SLAVE NUMBER
;CONTINUE SELECTION
;SEE IF FOUND ANY SLAVES
;IF NOT: BR
;SET NUMBER OF UNITS
;SET EOT CNTR
;TERMINATE SLAVE TABLE
;RETURN TO SEQ

```

```

4616
4617
4618
4619 022422 005037 000654
4620 022426 012701 000750
4621 022432 052721 001700
4622 022436 005111
4623 022440 001402
4624 022442 005111
4625 022444 000772
4626 022446 005111
4627 022450 004737 005174
4628 022454 012737 000006 000740
4629 022462 012737 174000 000556
4630 022470 012737 000100 000554
4631 022476 013737 000736 000550
4632 022504 012737 000001 000560
4633 022512 005037 000564
4634 022516 005037 000566
4635 022522 004737 003416
4636 022526 012737 000010 000560
4637 022534 004737 003416
4638 022540 012737 000014 000560
4639 022546 004737 003416
4640 022552 005737 000650
4641 022556 001411
4642 022560 012737 177777 000740
4643 022566 012737 153624 000624
4644 022574 012737 032561 000626
4645 022602 012737 177777 000560
4646 022610 004737 003416
4647 022614 000207

;SUBROUTINE TO SELECT NRZ AUTO TEST MODE*****
AMOD1: CLR BLCNTR ;ASSURE BLOCK COUNTER IS 0
MOV #UN1,R1 ;GET START OF SLAVE TABLE
AMOD1A: BIS #1700,(R1) ;SET ALL SLAVE TO NRZ,NORM,000
COM (R1)
BEQ AMOD1B ;IF FILLED ALL SLAVES: BR
COM (R1)
BR AMOD1A ;ELSE DO ALL
AMOD1B: COM (R1)
JSR PC,RWINDA ;GO REWIND ALL AVAIL SLAVES
MOV #6,ABLCNT ;SET NUMBER OF BLOCKS FOR MODE 1
MOV #-4000,FCNT ;SET FC = 4000
MOV #100,RCNT ;SET REC CNTR = 100
MOV ADRVN,DVN ;SELECT DRIVE
MOV #1,PATRN ;SELECT PATTERN 1
CLR TMEX ;ASSURE NO TMK
CLR INTRF ;ASSURE NORMAL READ
JSR PC,STAUTO ;GO DO AUTO MODE 1
MOV #10,PATRN ;SELECT PATTERN 10
JSR PC,STAUTO ;GO DO PATTERN 10
MOV #14,PATRN ;SELECT PATTERN 14
JSR PC,STAUTO
TST NRZOF ;SEE IF NRZ ONLY
BEQ AMOD1C ;IF NOT: BR
MOV #-1,ABLCNT ;FORCE TO EOT
MOV #153624,RANBAS ;RESET RANDOM DATA BASE
MOV #32561,RANSAV ;SELECT AUTO RANDOM DATA
AMOD1C: MOV #-1,PATRN
JSR PC,STAUTO
RTS PC ;RETURN TO SEQ

```

```

4648
4649
4650
4651 022616 005737 003042
4652 022622 001003
4653 022624 005737 000650
4654 022630 001057
4655 022632 005037 000654
4656 022636 012701 000750
4657 022642 042711 001700
4658 022646 052721 002300
4659 022652 005111
4660 022654 001402
4661 022656 005111
4662 022660 000770
4663 022662 005111
4664 022664 004737 005174
4665 022670 012737 000006 000740
4666 022676 012737 174000 000556
4667 022704 012737 000100 000554
4668 022712 012737 000010 000560
4669 022720 004737 003416
4670 022724 012737 000014 000560
4671 022732 004737 003416
4672 022736 012737 000015 000560
4673 022744 004737 003416
4674 022750 012737 177777 000740
4675 022756 012737 177777 000560
4676 022764 004737 003416
4677 022770 000207
4678
4679

```

```

;SUBROUTINE TO SELECT PE AUTO TEST MODE*****
AMOD2: TST CHNFLG ;++G BRANCH IF IN CHAIN MODE
      BNE 1# ;++G
      TST NRZOF ;SEE IF NRZ ONLY
      BNE AMOD2X ;IF SO: BR
1#: CLR BLCNTR ;CLEAR BLOCK CNTR
     MOV #UN1,R1 ;SET START OF SLAVE TABLE
AMOD2A: BIC #1700,(R1) ;CLEAR NRZ
        BIS #2300,(R1)+ ;SET TO PE NORM, ODD
        COM (R1) ;SEE IF END OF TABLE
        BEQ AMOD2B ;IF SO: BR
        COM (R1)
        BR AMOD2A ;CONTINUE
AMOD2B: COM (R1)
        JSR PC,RWINDA ;REWIND ALL SLAVES
        MOV #6,ABL CNT ;SET AUTO BLOCK COUNT
        MOV #-4000,FMCNT ;SET FC = 4000
        MOV #100,RCNT ;SET REC CNTR TO 100
        MOV #10,PATRN ;SELECT PATTERN 10
        JSR PC,STAUTO ;GO DO AUTO SEQ
        MOV #14,PATRN ;SELECT PATTERN 14
        JSR PC,STAUTO
        MOV #15,PATRN ;SELECT PATTERN 15
        JSR PC,STAUTO
        MOV #-1,ABL CNT ;FORCE TO END OF TAPE
        MOV #-1,PATRN ;SELECT AUTO RANDOM DATA
        JSR PC,STAUTO
AMOD2X: RTS PC ;RETURN TO SEQ

```

4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696 022772 012704 025160
4697 022776 104000
4698 023000 013703 000550
4699 023004 104002
4700 023006 012704 025144
4701 023012 104000
4702 023014 013703 000552
4703 023020 042703 177770
4704 023024 104002
4705 023026 012704 026371
4706 023032 104000
4707 023034 013703 000552
4708 023040 000303
4709 023042 042703 177770
4710 023046 104002
4711 023050 012704 026375
4712 023054 104000
4713 023056 005003
4714 023060 032737 000010 000552
4715 023066 001402
4716 023070 012703 000001
4717 023074 104002
4718 023076 012704 026401
4719 023102 104000
4720 023104 013703 000552
4721 023110 000241
4722 023112 006003
4723 023114 006003
4724 023116 006003
4725 023120 006003
4726 023122 042703 177760
4727 023126 104002
4728 023130 012704 025121
4729 023134 104000
4730 023136 032777 000400 155442
4731 023144 001406
4732 023146 012737 000122 000636
4733 023154 004737 023724
4734 023160 000411
4735 023162 065737 000734

```

;*****
;ERROR HEADER PRINT SUBROUTINE:
;
;THIS ROUTINE IS USED TO PRINT OUT A HEADER
;WITH EACH ERROR MESSAGE. THE PRINT IS IN TWO
;LINES AND CONTAINS THE FOLLOWING INFORMATION.
;LINE 1: DRIVE NO. SLAVE NO. DENSITY PARITY FORMAT
;LINE 2: CURRENT BLOCK NUMBER, RECORD NUMBER IN
;WHICH THE ERROR OCCURED PLUS THE TOTAL NUMBER
;OF RECORDS IN THIS BLOCK, THE RECORD SIZE (NUMBER
;OF CHARACTERS), AND THE ERROR TYPE (READ,WRITE, SPACE, ETC)
;PLUS THE TAPE DIRECTION (FORWARD OR REVERSE).
;ALL NUMBERS ARE IN OCTAL.
;*****
PAPRT:  MOV    #MSG12,R4
        TTOUTT                ;PRINT DRIVE HEADER
        MOV    DVN,R3
        OCTPP                ;PRINT DRIVE NUMBER
        MOV    #MSG11,R4
        TTOUTT                ;PRINT UNIT HEADER
        MOV    UDES,R3
        BIC    #177770,R3
        OCTPP                ;PRINT UNIT NUMBER
        MOV    #MSG60,R4
        TTOUTT                ;PRINT DENSITY TAG
        MOV    UDES,R3
        SWAB   R3
        BIC    #177770,R3
        OCTPP                ;PRINT DENSITY
        MOV    #MSG61,R4
        TTOUTT                ;PRINT PARITY TAG
        CLR    R3
        BIT    #10,UDES
        BEQ    PAPRT0
        MOV    #1,R3
        PAPRT0: OCTPP                ;PRINT PARITY
        MOV    #MSG62,R4
        TTOUTT                ;PRINT FORMAT TAG
        MOV    UDES,R3
        CLC
        ROR   R3
        ROR   R3
        ROR   R3
        ROR   R3
        BIC    #177760,R3
        OCTPP                ;PRINT FORMAT
        MOV    #MSG8,R4
        TTOUTT                ;PRINT PATRN TAG
        BIT    #400,ASWR
        BEQ    PAPRTB
        MOV    #122,TOB
        JSR   PC,TOB
        BR    PAPRTU
        PAPRTB: TST    ASEQF                ;SEE IF AUTO SEQ

```


4775
4776
4777
4778
4779
4780
4781
4782
4783
4784 023356 063737 000626 000624 RANG:
4785 023364 063737 000624 000626
4786 023372 023701 000626
4787 023376 101367
4788 023400 020237 000626
4789 023404 101364
4790 023406 000207
4791

```

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

```

```

4792 ;*****
4793 ;TTY ENTRY SUBROUTINE;
4794 ;
4795 ;THIS SUBROUTINE IS USED BY THE TEST CONDITION
4796 ;ENTRY ROUTINE TO READ THE RESPONSE ENTERED
4797 ;AT THE TTY AND CHECK THEM FOR LEGALITY AND
4798 ;LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL
4799 ;(0-7) AND MUST FALL WITHIN THE LIMITS SET BY
4800 ;THE CALLING ROUTINE.
4801 ;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
4802 ;A QUESTION MARK IS TYPED (?) AND THE RESPONSE
4803 ;MAY BE REENTERED.
4804 ;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
4805 ;MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
4806 ;CARRIAGE RETURN
4807 ;*****
4808
4809 023410 005037 000642 TTR: CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
4810 023414 005000 CLR RO
4811 023416 104010 TTR0: TTINN ;GO READ CHARACTER
4812 023420 122737 000015 000640 CMPB #15,TIB ;++G SEE IF CR
4813 023426 001004 BNE TTR1 ;IF NOT: BR
4814 023430 005737 000642 TST TEMP1 ;SEE IF FIRST CHARACTER
4815 023434 001436 BEQ TTR5 ;IF SO: BR
4816 023436 000426 BR TTR2 ;++G ELSE GO LOAD VALUE
4817 023440 122737 000060 000640 TTR1: CMPB #60,TIB ;++G SEE IF CHAR IS LESS THAN 0
4818 023446 101401 BLOS TTR1A ;IF NOT: BR
4819 023450 000431 BR T1NER ;++G ELSE GO TO ERROR
4820 023452 122737 000070 000640 TTR1A: CMPB #70,TIB ;++G SEE IF CHAR IS GREATER THAN 7
4821 023460 101001 BHI TTR1B ;IF NOT: BR
4822 023462 000424 BR T1NER ;++G ELSE GO TO ERROR
4823 023464 005237 000642 TTR1B: INC TEMP1 ;SET FIRST CHARACTER FLAG
4824 023470 006300 ASL RO
4825 023472 006300 ASL RO ;SHIFT 3 LEFT
4826 023474 006300 ASL RO
4827 023476 042737 177770 000640 BIC #177770,TIB ;STRIP ASCII
4828 023504 053700 000640 BIS TIB,RO ;LOAD CHARACTER
4829 023510 005301 DEC R1 ;SEE IF DONE
4830 023512 001341 BNE TTR0 ;IF NOT: BR
4831 023514 020002 TTR2: CMP RO,R2 ;SEE IF EXCEEDED MAXIMUM LIMIT
4832 023516 101401 BLOS TTR3 ;IF NOT: BR
4833 023520 000405 BR T1NER ;++G ELSE GO TO ERROR
4834 023522 020300 TTR3: CMP R3,RO ;SEE IF BELOW MINIMUM LIMIT
4835 023524 101401 BLOS TTR4 ;IF NOT: BR
4836 023526 000402 BR T1NER ;++G ELSE GO TO ERROR
4837 023530 010015 TTR4: MOV RO,(R5) ;LOAD VALUE
4838 023532 000207 TTR5: RTS PC ;EXIT
4839 023534 012704 026103 T1NER: MOV #MSG43,R4
4840 023540 104000 TTOUTT ;PRINT?
4841 023542 162716 000020 SUB #20,(SP) ;RESET SP TO START OF VALUE ROUTINE
4842 023546 000207 RTS PC ;REDO VALUE ENTRY

```

```

4843
4844 ;TTY READ SUBROUTINE*****
4845
4846 023550 005277 155034 TTIN: INC @TKS
4847 023554 105777 155030 TTIN1: TSTB @TKS
4848 023560 100375 BPL TTIN1
4849 023562 017737 155024 000640 MOV @TKB,TIB
4850 023570 042737 000200 000640 BIC @200,TIB ;+G STRIP PARITY BIT
4851 023576 105777 155012 TTIN2: TSTB @TPS
4852 023602 100375 BPL TTIN2
4853 023604 113777 000640 155004 MOVB TIB,@TPB
4854 023612 000207 RTS PC
4855
4856 ;TTY OUTPUT SUBROUTINE*****
4857
4858 023614 112437 000636 TTOUT: MOVB (R4),TOB
4859 023620 122737 000043 000636 CMPB @43,TOB
4860 023626 001476 BEQ TEX
4861 023630 122737 000045 000636 CMPB @45,TOB
4862 023636 001407 BEQ TCRLF
4863 023640 122737 000041 000636 CMPB @41,TOB
4864 023646 001467 BEQ TBELL ;DO BELL
4865 023650 004737 023724 JSR PC,TOG
4866 023654 000757 BR TTOUT
4867 023656 112737 000015 000636 TCRLF: MOVB @15,TOB
4868 023664 004737 023724 JSR PC,TOG
4869 023670 012703 000006 MOV @6,R3
4870 023674 005037 000636 TCRLFA: CLR TOB
4871 023700 004737 023724 JSR PC,TOG
4872 023704 005303 DEC R3
4873 023706 001372 BNE TCRLFA ;DO FILLERS
4874 023710 112737 000012 000636 MOVB @12,TOB
4875 023716 004737 023724 JSR PC,TOG
4876 023722 000734 BR TTOUT
4877
4878 023724 105777 154660 TOG: TSTB @TKS ;+JH SEE IF INPUT AT KEYBOARD
4879 023730 100024 BPL 3$ ;+JH BRANCH IF NOT
4880 023732 117737 154654 000747 MOVB @TKB,@CTRLS+1 ;+JH MOVE CHARACTER INTO BUFFER
4881 023740 142737 000200 000747 BICB @200,@CTRLS+1 ;+JH AND CLEAR PARITY BIT.
4882 023746 122737 000023 000747 CMPB @23,@CTRLS+1 ;+JH SEE IF CHARACTER IS XOFF (+S)
4883 023754 001004 BNE 2$ ;+JH BRANCH IF NOT
4884 023756 112737 000377 000746 MOVB @377,@CTRLS ;+JH ELSE SET XOFF FLAG
4885 023764 000757 BR TOG ;+JH AND KEEP CHECKING FOR XON
4886 023766 122737 000021 000747 2$: CMPB @21,@CTRLS+1 ;+JH SEE IF CHARACTER IS XON (+Q)
4887 023774 001002 BNE 3$ ;+JH BRANCH IF NOT
4888 023776 105037 000746 CLRB @CTRLS ;+JH ELSE CLEAR XOFF FLAG
4889 024002 105737 000746 3$: TSTB @CTRLS ;+JH SEE IF WE'RE IN XON MODE
4890 024006 100746 BMI TOG ;+JH BRANCH IF NOT
4891 024010 105777 154600 TSTB @TPS
4892 024014 100343 BPL TOG
4893 024016 113777 000636 154572 MOVB TOB,@TPB
4894 024024 000207 RTS PC
4895 024026 012703 000002 TBELL: MOV @2,R3
4896 024032 012737 000007 000636 TBELA: MOV @7,TOB
4897 024040 004737 023724 JSR PC,TOG
4898 024044 005303 DEC R3

```

4899 024046 001371
4900 024050 000661
4901
4902

BNE TBELA
BR TTOUT

;OCTAL OUTPUT SUBROUTINE*****

```

4903
4904
4905 024052 005037 024304      OCTP:  CLR      OFL          ;CLEAR FLAG FOR LEADING ZERO
4906 024056 000403              BR      OCTPE1
4907 024060 012737 000001 024304 OCTPE:  MOV      #1,OFL
4908 024066 010304              OCTPE1: MOV      R3,R4          ;SEE IF NUMBER IS ZERO
4909 024070 001006              BNE     OCTPO          ;IF NOT ZERO: BR
4910 024072 005737 024304      TST     OFL
4911 024076 001003              BNE     OCTPO
4912 024100 004737 024264      JSR     PC,OCTPG1      ;ELSE PRINT ZERO
4913 024104 000450              BR      OCTP3          ;...G SPACE AND EXIT
4914 024106 032704 100000      OCTPO:  BIT      #100000,R4 ;SEE IF MSD = 1
4915 024112 001406              BEQ     OCTP1          ;IF NOT: BR
4916 024114 012704 000001      MOV     #1,R4
4917 024120 004737 024242      JSR     PC,OCTPG      ;PRINT 1
4918 024124 000137 024136      JMP     OCTP2
4919 024130 005004              OCTP1:  CLR      R4
4920 024132 004737 024242      JSR     PC,OCTPG      ;PRINT 0
4921 024136 010304              OCTP2:  MOV      R3,R4
4922 024140 006004              ROR     R4
4923 024142 006004              ROR     R4
4924 024144 006004              ROR     R4          ;POSITION DIGIT
4925 024146 006004              ROR     R4
4926 024150 000304              SWAB   R4
4927 024152 004737 024242      JSR     PC,OCTPG      ;PRINT ^IGIT 2
4928 024156 010304              MOV     R3,R4
4929 024160 006004              ROR     R4
4930 024162 000304              SWAB   R4
4931 024164 004737 024242      JSR     PC,OCTPG      ;PRINT DIGIT 3
4932 024170 010304              MOV     R3,R4
4933 024172 006104              ROL     R4
4934 024174 006104              ROL     R4
4935 024176 000304              SWAB   R4
4936 024200 004737 024242      JSR     PC,OCTPG      ;PRINT DIGIT 4
4937 024204 010304              MOV     R3,R4
4938 024206 006004              ROR     R4
4939 024210 006004              ROR     R4
4940 024212 006004              ROR     R4
4941 024214 004737 024242      JSR     PC,OCTPG
4942 024220 010304              MOV     R3,R4
4943 024222 004737 024242      JSR     PC,OCTPG      ;PRINT DIGIT 5
4944 024226 012737 000240 000636 OCTP3:  MOV      #240,TOB
4945 024234 004737 023724      JSR     PC,TOG
4946 024240 000207              RTS     PC          ;EXIT
4947 024242 042704 177770      OCTPG:  BIC      #177770,R4
4948 024246 001004              BNE     OCTPGO
4949 024250 005737 024304      TST     OFL
4950 024254 001001              BNE     OCTPGO
4951 024256 000207              RTS     PC
4952 024260 005237 024304      OCTPGO: INC     OFL
4953 024264 052704 000260      OCTPG1: BIS      #260,R4
4954 024270 010437 000636      MOV     R4,TOB
4955 024274 004737 023724      JSR     PC,TOG
4956 024300 010304              MOV     R3,R4
4957 024302 000207              RTS     PC
4958 024304 000000      OFL:   0          ;FIRST CHAR FLAG

```



```

5015
5016 024522 000000
5017 024524 000000
5018 024526 000000
5019 024530 022737 000176 000606 CKSWR:
5020 024536 001123
5021 024540 105777 154044
5022 024544 100120
5023 024546 017737 154040 000640
5024 024554 042737 177600 000640
5025 024562 022737 000007 000640
5026 024570 001106
5027 024572 012704 027556 CNTG:
5028 024576 104000
5029 024600 012704 027562 CNTLU:
5030 024604 104000
5031 024606 017703 153774
5032 024612 004737 024060
5033 024616 012704 027571
5034 024622 104000
5035 024624 005037 024522
5036 024630 012737 000007 024524
5037 024636 104010
5038 024640 122737 000025 000640
5039 024646 001001
5040 024650 000753
5041 024652 122737 000015 000640
5042 024660 001013
5043 024662 012737 000200 024526
5044 024670 012704 027601
5045 024674 104000
5046 024676 022737 000007 024524
5047 024704 001035
5048 024706 000237
5049 024710 122737 000060 000640
5050 024716 003004
5051 024720 122737 000067 000640
5052 024726 002004
5053 024730 012704 026103
5054 024734 104000
5055 024736 000744
5056 024740 006337 024522
5057 024744 006337 024522
5058 024750 006337 024522
5059 024754 142737 000060 000640
5060 024762 153737 000640 024522
5061 024770 005337 024524
5062 024774 001755
5063 024776 000717
5064 025000 013777 024522 153600
5065 025006 000207
5066
5067
5068 025010 000000
5069 025012 104004
5070 025014 000207

```

```

;LOCATIONS USED:
TEMPST: .WORD 0
COUNT: .WORD 0
RDSW: .WORD 0
;SOFTWARE SWITCH REG PRESENT
;NO, GET OUT
;YES, WAIT FOR
;READY, GET CHARACTER
;AND STRIP OFF
;THE GARBAGE
;IS IT A <'G>
;GO READ A CHARACTER
;IS IT A 'U'?
;BRANCH IF NOT
;START OVER
;IS IT A <CR>?
;BRANCH IF NOT
;WAS IT FIRST CHARACTER
;CHANGE SWR IF NOT FIRST ONE
;GET OUT
;START OVER IF NOT LEGAL CHARACTER
;GET NITTY-GRITTY
;ONLY WANT 6 DIGITS
;CHANGE SWITCH REGISTER CONTENTS
;RETURN TO BODY OF PROGRAM
;HALT HANDLER*****
STOP: HALT
CKSWRR
RTS PC

```

```

;SWREG,SWR
OUT
;TKS
OUT
;TKB,TIB
;177600.TIB
;7,TIB
;CNTG,R4
;MSWR,R4
;SWR,R3
PC,OCPE
;MNEW,R4
CLR TEMPST
MOV #7,COUNT
TTINN
CMPB #25,TIB
BNE 2#
BR CNTLU
CMPB #15,TIB
BNE 4#
MOV #200,RDSW
MOV #MCRLF,R4
TTOUTT
CMP #7,COUNT
BNE 7#
BR OUT
CMPB #60,TIB
BGT 5#
CMPB #67,TIB
BGE 6#
MOV #MSG43,R4
TTOUTT
BR 3#
ASL TEMPST
ASL TEMPST
ASL TEMPST
BICB #60,TIB
BISB TIB,TEMPST
DEC COUNT
BEQ 5#
BR 1#
MOV TEMPST,@SWR
RTS PC
;CHECK FOR CONTROL G

```

```

5071
5072
5073
5074
5075 025016 016677 000002 153560 TRAP30: MOV 2(6),BPSW ;ADJUST PSW
5076 025024 011666 000002 MOV BSP,2(6) ;PLACE RETURN ADDRESS OVER PSW
5077 025030 162716 000002 SUB #2,BSP ;SUB. 2 FROM RETURN ADDRESS
5078 025034 013646 MOV B(6)+,-(6)
5079 025036 062716 121044 ADD #TABLE-104000,BSP ;GET SUBROUTINE STARTING ADDRESS
5080 025042 013607 MOV B(SP)+,PC ;GO TO SUBROUTINE
5081 025044 023614 TABLE: TTOUT
5082 025046 024052 OCTP
5083 025050 024530 CKSWR
5084 025052 025010 STOP
5085 025054 023550 TTIN
5086 104000 TTOUTT= 104000
5087 104002 OCTPP= 104002
5088 104004 CKSWRR= 104004
5089 104006 STOPP= 104006
5090 104010 TTINN= 104010
5091

```



```

5092
5093                ,ERROR MESSAGES*****
5094
5095 025056 042052 020105    043 MSG1: .ASCII /*DE */
5096
5097 025063    045 035507 021440 MSG2: .ASCII /*G; */
5098
5099 025070 041045 020073    043 MSG3: .ASCII /*B; */
5100
5101 025075    045 047103 021440 MSG4: .ASCII /*CN */
5102
5103 025102 053452 020105    043 MSG5: .ASCII /*WE */
5104
5105 025107    052 042522 021440 MSG6: .ASCII /*RE */
5106
5107 025114 051052 020123    043 MSG7: .ASCII /*RS */
5108
5109 025121    052 040520 051124 MSG8: .ASCII /*PATRN */
5110 025126 020116    043
5111 025131    040 047123 020072 MSG9: .ASCII / SN: */
5112 025136    043
5113 025137    052 042523 021440 MSG10: .ASCII /*SE */
5114
5115 025144 051452 040514 042526 MSG11: .ASCII /*SLAVE NO. */
5116 025152 047040 027117 021440
5117
5118 025160 042045 044522 042526 MSG12: .ASCII /*DRIVE NO. */
5119 025166 047040 027117 021440
5120
5121 025174 025045 047102 021440 MSG13: .ASCII /**BN */
5122
5123 025202 051052 020116    043 MSG14: .ASCII /*RN */
5124
5125 025207    045 020041 020040 MSG15: .ASCII /*!          BAD RECORD*/
5126 025214 020040 020040 020040
5127 025222 041040 042101 051040
5128 025230 041505 051117 022504
5129 025236 021445
5130
5131 025240 043040    043    MSG16: .ASCII / F*/
5132
5133 025243    040 021522    MSG17: .ASCII / R*/
5134
5135 025246 020041 047505 020124 MSG20: .ASCII /*! EOT NO: */
5136 025254 047516 020072    043
5137
5138
5139 025261    045 047111 042524 MSG21: .ASCII /*INTERCHANGE READ = */
5140 025266 041522 040510 043516
5141 025274 020105 042522 042101
5142 025302 036440 021440
5143
5144 025306 020445 046111 042514 MSG22: .ASCII /*!ILLEGAL BOT: HALT*/
5145 025314 040507 020114 047502
5146 025322 035124 044040 046101
5147 025330 022524    043

```

5148										
5149	025333	045	051503	020061	MSG23:	.ASCII	/#CS1 #/			
5150	025340	043								
5151										
5152	025341	045	041527	021440	MSG23A:	.ASCII	/#WC #/			
5153										
5154	025346	041045	020101	043	MSG23B:	.ASCII	/#BA #/			
5155										
5156	025353	045	041506	021440	MSG23C:	.ASCII	/#FC #/			
5157										
5158	025360	041445	031123	021440	MSG23D:	.ASCII	/#CS2 #/			
5159										
5160	025366	042045	020123	043	MSG23E:	.ASCII	/#DS #/			
5161										
5162	025373	045	051105	021440	MSG23F:	.ASCII	/#ER #/			
5163										
5164	025400	040445	020123	043	MSG23G:	.ASCII	/#AS #/			
5165										
5166	025405	045	045503	021440	MSG23H:	.ASCII	/#CK #/			
5167										
5168	025412	042045	020102	043	MSG23I:	.ASCII	/#DB #/			
5169										
5170	025417	045	051115	021440	MSG23J:	.ASCII	/#MR #/			
5171										
5172	025424	042045	020124	043	MSG23K:	.ASCII	/#DT #/			
5173										
5174	025431	045	041524	021440	MSG23L:	.ASCII	/#TC #/			
5175										
5176	025436	051445	020116	043	MSG23M:	.ASCII	/#SN #/			
5177										
5178	025443	045	047041	020117	MSG24:	.ASCII	/#!NO INTERRUPT#0/			
5179	025450	047111	042524	051122						
5180	025456	050125	022524	043						
5181										
5182	025463	045	047041	020117	MSG25:	.ASCII	/#!NO MOL: HALT#0/			
5183	025470	047515	035114	044040						
5184	025476	046101	022524	043						
5185										
5186	025503	045	051104	050117	MSG26:	.ASCII	/#DROPS: #/			
5187	025510	035123	021440							
5188										
5189	025514	050045	041511	051513	MSG27:	.ASCII	/#PICKS: #/			
5190	025522	020072	043							
5191										
5192	025525	045	043		MSG28:	.ASCII	/#0/			
5193	025527	045	052045	030115	MSG30:	.ASCII	'#TM02-TU16/TE16 AUTO SEQUENCE (CZTUJAJ0)#0' ;*G			
5194	025534	026462	052524	033061						
5195	025542	052057	030505	020066						
5196	025550	052501	047524	051440						
5197	025556	050505	042525	041516						
5198	025564	020105	041450	052132						
5199	025572	040525	030112	022451						
5200	025600	043								
5201	025601	045	041445	052132	MSG31:	.ASCII	'#CZTUJAJ0 TM02-TU16/TE16 RELIAB#0'			
5202	025606	040525	030112	052040						
5203	025614	030115	026462	052524						

5260	026162	020445	047516	041040	MSG48:	.ASCII	/#NO BOT ON REWIND: HALT#0/
5261	026170	052117	047440	020116			
5262	026176	042522	044527	042116			
5263	026204	020072	040510	052114			
5264	026212	021445					
5265							
5266	026214	047040	052117	040440	MSG49:	.ASCII	/ NOT AVAIL #/
5267	026222	040526	046111	021440			
5268	026230	044440	046114	043505	MSG50:	.ASCII	/ ILLEGAL DRIVE TYPE #/
5269	026236	046101	042040	044522			
5270	026244	042526	052040	050131			
5271	026252	020105	043				
5272	026255	045	042045	044522	MSG52:	.ASCII	/#DRIVE NUMBER = #/
5273	026262	042526	047040	046525			
5274	026270	042502	020122	020075			
5275	026276	043					
5276							
5277	026277	045	047506	046522	MSG53:	.ASCII	/#FORMAT = #/
5278	026304	052101	036440	021440			
5279							
5280	026312	053452	020105	046524	MSG54:	.ASCII	/#WE TM#/
5281	026320	043					
5282							
5283	026321	052	042523	052040	MSG55:	.ASCII	/#SE TM#/
5284	026326	021515					
5285							
5286	026330	052040	021515		MSG56:	.ASCII	/ TM#/
5287							
5288	026334	047040	047117	042455	MSG57:	.ASCII	/ NON-EXIST SLAVE#/
5289	026342	044530	052123	051440			
5290	026350	040514	042526	043			
5291	026355	045	051103	020103	MSG58:	.ASCII	/#CRC #/
5292	026362	043					
5293	026363	045	051114	020103	MSG59:	.ASCII	/#LRC #/
5294	026370	043					
5295	026371	052	020104	043	MSG60:	.ASCII	/#D #/
5296	026375	052	020120	043	MSG61:	.ASCII	/#P #/
5297	026401	052	020106	043	MSG62:	.ASCII	/#F #/
5298							
5299	026405	045	047452	044522	MSG64:	.ASCII	/#ORIGINAL ERROR#0/
5300	026412	044507	040516	020114			
5301	026420	051105	047522	025122			
5302	026426	043					
5303							
5304	026427	045	042522	051124	MSG65:	.ASCII	/#RETRY: #/
5305	026434	035131	021440				
5306							
5307	026440	020452	042523	051040	MSG66:	.ASCII	/#!SE RTRY #/
5308	026446	051124	020131	043			
5309							
5310	026453	052	042441	040522	MSG67:	.ASCII	/#!ERASE#/
5311	026460	042523	043				
5312							
5313	026463	045	042522	042522	MSG68:	.ASCII	/#REREV: #/
5314	026470	035126	021440				
5315	026474	052045	050101	020105	MSG69:	.ASCII	/#TAPE MARK = #/

5316	026502	040515	045522	036440			
5317	026510	021440					
5318							
5319	026512	020445	047516	042040	MSG/0:	.ASCII	/#!NO DRY FROM REWIND: HALT#/
5320	026520	054522	043040	047522			
5321	026526	020115	042522	044527			
5322	026534	042116	020072	040510			
5323	026542	052114	021445				
5324	026546	047040	047117	042455	MSG71:	.ASCII	/NON-EXIST DRIVE#/
5325	026554	044530	052123	042040			
5326	026562	044522	042526	043			
5327	026567	045	042522	053506	MSG72:	.ASCII	/#REFWD: #/
5328	026574	035104	021440				
5329	026600	053445	042524	051122	MSG73:	.ASCII	/#WTERR: #/
5330	026606	020072	043				
5331	026611	045	042522	044507	MSG74:	.ASCII	/#REGISTER START = #/
5332	026616	052123	051105	051440			
5333	026624	040524	052122	036440			
5334	026632	021440					
5335	026634	053045	041505	047524	MSG75:	.ASCII	/#VECTOR = #/
5336	026642	020122	020075	043			
5337	026647	045	042504	042522	MSG76:	.ASCII	/#DEREV: #/
5338	026654	035126	021440				
5339	026660	042045	043105	042127	MSG77:	.ASCII	/#DEFWD: #/
5340	026666	020072	043				
5341	026671	045	047041	047117	MSG78:	.ASCII	/#!NON-RETRYABLE WRITE ERROR: ER #/
5342	026676	051055	052105	054522			
5343	026704	041101	042514	053440			
5344	026712	044522	042524	042440			
5345	026720	051122	051117	020072			
5346	026726	051105	021440				
5347	026732	020445	047516	026516	MSG79:	.ASCII	/#!NON-RETRYABLE READ ERROR: ER #/
5348	026740	042522	051124	040531			
5349	026746	046102	020105	042522			
5350	026754	042101	042440	051122			
5351	026762	051117	020072	051105			
5352	026770	021440					
5353	026772	020445	042441	042116	MSG100:	.ASCII	/#!END OF PASS #/
5354	027000	047440	020106	040520			
5355	027006	051523	022440	043			
5356	027013	045	025052	025052	MSG101:	.ASCII	/#*****#/
5357	027020	025052	025052	022452			
5358	027026	043					
5359	027027	052	046524	031060	MSG102:	.ASCII	/#TM02 #/
5360	027034	021440					
5361	027036	051452	040514	042526	MSG103:	.ASCII	/#SLAVES #/
5362	027044	020123	043				
5363	027047	045	052501	047524	MSG104:	.ASCII	/#AUTO CONT: #/
5364	027054	041440	047117	035124			
5365	027062	021440					
5366	027064	051045	041505	053117	MSG105:	.ASCII	/#RECOVERED#/
5367	027072	051105	042105	043			
5368	027077	052	020441	040502	MSG106:	.ASCII	/#!BAD TAPE OVERFLOW#/
5369	027104	020104	040524	042520			
5370	027112	047440	042526	043122			
5371	027120	047514	021527				

5372	027124	051045	053505	047111	MSG16A: .ASCII	/#RLWIND TAPE; RESTART AT BLOCK ONE //
5373	027132	020104	040524	042520		
5374	027140	020073	042522	052123		
5375	027146	051101	020124	052101		
5376	027154	041040	047514	045503		
5377	027162	047440	042516	043		
5378	027167	045	020441	047125	MSG107: .ASCII	/#!!UNRECOVERABLE BAD SPOT/
5379	027174	042522	047503	042526		
5380	027202	040522	046102	020105		
5381	027210	040502	020104	050123		
5382	027216	052117				
5383	027220	041045	042101	051040	.ASCII	/#BAD RECORD LEFT ON TAPE#/
5384	027226	041505	051117	020104		
5385	027234	042514	052106	047440		
5386	027242	020116	040524	042520		
5387	027250	021445				
5388	027252	047045	055122	047440	MSG108: .ASCII	/#NRZ ONLY: #/
5389	027260	046116	035131	021440		
5390	027266	020452	050041	051517	MSG109: .ASCII	/#!!POSITION LOST IN RETRY#/
5391	027274	052111	047511	020116		
5392	027302	047514	052123	044440		
5393	027310	020116	042522	051124		
5394	027316	021531				
5395	027320	051445	051525	042520	MSG110: .ASCII	/#SUSPECT BAD TAPE#/
5396	027326	052103	041040	042101		
5397	027334	052040	050101	021505		
5398	027342	051045	050105	040505	MSG111: .ASCII	/#REPEAT: #/
5399	027350	035124	021440			
5400	027354	041040	042101	052040	MSG112: .ASCII	/ BAD TAPE SPOTS#/
5401	027362	050101	020105	050123		
5402	027370	052117	022523	043		
5403						
5404	027375	045	051440	043117	MSG113: .ASCII	/# SOFT: #/
5405	027402	035124	021440			
5406						
5407	027406	020045	040510	042122	MSG114: .ASCII	/# HARD: #/
5408	027414	020072	043			
5409						
5410	027417	045	020441	040510	MSG115: .ASCII	/#!!HARD READ ERROR#/
5411	027424	042122	051040	040505		
5412	027432	020104	051105	047522		
5413	027440	021522				
5414	027442	020445	047125	052111	MSG116: .ASCII	/#!UNIT REWINDING: TEST WILL START AT BOT#/
5415	027450	051040	053505	047111		
5416	027456	044504	043516	020072		
5417	027464	042524	052123	053440		
5418	027472	046111	020114	052123		
5419	027500	051101	020124	052101		
5420	027506	041040	052117	043		
5421	027513	045	042522	047515	MSG120: .ASCII	/#REMOVE TMDP FROM UNIT UNDER TEST#/
5422	027520	042526	052040	042115		
5423	027526	020120	051106	046517		
5424	027534	052440	044516	020124		
5425	027542	047125	042504	020122		
5426	027550	042524	052123	021445		
5427	027556	057045	021507		!CNTG: .ASCII	/#+G#/

5428	027562	051445	051127	020075	#MSWR: .ASCII /#SWR= 0/	
5429	027570	043				
5430	027571	040	047040	053505	#MNEW: .ASCII / NEW= 0/	
5431	027576	020075	043			
5432	027601	045	043		MCRLF: .ASCII /#0/	
5433						
5434		027604			.EVEN	
5435	027604	000000			WDATA: 0	;WRITE BUFFER
5436						
5437		033612			.*.4004	
5438	033612	000000			RDATA: 0	;READ BUFFER
5439						
5440		000001			.END	

DATA4	003004	1838#							
DATA5	003006	1839#							
DATA6	003010	1840#							
DATA7	003012	1841#							
DATBL	002772	1833#	3457						
DATER1	001132	1689#	2044	3839*					
DATR	015104	1975	3441	3637#					
DATRO	015122	3641#	3644						
DATO	014414	1834	3479#						
DATOA	014446	3486#	3494	3498	3501				
DATOB	014454	3487#	3488						
DATOC	014520	3492	3499#						
DATOD	014526	3502#	3510						
DATOE	014536	3504#	3509						
DATOF	01.550	3506	3508#						
DAT1	014560	1835	3515#						
DAT1A	014564	3516#	3525	3548	3553	3601			
DAT1B	014570	3517#	3519						
DAT10	014702	1842	3568#						
DAT10A	014714	3571#	3575						
DAT11	014732	1843	3580#						
DAT11A	014740	3582#	3585						
DAT12	014752	1844	3590#						
DAT12A	014762	3592#	3595						
DAT13	014774	1845	3600#						
DAT14	015004	1846	3605#						
DAT14A	015016	3608#	3612						
DAT15	015034	1847	3617#						
DAT15A	015040	3618#	3624						
DAT15B	015050	3620#	3622						
DAT2	014600	1836	3524#						
DAT3	014604	1837	3529#						
DAT3A	014612	3531#	3542						
DAT3B	014616	3532#	3535						
DAT4	014630	1838	3540#						
DAT5	014640	1839	3547#						
DAT6	014646	1840	3552#						
DAT7	014654	1841	3557#						
DAT7A	014670	3560#	3563						
DB	000532	1534#							
DCHK	015546	2783	2978	3754#					
DCKO	015576	3758	3761#						
DEREV1	001172	1711#	2064	3841*					
DEREX	016624	3912	3933	3935	3943	3951	3954	3956#	
DEREX1	016656	3957	3960	3962	3964#				
DERFL	000704	1593#	3755*	3832	3965*				
DERR	016162	3825	3872#						
DERR0	016172	3874#	3963						
DERR0A	016222	3876	3881#						
DERR0B	016254	3887	3890#						
DERR0C	016300	3894	3897#						
DERR0D	016302	3896	3898#						
DERR1	016330	3902	3905#						
DERR2	016332	3904	3906#						
DERR3	016346	3909#							
DERR4	016350	3873	3908	3910#					

DERR4A	016510	3927	3936#																
DERR4B	016552	3921	3944#																
DERR5	016606	3948	3952#																
DERR6	016620	3924	3946	3955#															
DFX	016160	3833	3835	3840	3842#														
DF0	016052	3781	3820#	3829															
DFOA	015746	3791	3793#	3830															
DFOA0	015770	3797	3799#																
DFOA1	016004	3802	3804#																
DFOA2	016020	3807	3809#																
DFOA3	016034	3812	3814#																
DFOA4	016040	3794	3816#																
DF0B	015706	3782#																	
DF0B0	015730	3785	3788#																
DF0C	015666	3774	3778#																
DF0C0	015676	3764	3766	3768	3780#														
DF0D	015652	3770	3775#																
DF0E	015644	3772#	3777																
DF0F	015636	3769#	3773																
DF1	016064	3817	3821	3824#															
DF2	016074	3819	3823	3826#															
DF3	016112	3827	3831#																
DF4	016154	3838	3841#																
DOUT	024306	3898	3906	4962#	4979	4981													
DOUTD	024374	4977#																	
DOUT1	024322	4965#	4966	4975															
DOUT2	024354	4969	4972#																
DOUT3	024362	4971	4973#																
DPC	017000	4004#	4050																
DPCG	017006	4005	4007#																
DPC0	017014	4008#	4042																
DPC0A	017056	4016	4018#																
DPC1	017064	4011	4020#																
DPC1A	017112	4024	4026#																
DPC2	017154	4009	4014	4022	4036#														
DPC2A	017116	4019	4027#																
DPC2B	017140	4031#	4034																
DPC3	017204	4039	4043#																
DPPRT	017252	2015	4018	4026	4052#														
DPPRTX	017406	4072	4075#																
DPPRT0	017322	4060#	4065																
DPPRT1	017346	4063	4066#																
DPPRT2	017362	4069#	4074																
DROP	016770	3999	4002#																
DRPK	016756	3995	3999#																
DRPKF	016670	3824	3986#																
DRP1	001012	1634#	3993	4028	4046	4055													
DRP2	001014	1635#																	
DRP3	001016	1636#																	
DRP4	001020	1637#																	
DRP5	001022	1638#																	
DRP6	001024	1639#																	
DRP7	001026	1640#																	
DRP8	001030	1641#																	
DRVER	021142	2369#	2739#	4185#	4237	4261	4312	4352#											
DS	000522	1530#	1984	1998	2074	2107	2136	2159	2161	2232	2250	2252	2312	2366					

CZTUARO TMO2-TU16/TE16 RELIAB
CZTUAR.P11 25-MAY-84 11:33MACY11 30(1046) 25-MAY-84 11:41 PAGE 127
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0126

MSG14	025202	2540	4745	51230
MSG15	025207	3915	51250	
MSG16	025240	4376	51310	
MSG16A	027124	2131	53720	
MSG17	025243	4373	51330	
MSG2	025063	3891	50970	
MSG20	025246	2125	51350	
MSG21	025261	3356	51390	
MSG22	025306	2728	51440	
MSG23	025333	4253	51490	
MSG23A	025341	4287	51520	
MSG23B	025346	4273	51540	
MSG23C	025353	4283	51560	
MSG23D	025360	4257	51580	
MSG23E	025366	4263	51600	
MSG23F	025373	4267	51620	
MSG23G	025400	51640		
MSG23H	025405	4320	51660	
MSG23I	025412	51680		
MSG23J	025417	51700		
MSG23K	025424	51720		
MSG23L	025431	51740		
MSG23M	025436	51760		
MSG24	025443	4477	51780	
MSG25	025463	4424	51820	
MSG26	025503	4052	51860	
MSG27	025514	4067	51890	
MSG28	025525	2549	2564	51920
MSG3	025070	3899	50990	
MSG30	025527	3179	51930	
MSG31	025601	3176	52010	
MSG32	025643	3233	52080	
MSG33	025664	3270	52120	
MSG34	025700	3282	52150	
MSG35	025713	3315	52180	
MSG36	025734	3325	52220	
MSG37	025760	3337	52270	
MSG38	026003	3365	52310	
MSG4	025075	3881	51010	
MSG40	026023	3376	52340	
MSG41	026051	3385	52390	
MSG42	026063	3394	52420	
MSG43	026103	4839	5053	52460
MSG44	026107	2984	52480	
MSG45	026136	52530		
MSG46	026150	52560		
MSG47	026155	52580		
MSG48	026162	2165	2259	52600
MSG49	026214	1989	2079	52660
MSG5	025102	2306	4418	51030
MSG50	026230	3262	52680	
MSG52	026255	3216	52720	
MSG53	026277	3294	52770	
MSG54	026312	2358	4247	52800
MSG55	026321	3037	52830	
MSG56	026330	4249	52860	

TEX	024024	4860	4894*											
TIB	000640	1575*	4812	4817	4820	4827*	4828	4849*	4850*	4853	5023*	5024*	5025	5038
		5041	5049	5051	5059*	5060								
TINER	023534	4819	4822	4833	4836	4839*								
TINF	000634	1573*	1892*	1897*	1901*	2175*	3166							
TINP	012402	1940	3166*											
TINPA	012412	3167	3169*											
TINPB	012432	3173*	3175											
TINPB0	012616	3207*	3210											
TINPB1	012456	3178	3180*											
TINPC	012644	3182	3212	3215*	3232									
TINPX	014144	3375	3403*											
TINPO	012750	3226	3233*	3245	3259	3310								
TINPOA	013024	3244	3246*											
TINPOB	013042	3242	3249*											
TINPOC	013102	3255*												
TINPOD	013124	3256	3260*											
TINPOE	013150	3261	3266*											
TINP1	013166	3270*												
TINP2	013244	3278	3282*											
TINP2A	013322	3290	3294*											
TINP2B	013400	3302	3306*											
TINP2C	013426	3248	3308	3311*										
TINP3	013446	3315*												
TINP4	013776	3374*	4509											
TKB	000612	1561*	2982	2996	3000*	4488	4849	4880	5023					
TKS	000610	1560*	1951*	2911*	2980	3005*	4846*	4847	4878	5021				
TMAX	000564	1550*	2355	2693	2805	3029	3349	3351	4633*					
TMFLG	000676	1590*	2357*	2401*	2417	2690*	2695*	2721	2734	2781	2793*	2807	2809*	2812*
		2923	2947	2974	4137	4159	4166	4177	4192	4245	4344	4753		
TOB	000636	1574*	1902	4277*	4298*	4308*	4732*	4762*	4858*	4859	4861	4863	4867*	4870*
		4874*	4893	4896*	4944*	4954*	4962*	4964*	4967	4968	4973*	5005*	5007*	
TGG	023724	4278	4299	4309	4733	4763	4865	4868	4871	4875	4878*	4885	4890	4892
		4897	4945	4955	5008									
TPB	000616	1563*	4853*	4893*	4970*	4972*								
TPOS	014154	3281	3293	3305	3409*	3412								
TPOS1	014166	3251	3413*											
TPS	000614	1562*	4851	4891	4965									
TRAP30	025016	1468	5075*											
TSTAL	000576	1555*	2004	2007	2493	2617	2642	3025	3056	3396	3398			
TTIN	023550	4846*	5085											
TTINN -	104010	4811	5037	5090*										
TTINT	021632	1484	4487*											
TTINTO	021734	4491	4508*											
TTIN1	023554	4847*	4848											
TTIN2	023576	4851*	4852											
TTOUT	023614	4858*	4866	4876	4900	5081								
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	4530	4537	4550	4552	4556	4697	4701	4706	4712
		4719	4729	4742	4746	4767	4840	5028	5030	5034	5045	5054	5086*	

\$CHAIN 13610 1870
.\$ACT1 13610 1472
.\$EOP 13610 2188

. ABS. 033614 000

ERRORS DETECTED: 0

CZTUJ,CZTUJ/CRF/SOL/NL:TOC=CZTUJ.SML/ML,CZTUJ.P11
RUN-TIME: 4 9 1 SECONDS
RUN-TIME RATIO: 20/16=1.2
CORE USED: 15K (30 PAGES)