

TM02, TE16,
TU16

RELIAB
CZTUAIO

AH-9448I-MC
FICHE 1 OF 1

JUL 1982
COPYRIGHT TO 74-82
MADE IN USA



A large grid of 10 columns and 20 rows of small, illegible text blocks, likely representing a data table or a series of microfilm frames. The text is too small to be read accurately.



.REM %

IDENTIFICATION

PRODUCT CODE: AC-94471-MC
PRODUCT TITLE: CZTUAIO TMO2-TU16/TE16 RELIAB
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: R.B. BARNES
PRODUCT DATE: 29-MAR-82 B.T. LEBLANC

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) 1974,1982 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 RM AND TMO2.

HOWEVER; THE RM 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 RM OR TMO2.

2. REQUIREMENTS (HARDWARE)

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

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING BINARY TAPES

4. STARTING PROCEDURE

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

- A. 200(8): THIS ADDRESS MUST BE USED ON INITIAL START FROM LOAD AS ALL PARAMETERS ARE ENTERED FROM HERE. REQUESTS ARE PRINTED ON THE TELETYPE FOR ENTRY OF RH STARTING ADDRESS, VECTOR ADDRESS, DRIVE NUMBER(TMO2 ADDRESS), SLAVE NUMBER, DENSITY, PARITY, FORMAT, RECORD COUNT, CHARACTER COUNT, PATTERN NUMBER, TAPE MARK AND STALL FOR READ, WRITE, AND TURNAROUND. ALL 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 PENTERED. 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 NECFSSARY 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 MRZ ONLY. SEE ALSO SECTION 11 FOR DETAILS.
- E. 300(8): THIS ADDRESS IS TO BE USED AS A RESTART ONLY AND WILL PERFORM JUST AS IN 200(8) EXCEPT THAT THE PARAMETER INPUT LIST IS SHORTENED. THE SHORT PARAMETER LIST CONSISTS OF DRIVE NUMBER, SLAVE NUMBER, DENSITY, PARITY, FORMAT, RECORD COUNT, CHARACTER COUNT, PATTERN, TAPE MARK, AND INTERCHANGE READ.

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

- REGISTER START: THE RESPONSE REQUIRED FOR THIS REQUEST IS TO ENTER THE ADDRESS OF THE FIRST RH REGISTER (CS1) AS A SIX DIGIT UNIBUS ADDRESS.
- VECTOR ADDRESS: THE RESPONSE FOR THIS REQUEST IS TO ENTER THE INTERRUPT VECTOR ADDRESS USED BY THE RH AS A THREE (3) DIGIT ADDRESS.
- DRIVE NUMBER: THE DRIVE NUMBER (MASSBUS ADDRESS OF THE TM02) IS ENTERED AS ONE (1) OCTAL CHARACTER AND MUST BE WITHIN THE LIMITS OF 0 THROUGH 7.
- SLAVE NUMBER: THE SLAVE NUMBER IS ENTERED AS ONE (1) OCTAL CHARACTER AND MUST BE WITHIN THE LIMITS OF 0 THROUGH 7. WHEN THE SLAVE NUMBER HAS BEEN ENTERED AND IS LEGAL, THE PROGRAM TESTS FOR THE PRESENCE OF A SLAVE OF THAT NUMBER. IF THE SLAVE IS AVAILABLE A PRINTOUT OF 7 CHANNEL, IF APPLICABLE, AND ITS SERIAL NUMBER (IN BCD) WILL BE MADE TO ASSIST THE OPERATOR IN SETTING OF DENSITY, PARITY, AND FORMAT. A CHECK IS MADE FOR THE PROPER SETTING OF THE DRIVE TYPE REGISTER; IF WRONG, A MESSAGE IS PRINTED FOR INFORMATION ONLY. IF THE SLAVE IS NOT AVAILABLE, A MESSAGE STATING SO WILL BE PRINTED AND A NEW SLAVE NUMBER REQUEST WILL BE ISSUED. WHEN A GOOD SLAVE NUMBER HAS BEEN ENTERED, REQUESTS FOR OPERATING DENSITY, PARITY AND FORMAT ARE MADE FOR THAT SLAVE AND SHOULD BE RESPONDED TO ACCORDING TO THAT PARTICULAR SLAVE'S NEEDS. AS MANY AS EIGHT (8) SLAVE NUMBER REQUESTS MAY BE USED, HOWEVER, AT LEAST ONE MUST BE USED. THE SLAVE NUMBERS AND THEIR RESPECTIVE DENSITY, PARITY AND FORMAT MAY BE ENTERED IN ANY ORDER. THE INFORMATION FOR EACH SLAVE ENTERED IS LOADED INTO A TABLE FOR REFERENCE IN TESTING. IF LESS THAN EIGHT (8) SLAVES ARE REQUIRED, THEN RESPONDING TO THE SLAVE NUMBER REQUEST WITH A CARRIAGE RETURN WILL TERMINATE THE SLAVE ENTRIES AND CONTINUE TO THE NEXT PARAMETER. IT SHOULD BE REMEMBERED THAT AT LEAST ONE SLAVE NUMBER REQUEST

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

DENSITY:

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

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

PARITY:

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

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

FORMAT:

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

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

RECORD COUNT:

THIS REQUEST IS RESPONDED TO BY A SIX (6) CHARACTER OCTAL NUMBER FROM 1 TO 177777. REMEMBER LEADING ZEROS ARE NOT REQUIRED AND IF LESS THAN SIX CHARACTERS ARE ENTERED, A CARRIAGE RETURN WILL TERMINATE THE RESPONSE. THE RECORD COUNT IS USED IN CONJUNCTION WITH THE CHARACTER COUNT TO ESTABLISH A BLOCKING FACTOR FOR USE IN READ OR WRITE CYCLES.

CHARACTER COUNT:

THIS RESPONSE IS ENTERED AS FOUR (4) OCTAL CHARACTERS WITHIN THE LIMITS OF 20 THRU 4000. AGAIN LEADING ZEROS ARE NOT REQUIRED AND A CARRIAGE RETURN TERMINATES A LESS THAN FOUR (4) CHARACTER RESPONSE. THE CHARACTER COUNT IN CONJUNCTION WITH THE RECORD COUNT IS USED TO ESTABLISH THE BLOCK SIZE (CHARACTERS PER RECORD, AND RECORDS PER BLOCK) USED IN READ AND WRITE CYCLES. THE SAME BLOCKING IS USED ON ALL AVAILABLE UNITS.

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

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

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

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

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

READ: THE TIME DELAY BETWEEN EACH RECORD READ

WRITE: THE TIME DELAY BETWEEN EACH RECORD WRITTEN

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

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

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

SAMPLE START AT 200(8):

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

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

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

TU16 TAPE DRIVE TEST
ENTER CONDITIONS IN OCTAL

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

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

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

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

4.1 AUTOMATIC MODE OPERATION

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

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

5. DATA PATTERNS

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

THE FOLLOWING IS A LIST OF THE DATA PATTERNS AVAILABLE:

DATA0: EXTERNAL INPUT THRU HIGH SPEED READER (SEE DTC)
DATA1: ALL ONE BITS IN ALL CHARACTERS
DATA2: ALL ZERO BITS IN ALL CHARACTERS
DATA3: A ONE BIT WALKING FROM RIGHT TO LEFT IN A FIELD OF ZEROS
DATA4: A ZERO BIT WALKING FROM RIGHT TO LEFT IN A FIELD OF ONES.
DATA5: ALTERNATING ONE AND ZERO BITS IN EACH CHARACTER
DATA6: ALTERNATING ZERO AND ONE BITS IN EACH CHARACTER
DATA7: SAME AS DATA5 BUT WITH EVERY OTHER CHARACTER COMPLEMENTED
DATA10: WALKING ONE/ALL ONE IN ALTERNATING CHARACTERS
DATA11: INCREMENTING CHARACTERS (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) TYPING 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
(00200)0=USE FIXED CHARACTER COUNT

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

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

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

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

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

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

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

SWITCH EXPLANATION AND EXAMPLES:

SW0-3:

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

EXAMPLES: 0-3

- A. SW0=0, SW1=0, SW2=1, SW3=1
WRITE ONLY X RECORDS OF Y CHARACTERS
- B. SW0=0, SW1=0, SW2=1, SW3=0
WRITE THEN BACKSPACE AND READ FORWARD X RECORDS
- C. SW0=0, SW1=0, SW2=0, SW3=1
WRITE THEN READ REVERSE X RECORDS.
- D. SW0=0, SW1=0, SW2=0, SW3=0
WRITE THEN READ REVERSE AND READ FORWARD X RECORDS
- E. SW0=0, SW1=1, SW2=0, SW3=0
WRITE THEN BACKSPACE AND READ FORWARD THEN REVERSE
- F. SW0=1, SW1=0, SW2=1, SW3=0
READ TAPE FORWARD X RECORDS
- G. SW0=1, SW1=0, SW2=0, SW3=1
READ TAPE REVERSE X RECORDS
- H. SW0=1, SW1=0, SW2=0, SW3=0
READ TAPE REVERSE THEN FORWARD
- I. SW0=1, SW1=1, SW2=0, SW3=0
READ TAPE FORWARD THEN REVERSE

SW4:

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

SW5:

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

SW6-8:

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

SW9:

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

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

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

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

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

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

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

9. ERROR PRINTOUTS

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

A. OPERATION ERRORS:

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

1. READ/WRITE STATUS ERRORS: THESE ARE DETECTED BY EITHER THE 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 .PE 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 CO'TINUE.
5. NO BOT ON REWIND: AS EACH SLAVE IS REWOUND A CHECK
IS MADE TO ASSURE THAT PROPER POSITION AT BOT
IS ESTABLISHED. IF BOT IS NOT SET UPON COMPLETION OF
A REWIND, AN ERROR IS PRINTED AND THE PROGRAM
WILL HALT. PRESS CONTINUE TO RESUME TESTING.
6. POSITION ERROR: IF POSITION IS LOST DURING A RETRY,
A MESSAGE IS PRINTED, THE TAPE REWOUND,
AND REMOVED FROM TESTING UNTIL ALL ARE
RESTARTED AT BLOCK ONE.
7. BAD TAPE OVERFLOW: IF 20(8) BAD TAPE SPOTS ARE FOUND,
A MESSAGE IS PRINTED, THE TAPE REWOUND,
AND REMOVED FROM TESTING UNTILL ARE
RESTARTED AT BLOCK ONE.
8. HARD READ ERROR: IF ANY HARD READ ERROR IS ENCOUNTERED
DURING A RETRY, A MESSAGE IS PRINTED
REGARDLESS OF THE SETTING OF SW10.
9. NON-RETRYABLE: IF ANY NON-RETRYABLE ERROR IS ENCOUNTERED, A
MESSAGE IS PRINTED REGARDLESS OF THE SETTING OF SW10.

D. EXAMPLES:

GLOSSARY:

BN = CURRENT BLOCK NUMBER
RN = CURRENT RECORD NUMBER
RS = RECORD SIZE, IN FRAMES
WE = WRITE STATUS ERROR
RE = READ STATUS ERROR
SE = SPACE ERROR
TM = TAPE MARK
F = FORWARD
R = REVERSE
CS1 = RH/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 14427C
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 1111111
B 1011111
CN 6
G 1111111
B 1011111

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

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

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

BAD RECORD

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

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

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

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

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

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

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

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

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

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

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

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

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

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

10. STATISTICS PRINTOUT

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

STATISTICS PRINT EXAMPLE (A HEADER WILL PRECEED THE STATS)

DROPS: 0 3 0 0 0 6 45 0
PICKS: 1 0 0 0 0 0 0 2
RETRY: 1
WTERR: 2
REFWD: 3
SOFT: 2
HARD: 1
DEFWD: 0
REREV: 4
SOFT: 1
HARD: 3
DEREV: 0
2 BAD TAPE SPOTS
0 *RN 1 *RN 2
UN 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 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 COMPATIBILITY. 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 C, 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.

THEY ARE, YOU'LL LIKE IT.

13. LISTING

x

CZTUAIO TMO2-TU16/TE16 RELIAB
CZTUAL.P11 16-JUN-82 10:46

MACY11 30(1046) 16-JUN-82 11:10 F 3
PAGE 31

SEQ 0031

1398
1399
1400
1401
1402
1403
1404

: 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


```
1451                                     ;REGISTER EQUIVS*****
1452                                     RO=R0
1453 000000                               R1=R1
1454 000001                               R2=R2
1455 000002                               R3=R3
1456 000003                               R4=R4
1457 000004                               R5=R5
1458 000005                               SP=R6
1459 000006                               PC=R7
1460 000007                               NOP=240
1461 000240
1462
1463                                     ;TRAP CATCHERS*****
1464                                     =30
1465 000030 024676                       TRAP30
1466 000032 000032                       =32
1467 000032 000340                       340
1468
1469
1470                                     ;ACT11 HOOK *****
1471 000034                               $SVPC=.
1472 000046                               ;SAVE CURRENT LOCATION CTR
1473 000046 005116                       .WORD SENDAD
1474 000052 000052                       ;SET LOCATION 46
1475 000052 000000                       .WORD 0
1476 000034                               ;SET LOCATION 52 = 0
1477                                     ;RESTORE LOCATION CTR
1478
1479                                     ;TTY INTERRUPT VECTOR*****
1480 000060 000060                       =60
1481 000060 021630                       TTINT
1482 000062 000000                       ;TTY INTERRUPT HANDLER ADDRESS
1483 0
1484
1485                                     ;SOFTWARE SWITCH REGISTER LOC. 176*****
1486
1487 000176 000176                       =176
1488 000176 000000                       SWREG: 0
1489                                     ;SOFTWARE SWITCH REGISTER
1490
1491                                     ;START ADDRESS*****
1492 000200 000200                       =200
1493 000200 000137 003026                 JHP START
1494                                     ;ENTER PARAMETERS VIA TTY
1495 000204 000204                       =204
1496 000204 000137 003152                 JHP STARTC
1497                                     ;USE FIXED PARAMETERS; HOLD DATA
1498 000210 000210                       =210
1499 000210 005037 015150                 CLR RDFL
1500 000214 000137 003160                 JHP STARTA
1501                                     ;USE FIXED PARAMETERS; NEW DATA
1502
1503                                     ;FAG TAPE INTERRUPT VECTOR*****
1504 000224 000224                       =224
1505 000224 021714                       HTINT
1506 000226 000340                       ;FAG TAPE INTERRUPT HANDLER ADDRESS
1507 340
```

:507
1508
1509
1510
1511 000240 000240
1512 000244 000137 000734 003136

:AUTO SEQUENCE START*****
: =240
INC ASEQF :SET AUTO SEQUENCE FLAG
JMP STAUT :GO TO START OF AUTO SEQUENCE

```
1513 ;SHORT CONVERSATION RESTART*****
1514
1515 ;=300
1516 000300 000300 014150 INC SCVFL ;SET SHORT CONVERSATION FLAG
1517 0003C4 000137 003026 JMP START ;ENTER SHORT PARAMETER LIST
1518
1519 ;=510
1520 ;TU16/TE16 REGISTER EQUIVS*****
1521
1522 000510 172440 C1: 172440
1523 000512 172442 WC: 172442
1524 000514 172444 BA: 172444
1525 000516 172446 FC: 172446
1526 000520 172450 CS: 172450
1527 000522 172452 DS: 172452
1528 000524 172454 ER: 172454
1529 000526 172456 AS: 172456
1530 000530 172460 CC: 172460
1531 000532 172462 DB: 172462
1532 000534 172464 MR: 172464
1533 000536 172466 DT: 172466
1534 000540 172470 SN: 172470
1535 000542 172472 C2: 172472
1536
1537 ;CONSTANTS*****
1538
1539 000544 172440 REGS: 172440 ;STARTING REGISTER ADDRESS (CS1)
1540 000546 000224 VECT: 224 ;VECTOR ADDRESS (RM INTERRUPT)
1541 000550 000000 DVN: 0 ;DRIVE NUMBER
1542 000552 000000 UDES: 0 ;UNIT DESCRIPTION (PARITY,DENSITY,UNIT,FORMAT)
1543 000554 000100 RCNT: 100 ;RECORD COUNTER
1544 000556 177600 FMCNT: 177600 ;NUMBER OF CHAR (4 - 4000) OCTAL IN TWOS COMPLEMENT
1545 000560 000001 PATRN: 1 ;DATA PATTERN SELECTOR (0 - 15) OCTAL
1546 000562 000002 RDCMD: 2 ;READ COMMAND
1547 000564 000000 TMEX: 0 ;TAPE MARK FLAG: 1=TM 0=NO TM
1548 000566 000000 INTRF: 0 ;INTERCHANGE READ 1=YES 0=NO
1549 000570 000000 SING: 0 ;SINGLE PASS 1=YES 0=NO
1550 000572 000001 RSTAL: 1 ;READ STALL
1551 000574 000001 WSTAL: 1 ;WRITE STALL
1552 000576 000001 TSTAL: 1 ;TURN AROUND STAL
1553 000600 002000 YSTAL: 2000 ;YOZZLE STAL
1554 000602 000010 RETRY: 10 ;READ RETRY NUMBER
1555 000604 177776 PSW: 177776 ;PROCESSOR STATUS
1556 000606 177570 SWR: 177570 ;CONSOLE SWITCHES
1557 000610 177560 TKS: 177560 ;TTY READ STATUS REGISTER
1558 000612 177562 TKB: 177562 ;TTY READ BUFFER
1559 000614 177564 TPS: 177564 ;TTY PUNCH STATUS REGISTER
1560 000616 177566 TPB: 177566 ;TTY PUNCH OUTPUT REGISTER
1561 000620 177550 PRS: 177550 ;H/S READER STATUS REGISTER
1562 000622 177552 PRB: 177552 ;H/S READER BUFFER
1563 000624 153624 RANBAS: 153624 ;RANDOM NUMBER GENERATOR BASE
1564 000626 032561 RANSAB: 032561 ;RANDOM NUMBER BUFFER
1565 000630 000000 RCSAV: 0 ;RECORD COUNT SAVE
1566 000632 000000 FCSAV: 0 ;FRAME COUNT SAVE
```



```
1567  
1568  
1569  
1570 000634 000000  
1571 000636 000000  
1572 000640 000000  
1573 000642 000000  
1574 000644 000000  
1575 000646 000000  
1576 000650 000000  
1577 000652 000000  
1578 000654 000000  
1579 000656 000000  
1580 000660 000000  
1581 000662 000000  
1582 000664 000000  
1583 000666 000000  
1584 000670 000000  
1585 000672 000000  
1586 000674 000000  
1587 000676 000000  
1588 000700 000000  
1589 000702 000000  
1590 000704 000000  
1591 000706 000000  
1592 000710 000000  
1593 000712 000000  
1594 000714 000000  
1595 000716 000000  
1596 000720 000000  
1597 000722 000000  
1598 000724 000000  
1599 000726 000000  
1600 000730 000000  
1601 000732 000000  
1602 000734 000000  
1603 000736 000000  
1604 000740 000000  
1605 000742 000000  
1606 000744 000000
```

: FLAGS AND COUNTERS*****

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

```
1607  
1608  
1609  
1610 000746 000000 UN1: 0 ;UNIT ORDER AND DESCRIPTION TABLE *****  
1611 000750 000000 UN2: 0 ;THIS TABLE IS LOADED  
1612 000752 000000 UN3: 0 ;WITH UNIT NUMBERS AND  
1613 000754 000000 UN4: 0 ;THEIR DESCRIPTIONS IN  
1614 000756 000000 UN5: 0 ;THE ORDER THAT THEY  
1615 000760 000000 UN6: 0 ;WILL BE TESTED  
1616 000762 000000 UN7: 0  
1617 000764 000000 UN8: 0  
1618 000766 177777 UNX: -1
```

```
1619  
1620  
1621  
1622 000770 001210  
1623 000772 001230  
1624 000774 001250  
1625 000776 001270  
1626 001000 001310  
1627 001002 001330  
1628 001004 001350  
1629 001006 001370  
1630 001010 001410  
1631 001012 001430  
1632 001014 001450  
1633 001016 001470  
1634 001020 001510  
1635 001022 001530  
1636 001024 001550  
1637 001026 001570  
1638  
1639  
1640  
1641 001030 001610  
1642 001032 001714  
1643 001034 002020  
1644 001036 002124  
1645 001040 002230  
1646 001042 002334  
1647 001044 002440  
1648 001046 002544  
1649  
1650  
1651  
1652 001050 000000  
1653 001052 000000  
1654 001054 000000  
1655 001056 000000  
1656 001060 000000  
1657 001062 000000  
1658 001064 000000  
1659 001066 000000  
1660  
1661  
1662
```

:UNIT DROPS AND PICKS POINTERS*****

```
PIK1: BP00  
PIK2: BP10  
PIK3: BP20  
PIK4: BP30  
PIK5: BP40  
PIK6: BP50  
PIK7: BP60  
PIK8: BP70  
DRP1: BD00  
DRP2: BD10  
DRP3: BD20  
DRP4: BD30  
DRP5: BD40  
DRP6: BD50  
DRP7: BD60  
DRP8: BD70
```

:UNIT BAD TAPE POINTERS*****

```
BTADDR: BT00  
BT01  
BT02  
BT03  
BT04  
BT05  
BT06  
BT07
```

:UNIT WRITE RETRY COUNTER*****

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

:UNIT WRITE ERRORS*****

1663	001070	000000	WTER1:	0
1664	001072	000000	WTER2:	0
1665	001074	000000	WTER3:	0
1666	001076	000000	WTER4:	0
1667	001100	000000	WTER5:	0
1668	001102	000000	WTER6:	0
1669	001104	000000	WTER7:	0
1670	001106	000000	WTER8:	0
1671				
1672				
1673				:UNIT READ FORWARD ERRORS*****
1674	001110	000000	RDER1:	0
1675	001112	000000	RDER2:	0
1676	001114	000000	RDER3:	0
1677	001116	000000	RDER4:	0
1678	001120	000000	RDER5:	0
1679	001122	000000	RDER6:	0
1680	001124	000000	RDER7:	0
1681	001126	000000	RDER8:	0
1682				
1683				:UNIT DATA ERRORS FORWARD*****
1684				
1685	001130	000000	DATER1:	0
1686	001132	000000		0
1687	001134	000000		0
1688	001136	000000		0
1689	001140	000000		0
1690	001142	000000		0
1691	001144	000000		0
1692	001146	000000		0
1693				
1694				:UNIT READ REVERSE ERRORS*****
1695				
1696	001150	000000	RDERR1:	0
1697	001152	000000		0
1698	001154	000000		0
1699	001156	000000		0
1700	001160	000000		0
1701	001162	000000		0
1702	001164	000000		0
1703	001166	000000		0
1704				
1705				:UNIT DATA ERRORS REVERSE*****
1706				
1707	001170	000000	DEREV1:	0
1708	001172	000000		0
1709	001174	000000		0
1710	001176	000000		0
1711	001200	000000		0
1712	001202	000000		0
1713	001204	000000		0
1714	001206	000000		0

			:DROPS + PICKS PER CHANNEL PER UNIT*****
1715			
1716			
1717	001210	000000	BP00: 0
1718		001230	0 =.+16
1719	001230	000000	BP10: 0
1720		001250	0 =.+16
1721	001250	000000	BP20: 0
1722		001270	0 =.+16
1723	001270	000000	BP30: 0
1724		001310	0 =.+16
1725	001310	000000	BP40: 0
1726		001330	0 =.+16
1727	001330	000000	BP50: 0
1728		001350	0 =.+16
1729	001350	000000	BP60: 0
1730		001370	0 =.+16
1731	001370	000000	BP70: 0
1732		001410	0 =.+16
1733	001410	000000	BD00: 0
1734		001430	0 =.+16
1735	001430	000000	BD10: 0
1736		001450	0 =.+16
1737	001450	000000	BD20: 0
1738		001470	0 =.+16
1739	001470	000000	BD30: 0
1740		001510	0 =.+16
1741	001510	000000	BD40: 0
1742		001530	0 =.+16
1743	001530	000000	BD50: 0
1744		001550	0 =.+16
1745	001550	000000	BD60: 0
1746		001570	0 =.+16
1747	001570	000000	BD70: 0
1748		001610	0 =.+16
1749			
1750			

```
1751
1752
1753 ;UNIT BAD TAPE COUNTER:16 PER SLAVE*****
1754 001610 000000 BT00: 0
1755 001714 000000 BT01: 0
1756 001714 000000 BT01: 0
1757 002020 000000 BT02: 0
1758 002020 000000 BT02: 0
1759 002124 000000 BT03: 0
1760 002124 000000 BT03: 0
1761 002230 000000 BT04: 0
1762 002230 000000 BT04: 0
1763 002334 000000 BT05: 0
1764 002334 000000 BT05: 0
1765 002440 000000 BT06: 0
1766 002440 000000 BT06: 0
1767 002544 000000 BT07: 0
1768 002544 000000 BT07: 0
1769 002650 000000
1770
1771 ;UNIT END OF TAPE COUNTERS 1 PER SLAVE*****
1772
1773 002650 000000 EOTCO: 0
1774 002652 000000
1775 002654 000000
1776 002656 000000
1777 002660 000000
1778 002662 000000
1779 002664 000000
1780 002666 000000
1781
1782 ;UNIT READ FORWARD SOFT ERROR*****
1783
1784 002670 000000 RFSOFT: 0
1785 002672 000000
1786 002674 000000
1787 002676 000000
1788 002700 000000
1789 002702 000000
1790 002704 000000
1791 002706 000000
1792
1793 ;UNIT READ REVERSE SOFT ERROR*****
1794
1795 002710 000000 RRSOFT: 0
1796 002712 000000
1797 002714 000000
1798 002716 000000
1799 002720 000000
1800 002722 000000
1801 002724 000000
1802 002726 000000
1803
```

C
C

```
1804
1805
1806
1807 002730 000000
1808 002732 000000
1809 002734 000000
1810 002736 000000
1811 002740 000000
1812 002742 000000
1813 002744 000000
1814 002746 000000
1815
1816
1817
1818 002750 000000
1819 002752 000000
1820 002754 000000
1821 002756 000000
1822 002760 000000
1823 002762 000000
1824 002764 000000
1825 002766 000000
1826
1827
1828
1829 002770 002770
1830 002772 014412
1831 002774 014556
1832 002776 014576
1833 003000 014602
1834 003002 014626
1835 003004 014636
1836 003006 014644
1837 003010 014652
1838 003012 014700
1839 003014 014730
1840 003016 014750
1841 003020 014772
1842 003022 015002
1843 003024 015032
1844
```

:UNIT READ FORWARD HARD ERROR*****
RFHARD: 0
0
0
0
0
0
0
0
0

:UNIT READ REVERSE HARD ERROR*****
RRHARD: 0
0
0
0
0
0
0
0
0

:DATA PATTERN GENERATORS*****
DATBL: . :ENTRY TABLE
DATA0: DAT0 :EXTERNAL INPUT FROM H/S READER(SEE MAINDEC-11-DZTUF)
DATA1: DAT1 :ALL ONES
DATA2: DAT2 :ALL ZEROS
DATA3: DAT3 :WALKING ONE
DATA4: DAT4 :WALKING ZERO
DATA5: DAT5 :ALTERNATING ONE/ZERO
DATA6: DAT6 :ALTERNATING ZERO/ONE
DATA7: DAT7 :ALTERNATING ONE/ZERO IN ALTERNATING CHARACTERS
DATA10: DAT10 :WALKING ONE/ALL ONE IN ALTERNATING CHARACTERS
DATA11: DAT11 :ALL BITS 0-377
DATA12: DAT12 :ALL BITS 377-0
DATA13: DAT13 :ALTERNATING CHARACTERS 0 AND 377
DATA14: DAT14 :WALKING ZERO/ALL ZERO IN ALTERNATING CHARACTERS
DATA15: DAT15 :AUTO SEQUENCE PATTERN 0,0,-1,-1,-1,0,0

1845
1846
1847
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

.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 SPECIFIED DATA PATTERN,
:GENERATE ANY RANDOM NUMBER AND EXECUTE
:THE TEST CYCLE REQUESTED BY THE SWITCH SETTING.
:AT THE END OF THE TEST CYCLE THE UNIT IS SELECTED
:AND CHECKED FOR AVAILABILITY AND THE NEXT CYCLE IS
:EXECUTED ON IT.
:THE READ WRITE STATS MAY BE PRINTED AT THE END OF
:EACH TEST CYCLE VIA CONSOLE SWITCH FOURTEEN (14).
:*****

:START 200 & 300 *****
START: MOV #500,SP ;++G SET STACK PTR
CLR ASEQF ;CLEAR AUTO SEQUENCE FLAG
CLR (PC)+ ;CLEAR CHAIN INDICATOR
CHNFLG: .WORD 0 ;CHAIN MODE INDICATOR
;1/0 = CHAIN/NOT CHAIN MODE
CMP #SENDAD,@#42 ;BRANCH IF LOADED VIA ACT11 CHAIN MODE
BEQ 50\$
TST @#42 ;BRANCH IF IN DUMP MODE
BEQ 52\$
BR 51\$
50\$: MOV #SWREG,SWR ;INVOKE SOFTWARE SWR
MOV #100000,@SWR ;HALT ON ERROR
51\$: INC CHNFLG ;SET CHNFLG = CHAIN MODE
JMP 3\$;GO TO CHAIN ADDRESS
52\$: CMPB #6,@#41 ;++G BRANCH IF NOT LOADED VIA TMDP
BNE STAUT
MOV #MSG120,R4
TTOUTT ;++G ADVISE USER TO REMOVE TMDP
BR STAUT ;++G
3\$: INC ASEQF ;++G SET AUTO SEQUENCE FLAG
JMP SUSWR ;CHECK AND SET UP HRD/SOFT SWITCH REG ++ C.W

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

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

:START 210 *****
STARTA: CLR TINF ;CLEAR TTY ENTRY FLAG
STARTB: MOV #IOB,RO
MOV #37,R1
STARTO: CLR (RO)+ ;CLEAR FLAGS AND COUNTERS

1901	003176	005301		DEC	R1	
1902	003200	001375		BNE	STARTO	
1903	003202	012706	000500	MOV	#500,SP	:SET STACK POINTER
1904	003202	004739	004372	JSR	PC,RANSET	:GO RESET RANDOM BASE
1905	003212	012700	001050	MOV	#R1Y1,R0	
1906	003216	012701	000750	MOV	#750,R1	
1907	003222	005020		STARTF: CLR	(R0)+	:CLEAR STATISTIC COUNTERS
1908	003224	005301		DEC	R1	
1909	003226	001375		BNE	STARTF	
1910	003230	012737	177777 014406	MOV	#-1,PATS	:PRESET PATTERN
1911	003236	005037	000744	CLR	EOPB1	
1912	003242	012737	000001 000654	STARTE: MOV	#1,BLCNTR	:PRESET BLOCK COUNTER
1913	003250	012706	000500	STARTD: MOV	#500,SP	
1914	003254	012777	000340 175322	MOV	#340,@PSW	
1915	003262	013746	000006	SUSWR: MOV	@#6,-(SP)	:SAVE VECTORS
1916	003266	013746	000004	MOV	@#4,-(SP)	
1917	003272	012737	003312 000004	MOV	#1,@#4	:SET UP FOR TIMEOUT
1918	003280	022777	177777 175300	CMP	#-1,@SWR	:REFERENCE HARDWARE SWITCH REGISTER
1919	003306	001402		BEQ	2\$	
1920	003310	000404		BR	3\$	
1921	003312	022626		1\$: CMP	(SP)+,(SP)+	:ADJUST STACK
1922	003314	012737	000176 000606	2\$: MOV	#SWREG,SWR	:POINT TO SOFTWARE SWITCH REG
1923	003322	012637	000004	3\$: MOV	(SP)+,@#4	:RESTORE VECTORS
1924	003326	012637	000006	MOV	(SP)+,@#6	
1925	003332	022737	000176 000606	CMP	#SWREG,SWR	:IS SWREG SELECTED
1926	003340	001020		BNE	4\$	
1927	003342	005737	000744	TST	EOPB1	
1928	003346	001015		BNE	4\$	
1929	003350	005037	000744	CLR	EOPB1	
1930	003354	022737	005116 000042	CMP	#SENDAD,@#42	:ACT MODE? ++ C.W
1931	003362	001402		BEQ	6\$:BRANCH - IF YES ++ C.W
1932	003364	004737	024460	JSR	PC,CNTLU	:CHECK FOR CONTROL G
1933	003370	005737	000734	6\$: TST	ASEQF	:AUTO SEQ MODE? ++ C.W
1934	003374	001402		BEQ	4\$:BRANCH - IF NO ++ C.W
1935	003376	000137	022016	JMP	ASEQO	:GO DO AUTO SEQ ++ C.W
1936	003402	004737	012400	4\$: JSR	PC,TINP	:GO GET PARAMETERS FROM TTY
1937	003406	012777	000040 175104	MOV	#40,@CS	:INITIALIZE
1938	003414	005000		STAUTO: CLR	R0	:POINT TO FIRST ENTRY
1939	003416	022760	177777 000746	1\$: CMP	#-1,UN1(R0)	:++G BRANCH IF LAST ENTRY
1940	003424	001406		BEQ	2\$	
1941	003426	042760	100000 000746	BIC	#100000,UN1(R0)	:CLEAR EOT FLAG
1942	003434	062700	000002	ADD	#2,R0	:POINT TO NEXT UNIT ENTRY
1943	003440	000766		BR	1\$:++G CONTINUE CLEARING
1944	003442	013703	005156	2\$: MOV	REOTC,R3	
1945	003446	000303		SWAB	R3	
1946	003450	110337	005156	MOV	R3,REOTC	:RESTORE EOT CNTR
1947	003454	012777	000100 175126	START1: MOV	#100,@TKS	:SET TTY INTERRUPT ENABLE
1948	003462	013700	000674	MOV	UNP,R0	:R0 = UN'T TABLE POINTER
1949	003466	022760	177777 000746	STAR1A: CMP	#-1,UN1(R0)	:++G BRANCH IF LAST ENTRY
1950	003474	001404		BEQ	STAR1B	:IF LAST UNIT IN STRING: BR
1951	003476	016037	000746 000552	MOV	UN1(R0),UDES	:LOAD NEXT UNIT DESCRIPTION
1952	003504	000446		BR	START4	:++G
1953	003506	005237	000654	STAR1B: INC	BLCNTR	:BUMP BLOCK COUNTER
1954	003512	005737	000734	TST	ASEQF	:SEE IF AUTO SEQ
1955	003516	001411		BEQ	STAR1C	:IF NOT: BR
1956	003520	023737	000654 000740	CMP	BLCNTR,ABL CNT	:SEE IF DONE SEQ

1957	003526	001005				BNE	STAR1C		:IF NOT: BR
1958	003530	005037	000654			CLR	BLCNTR		:RESET BLOCK CNTR
1959	003534	005037	000674			CLR	UNP		:RESET UNIT POINTER
1960	003540	000207				RTS	PC		:RETURN TO AUTO SEQ
1961									
1962	003542	005037	000674			STAR1C:	CLR	UNP	
1963	003546	005000				CLR	RO		
1964	003550	016037	000746	000552		MOV	UN1(RO), UDES		:LOAD FIRST UNIT DESCRIPTION
1965	003556	105777	175024			TSTB	@SWR		:++G BRANCH IF NOT RANDOM RECORD
1966	003562	100003				BPL	START2		:++G SIZE REQUESTED.
1967	003567	001402				BEQ	START2		:IF NOT: BR
1968	003566	004737	012314			JSR	PC, CCNTR		:GO GENERATE RANDOM RECORD SIZE
1969	003572	032777	000400	175006		START2:	BIT	#400, @SWR	:SEE IF RANDOM DATA
1970	003600	001402				BEQ	START3		:IF NOT: BR
1971	003602	004737	015102			JSR	PC, DATR		:GO GENERATE RANDOM DATA
1972	003606	032777	000100	174772		START3:	BIT	#100, @SWR	:SEE IF RANDOM RECORD COUNT
1973	003614	001402				BEQ	START4		:IF NOT: BR
1974	003616	004737	012354			JSR	PC, RCNTR		:GO GENERATE RANDOM RECORD COUNT
1975	003622	005760	000746			START4:	TST	UN1(RO)	:++G BRANCH IF NOT AT EOT
1976	003626	100003				BPL	STAR40		:IF NOT: BR
1977	003630	000137	004360			JMP	START7		:ELSE GO TO NEXT UNIT
1978	003634	013777	000550	174656		STAR40:	MOV	DVN, @CS	:SET DRIVE NUMBER
1979	003642	013777	000552	174672		MOV	UDES, @C2		:SET UNIT NUMBER
1980	003650	105777	174646			TSTB	@DS		:++G BRANCH IF UNIT AVAIL
1981	003654	100412				BMI	STAR4A		
1982	003656	005337	000666			DEC	STAL		
1983	003662	001327				BNE	START4		:AWAIT TUR
1984	003664	004737	022736			JSR	PC, PAPRT		:PRINT HEADER
1985	003670	012704	026074			MOV	#MSG49, R4		
1986	003674	104000				TTOUTT			:PRINT NOT AVAIL
1987	003676	104006				STOPP			:STOP
1988	003700	000750				BR	START4		:++G RETRY
1989	003702	013746	000552			STAR4A:	MOV	UDES, -(SP)	:GET UNIT DESCRIPTION
1990	003706	042716	175400			BIC	#175400, (SP)		:++G CLEAR ALL BUT FORMAT BITS
1991						:CMP	#1700, (SP)+		:++G BRANCH IF NRZ
1992	003712	032726	002000			BIT	#2000, (SP)+		:++H BRANCH IF NZR
1993	003716	001406				BEQ	1\$:++G
1994	003720	032777	000040	174574		BIT	#40, @DS		:++G BRANCH IF SLAVE IN PE FORMAT
1995	003726	001002				BNE	1\$:++G
1996	003730	000137	004360			JMP	START7		:++G GO TO NEXT UNIT
1997	003734	004737	014200			1\$:	JSR	PC, DSUP	:GO SET UP WRITE DATA
1998	003740	004737	005160			JSR	PC, RWND		:REWIND
1999	003744	004737	005592			JSR	PC, WRITE		:WRITE
2000	003750	013737	000666			MOV	T\$AL, STAL		:SET TURN AROUND DELAY
2001	003756	004737	01604			JSR	PC, STALL		:DELAY
2002	003762	004737	00714			JSR	PC, RSEQ		:GO TO READ SEQUENCER
2003	003766	013737	000576	000666		MOV	T\$AL, STAL		:SET TURN AROUND DELAY
2004	003774	004737	012304			JSR	PC, STALL		:DELAY
2005	004000	032777	040000	174600		BIT	#40000, @SWR		:SEE IF SHOULD PRINT STATISTICS
2006	004006	001541				BEQ	START5		:IF NOT: BR
2007	004010	012700	000001			MOV	#1, RO		:SET RECORD COUNTER TO 1
2008	004014	004737	022736			JSR	PC, PAPRT		:PRINT CYCLE NUMBER
2009	004020	004737	004030			JSR	PC, STP		:GO PRINT STATS
2010	004024	000137	004276			JMP	STPX		
2011	004030	004737	017250			STAR:	JSR	PC, DPPRT	:PRINT DROPS AND PICKS
2012	004034	012704	026307			MOV	#MSG65, R4		

2013	004040	104000	TTOUTT	:PRINT RETRY TOTAL
000000	000000	013704	MOV	UNP,R4
000000	004404	016403	MOV	RTYf(R4),R3
000000	000000	104000	OCTPP	:PRINT RETRIES
000000	004404	012700	MOV	#MSG73,R4
000000	000000	104000	TTOUTT	:PRINT WRITE ERROR TAG
000000	004404	013704	MOV	UNP,R4
000000	000000	016403	MOV	WTER1(R4),R3
000000	004404	104000	OCTPP	:PRINT WRITE ERRORS
000000	000000	012700	MOV	#MSG72,R4
000000	004404	104000	TTOUTT	:PRINT READ FORWARD ERROR TAG
000000	000000	013704	MOV	UNP,R4
000000	004404	016403	MOV	RDER1(R4),R3
000000	000000	104000	OCTPP	:PRINT READ FORWARD ERRORS
000000	004404	012700	MOV	#MSG113,R4
000000	000000	104000	TTOUTT	:PRINT SOFT TAG
000000	004404	013704	MOV	UNP,R4
000000	000000	016403	MOV	RFSOFT(R4),R3
000000	004404	104000	OCTPP	:PRINT FORWARD SOFT ERRORS
000000	000000	012700	MOV	#MSG114,R4
000000	004404	104000	TTOUTT	:PRINT HARD TAG
000000	000000	013704	MOV	UNP,R4
000000	004404	016403	MOV	RFHARD(R4),R3
000000	000000	104000	OCTPP	:PRINT HARD FORWARE ERRORS
000000	004404	012700	MOV	#MSG77,R4
000000	000000	104000	TTOUTT	:PRINT DATA ERROR FORWARD TAG
000000	004404	013704	MOV	UNP,R4
000000	000000	016403	MOV	DATER1(R4),R3
000000	004404	104000	OCTPP	:PRINT DATA ERROR FORWARD NUMBER
000000	000000	012700	MOV	#MSG68,R4
000000	004404	104000	TTOUTT	:PRINT READ ERROR REVERSE TAG
000000	000000	013704	MOV	UNP,R4
000000	004404	016403	MOV	RDER1(R4),R3
000000	000000	104000	OCTPP	:PRINT REVESE ERROR NUMBER
000000	004404	012700	MOV	#MSG113,R4
000000	000000	104000	TTOUTT	:PRINT SOFT TAG
000000	004404	013704	MOV	UNP,R4
000000	000000	016403	MOV	RRSOFT(R4),R3
000000	004404	104000	OCTPP	:PRINT REVERSE SOFT ERROR
000000	000000	012700	MOV	#MSG114,R4
000000	004404	104000	TTOUTT	:PRINT HARD TAG
000000	000000	013704	MOV	UNP,R4
000000	004404	016403	MOV	RRHARD(R4),R3
000000	000000	104000	OCTPP	:PRINT DATA ERROR REVERSE TAG
000000	004404	012700	MOV	#MSG76,R4
000000	000000	104000	TTOUTT	:PRINT DATA REVERSE ERROR NUMBER
000000	004404	013704	MOV	UNP,R4
000000	000000	016403	MOV	DEREV1(R4),R3
000000	004404	104000	OCTPP	:PRINT DATA REVERSE ERROR NUMBER
000000	000000	012700	RTS	PC
000000	004404	104000	STPX: INC	BTSTF
000000	000000	013704	JSR	PC,BTPRT
000000	004404	016403	CLR	BTSTF
000000	000000	012700	START5: MOV	@SWR,R0
000000	004404	013704	BIC	#177762,R0
000000	000000	016403	CMP	#15,R0
000000	004404	104000		:SEE IF HAVE READ OR WRITE

```

2069 004326 001417
2070 004330 105777 174166
2071 004334 100411
2072 004336 005337 000666
2073 004342 001372
2074 004344 004737 022736
2075 004350 012704 026074
2076 004354 104000
2077 004356 104006
2078 004360 067737 000002 000674
2079 004366 000137 003454
2080
2081
2082
2083
2084 004372 012737 153624 000624
2085 004400 012737 032561 000626
2086 004406 013737 000630 000554
2087 004414 013737 000632 000556
2088 004422 000207

START6: BEQ START8 ;IF NOT: BR
          TSTB @DS ;++G BRANCH IF HAVE UNIT READY
          BMI START7 ;++G
          DEC STAL
          BNE START6 ;DELAY FOR TUR
          JSR PC,PAPRT ;PRINT HEADER
          MOV #MSG49,R4
          TTOUTT ;PRINT NOT AVAIL
          STOP ;STOP
START7: ADD #2,UNP ;POINT TO NEXT UNIT
START8: JMP START1 ;CONTINUE

;RANDOM BASE RESET*****
RANSET: MOV #153624,RANBAS ;RESET BASE
          MOV #32561,RANSAV ;RESET BUFFER
          MOV RCSAV,RCNT ;RESET RECORD COUNT
          MOV FCSAV,FCNT ;RESET FRAME COUNT
          RTS PC
  
```

2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144

```
004424 013777 000552 174110 REOT: MOV UDES,BC2 ;LOAD COMMAND REGISTER
004432 013777 000011 174050 MOV #11,BC1 ;DRIVE CLEAR
004440 105777 174056 1S: TSTB @DS ;++G WAIT FOR DRIVE READY
004444 100375 BPL 1S ;AWAIT DRY
004446 012777 000007 174034 MOV #7,BC1 ;START REWIND
004454 005737 000724 TST BTFLG ;SEE IF BAD TAPE OVERFLOW REWIND
004460 001004 BNE REOT1A ;IF SO: BR
004462 013700 MOV EOTREC,R0
004464 042700 100000 BIC #100000,R0 ;SET RECORD NUMBER OF EOT
004472 005037 000660 REOT1A: CLR EOTREC ;++G CLEAR EOT IND & REC CTR
004474 004737 022736 JSR PC,PAPRT ;PRINT HEADER
004476 022737 000002 000724 CMP #2,BTFLG ;SEE IF POSITION ERROR
004478 001003 BNE REOT1B ;IF NOT: BR
004480 012704 MOV #MSG109,R4 ;SET POSITION ERROR MSG
004482 000406 BR REOT1F
004484 022737 000001 000724 REOT1B: CMP #1,BTFLG ;SEE IF BAD TAPE OVERFLOW
004486 001004 BNE REOT1C ;IF NOT: BR
004488 012704 MOV #MSG106,R4 ;SET BAD TAPE OVERFLOW MSG
004490 104000 REOT1F: TTOUTT ;PRINT REWIND REASON
004492 000412 BR REOT1E
004494 012704 025126 REOT1C: MOV #MSG20,R4 ;SET EOT MSG
004496 104000 REOT1D: TTOUTT ;PRINT MSG
004498 013704 000674 MOV UNP,R4
004500 005264 002650 INC EOTC(R4) ;BUMP CNTR
004502 016403 002650 MOV EOTC(R4),R3
004504 104000 OCTPP ;PRINT EOT CNTR
004506 012704 027004 REOT1E: MOV #MSG16A,R4
004508 104000 TTOUTT ;PRINT RESTART MSG
004510 005037 000724 CLR BTFLG ;CLEAR BAD TAPE FLAG
004512 004737 004030 JSR PC,STP ;PRINT STATS
004514 004737 007324 JSR PC,BTPRT ;PRINT BAD TAPE STATS
004516 105777 173710 REOT2: TSTB @DS ;++G BRANCH IF UNIT IS READY
004518 100414 BMI REOT2A
004520 005337 000666 DEC STAL
004522 001372 BNE REOT2 ;WAIT DRY
004524 012737 024767 000652 MOV #MSG6,FMADDR
004526 004737 022736 JSR PC,PAPRT ;PRINT HEADER
004528 012704 026251 MOV #MSG60,R4
004530 104000 TTOUTT ;PRINT NO DRIVE READY
004532 104000 STOPP
004534 105337 05156 REOT2A: DECB REOTC ;SEE IF LAST UNIT TO REACH EOT
004536 011410 BEQ REOT3 ;IF SO: BR
004538 013700 000674 MOV UNP,R0
004540 052760 100000 000746 BIS #100000,UN1(R0) ;SET EOT FLAG
```

2145	004664	005726		TST	(SP)+	:RESET STACK POINTER
2146	004666	000137	004360	JMP	START7	:GO TO NEXT UNIT
2147	004668	000337	005156	REOT3:	SWAB	REOTC
2148	004670	013700	005156		MOV	REOTC,R0
2149	004672	000337	005156		SWAB	REOTC
2150	004674	110037	005156		MOVB	R0,REOTC
2151	004676	005037	000674		UNP	:RESTORE EOT UNIT COUNTER
2152	004678	013700	000674		CLR	UNP,R0
2153	004680	016037	000746	000552	REOT4:	MOV UN1(RC),UDES
2154	004682	013700	000552	173604		:LOAD UNIT DESCRIPTION
2155	004684	032777	000552	173556	REOT5:	MOV UDES,@C2
2156	004686	001374	020000		BIT	#20000,@DS
2157	004688	032777	000002	173546		BNE REOT5
2158	004690	001012			BIT	#2,@DS
2159	004692	012700	000001		BNE	REOT6
2160	004694	004737	022736		MOV	#1,R0
2161	004696	012704	026042		JSR	PC,PAPRT
2162	004698	104000			MOV	#MSG48,R4
2163	004700	104006			TTOUTT	:PRINT BOT ERROR
2164	004702	013700	000674		STOPP	
2165	004704	042760	100000	000746	REOT6:	BIC #100000,UN1(R0)
2166	004706	062737	000002	000674		:CLEAR EOT FLAG
2167	004708	013700	000674		ADD	#2,UNP
2168	004710	022760	177777	000746	MOV	UNP,R0
2169	004712	001334			CMP	#-1,UN1(R0)
2170	004714	005037	000674		BNE	REOT4
2171	004716	005037	000634		REOT7:	CLR UNP
2172	004718	005037	000734		CLR	TINF
2173	004720	001402			TST	ASEQF
2174	004722	005726			BEQ	REOTX
2175	004724	000412			TST	(SP)+
2176	004726				BR	TEND
2177	005054	004737	004372		REOTX:	JSR PC,RANSET
2178	005060	012737	177777	014406		:GO RESET RANDOM BASE
2179	005066	005037	015150		MOV	#-1,PATS
2180	005072	005737	000570		CLR	RDFL
2181	005078	001420			TST	SPFLG
2182	005100	012704	026652		BEQ	REOTXX
2183	005104	104000			TEND:	MOV #MSG100,R4
2184	005106	013700	000042		TTOUTT	:PRINT END OF PASS
2185	005112	001405			MOV	@#42,R0
2186	005114	000005			BEQ	HERE
2187	005116	004710			RESET	:GET ACT11 RETURN ADDRESS
2188	005120	000240			SENDAD:	JSR PC,(R0)
2189	005122	000240			NOP	
2190	005124	000240			NOP	
2191	005126	005737	000734		HERE:	TST ASEQF
2192	005128	001401			BEQ	1\$
2193	005130	000207			RTS	PC
2194	005132	104006			1\$:	STOPP
2195	005140	012704	026652		REOTXX:	MOV #MSG100,R4
2196	005144	104000			TTOUTT	:GET END OF PASS MESSAGE ++ C.W
2197	005146	005237	000744		INC	EOPB1
2198	005152	000137	003242		JMP	STARTE
2199	005156	000000			REOTC:	0

2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255

005160 032777 001000 173420
005166 001001
005170 000207
005172 013737 000674 000714
005200 005037 000674
005204 005037 000660
005210 000337 005156
005214 013700 005156
005220 000337 005156
005224 110037 005156
005228 013700 000674
005232 022760 177777 000746
005236 001445
005240 005760 000746
005244 100637
005248 016037 000746 000552
005252 013777 000552 173254
005256 012777 000011 173214
005260 012777 000007 173206
005264 105777 173214
005268 100614
005272 005337 000666
005276 001337
005280 012737 024767 000652
005284 004737 022736
005288 012704 026372
005292 104000
005296 104006
005300 042760 100000 000746
005304 062737 000002 000674
005308 000725
005312 005037 000674
005316 013700 000674
005320 022760 177777 000746
005324 001436
005328 016037 000746 000552
005332 013777 000552 173150
005336 032777 020000 173102
005340 001374
005344 032777 000002 173072
005348 001407
005352 062737 000002 000674
005356 012777 000011 173042
005360 000745
005364 012700 000001
005368 004737 022736
005372 012704 026042

```
*****  
:REWIND ALL AVAIL TAPES:  
:THIS ROUTINE: ENTERED VIA CONSOLE SWITCH NINE (9),  
:WILL REWIND ALL AVAILABLE TAPES TO BOT NO MATTER  
:WHERE THEY ARE CURRENTLY POSITIONED AND RESUME TESTING  
:ON THE CURRENTLY SELECTED UNIT.  
*****  
RWND: BIT #1000,@SWR :SEE IF SHOULD REWIND  
BNE RWNDA :IF SO: BR  
RTS PC :ELSE EXIT  
RWNDA: MOV UNP,UPS :SAVE UNIT POINTER  
CLR UNP :CLEAR POINTER  
CLR EOTREC :CLEAR EDT FLAG  
SWAB REOTC  
MOV REOTC,R0  
SWAB REOTC  
MOVB R0,REOTC :RESTORE EOT UNIT COUNTER  
RWND0: MOV UNP,R0 :POINT TO UNIT ENTRY  
CMP #-1,UN1(R0) :++G BRANCH IF LAST ENTRY  
BEQ RWND2 :IF SO: BR  
TST UN1(R0) :++G BRANCH IF ALREADY REWINDING  
BMI RWND1A :++G  
MOV UN1(R0),UDES :SET UNIT DESCRIPTION  
MOV UDES,@C2 :LOAD COMMAND REGISTER  
MOV #11,@C1 :DRIVE CLEAR  
MOV #7,@C1 :START REWIND  
RWND1: TSTB @DS :++G WAIT FOR DRIVE READY  
B:II RWND1A :IF DRY: BR  
DEC STAL :AWAIT DRY  
BNE RWND1  
MOV #MSG6,EMADDR :PRINT HEADER  
JSR PC,PAPRT  
MOV #MSG70,R4 :PRINT NO DRIVE READY  
TTOUTT  
STOPP  
RWND1A: BIC #100000,UN1(R0) :CLEAR EOT FLAG  
ADD #2,UNP :BUMP POINTER  
BR RWND0 :++G DO NEXT UNIT  
RWND2: CLR UNP :CLEAR POINTER  
RWND3: MOV UNP,R0 :POINT TO UNIT ENTRY  
CMP #-1,UN1(R0) :++G BRANCH IF LAST ENTRY  
BEQ RWNDX :IF SO: BR  
MOV UN1(R0),UDES :SET UNIT DESCRIPTION  
MOV UDES,@C2 :LOAD COMMAND REGISTER  
RWND4: BIT #20000,@DS :AWAIT PIP RESET  
BNE RWND4 :SEE IF HAVE BOT  
BIT #2,@DS :IF NOT: BR  
BEQ RWND6 :BUMP POINTER  
RWND5: ADD #2,UNP :BUMP POINTER  
MOV #11,@C1 :DRIVE CLEAR  
BR RWND3 :++G DO NEXT UNIT  
RWND6: MOV #1,R0 :PRINT HEADER  
JSR PC,PAPRT  
MOV #MSG48,R4
```


2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320

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

```
005522 032777 000001 173056 WRITE: BIT #1,@SWR ;SEE IF SHOULD WRITE  
005530 001402 BEQ W1E ;  
005532 000137 JMP WEX ;IF NOT: BR  
005536 013700 000554 WRTE: MOV RCNT,R0 ;R0=RECORD COUNT  
005542 012737 024762 000652 WO: MOV #MSG5,EMADDR ;SET ERROR MSG ADDRESS  
005550 013777 000556 172740 MOV #FMCNT,@FC ;LOAD CHAR COUNT  
005556 012777 027464 172730 MOV #WDATA,@BA ;SET DATA ADDR  
005564 112737 000060 000672 MOVB #60,MTC1 ;SET WRITE OP COMMAND  
005572 012737 005604 000662 MOV #W1,RTRN ;SET RETURN ADDRESS  
005600 000137 021216 JMP TAPG ;GO EXECUTE COMMAND  
005604 C32777 002000 172710 W1: BIT #2000,@DS ;SEE IF EOT  
005612 001414 BEQ W2 ;IF NOT AT EOT: BR  
005614 005737 000660 TST EOTREC ;++G BRANCH IF WRITTEN PAST EOT  
005620 100411 BMI W2 ;++G  
005622 010037 000660 MOV R0,EOTREC ;SAVE EOT RECORD NUMBER  
005626 052737 100000 000660 BIS #100000,EOTREC ;++G SET EOT FLAG  
005634 005337 000660 DEC EOTREC ;++G ADJUST RECORD COUNT  
005640 012700 000002 MOV #2,R0 ;++G SET R0 TO WRITE 1 MORE RECORD  
005644 032777 010000 172734 W2: BIT #10000,@SWR ;SEE IF SHOULD CHECK ERRORS  
005652 001002 BNE W3 ;IF NOT: BR  
005654 004737 017406 JSR PC,ERCHK ;GO CHECK ERRORS  
005660 013737 000574 000666 W3: MOV #STAL,STAL ;SET DELAY  
005666 004737 012304 JSR PC,STALL ;DELAY
```


2321	005672	005737	000712		TST	RTYFL	:SEE IF RETRY TIME
2322	005676	001401			BEQ	W3A	:IF NOT: BR
2323	005700	000207			RTS	PC	:ELSE RETURN
2324	005702	005737	000706	W3A:	TST	SERFL	:SEE IF WRITE ERROR
2325	005706	001450			BEQ	W5	:IF NOT: BR
2326	005710	013704	000674		MOV	UNP,R4	
2327	005714	005264	001070		INC	WTER1(R4)	:BUMP WRITE ERROR
2328	005720	005037	000706		CLR	SERFL	:CLEAR STATUS ERROR FLAG
2329	005724	032777	000020	172654	BIT	#20,BSWR	:SEE IF RETRY
2330	005732	001456			BEQ	W5	:IF NOT: BR
2331	005734	013733	000722		MOV	ERSAV,R3	
2332	005740	042703	102700		BIC	#102700,R3	:MASK UNRECOVERABLE ERROR
2333	005744	001410			BEQ	W4	:IF SO: BR
2334	005746	004737	022736		JSR	PC,PAPRT	:PRINT HEADER
2335	005752	012704	026551		MOV	#MSG78,R4	
2336	005756	104000			TTOUTT		:PRINT NON-RETRYABLE ERROR TAG
2337	005760	004737	011260		JSR	PC,NRTP	:PRINT ER FOR NON-RETRYABLE
2338	005764	000421			BR	W5	
2339	005766	013704	000674	W4:	MOV	UNP,R4	
2340	005772	005264	001050		INC	RTY1(R4)	:BUMP RETRY CNTR
2341	005776	032777	002000	172602	BIT	#2000,BSWR	:SEE IF PRINT ERRORS
2342	006004	001003			BNE	W4A	:IF NOT: BR
2343	006006	012704	026265		MOV	#MSG64,R4	
2344	006012	104000			TTOUTT		:PRINT ORIGINAL ERROR TAG
2345	006014	005037	000702	W4A:	CLR	RTCNT	:CLEAR RETRY NUMBER
2346	006020	005037	000700		CLR	RPCNT	:CLEAR REPEAT COUNTER
2347	006024	004737	006362		JSR	PC,WRTY	:GO RETRY WRITE ERROR
2348	006030	005037	000712	W5:	CLR	RTYFL	:CLEAR RETRY COUNTER
2349	006034	005300			DEC	R0	:SEE IF DONE ALL
2350	006036	001241			BNE	W0	:IF NOT: BR
2351	006040	005737	000564	W6:	TST	TMEX	:SEE IF TM
2352	006044	001525			BEQ	WEX	:IF NOT: BR
2353	006046	005237	000676		INC	TMFLG	:SET TM FLAG
2354	006052	012737	026172	000652	MOV	#MSG54,EMADDR	:POINT TO TM ERROR MSG
2355	006060	012737	000026	000672	MOV	#26,MTC1	:SET TM OP CODE
2356	006066	012777	000000	172422	MOV	#0,BFC	:LOAD FRAME COUNTER
2357	006074	012777	027464	172412	MOV	#WDATA,BBA	:LOAD BUS ADDRESS
2358	006102	012737	006114	000662	MOV	#WTMO,RTN	:SAVE RETURN ADDRESS
2359	006110	000137	021216		JMP	TAPG	:WRITE TM
2360	006114	032777	010000	172464	BIT	#10000,BSWR	:SEE IF SHOULD CHECK ERRORS
2361	006122	001076			BNE	WEX	
2362	006134	032777	000004	172370	BIT	#4,BDS	:SEE IF TM STATUS
2363	006132	001011			BNE	WTM1	:IF SO: BR
2364	006134	012737	027464	021132	MOV	#WDATA,CADER	:SET EXPT BUS ADDRESS
2365	006142	012737	000001	021140	MOV	#1,DRVER	:INDICATE ERROR
2366	006150	004737	020234		JSR	PC,ERPT	:PRINT TM ERROR
2367	006154	000404			BR	WTM2	
2368	006156	012703	027464	WTM1:	MOV	#WDATA,R3	:SET EXPT ADDRESS
2369	006162	004737	017504		JSR	PC,ER2	:GO CHECK FOR OTHER ERRORS
2370	006166	005737	000712	WTM2:	TST	RTYFL	:SEE IF RETRY
2371	006172	001401			BEQ	WTM3	:IF NOT: BR
2372	006174	000207			RTS	PC	:ELSE RETURN TO RETRY ROUTINE
2373	006176	005737	000706	WTM3:	TST	SERFL	:SEE IF WRITE ERROR
2374	006203	001446			BEQ	WEX	:IF NOT: BR
2375	006204	013704	000674		MOV	UNP,R4	
2376	006210	005264	001070		INC	WTER1(R4)	:BUMP WRITE ERROR

2377	006214	032777	000020	172364	BIT	#20,@SWR	:SEE IF SHOULD RETRY
2378	006223	001436			BEQ	WEX	:IF NOT: BR
2379	006224	013703	000722		MOV	ERSAV,R3	
2380	006225	042703	102700		BIC	#102700,R3	:MASK UNRECOVERABLE ERROR
2381	006226	001410			BEQ	WTM4	:IF SO: BR
2382	006227	004737	022736		JSR	PC,PAPRT	:PRINT HEADER
2383	006228	012704	026551		MOV	#MSG78,R4	
2384	006229	104000			TTOUTT		:PRINT UNRETRYABLE TAG
2385	006230	004737	011260		JSR	PC,NRTP	:PRINT ER FOR NON-RETRYABLE
2386	006231	000421			BR	WEX	
2387	006232	005037	000700	WTM4:	CLR	RPCNT	:CLEAR REPEAT CNTR
2388	006233	013704	000674		MOV	UNP,R4	
2389	006234	003264	001050		INC	RTY1(R4)	:BUMP RETRY CNTR
2390	006235	003037	000702		CLR	RTCNT	:CLEAR RETRY CNTR
2391	006236	032777	002000	172302	BIT	#2000,@SWR	:SEE IF PRINT ERRORS
2392	006237	001003			BNE	WTM4A	:IF NOT: BR
2393	006238	012704	026265		MOV	#MSG64,R4	
2394	006239	104000			TTOUTT		:PRINT ORIGINAL ERROR TAG
2395	006240	004737	006362	WTM4A:	JSR	PC,WRTY	:GO DO RETRY
2396	006241	005037	000712	WEX:	CLR	RTYFL	:CLEAR RETRY FLAG
2397	006242	005037	000676		CLR	TMFLG	:CLEAR TAPE MARK FLAG
2398	006243	005737	000660		TST	EOTREC	:++G BRANCH IF NOT AT EOT
2399	006244	100011			BPL	WRWX	:++G
2400	006245	017703	172244	WRW:	MOV	@SWR,R3	
2401	006246	042703	177763		BIC	#177763,R3	
2402	006247	022703	000014		CMP	#14,R3	:SEE IF WRITE ONLY
2403	006248	001003			BNE	WRWX	:IF NOT: BR
2404	006249	000137	004424		JMP	REOT	:ELSE REWIND
2405	006250	U00207		WRWX:	RTS	PC	:EXIT

2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461

006362 012737 000001 000712
006370 004737 006764
006376 005737 000676
006400 001003
006402 004737 005542
006406 000402
006410 004737 006052
006414 005737 000706
006420 001024
006422 005737 000700
006426 022737 000004 000700
006434 001355
006436 032777 002000 172142
006444 001011
006446 012704 026744
006454 104000
006456 012704 026307
006460 104000
006462 013703 000702
006464 104002
006470 005207
006472 005737 000646
006474 013703 000722
006476 042703 102700
006482 001413
006484 004737 022736
006486 012704 026551
006490 104000
006492 004737 011260
006494 012737 000001 000646
006496 000407
006498 032777 002000 172042 WRTY2A:
006502 001025
006504 012704 027200
006506 104000
006508 012704 026307 WRTY2B:
006510 104000
006512 013703 000702
006514 104002
006516 012704 027222
006518 104000
006520 013703 000700
006522 104002
006524 005737 000646
006526 001403
006528 005037 000646
006530 005207
006532 005737 000702 WRTY3:
006534 001004
006536 013704 000674
006538 005364 001070

```
*****  
:WRITE ERROR RETRY  
*****  
WRTY:  MOV #1,RTYFL      :SET RETRY FLAG  
WRTY0: JSR PC,WRTSB     :GO SPACE REVERSE FOR REPEAT  
      TST TMFLG         :SEE IF TAPE MARK TIME  
      BNE WRTYTM        :IF SO: BR  
      JSR PC,W0         :REWRITE RECORD  
      BR WRTYR          :GO ON  
WRTYTM: JSR PC,WTM      :GO WRITE TAPE MARK AGAIN  
WRTYR:  TST SERFL       :REWRITE GOOD  
      BNE WRTY2        :IF NOT: BR  
      INC RPCNT        :BUMP REPEAT COUNTER  
      CMP #4,RPCNT     :SEE IF FOUR GOOD REPEATS  
      BNE WRTY0        :IF NOT: REPEAT  
      BIT #2000,@SWR   :SEE IF PRINT  
      BNE WRTY1        :IF NOT: BR  
      MOV #MSG105,R4  
      TTOUTT           :PRINT RECOVERED MESSAGE  
      MOV #MSG65,R4  
      TTOUTT           :PRINT RETRY TAG  
      MOV RTCNT,R3  
      OCTPP            :PRINT RETRY NUMBER  
WRTY1:  RTS PC          :RESUME TESTING  
WRTY2:  CLR TEMP3       :++G CLEAR RECOVERABLE ERROR FLAG  
      MOV ERSV,R3      :GET ER  
      BIC #102700,R3   :MASK RECOVERABLE BITS  
      BEQ WRTY2A       :IF RECOVERABLE: BR  
      JSR PC,PAPRT     :PRINT HEADER  
      MOV #MSG78,R4  
      TTOUTT           :PRINT NON-RECOVERABLE MSG  
      JSR PC,WRTP      :PRINT ER  
      MOV #1,TEMP3     :SET FLAG  
      BR WRTY2B  
WRTY2A: BIT #2000,@SWR :SEE IF PRINT  
      BNE WRTY3        :IF NOT: BR  
      MOV #MSG110,R4  
      TTOUTT           :PRINT BAD TAPE SUSPECT  
WRTY2B: MOV #MSG65,R4  
      TTOUTT           :PRINT RETRY TAG  
      MOV RTCNT,R3  
      OCTPP            :PRINT RETRY NUMBER  
      MOV #MSG111,R4  
      TTOUTT           :PRINT REPEAT TAG  
      MOV RPCNT,R3  
      OCTPP            :PRINT REPEAT NUMBER  
      TST TEMP3        :SEE IF DID NON-RECOVERABLE  
      BEQ WRTY3        :IF NOT: BR  
      CLR TEMP3        :CLEAR FLAG  
      RTS PC           :EXIT  
WRTY3:  TST RICNT      :SEE IF FIRST RETRY  
      BNE WRTY3A       :IF NOT: BR  
      MOV UNP,R4  
      DEC WTR1(R4)     :DECREMENT WRITE ERROR CNTR
```

```
2462 006636 013704 000674 WRTY3A: MOV UNP,R4 :GET UNIT NUMBER
2463 006652 016437 001030 000730 MOV BTADDR(R4),BTPT :GET ADDRESS OF UNIT BAD TAPE CNTR
2464 006650 017704 172054 MOV @BTPT,R4 :GET COUNTER
2465 006654 005724 TST (R4)+ :SET POINTER OFFSET
2466 006656 010477 172046 MOV R4,@BTPT
2467 006662 013703 000730 MOV BTPT,R3
2468 006666 060304 ADD R3,R4 :SET ABSOLUTE POINTER
2469 006670 013714 000654 MOV BL(CNTR,(R4)) :SET BLOCK NUMBER
2470 006674 062704 000040 ADD #40,R4 :ADD RCNT OFFSET
2471 006700 013714 000554 MOV RCNT,(R4)
2472 006704 160014 SUB R0,(R4) :SET RECORD NUMBER
2473 006706 005214 INC (R4) :CORRECT RECORD NUMBER
2474 006710 022777 000040 172012 CMP #40,@BTPT :SEE IF TOO MANY BAD SPOTS
2475 006716 001002 BNE WRTY4 :IF NOT: BR
2476 006720 000137 007160 JMP BTOV :ELSE GO TO BAD TAPE OVERFLOW
2477 006724 005237 000702 WRTY4: INC RTCNT :BUMP RETRY COUNTER
2478 006730 022737 000004 000702 CMP #4,RTCNT :SEE IF DONE < RETRIES
2479 006736 001410 BEQ WRTY5 :IF SO: BR
2480 006740 013704 000674 MOV UNP,R4
2481 006744 005264 001050 INC RTY1(R4) :BUMP RETRY COUNTER
2482 006750 005237 000732 INC ERTFL :SET ERASE FLAG
2483 006754 000137 006370 JMP WRTY4
2484 006760 000137 007400 WRTY5: JMP BTUR :ELSE GO TO BAD TAPE UNRECOVERABLE
2485
2486 :WRITE RETRY BACKSPACE-ERASE SUBROUTINE*****
2487
2488 006764 005037 000706 WRTSB: CLR SERFL :CLEAR FLAG
2489 006770 013737 000576 000666 MOV TSTAL,STAL
2490 006776 004737 012304 JSR PC,STALL :DO TURN AROUND DELAY
2491 007002 012737 026320 000652 MOV #MSG66,EMADDR :SET ERROR CODE
2492 007010 012777 177777 171500 MOV #-1,@FC :SET TO BACKSPACE 1 RECORD
2493 007016 012777 033472 171470 MOV #RDATA,@BA :SET BA
2494 007024 004737 012234 JSR PC,BKRT :GO BACKSPACE
2495 007030 005737 000706 TST SERFL :SEE IF ERROR
2496 007036 001406 BEQ WRTSB1 :IF NOT: BR
2497 007042 012737 000002 000724 WRTSB0: MOV #2,BTFLG :SET FLAG
2498 007048 022626 CMP (SP)+,(SP)+ :RESET STACK
2499 007054 000137 004424 JMP REOT :GO REWIND AND REMOVE FROM TESTING
2500 007060 005737 000732 WRTSB1: TST ERTFL :SEE IF SHOULD ERASE
2501 007066 001001 BNE WRTSB2 :IF SO: BR
2502 007072 000201 RTS PC :RETURN
2503 007078 005037 000732 WRTSB2: CLR ERTFL :CLEAR ERASE FLAG
2504 007084 005037 000700 CLR RPCNT :CLEAR REPEAT CNTR
2505 007090 005037 000706 CLR SERFL :CLEAR FLAG
2506 007096 012737 026333 000652 MOV #MSG67,EMADDR :SET ERROR CODE
2507 007102 005037 171406 CLR @FC :CLEAR FRAME COUNT
2508 007108 012737 000024 000672 MOV #24,MTC1 :SET ERASE OP-CODE
2509 007114 012777 027464 171370 MOV #WDATA,@BA :SET BA
2510 007120 012737 007136 000662 MOV #WRTSB3,RTRN :SET RETURN ADDRESS
2511 007126 000137 021216 JMP TAPG :GO ERASE
2512 007132 012703 027464 WRTSB3: MOV #WDATA,R3 :SET EXPT BA
2513 007138 004737 017504 JSR PC,ER2 :GO CHECK ERRORS
2514 007144 005737 000706 TST SERFL :SEE IF ERROR
2515 007150 001737 BEQ WRTSB1 :IF NOT: BR
2516 007156 000137 JMP WRTSB0
```

```
2518 ;BAD TAPE OVERFLOW SUBROUTINE*****
2519
2520 007160 005037 000712
2521 007164 013737 000001 000724 BTOV: CLR RTYFL ;CLEAR RETRY FLAG
2522 007172 005726 ;SET BAD TAPE OVERFLOW FLAG
2523 007174 000137 ;++G ADJUST STACK
2524 007200 013701 004424 ;GO REWIND AND REMOVE FROM TESTING
2525 007204 005721 000730 BTOV0: MOV BTPT,R1 ;SET TABLE POINTER
2526 007206 005000
2527 007210 010003 BTOV1: CLR R0
2528 007212 000241 MOV R0,R3
2529 007214 006003 ROR R3 ;R3=R3/2 FOR CORRECT NUMBER
2530 007216 104002 OCTPP ;PRINT ENTRY NUMBER
2531 007220 012704 025054 MOV #MSG13,R4
2532 007224 105724 TSTB (R4)+ ;SKIP CR/LF
2533 007226 104000 TTOUTT ;PRINT BLOCK NUMBER TAG
2534 007230 011103 MOV (R1),R3
2535 007232 104002 OCTPP ;PRINT BLOCK NUMBER
2536 007234 012704 025062 MOV #MSG14,R4
2537 007240 104000 TTOUTT ;PRINT RECORD NUMBER TAG
2538 007242 062701 000040 ADD #40,R1 ;SET POINTER OFFSET FOR RECOED NUMBER
2539 007246 012103 MOV (R1)+,R3
2540 007250 104002 OCTPP ;PRINT RECORD NUMBER
2541 007252 162701 000040 SUB #40,R1 ;RESET POINTER FOR BLOCK NUMBER
2542 007256 005720 TST (R0)+
2543 007260 020077 171444 CMP R0,BTPT ;SEE IF DONE
2544 007264 001404 BEQ BTOV2 ;IF SO: BR
2545 007266 012704 025405 MOV #MSG28,R4
2546 007270 104000 TTOUTT ;DO CR/LF
2547 007274 000745 BR BTOV1 ;CONTINUE
2548 007276 005737 000726 BTOV2: TSI BTSTF ;SEE IF STAT ONLY PRINT
2549 007302 001007 BNE BTOVX ;IF SO: BR
2550 007304 012703 000041 MOV #41,R3 ;SET SIZE OF TABLE
2551 007310 013704 000730 MOV BTPT,R4 ;SET POINTER
2552 007314 005024 BTOV3: CLR (R4)+ ;CLEAR TABLE
2553 007316 005303 DEC R3 ;SEE IF DONE
2554 007320 001375 BNE BTOV3 ;IF NOT: BR
2555 007322 000207 B.OVX: RTS PC ;RETURN
2556
```

2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581

007324 012704 025405
007330 104000
007332 013704 000674
007336 C16437 001030
007344 017703 171360
007350 000641
007352 006003
007354 104002
007356 012704 027234
007362 104000
007364 005777 171340
007370 001001
007372 000207
007374 000137 007200

007400 004737 022736
007404 012704 027047
007410 104000
007412 000207

000730

```
                ;BAD TAPE STATISTIC PRINT*****  
BTPRT:  MOV      #MSG28,R4  
        TTOUTT                    ;DO CR/LF  
        MOV      UNP,R4  
        MOV      BTADDR(R4),BTPT ;SET TABLE POINTER  
        MOV      @BTPT,R3  
        CLC  
        ROR      R3                ;CORRECT NUMBER  
        OCTPP                    ;PRINT NUMBER OF BAD SPOTS  
        MOV      #MSG112,R4  
        TTOUTT                    ;PRINT BAD TAPE TAG  
        TST      @BTPT              ;SEE IF ANY BAD SPOTS  
        BNE      BTPRT1            ;IF SO: BR  
        RTS      PC                ;ELSE RETURN  
BTPRT1: JMP      BTOVO            ;PRINT STATS  
  
                ;BAD TAPE UNRECOVERABLE SUBROUTINE*****  
BTUR:   JSR      PC,PAPRT          ;PRINT HEADER  
        MOV      #MSG107,R4  
        TTOUTT                    ;PRINT UNRECOVERABLE BAD SPOT MSG  
        RTS      PC                ;RESUME TESTING
```

2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637

```
007414 012737 000002 000562 RSEQ:
007422 017704 171160
007426 02704 177763
007432 005704
007434 001004
007436 032777 000002 171142
007444 001051
007446 032777 000004 171132 P-R:
007454 001005
007456 012737 010000 000562
007464 004737 007732
007470 032777 000010 171110 RSF:
007476 001026
007500 032737 010000 000562
007506 001407
007510 013737 000576 000666
007516 004737 012504
007522 000137 007542
007526 032777 000001 171052 RSF0:
007534 001002
007536 004737 012056
007538 012737 000002 000562 RSF1:
007540 004737 007732
007542 005737 000660 RSEX:
007546 100002
007548 000137 004424
007550 000207
007552
007554
007556
007558
007560
007562
007564
007566
007568
007570
007572
007574
007576
007578
007580
007582
007584
007586
007588
007590
007592
007594
007596
007598
007600
007602
007604
007606
007608
007610
007612
007614
007616
007618
007620
007622
007624
007626
007628
007630
007632
007634
007636
007638
007640
007642
007644
007646
007648
007650
007652
```

```
*****
:READ SEQUENCER:
:THIS ROUTINE IS USED TO DETERMINE THE SEQUENCE
:IN WHICH READ TAPE OPERATIONS ARE TO BE PERFORMED.
:THIS IS NECESSARY WHEN THE UNIT BEING TESTED IS
:CAPABLE OF READING DATA IN BOTH THE FORWARD AND
:REVERSE DIRECTIONS. CONSOLE SWITCHES ONE (1), TWO (2),
:AND THREE (3) ARE USED TO DETERMINE THE READ SEQUENCE.
:CONSOLE SWITCH ONE (1) DETERMINES WHETHER TO READ
:THE BLOCK OF DATA FORWARD FIRST OR REVERSE FIRST.
:SWITCH TWO (2) DISALLOWS READING IN THE REVERSE
:DIRECTION AND SWITCH THREE (3) DISALLOWS READING IN
:THE FORWARD DIRECTION.
*****

MOV #2,RDCMD
MOV @SWR,R4 :READ SWITCHES
BIC #177763,R4 :MASK READ BITS
TST R4 :SEE IF BOTH READS
BNE RSR :IF NOT: BR
BIT #2,@SWR :SEE IF READ REVERSE FIRST
BNE RSFR :IF NOT: BR
BIT #4,@SWR :SEE IF SHOULD READ REVERSE
BNE RSF :IF NOT: BR
MOV #10000,RDCMD :LOAD READ REVERSE COMMAND
JSR PC,READ :GO READ REVERSE
BIT #10,@SWR :SEE IF SHOULD READ FORWARD
BNE RSEX :IF NOT: BR
BIT #10000,RDCMD :SEE IF HAVE READ REVERSE
BEQ RSF0 :IF NOT: BR
MOV TSTAL,STAL
JSR PC,STALL :DO READ STALL
JMP RSF1
BIT #1,@SWR :SEE IF WRITE
BNE RSF1 :IF NOT: BR
JSR PC,BKSP :GO BACKSPACE
MOV #2,RDCMD :LOAD READ FORWARD COMMAND
JSR PC,READ :GO READ
TST EOTREC :++G BRANCH IF NOT AT EOT
BPL 1$ :++G
JMP REOT :++G ELSE GO REWIND
RTS PC :++G EXIT

MOV #10000,RDCMD
BIT #10,@SWR :SEE IF SHOULD READ FORWARD
BNE RSFR1 :IF NOT: BR
BIT #1,@SWR :SEE IF WRITE
BNE RSFR0 :IF NOT: BR
JSR PC,BKSP :GO BACKSPACE TO START
MOV #2,RDCMD :LOAD READ FORWARD COMMAND
JSR PC,READ :GO READ FORWARD
BIT #4,@SWR :SEE IF SHOULD READ REVERSE
BNE RSEX :IF NOT: BR
BIT #10000,RDCMD
BNE RSFR2 :IF READ REVERSE: BR
```

2638	007654	013737	000576	000666	MOV	TSTAL, STAL	:DO READ STALL
2639	007662	004737	012304		JSR	PC, STALL	
2640	007666	012737	010000	000562	RSFR2: MOV	#10000, RDCMD	:LOAD READ REVERSE
2641	007674	004737	007732		JSR	PC, READ	:GO READ REVERSE
2642	007700	005737	000660		TST	EOTREC	:JEE IF AT END OF TAPE
2643	007704	100011			BPL	RSFRX	:++G IF NOT: BR
2644	007706	163737	000554	000660	SUB	RCNT, EOTREC	
2645	007714	005437	000660		NEG	EOTREC	:SET TO PROPER RECORD NUMBER
2646	007720	005237	000660		INC	EOTREC	
2647	007724	000137	004424		JMP	REOT	:ELSE GO TO REWIND
2648	007730	000207			RTS	PC	:EXIT
2649							

2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705

007732 013700 000554
007736 005737 000660
007742 100013
007744 032737 010000 000562
007752 001407
007754 042737 100000 000660
007762 013703 000660
007766 160300
007770 005200
007772 012737 024767 000652
010000 005037 000676
010004 032737 010000 000562
010012 001407
010014 005737 000564
010016 001407
010018 005237 000676
010020 005200
010022 013777 000556 170460
010024 012777 033472 170450
010026 012737 010000 000562
010028 001407
010030 013703 000556
010032 005200
010034 012737 000020 000552
010036 001407
010038 005200
010040 005737 170412
010102 012737 000076 000672
010110 000403

:READ ROUTINE:
:THIS ROUTINE PERFORMS THE READ OPERATION DETERMINED
:BY THE READ SEQUENCE ROUTINE ONE RECORD AT A TIME.
:AT THE END OF EACH READ OPERATION THE STATUS REGISTER
:IS SCANNED FOR EITHER END OF TAPE OR BEGINNING OF TAPE.
:IF EOT WAS REACHED, CONTROL WILL BE PASSED TO
:THE EOT SUBROUTINE TO REWIND THE UNIT AND FLAG IT
:UNAVAILABLE UNTIL ALL UNITS HAVE REACHED EOT.
:IF BOT WAS REACHED AN ERROR IS PRINTED AND THE
:PROGRAM WILL HALT. TESTING MAY BE RESUMED BY PRESSING
:THE CONTINUE SWITCH.
:IF A TAPE MARK IS EXPECTED (TM=1) THEN THE
:READ ROUTINE EXPECTS THE FIRST RECORD OF A
:READ REVERSE TO BE A TM, AND THE LAST RECORD
:OF A READ FORWARD TO BE A TM. REMEMBER
:THAT THE TM ADDS ONE (1) TO THE TCAL NUMBER
:OF RECORDS IN A BLOCK.
:CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13) DETERMINE WHETHER
:OR NOT TO CHECK FOR STATUS ERRORS (11) OR DATA ERRORS (13),
:CONSOLE SWITCH FIVE (5) IS USED TO CAUSE A CONTINUOUS
:READ AND SPACE (FORWARD OR REVERSE) OF THE CURRENT
:RECORD ON TAPE (YOZZLE).

READ: MOV RCNT,R0 :LOAD REC CNTR
TST EOTREC :SEE IF EOT
BPL RDA :IF NOT: BR
BIT #10000,RDCMD :SEE IF READ FORWARD
BEQ RDA :IF SO: BR
BIC #10000,EOTREC :CLEAR FLAG
MOV EOTREC,R3 :GET MODIFIED RECORD COUNT
SUB R3,R0 :SET RECORD AT
INC R0 :SET TO PROPER NUMBER OF RECORDS
RDA: MOV #MSG6,EMADDR :SET ERROR MSG ADDRESS
CLR TMFLG
BIT #10000,RDCMD
BEQ RDO :IF READ FORWARD: BR
TST TMEX :SEE IF TM
BEQ RDO :IF NOT: BR
INC TMFLG :SET TM FLAG
INC R0
RDO: MOV FMCNT,@FC :LOAD CHAR CNTR
MOV #RDATA,@BA :LOAD DATA ADDR
BIT #10000,RDCMD :SEE IF READ REVERSE
BEQ RDA :IF NOT: BR
MOV FMCNT,R3
COM R3
BIT #20,UDES :SEE IF CORE DUMP
BEQ RD1 :IF NOT: BR
CLC
ROR R3 :R3 = FC/2
ADD R3,@BA :SET REVERSE BUS ADDRESS
MOV #76,MC1 :SET READ REVERSE
BR RD1B

```

2706 0101112 012737 000070 000672 RD1A: MOV #70,MTC1 ;SET READ FORWARD
2707 0101120 012737 010132 000662 RD1B: MOV #RD2,RTN ;SET INTERRUPT RETURN ADDRESS
2708 0101126 000137 021216 RD1D: JMP TAPG ;GO EXECUTE TAPE COMMAND
2709 0101132 032737 010000 000562 RD2: BIT #10000,RDCMD ;SEE IF READ REVERSE
2710 0101140 001024 RD3 BNE RD3 ;IF SO: BR
2711 0101146 032777 000020 170352 BIT #20,ADS ;AWAIT SWDN
2712 0101152 001140 RD2B: BEQ RD2B ;AWAIT TUR
2713 0101158 032777 000020 170342 RD2A: BIT #20,ADS ;SEE IF EOT
2714 0101164 001157 RD2B: BNE RD2A ;IF NOT: BR
2715 0101170 032777 002000 170332 RD2B: BIT #2000,ADS ;SEE IF TM
2716 0101176 001141 BEQ RD3 ;IF SO: BR
2717 0101182 001172 000676 IST TMFLG ;SEE IF TM
2718 0101188 001100 BNE RD3 ;IF SO: BR
2719 0101194 010035 000660 MOV R0,EOTREC
2720 0101200 032737 100000 000660 RD3: BIS #10000,EOTREC ;SET EOT FLAG
2721 0101206 001141 000002 170302 RD3: BIT #2,ADS ;SEE IF AT LOAD POINT
2722 0101212 001172 IF NOT: BR
2723 0101218 001172 022736 JSR PC,PAPRT ;PRINT CYCLE NUMBER
2724 0101224 012704 025166 MOV #MSG22,R4
2725 0101230 104000 TTOUTT ;PRINT BOT ERROR
2726 0101236 104006 STOPP
2727 0101242 000137 003160 JMP STARTA ;RESTART
2728 0101248 032777 004000 170336 RD4: BIT #4000,ASWR ;SEE IF SHOULD CHECK ERRORS
2729 0101254 001121 BNE RD5 ;IF NOT: BR
2730 0101260 001121 000676 YST TMFLG ;IF NO TM EXPT: BR
2731 0101266 001472 BEQ RD4B ;IF TM RECVD: BR
2732 0101272 032777 000004 170234 BIT #4,ADS ;SAVE EXPT BUS ADDRESS
2733 0101278 001024 BNE RD4A ;SET TM STATUS ERROR FLAG
2734 0101284 012737 033472 021132 MOV #RDATA,CADER ;GO PRINT TM ERROR
2735 0101290 012737 000002 021140 MOV #2,DRVER
2736 0101296 004737 020234 JSR PC,ERPT
2737 0101302 013704 000674 MOV UNP,R4
2738 0101308 032737 010000 000562 BIT #10000,RDCMD ;SEE IF READ REVERSE
2739 0101314 001140 BEQ 1$ ;IF NOT: BR
2740 0101320 005264 001150 INC RDERR1(R4) ;BUMP READ REVERSE ERROR
2741 0101326 000502 BR RD6
2742 0101332 005264 001110 1$: INC RDER1(R4) ;BUMP READ FORWARD ERROR
2743 0101338 000477 BR RD6
2744 0101344 012703 033472 RD4A: MOV #RDATA,R3
2745 0101350 032737 010000 000562 BIT #10000,RDCMD ;SEE IF READ REVERSE
2746 0101356 001007 BNE RD4A0 ;IF SO: BR
2747 0101362 032737 002000 000552 BIT #2000,UDES ;SEE IF IN PE
2748 0101368 001025 BNE RD4A2 ;IF SO: BR
2749 0101374 0062703 ADD #2,R3
2750 0101380 001422 BR RD4A2
2751 0101386 013704 000556 RD4A0: MOV FMCNT,R4
2752 0101392 005264 COM R4
2753 0101398 032737 000020 000552 BIT #20,UDES ;SEE IF CORE DUMP
2754 0101404 001140 BEQ RD4A1 ;IF NOT: BR
2755 0101410 000241 CLC
2756 0101416 006004 ROR R4 ;SET TO FC/2
2757 0101422 006003 RD4A1: ADD R4,R3 ;SET EXPT BUS ADDRESS
2758 0101428 042703 000001 BIC #1,R3 ;MAKE EXPT ADDRESS EVEN
2759 0101434 032737 002000 000552 BIT #2000,UDES ;SEE IF IN PE
2760 0101440 001002 BNE RD4A2 ;IF SO: BR
2761 0101446 162703 000002 SUB #2,R3
    
```

2762	010436	004737	017504		RD4A2:	JSR	PC,ER2		
2763	010442	000402				BR	RD4C		
2764	010445	004737	017406		RD4B:	JSR	PC,ERCHK	:GO CHECK ERRORS	
2765	010450	005737	000706		RD4C:	TST	SERFL		
2766	010456	001417				BEQ	RD5	:IF NO ERROR: BR	
2767	010456	013704	000674			MOV	UNP,R4		
2768	010462	032737	010000	000562		BIT	#10000,RDCMD	:SEE IF READ REVERSE	
2769	010470	001003				BNE	RD4D	:IF SO: BR	
2770	010472	005267	001110			INC	RDER1(R4)	:BUMP READ FORWARD ERROR	
2771	010476	000402				BR	RD4E		
2772	010480	005267	001150		RD4D:	INC	RDERR1(R4)	:BUMP READ REVERSE ERROR	
2773	010484	004737	010706		RD4E:	JSR	PC,RDRTY	:GO RETRY	
2774	010490	005037	000712			CLR	RTYFL	:CLEAR RETRY FLAG	
2775	010494	032777	020000	170064	RD5:	BIT	#20000,@SWR	:SEE IF SHOULD DO DATA CHECK	
2776	010498	001003				BNE	RD6	:IF NOT: BR	
2777	010504	005737	000676			TST	TMFLG		
2778	010508	001002				BNE	RD6		
2779	010514	004737	015544			JSR	PC,DCHK	:GO CHECK DATA	
2780	010520	005037	000706		RD6:	CLR	SERFL	:CLEAR STATUS ERROR FLAG	
2781	010524	004737	014352			JSR	PC,DS3	:CLEAR BUFFER	
2782	010528	032777	000040	170032		BIT	#40,@SWR	:SEE IF SHOULD YOZZLE	
2783	010534	001402				BEQ	RD7	:IF NOT: BR	
2784	010536	004737	011274			JSR	PC,YOZ	:ELSE GO YOZZLE	
2785	010542	013737	000572	000666	RD7:	MOV	RSTAL,STAL	:SET DELAY	
2786	010570	004737	012304			JSR	PC,STALL	:STALL	
2787	010574	032737	010000	000562		BIT	#10000,RDCMD	:SEE IF READ REVERSE	
2788	010602	001403				BEQ	RD7A	:IF NOT: BR	
2789	010604	005037	000676			CLR	TMFLG	:CLEAR TAPE MARK FLAG	
2790	010610	000405				BR	RD10		
2791	010612	005737	000660		RD7A:	TST	EOTREC	:SEE IF EOT FOUND	
2792	010616	100002				BPL	RD10	:IF NOT: BR	
2793	010620	012700	000001			MOV	#1,R0	:SET TO EOT	
2794	010624	005300			RD10:	DEC	R0		
2795	010626	001402				BEQ	RD11	:IF DONE ALL: BR	
2796	010630	000137	010030			JMP	RDO		
2797	010634	032737	010000	000562	RD11:	BIT	#10000,RDCMD	:SEE IF READ REVERSE	
2798	010642	001016				BNE	RDEX	:IF SO: BR	
2799	010644	005737	000660			TST	EOTREC	:SEE IF FOUND EOT	
2800	010650	100413				BMI	RDEX	:IF SO: BR	
2801	010652	005737	000564			TST	TMEX	:SEE IF TM EXPECTED	
2802	010656	001410				BEQ	RDEX	:IF NOT: BR	
2803	010660	005737	000676			TST	TMFLG	:SEE IF TM FOUND	
2804	010664	001005				BNE	RDEX	:IF SO: BR	
2805	010666	005237	000676			INC	TMFLG	:ELSE SET FLAG	
2806	010672	005200				INC	R0	:SET RECORD COUNT TO ONE	
2807	010674	000137	010030			JMP	RDO	:GO READ TM	
2808	010700	005037	000676		RDEX:	CLR	TMFLG		
2809	010704	000207			RDX:	RTS	PC	:EXIT	

2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865

010706 032777 000020 167672
010714 001001
010716 000207
010720 013703 000722
010724 042703 102700
010730 001410
010732 004737 022736
010736 012704 026612
010742 104000
010744 004737 011260
010750 000207
010752 032777 002000 167626
010760 001003
010762 012704 026265
010764 104000
010770 005037 000702
010774 005037 000706
011000 012737 000002 000712
011006 004737 011274
011012 005737 000706
011016 001031
011020 032777 002000 167560
011026 001011
011030 012704 026744
011034 104000
011036 012704 026307
011042 104000
011044 013703 000702
011050 104002
011052 013704 000674
011056 032737 010000 000562
011064 001003
011066 005264 002670
011072 000402
011074 005264 002710
011100 000207
011102 005037 000646
011106 013703 000722
011112 042703 102700
011116 001413
011120 004737 022736
011124 012704 026612
011130 104000
011132 004737 011260
011136 012737 000001 000646

```
*****  
:READ ERROR RETRY SUBROUTINE:  
:THIS SUBROUTINE WILL RETRY ALL DATA RELATED  
:READ ERRORS UP TO EIGHT (8) TIMES. IF ALL  
:FOUR RETRIES ARE BAD, IT IS CONSIDERED  
:A HARD ERROR. IF ANY ARE GOOD, IT IS A  
:SOFT ERROR. RETRIES MAY BE INHIBITED  
:VIA SWITCH FOUR (SW4=0: INHIBIT RETRIES)  
*****  
RDRTY: BIT #20,@SWR :SEE IF RETRY INHIBITED  
BNE RDRT0 :IF NOT: BR  
RTS PC :ELSE RETURN  
RDRT0: MOV ERSAV,R3  
BIC #102700,R3 :MARK NON-RECOVERABLE ERROR BITS  
BEQ RDRT1 :IF NOT: BR  
JSR PC,PAPRT :PRINT HEADER  
MOV #MSG79,R4  
TTOUTT :PRINT NON-RECOVERABLE MESSAGE  
JSR PC,NRTP :PRINT ER FOR NON-RETRYABLE ERROR  
RTS PC :RETURN  
RDRT1A: BIT #2000,@SWR :SEE IF PRINT INHIBITED  
RDRT1: BNE RDRT1B :IF SO: BR  
MOV #MSG64,R4  
TTOUTT :PRINT ORIGINAL ERROR TAG  
RDRT1B: CLR RTCNT :CLEAR RETRY COUNTER  
RDRTG: CLR SERFL :CLEAR STATUS ERROR FLAG  
MOV #2,RTYFL :SET READ RETRY FLAG  
JSR PC,YOZ :GO TO YOZZLE TO RETRY READ  
TST SERFL :SEE IF RETRY ERROR  
BNE RDRT5 :IF SO: BR  
RDRT2: BIT #2000,@SWR  
BNE RDRT2  
MOV #MSG105,R4  
TTOUTT :PRINT RECOVERED MESSAGE  
MOV #MSG65,R4 :PRINT RETRY TAG  
MOV RTCNT,R3 :PRINT RETRY NUMBER  
OCTPP :PRINT RETRY NUMBER  
RDRT2: MOV UNP,R4  
BIT #10000,RDCMD :SEE IF READ REVERSE  
BNE RDRT3 :IF SO: BR  
INC RFSOFT(R4) :ELSO BUMP FORWARD SOFT ERROR COUNTER  
BR RDRT4  
RDRT3: INC RRSOFT(R4) :BUMP ERRORS SOFT CNTR  
RDRT4: RTS PC :RETURN  
RDRT5: CLR TEMP3 :++G CLEAR RECOVERABLE ERROR INDICATOR  
MOV ERSAV,R3 :GET ER  
BIC #102700,R3 :MASK RECOVERABLE BITS  
BEQ RDRT5A :IF RECOVERABLE: BR  
JSR PC,PAPRT :PRINT HEADER  
MOV #MSG79,R4  
TTOUTT :PRINT NON-RECOVERABLE MSG  
JSR PC,NRTP :PRINT ER  
MOV #1,TEMP3 :SET FLAG
```

2866	C11144	000404								
2867	011146	032777	002000	167432	RDRT5A:	BR	RDRT5B			
2868	011154	001014				BIT	#2000,@SWR		:SEE IF PRINT INHIBITED	
2869	011156	012704	026307		RDRT5B:	BNE	RDRT6		:IF SO: BR	
2870	011162	104000				MOV	#MSG65,R4			
2871	011164	013703	000702			TTOUTT			:PRINT RETRY TAG	
2872	011170	104003				MOV	RTCNT,R3			
2873	011172	005737	000646			OCTPP			:PRINT PTRY NUMBER	
2874	011176	001403				TST	TEMP3		:SEE IF ID NON-RECOVERABLE	
2875	011200	005037	000646			BEQ	RDRT6		:IF NOT: BR	
2876	011204	000207				CLR	TEMP3		:CLEAR FLAG	
2877	011206	005237	000702		RDRT6:	RTS	PC		:EXIT	
2878	011212	023737	000702	000602		INC	RTCNT			
2879	011220	001265				CMP	RTCNT,RETRY		:SEE IF DONE 8 RETRIES	
2880	011222	012704	027277			BNE	RDRT6		:IF NOT: BR	
2881	011226	104000				MOV	#MSG115,R4			
2882	011230	013704	000674			TTOUTT			:PRINT HARD ERROR MESSAGE	
2883	011234	032737	010000	000562		MOV	UNP,R4			
2884	011242	001003				BIT	#10000,RDCMD		:SEE IF READ REVERSE	
2885	011244	005264	002730			BNE	RDRT7		:IF SO: BR	
2886	011250	000402				INC	RFHARD(R4)		:BUMP FORWARD HARD ERROR CNTR	
2887	011252	005264	002750		RDRT7:	BR	RDRTX			
2888	011256	000207			RDRTX:	INC	RRHARD(R4)		:BUMP REVERSE HARD ERROR CNTR	
2889						RTS	PC		:RETURN	
2890	011260	013703	000722		NRTP:	MOV	ERSAV,R3		:GET ER REGISTER	
2891	011264	104002				OCTPP			:PRINT ER	
2892	011266	004737	021156			JSR	PC,FRPRT		:PRINT F OR R	
2893	011272	000207				RTS	PC		:RETURN	

2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949

011274 012777 000001 167306
011302 013737 000600 000666
011310 004737 012304
011314 012777 177777 167174
011322 032737 010000 000562
011330 001404
011332 112737 000030 000672
011340 000403
011342 112737 000032 000672
011350 012737 011370 000662
011356 012737 177775 000666
011364 000137 021216
011370 005737 000676
011374 001404
011376 012737 040000 000666
011404 000403
011406 013737 000600 000666
011414 004737 012304
011420 012777 033472 167066
011426 032737 010000 000562
011434 001417
011436 013703 000556
011442 005103
011444 032737 000020 000552
011452 001402
011457 000241
011456 006003
011460 060377 167030
011464 012737 000076 000672
011472 000403
011474 012737 000070 000672
011502 013777 000556 167006
011510 012737 011522 000662
011516 000137 021216
011522 032777 004000 167056
011530 001051
011532 005737 000676
011536 001444
011540 032737 010000 000562
011546 001426
011550 012703 033472
011554 013704 000556
011560 005104

```
*****  
:YOZZLE SUBROUTINE:  
*****  
:THIS SUBROUTINE, ENTERED VIA SWITCH FIVE (5), IS USED TO PERFORM  
:A CONTINUOUS READ AND SPACE OVER OF THE CURRENT RECORD ON TAPE.  
:FULL STATUS AND DATA CHECKING MAY BE PERFORMED  
:OR NOT VIA CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13).  
:A SOFTWARE DELAY IS PERFORMED BETWEEN EACH READ  
:AND SPACE OPERATION AND MAY BE VARIED BY TYPING  
:CNTRL C ON THE TTY AND ENTERING A VALUE IN RESPONSE  
:TO THE PRINTED REQUEST.  
*****  
YOZ:  MOV #1,@TKS ;SET TTY ENABLE  
      MOV YSTAL,STAL  
      JSR PC,STALL ;DO YOZZLE STALL  
YOZO:  MOV #-1,@FC ;SET TO 1 RECORD SPACING  
      BIT #10000,RDCMD ;SEE IF READ REVERSE  
      BEQ YOZA ;IF NOT: BR  
      MOVB #30,MTC1 ;SET TO SPACE FORWARD  
      BR YOZB  
YOZA:  MOVB #32,MTC1 ;SET TO SPACE REVERSE  
YOZB:  MOV #YOZC,RTRN ;SET RETURN ADDRESS  
      MOV #177775,STAL ;SET TIME MULTIPLIER  
      JMP TAPG ;GO YOZZLE  
YOZC:  TST TMFLG ;SEE IF TM  
      BEQ 1$ ;IF NOT: BR  
      MOV #40000,STAL ;SET TM STALL  
      BR 2$  
1$:  MOV YSTAL,STAL  
2$:  JSR PC,STALL ;DO YOZZLE STALL  
      MOV #RDATA,@BA ;SET BUS ADDRESS  
      BIT #10000,RDCMD ;SEE IF READ REVERSE  
      BEQ YOZC1 ;IF NOT: BR  
      MOV FMCNT,R3  
      COM R3  
      BIT #20,UDES ;SEE IF CORE DUMP  
      BEQ YOZC0 ;IF NOT: BR  
      CLC  
      ROR R3 ;R3 = FC/2  
YOZC0:  ADD R3,@BA ;SET REVERSE BUS ADDRESS  
      MOV #76,MTC1 ;SET READ REVERSE  
      BR YOZC2  
YOZC1:  MOV #70,MTC1 ;SET READ FORWARD  
YOZC2:  MOV FMCNT,@FC ;SET CHARACTER COUNT  
      MOV #YOZD,RTRN ;SET RETURN ADDRESS  
      JMP TAPG ;GO READ  
YOZD:  BIT #4000,@SWR ;SEE IF SHOULD CHECK ERRORS  
      BNE YOZE ;IF NOT: BR  
      TST TMFLG ;SEE IF TAPE MARK TIME  
      BEQ YOZD1 ;IF NOT: BR  
      BIT #10000,RDCMD ;SEE IF READ REVERSE  
      BEQ YOZD0 ;IF NOT: BR  
      MOV #RDATA,R3  
      MOV FMCNT,R4  
      COM R4
```

```

2950 011562 032737 000020 000552 BIT #20,UDES ;SEE IF CORE DUMP
2951 011570 001402 BEQ YOZD4 ;IF NOT: BR
2952 011572 000241 CLC
2953 011574 006004 ROR R4 ;SET TO FC/2
2954 011576 060403 YOZD4: ADD R4,R3 ;SET EXPT BUS ADDRESS
2955 011600 042703 000001 BIC #1,R3 ;MAKE EXPT ADDRESS EVEN
2956 011604 032737 002000 000552 BIT #2000,UDES ;SEE IF PE
2957 011612 001001 BNE YOZD2 ;IF SO: BR
2958 011614 005743 TST -(R3) ;SET EXPT BA
2959 011616 004737 017504 YOZD2: JSR PC,ER2 ;GO CHECK ERRORS
2960 011622 000430 BR YOZF
2961 011624 012703 033472 YOZD0: MOV #RDATA,R3
2962 011630 032737 002000 000552 BIT #2000,UDES ;SEE IF PE
2963 011636 001001 BNE YOZD3 ;IF SO: BR
2964 011640 005723 TST (R3)+ ;SET EXPT BA
2965 011642 004737 017504 YOZD3: JSR PC,ER2 ;GO CHECK ERRORS
2966 011646 000416 BR YOZF
2967 011650 004737 017406 YOZD1: JSR PC,ERCHK ;ELSE GO CHECK ERRORS
2968 011654 032777 020000 166724 YOZE: BIT #2000,@SWR ;SEE IF SHOULD CHECK DATA
2969 011662 001010 BNE YOZF ;IF NOT: BR
2970 011664 005737 000676 TST TMFLG ;SEE IF TAPE MARK
2971 011670 001005 BNE YOZF ;IF SO: BR
2972 011672 005737 000712 TST RTYFL ;SEE IF RETRY
2973 011676 001004 BNE YOZF0 ;IF SO: BR
2974 011700 004737 015544 JSR PC,DCHK ;ELSE GO CHECK DATA
2975 011704 004737 014352 YOZF: JSR PC,DS3 ;GO CLEAR DATA AREA
2976 011710 105777 166674 YOZF0: TSTB @TKS ;SEE IF HAVE NEW STALL VALUE
2977 011714 100032 BPL YOZG ;IF NOT: BR
2978 011716 122777 000203 166666 CMPB #203,@TKB ;SEE IF CONT C
2979 011724 001026 BNE YOZG ;IF NOT: BR
2980 011726 012704 025767 MOV #MSG44,R4
2981 011732 104000 TTOUTT ;PRINT YSTALL REQUEST
2982 011734 013703 00C600 MOV YSTAL,R3
2983 011740 104002 OCTPP ;PRINT PRESENT STALL
2984 011742 010037 000646 MOV R0,TEMP3 ;SAVE R0(REC CNT)
2985 011746 012705 000600 MOV #YSTAL,R5 ;SET ADDRESS OF YSTL
2986 011752 012701 000006 MOV #6,R1 ;SET NUMBER OF CHAR TO INPUT
2987 011756 012702 177777 MOV #-1,R2 ;SET MAXIMUM LIMIT
2988 011762 012703 002000 MOV #2000,R3 ;SET MINIMUM LIMIT
2989 011766 004737 023354 JSR PC,TTR ;GO GET VALUE
2990 011772 013700 000646 MOV TEMP3,R0 ;RESTORE R0(REC CNTR)
2991 011776 000137 011274 JMP YOZ ;RESTART YOZZLE
2992 012002 122777 000207 166602 YOZG: CMPB #207,@TKB ;CHECK FOR CNTL G
2993 012010 001010 BNE YOZI
2994 012012 022737 000176 000606 CMP #SWREG,SWR ;IS SWREG SELECTED
2995 012014 001004 BNE YOZI
2996 012016 005077 166564 CLR @TKB ;CLEAR CNTL G OUT OF BUFFER
2997 012018 004737 024452 JSR PC,CNTG ;GO CHANGE SWREG
2998 012022 032777 000040 166546 YOZI: BIT #40,@SWR ;SEE IF SHOULD CONTINUE YOZZLE
2999 012024 001402 BEQ YOZH ;IF NOT: BR
3000 012026 000137 011314 JMP YOZO
3001 012028 012777 000100 166534 YOZH: MOV #100,@TKS ;SET TTY INTERRUPT ENABLE
3002 012034 000207 RTS PC ;EXIT
    
```

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

: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.

```
012056 013737 000576 000666 BKSP: MOV TSTAL,STAL
012064 004737 012304 JSR PC,STALL ;DO TURN AROUND STALL
012070 012737 025017 000652 MOV #MSG10,FMADDR
012076 012777 033472 16641C P / #RDATA,@BA
012104 005737 000564 TST TMEX ;SEE IF TM
012110 001440 BEQ B0 ;IF NOT: BR
012112 012777 177777 166376 MOV #-1,@FC
012120 012737 000032 000672 MOV #32,MTC1
012126 012737 012140 000662 MOV #BKTM,RTRN
012134 000137 021216 JMP TAPG ;SPACE TO TM
012140 032777 010000 166440 BKTM: BIT #10000,@SWR ;SEE IF SHOULD CHECK ERROR
012150 001021 BNE B0 ;IF NOT: BR
012156 012737 026201 000652 MOV #MSG55,EMADDR
012164 032777 000004 166336 BIT #4,@DS ;SEE IF TM
012172 001006 BNE BK1M0 ;IF SO: BR
012178 012737 033472 021132 MOV #RDATA,CADER
012186 004737 020234 JSR PC,ERP1 ;PRINT ERROR
012194 000404 BR B0
012200 012703 033472 BK1M0: MOV #RDATA,R3
012206 004737 017504 JSR PC,ER2
012212 013700 000554 B0: MOV RCNT,R0
012218 005100 COM R0 ;BUILD SPACE AMOUNT
012224 005200 INC R0
012230 012737 025017 000652 MOV #MSG10,EMADDR ;SET ERROR MESSG ADDRESS
012236 010077 166262 RO,@FC
012242 012737 000032 000672 BKRT: MOV #32,MTC1 ;SET SPACE REVERSE
012248 012737 012260 000662 MOV #B1,RTRN ;SET RETURN ADDRESS
012254 010037 000666 MOV R0,STAL ;SET INTERRUPT TIME MULTIPLIER
012260 030137 021216 JMP TAPG ;GO DO SPACE
012266 012703 033472 B1: MOV #RDATA,R3
012272 004737 017504 JSR PC,ER2
012278 013737 000576 000666 B2: MOV TSTAL,STAL ;DO STALL
012284 004737 012304 JSR PC,STALL ;STALL
012302 000207 RTS PC ;EXIT
```


3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077

```
.....  
: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 250(8) OR MAY BE MODIFIED  
:AT ANY TIME BY ENTERING CNTRL C ON THE TTY AND  
:INSERTING NEW VALUES IN RESPONSE TO THE REQUEST.  
:THE READ STALL AND THE WRITE STALL ARE DELAYS  
:EXECUTED BETWEEN EACH RECORD OF THE DATA BLOCK.  
:THE TURN AROUND STALL IS EXECUTED EACH TIME  
:THE DIRECTION OF TAPE MOVEMENT IS CHANGED AND  
:ALSO EACH TIME THE TAPE OPERATION CHANGES FROM  
:WRITE TO READ OR READ TO WRITE. THE YOZZLE  
:STALL IS EXECUTED ONLY DURING THE YOZZLE ROUTINE.  
.....
```

012304 005337 000666
012310 001375
012312 000207

STALL: DEC STAL
 3NE STALL :DELAY
 RTS PC :EXIT

3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115

012314	012701	177760	
012320	012702	174000	
012324	004737	023322	
012330	042737	000001	000626
012336	013737	000626	000556
012344	012737	177777	014406
012352	000207		

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

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

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

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

3116
3117
3118
3119
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
3166
3167
3168
3169
3170
3171

```

*****
: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 ^ SO 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.
*****

```

```

012400 005737 000634
012404 001001
012406 000207
012410 005037 000574
012414 005037 005156
012420 012700 000010
012424 012701 000746
012430 005021
012434 005300
012437 001375

```

```

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

```

3172	012436	012704	025461	MOV	#MSG31,R4	
3173	012442	005737	000734	TST	ASEQF	:SEE IF AUTO SEQ
3174	012446	001402		BEQ	TINPB1	:IF NOT: BR
3175	012450	012704	025407	MOV	#MSG30,R4	:SET AUTO SEQ HDR
3176	012454	104000		TINPB1: TTOUTT		:PRINT PROGRAM NAME
3177	012456	005737	014150	TST	SCVFL	:SEE IF SHORT CONVERSATION
3178	012462	001067		BNE	TINPC	:IF SO: BR
3179	012466	012704	026471	MOV	#MSG74,R4	
3180	012470	104000		TTOUTT		:REQUEST STARTING REGISTER ADDRESS
3181	012472	013703	000544	MOV	REGS,R3	
3182	012476	104002		OCTPP		:PRINT CURRENT REG START
3183	012480	012703	000544	MOV	#REGS,R5	:SAVE ADDRESS LOCATION
3184	012484	012701	000006	MOV	#6,R1	:SET SIZE OF ENTRY
3185	012488	012701	176400	MOV	#176400,R2	:SET UPPER LIMIT
3186	012492	012702	172300	MOV	#172300,R3	:SET LOWER LIMIT
3187	012496	004737	023354	JSR	PC,TTR	:GO GET RESPONSE
3188	012500	012704	026514	MOV	#MSG75,R4	
3189	012504	104000		TTOUTT		:GO REQUEST VECTOR ADDRESS
3190	012508	013703	000546	MOV	VECT,R3	
3191	012512	104002		OCTPP		:PRINT CURRENT VECTOR
3192	012516	012703	000546	MOV	#VECT,R5	:SET SAVE LOCATION
3193	012520	012701	000003	MOV	#3,R1	:SET SIZE OF ENTRY
3194	012524	012702	000224	MOV	#224,R2	:SET UPPER LIMIT
3195	012528	012703	000150	MOV	#150,R3	:SET LOWER LIMIT
3196	012532	004737	023354	JSR	PC,TTR	:GO GET RESPONSE
3197	012536	013700	000546	MOV	VECT,R0	:GET VECTOR ADDRESS
3198	012540	012720	021714	MOV	#MTINT,(R0)+	:LOAD VECTOR WITH HANDLER ADDRESS
3199	012544	012710	000340	MOV	#340,(R0)	:LOAD PRIORITY LEVEL
3200	012548	012700	000544	MOV	REGS,R0	:GET STARTING REGISTER ADDRESS
3201	012552	012701	000016	MOV	#16,R1	:SET NUMBER OF REGISTERS
3202	012556	012702	000510	MOV	#C1,R2	:GET FIRST ADDRESS LOCATION
3203	012560	010022		TINPB0: MOV	R0,(R2)+	:BUILD TABLE OF ADDRESSES
3204	012564	004737	000002	ADD	#2,R0	:BUMP ADDRESS
3205	012568	001350		DEC	R1	:SEE IF DONE
3206	012572	001353		BNE	TINPB0	:IF NOT: BR
3207	012576	005737	000734	TST	ASEQF	:SEE IF AUTO SEQ
3208	012580	001403		BEQ	TINPC	:IF NOT: BR
3209	012584	005726		TST	(SP)+	:RESET STACK POINTER
3210	012588	000137	021732	JMP	ASEQ	:GO TO AUTO SEQUENCE
3211	012592	012777	000040	165650 TINPC: MOV	#40,@CS	:INITIALIZE
3212	012596	012704	026135	MOV	#MSG52,R4	
3213	012600	104000		TTOUTT		:REQUEST DRIVE NUMBER
3214	012604	012705	000550	MOV	#DVN,R5	:GET ADDRESS
3215	012608	012701	000001	MOV	#1,R1	:SET SIZE OF RESPONSE
3216	012612	012702	000007	MOV	#7,R2	:SET UPPER LIMIT
3217	012616	012703	000000	MOV	#0,R3	:SET LOWER LIMIT
3218	012620	004737	023354	JSR	PC,TTR	:GO GET DRIVE NUMBER
3219	012624	013777	000550	165610 MOV	DVN,@CS	
3220	012628	005777	165574	TST	@C1	:ACCESS DRIVE
3221	012632	005777	010000	165576 BIT	#10000,@CS	:SEE IF NED
3222	012636	001411		BEQ	TINP0	:IF NOT: BR
3223	012640	012704	026426	MOV	#MSG71,R4	
3224	012644	104000		TTOUTT		:PRINT NED
3225	012648	013704	000510	MOV	C1,R4	
3226	012652	005204		INC	R4	
3227	012656	152714	000100	BISB	#100,(R4)	:CLEAR TRE

3228	012744	000736			BR	TINPC		:++G RETRY DVN
3229	012746	012704	025523	TINPO:	MOV	#MSG32,R4		
3230	012752	104000			TTOUTT			:PRINT UNIT NUMBER REQUEST
3231	012754	005037	000644		CLR	TEMP2		:CLEAR BUFFER
3232	012760	012703	000644		MOV	#TEMP2,R5		:SET UNIT DESCRIPTION BUFFER ADDRESS
3233	012762	012701	000001		MOV	#1,R1		:SET NUMBER OF CHARACTERS TO INPUT
3234	012764	012701	000007		MOV	#7,R2		:SET MAXIMUM LIMIT
3235	012770	012701	000000		MOV	#0,R3		:SET MINIMUM LIMIT
3236	012772	004737	000000		JSR	PC,TTR		:GO GET UNIT NUMBER
3237	013000	005737	023354		TST	TEMP1		:SEE IF HAVE NEW PARAMETER
3238	013010	001013	000642		BNE	TINPOB		:IF SO: BR
3239	013012	005737	000674		TST	UNP		:SEE IF FIRST ENTRY
3240	013016	001001			BNE	TINPOA		:IF NOT: BR
3241	013020	000752			BR	TINPO		:++G ELSE RETRY
3242	013022	013700	000674	TINPOA:	MOV	UNP,R0		
3243	013026	012760	177777	000746	MOV	#-1,UN1(R0)		:SET END UNIT TABLE
3244	013030	000137	013424		JMP	TINP2C		:GO GET RECORD COUNT
3245	013034	013700	000674	TINPOB:	MOV	UNP,R0		
3246	013040	042760	000007	000746	BIC	#7,UN1(R0)		:CLEAR UNIT NUMBER
3247	013042	004737	014164		JSR	PC,TPOS1		:GO LOAD UNIT NUMBER TO PROPER POSITION
3248	013046	012777	000040	165434	MOV	'40,@CS		
3249	013052	013777	000550	165426	MOV	'N,@CS		
3250	013072	016077	000746	165442	MOV	(R0),@C2		:LOAD UNIT NUMBER
3251	013080	012777	002000	165430	TINPOC:	BIT	-2000,@T	:SEE IF SLAVE PRESENT
3252	013100	001005			BNE	TINPOD		:IF SO: BR
3253	013110	012704	026214		MOV	#MSG57,R4		
3254	013112	104000			TTOUTT			:PRINT NON-EXIST SLAVE
3255	013116	000137	012746		JMP	TINPO		:REDO
3256	013122	022777	142011	165406	TINPOD:	CMP	#142011,@T	:++G SEE IF 9TRK TM02,TU16/TE16
3257	013130	001406			BEQ	TINPOE		:IF SO: BR
3258	013132	012704	026110		MOV	#MSG50,R4		:ILLEGAL DRIVE TYPE
3259	013136	104000			TTOUTT			:GO PRINT
3260	013140	017703	165372		MOV	@DT,R3		
3261	013144	104002			OCTPP			:PRINT DRIVE TYPE REGISTER
3262	013146	012704	025011		TINPOE:	MOV	#MSG9,R4	
3263	013152	104000			TTOUTT			:PRINT SERIAL NUMBER TAG
3264	013154	017703	165360		MOV	@SN,R3		
3265	013156	004737	024300		JSR	PC,SNPT		:PRINT SERIAL NUMBER
3266	013160	012704	025544		TINP1:	MOV	#MSG33,R4	
3267	013162	104000			TTOUTT			:PRINT DENSITY REQUEST
3268	013164	005037	000644		CLR	TEMP2		:CLEAR BUFFER
3269	013170	012701	000001		MOV	#1,R1		:SET NUMBER OF CHARACTERS TO INPUT
3270	013172	012702	000007		MOV	#7,R2		:SET MAXIMUM LIMIT
3271	013174	012703	000000		MOV	#0,R3		:SET MINIMUM LIMIT
3272	013176	004737	023354		JSR	PC,TTR		:GO GET DENSITY
3273	013178	005737	000642		TST	TEMP1		:SEE IF HAVE NEW PARAMETER
3274	013180	001406			BEQ	TINP2		:IF NOT: BR
3275	013182	022737	003400	000552	BIC	#3400,UDES		:ELSE CLEAR OLD PARAMETER
3276	013184	012703	000010		MOV	#10,R3		:SET POSITION FACTOR
3277	013186	004737	014152		JSR	PC,TPOS		:GO LOAD DENSITY INTO PROPER POSITION
3278	013190	012704	025560		TINP2:	MOV	#MSG34,R4	
3279	013192	104000			TTOUTT			:PRINT PARITY REQUEST
3280	013194	005037	000644		CLR	TEMP2		:CLR BUFFER
3281	013196	012701	000001		MOV	#1,R1		:SET NUMBER OF CHARACTERS TO INPUT
3282	013198	012702	000001		MOV	#1,R2		:SET MAXIMUM LIMIT
3283	013200	012703	000000		MOV	#0,R3		:SET MINIMUM LIMIT

3284	013270	004737	023354	JSR	PC,TTR	:GO INPUT PARITY
3285	013274	005737	000642	TST	TEMP1	:SEE IF HAVE NEW PARAMETER
3286	013300	001407		BEG	TINP2A	:IF NOT: BR
3287	013300	001407	000010	BIC	#10,UDES	:ELSE CLEAR OLD PARAMETER
3288	013300	001407	000003	MOV	#3,R3	:SET POSITION FACTOR
3289	013300	001407	014155	JSR	PC,TPOS	:CO LOAD PARITY TO PROPER POSITION
3290	013300	001407	026157	TINP2A: MOV	#MSG53,R4	
3291	013300	104000		TTOUTT		:REQUEST FORMAT
3292	013300	005037	000644	CLR	TEMP2	
3293	013300	012701	000002	MOV	#2,R1	
3294	013300	012702	000016	MOV	#16,R2	
3295	013300	012702	000014	MOV	#14,R3	
3296	013300	004737	023354	JSR	PC,TTR	:GO GET FORMAT
3297	013300	005737	000642	TST	TEMP1	:SEE IF NEW PARAMETER
3298	013300	001407		BEG	TINP2B	:IF NOT: BR
3299	013300	042737	000170	BIC	#170,UDES	
3300	013300	012703	000004	MOV	#4,R3	
3301	013300	004737	014152	JSR	PC,TPOS	
3302	013300	005237	005156	TINP2B: INC	REOTC	:BUMP EOT UNIT COUNTER
3303	013300	022737	000016	CMP	#16,UNP	:SEE IF DONE UNITS
3304	013300	001407		BEG	TINP2C	:IF SO: BR
3305	013300	062737	000002	ADD	#2,UNP	:POINT TO NEXT UNIT
3306	013300	000137	012746	JMP	TINP0	:ELSE LOOK FOR NEXT UNIT
3307	013300	005037	000674	TINP2C: CLR	UNP	:CLEAR UNIT POINTER
3308	013300	005037	005156	MOV	REOTC,R0	
3309	013300	005037	005156	SWAB	REOTC	
3310	013300	005037	005156	MCVB	R0,REOTC	:SET UNIT EOT COUNTER
3311	013300	012704	025573	TINP3: MOV	#MSG35,R4	
3312	013300	104000		TTOUTT		:PRINT RECORD COUNT REQUEST
3313	013300	013300	000554	MOV	RCNT,R3	
3314	013300	012703		OCTPP		:PRINT RECORD COUNT
3315	013300	012703	000554	MOV	#RCNT,R5	:SET RECORD COUNT ADDRESS
3316	013300	012703	000006	MOV	#6,R1	:SET NUMBER OF CHARACTERS TO INPUT
3317	013300	012703	177777	MOV	#-1,R2	:SET MAXIMUM LIMIT
3318	013300	012703	000001	MOV	#1,R3	:SET MINIMUM LIMIT
3319	013300	004737	023354	JSR	PC,TTR	:GO GET RECORD COUNT
3320	013300	013300	000554	MOV	RCNT,RCSAV	:SAVE RECORD COUNT
3321	013300	012704	025614	MOV	#MSG36,R4	
3322	013300	104000		TTOUTT		:PRINT CHARACTER COUNT REQUEST
3323	013300	005037	000556	NEG	FCNT	
3324	013300	013703	000556	MOV	FCNT,R3	
3325	013300	104000		OCTPP		:PRINT CHAR COUNT
3326	013300	012703	000556	MOV	#FCNT,R5	:SET CHARACTER COUNT ADDRESS
3327	013300	012701	000006	MOV	#6,R1	:SET NUMBER OF CHARACTERS TO INPUT
3328	013300	012701	004000	MOV	#4000,R2	:SET MAXIMUM LIMIT
3329	013300	012701	000004	MOV	#4,R3	:SET MINIMUM LIMIT
3330	013300	004737	023354	JSR	PC,TTR	:GO GET CHARACTER COUNT
3331	013300	005037	000556	NEG	FCNT	:SET TO TWO'S COMPLIMENT
3332	013300	013737	000556	MOV	FCNT,FCSAV	:SAVE FRAME COUNT
3333	013300	012704	025640	MOV	#MSG37,R4	:PRINT PATTERN NUMBER REQUEST
3334	013300	104000		TTOUTT		
3335	013300	013703	000560	MOV	PATRN,R3	
3336	013300	104000		OCTPP		:PRINT PATTERN
3337	013300	005037	014554	CLR	DOFL	:CLEAR EXTERNAL DATA FLAG
3338	013300	012705	000560	MOV	#PATRN,R5	:SET PATTERN NUMBER ADDRESS
3339	013300	012701	000002	MOV	#2,R1	:SET NUMBER OF CHARACTERS TO INPUT

3340	013620	012702	000015	MOV	#15,R2	:SET MAXIMUM LIMIT
3341	013624	012703	000000	MOV	#0,R3	:SET MINIMUM LIMIT
3342	013630	004737	023354	JSR	PC,ITR	:GO GET PATTERN NUMBER
3343	013636	012704	023354	MOV	#MSG69,R4	
3344	013640	104000		TTOUTT		:REQUEST TM
3345	013646	013703	000564	MOV	TMEX,R3	
3346	013650	104002		OCTPP		:PRINT CURRENT TM FLAG SETTING
3347	013656	012705	000564	MOV	#TMEX,R5	:GET TM FLAG ADDRESS
3348	013660	012701	000001	MOV	#1,R1	:SET SIZE OF RESPONSE
3349	013666	012702	000001	MOV	#1,R2	:SET UPPER LIMIT
3350	013670	012703	000000	MOV	#0,R3	:SET LOWER LIMIT
3351	013676	004737	023354	JSR	PC,TTR	:TM 1=YES
3352	013680	012704	025141	MOV	#MSG21,R4	
3353	013686	104000		TTOUTT		:REQUEST INTERCHANGE READ
3354	013690	013703	000566	MOV	INTRF,R3	
3355	013696	104002		OCTPP		:PRINT CURRENT SETTING
3356	013700	012705	000566	MOV	#INTRF,R5	:GET FLAG ADDRESS
3357	013706	012701	000001	MOV	#1,R1	:SET SIZE OF RESPONSE
3358	013710	012702	000001	MOV	#1,R2	:SET UPPER LIMIT
3359	013716	012703	000000	MOV	#0,R3	:SET LOWER LIMIT
3360	013720	004737	023354	JSR	PC,TTR	:GO GET RESPONSE
3361	013726	012704	025663	MOV	#MSG38,R4	
3362	013730	104000		TTOUTT		:REQUEST SINGLE PASS
3363	013736	013703	000570	MOV	SPFLG,R3	
3364	013740	104002		OCTPP		:PRINT CURRENT SETTING
3365	013746	012705	000570	MOV	#SPFLG,R5	:SET ADDRESS OF FLAG
3366	013750	012701	000001	MOV	#1,R1	:SET SIZE OF RESPONSE
3367	013756	012702	000001	MOV	#1,R2	:SET UPPER LIMIT
3368	013760	012703	000000	MOV	#0,R3	:SET LOWER LIMIT
3369	013766	004737	023354	JSR	PC,TTR	:GO GET RESPONSE
3370	013770	005737	014150	TST	SCVFL	:SEE IF SHORT CONVERSATION
3371	014000	001060		BNE	TINPX	:IF SO: BR
3372	014002	012704	025703	MOV	#MSG40,R4	
3373	014006	104000		TTOUTT		:PRINT READ STALL REQUEST
3374	014010	013703	000572	MOV	RSTAL,R3	
3375	014014	104002		OCTPP		:PRINT READ STALL
3376	014016	012705	000572	MOV	#RSTAL,R5	:SET READ STALL ADDRESS
3377	014022	012701	000006	MOV	#6,R1	:SET NUMBER OF CHARACTERS TO INPUT
3378	014026	012702	177777	MOV	#-1,R2	:SET MAXIMUM LIMIT
3379	014032	012703	000001	MOV	#1,R3	:SET MINIMUM LIMIT
3380	014036	004737	023354	JSR	PC,TTR	:GO GET READ STALL
3381	014042	012704	025731	MOV	#MSG41,R4	
3382	014046	104000		TTOUTT		:PRINT WRITE STALL REQUEST
3383	014050	013703	000574	MOV	WSTAL,R3	
3384	014054	104002		OCTPP		:PRINT READ STALL
3385	014056	012705	000574	MOV	#WSTAL,R5	:SET WRITE STALL ADDRESS
3386	014062	012701	000006	MOV	#6,R1	:SET NUMBER OF CHARACTERS TO INPUT
3387	014066	012702	177777	MOV	#-1,R2	:SET MAXIMUM LIMIT
3388	014072	012703	000001	MOV	#1,R3	:SET MINIMUM LIMIT
3389	014076	004737	023354	JSR	PC,TTR	:GO GET WRITE STALL
3390	014102	012704	025743	MOV	#MSG42,R4	
3391	014106	104000		TTOUTT		:PRINT TURN AROUND STALL REQUEST
3392	014110	013703	000576	MOV	TSTAL,R3	
3393	014114	104002		OCTPP		:PRINT TA STALL
3394	014116	012705	000576	MOV	#TSTAL,R5	:SET TURN AROUND STALL ADDRESS
3395	014122	012701	000006	MOV	#6,R1	:SET NUMBER OF CHARACTERS TO INPUT

TINP4:

```

3396 014126 012702 177777      MOV      #-1,R2      ;SET MAXIMUM LIMIT
3397 014132 012703 000001      MOV      #1,R3      ;SET MINIMUM LIMIT
3398 014136 004737 023354      JSR      PC,TTR     ;GO GET TURN AROUND STALL
3399 014142 005037 014150      TINPX:  CLR      SCVFL ;CLEAR SHORT CONVERSATION FLAG
3400 014146 000207              RTS      PC         ;EXIT
3401 014150 000000      SCVFL:  0          ;SHORT CONVERSATION FLAG
3402
3403
3404
3405 014152 000241              ;UNIT DESCRIPTION POSITIONING SUBROUTINE*****
3406 014154 006137 000644      TPOS:   CLC      ;POSITION CHARACTER
3407 014160 005303              DEC      R3        ;SEE IF DONE
3408 014162 001373              BNE     TPOS      ;IF NOT: BR
3409 014164 013700 000674      TPOS1:  MOV      UNP,R0 ;LOAD UNIT PCINTER
3410 014170 053760 000644 000746  BIS     TEMP2,UN1(R0) ;LOAD CHARACTER INTO UN1(R0)
3411 014176 000207              RTS      PC         ;EXIT
3412
  
```


3413
3414
3415
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

014200 005737 015150
014204 001044
014206 005737 000734
014212 001406
014214 005737 000560
014220 100003
014222 004737 015102
014226 000207
014230 023737 000560 014406
014236 001014
014240 013703 000552
014244 042703 177767
014250 023703 014410
014254 001404
014256 010337 014410
014262 004737 015152
014266 000207
014270 012703 027464
014274 013701 000560
014300 010137 014406
014304 062701 000001
014310 006301
014312 004771 002770
014316 032777 010000 164212
014324 001410
014326 012702 002002
014330 012701 027464
014334 042721 140300
014338 001302
014342 001374
014346 004737 015152
014352 012702 002000
014356 012701 033472
014360 001302
014364 001374
014370 013737 000552 014410
014376 042737 177767 014410

```
*****  
: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.  
*****  
DSUP: TST RDFL ;SEE IF DID RANDOM DATA  
BNE DS1 ;++G F SO BRANCH  
DSO: TST ASEOF ;SEE IF AUTO SEQ  
BEQ DSOC ;IF NOT: BR  
TST PATRN ;SEE IF AUTO RANDOM  
BPL DSOC ;IF NOT: BR  
JSR PC,DATR ;ELSE GO GENERATE RANDOM DATA  
RTS PC ;RETURN  
DSOC: CMP PATRN,PATS ;SEE IF NEW PATTERN  
BNE 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,PARS ;SAVE PARITY  
JSR PC,CRCLRC ;GO GENERATE EXPT CRC/LRC  
DSOB: RTS PC  
DSOA: MOV #WDATA,R3 ;R3 = ADDR OF WRITE BUFFER  
MOV PATRN,R1 ;R1 = PATTERN SELECTOR  
MOV R1,PATS  
ADD #1,R1 ;BUMP POINTER  
ASL R1 ;++G MAKE PATTERN SELECTOR EVEN  
JSR PC,@DATBL(R1) ;GO GENERATE PATTERN  
DS1: BIT #1000,@DT ;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,CRCLRC ;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,PARS ;GET UNIT DESCRIPTION  
BIC #177767,PARS ;MASK PARITY
```

```

3469 014404 000207
3470 014406 177777
3471 014410 000000
3472
3473
3474
3475 014412 005737 014554
3476 014416 001401
3477 014420 000207
3478 014422 012737 000001 014554 1$:
3479 014430 005077 164166
3480 014434 005077 164160
3481 014440 005037 000642
3482 014444 052777 000001 164146 DATOA:
3483 014452 105777 164142 DATOB:
3484 014456 100375
3485 014460 005001
3486 014462 117701 164134
3487 014466 005737 000642
3488 014472 001011
3489 014474 105701
3490 014476 001762
3491 014500 012737 000001 000642
3492 014506 010137 000644
3493 014512 010102
3494 014514 000753
3495 014516 110123
3496 014520 005302
3497 014522 001350
3498 014524 012701 027464
3499 014530 013702 000644
3500 014534 112123
3501 014536 022703 033472
3502 014542 003001
3503 014544 000207
3504 014546 005302
3505 014550 001371
3506 014552 000764
3507 014554 000000
3508

PATS: RTS PC :EXIT
PARS: -1 :PATTERN NUMBER SAVE
0

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

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

```
3509 ;ALL ONES*****
3510
3511 014556 012701 177777 DAT1: MOV #1,R1 ;R1=DATA
3512 014562 012702 002002 DAT1A: MOV #2002,R2 ;R2=WORD COUNT +2
3513 014566 010123 DAT1B: MOV R1,(R3)+ ;LOAD BUFFER
3514 014570 005302 DEC R2 ;SEE IF DONE
3515 014572 001375 BNE DAT1B ;IF NOT: BR
3516 014574 000207 RTS PC ;++G RETURN
3517
3518 ;ALL ZEROS*****
3519
3520 014576 005091 DAT2: CLR R1 ;R1=DATA
3521 014660 000770 BR DAT1A ;++G LOAD BUFFER
3522
3523 ;WALKING ONE*****
3524
3525 014602 012701 000001 DAT3: MOV #1,R1 ;R1=DATA
3526 014606 000241 CLC
3527 014610 012702 004004 DAT3A: MOV #4004,R2 ;R2=CHARACTER COUNT+4
3528 014614 110123 DAT3B: MOV R1,(R3)+ ;LOAD BUFFER
3529 014616 106101 ROLB R1 ;SET NEXT CHARACTER
3530 014620 005302 DEC R2 ;SEE IF DONE
3531 014622 001374 BNE DAT3B ;IF NOT: BR
3532 014624 000207 RTS PC ;++G RETURN
3533
3534 ;WALKING ZERO*****
3535
3536 014626 012701 000376 DAT4: MOV #376,R1 ;R1=START OF DATA
3537 014632 000261 SEC
3538 014634 000765 BR DAT3A ;++G LOAD BUFFER
3539
3540 ;ALTERNATING ONE/ZERO*****
3541
3542
3543 014636 012701 052525 DAT5: MOV #52525,R1 ;R1=DATA
3544 014642 000747 BR DAT1A ;++G LOAD BUFFER
3545
3546 ;ALTERNATING ZERO/ONE*****
3547
3548 014644 012701 125252 DAT6: MOV #125252,R1 ;R1=DATA
3549 014650 000744 BR DAT1A ;++G LOAD BUFFER
3550
3551 ;ONE/ZERO IN ALTERNATING WORDS*****
3552
3553 014652 012701 125252 DAT7: MOV #125252,R1 ;SET WORD 1
3554 014656 012702 052525 MOV #52525,R2 ;SET WORD 2
3555 014662 012704 001002 MOV #1002,R4 ;SET NUMBER OF ENTRIES
3556 014666 010123 DAT7A: MOV R1,(R3)+ ;LOAD WORD 1
3557 014670 010223 MOV R2,(R3)+ ;LOAD WORD 2
3558 014672 005304 DEC R4 ;SEE IF DONE
3559 014674 001374 BNE DAT7A ;IF NOT: BR
3560 014676 000207 RTS PC ;++G RETURN
3561
```

```

3562                                     :WALKING ONE/ALL ONE IN ALTERNATING CHARS****
3563
3564 014700 012702 002002  DAT10: MOV #2002,R2      :SET BUFFER SIZE
3565 014704 012701 000001      MOV #1,R1          :SET WALK BASE
3566 014710 000241
3567 014712 012713 177400  DAT10A: MOV #177400,(R3) :LOAD ALL ONE BYTE
3568 014716 050123           BIS R1,(R3)+      :LOAD WALK BYTE
3569 014720 106101           ROLB R1          :WALK ONE
3570 014722 005302           DEC R2
3571 014724 001372           BNE DAT10A       :DO FULL BUFFER
3572 014726 000207           RTS PC          :++G RETURN
3573
3574                                     :ALL BITS 0-377* *****
3575
3576 014730 005001
3577 014732 012702 004004  DAT11: CLR R1          :R1=STARTING DATA
3578 014736 110123           MOV #4004,R2     :R2=CHARACTER COUNT+4
3579 014740 105201           DAT11A: MOVB R1,(R3)+ :LOAD BUFFER
3580 014742 005302           INCB R1          :BUMP DATA
3581 014744 001374           DEC R2           :SEE IF DONE
3582 014746 000207           BNE DAT11A       :IF NOT: BR
3583                                     :++G RETURN
3584
3585                                     :ALL BITS 377-0*****
3586 014750 012701 000377  DAT12: MOV #377,R1     :R1=STARTING DATA
3587 014754 012702 004004  MOV #4004,R2     :R2=CHARACTER COUNT+4
3588 014760 110123           DAT12A: MOVB R1,(R3)+ :LOAD BUFFER
3589 014762 105301           DECB R1          :BUMP DATA
3590 014764 005302           DEC R2           :SEE IF DONE
3591 014766 001374           BNE DAT12A       :IF NOT: BR
3592 014770 000207           RTS PC          :++G RETURN
3593
3594                                     :ALTERNATING CHARACTERS 0 AND 377*****
3595
3596 014772 012701 000377  DAT13: MOV #377,R1   :R1 = DATA
3597 014776 000137 014562  JMP DAT1A        :LOAD BUFFER
3598
3599                                     :WALKING ZERO/ALL ZERO IN ALTERNATING CHARS*****
3600
3601 015002 012702 002002  DAT14: MOV #2002,R2   :SET BUFFER SIZE
3602 015006 012701 000376  MOV #376,R1      :SET WALK BASE
3603 015012 000261
3604 015014 010113           DAT14A: MOV R1,(R3)   :LOAD WALK BYTE
3605 015016 042723 177400  BIC #177400,(R3)+ :CLEAR HIGH BYTE
3606 015022 106101           ROLB R1          :WALK ZERO BIT
3607 015024 005302           DEC R2
3608 015026 001372           BNE DAT14A       :FILL BUFFER
3609 015030 000207           RTS PC          :++G RETURN
3610

```

3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644

:AUTO SEQUENCE PATTERN*****

```
DAT15: MOV #200,R2 ;SET NUMBER OF ENTRIES
DAT15A: MOV #APATS,R1 ;SET START OF PATTERN
MOV #10,R4 ;SET SIZE OF PATTERN
DAT15B: MOV (R1)+,(R3)+ ;FILL BUFFER
DEC R4 ;SEE IF DONE PATTERN
BNE DAT15B ;IF NOT: BR
DEC R2 ;SEE IF DONE BUFER
BNE DAT15A ;IF NOT: BR
RTS PC ;++G RETURN

APATS: 0
177400
377
0
-1
377
177400
-1
```

:RANDOM DATA GENERATOR SUBROUTINE*****

```
DATR: MOV FMCNT,R4 ;SET NUMBER OF FRAMES
MOV #WDATA,R3 ;SET ADDRESS OF START OF BUFFER
MOV #-1,R1 ;SET HIGH LIMIT
CLR R2 ;SET LOW LIMIT
DATRO: JSR PC,RANG ;GO GENERATE NUMBER
MOV RANSV,(R3)+ ;LOAD BUFFER
INC R4 ;SEE IF DONE WHOLE BUFFER
BNE DATRO ;IF NOT: BR
JSR PC,DS1 ;GO CHECK FOR 7 CH
MOV #1,RDFL ;SET RANDOM DATA FLAG
RTS PC ;EXIT

RDFL: 0 ;RANDOM DATA SELECT FLAG
```

000200
015062
000010
000530
001375
005302
001367
000207
000000
177400
000377
000000
177777
000377
177400
177777
000556
027464
177777
023322
000626
014316
000001 015150
000207
000000

3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700

015152 013700 000556
015156 005400
015160 012701 027464
015164 005037 015534
015170 111104
015172 004737 015362
015176 004737 015510
015202 000241
015204 006004
015206 103014
015210 052704 000400
015214 000241
015216 010405
015220 042705 177703
015224 005105
015226 042705 177703
015228 042704 000074
015232 050504
015234 010405 015534
015236 005300
015238 001401
015240 000747
015242 013704 015534
015244 005137 015534
015246 042737 177050 015534
015248 042704 177727
015250 050437 015534
015252 013737 015534 015536
015254 013700 000556
015256 005400
015258 012701 027464
015260 005037 015534
015262 111104
015264 004737 015362
015266 004737 015510
015268 005300
015270 001371
015272 013704 015536
015274 004737 015510
015276 013737 015534 015540
015278 000207
015280 005704
015282 001010
015284 032737 000010 000552
015286 001404
015288 012704 000420
015402 005201

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

```
3701 015404 000207  
3702 015406 005037 015542  
3703 015412 012703 000010  
3704 015416 032703 000001  
3705 015422 001402  
3706 015424 005237 015542  
3707 015430 000241  
3708 015432 006004  
3709 015434 005303  
3710 015436 001367  
3711 015440 112104  
3712 015442 042704 177400  
3713 015446 032737 000001 015542  
3714 015454 001005  
3715 015456 032737 000010 000552  
3716 015464 001406  
3717 015466 000207  
3718 015470 032737 000010 000552  
3719 015476 001001  
3720 015500 000207  
3721 015502 052704 000400  
3722 015506 000207  
3723 015510 010446  
3724 015512 043716 015534  
3725 015516 040437 015534  
3726 015522 052637 015534  
3727 015526 013704 015534  
3728 015532 000207  
3729  
3730 015534 000000  
3731 015536 000000  
3732 015540 000000  
3733 015542 000000  
3734
```

```
CLPE: RTS PC :RETURN  
CLR PARCNT :CLEAR BIT COUNTER  
MOV #10,R3 :SET NUMBER OF BITS  
CLP0: BIT #1,R4 :SEE IF ONE BIT  
BEQ CLP1 :IF NOT: BR  
INC PARCNT :BUMP COUNTER  
CLP1: CLC :ROTATE TO NEXT BIT  
ROR R4  
DEC R3  
BNE CLP0 :CONTINUE FOR ALL BITS  
MOVB (R1)+,R4  
BIC #177400,R4  
BIT #1,PARCNT :SEE IF ODD NUMBER OF ONE BITS  
BNE CLP2 :IF SO: BR  
BIT #10,UDES :SEE IF SHOULD BE EVEN PARITY  
BEQ CLP3 :IF NOT: BR  
RTS PC :ELSE EXIT  
CLP2: BIT #10,UDES :SEE IF SHOULD BE ODD PARITY  
BNE CLP3 :IF NOT: BR  
RTS PC :ELSE EXIT  
CLP3: BIS #400,R4 :SET PARITY BIT  
RTS PC  
XOR: MOV R4, -(SP)  
BIC XORS, (SP)  
BIC R4, XORS :XOR SUBROUTINE: R4 WITH XORS  
BIS (SP)+, XORS  
MOV XORS, R4  
RTS PC  
XORS: 0 :XOR SAVE  
EXCRC: 0 :EXPECTED CRC  
EXLRC: 0 :EXPECTED LRC  
PARCNT: 0 :PARITY COUNTER
```

3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790

015544 005037 000656
015550 005037 000704
015554 013705 000556
015560 032737 000020 000552
015566 001402
015570 000261
015572 006005
015574 012701 027464
015600 012702 033472
015604 032737 000010 000552
015612 001430
015614 032737 000020 000552
015622 001024
015624 032737 002000 000552
015632 001020
015634 105711
015636 001404
015640 005201
015642 005205
015644 001373
015646 000406
015650 112721 000020
015654 012737 177777 014406
015662 000767
015664 013705 000556
015670 012701 027464
015674 032737 010000 000562
015678 001462
015682 013704 000556
015686 005404
015690 032737 000020 000552
015694 001402
015698 000241
015702 001404
015706 040401
015710 040402
015714 032737 000001 000556
015718 001401
015722 105722
015726 032737 000020 000552
015730 001431

```
*****  
:DATA CHECK SUBROUTINE:  
*****  
:THIS SUBROUTINE IS USED TO COMPARE EACH CHARACTER  
:OF DATA READ FROM TAPE WITH THE EXPECTED CHARACTER.  
:ANY ERROR DETECTED WILL CAUSE CONTROL TO BE  
:PASSED TO AN ERROR PRINT SUBROUTINE AND A  
:SUBROUTINE TO ACCUMULATE THE NUMBER OF BITS  
:DROPPED AND PICKED UP FROM EACH CHARACTER.  
:THE NUMBER OF HEAD ERRORS IS ALSO ACCUMULATED.  
:DATA CHECKING MAY BE TERMINATED BY USE OF  
:CONSOLE SWITCH THIRTEEN (13).  
*****  
DCHK: CLR BBC :CLEAR BAD RECORD CNTR  
CLR DERFL :CLEAR DATA ERROR FLAG  
MOV FMCNT,R5 :LOAD CHAR COUNT  
BIT #20,UDES :SEE IF CORE DUMP  
BEQ DCHKO :IF NOT: BR  
SEC  
ROR R5 :R5 = FC/2  
DCHKO: MOV #WDATA,R1 :SET WRITE DATA ADDR  
MOV #RDATA,R2 :SET READ DATA ADDR  
BIT #10,UDES :SEE IF EVEN PARITY  
BEQ DFOCO :IF NOT: BR  
BIT #20,UDES :SEE IF CORE DUMP PARITY  
BNE DFOCO :IF SO: BR  
BIT #2000,UDES :SEE IF PE MODE  
BNE DFOCO :IF SO: BR  
DFGF: TSTB (R1) :SEE IF 0 CHAR  
BEQ DFOD :IF SO: BR  
INC R1 :BUMP POINTER  
DFOE: INC R5 :SEE IF DONE  
BNE DFOF :IF NOT: BR  
BR DFOD :ELSE CONTINUE  
DFOD: MOVB #20,(R1)+ :SET 20 IN PLACE OF 0  
MOV #-1,PATS :SET PATTERN GENERATE FLAG  
BR DFOE  
DFOC: MOV FMCNT,R5 :RESET CHAR CNT  
MOV #WDATA,R1 :RESET DATA ADDRESS  
DFOCO: BIT #10000,RDCMD :SEE IF READ REVERSE  
BEQ DFO :IF NOT: BR  
DFOB: MOV FMCNT,R4 :GET FRAME COUNT  
NEG R4 :SET TO WHOLE NUMBER  
BIT #20,UDES :SEE IF CORE DUMP  
BEQ DFOBO :IF NOT: BR  
CLC  
ROR R4 :SET TO FC/2  
DFOBO: ADD R4,R1 :POINT TO START OF WRITE DATA  
ADD R4,R2 :POINT TO START OF READ DATA  
BIT #1,FMCNT :SEE IF ODD FRAME COUNT  
BEQ DFOA :IF NOT: BR  
TSTB (R2)+ :BUMP POINTER  
DFOA: BIT #20,UDES :SEE IF CORE DUMP  
BEQ DFOA4 :IF NOT: BR
```


3791	015754	000241		CLC		
3792	015756	132742	000001	BITB	#1, -(R2)	:SEE IF BIT 0 = 1
3793	015762	001401		BEQ	DF0A0	:IF NOT: BR
3794	015764	000261		SEC		
3795	015766	106012		DF0A0:	RORB	(R2)
3796	015770	000241		CLC		
3797	015772	132712	000001	BITB	#1, (R2)	
3798	015776	001401		BEQ	DF0A1	
3799	016000	000261		SEC		
3800	016002	106012		DF0A1:	RORB	(R2)
3801	016004	000241		CLC		:POSITION BITS FOR REVERSE CORE DUMP
3802	016006	132712	000001	BITB	#1, (R2)	
3803	016012	001401		BEQ	DF0A2	
3804	016014	000261		SEC		
3805	016016	106012		DF0A2:	RORB	(R2)
3806	016020	000241		CLC		
3807	016022	132712	000001	BITB	#1, (R2)	
3808	016026	001401		BEQ	DF0A3	
3809	016030	000261		SEC		
3810	016034	106012		DF0A3:	RORB	(R2)
3811	016036	005205		INC	R2	:RESET POINTER
3812	016038	124142		DF0A4:	CMPB	-(R1), -(R2)
3813	016040	001010		BNE	DF1	:TEST DATA CHARACTER
3814	016042	105037	000656	CLRB	BBC	:IF NOT GOOD: BR
3815	016046	000411		BR	DF2	:CLEAR BAD RECORD COUNTER
3816	016050	122122		DF0:	CMPB	(R1)+, (R2)+
3817	016052	001003		BNE	DF1	:CHECK DATA
3818	016054	105037	000656	CLRB	BBC	:IF BAD: BR
3819	016060	000404		BR	DF2	:CLEAR BAD RECORD CNTR
3820	016062	004737	016666	DF1:	JSR	PC, DRPKF
3821	016066	004737	016160	DF2:	JSR	PC, DERR
3822	016072	005205		INC	R5	:GO GET DROPS AND PICKS
3823	016074	001405		BEQ	DF3	:GO DO PRINT
3824	016076	032737	010000 000562	BIT	#10000, RDCMD	:BUMP CHAR CNTR
3825	016104	001761		BEQ	DF0	:IF DONE ALL: BR
3826	016106	000716		SR	DF0A	:SEE IF READ REVERSE
3827	016110	005037	000664	DF3:	CLR	DF0A
3828	016114	005737	000704	TST	HDRFL	:ELSE CONTINUE READ REV
3829	016120	001416		BEQ	DFX	:CLEAR HEADER FLAG
3830	016122	005737	000706	TST	SERFL	:SEE IF HAD DATA ERROR
3831	016126	001013		BNE	DFX	:IF NOT: BR
3832	016130	013704	000674	MOV	UNP, R4	IF NOT DATA ERROR ONLY: BR
3833	016134	032737	010000 000562	BIT	#10000, RDCMD	:SEE IF READ REVERSE
3834	016142	001003		BNE	DF4	:IF SO: BR
3835	016144	005264	001130	INC	DATER1(R4)	:BUMP DATA ERROR FORWARD COUNTER
3836	016150	000402		BR	DFX	
3837	016152	005264	001170	DF4:	INC	DEREV1(R4)
3838	016156	000207		DFX:	RTS	PC
3839						:BUMP REVERSE DATA ERROR
						:EXIT

3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895

016160 032777 002000 162420
016166 031067
016170 005237 000670
016174 005737 000664
016200 001007
016202 004737 022736
016204 012704 024736
016212 104000
016214 004737 021156
016220 012704 024755
016224 104000
016226 010203
016230 162703 033472
016234 005303
016236 032737 010000 000562
016244 001402
016246 010503
016250 005103
016252 104002
016254 012704 024743
016260 104000
016262 032737 010000 000562
016270 001402
016272 111103
016274 000401
016276 114103
016300 004737 024166
016304 012704 024750

DERR: BIT #2000, @SWR ;SEE IF SHOULD PRINT ERRORS
BNE DERR4 ;++G BRANCH IF NOT
DERRO: INC PFLG ;SET PRINT FLAG
TST HDRFL ;SEE IF HAVE PRINTED HEADER
BNE DERROA ;IF SO: BR
JSR PC, PAPRT ;PRINT CYCLE NUMBER
MOV #MSG1, R4 ;LOAD ERROR MSG ADDR
TTOUTT ;PRINT ERROR
JSR PC, FRPRT ;PRINT F OR R
DERROA: MOV #MSG4, R4
TTOUTT ;PRINT CHAR NO. HEADER
MOV R2, R3
SUB #RDATA, R3 ;POINT TO CHAR
DEC R3
BIT #10000, RDCMD ;SEE IF READ REVERSE
BEQ DERROB ;IF NOT: BR
MOV R5, R3 ;GET CHAR NUMBER
COM R3
DERROB: OCTPP ;PRINT CHAR NUMBER
MOV #MSG2, R4
TTOUTT ;PRINT EXPECTED DATA
BIT #10000, RDCMD ;SEE IF READ REVERSE
BEQ DERROC ;IF NOT: BR
MOVB (R1), R3 ;GET CHAR
BR DERROD
DERROC: MOVB -(R1), R3 ;LOAD EXPECTED DATA
DERROD: JSR PC, DOUT ;GO PRINT CHAR
MOV #MSG3, R4

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

3896	016310	104000			TTOUTT		:PRINT RECIEVED DATA
3897	016312	032737	010000	000562	BIT #10000,RDCMD		:SEE IF READ REVERSE
3898	016320	001402			BEQ DERR1		:IF NOT: BR
3899	016322	111203			MOVB (R2),R3		:GET CHAR
3900	016327	000401			BR DERR2		
3901	016330	114203			DERR1: MOVB -(R2),R3		
3902	016330	004737	024166		DERR2: JSR PC,DOUT		:PRINT BAD CHAR
3903	016334	032737	010C00	000562	BIT #10000,RDCMD		:SEE IF READ REVERSE
3904	016342	001001			BNE DERR4		:++G BRANCH IF NOT
3905	016344	122122			DERR3: CMPB (R1)+,(R2)+		:PESET POINTERS
3906	016346	105238	000656		DERR4: INCB BBC		:BUMP BAD RECORD CNTR
3907	016352	122737	000010	000656	CMPB #10,BBC		:SEE IF BLD BTH
3908	016360	001120			BNE DEREX		:IF NOT: BR
3909	016362	032777	002000	162216	BIT #2000,BSWR		:SEE IF PRINT INHIBIT
3910	016370	001003			BNE 18		:IF SO: BR
3911	016372	012704	025067		MOV #MSG15,R4		
3912	016376	104000			TTOUTT		:PRINT BLD BTH
3913	016400	105037	000656		18: CLR B BBC		:RESET BAD RECORD CNTR
3914	016404	000337	000656		SWAB BBC		:POSITION BLD BTH AMOUNT
3915	016410	105237	000656		INCB BBC		:BUMP AMOUNT
3916	016414	122737	000003	000656	CMPB #3,BBC		:SEE IF HAD 3 BLD BTHS
3917	016422	101052			BHI DERR4B		:IF NOT: BR
3918	016424	000337	000656		SWAB BBC		:REPOSITION BBC
3919	016430	022705	177767		CMP #177767,R5		:SEE IF ON LAST EIGHT CHARS
3920	016440	101470			BLOS DERR6		:IF SO: BR
3921	016444	012705	177767		MOV #177767,R5		:SET CHAR CNTR TO 8
3922	016448	032737	010000	000562	BIT #10000,RDCMD		:SEE IF READ REVERSE
3923	016450	001416			BEQ DERR4A		:IF NOT: BR
3924	016454	012701	027464		MOV #WDATA,R1		:GET START OF BUFFER
3925	016458	012702	033472		MOV #RDATA,R2		:GET START OF BUFFER
3926	016462	062701	000010		ADD #10,R1		
3927	016466	062702	000010		ADD #10,R2		:POINT TO START +10
3928	016470	032737	000001	000556	BIT #1,FM CNT		:SEE IF ODD FRAME COUNT
3929	016500	001450			BEQ DEREX		:IF NOT: BR
3930	016502	105722			TSTB (R2)+		:BUMP POINTER
3931	016504	000446			BR DEREX		
3932	016506	013737	000556	000642	DERR4A: MOV FM CNT,TEMP1		:LOAD CHAR COUNT
3933	016514	003437	000642		NEG TEMP1		:++G FORM TWO'S COMPLEMENT
3934	016520	162737	000010	000642	SUB #10,TEMP1		:POINT TO BUFFER -8
3935	016526	013701	000642		MOV TEMP1,R1		:POINT TO NEXT CHAR
3936	016532	062701	027464		ADD #WDATA,R1		:POINT TO NEXT WRITE CHAR
3937	016538	013702	000642		MOV TEMP1,R2		:POINT TO END OF READ DATA -8 FORWARD
3938	016542	062702	033472		ADD #RDATA,R2		:POINT TO NEXT CHAR
3939	016546	000423			BP DEREX		:EXIT
3940	016550	000337	000656		DERR4B: SWAB BBC		:REPOSITION BBC
3941	016554	062705	000024		ADD #24,R5		:SKIP 20 CHARS
3942	016560	103416			BCC DERR6		:IF EXCEED RECORD SIZE: BR
3943	016562	032737	010000	000562	BIT #10000,RDCMD		:SEE IF READ REVERSE
3944	016570	001405			BEQ DERR5		:IF NOT: BR
3945	016572	162701	000024		SUB #24,R1		
3946	016576	162702	000024		SUB #24,R2		:RESET POINTERS
3947	016602	000407			BR DEREX		
3948	016604	062701	000024		DERR5: ADD #24,R1		:SKIP 20 CHARS
3949	016610	062702	000024		ADD #24,R2		:SKIP FORWARD 20 CHARS
3950	016614	000402			BR DEREX		
3951	016616	012705	177777		DERR6: MOV #-1,R5		:SET TO EOR

3952	016622	005777	161760	DEREX:	TST	@SWR	:++G BRANCH IF CONTINUE ON ERROR
3953	016626	100012			BPL	DEREX1	:++G
3954	016630	104006			STOPP		
3955	016632	005737	000670		TST	PFLG	:SEE IF PRINTED
3956	016636	001006			BNE	DEREX1	:IF SO: BR
3957	016640	032777	002000	161740	BIT	#2000 @SWR	:SEE IF SHOULD PRINT
3958	016646	001002			BNE	DEREX1	:IF NOT: BR
3959	016650	000137	01617		JMP	DERRO	:ELSE PRINT
3960	016654	005037	000670	DEREX1:	CLR	PFLG	:CLEAR FLAG
3961	016660	005237	000704		INC	DERFL	:BUMP DATA ERROR FLAG
3962	016664	000207			RTS	PC	:RETURN
3963							

3964
3965
3966
3967
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

016656 005037 000642
016672 005037 000644
016676 005037 000646
016702 111137 000642
016706 111237 000644
016712 013704 000674
016716 016437 000770 000720
016724 016437 001010 000716
016732 032737 010000 000562
016740 001005
016742 124142
016744 112137 000642
016750 112237 000644
016754 004737 016766
016760 004737 017204
016764 000207
016766 113703 000642
016772 113704 000644
016776 140403
017000 001001
017002 000207
017004 012737 000010 000710
017012 132703 000001
017016 001453
017020 105737 000646
017024 001016
017026 005277 161664
017032 005777 161660
017036 100045
017040 032777 002000 161540
017046 001402
017050 004737 022736
017054 004737 017250
017060 000415
017062 005277 161632
017066 005777 161626
017072 100027
017074 032777 002000 161504

```
*****  
:DROPS AND PICKS SUBROUTINE:  
:THIS SUBROUTINE IS USED TO ACCUMULATE FROM  
:EACH BAD DATA CHARACTER FOUND THE NUMBER  
:OF BITS WHICH WERE EITHER DROPPED OR PICKED UP.  
:TWO COUNTERS PER SLAVE ARE USED TO ACCUMULATE THIS  
:INFORMATION AND CAN STORE UP TO 32K DROPS  
:OR PICKS BEFORE OVERFLOWING. IF OVERFLOW IS  
:ABOUT TO OCCUR, THESE ACCUMULATORS ARE  
:PRINTED IN OCTAL AND RESET TO ZERO.  
:THE CONTENTS OF THE ACCUMULATORS MAY BE  
:DISPLAYED AT ANY TIME BY SETTING CONSOLE  
:SWITCH FOURTEEN TO A ONE (1). THE PRINTOUT WILL OCCUR  
:AT THE END OF THE CURRENT BLOCK CYCLE.  
*****  
DRPKF: CLR TEMP1  
CLR TEMP2  
CLR TEMP3  
MOVB (R1),TEMP1 :LOAD GOOD CHAR  
MOVB (R2),TEMP2 :LOAD BAD CHAR  
MOV UNP R4  
MOV PIK1(R4),BPKP  
MOV DRP1(R4),BDPP  
BIT #10000,R0CMD :SEE IF READ REVERSE  
BNE DRPK :IF SO: BR  
CMPB -(R1),-(R2) :POINT TO CHAR  
MOVB (R1)+,TEMP1 :LOAD GOOD CHAR  
MOVB (R2)+,TEMP2 :LOAD BAD CHAR  
DRPK: JSR PC,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  
DPC0: BITB #1,R3 :SEE IF DROPPED OR PICKED THIS BIT  
BEQ DPC2 :IF NOT: BR  
TSTB TEMP3 :SEE IF ON PICKS  
BNE DPC1 :IF SO: BR  
INC @BDPP :BUMP DROP CNTR  
TST @BDPP  
BPL DPC2 :IF NO OVERFLOW: BR  
BIT #2000,@SWR :SEE IF HAVE PRINTED DATA  
BEQ DPC0A :IF SO: BR  
JSR PC,PAPRT :PRINT CYCLE NUMBER  
DPC0A: JSR PC,DPPRT :PRINT DROPS AND PICKS  
BR DPC2A  
DPC1: INC @BPKP :BUMP PICK CNTR  
TST @BPKP :SEE IF OVERFLOW  
BPL DPC2 :IF NOT: BR  
BIT #2000,@SWR :SEE IF HAVE PRINTED DATA
```

4020	017102	001402			BEQ	DPC1A		:IF SO: BR
4021	017104	004737	022736		JSR	PC,PAPRT		:PRINT CYCLE NUMBER
4022	017110	004737	017250		DPC1A: JSR	PC,DPPRT		:PRINT DROPS AND PICKS
4023	017114	013704	000674		DPC2A: MOV	UNP,R4		
4024	017120	016403	001010		MOV	DRP1(R4),R3		:SET DROP POINTER
4025	017124	016402	000770		MOV	PIK1(R4),R4		:SET PICK PCINTER
4026	017130	012737	000010	000710	MOV	#10,BCNT		:SET NUMBER OF BITS
4027	017136	005023			DPC2B: CLR	(R3)+		:CLEAR DROPS
4028	017140	005024			CLR	(R4)+		:CLEAR PICK
4029	017142	005337	000710		DEC	BCNT		:SEE IF DONE
4030	017146	001373			BNE	DPC2B		:IF NOT: BR
4031	017150	000207			RTS	PC		:EXIT
4032	017152	000241			DPC2: CLC			
4033	017154	106003			RORB	R3		:GET NEXT BIT
4034	017156	005337	000710		DEC	BCNT		:SEE IF DONE
4035	017162	001407			BEQ	DPC3		
4036	017164	062737	000002	000720	ADD	#2,BPKP		
4037	017172	062737	000002	000716	ADD	#2,BDPP		
4038	017200	000704			BR	DPC0		:++G CONTINUE
4039	017202	000207			RTS	PC		:RETURN
4040	017204	013704	000674		PICK: MOV	UNP,R4		:GET UNIT POINTER
4041	017210	016437	000770	000720	MOV	PIK1(R4),BPKP		:SET PICK POINTER
4042	017216	016437	001010	000716	MOV	DRP1(R4),BDPP		:SET DROP POINTER
4043	017224	113704	000642		MOVB	TEMP1,R4		:R4 = GOOD CHAR
4044	017230	113703	000644		MOVB	TEMP2,R3		:R3 = BAD CHAR
4045	017234	112737	000001	000646	MOVB	#1,TEMP3		:SET PICK FLAG
4046	017242	004737	016776		JSR	PC,DPC		:GO CHECK PICKS
4047	017246	000207			RTS	PC		:EXIT
4048	017250	012704	025363		DPPRT: MOV	#MSG26,R4		
4049	017254	104000			TTOUTT			:PRINT DROP HEADER
4050	017256	013704	000674		MOV	UNP,R4		
4051	017262	016437	001010	000716	MOV	DRP1(R4),BDPP		:SET DROP POINTER
4052	017270	016437	000770	000720	MOV	PIK1(R4),BPKP		:SET PICK POINTER
4053	017276	062737	000016	000716	ADD	#16,BDPP		
4054	017304	062737	000016	000720	ADD	#16,BPKP		
4055	017312	012737	000010	000710	MOV	#10,BCNT		:SET NUMBER TO PRINT
4056	017320	017703	161372		DPPRT0: MOV	@BDPP,R3		
4057	017324	104002			OCTPP			:PRINT DROPS
4058	017326	005337	000710		DEC	BCNT		:SEE IF DONE
4059	017332	001404			BEQ	DPPRT1		:IF NOT: BR
4060	017334	162737	000002	000716	SUB	#2,BDPP		:BUMP POINTER
4061	017342	000766			BR	DPPRT0		:CONTINUE FOR ALL 8 BITS
4062	017344	012737	000010	000710	DPPRT1: MOV	#10,BCNT		:SET NUMBER TO PRINT
4063	017352	012704	025374		MOV	#MSG27,R4		
4064	017356	104000			TTOUTT			:PRINT PICK HEADER
4065	017360	017703	161334		DPPRT2: MOV	@BPKP,R3		
4066	017364	104002			OCTPP			:PRINT PICKS
4067	017366	005337	000710		DEC	BCNT		:SEE IF DONE
4068	017372	001404			BEQ	DPPRTX		:IF SO: BR
4069	017374	162737	000002	000720	SUB	#2,BPKP		:BUMP POINTER
4070	017402	000766			BR	DPPRT2		:CONTINUE FOR ALL 8 BITS
4071	017404	000207			DPPRTX: RTS	PC		:RETURN

4072
 4073
 4074
 4075
 4076
 4077
 4078
 4079
 4080
 4081
 4082
 4083
 4084
 4085
 4086
 4087
 4088
 4089
 4090
 4091
 4092
 4093
 4094
 4095
 4096
 4097
 4098
 4099
 4100
 4101
 4102
 4103
 4104
 4105
 4106
 4107
 4108
 4109
 4110
 4111
 4112
 4113
 4114
 4115
 4116
 4117
 4118
 4119
 4120
 4121
 4122
 4123
 4124
 4125
 4126
 4127

017406 013703 000556
 017412 032703 000001
 017416 001401
 017420 005303
 017422 005403
 017424 032737 000020 000552
 017426 001402
 017432 000241
 017434 006003
 017436 032737 000010 000672
 017440 001414
 017446 032737 010000 000562
 017450 001405
 017456 012703 033472
 017460 162703 000002
 017464 000405
 017470 062703 033472
 017472 000402
 017476 062703 027464
 017500 010337 021132
 017504 012704 000007
 017510 012701 021134
 017514 005302
 017520 005304
 017524 001375
 017526 020377 160762
 017532 001402
 017534 005237 021134
 017540 032737 000010 000672
 017546 001006
 017550 005777 160742

```

*****
:STATUS CHECK SUBROUTINE:
:
:THIS SUBROUTINE IS USED TO PERFORM A CHECK OF
:BOTHS THE MASSBUS CONTROLLER (RH11) AND THE TAPE
:CONTROLLER (TMO2). THE RH11 IS CHECKED FOR ERRORS
:AS REFLECTED IN REGISTERS CS1 AND CS2 AND ALSO THAT
:THE BUS ADDRESS (BA) AND WORD COUNT (WC) ARE
:CORRECT. THE TMO2 IS CHECKED FOR DRIVE STATUS (DS),
:DRIVE ERRORS (ER), AND PROPER FRAME COUNT. THE SPECIAL
:CHECK CHARACTERS (CRC+LRC) ARE ALSO CHECKED WHEN
:APPROPRIATE (IE: NRZ READ OR WRITE). CERTAIN TYPES
:OF DRIVE ERRORS IN PE OPERATION WILL BE ACCOMPANIED
:BY THE DISPLAY OF THE DEAD TRACK REGISTER (CC). THESE
:TYPES ARE ER BITS 15,10,7,6. THE PRINTOUTS OF BAD
:CRC,LRC,FC, AND BA WILL SHOW BOTH THE EXPECTED AND
:RECEIVED VALUES (IE: EXPT-RCVD). ONLY THOSE REGISTERS
:WHICH ARE IN ERROR WILL BE PRINTED AND ALL PRINTOUTS
:ARE IN OCTAL FORMAT WITH NO LEADING ZEROS. AS IN
:DATA ERRORS, STATUS ERRORS ARE PRECEDED BY HEADER
:DESCRIBING THE HARDWARE UNDER TEST, THE BLOCKING
:INFORMATION, AND THE ERROR TYPE.
*****
ERCHK: MOV FMCNT,R3 ;GET FRAME COUNT
BIT #1,R3 ;SEE IF ODD
BEQ ERO ;IF NOT: BR
DEC R3 ;BUMP COUNT
ER0: NEG R3
BIT #20,UDES ;SEE IF CORE DUMP
BEQ EROB ;IF NOT: BR
CLC
ROR R3 ;SET TO FC/2
ER0B: BIT #10,MTC1 ;SEE IF WRITE OP
BEQ ER1 ;IF SO: BR
BIT #10000,RDCMD
BEQ EROA
MOV #RDATA,R3
SUB #2,R3 ;SET POINTER
BR ER2
ER0A: ADD #RDATA,R3 ;BUILD EXPT READ ADDRESS
BR ER2
ER1: ADD #WDATA,R3 ;BUILD EXPT WRITE ADDRESS
ER2: MOV R3,CADEN ;SAVE ADDRESS
MOV #7,R4
MOV #BAER,R1
ER2A0: CLR (R1)+ ;CLEAR FLAGS
DEC R4
BNE ER2A0
CMP R3,@BA ;SEE IF ADDRESS OK
BEQ ER2A1 ;IF SO: BR
INC BAER ;SET BUS ADDRESS ERROR
ER2A1: BIT #10,MTC1 ;SEE IF WRITE OPER
BNE ER2B ;IF NOT: BR
ER2A: TST @FC ;SEE IF FC=0

```

4128	017554	001441			BEQ	ER3	:IF SO: BR
4129	017556	005237	021142		INC	FCER	:SET FC ERROR
4130	017562	000436			BR	ER3	:++G
4131	017564	032737	000040	000672	ER2B: BIT	#40,MTC1	:SEE IF SPACE OPER
4132	017572	001766			BEQ	ER2A	:IF SO: BR
4133	017574	005737	000676		TST	TMFLG	:SEE IF TM TIME
4134	017600	001011			BNE	ER2D	:IF SO: BR
4135	017602	013703	000556		MOV	FMCNT,R3	
4136	017606	005403			NEG	R3	:R3 = EXPT RECORJ SIZE
4137	017610	020377	160702		ER2C: CMP	R3,#FC	:SEE IF FC = EXPT
4138	017614	001421			BEQ	ER3	:IF SO: BR
4139	017616	005237	021142		INC	FCER	:SET FC ERROR FLAG
4140	017622	000416			BR	ER3	:++G
4141	017624	032737	002000	000552	ER2D: BIT	#2000,UDES	:SEE IF PE
4142	017632	001346			BNE	ER2A	:IF SO: BR
4143	017634	032737	010000	000562	BIT	#10000,RDCMD	:SEE IF READ REVERSE
4144	017642	001003			BNE	ER2E	:IF SO: BR
4145	017644	012703	000002		MOV	#2,R3	
4146	017650	000757			BR	ER2C	:LOOK FOR EXPT = 2
4147	017652	012703	000001		ER2E: MOV	#1,R3	
4148	017656	000754			BR	ER2C	:GO CHECK FC FOR TM
4149	017660	032777	160000	160622	ER3: BIT	#160000,#C1	:SEE IF COUNT ERROR
4150	017666	001441			BEQ	ER4	
4151	017670	017703	160624		MOV	#CS,R3	:GET CONT STATUS REG
4152	017674	042703	000307		BIC	#307,R3	:MASK OUT IR,OR,UNIT NO.
4153	017700	005703			TST	R3	:SEE IF ANY OTHER ERRORS
4154	017702	001407			BEQ	EP3A	:IF NOT: BR
4155	017704	005737	000676		TST	TMFLG	:SEE IF TAPE MARK TIME
4156	017710	001426			BEQ	ER3B	:IF NOT: BR
4157	017712	042703	001000		BIC	#1000,R3	:MASK MISSED TRANS
4158	017716	005703			TST	R3	:SEE IF ANY OTHER ERRORS
4159	017720	001022			BNE	ER3B	:IF SO: BR
4160	017722	032777	060000	160560	ER3A: BIT	#60000,#C1	:SEE IF EITHER TRE OR MCPE
4161	017730	001420			BEQ	ER4	:IF NOT: BR
4162	017732	005737	000676		TST	TMFLG	:SEE IF TM TIME
4163	017736	001413			BEQ	ER3B	:IF NOT: BR
4164	017740	017703	160560		MOV	#ER,R3	:GET ERROR REGISTER
4165	017744	032737	000010	000552	BIT	#10,UDES	:SEE IF EVEN PARITY
4166	017752	001402			BEQ	ER3A1	:IF NOT: BR
4167	017754	042703	000100		BIC	#100,R3	:MASK PAR
4168	017760	042703	001000		ER3A1: BIC	#1000,R3	:MASK FCE
4169	017764	001402			BEQ	ER4	:IF NO ERRORS EXCEPT FCE: BR
4170	017766	005237	021136		ER3B: INC	CONER	:SET CONT ERROR FLAG
4171	017772	032777	040000	160522	ER4: BIT	#40000,#DS	:SEE IF DRIVE ERROR
4172	020000	001420			BEQ	ER6	:IF NOT: BR
4173	020002	005737	000676		TST	TMFLG	:SEE IF TAPE MARK TIME
4174	020006	001413			BEQ	ER4A	:IF NOT: BR
4175	020010	017703	160510		MOV	#ER,R3	:GET ER
4176	020014	032737	000010	000552	BIT	#10,UDES	:SEE IF EVEN PARITY
4177	020022	001402			BEQ	ER4A1	:IF NOT: BR
4178	020024	042703	000100		BIC	#100,R3	:MASK PAR
4179	020030	042703	001000		ER4A1: BIC	#1000,R3	:MASK OUT FCE
4180	020034	001402			BEQ	ER6	:++G & BR IF NO OTHER ERR BITS ARE SET
4181	020036	005237	021140		ER4A: INC	DRVER	:SET DRIVER ERROR FLAG
4182	020042	032737	002000	000552	ER6: BIT	#2000,UDES	
4183	020050	001071			BNE	ERPT	:IF IN PE MODE: BR

4184	020052	032777	020000	160526	BIT	#20000,BSWR	:SEE IF NO DATA CHECK
4185	020060	0011065			BNE	ERPT	:IF NOT: BR (ALLOW READ OF UNKNOWN TAPES)
4186	020062	032737	000040	000672	BIT	#40,MTC1	:SEE IF WRITE OR READ OP
4187	020070	001461			BEQ	ERPT	:IF NOT: BR
4188	020072	005737	000676		TST	TMFLG	:SEE IF TAPE MARK TIME
4189	020076	001413			BEQ	ER6A	:IF NOT: BR
4190	020100	013713	015536	021154	MOV	EXCRC,CRCV	:SAVE CRC
4191	020106	013713	015540	021152	MOV	EXLRC,LRCV	:SAVE LRC
4192	020114	005037	015536		CLR	EXCRC	
4193	020120	012737	000023	015540	MOV	#23,EXLRC	:SET CRC/LRC FOR TM
4194	020126	032737	000060	000552	BIT	#60,UDES	:SEE IF FORMAT 14
4195	020134	001037			BNE	ERPT	:IF NOT: BR
4196	020136	017700	160366		MOV	@CC,R3	:GET CRC CHARACTER
4197	020142	042700	177000		BIC	#177000,R3	
4198	020146	023700	015536		C,IP	EXCRC,R3	
4199	020152	001402			BEQ	ER7	:IF CRC GOOD: BR
4200	020154	005237	021146		INC	CRCER	:SET ERROR FLAG
4201	020160	017700	160350		MOV	@MR,R3	:GET LRC
4202	020164	000300			SWAB	R3	
4203	020166	005700			TST	R3	
4204	020170	100000			BPL	ER10	
4205	020172	052700	000400		BIS	#400,R3	
4206	020176	042700	177000		BIC	#177000,R3	
4207	020202	023700	015540		CMP	EXLRC,R3	
4208	020206	001413			BEQ	ERPT	:IF LRC GOOD: BR
4209	020210	010337	021150		MOV	R3,ACTLRC	:SAVE ACTUAL LRC
4210	020214	005237	021144		INC	LRCER	:SET LRC ERROR FLAG
4211	020220	032737	010000	000562	BIT	#10000,RDCMD	:SEE IF READ REVERSE
4212	020226	001402			BEQ	ERPT	:IF NOT: BR
4213	020230	005037	021144		CLR	LRCER	:ELSE CLEAR LRC ERROR
4214	020234	015700	000006		MOV	#6,R3	
4215	020238	005037	000706		CLR	SERFL	:CLEAR ERROR FLAG
4216	020242	005037	000722		CLR	ERSAV	
4217	020246	012704	021134		MOV	#BAER,R4	
4218	020250	005724			TST	(R4)+	:SEE IF ANY ERROR
4219	020254	001004			BNE	ERPTG	:IF SO: BR
4220	020258	005303			DEC	R3	
4221	020262	001374			BNE	ERPTT	
4222	020266	000137	021076		JMP	ERPTT	
4223	020270	000137	000706		INC	SERFL	:SET ERROR FLAG
4224	020274	017237	160224	000722	MOV	@ER,ERSAV	:SAVE ERROR REGISTER
4225	020278	032777	002000	160276	BIT	#2000,BSWR	:SEE IF PRINT
4226	020282	001420			BEQ	ERPT0	:IF SO: BR
4227	020286	022737	000002	000712	CMP	#2,RTYFL	:SEE IF READ RETRY
4228	020290	001006			BNE	ERPTG1	:IF NOT: BR
4229	020294	003700	000702		MOV	RTCNT,R3	
4230	020298	003700			INC	R3	:BUMP RETRY COUNT
4231	020302	020306	000602		CMP	R3,RETRY	:SEE IF LAST RETRY
4232	020306	001406			BEQ	ERPT0	:IF SO: BR
4233	020310	022737	000002	021140	ERPTG1: CMP	#2,DRVER	:SEE IF TM STATUS ERROR
4234	020314	001406			BEQ	ERPT0	:IF SO: BR
4235	020318	000137	021000		JMP	ERPT0	
4236	020322	003700	000670		INC	PFLG	
4237	020326	004237	022736		JSR	PC,PAPRT	:PRINT HEADER
4238	020330	013704	000652		MOV	EMADDR,R4	
4239	020334	104000			TTOUTT		:PRINT ERROR TYPE

4240	020370	004737	021156		JSR	PC,FRPRT	;PRINT F OR R
4241	020374	005737	000676		TST	TMFLG	
4242	020400	0014607			BEQ	ERPT1	
4243	020402	022737	026172	000652	CMP	#MSG54,EMADDR	
4244	020410	0014603			BEQ	ERPT1	
4245	020412	012704	026210		MOV	#MSG56,R4	;PRINT TM
4246	020416	1040000			TTOUTT		
4247	020420	005737	021136	ERPT1:	TST	CONER	
4248	020424	0014614			BEQ	ERPT2	;IF NO CONT ERROR: BR
4249	020426	012704	025213		MOV	#MSG23,R4	
4250	020432	1040000			TTOUTT		;PRINT C1 TAG
4251	020434	017703	160050		MOV	@C1,R3	
4252	020440	1040002			OCTPP		;PRINT CONTROL 1
4253	020442	012704	025240		MOV	#MSG23D,R4	;PRINT CS TAC
4254	020446	1040000			TTOUTT		
4255	020450	017703	160044		MOV	@CS,R3	
4256	020454	1040002			OCTPP		;PRINT CONT STATUS
4257	020456	005737	021140	ERPT2:	TST	DRVER	
4258	020462	0014614			BEQ	ERPT3	;IF SO DRIVE ERROR: BR
4259	020464	012704	025246		MOV	#MSG23E,R4	
4260	020470	1040000			TTOUTT		;PRINT DS TAG
4261	020472	017703	160024		MOV	@DS,R3	
4262	020476	1040002			OCTPP		;PRINT DRIVE STATUS
4263	020500	012704	025253		MOV	#MSG23F,R4	
4264	020504	1040000			TTOUTT		;PRINT ER TAG
4265	020506	017703	160012		MOV	@ER,R3	
4266	020512	1040002			OCTPP		;PRINT DRIVE ERROR
4267	020514	005737	021134	ERPT3:	TST	BAER	
4268	020520	0014616			BEQ	ERPT4	;IF NO BA ERROR: BR
4269	020522	012704	025226		MOV	#MSG23B,R4	
4270	020526	1040000			TTOUTT		;PRINT BA TAG
4271	020530	017703	157760		MOV	@BA,R3	
4272	020534	1040002			OCTPP		;PRINT BUS ADDRESS
4273	020536	012704	000255	000636	MOV	#255,TOB	
4274	020542	004737	023670		JSR	PC,TOG	;PRINT /
4275	020550	013703	021132		MOV	CADER,R3	
4276	020554	1040002			OCTPP		;PRINT EXPT BUS ADDRESS
4277	020556	005737	021142	ERPT4:	TST	FCER	
4278	020562	0014606			BEQ	ERPT5	;IF NO FC ERROR: BR
4279	020564	012704	025233		MOV	#MSG23C,R4	
4280	020570	1040000			TTOUTT		;PRINT FC TAG
4281	020572	017703	157720		MOV	@FC,R3	
4282	020576	1040002			OCTPP		;PRINT FRAME COUNT
4283	020600	012704	025221	ERPT5:	MOV	#MSG23A,R4	
4284	020604	1040000			TTOUTT		;PRINT WC TAG
4285	020606	017703	157700		MOV	@WC,R3	
4286	020612	1040002			OCTPP		;PRINT WORD COUNT
4287	020614	005737	021146		TST	CRCR	
4288	020620	0014620			BEQ	ERPT5A	;IF NO CRC ERROR: BR
4289	020622	012704	026235		MOV	#MSG58,R4	
4290	020626	1040000			TTOUTT		;PRINT CRC TAG
4291	020630	017703	157674		MOV	@CC,R3	
4292	020634	012704	177000		BIC	#177000,R3	
4293	020640	1040002			OCTPP		;PRINT ACTUAL CRC
4294	020642	012704	000255	000636	MOV	#255,TOB	
4295	020650	004737	023670		JSR	PC,TOG	

```

4296 020654 013703 015536          MOV      EXCRC,R3
4297 020660 104002          OCTPP
4298 020662 005737 021144      ERPT5A:  TST      LRCER      ;PRINT EXPECTED CRC
4300 020670 001416          BEQ      ERPT6      ;IF NO LRC ERROR: BR
4301 020672 012704 026243      MOV      #MSG59,R4
4302 020674 104000          TTOUTT   ;PRINT LRC TAG
4303 020676 013703 021150      MOV      ACTLRC,R3
4304 020678 104002          OCTPP      ;PRINT ACTUAL LRC
4305 020704 012737 000255 000636      MOV      #255,TOB
4306 020712 004737 023670      JSR      PC,TOG
4307 020716 013703 015540      MOV      EXLRC,R3
4308 020722 104002          OCTPP      ;PRINT EXPECTED LRC
4309 020724 005737 021140      ERPT6:  TST      DRIVER
4310 020726 001416          BEQ      ERPT7      ;IF NO DRIVE ERROR: BR
4311 020728 003277 002000 000552      BIT      #2000,UDES
4312 020730 001416          BEQ      ERPT7      ;IF NO PE: BR
4313 020732 017704 157556      MOV      @ER,R4
4314 020734 002704 075477      BIC      #75477,R4 ;MASK OUT ALL BUT BITS 15,10,7,6
4315 020736 003704          TST      R4
4316 020738 001410          BEQ      ERPT7      ;IF NO CONDITIONALS SET: BR
4317 020740 012704 025265      MOV      #MSG23H,R4
4318 020742 104000          TTOUTT   ;PRINT CC TAG
4319 020744 017704 157540      MOV      @CC,R3
4320 020746 042703 177000      BIC      #177000,R3 ;MASK CC
4321 020748 104002          OCTPP      ;PRINT CHECK CHARACTERS
4322 020750 000277          NOP
4323 020752 003277 100000 157600      ERPT7:  ERPX0:  BIT      #100000,@SWR ;SEE IF STOP ON ERROR
4324 020754 001412          BEQ      EP^X      ;IF NOT: BR
4325 020756 104002          STOPP
4326 020758 005737 000670          TST      PFLG      ;SEE IF HAVE PRINTED
4327 020760 001006          BNE      ERPX      ;IF SO: BR
4328 020762 003277 002000 157560      BIT      #2000,@SWR ;SEE IF SHOULD PRINT
4329 020764 001002          BNE      ERPX      ;IF NOT: BR
4330 020766 000137 020352      JMP      ERPT0     ;PRINT ERROR
4331 020768 005037 000670          CLR      PFLG
4332 020770 012777 000011 157442      ERPX:   MOV      #11,@C1 ;DRIVE CLEAR
4333 020772 017704 157454      MOV      @AS,R4
4334 020774 010477 157450      MOV      R4,@AS   ;CLEAR AS
4335 020776 013704 000510      MOV      C1,R4
4336 020778 003204          INC      R4
4337 020780 152714 000100          BISB    #100,(R4) ;RESET TRE
4338 020782 013777 000552 157444      MOV      UDES,@C2 ;RESET TC
4339 020784 032737 000040 000672      ERPX1:  BIT      #40,RTC1
4340 020786 001411          BEQ      ERPX2
4341 020788 005737 000676          TST      TMFLG
4342 020790 001406          BEQ      ERPX2   ;IF NOT TM TIME: BR
4343 020792 013737 021154 015536      MOV      CRCSV,EXCRC ;RESTORE CRC
4344 020794 013737 021152 015540      MOV      LRCSV,EXLRC ;RESTORE LRC
4345 020796 000207          RTS
4346 020798 000000      ERPX2:  CADER:  0
4347 020800 000000      BAER:   0
4348 020802 000000      CONER:  0
4349 020804 000000      DRIVER: 0
4350 020806 000000      FCER:   0
4351 020808 000000      LRCER:  0
4352 020810 000000      CRCER:  0
    
```

4352 021150 000000
4353 021152 000000
4354 021154 000000
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375

ACTLRC: 0
LRCSV: 0
CRCSV: 0

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

021156	032737	000010	000672	FRPRT:	BIT	#10,MTC1	:SEE IF WRITE COMMAND
021164	001413				BEQ	FREX	:IF SO: BR
021166	032737	000002	000672		BIT	#2,MTC1	:SEE IF REVERSE
021174	001404				BEQ	FRO	:IF NOT: BR
021176	012704	025123			MOV	#MSG17,R4	
021202	104000				TTOUTT		:PRINT R
021204	000403				BR	FREX	
021206	012704	025120		FRO:	MOV	#MSG16,R4	
021212	104000				TTOUTT		:PRINT F
021214	000207			FREX:	RTS	PC	:EXIT

4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431

021216 005037 000642
021222 013777 000550 157270
021230 032777 010000 157264
021236 00 026
021240 005237 000642
021244 001371
021246 004737 022736
021252 032737 000010 000672
021260 001006
021262 012704 024762
021266 104000
021270 000405
021272 012704 024767
021276 104000
021280 004737 021156
021284 012704 025343
021290 104000
021294 104006
021298 032777 020000 157200
021302 001411
021306 004737 022736
021310 012704 027322
021314 104000
021318 032777 020000 157156
021322 001374
021326 022737 000026 000672
021334 001003

TAPG: CLR TEMP1
MOV DVN,BCS :SET DRIVE NO.
BIT #10000,ADS :SEE IF HAVE MOL
B'E TAPG3 :IF SO: BR
I TEMP1 :SEE IF TIMED OUT
BNE TAPG0 :WAIT FOR READY
JSR PC,PAPRT :PRINT CYCLE NUMBER
BIT #10,MTC1 :SEE IF WRITE OP
BNE TAPG1 :IF NOT: BR
MOV #MSG5,R4
TTOUTT :PRINT WRITE ERR
BR TAPG2
TAPG1: MOV #MSG6,R4
TTOUTT :PRINT READ ERR
JSR PC,FRPRT :PRINT F OR R
TAPG2: MOV #MSG25,R4
TTOUTT :PRINT NO MOL ERR
STOPP
TAPG3: BIT #20000,ADS :SEE IF PIP RESET
BEQ TAPG3F :IF SO: BR
JSR PC,PAPRT :PRINT HEADER
MOV #MSG116,R4
TTOUTT :PRINT REWINDING MESSAGE
1\$: BIT #20000,ADS
BNE 1\$:AWAIT PIP RESET
TAPG3F: C'P #26,MTC1 :SEE IF WRITE TM
BNE TAPG3A :IF NOT: BR

:TAPE COMMAND EXECUTE SUBROUTINE:
:
:THIS SUBROUTINE IS USED TO EXECUTE THE
:MAG TAPE COMMAND DESCRIBED BY THE READ
:OR WRITE ROUTINE. THE FINAL COMMAND IS
:SENT TO THE DEVICE REGISTER ALCO WITH THE
:INTERRUPT ENABLE AND GO BITS.
:ONCE THE COMMAND IS ISSUED, AN INTERRUPT
:TIMER IS STARTED AND IF NO INTERRUPT IS RETURNED
:BEFORE TIME OUT OCCURS, AN ERROR WILL BE
:PRINTED AND THE PROGRAM STOPPED. TESTING MAY
:BE RESUMED BY PRESSING THE CONTINUE SWITCH.
:TWO INTERRUPT HANDLERS ARE USED, ONE FOR MAG TAPE
:AND ANOTHER FOR TELETYPE (TTY).
:UPON RECEIPT OF A MAG TAPE INTERRUPT, HOUSEKEEPING
:IS PERFORMED AND CONTROL RETURNED TO THE CALLING
:ROUTINE (READ WRITE, ETC).
:RECEIPT OF A TTY INTERRUPT WILL CAUSE THE
:PROGRAM TO CHECK FOR ENTRY OF A CNTRL C CHARACTER.
:IF NOT CNTRL C, THEN CONTINUATION OF WAIT FOR MAG
:TAPE INTERRUPT IS RETURNED. IF, HOWEVER, THE TTY
:INTERRUPT WAS CAUSED BY ENTRY OF A CNTRL C,
:THEN AT THIS TIME REQUESTS FOR NEW STALL VALUES
:ARE PRINTED AND THE RESPONSES ENTERED. RESUMPTION
:OF TAPE INTERRUPT WAIT IS THEN RESUMED.

4432	021356	012704	177777		MOV	#-1,R4	:ELSE SET FC FOR -1
4433	021356	000406			BR	TAPG3B	
4434	021356	013704	000556		TAPG3A: MOV	FMCNT,R4	
4435	021370	032704	000001		BIT	#1,R4	
4436	021374	001401			BEQ	TAPG3B	
4437	021376	005304			DEC	R4	
4438	021400	000261			TAPG3B: SEC		
4439	021402	006004			ROR	R4	:SET WC = FC/2 FOR NORMAL FORMAT
4440	021404	032733	000020	000552	BIT	#20,UDES	:SEE IF CORE DUMP FORMAT
4441	021412	001402			BEQ	TAPG3C	:IF NOT: BR
4442	021414	000261			SEC		
4443	021416	006004			ROR	R4	:SET WC = FC/4 FOR CORE DUMP
4444	021420	010477	157066		TAPG3C: MOV	R4,@WC	:SET WORD COUNT
4445	021424	012777	000011	157056	MOV	#1,@C1	:DRIVE CLEAR
4446	021432	017777	157060	157056	MOV	@FC,@FC	:RESET FC LOADED
4447	021440	005737	000566		TST	INTR	:SEE IF INTERCHANGE READ
4448	021444	001407			BEQ	TAPG3D	:IF NOT: BR
4449	021446	032737	000040	000672	BIT	#40,MTC1	:SEE IF READ OP
4450	021454	001403			BEQ	TAPG3D	:IF NOT: BR
4451	021456	012777	000003	157050	MOV	#3,@MR	:SET INTERCHANGE READ MAINT. MODE
4452	021464	013704	000672		TAPG3D: MOV	MTC1,R4	:GET COMMA
4453	021470	042704	177707		BIC	#177707,R4	:MASK OP CODE
4454	021474	022704	000030		CMP	#30,R4	:SEE IF SPACE OP CODE
4455	021500	001403			BEQ	TAPG3E	:IF SO: BR
4456	021502	012737	177740	000666	MOV	#-40,STAL	:SET INTERRUPT DELAY MULT TO 40
4457	021510	052737	000101	000672	TAPG3E: BIS	#101,MTC1	:SET INTERRUPT ENABLE AND GO
4458	021516	000240			NOP		
4459	021520	013777	000672	156762	MOV	MTC1,@C1	:EXECUTE COMMAND
4460	021526	005077	157052		CLR	@PSW	:CLEAR PRIORITY
4461	021532	005037	000642		CLR	TEMP1	
4462	021536	005237	000642		TAPG4: INC	TEMP1	:SEE IF HAVE TIMED OUT
4463	021542	001377			BNE	TAPG4	:IF NOT: BR
4464	021544	005237	000666		INC	STAL	
4465	021550	001377			BNE	TAPG4	:DO TIME DELAY MULTIPLIER
4466	021552	012777	000340	157024	TAPG5: MOV	#340,@PSW	:RESET PRIORITY
4467	021560	032777	002000	157020	BIT	#2000,@SWR	:SEE IF SHOULD PRINT ERRORS
4468	021566	001012			BNE	TAPG6	:IF NOT: BR
4469	021570	004737	022736		JSR	PC,PAPRT	:PRINT CYCLE NUMBER
4470	021574	013704	000652		MOV	EMADDR,R4	
4471	021600	104000			TTOUTT		:PRINT ERROR OP
4472	021602	004737	021156		JSR	PC,FRPRT	:PRINT F OR R
4473	021606	012704	025323		MOV	#MSG24,R4	
4474	021612	104000			TTOUTT		:PRINT NO INTERRUPT
4475	021614	005777	156766		TAPG6: TST	@SWR	:++C BRANCH IF CONTINUE ON ERROR
4476	021620	100001			BPL	TAPG7	
4477	021622	104000			STOPP		
4478	021624	C00137	021716		TAPG7: JMP	MTINTA	:RETURN TO CALLING ROUTINE
4479							

```
4480  
4481  
4482  
4483 021630 012777 000340 156746 TTINT: MOV #340, @PSW ;RESET PSW  
4484 021636 017746 156750 MOV @1KB, -(SP) ;++G GET CHARACTER  
4485 021642 042716 000200 BIC #200, (SP) ;++G STRIP PARITY BIT  
4486 021648 122716 000003 CMPB #3, (SP) ;++G SEE IF CONT C  
4487 021652 001412 BEQ TTINTO ;IF SO: BR  
4488 021654 122716 000007 CMPB #7, (SP) ;++G CHECK FOR CNTL G  
4489 021660 001013 BNE RETURN  
4490 021662 022737 000176 000606 CMP #SWREG, SWR ;IS SOFTWARE SWITCH REGISTER USED  
4491 021670 001007 BNE RETURN ;NO, GET OUT  
4492 021672 004737 024452 JSR PC, CNTG ;GO CHANGE SWREG  
4493 021676 000404 BR RETURN ;++G GO TO EXIT  
4494  
4495 021700 010046 TTINTO: MOV RO, -(SP) ;++G SAVE RO(REC CNTR)  
4496 021702 004737 013774 JSR PC, TINTP4 ;GO GET STALL VALUES  
4497 021706 012600 MOV (SP)+, RO ;++G RESTORE RO(REC CNTR)  
4498 021710 005726 RETURN: TST (SP)+ ;++G POP CHAR OFF STACK  
4499 021712 000002 RTI ;RETURN  
4500  
4501 ;MAG TAPE INTERRUPT HANDLER*****  
4502  
4503 021714 000240 MTINT: NOP  
4504 021716 042777 000037 156610 MTINTA: BIC #37, @MR ;CLEAR MAINT MODE  
4505 021724 013716 000662 MOV RTRN, (SP) ;++G GET RETURN ADDRESS  
4506 021730 000002 RTI ;++G RETURN
```

4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
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

021732
021733
021734
021735
021736
021737
021738
021739
021740
021741
021742
021743
021744
021745
021746
021747
021748
021749
021750
021751
021752
021753
021754
021755
021756
021757
021758
021759
021760
021761
021762
021763
021764
021765
021766
021767
021768
021769
021770
021771
021772
021773
021774
021775
021776
021777
021778
021779
021780
021781
021782
021783
021784
021785
021786
021787
021788
021789
021790
021791
021792
021793
021794
021795
021796
021797
021798
021799
021800
022160
022161
022162
022163
022164
022165
022166
022167
022168
022169
022170
022171
022172
022173
022174
022175
022176
022177
022178
022179
022180
022181
022182
022183
022184
022185
022186
022187
022188
022189
022190
022191
022192
022193
022194
022195
022196
022197
022198
022199
022200

```

:*****
: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      ;PRINT NRZ ONLY REQUEST
      TTOUTT                               ;SET ADDRESS OF FLAG
      MOV    #NRZOF,R5      ;SET SIZE OF ENTRY
      MOV    #1,R1          ;SET UPPER LIMIT
      MOV    #1,R2          ;SET LOWER LIMIT
      MOV    #0,R3          ;GO GET RESPONSE
      JSR    PC,TTR
      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    B#42           ;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$:  MOV    #MSG101,R4
      TTOUTT                               ;PRINT DIVIDER
      MOV    #MSG102,R4
      TTOUTT                               ;PRINT TM02 NUMBER
      MOV    ADRVN,R3
      OCTPP                              ;PRINT TMUC
      MOV    #MSG103,R4
      TTOUTT                               ;PRINT SLAVE HDR
2$:  MOV    #UH1,R0          ;POINT TO START OF SLAVE TABLE
ASFQ2: TST    (R0)          ;SEE IF END
      BMI    ASEQ3         ;IF SO: BR
      MOV    (R0)+,R3
      OCTPP                              ;PRINT SLAVE TABLE
      BR     ASEQ2         ;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    ASEQCF        ;CONTINUE
      BNE    ASEQ0         ;GO START AGAIN
1$:  HALT

```



```
4562  
4563  
4564  
4565  
4566  
4567  
4568  
4569  
4570  
4571  
4572  
4573  
4574  
4575  
4576  
4577  
4578  
4579  
4580  
4581  
4582  
4583  
4584  
4585  
4586  
4587  
4588  
4589  
4590  
4591  
4592  
4593  
4594  
4595  
4596  
4597  
4598  
4599  
4600  
4601  
4602
```

:SUBROUTINE TO SEIECT AUTO SEQUENCE HARDWARE*****

```
0222162 005037 005156 HRDS: CLR REOTC ;CLEAR EOT UNIT CNTR  
0222162 005037 000642 CLR TEMP1  
0222162 012777 000640 156320 MOV #40,BCS ;INIT  
0222162 012777 000736 156312 MOV ADRVN,BCS ;SET DRIVE  
0222162 012701 156324 MOV @DT,R1 ;READ DRIVE TYPE  
0222162 012777 010000 156300 BIT #10000,BCS ;TEST FOR NON-EXISTANT DRIVE  
0222162 001403 BEQ HRDS1 ;IF DRIVE AVAIL: BR  
0222162 000137 HRDS0: TST (SP)+ ;RESET STACK POINTER  
0222162 042701 022122 JMP ASEQ4 ;GO SEE IF TRIED ALL DRIVES  
0222162 042701 002007 HRDS1: BIC #2007,R1 ;MASK SLAVE TYPE  
0222162 042701 140010 CMP #140010,R1 ;++G SEE IF TU16/TE16 TAPE  
0222162 001370 BNE HRDS0 ;IF NOT: BR  
0222162 005000 CLR R0  
0222162 012701 000746 MOV #UN1,R1 ;SET START OF SLAVE TABLE  
0222162 005737 003040 TST CHNFLG ;++G BRANCH IF NOT IN CHAIN MODE  
0222162 001410 BEQ HRDS2 ;++G BRANCH IF NOT LOADED VIA TMDP  
0222162 122737 000006 000041 CMPB #6,@#41  
0222162 001004 BNE HRDS2 ;++G BRANCH IF NOT DRIVE 0  
0222162 005737 000736 TST ADRVN ;++G  
0222162 001001 BNE HRDS2 ;++G DO NOT TEST DRIVE 0 SLAVE 0  
0222162 005200 INC R0 ;++G IF TMDP CHAIN  
0222276 010077 156240 HRDS2: MOV R0,@C2 ;SELECT SLAVE  
0222302 032777 010000 156212 BIT #10000,@DS ;SEE IF SLAVE AVAIL FOR TEST(MOL)  
0222310 001403 BEQ HRDS3 ;IF NOT: BR  
0222312 005237 000642 INC TEMP1 ;SET SLAVE FOUND FLAG  
0222316 010021 MOV R0,(R1)+ ;LOAD SLAVE TABLE  
0222320 022700 000007 HRDS3: CMP #7,R0 ;SEE IF DONE ALL SLAVES  
0222324 001402 BEQ HRDS4 ;IF SO: BR  
0222326 005200 INC R0 ;ELSE BUMP SLAVE NUMBER  
0222330 000762 BR HRDS2 ;CONTINUE SELECTION  
0222332 005737 000642 HRDS4: TST TEMP1 ;SEE IF FOUND ANY SLAVES  
0222336 001731 BEQ HRDS0 ;IF NOT: BR  
0222340 013737 000642 005156 MOV TEMP1,REOTC ;SET NUMBER OF UNITS  
0222346 000337 000642 SWAB TEMP1  
0222352 053737 000642 005156 BIS TEMP1,REOTC ;SET EOT CNTR  
0222360 012711 177777 MOV #-1,(R1) ;TERMINATE SLAVE TABLE  
0222364 000207 RTS PC ;RETURN TO SEQ
```

```

4603
4604
4605 ;SUBROUTINE TO SELECT NRZ AUTO TEST MODE*****
4606 022366 005037 000654 AMOD1: CLR BLCNTR ;ASSURE BLOCK COUNTER IS 0
4607 022372 012701 000746 AMOD1: MOV #UN1,R1 ;GET START OF SLAVE TABLE
4608 022376 052721 001700 AMOD1A: BIS #1700,(R1)+ ;SET ALL SLAVE TO NRZ,NORM,ODD
4609 022402 005111 ;
4610 022404 001402 ;
4611 022406 005111 ;
4612 022410 000772 ;
4613 022412 005111 ;
4614 022414 004737 005172 AMOD1B: COM AMOD1A ;IF FILLED ALL SLAVES: BR
4615 022420 012737 000006 000740 JSR PC,RWDA ;GO REWIND ALL AVAIL SLAVES
4616 022426 012737 174000 003556 MOV #6,ABLCNT ;SET NUMBER OF BLOCKS FOR MODE 1
4617 022432 012737 000100 003554 MOV #-4000,FCNT ;SET FC = 4000
4618 022438 012737 000736 003550 MOV #100,RCNT ;SET REC CNTR = 100
4619 022442 012737 000001 000560 MOV ADRVN,DVN ;SELECT DRIVE
4620 022446 005037 000564 CLR TMEX ;SELECT PATTERN 1
4621 022450 005037 000566 CLR INTRF ;ASSURE NO TMK
4622 022454 004737 000010 000560 JSR PC,STAUTO ;ASSURE NORMAL READ
4623 022458 004737 003414 000560 MOV #10,PATRN ;GO DO AUTO MODE 1
4624 022500 012737 000014 000560 JSR PC,STAUTO ;SELECT PATTERN 10
4625 022504 012737 003414 000560 MOV #14,PATRN ;GO DO PATTERN 10
4626 022512 004737 003414 JSR PC,STAUTO ;SELECT PATTERN 14
4627 022516 005737 000650 TST NRZOF ;SEE IF NRZ ONLY
4628 022522 001411 BEQ AMOD1C ;IF NOT: BR
4629 022526 012737 177777 000740 MOV #-1,ABLCNT ;FORCE TO EOT
4630 022532 012737 153624 000624 MOV #153624,RANBAS
4631 022540 012737 032561 000626 MOV #32561,RANSAV ;RESET RANDOM DATA BASE
4632 022546 012737 177777 000560 AMOD1C: MOV #-1,PATRN ;SELECT AUTO RANDOM DATA
4633 022554 004737 003414 JSR PC,STAUTO
4634 022560 000207 RTS ;RETURN TO SEQ
  
```

```
4635  
4636  
4637  
4638 022562 005737 003040  
4639 022566 001003  
4640 022570 005737 000650  
4641 022574 001057  
4642 022576 005037 000654  
4643 022602 012701 000746  
4644 022606 042711 001700  
4645 022612 052721 002300  
4646 022616 005111  
4647 022620 001402  
4648 022622 005111  
4649 022624 005770  
4650 022626 005111  
4651 022630 004737 005172  
4652 022634 012737 000006 000740  
4653 022642 012737 174000 000556  
4654 022650 012737 000100 000554  
4655 022656 012737 000010 000560  
4656 022664 004737 003414  
4657 022670 012737 000014 000560  
4658 022676 004737 003414  
4659 022702 012737 000015 000560  
4660 022710 004737 003414  
4661 022714 012737 177777 000740  
4662 022722 012737 177777 000560  
4663 022730 004737 003414  
4664 022734 000207  
4665  
4666
```

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

4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722

022736 012704 025040
022742 104000
022744 013703 000550
022750 104002
022752 012704 025024
022756 104000
022760 013703 000552
022764 042703 177770
022770 104002
022772 012704 026251
022776 104000
023000 013703 000552
023004 000303
023006 042703 177770
023012 104002
023014 012704 026255
023020 104000
023022 005003
023024 032737 000010 000552
023032 001402
023034 012703 000001
023040 104002
023042 012704 026261
023046 104000
023050 013703 000552
023054 000241
023056 006003
XXXXXX
023062 006003
023064 006003
023066 042703 177760
023072 104002
023074 012704 025001
023100 104000
023102 032777 000400 155476
023110 001406
023112 012737 000122 000636
023120 004737 023670
023124 000411
023126 005737 000734

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

PAPRT: MOV #MSG12,R4
TTOUTT ;PRINT DRIVE HEADER
MOV DVN,R3
OCTPP ;PRINT DRIVE NUMBER
MOV #MSG11,R4
TTOUTT ;PRINT UNIT HEADER
MOV UDES,R3
BIC #177770,R3
OCTPP ;PRINT UNIT NUMBER
MOV #MSG60,R4
TTOUTT ;PRINT DENSITY TAG
MOV UDES,R3
SWAB R3
BIC #177770,R3
OCTPP ;PRINT DENSITY
MOV #MSG61,R4
TTOUTT ;PRINT PARITY TAG
CLR R3
BIT #10,UDES
BEQ PAPRT0
MOV #1,R3
PAPRT0: OCTPP ;PRINT PARITY
MOV #MSG62,R4
TTOUTT ;PRINT FORMAT TAG
MOV UDES,R3
CLC
ROR R3
ROR R3
ROR R3
ROR R3
BIC #177760,R3
OCTPP ;PRINT FORMAT
MOV #MSG8,R4
TTOUTT ;PRINT PATRN TAG
BIT #400,@SWR ;SEE IF RANDOM DATA
BEQ PAPRTB ;IF NOT: BR
MOV #122,TOB
JSR PC,TOB ;PRINT R
BR PAPRTD
PAPRTB: TST ASEOF ;SEE IF AUTO SEQ

4723	023132	001403									
4724	023134	005737	000560								
4725	023140	100764									
4726	023142	013703	000560								
4727	023146	104002									
4728	023150	012704	025054								
4729	023154	104000									
4730	023156	013703	000654								
4731	023162	104002									
4732	023167	012704	025062								
4733	023170	104000									
4734	023172	010003									
4735	023174	032737	000010	000672							
4736	023202	001404									
4737	023204	032737	010000	000562							
4738	023212	001016									
4739	023216	013703	000554								
4740	023220	005737	000676								
4741	023224	001010									
4742	023226	022737	012260	000662							
4743	023234	001003									
4744	023238	005737	000712								
4745	023242	001402									
4746	023244	160003									
4747	023246	005203									
4748	023250	104002									
4749	023252	012737	000055	000636							
4750	023258	004737	023670								
4751	023260	013703	000554								
4752	023270	104002									
4753	023272	012704	024774								
4754	023276	104000									
4755	023300	013703	000556								
4756	023304	005303									
4757	023306	005103									
4758	023310	104002									
4759	023312	012737	000001	000664							
4760	023320	000207									
4761											

```

BEQ PAPER:
TST PATRN :IF NOT: BR
BMI PAPER: :SEE IF AUTO RANDOM
PAPER: MOV PAPER,R3 :IF SO: BR
OCTPP :PRINT PATRN NUMBER
PAPER: MOV #MSG13,R4
TTOUTT :PRINT BLOCK NO. HEADER
MOV BLCNTR,R3
OCTPP :PRINT NUMBER
MOV #MSG14,R4
TTOUTT :PRINT REC NO. HEADER
MOV R0,R3
BIT #10,MTC1 :SEE IF WRITE OPERATION
BEQ PAPER: :IF SO: BR
BIT #10000,RDCMD :SEE IF READ REVERSE
PAPER: MOV PAPER: :IF SO: BR
RCNT,R3
TST TMFLG :SEE IF TAPE MARK TIME
BNE PAPER: :IF SO: BR
CMP #B1,RTRN
BNE PAPER: :IF NOT BACK SPACE: BR
TST RTYFL
BEQ PAPER: :IF NOT RETRY: BP
PAPER: SUB R0,R3 :GET RECORD NUMBER
PAPER: INC R3
PAPER: OCTPP :PRINT RECORD NUMBER
MOV #55,TOB :LOAD DASH (-)
JSR PC,TOG :PRINT DASH (-)
MOV RCNT,R3
OCTPP :PRINT RECORD COUNT
MOV #MSG7,R4
TTOUTT :PRINT RECORD SIZE HEADER
MOV FMCNT,R3 :GET CHARACTER COUNT
DEC R3
COM R3 :REMOVE TWOS COMPLEMENT
OCTPP :PRINT RECORD SIZE
MOV #1,HDRFL :SET HEADER FLAG
RTS PC :RETURN
  
```

4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778

023322	063737	000626	000624	RANG:
023330	063737	000624	000626	
023336	023701	000626		
023342	101367			
023344	020237	000626		
023350	101364			
023352	000207			

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

4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829

023354 005037 000642
023360 005000
023362 104010
023364 122737 000015 000640
023372 001004
023374 005737 000642
023400 001436
023402 000426
023404 122737 000060 000640
023412 101401
023414 000431
023416 122737 000070 000640
023424 101001
023426 000424
023430 005237 000642
023434 006300
023436 006300
023440 006300
023442 042737 177770 000640
023450 053700 000640
023454 005301
023456 001341
023460 020002
023462 101401
023464 000405
023466 020300
023470 101401
023472 000402
023474 010015
023476 000207
023500 012704 025763
023504 104000
023506 162716 000020
023512 000207

```
*****  
:TTY ENTRY SUBROUTINE:  
:THIS SUBROUTINE IS USED BY THE TEST CONDITION  
:ENTRY ROUTINE TO READ THE RESPONSE ENTERED  
:A THE TTY AND CHECK THEM FOR LEGALITY AND  
:LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL  
:(0-7) AND MUST FALL WITHIN THE LIMITS SET BY  
:THE CALLING ROUTINE.  
:IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,  
:A QUESTION MARK IS TYPED (?) AND THE RESPONSE  
:MAY BE REENTERED.  
:ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND  
:MAY BE TERMINATED AT LESS THAN SIX BY TYPING A  
:CARRIAGE RETURN  
*****  
TTR: CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG  
CLR RO  
TTR0: TTINN ;GO READ CHARACTER  
CMPB #15,TIB ;++G SEE IF CR  
BNE TTR1 ;IF NOT: BR  
TST TEMP1 ;SEE IF FIRST CHARACTER  
BEQ TTR5 ;IF SO: BR  
BR TTR2 ;++G ELSE GO LOAD VALUE  
000060 000640 TTR1: CMPB #60,TIB ;++G SEE IF CHAR IS LESS THAN 0  
BLOS TTR1A ;IF NOT: BR  
BR TTR1 ;++G ELSE GO TO ERROR  
000070 000640 TTR1A: CMPB #70,TIB ;++G SEE IF CHAR IS GREATER THAN 7  
BHI TTR1B ;IF NOT: BR  
BR TTR1 ;++G ELSE GO TO ERROR  
000642 TTR1B: INC TEMP1 ;SET FIRST CHARACTER FLAG  
ASL RO  
ASL RO ;SHIFT 3 LEFT  
ASL RO  
000640 000640 BIC #177770,TIB ;STRIP ASCII  
BIS TIB,RO ;LOAD CHARACTER  
DEC R1 ;SEE IF DONE  
BNE TTR0 ;IF NOT: BR  
000642 TTR2: CMP RO,R2 ;SEE IF EXCEEDED MAXIMUM LIMIT  
BLOS TTR3 ;IF NOT: BR  
BR TTR1 ;++G ELSE GO TO ERROR  
000640 TTR3: CMP R3,RO ;SEE IF BELOW MINIMUM LIMIT  
BLOS TTR4 ;IF NOT: BR  
BR TTR1 ;++G ELSE GO TO ERROR  
000640 TTR4: MOV RO,(R5) ;LOAD VALUE  
TTR5: RTS PC ;EXIT  
000640 TTR5: MOV #MSG43,R4  
TINER: TTOUTT ;PRINT?  
SUB #20,(SP) ;RESET SP TO START OF VALUE ROUTINE  
RTS PC ;REDO VALUE ENTRY
```

4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875
4876

023514 005277 155070
023520 105777 155064
023526 100375
023532 017737 155060 000640
023538 042737 000200 000640
023544 105777 155048
023550 100375
023556 113777 000640 155040
000207

023560 112437 000636
023564 122737 000043 000636
023572 001444
023576 122737 000045 000636
023580 001407
023584 122737 000041 000636
023588 001435
023592 004737 023670
023596 000757
023600 112737 000015 000636
023604 004737 023670
023608 012703 000006
023612 005303 000636
023616 004737 023670
023620 005303
023624 001372
023628 112737 000012 000636
023632 004737 023670
023636 000734

023670 105777 154720
023674 100375
023676 113777 000636 154712
023704 000207
023706 012703 000002
023712 012737 000007 000636
023720 004737 023670
023724 005303
023726 001371
023730 000713

:TTY READ SUBROUTINE*****
TTIN: INC @TKS
ITIN1: TSTB @TKS
BPL TTI1
MOV @TKB,TIB
BIC #200,TIB ;++G STRIP PARITY BIT
TTIN2: TSTB @TPS
BPL TTI2
MOVB TIB,@TPB
RTS PC

:TTY OUTPUT SUBROUTINE*****
TTOUT: MOVB (R4)+,TOB
CMPB #43,TOB
BEQ TEX
CMPB #45,TOB
BEQ TCRLF
CMPB #41,TOB
BEQ TBELL ;DO BELL
JSR PC,TOG
BR TTOUT
TCRLF: MOVB #15,TOB
JSR PC,TOG
MOV #6,R3
TCRLFA: CLR TOB
JSR PC,TOG
DEC R3
BNE TCRLFA ;DO FILLERS
MOVB #12,TOB
JSR PC,TOG
BR TTOUT

TOG: TSTB @TPS
BPL TOG
MOVB TOB,@TPB
TEX: RTS PC
TBELL: MOV #2,R3
TBELA: MOV #7,TOB
JSR PC,TOG
DEC R3
BNE TBELA
BR TTOUT


```

4877                                     :OCTAL OUTPUT SUBROUTINE*****
4878
4879 023732 005037 024164      OCTP:  CLR   OFL           ;CLEAR FLAG FOR LEADING ZERO
4880 023733 000403
4881 023734 012737 000001 024164  OCTPE: MOV   #1,OFL
4882 023735 010304      OCTPE1: MOV   R3,R4           ;SEE IF NUMBER IS ZERO
4883 023736 001006      OCTPE1: BNE   OCTP0           ;IF NOT ZERO: BR
4884 023737 005737 024164      TST   OFL
4885 023738 001003      BNE   OCTP0
4886 023739 004737 024144      JSR   PC,OCTPG1        ;ELSE PRINT ZERO
4887 023740 000450      BR    OCTP3           ;++G SPACE AND EXIT
4888 023741 032704 100000      OCTP0: BIT   #100000,R4 ;SEE IF MSD = 1
4889 023742 001406      BEQ   OCTP1           ;IF NOT: BR
4890 023743 012704 000001      MOV   #1,R4
4891 024000 004737 024122      JSR   PC,OCTPG        ;PRINT 1
4892 024001 000137 024016      JMP   OCTP2
4893 024010 005004      OCTP1: CLR   R4
4894 024011 004737 024122      JSR   PC,OCTPG        ;PRINT 0
4895 024012 010304      OCTP2: MOV   R3,R4
4896 024013 006004      ROR   R4
4897 024014 006004      ROR   R4
4898 024015 006004      ROR   R4           ;POSITION DIGIT
4899 024016 006004      ROR   R4
4900 024017 000304      SWAB  R4
4901 024018 004737 024122      JSR   PC,OCTPG        ;PRINT DIGIT 2
4902 024019 010304      MOV   R3,R4
4903 024020 006004      ROR   R4
4904 024021 000304      SWAB  R4
4905 024022 004737 024122      JSR   PC,OCTPG        ;PRINT DIGIT 3
4906 024023 010304      MOV   R3,R4
4907 024024 006104      ROL   R4
4908 024025 006104      ROL   R4
4909 024026 000304      SWAB  R4
4910 024027 004737 024122      JSR   PC,OCTPG        ;PRINT DIGIT 4
4911 024028 010304      MOV   R3,R4
4912 024029 006004      ROR   R4
4913 024030 006004      ROR   R4
4914 024031 006004      ROR   R4
4915 024032 004737 024122      JSR   PC,OCTPG
4916 024100 010304      MOV   R3,R4
4917 024101 004737 024122      JSR   PC,OCTPG        ;PRINT DIGIT 5
4918 024102 012737 000240 000636  OCTP3: MOV   #240,TOB
4919 024103 004737 023670      JSR   PC,TOG
4920 024104 000207      RTS   PC           ;PRINT SPACE
4921 024105 042704 177770      OCTPG: BIC   #177770,R4 ;EXIT
4922 024106 001004      BNE   OCTPG0
4923 024107 005737 024164      TST   OFL
4924 024108 001001      BNE   OCTPG0
4925 024109 000207      RTS   PC
4926 024110 003237 024164      OCTPG0: INC  OFL
4927 024111 052704 000260      OCTPG1: BIS   #260,R4
4928 024112 010437 000636      MOV   R4,TOB
4929 024113 004737 023670      JSR   PC,TOG
4930 024114 010304      MOV   R3,R4
4931 024115 000207      RTS   PC
4932 024116 000000      UFL:  0           ;FIRST CHAR FLAG
  
```

4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988

024166 005037 000636
024172 012704 000010
024176 110337 000636
024202 105777 154406
024206 100375
024210 132737 000200 000636
024216 105737 000636
024222 100004
024226 012777 000061 154364
024232 000403
024236 012777 000060 154354
024242 006137 000636
024246 005304
024250 001354
024254 000307
024260 000303 000646
024262 004737 024166
024266 013703 000646
024272 004737 024166
024276 000207

000636

154364

154354

000646

024166

000646

024166

024356

024356

024356

024356

000260 000636

042704 177760

050437 000636

004737 023670

000207

:DATA CHARACTER OUTPUT SUBROUTINE*****

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

:++G TU16/TE16 SERIAL NUMBER PRINT SUBROUTINE*****

```
SNPT: MOV R3,R4
SWAB R4
ROR R4
ROR R4
ROR R4
ROR R4
ROR R4
JSR PC,SNPG ;PRINT FIRST DIGIT
MOV R3,R4
SWAB R4
JSR PC,SNPG ;PRINT SECOND DIGIT
MOV R3,R4
ROR R4
ROR R4
ROR R4
ROR R4
ROR R4
JSR PC,SNPG ;PRINT THIRD DIGIT
MOV R3,R4
JSR PC,SNPG ;PRINT FOURTH DIGIT
RTS PC
SNPG: MOV #260,TOB ;SET NUMBER BASE
BIC #177760,R4 ;MASK NUMBER
BIS R4,TOB ;BUILD DIGIT
JSR PC,TOG ;GO TYPE
RTS PC ;RETURN
```

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

4989					: LOCATIONS USED:	
4990	024402	000000			TEMPST: .WORD	0
4991	024404	000000			COUNT: .WORD	0
4992	024406	000000			RDSW: .WORD	0
4993	024410	022737	J00176	000606	CKSWR: CMP	#SWREG,SWR
4994	024416	001123			BNE	OUT
4995	024420	105777	154164		TSTB	@TKS
4996	024424	100120			BPL	OUT
4997	024426	017737	154160	000640	MOV	@TKB,TIB
4998	024434	042737	177600	000640	BIC	#177600,TIB
4999	024442	022737	000007	000640	CMP	#7,TIB
5000	024450	001106			BNE	OUT
5001	024452	012704	027436		CNTG: MOV	#SCNTG,R4
5002	024454	104000			TTOUTT	
5003	024460	012704	027442		CNTLU: MOV	#SMSWR,R4
5004	024462	104000			TTOUTT	
5005	024466	017703	154114		MOV	AS'R,R3
5006	024472	004737	023740		JSR	PC,OCTPE
5007	024474	012704	027451		MOV	#SMNEW,R4
5008	024502	104000			TTOUTT	
5009	024510	005037	024402		\$READ: CLR	TEMPST
5010	024516	012737	000007	024404	MOV	#7,COUNT
5011	024518	104010			1\$: TTINN	
5012	024520	122737	000025	000640	CMPB	#25,TIB
5013	024526	001001			BNE	2\$
5014	024530	000737			3\$: BR	CNTLU
5015	024532	122737	000015	000640	2\$: CMPB	#15,TIB
5016	024534	001011			BNE	4\$
5017	024536	012737	000200	024406	MOV	#200,RDSW
5018	024538	012704	027461		MOV	#MCRLF,R4
5019	024540	104000			TTOUTT	
5020	024542	022737	000007	024404	CMP	#7,COUNT
5021	024544	001037			BNE	7\$
5022	024546	000437			BR	OUT
5023	024550	122737	000060	000640	4\$: CMPB	#60,TIB
5024	024552	003004			BGT	5\$
5025	024556	122737	000067	000640	CMPB	#67,TIB
5026	024600	002004			BGE	6\$
5027	024610	012704	025763		5\$: MOV	#MSG43,R4
5028	024614	104000			TTOUTT	
5029	024616	000742			BR	3\$
5030	024620	006337	024402		6\$: ASL	TEMPST
5031	024624	006337	024402		ASL	TEMPST
5032	024630	006337	024402		ASL	TEMPST
5033	024634	142737	000060	000640	BICB	#60,TIB
5034	024636	153737	000640	024402	BISB	TIB,TEMPST
5035	024650	005337	024404		DEC	COUNT
5036	024652	001737			BEG	5\$
5037	024656	000717			BR	1\$
5038	024660	013777	024402	153720	7\$: MOV	TEMPST,@SWR
5039	024666	000207			OUT: RTS	PC
5040						:HALT HANDLER*****
5041	024670	000000			STOP: HALT	
5042	024672	104004			CKSWR	
5043	024674	000207			RTS	PC

: 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
: CHECK FOR CONTROL G

5045
5046
5047
5048
5049
5050
5051
5052
5053
5054
5055
5056
5057
5058
5059
5060
5061
5062
5063
5064
5065

024676 016677 000002 153700
024704 011666 000002
024710 162716 000002
024714 013646
024716 062716 120724
024722 013607
024724 023360
024726 023332
024730 024410
024732 024670
024734 023314
104000
104002
104004
104006
104010

;TRAP HANDLER*****

```
TRAP30: MOV 2(6),@PSW ;ADJUST PSW
MOV @SP,2(6) ;PLACE RETURN ADDRESS OVER PSW
SUB #2,@SP ;SUB. 2 FROM RETURN ADDRESS
MOV @6)+-(6)
ADD @TABLE-104000,@SP ;GET SUBROUTINE STARTING ADDRESS
MOV @SP)+,PC ;GO TO SUBROUTINE

TABLE: TTOUT
OCTP
CKSWR
STOP
TTIN

TTOUT= 104000
OCTP= 104002
CKSWR= 104004
STOP= 104006
TTIN= 104010
```

```

5066
5067
5068 :ERROR MESSAGES*****
5069 024736 042052 020105 043 MSG1: .ASCII /DE #/
5070
5071 024743 045 035507 021440 MSG2: .ASCII /XG; #/
5072
5073 024750 041045 020073 043 MSG3: .ASCII /XB; #/
5074
5075 024755 045 047103 021440 MSG4: .ASCII /XCN #/
5076
5077 024762 053452 020105 043 MSG5: .ASCII /WE #/
5078
5079 024767 052 042522 021440 MSG6: .ASCII /RE #/
5080
5081 024774 051052 020123 043 MSG7: .ASCII /RS #/
5082
5083 025001 052 040520 051124 MSG8: .ASCII /PATRN #/
5084 025006 020116 043
5085 025011 040 047123 020072 MSG9: .ASCII /SN: #/
5086 025016 043
5087 025017 052 042523 021440 MSG10: .ASCII /SE #/
5088
5089 025024 051452 040514 042526 MSG11: .ASCII /SLAVE NO. #/
5090 025032 047040 027117 021440
5091
5092 025040 042045 044522 042526 MSG12: .ASCII /XDRIVE NO. #/
5093 025046 047040 027117 021440
5094
5095 025054 025045 047102 021440 MSG13: .ASCII /XBN #/
5096
5097 025062 051052 020116 043 MSG14: .ASCII /RN #/
5098
5099 025067 045 020041 020040 MSG15: .ASCII /X! BAD RECORDX#/
5100 025074 020040 020040 020040
5101 025102 041040 042101 051040
5102 025110 041505 0511.7 022504
5103 025116 021445
5104
5105 025120 043040 043 MSG16: .ASCII /F#/
5106
5107 025123 040 041522 MSG17: .ASCII /R#/
5108
5109 025126 020041 047505 020124 MSG20: .ASCII /! EOT NO: #/
5110 025134 047516 020072 043
5111
5112
5113 025141 045 047111 042524 MSG21: .ASCII /XINTERCHANGE READ = #/
5114 025146 041522 040510 043516
5115 025154 020105 042522 042101
5116 025162 036440 021440
5117
5118 025166 020445 046111 042514 MSG22: .ASCII /X!ILLEGAL BOT: HALTX#/
5119 025174 040507 020114 047502
5120 025202 035124 044040 046101
5121 025210 022524 043
    
```

5122	025213	045	051503	020061	MSG23:	.ASCII	/XCS1 #/	
5123	025220	043						
5124	025221	045	041527	021440	MSG23A:	.ASCII	/XWC #/	
5125	025226	041045	020101	043	MSG23B:	.ASCII	/XBA #/	
5126	025233	045	041506	021440	MSG23C:	.ASCII	/XFC #/	
5127	025240	041445	031123	021440	MSG23D:	.ASCII	/XCS2 #/	
5128	025246	042045	020123	043	MSG23E:	.ASCII	/XDS #/	
5129	025253	045	051105	021440	MSG23F:	.ASCII	/XER #/	
5130	025260	040445	020123	043	MSG23G:	.ASCII	/XAS #/	
5131	025265	045	045503	021440	MSG23H:	.ASCII	/XCK #/	
5132	025272	042045	020102	043	MSG23I:	.ASCII	/XDB #/	
5133	025277	045	051115	021440	MSG23J:	.ASCII	/XMR #/	
5134	025304	042045	020124	043	MSG23K:	.ASCII	/XDT #/	
5135	025311	045	041524	021440	MSG23L:	.ASCII	/XTC #/	
5136	025316	051445	020116	043	MSG23M:	.ASCII	/XSN #/	
5137	025323	045	047041	020117	MSG24:	.ASCII	/X!NO INTERRUPTX#/	
5138	025330	047111	042524	051122				
5139	025336	050125	022524	043				
5140	025343	045	047041	020117	MSG25:	.ASCII	/X!NO MOL: HALTX#/	
5141	025350	047515	035114	044040				
5142	025356	046101	022524	043				
5143	025363	045	051104	050117	MSG26:	.ASCII	/XDROPS: #/	
5144	025370	035123	021440					
5145	025374	050045	041511	051513	MSG27:	.ASCII	/XPICKS: #/	
5146	025402	020072	043					
5147	025405	045	043		MSG28:	.ASCII	/X#/	
5148	025407	045	052043	030115	MSG30:	.ASCII	'XXTMO2-TU16/TE16 AU"O SEQUENCE (CZTUAIO)X#' ;++G	
5149	025414	026462	052524	033061				
5150	025422	032057	030505	020066				
5151	025430	030501	047524	051440				
5152	025438	030503	042523	041516				
5153	025445	020103	041450	052132				
5154	025452	040523	030111	022451				
5155	025461	045	041445	052132	MSG31:	.ASCII	'XXCZTUAIO TMO2-TU16/TE16 RELIABX#'	
5156	025466	040525	030111	052040				
5157	025474	030115	026462	052524				

5178	025502	033061	052057	030505	
5179	025510	020066	042522	042517	
5180	025516	041101	022445	043	
5181					
5182	025523	045	046123	053101	MSG32: .ASCII /%SLAVE NUMBER = #/
5183	025530	020105	052516	041115	
5184	025536	051105	036440	021440	
5185					
5186	025544	042045	047105	044523	MSG33: .ASCII /%DENSITY = #/
5187	025552	054524	036440	021440	
5188					
5189	025560	050045	051101	052111	MSG34: .ASCII /%PARITY = #/
5190	025566	020131	020075	043	
5191					
5192	025573	045	042522	047503	MSG35: .ASCII /%RECORD COUNT = #/
5193	025600	042122	041440	052517	
5194	025606	052116	036440	021440	
5195					
5196	025614	041445	040510	040522	MSG36: .ASCII /%CHARACTER COUNT = #/
5197	025622	052103	051105	041440	
5198	025630	052517	052116	036440	
5199	025636	021440			
5200					
5201	025640	050045	052101	042524	MSG37: .ASCII /%PATTERN NUMBER = #/
5202	025646	047122	047040	046525	
5203	025654	042502	020122	020075	
5204	025662	043			
5205	025668	045	044523	043516	MSG38: .ASCII /%SINGLE PASS = #/
5206	025670	042514	050040	051501	
5207	025676	020123	020075	043	
5208	025703	045	047105	042524	MSG40: .ASCII /%ENTER STALLS%READ = #/
5209	025710	020122	052123	046101	
5210	025716	051514	051045	040505	
5211	025724	020104	020075	043	
5212					
5213	025731	045	051127	052111	MSG41: .ASCII /%WRITE = #/
5214	025736	020105	020075	043	
5215					
5216	025743	045	052524	047122	MSG42: .ASCII /%TURN AROUND = #/
5217	025750	040440	047522	047125	
5218	025756	020104	020075	043	
5219					
5220	025763	045	022477	043	MSG43: .ASCII /%?%#/
5221					
5222	025767	045	047105	042524	MSG44: .ASCII /%ENTER YOZZLE STALL = #/
5223	025774	020122	047531	055132	
5224	026002	042514	051440	040524	
5225	026010	046114	036440	021440	
5226					
5227	026016	042445	051122	040440	MSG45: .ASCII /%ERR AMT #/
5228	026024	052115	021440		
5229					
5230	026030	043045	020103	043	MSG46: .ASCII /%FC #/
5231					
5232	026035	045	040503	021440	MSG47: .ASCII /%CA #/
5233					

5234	026042	020445	047516	041040	MSG48:	.ASCII	/%!NO BOT ON REWIND: HALT%#/ 5235 026050 052117 047440 020116 5236 026056 042522 044527 042116 5237 026064 020072 040510 052114 5238 026072 021445
5240	026074	047040	052117	040440	MSG49:	.ASCII	/ NOT AVAIL #/ 5241 026102 040526 046111 021440 5242 026110 044440 046114 043505 5243 026116 046101 042040 044522 5244 026124 042526 052040 050131 5245 026132 020105 043 5246 026138 045 042045 044522 5247 026142 042526 047040 046522 5248 026150 042502 020122 020075 5249 026156 043
5251	026157	045	047506	046522	MSG53:	.ASCII	/%FORMAT = #/ 5252 026164 052101 036440 021440
5255	026172	053452	020105	046524	MSG54:	.ASCII	/*WE TM#/ 5256 026200 043
5258	026201	052	042523	052040	MSG55:	.ASCII	/*SE TM#/ 5259 026206 021515
5260	026210	052040	021515		MSG56:	.ASCII	/ TM#/ 5261 026214 047040 047117 042455 5262 026222 044530 052123 051440 5263 026230 040514 042526 043 5264 026238 045 051103 020103 5265 026242 043 5266 026248 043 5267 026250 043 051114 020103 5268 026251 052 020104 043 5269 026255 052 020120 043 5270 026261 052 020106 043
5273	026265	045	047452	044522	MSG64:	.ASCII	/%*ORIGINAL ERROR*#/ 5274 026272 044507 040516 020114 5275 026300 051105 047522 025122 5276 026306 043
5278	026307	045	042522	051124	MSG65:	.ASCII	/%RETRY: #/ 5279 026314 035131 021440
5281	026320	020452	042523	051040	MSG66:	.ASCII	/*!SE RTRY #/ 5282 026326 051124 020131 043
5284	026333	052	042441	040522	MSG67:	.ASCII	/*!ERASE#/ 5285 026340 042523 043
5288	026343	045	042522	042522	MSG68:	.ASCII	/%REREV: #/ 5289 026350 035126 021440
5289	026354	052045	050101	020105	MSG69:	.ASCII	/%TAPE MARK = #/

5290	026362	040515	045522	036440	
5291	026370	021440			
5292					
5293	026372	020445	047516	042040	MSG70: .ASCII /%!NO DRY FROM REWIND: HALT%#/ 043
5294	026400	052525	043040	047522	
5295	026406	020115	042522	044527	
5296	026414	042116	020072	040510	
5297	026422	052116	021440		
5298	026426	047040	047117	042455	MSG71: .ASCII / NON-EXIST DRIVE#/ 043
5299	026434	044530	052123	042040	
5300	026442	044530	042522	043	
5301	026447	042522	042522	053506	MSG72: .ASCII /%REFWD: #/ 043
5302	026454	035104	021440		
5303	026460	053445	042522	051122	MSG73: .ASCII /%WTERR: #/ 043
5304	026466	020072	043		
5305	026471	042522	042522	044507	MSG74: .ASCII /%REG STER START = #/ 043
5306	026476	052123	051105	051440	
5307	026504	040524	052122	036440	
5308	026512	021440			
5309	026514	053045	041505	047524	MSG75: .ASCII /%VECTOR = #/ 043
5310	026522	020123	020075	043	
5311	026527	042504	042504	042522	MSG76: .ASCII /%DEREV: #/ 043
5312	026534	035126	021440		
5313	026540	042045	043105	042127	MSG77: .ASCII /%DEFWD: #/ 043
5314	026546	020072	043		
5315	026551	047041	047041	047117	MSG78: .ASCII /%!NON-RETRYABLE WRITE ERROR: ER #/ 043
5316	026556	051055	052105	054522	
5317	026564	041101	042514	053440	
5318	026572	044522	042524	042440	
5319	026600	051122	051117	020072	
5320	026606	051105	021440		
5321	026612	020445	047516	026516	MSG79: .ASCII /%!NON-RETRYABLE READ ERROR: ER #/ 043
5322	026620	042522	051124	040531	
5323	026626	046102	020105	042522	
5324	026634	042101	042440	051122	
5325	026642	051117	020072	051105	
5326	026650	021440			
5327	026652	020445	042441	042116	MSG100: .ASCII /%!!END OF PASS %#/ 043
5328	026660	047440	020106	040520	
5329	026666	051522	022440	043	
5330	026673	042502	025052	025052	MSG101: .ASCII /%*****%#/ 043
5331	026700	025052	025052	022452	
5332	026707	052525	046524	031060	MSG102: .ASCII /*TMO2 #/ 043
5333	026714	021440			
5334	026716	051453	040514	042526	MSG103: .ASCII /*SLAVES #/ 043
5335	026724	020123	043		
5336	026727	042501	052501	047524	MSG104: .ASCII /%AUTO CONT: #/ 043
5337	026734	041440	047117	035124	
5338	026742	021440			
5339	026744	051045	041505	053117	MSG105: .ASCII /%RECOVERED#/ 043
5340	026752	051105	042105	043	
5341	026759	052525	020441	040502	MSG106: .ASCII /*!!BAD TAPE OVERFLOW#/ 043
5342	026764	020104	040524	042520	
5343	026772	047440	042526	043122	
5344	027000	047514	021527		


```
5402 027442 051445 051127 020075 SMSWR: .ASCII /XSWH= #/  
5403 027450 043  
5404 027451 040 047040 053505 SMNEW: .ASCII / NEW= #/  
5405 027456 020075 043  
5406 027461 045 043 MCRLF: .ASCII /X#/  
5407  
5408 027464  
5409 027464 000000 WDATA: 0 EVEN ;WRITE BUFFER  
5410  
5411 033472 033472  
5412 033472 000000 RDATA: 0 =.+4004 ;READ BUFFER  
5413  
5414 000001 .END
```


DATA4	003002	1834#							
DATA5	003004	1835#							
DATA6	003006	1836#							
DATA7	003010	1837#							
DATBL	002770	1829#	3453						
DATER1	001130	1685#	2040	3835*					
DATR	015102	1971#	2237	3633#					
DATRO	015120	3637#	2640						
DATO	014412	1830#	2475#						
DATOA	014444	3482#	2490	3494	3497				
DATOB	014452	4683#	2484						
DATOC	014516	4688#	2493						
DATOD	014524	4689#	2498						
DATOE	014534	4690#	2500						
DATOF	C:4546	4692#	2504						
DAT1	014556	1831#	2511						
DAT1A	014562	1832#	2511	3544	3549	3597			
DAT1B	014566	3511#	2513						
DAT10	014700	3583#	2576						
DAT10A	014712	3586#	2577						
DAT11	014730	1839#	2576						
DAT11A	014736	3578#	2581						
DAT12	014750	1840#	2586						
DAT12A	014760	3588#	2591						
DAT13	014772	1841#	2596						
DAT14	015002	1842#	2601						
DAT14A	015014	3604#	2608						
DAT15	015032	1843#	2613						
DAT15A	015036	3614#	2620						
DAT15B	015046	3616#	2618						
DAT2	014576	1833#	2520#						
DAT3	014602	1833#	2525#						
DAT3A	014610	3527#	2528#						
DAT3B	014614	3528#	2531						
DAT4	014626	1834#	2536#						
DAT5	014636	1835#	2543#						
DAT6	014644	1836#	2548#						
DAT7	014652	1837#	2553#						
DAT7A	014666	3556#	2559						
DB	000532	1531#							
DCHK	015544	2779	2974	3750#					
DCKO	015574	3754	3757#						
DEREV1	001170	1707#	2060	3837*					
DEREX	016622	3908	3929	3931	3939	3947	3950	3952#	
DEREX1	016654	3953	3956	3958	3960#				
DERFL	000704	1590#	2751*	3828	3961*				
DERR	016160	3821	3858#						
DERRO	016170	3870#	3959						
DERROA	016220	3872	3877#						
DERROB	016232	3883	3886#						
DERROC	016276	3890	3893#						
DERROD	016300	3892	3894#						
DERR1	016326	3898	3901#						
DERR2	016330	3900	3902#						
DERR3	016344	3905#							
DERR4	016346	3869	3904	3906#					

CZTUAIO TMO2-TU16/TE16 RELIAB
 CZTUA1.5:11 16-JUN-82 10:48

MACY11 30(1046) 16-JUN-82 11:10 F 10
 CROSS REFERENCE TABLE -- USER SYMBOLS PAGE 123

SEQ 0122

DERR4A	016506	3923	3932#																	
DERR4B	016550	3917	3940#																	
DERR5	016604	3944	3948#																	
DERR6	016616	3920	3942#	3951#																
DFX	016156	3829	3831#	3836#	3838#															
DF0	016050	3777	3816#	3825#																
DF0A	015744	3787	3789#	3826#																
DF0A0	015766	3793	3795#																	
DF0A1	016002	3798	3800#																	
DF0A2	016016	3803	3805#																	
DF0A3	016032	3808	3810#																	
DF0A4	016036	3790	3812#																	
DF0B	015704	3778#																		
DF0B0	015726	3781	3784#																	
DF0C	015664	3770	3774#																	
DF0C0	015674	3760	3762#	3764	3776#															
DF0D	015650	3766	3771#																	
DF0E	015642	3768#	3773#																	
DF0F	015634	3765#	3769#																	
DF1	016062	3813#	3817#	3820#																
DF2	016072	3815#	3819#	3822#																
DF3	016110	3823#	3827#																	
DF4	016152	3834#	3837#																	
DOUT	024166	3894#	3902#	4936#	4953	4955														
DOUTD	024234	4951#																		
DOUT1	024202	4939#	4940	4949																
DOUT2	024234	4943#	4946#																	
DOUT3	024242	4945#	4947#																	
DPC	016776	4000#	4046#																	
DPCG	017004	4001#	4003#																	
DPC0	017012	4004#	4038#																	
DPCJA	017054	4012#	4014#																	
DPC1	017062	4007#	4016#																	
DPC1A	017110	4020#	4022#																	
DPC2	017152	4005#	4010#	4018	4032#															
DPC2A	017114	4015#	4023#																	
DPC2B	017136	4027#	4030#																	
DPC3	017202	4035#	4039#																	
DPPRT	017250	2011	4014#	4022	4048#															
DPPRTX	017404	4068#	4071#																	
DPPRT0	017320	4056#	4061#																	
DPPRT1	017344	4059#	4062#																	
DPPRT2	017360	4065#	4070#																	
DROP	016766	3995#	3998#																	
DRPK	016754	3991#	3995#																	
DRPKF	016666	3820#	3982#																	
DRP1	001010	1630#	3989#	4024	4042	4051														
DRP2	001012	1631#																		
DRP3	001014	1632#																		
DPP4	001016	1633#																		
DFP5	001020	1634#																		
DFP6	001022	1635#																		
DFP7	001024	1636#																		
DFP8	001026	1637#																		
DRVER	021140	2365#	2735*	4181*	4233	4257	4308	4348#												
DS	000522	1527#	1980	1994	2070	2103	2132	2155	2157	2228	2246	2248	2308	2362						

CZTUAIO TM02-TU16 TE16 RELIAB
CZTUA1.P11 16-JUN-82 10:46

MACY11 30(1046) 16-JUN-82 11:10 I 10
CROSS REFERENCE TABLE -- USER SYMBOLS PAGE 126

SEQ 0125

MSG14	025062	2536	4732	5097#
MSG15	000000	3911	5099#	
MSG16	000000	4372	5105#	
MSG16A	000000	2127	5366#	
MSG17	000000	4212	5107#	
MSG17A	000000	4286	5107#	
MSG18	000000	4286	5107#	
MSG19	000000	4286	5109#	
MSG20	000000	4286	5109#	
MSG21	000000	4286	5109#	
MSG22	000000	4286	5109#	
MSG23	000000	4286	5109#	
MSG24	000000	4286	5109#	
MSG25	000000	4286	5109#	
MSG26	000000	4286	5109#	
MSG27	000000	4286	5109#	
MSG28	000000	4286	5109#	
MSG29	000000	4286	5109#	
MSG30	000000	4286	5109#	
MSG31	000000	4286	5109#	
MSG32	000000	4286	5109#	
MSG33	000000	4286	5109#	
MSG34	000000	4286	5109#	
MSG35	000000	4286	5109#	
MSG36	000000	4286	5109#	
MSG37	000000	4286	5109#	
MSG38	000000	4286	5109#	
MSG39	000000	4286	5109#	
MSG40	000000	4286	5109#	
MSG41	000000	4286	5109#	
MSG42	000000	4286	5109#	
MSG43	000000	4286	5109#	
MSG44	000000	4286	5109#	
MSG45	000000	4286	5109#	
MSG46	000000	4286	5109#	
MSG47	000000	4286	5109#	
MSG48	000000	4286	5109#	
MSG49	000000	4286	5109#	
MSG50	000000	4286	5109#	
MSG51	000000	4286	5109#	
MSG52	000000	4286	5109#	
MSG53	000000	4286	5109#	
MSG54	000000	4286	5109#	
MSG55	000000	4286	5109#	
MSG56	000000	4286	5109#	

5140#

5166#

5220#

5234#
5240#
5077#

5254#

TEX	023704	4847	4868#																	
TIB	000640	1572#	4799#	4804	4807	4814*	4815	4836*	4837*	4840	4997*	4998*	4999	5012						
TINER	023:00	4806#	4809#	4820	4823*	4826#														
TINF	000634	1170#	1848#	1893*	1897*	2171*	3162													
TINP	012200	1193#	1197#																	
TINPA	012410	1163#	1172#																	
TINPB	012430	1166#	1191#																	
TINPB0	012614	1203#	1172#																	
TINPB1	012654	1176#	1176#																	
TINPC	012642	1178#	3208#	3211#	3228															
TINPX	012742	1171#	3208#																	
TINPO	012746	1172#	3229#	3241	3255	3306														
TINPOA	013022	1172#	3229#																	
TINPOB	013040	1172#	3229#																	
TINPOC	013100	1172#	3229#																	
TINPOD	013122	1172#	3229#	3256#																
TINPOE	013146	1172#	3229#	3262#																
TINP1	013164	1172#	3229#																	
TINP2	013242	1172#	3229#	3278#																
TINP2A	013320	1172#	3229#	3290#																
TINP2B	013376	1172#	3229#	3302#																
TINP2C	013424	1172#	3229#	3304#	3307#															
TINP3	013444	1172#	3229#																	
TINP4	013774	1172#	3229#	4496																
TKB	000612	1535#	2978#	2992	2996*	4484	4836	4997												
TKS	000610	1535#	2978#	2907*	2976*	3001*	4833*	4854	4995											
TMAX	000564	1547#	2351#	2689	2801	3025	3345	3347	4620*											
TMLG	000676	1587#	2353#	2397*	2413	2686*	2691*	2717	2730*											
TOB	000636	2919	2943#	2970	4133	4155	4162	4173	4188	2777	2789*	2803	2805*	2808*						
TOG	023670	4861*	4867*	4870*	4918*	4928*	4936*	4938*	4941	4846	4848	4850	4854*	4857*						
TPB	000616	1560#	4840*	4867*	4944*	4946*														
TPOS	014152	3277#	3289#	3301	3405#	3408														
TPOS1	014164	3247#	3409#																	
TPS	000614	1559#	4838#	4865	4939															
TRAP30	024676	1465#	2049#																	
TSTAL	000576	1552#	2000#	2003	2489	2613	2638	3021	3052	3392	3394									
TTIN	023514	4833#	5059#																	
TTINN =	104010	4798#	5011#	5064#																
TTINT	021630	1481#	4483#																	
TTINTO	021700	4487#	4495#																	
TTIN1	023520	4833#	4833#																	
TTIN2	023542	4833#	4833#																	
TTOUT	023560	4845#	4853#	4863	4874	5055														
TTOUT =	104000	1882#	1986#	2013	2018	2023	2028	2033	2038	2043	2048	2053	2058	2076						
		2119#	2122#	2128	2139	2162	2183	2196	2238	2243	2248	2253	2258	2276						
		2226#	2228#	2238	2245	2247	2251	2253	2257	2263	2268	2273	2278	2296						
		2282#	2283#	2283	2287	2287	2287	2288	2291	2296	2301	2306	2311	2316						
		2320#	2321#	2321	2325	2325	2325	2326	2327	2332	2337	2342	2347	2352						
		2354#	2355#	2355	2359	2359	2359	2360	2361	2366	2371	2376	2381	2386						
		4254	4260	4264	4270	4270	4280	4280	4284	4290	4290	4294	4294	4298						
		4421	4427	4471	4474	4517	4524	4527	4527	4537	4537	4537	4537	4537						
		4706	4716	4729	4733	4754	4827	5002	5004	5008	5019	5028	5060#							

CZTUA10 TMO2-TU16/TE16 RELIAB
CZTUA1.P11 16-JUN-82 10:46

MACY11 30(1046) 16-JUN-82 11:10 F 11
PAGE 137
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0135

SCHAIN 1358# 1866
.SACT1 1358# 1469
.SEOP 1358# 2184

. ABS. 033474 000

ERRORS DETECTED: 0

CZTUA1,CZTUA1.LST/CRF/SOL/NL:TOC=CZTUA1.SML/ML,CZTUA1.P11

RUN-TIME: 491 SECONDS

RUN-TIME RATIO: 59/15=3.7

CORE USED: 15K (30 PAGES)