

TM02, TE16,
TU16

RELIAB
CZTUAIO

AH-9448I-MC
FICHE 1 OF 1

JUL 1982
COPYRIGHT TO 74-82
MADE IN USA



.REM %

IDENTIFICATION

PRODUCT CODE: AC-94471-MC
PRODUCT TITLE: CZTUAIO TM02-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 CORPORATION

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

TABLE OF CONTENTS

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

1. ABSTRACT

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

THE PROGRAM PROVIDES FOR TESTING OF ALL TAPE DRIVE FUNCTIONS SUCH AS WRITING, READING, REWINDING, TAPE POSITIONING, EOT - BOT SENSING AND ASSUMES A GOOD 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(TM02 ADDRESS), SLAVE NUMBER, DENSITY, PARITY, FORMAT, RECORD COUNT, CHARACTER COUNT, PATTERN NUMBER, TAPE MARK AND STALL FOR READ, WRITE, AND TURNAROUND. ALL RESPONSES SHOULD BE MADE IN OCTAL AND WITHIN THE LIMITS OF THE PARAMETER. A QUESTION MARK (?) WILL BE TYPED IF ANY CHARACTER ENTERED IS NOT BETWEEN 0 THRU 7 (OCTAL). THE CHARACTER MAY BE RETYPED FOLLOWING THE QUESTION MARK. IF THE RESPONSE IS NOT WITHIN ITS LIMITS, A QUESTION MARK (?) IS TYPED AND THE ENTIRE RESPONSE MAY BE ENTERED. SOME RESPONSES REQUIRE MORE THAN ONE (1) CHARACTER, BUT NONE REQUIRES MORE THAN SIX (6). RESPONSES OF MORE THAN ONE CHARACTER NEED NOT HAVE LEADING ZEROS AND SHOULD BE TERMINATED BY A CARRIAGE RETURN IF LESS THAN THE MAXIMUM NUMBER OF CHARACTERS IS INPUT.
- B. 204(8): THIS ADDRESS SHOULD BE USED ANYTIME A RESTART OF THE PROGRAM IS NECESSARY AND THE PARAMETERS ENTERED AT THE INITIAL START OF 200(8) NEED NOT BE CHANGED. ALSO NOTE THAT ANY DATA PATTERN WHICH HAD BEEN GENERATED BY SETTING THE RANDOM DATA SWITCH (CONSOLE SWITCH EIGHT) WILL NOT BE OVERWRITTEN AND THEREFORE IS HELD IN CORE FOR USE UNTIL CONSOLE SWITCH EIGHT(8) IS AGAIN SET AND THAT ALL STATISTICS WILL BE RETAINED.
- C. 210(8): THIS ADDRESS IS THE SAME AS USING 204(8) IN THAT THE PREVIOUSLY SET PARAMETERS ARE USED; HOWEVER, THE DATA PATTERN IS RETURNED TO THE FIXED PATTERN ORIGINALLY CALLED FOR AT THE 200(8) START AND ALL STATISTICS ARE CLEARED TO ZERO.
- D. 240(8): THIS IS A SPECIAL ADDRESS WHICH WILL CAUSE THE PROGRAM TO EXECUTE A PREDETERMINED TEST PLAN ON ALL AVAILABLE DRIVES AND SLAVES. THE ONLY INPUT REQUIRED BY THE OPERATOR IS A RESPONSE TO REQUESTS FOR THE RH ADDRESS, VECTOR ADDRESS, CONTINUOUS OPERATION OF THE SEQUENCE, AND NAZ 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
PREFS. AS MANY AS EIGHT (8) SLAVE
NUMBER REQUESTS MAY BE USED, HOW-
EVER, AT LEAST ONE MUST BE USED.
THE SLAVE NUMBERS AND THEIR RESPECTIVE
DENSITY, PARITY AND FORMAT MAY BE ENTERED
IN ANY ORDER. THE INFORMATION FOR
EACH SLAVE ENTERED IS LOADED INTO A
TABLE FOR REFERENCE IN TESTING.
IF LESS THAN EIGHT (8) SLAVES ARE
REQUIRED, THEN RESPONDING TO THE
SLAVE NUMBER REQUEST WITH A CARRIAGE
RETURN WILL TERMINATE THE SLAVE
ENTRIES AND CONTINUE TO THE NEXT
PARAMETER. IT SHOULD BE REMEMBERED
THAT AT LEAST ONE SLAVE NUMBER REQUEST

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

DENSITY:

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

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

PARITY:

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

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

FORMAT:

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

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

RECORD COUNT:

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

CHARACTER COUNT:

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

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

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

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

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

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

READ: THE TIME DELAY BETWEEN EACH RECORD READ

WRITE: THE TIME DELAY BETWEEN EACH RECORD WRITTEN

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

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

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

SAMPLE START AT 200(8):

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

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

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

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

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

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

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

4.1 AUTOMATIC MODE OPERATION

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

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

5. DATA PATTERNS

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

THE FOLLOWING IS A LIST OF THE DATA PATTERNS AVAILABLE:

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

6. RANDOMIZATION

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

- A. RANDOM DATA: (CONSOLE SWITCH 8)
GENERATES AN ENTIRE BUFFER, CHARACTER BY
CHARACTER, OF RANDOM DATA WHEN SWITCH 8
IS SET TO A ONE. ONCE SET, THE RESETTING OF
SWITCH 8 CAUSES THE LAST GENERATED PATTERN
TO BE RETAINED IN CORE. A RESTART AT LOCATION
200(8) OR 210(8) WILL CAUSE REVERSION OF THE
DATA TO THE FIXED PATTERN REQUESTED INITIALLY.
A RESTART AT LOCATION 204(8) WILL HOLD
THE LAST GENERATED PATTERN IN CORE UNTIL SWITCH
8 IS AGAIN SET.
ALTHOUGH THE DATA IS GENERATED AS RANDOM,
THE PROGRESSION OF RANDOM CHARACTERS IS ALWAYS
THE SAME FROM THE OUTSET OF RANDOMIZATION.
THEREFORE IT IS POSSIBLE TO GENERATE ONE TAPE REEL
OF RANDOM DATA ON ONE UNIT, RESTART THE
PROGRAM TO RE-ESTABLISH THE OUTSET POINT, AND
READ THE RANDOM TAPE REEL ON ANOTHER UNIT
FOR COMPATABILITY TESTING. IN MULTIDRIVE SYSTEMS THE SAME
BLOCK OF DATA, WHETHER RANDOM OR FIXED, IS
WRITTEN OR READ ON EACH AVAILABLE UNIT IN THE
ORDER THAT THEY WERE ENTERED, BEFORE BEING CHANGED.
- B. RANDOM CHARACTER COUNT: (CONSOLE SWITCH 7)
GENERATES A DIFFERENT NUMBER OF CHARACTERS
PER RECORD TO BE WRITTEN ON EACH BLOCK CYCLE.
THE SAME NUMBER OF CHARACTERS PER RECORD IS
WRITTEN OR READ ON EACH AVAILABLE UNIT BEFORE
BEING CHANGED. RESETTING SWITCH 7 HOLDS THE
LAST VALUE GENERATED.
- C. RANDOM RECORD COUNT: (CONSOLE SWITCH 6)
GENERATES A DIFFERENT NUMBER OF RECORDS
FOR EACH BLOCK OF DATA WRITTEN OR READ ON
EACH BLOCK CYCLE. THE SAME NUMBER OF RECORDS
IS WRITTEN OR READ ON EACH AVAILABLE UNIT BEFORE
BEING CHANGED. RESETTING SWITCH 6 HOLDS LAST VALUE
GENERATED.

7. DYNAMIC PARAMETERS:

THE THREE (3) STALL VALUES ARE CONSIDERED TO BE DYNAMIC PARAMETERS AS THEY MAY BE CHANGED WHILE THE PROGRAM IS RUNNING BY TYPING A CONTROL C CHARACTER AT THE TELETYPE. AS SOON AS THE BUS IS RELEASED BY THE MAG TAPE OPERATION IN PROGRESS, THE PROGRAM WILL RESPOND TO THE CONTROL C INPUT BY TYPING A REQUEST FOR NEW STALL PARAMETERS. THE LAST VALUES THAT WERE ENTERED WILL BE PRINTED AS THE STORED VALUES AND MAY BE CHANGED BY ENTERING NEW VALUES OR LEFT UNCHANGED BY TYPING A CARRIAGE RETURN. THE YOZZLE STALL IS ALSO DYNAMIC AND CAN BE CHANGED BY TYPING A CNTRL C WHILE DOING A YOZZLE. A YOZZLE STALL REQUEST WILL BE PRINTED AND SHOULD BE RESPONDED TO WITH THE DESIRED VALUE.

8. CONSOLE SWITCH SETTINGS

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

CONTROL:

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

- 1) 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
(000200)0=USE FIXED CHARACTER COUNT

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

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

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

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

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

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

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

SWITCH EXPLANATION AND EXAMPLES:

SW0-3:

THESE SWITCHES ARE USED TO CONTROL THE SEQUENCE OF MAG TAPE OPERATIONS PERFORMED ON EACH AVAILABLE UNIT. THE BLOCK OF DATA DESCRIBED THROUGH THE RESPONSES TO TELETYPE REQUESTS AT INITIAL START WILL BE EITHER WRITTEN OR READ FROM EACH AVAILABLE UNIT IN THE ORDER THAT THEY WERE ENTERED. THE SEQUENCE OF OPERATIONS IS CALLED A CYCLE, AND WILL BE PERFORMED CONTINUOUSLY UNTIL STOPPED BY THE OPERATOR. WHEN END OF TAPE IS REACHED, THE UNIT WILL BE REWOUND AND FLAGGED AS UNAVAILABLE FOR TEST UNTIL ALL UNITS HAVE REACHED 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 PROCEEDED BY A TWO LINE HEADER WHICH CONTAINS THE DRIVE NUMBER, SLAVE NUMBER, DENSITY, PARITY, AND FORMAT ON THE FIRST LINE, AND THE BLOCK NUMBER, RECORD NUMBER, RECORD SIZE, AND ERROR TYPE ON THE SECOND.

A. OPERATION ERRORS:

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

1. READ/WRITE STATUS ERRORS: THESE ARE DETECTED BY EITHER THE TMO2 ITSELF OR BY THE MASSBUS CONTROLLER. ALL STATUS ERRORS WILL BE REPORTED.
2. TAPE POSITION ERRORS: THESE ARE INDICATED BY AN INCORRECT SPACE OR REWIND OPERATION IN WHICH TAPE POSITION BECOMES UNRELIABLE.

B. DATA ERRORS:

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

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

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

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

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

DRIVE NO. 2 *SLAVE NO. 3 *D 3 *P 1 *F 14 *PATRN 2
*BN 12 *RN 10-25 *RS 20 *RE F
CS1 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 ACCOMPANING
STATUS ERROR, WHICH RESULTED IN A BAD RECORD.

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

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

BAD RECORD

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

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

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

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

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

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

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

EXAMPLE 9: IF , DURING A WRITE RETRY THE BACKSPACE
OR 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
 REFW: 3
 SOFT: 2
 HARD: 1
 DEFWD: 0
 REREV: 4
 SOFT: 1
 HARD: 3
 DEREV: 0
 2 BAD TAPE SPOTS
 0 *RN 1 *RN 2
 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
 REFW: 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 COMPATABILITY. SCOPING OF TAPE UNITS BECOMES SIMPLE; BY SETTING THE DESIRED OPERATION AND ITS PARAMETER, A UNIT MAY BE CONTINUOUSLY EXERCISED IN ANY MANNER DESIRED. BY USING THE VARIOUS ERROR CONTROL SWITCHES AND ENTERING THE NEEDED STALL, ANY FUNCTION CAN BE SCOPED RATHER EASILY. RELIABILITY TESTING CAN BE PERFORMED BY USE OF THE RANDOMIZATION CAPABILITY. PERHAPS A CYCLE OF RANDOM TESTING MIGHT BE SET UP AND ALLOWED TO RUN FOR SOME PERIOD OF TIME, THE STATISTICAL COLLECTION OF DROPS AND PICKS IS THEN SIGNIFICANT. INTERMITTANT PROBLEMS CAN BE FOUND BY SETTING THE DESIRED OPERATION IN MOTION AND DISALLOWING ERROR PRINTOUTS WHILE ALLOWING A HALT ON ERROR. THE ERROR THAT CAUSED THE HALT CAN BE PRINTED BY RESETTING CONSOLE SWITCH TEN AND PRESSING CONTINUE. IF SOME PARTICULAR DATA PATTERN SHOULD BE CAUSING DATA ERROR, USE 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.

THAT IS, YOU'LL LIKE IT.

13. LISTING

X

```

1343 .LIST BIN LOC SEQ
1344 .TITLE CZTUA10 TMO2-TU16/TE16 RELIAB
1345 :ZZ - CZTUA10
1346 :21 APRIL 76
1347 :R. BARNES
1348
1349 :REVISED (++) J.G.ADAMS MAY 1977
1350 :++G
1351 :++G
1352 :++G
1353 :++G
1354 :++G
1355 :++G
1356 :++G
1357 :++G
1358
1359 .MCALL .SACT11,.SEOP,$CHAIN ;++G ACT11 HOOKS
1360 .NLIST MC ;++G DO NOT LIST MACRO CALLS
1361 .LIST ME ;++G LIST MACRO EXPANSIONS
1362 .ENABLE ABS,AMA ;++G ENABLE ABS AND MODE '37'
1363
1364 :CONSOLE SWITCHES*****
1365
1366 :SW15: 1=STOP ON ERROR
1367 : 0=CONTINUE ON ERROR
1368
1369 :SW14: 1=PRINT READ/WRITE STATS
1370 : 0=DO NOT PRINT STATS
1371
1372 :SW13: 1=DO NOT CHECK DATA
1373 : 0=CHECK DATA
1374 :SW12: 1=DO NOT CHECK WRITE ERRORS
1375 : 0=CHECK WRITE ERRORS
1376 :SW11: 1=DO NOT CHECK READ ERRORS
1377 : 0=CHECK READ ERRORS
1378 :SW10: 1=DO NOT PRINT ERRORS
1379 : 0=PRINT ERRORS
1380
1381 :SW9: 1=REWIND TAPE
1382 : 0=DO NOT REWIND
1383
1384 :SW8: 1=USE RANDOM DATA
1385 : 0=USE FIXED DATA PATTERN
1386 :SW7: 1=USE RANDOM CHARACTER COUNT
1387 : 0=USE FIXED CHAR COUNT
1388 :SW6: 1=USE RANDOM RECORD COUNT
1389 : 0=USE FIXED RECORD COUNT
1390
1391 :SW5: 1=YOZZLE ON CURRENT RECORD
1392 : 0=DO NOT YOZZLE
1393
1394 :SW4: 1=DO BOTH READ AND WRITE RETRIES
1395 : 0=INHIBIT RETRIES
1396
1397 :SW3: 1=DO NOT READ FORWARD

```

CZTUA10 TMO2-TU16/TE16 RELIAB MACY11 30(1046) 16-JUN-82 11:10 F 3
CZTUA1.P11 16-JUN-82 10:46 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

[illegible]

```

1451                                     ;REGISTER EQUIVS*****
1452                                     R0=X0
1453                                     R1=X1
1454                                     R2=X2
1455                                     R3=X3
1456                                     R4=X4
1457                                     R5=X5
1458                                     SP=X6
1459                                     PC=X7
1460                                     NOP=240
1461
1462                                     ;TRAP CATCHERS*****
1463                                     =30
1464 000030 024676 TRAP30
1465 000032 000340 =32
1466 000032 000340 340
1467
1468
1469
1470 ;ACT11 HOOK *****
1471 $SVPC= . ;SAVE CURRENT LOCATION CTR
1472 =46
1473 000046 005116 .WORD $ENDAD ;SET LOCATION 46
1474 000052 000052 =52
1475 000052 000000 .WORD 0 ;SET LOCATION 52 = 0
1476 000034 =$SVPC ;RESTORE LOCATION CTR
1477
1478
1479 ;TTY INTERRUPT VECTOR*****
1480 =60
1481 000060 021630 TTINT ;TTY INTERRUPT HANDLER ADDRESS
1482 000062 000000 0
1483
1484
1485 ;SOFTWARE SWITCH REGISTER LOC. 176*****
1486 =176
1487 000176 000000 SWREG: 0 ;SOFTWARE SWITCH REGISTER
1488
1489
1490 ;START ADDRESS*****
1491 =200
1492 000200 000137 003026 JHP START ;ENTER PARAMETERS VIA TTY
1493
1494 =204
1495 000204 000137 003152 JHP STARTC ;USE FIXED PARAMETERS; HOLD DATA
1496
1497 =210
1498 000210 005037 015150 CLR RDFL
1499 000214 000137 003160 JHP STARTA ;USE FIXED PARAMETERS; NEW DATA
1500
1501
1502 ;FAG TAPE INTERRUPT VECTOR*****
1503 =224
1504 000224 021714 HTINT ;FAG TAPE INTERRUPT HANDLER ADDRESS
1505 000226 000340 340
1506

```

CZTUAIO TM02-TU16/TE16 RELIAB MACY11 30(1046) 16-JUN-82 11:10 ¹₃ PAGE 34
CZTUA1.P11 16-JUN-82 10:46

SEQ 0034

1507
1508
1509
1510
1511
1512

000240 000240
000240 000240
000240 000240
000240 000240
000240 000240
000240 000240

:AUTO SEQUENCE START*****

: =240

INC ASEQF
JMP STAUT

:SET AUTO SEQUENCE FLAG
: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      IIRF: 0          :INTERCHANGE READ 1=YES 0=NO
1549 000570 000000      SFG: 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
EMACR: 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
BT_LG: 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
1611 000750 000000 UN2: 0
1612 000752 000000 UN3: 0
1613 000754 000000 UN4: 0
1614 000756 000000 UN5: 0
1615 000760 000000 UN6: 0
1616 000762 000000 UN7: 0
1617 000764 000000 UN8: 0
1618 000766 177777 UNX: -1
  
```

:UNIT ORDER AND DESCRIPTION TABLE *****

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

```

1619
1620
1621
1622 000770 001210 PIK1: BP00
1623 000772 001230 PIK2: BP10
1624 000774 001250 PIK3: BP20
1625 000776 001270 PIK4: BP30
1626 001000 001310 PIK5: BP40
1627 001002 001330 PIK6: BP50
1628 001004 001350 PIK7: BP60
1629 001006 001370 PIK8: BP70
1630 001010 001410 DRP1: BD00
1631 001012 001430 DRP2: BD10
1632 001014 001450 DRP3: BD20
1633 001016 001470 DRP4: BD30
1634 001020 001510 DRP5: BD40
1635 001022 001530 DRP6: BD50
1636 001024 001550 DRP7: BD60
1637 001026 001570 DRP8: BD70
  
```

:UNIT DROPS AND PICKS POINTERS*****

```

1638
1639
1640
1641 001030 001610 BTADDR: BT00
1642 001032 001714 BT01
1643 001034 002020 BT02
1644 001036 002124 BT03
1645 001040 002230 BT04
1646 001042 002334 BT05
1647 001044 002440 BT06
1648 001046 002544 BT07
  
```

:UNIT BAD TAPE POINTERS*****

```

1649
1650
1651
1652 001050 000000 RTY1: 0
1653 001052 000000 RTY2: 0
1654 001054 000000 RTY3: 0
1655 001056 000000 RTY4: 0
1656 001060 000000 RTY5: 0
1657 001062 000000 RTY6: 0
1658 001064 000000 RTY7: 0
1659 001066 000000 RTY8: 0
  
```

:UNIT WRITE RETRY COUNTER*****

```

1660
1661
1662
  
```

: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			:UNIT READ FORWARD ERRORS*****	
1673				
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	BP10: 0 =.+16
1719	001230	000000	BP20: 0 =.+16
1720		001250	BP30: 0 =.+16
1721	001250	000000	BP40: 0 =.+16
1722		001270	BP50: 0 =.+16
1723	001270	000000	BP60: 0 =.+16
1724		001310	BP70: 0 =.+16
1725	001310	000000	BD00: 0 =.+16
1726		001330	BD10: 0 =.+16
1727	001330	000000	BD20: 0 =.+16
1728		001350	BD30: 0 =.+16
1729	001350	000000	BD40: 0 =.+16
1730		001370	BD50: 0 =.+16
1731	001370	000000	BD60: 0 =.+16
1732		001410	BD70: 0 =.+16
1733	001410	000000	
1734		001430	
1735	001430	000000	
1736		001450	
1737	001450	000000	
1738		001470	
1739	001470	000000	
1740		001510	
1741	001510	000000	
1742		001530	
1743	001530	000000	
1744		001550	
1745	001550	000000	
1746		001570	
1747	001570	000000	
1748		001610	
1749			
1750			

```

1751
1752                ;UNIT BAD TAPE COUNTER:16 PER SLAVE*****
1753
1754 001610 000000    BT00: 0
1755                0=+.102
1756 001714 000000    BT01: 0
1757                0=+.102
1758 002020 000000    BT02: 0
1759                0=+.102
1760 002124 000000    BT03: 0
1761                0=+.102
1762 002230 000000    BT04: 0
1763                0=+.102
1764 002334 000000    BT05: 0
1765                0=+.102
1766 002440 000000    BT06: 0
1767                0=+.102
1768 002544 000000    BT07: 0
1769                0=+.102
1770                0=+.102
1771                ;UNIT END OF TAPE COUNTERS 1 PER SLAVE*****
1772
1773 002650 000000    EOTC0: 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
  
```

```

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
0
0

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

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

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

```

```
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864 003026 012706 000500  
1865 003032 005037 000734  
1866 003036 005027  
1867 003040 000000  
1868  
1869 003042 022737 005116 000042  
1870 003050 001404  
1871 003052 005737 000042  
1872 003056 001413  
1873 003060 000406  
1874 003062 012737 000176 000606  
1875 003070 012777 100000 175510  
1876 003076 005237 003040  
1877 003102 000137 003126  
1878 003106  
1879 003106 122737 000006 000041  
1880 003114 001010  
1881 003116 012704 027373  
1882 003122 104000  
1883 003124 000404  
1884 003126 005237 000734  
1885 003132 000137 003262  
1886  
1887  
1888 003136 012737 000001 000634  
1889 003144 005037 015150  
1890 003150 000405  
1891  
1892  
1893 003152 005037 000634  
1894 003156 000434  
1895  
1896  
1897 003160 005037 000634  
1898 003164 012700 000634  
1899 003170 012701 000037  
1900 003174 005020  
  
: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 STAUTB ;++G  
  
:START 204 *****  
STARTC: CLR TINF ;CLEAR TTY INPUT FLAG  
BR STAUTD ;++G  
  
:START 210 *****  
STARTA: CLR TINF ;CLEAR TTY ENTRY FLAG  
STARTB: MOV #10B,R0  
MOV #37,R1  
STARTO: CLR (R0)+ ;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	004737	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,SPSW	
1915	003262	013746	000006	SUSWR: MOV	#6,-(SP)	;SAVE VECTORS
1916	003266	013746	000004	MOV	#4,-(SP)	
1917	003272	012737	003312 000004	MOV	#18,#4	;SET UP FOR TIMEOUT
1918	003300	022777	177777 175300	MOV	#-1,ASWR	;REFERENCE HARDWARE SWITCH REGISTER
1919	003306	001402		CMP	2\$	
1920	003310	000404		BEQ	3\$	
1921	003312	022626		BR	3\$	
1922	003314	012737	000176 000606	1\$: CMP	(SP)+,(SP)+	;ADJUST STACK
1923	003322	012637	000004	2\$: MOV	#SWREG,SWR	;POINT TO SOFTWARE SWITCH REG
1924	003326	012637	000006	3\$: MOV	(SP)+,#4	;RESTORE VECTORS
1925	003332	022737	000176 000606	MOV	(SP)+,#6	
1926	003340	001020		CMP	#SWREG,SWR	;IS SWREG SELECTED
1927	003342	005737	000744	BNE	4\$	
1928	003346	001015		TST	EOPB1	
1929	003350	005037	000744	BNE	4\$	
1930	003354	022737	005116 000042	CLR	EOPB1	
1931	003362	001402		CMP	#SENDAD,#42	;ACT MODE? ++ C.W
1932	003364	004737	024460	BEQ	6\$	;BRANCH - IF YES ++ C.W
1933	003370	005737	000734	JSR	PC,CNTLU	;CHECK FOR CONTROL G
1934	003374	001402		6\$: TST	ASEQF	;AUTO SEQ MODE? ++ C.W
1935	003376	000137	022016	BEQ	4\$	;BRANCH - IF NO ++ C.W
1936	003402	004737	012400	JMP	ASEQ0	;GO DO AUTO SEQ ++ C.W
1937	003406	012777	000040 175104	4\$: JSR	PC,TINP	;GO GET PARAMETERS FROM TTY
1938	003414	005000		MOV	#40,ACS	;INITIALIZE
1939	003416	022760	177777 000746	STAUTO: CLR	R0	;POINT TO FIRST ENTRY
1940	003424	001406		1\$: CMP	#-1,UN1(R0)	;++G BRANCH IF LAST ENTRY
1941	003426	042760	100000 000746	BEQ	2\$	
1942	003434	062700	000002	BIC	#100000,UN1(R0)	;CLEAR EOT FLAG
1943	003440	000766		ADD	#2,R0	;POINT TO NEXT UNIT ENTRY
1944	003442	013703	005156	BR	1\$	;++G CONTINUE CLEARING
1945	003446	000303		2\$: MOV	REOTC,R3	
1946	003450	110337	005156	SWAB	R3	
1947	003454	012777	000100 175126	MOVB	R3,REOTC	;RESTORE EOT CNTR
1948	003462	013700	000674	START1: MOV	#100,ATKS	;SET TTY INTERRUPT ENABLE
1949	003466	022760	177777 000746	MOV	UNP,R0	;R0 = UN1 TABLE POINTER
1950	003474	001404		STAR1A: CMP	#-1,UN1(R0)	;++G BRANCH IF LAST ENTRY
1951	003476	016037	000746 000552	BEQ	STAR1B	;IF LAST UNIT IN STRING: BR
1952	003504	000446		MOV	UN1(R0),UDES	;LOAD NEXT UNIT DESCRIPTION
1953	003506	005237	000654	BR	START4	;++G
1954	003512	005737	000734	STAR1B: INC	BLCNTR	;BUMP BLOCK COUNTER
1955	003516	001411		TST	ASEQF	;SEE IF AUTO SEQ
1956	003520	023737	000654 000740	BEQ	STAR1C	;IF NOT: BR
				CMP	BLCNTR,ABLCNT	;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(R0), 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 003580 001402 BEQ START3 ;IF NOT: BR
1971 003582 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(R0) ;++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 DVM, @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 TIOUTT ;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) ;++G 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 005553 JSR PC, WRITE ;WRITE
2000 003750 013737 000666 MOV TSAL, STAL ;SET TURN AROUND DELAY
2001 003756 004737 015044 JSR PC, STALL ;DELAY
2002 003762 004737 007514 JSR PC, RSEQ ;GO TO READ SEQUENCER
2003 003766 013737 000576 000666 MOV TSAL, 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, R0 ;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 STP: JSR PC, DPPRT ;PRINT DROPS AND PICKS
2012 004034 012704 026307 MOV #MSG65, R4
```

2013	004040	104000		TTOUTT	:PRINT RETRY TOTAL
2014	004042	013704	000674	MOV	UNP,R4
2015	004044	016403	001050	MOV	RTY1(R4),R3
2016	004046	104002		OCTPP	:PRINT RETRIES
2017	004048	012704	026460	MOV	#MSG73,R4
2018	004050	104000		TTOUTT	:PRINT WRITE ERROR TAG
2019	004052	013704	000674	MOV	UNP,R4
2020	004054	016403	001070	MOV	WTER1(R4),R3
2021	004056	104002		OCTPP	:PRINT WRITE ERRORS
2022	004058	012704	026447	MOV	#MSG72,R4
2023	004060	104000		TTOUTT	:PRINT READ FORWARD ERROR TAG
2024	004062	013704	000674	MOV	UNP,R4
2025	004064	016403	001110	MOV	RDER1(R4),R3
2026	004066	104002		OCTPP	:PRINT READ FORWARD ERRORS
2027	004068	012704	027255	MOV	#MSG113,R4
2028	004070	104000		TTOUTT	:PRINT SOFT TAG
2029	004072	013704	000674	MOV	UNP,R4
2030	004074	016403	002670	MOV	RFSOFT(R4),R3
2031	004076	104002		OCTPP	:PRINT FORWARD SOFT ERRORS
2032	004078	012704	027266	MOV	#MSG114,R4
2033	004080	104000		TTOUTT	:PRINT HARD TAG
2034	004082	013704	000674	MOV	UNP,R4
2035	004084	016403	002730	MOV	RFHARD(R4),R3
2036	004086	104002		OCTPP	:PRINT HARD FORWARE ERRORS
2037	004088	012704	026540	MOV	#MSG77,R4
2038	004090	104000		TTOUTT	:PRINT DATA ERROR FORWARD TAG
2039	004092	013704	000674	MOV	UNP,R4
2040	004094	016403	001130	MOV	DATER1(R4),R3
2041	004096	104002		OCTPP	:PRINT DATA ERROR FORWARD NUMBER
2042	004098	012704	026343	MOV	#MSG68,R4
2043	004100	104000		TTOUTT	:PRINT READ ERROR REVERSE TAG
2044	004102	013704	000674	MOV	UNP,R4
2045	004104	016403	001150	MOV	RDER1(R4),R3
2046	004106	104002		OCTPP	:PRINT REVESE ERROR NUMBER
2047	004108	012704	027255	MOV	#MSG113,R4
2048	004110	104000		TTOUTT	:PRINT SOFT TAG
2049	004112	013704	000674	MOV	UNP,R4
2050	004114	016403	002710	MOV	RRSOFT(R4),R3
2051	004116	104002		OCTPP	:PRINT REVERSE SOFT ERROR
2052	004118	012704	027266	MOV	#MSG114,R4
2053	004120	104000		TTOUTT	:PRINT HARD TAG
2054	004122	013704	000674	MOV	UNP,R4
2055	004124	016403	002750	MOV	RRHARD(R4),R3
2056	004126	104002		OCTPP	
2057	004128	012704	026527	MOV	#MSG76,R4
2058	004130	104000		TTOUTT	:PRINT DATA ERROR REVERSE TAG
2059	004132	013704	000674	MOV	UNP,R4
2060	004134	016403	001170	MOV	DEREV1(R4),R3
2061	004136	104002		OCTPP	:PRINT DATA REVERSE ERROR NUMBER
2062	004138	000207		RTS	PC
2063	004140	005237	000726	STPX: INC	BTSTF
2064	004142	004737	007324	JSR	PC,BTPRT
2065	004144	005037	000726	CLR	BTSTF
2066	004146	017700	173270	START5: MOV	@SWR,R0
2067	004148	042700	177762	BIC	#177762,R0
2068	004150	022700	000015	CMP	#15,R0

```

:RETURN
:SET STAT ONLY PRINT
:PRINT BAD TAPE STATS
:CLEAR FLAG
:LOAD SWR
:MASK READ/RITE SWITCHES
: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 004372 012737 153624 000624
2084 004400 012737 032561 000626
2085 004406 013737 000630 000554
2086 004414 013737 000632 000556
2087 004422 000207
2088

      BEQ      START8      ;IF NOT: BR
      TSTB     BDS         ;++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
      STOPP
      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 1S: MOV #11,BC1 ;DRIVE CLEAR
004440 105777 174056 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 ;SET RECORD NUMBER OF EOT
004464 042700 100000 BIC #100000,R0 ;++G CLEAR EOT IND & REC CTR
004466 005037 000660 REOT1A: CLR EOTREC ;PRINT HEADER
004472 004737 022736 JSR PC,PAPRT ;SEE IF POSITION ERROR
004474 022737 000002 CMP #2,BTFLG ;IF NOT: BR
004476 001003 BNE REOT1B ;SET POSITION ERROR MSG
004478 012704 027146 MOV #MSG109,R4 ;SEE IF BAD TAPE OVERFLOW
004480 000406 BR REOT1F ;IF NOT: BR
004482 022737 000001 000724 REOT1B: CMP #1,BTFLG ;SET BAD TAPE OVERFLOW MSG
004484 001004 BNE REOT1C ;PRINT REWIND REASON
004486 012704 026757 MOV #MSG106,R4 ;SET EOT MSG
004488 104000 REOT1F: TTOUTT BR REOT1E ;PRINT MSG
004490 012704 025126 REOT1C: MOV #MSG20,R4 ;BUMP CNTR
004492 104000 REOT1D: TTOUTT MOV UNP,R4 ;PRINT EOT CNTR
004494 013704 000674 INC EOTC(R4) ;PRINT RESTART MSG
004496 005264 002650 MOV EOTC(R4),R3 ;CLEAR BAD TAPE FLAG
004498 016403 002650 OCTPP ;PRINT STATS
004500 012704 027004 REOT1E: MOV #MSG16A,R4 ;PRINT BAD TAPE STATS
004502 104000 TTOUTT ;++G BRANCH IF UNIT IS READY
004504 005037 000724 CLR BTFLG ;WAIT DRY
004506 004737 004030 JSR PC,STP ;PRINT HEADER
004508 004737 007324 JSR PC,BTPRT ;PRINT NO DRIVE READY
004510 105777 173710 REOT2: TSTB @DS ;SEE IF LAST UNIT TO REACH EOT
004512 100414 BMI REOT2A ;IF SO: BR
004514 005337 000666 DEC STAL ;SET EOT FLAG
004516 001372 024767 000652 BNE REOT2 ;SEE IF LAST UNIT TO REACH EOT
004518 012737 022736 MOV #MSG6,FMADDR ;PRINT EOT FLAG
004520 004737 026251 JSR PC,PAPRT ;PRINT EOT FLAG
004522 012704 104000 MOV #MSG60,R4 ;PRINT EOT FLAG
004524 104000 TTOUTT STOPP ;PRINT EOT FLAG
004526 105337 05156 REOT2A: DECB REOTC ;SEE IF LAST UNIT TO REACH EOT
004528 013700 000674 BEQ REOT3 ;IF SO: BR
004530 052760 100000 000746 MOV UNP,R0 ;SET EOT FLAG
004532 052760 100000 BIS #100000,UN1(R0) ;SET EOT FLAG
```

```
2145 004664 005726 TST (SP)+ ;RESET STACK POINTER
2146 004666 000137 JMP START7 ;GO TO NEXT UNIT
2147 004668 000337 REOT3: SWAB REOTC
2148 004670 013700 MOV REOTC,R0
2149 004672 000337 SWAB REOTC
2150 004674 110037 MOV R0,REOTC ;RESTORE EOT UNIT COUNTER
2151 004676 005037 CLR UNP
2152 004678 013700 MOV UNP,R0 ;POINT TO FIRST UNIT
2153 004680 016037 REOT4: MOV UN1(RC),UDES ;LOAD UNIT DESCRIPTION
2154 004682 013777 MOV UDES,R0 ;LOAD COMMAND REGISTER
2155 004684 032777 REOT5: BIT #20000,R0
2156 004686 001374 BNE REOT5 ;AWAIT PIP RESET
2157 004688 032777 BIT #2,R0 ;SEE IF HAVE BOT
2158 004690 001012 BNE REOT6 ;IF SO: BR
2159 004692 012700 MOV #1,R0
2160 004694 004737 JSR PC,PAPRT ;PRINT HEADER
2161 004696 012704 MOV #MSG48,R4
2162 004698 104000 TTOUTT ;PRINT BOT ERROR
2163 004700 104006 STOPP
2164 004702 013700 MOV R0
2165 005002 042760 REOT6: BIC #100000,UN1(R0) ;CLEAR EOT FLAG
2166 005010 062737 ADD #2,UNP
2167 005016 013700 MOV UNP,R0 ;POINT TO NEXT UNIT
2168 005022 022760 CMP #-1,UN1(R0) ;++G BRANCH IF NOT LAST UNIT
2169 005030 001334 BNE REOT4 ;++
2170 005032 005037 REOT7: CLR UNP ;CLEAR UNIT POINTER
2171 005036 005037 CLR TINF ;CLEAR TTY INPUT FLAG
2172 005042 005737 TST ASEQF ;SEE IF AUTO SEQ
2173 005046 001402 BEQ REOTX ;IF NOT: BR
2174 005050 005726 TST (SP)+ ;RESET STACK POINTER
2175 005052 000412 BR TEND ;GO DO END OF PASS ++ C.W
2176
2177 005054 004737 REOTX: JSR PC,RANSET ;GO RESET RANDOM BASE
2178 005060 012737 MOV #-1,PATS ;PRESET PATTERN
2179 005066 005037 CLR RDFL ;CLEAR RANDOM FLAG
2180 005072 005737 TST SPFLG ;SEE IF SINGLE PASS
2181 005076 001420 BEQ REOTXX ;IF NOT: BR
2182 005100 012704 TEND: MOV #MSG100,R4
2183 005104 104000 TTOUTT ;PRINT END OF PASS
2184 005106 013700 MOV @#42,R0 ;GET ACT11 RETURN ADDRESS
2185 005112 001405 BEQ HERE ;BRANCH IF NOT ACT11
2186 005114 000005 RESET
2187 005116 004710 SENDAD: JSR PC,(R0)
2188 005120 000240 NOP
2189 005122 000240 NOP
2190 005124 000240 NOP
2191 005126 005737 HERE: TST ASEQF ;AUTO MODE? ++ C.W
2192 005132 001401 BEQ 1$ ;BRANCH - IF NO ++ C.W
2193 005134 000207 RTS PC ;RETURN TO AUTO SEQ ++ C.W
2194 005136 104006 1$: STOPP
2195 005140 012704 REOTXX: MOV #MSG100,R4 ;GET END OF PASS MESSAGE ++ C.W
2196 005144 104000 TTOUTT ;PRINT MESSAGE ++ C.W
2197 005146 005237 INC EOPB1
2198 005152 000137 JMP STARTE ;RESTART AT BLOCK NUMBER ONE
2199 005156 000000 REOTC: 0 ;EOT UNIT COUNTER
```

```

2200 *****
2201 :REWIND ALL AVAIL TAPES:
2202 :
2203 :THIS ROUTINE: ENTERED VIA CONSOLE SWITCH NINE (9),
2204 :WILL REWIND ALL AVAILABLE TAPES TO BOT NO MATTER
2205 :WHERE THEY ARE CURRENTLY POSITIONED AND RESUME TESTING
2206 :ON THE CURRENTLY SELECTED UNIT.
2207 *****
2208
2209 005160 032777 001000 173420 RWND: BIT #1000,ASWR :SEE IF SHOULD REWIND
2210 005166 001001 BNE RWND4 :IF SO: BR
2211 005170 000207 RTS PC :ELSE EXIT
2212 005172 013737 000674 000714 RWND4: MOV UNP,UPS :SAVE UNIT POINTER
2213 005200 005037 000674 CLR UNP :CLEAR POINTER
2214 005204 005037 000660 CLR EOTREC :CLEAR EDT FLAG
2215 005210 000337 005156 SWAB REOTC
2216 005214 013700 005156 MOV REOTC,R0
2217 005220 000337 005156 SWAB REOTC
2218 005224 110037 005156 MOV R0,REOTC :RESTORE EOT UNIT COUNTER
2219 005230 000674 000674 RWND0: MOV UNP,R0 :POINT TO UNIT ENTRY
2220 005236 022760 177777 000746 CMP #-1,UN1(R0) :++G BRANCH IF LAST ENTRY
2221 005242 001445 BEQ RWND2 :IF SO: BR
2222 005246 005760 000746 TST UN1(R0) :++G BRANCH IF ALREADY REWINDING
2223 005250 100433 BMI RWND1A :++G
2224 005254 016037 000746 000552 MOV UN1(R0),UDES :SET UNIT DESCRIPTION
2225 005258 013777 000552 173254 MOV UDES,AC2 :LOAD COMMAND REGISTER
2226 005262 012777 000011 173214 MOV #11,AC1 :DRIVE CLEAR
2227 005266 012777 000007 173206 MOV #7,AC1 :START REWIND
2228 005270 105777 173214 RWND1: TSTB ADS :++G WAIT FOR DRIVE READY
2229 005274 100414 BMI RWND1A :IF DRY: BR
2230 005278 005337 000666 DEC STAL
2231 005282 001372 BNE RWND1 :AWAIT DRY
2232 005286 012737 024767 000652 MOV #MSG6,EMADDR
2233 005290 004737 022736 JSR PC,PAPRT :PRINT HEADER
2234 005294 012704 026372 MOV #MSG70,R4
2235 005298 104000 TTOUTT :PRINT NO DRIVE READY
2236 005302 104006 STOPP
2237 005306 042760 100000 000746 RWND1A: BIC #100000,UN1(R0) :CLEAR EOT FLAG
2238 005310 062737 000002 000674 ADD #2,UNP :BUMP POINTER
2239 005314 000725 BR RWND0 :++G DO NEXT UNIT
2240 005318 005037 000674 RWND2: CLR UNP :CLEAR POINTER
2241 005322 013700 000674 RWND3: MOV UNP,R0 :POINT TO UNIT ENTRY
2242 005326 022760 177777 000746 CMP #-1,UN1(R0) :++G BRANCH IF LAST ENTRY
2243 005330 001436 BEQ RWNDX :IF SO: BR
2244 005334 016037 000746 000552 MOV UN1(R0),UDES :SET UNIT DESCRIPTION
2245 005338 013777 000552 173130 MOV UDES,AC2 :LOAD COMMAND REGISTER
2246 005342 032777 020000 173102 RWND4: BIT #20000,ADS
2247 005346 001374 BNE RWND4 :AWAIT PIP RESET
2248 005350 032777 000002 173072 BIT #2,ADS :SEE IF HAVE BOT
2249 005354 001407 BEQ RWND6 :IF NOT: BR
2250 005358 062737 000002 000674 RWND5: ADD #2,UNP :BUMP POINTER
2251 005362 012777 000011 173042 MOV #11,AC1 :DRIVE CLEAR
2252 005366 000745 BR RWND3 :++G DO NEXT UNIT
2253 005370 012700 000001 RWND6: MOV #1,R0
2254 005374 004737 022736 JSR PC,PAPRT :PRINT HEADER
2255 005378 012704 026042 MOV #MSG48,R4
  
```

L

SEQ 0050

Address	Hex	ASCII	Comment
2256	005464	104000	TTOUTT
2257	005466	104006	STOPP
2258	005470	000760	BR
2259	005472	013737	000714 000674 RWNDX: BR
2260	005500	013700	000674 000674 RWNDX: MOV
2261	005504	016037	000746 000552 RWNDX: MOV
2262	005512	013777	000552 173022 RWNDX: MOV
2263	005520	000207	000207 RWNDX: RTS
2264			PC

```

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
055532 000137 006320 JMP WEX ;IF NOT: BR
005536 013700 000554 WRTE: MOV RCNT, R0 ;R0=RECORD COUNT
005542 012737 024762 000652 W0: MOV #MSG5, EMADDR ;SET ERROR MSG ADDRESS
005557 013777 000556 172740 MOV #MCNT, @FC ;LOAD CHAR COUNT
005556 012777 027464 172730 MOV #WDATA, @BA ;SET DATA ADDR
005564 112737 000060 000672 MOV #60, MTC1 ;SET WRITE OP COMMAND
005572 012737 005604 000662 MOV #W1, RTRN ;SET RETURN ADDRESS
005600 000137 021216 JMP TAP6 ;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			REQ	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,MTCT	: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,RTRN	: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	006230	042703	102700		BIC	#102700,R3	:MASK UNRECOVERABLE ERROR
2381	006234	001410			BEQ	WTM4	:IF SO: BR
2382	006236	004737	022736		JSR	PC,PAPRT	:PRINT HEADER
2383	006242	012704	026551		MOV	#MSG78,R4	
2384	006246	104000			TTOUTT		:PRINT UNRETRYABLE TAG
2385	006250	004737	011260		JSR	PC,NRTP	:PRINT ER FOR NON-RETRYABLE
2386	006254	000421			BR	WEX	
2387	006256	005037	000700	WTM4:	CLR	RPCNT	:CLEAR REPEAT CNTR
2388	006262	013704	000674		MOV	UNP,R4	
2389	006266	005264	001050		INC	RTY1(R4)	:BUMP RETRY CNTR
2390	006272	005037	000702		CLR	RTCNT	:CLEAR RETRY CNTR
2391	006276	032777	002000	172302	BIT	#2000,@SWR	:SEE IF PRINT ERRORS
2392	006304	001003			BNE	WTM4A	:IF NOT: BR
2393	006306	012704	026265		MOV	#MSG64,R4	
2394	006312	104000			TTOUTT		:PRINT ORIGINAL ERROR TAG
2395	006314	004737	006362	WTM4A:	JSR	PC,WRTY	:GO DO RETRY
2396	006320	005037	000712	WEX:	CLR	RTYFL	:CLEAR RETRY FLAG
2397	006324	005037	000676		CLR	TMFLG	:CLEAR TAPE MARK FLAG
2398	006330	005737	000660		TST	EOTREC	:++G BRANCH IF NOT AT EOT
2399	006334	100011			BPL	WRWX	:++G
2400	006336	017703	172244	WRW:	MOV	@SWR,R3	
2401	006342	042703	177763		BIC	#177763,R3	
2402	006346	022703	000014		CMP	#14,R3	:SEE IF WRITE ONLY
2403	006352	001003			BNE	WRWX	:IF NOT: BR
2404	006354	000137			JMP	REOT	:ELSE REWIND
2405	006360	000207	004424	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
006374 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
006452 104000
006454 012704 026307
006460 104000
006462 013703 000702
006466 104002
006470 000207
006472 005037 000646
006474 013703 000722
006476 042703 102700
006502 001413
006510 004737 022736
006514 012704 026551
006520 104000
006522 004737 011260
006526 012737 000001 000646
006534 000407
006536 032777 002000 172042
006544 001025
006546 012704 027200
006552 104000
006554 012704 026307
006560 104000
006562 013703 000702
006566 104002
006570 012704 027222
006574 104000
006576 013703 000700
006602 104002
006604 005737 000646
006610 001403
006612 005037 000646
006616 000207
006620 005737 000702
006624 001004
006626 013704 000674
006632 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,BSWR    ;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 ERSAV,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,BSWR ;SEE IF PRINT
      BNE WRTY3        ;IF NOT: BR
      MOV #MSG110,R4
      TTOUTT           ;PRINT BAD TAPE SUSPECT
      WRTY2B: MOV #MSG65,R4 ;PRINT RETRY TAG
      TTOUTT           ;PRINT RETRY NUMBER
      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 RTCNT  ;SEE IF FIRST RETRY
      BNE WRTY3A       ;IF NOT: BR
      MOV UNP,R4
      DEC WTER1(R4)    ;DECREMENT WRITE ERROR CNTR
```

```
2462 006636 013704 000674 WRTY3A: MOV UNP,R4 :GET UNIT NUMBER
2463 006642 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 BLNTR,(R4) :SET BLOCK NUMBER
2470 006674 062704 000040 ADD #40,R4 :ADD RCNT OFFSET
2471 006670 013714 000554 MOV RCNT,(R4)
2472 006670 160014 SUB R0,(R4) :SET RECORD NUMBER
2473 006670 005214 INC (R4) :CORRECT RECORD NUMBER
2474 006670 022777 000040 172012 CMP #40,@BTPT :SEE IF TOO MANY BAD SPOTS
2475 006671 001002 BNE WRTY4 :IF NOT: BR
2476 006672 000137 007160 JMP BTOV :ELSE GO TO BAD TAPE OVERFLOW
2477 006672 005237 000702 WRTY4: INC RTCNT :BUMP RETRY COUNTER
2478 006673 022737 000004 000702 CMP #4,RTCNT :SEE IF DONE < RETRIES
2479 006673 001410 BEQ WRTY5 :IF SO: BR
2480 006674 013704 000674 MOV UNP,R4
2481 006674 005264 001050 INC RTY1(R4) :BUMP RETRY COUNTER
2482 006675 005237 000732 INC ERTFL :SET ERASE FLAG
2483 006675 000137 006370 JMP WRTY6 :DO NEXT RETRY
2484 006676 000137 007400 WRTY5: JMP BTUR :ELSE GO TO BAD TAPE UNRECOVERABLE
2485
2486 :WRITE RETRY BACKSPACE-ERASE SUBROUTINE*****
2487
2488 006676 005037 000706 WRTSB: CLR SERFL :CLEAR FLAG
2489 006677 013737 000576 000666 MOV TSTAL,STAL
2490 006677 004737 012304 JSR PC,STALL :DO TURN AROUND DELAY
2491 006677 012737 026320 000652 MOV #MSG66,EMADDR :SET ERROR CODE
2492 006677 177777 171500 MOV #1,@FC :SET TO BACKSPACE 1 RECORD
2493 006677 033472 171470 MOV #RDATA,@BA :SET BA
2494 006677 012234 JSR PC,BKRT :GO BACKSPACE
2495 006677 000706 TST SERFL :SEE IF ERROR
2496 006677 001406 BEQ WRTSB1 :IF NOT: BR
2497 006677 012737 000002 000724 WRTSB0: MOV #2,BTFLG :SET FLAG
2498 006677 026266 CMP (SP)+,(SP)+ :RESET STACK
2499 006677 000137 004424 JMP REOT :GO REWIND AND REMOVE FROM TESTING
2500 006677 005737 000732 WRTSB1: TST ERTFL :SEE IF SHOULD ERASE
2501 006677 001001 BNE WRTSB2 :IF SO: BR
2502 006677 000207 RTS PC :RETURN
2503 006677 005037 000732 WRTSB2: CLR ERTFL :CLEAR ERASE FLAG
2504 006677 000700 CLR RPCNT :CLEAR REPEAT CNTR
2505 006677 000706 CLR SERFL :CLEAR FLAG
2506 006677 012737 026333 000652 MOV #MSG67,EMADDR :SET ERROR CODE
2507 006677 171406 CLR @FC :CLEAR FRAME COUNT
2508 006677 012737 000024 000672 MOV #24,MTCT :SET ERASE OP-CODE
2509 006677 027464 171370 MOV #WDATA,@BA :SET BA
2510 006677 012737 007136 000662 MOV #WRTSB3,RTRN :SET RETURN ADDRESS
2511 006677 000137 021216 JMP TAPG :GO ERASE
2512 006677 012703 027464 WRTSB3: MOV #WDATA,R3 :SET EXPT BA
2513 006677 004737 017504 JSR PC,ER2 :GO CHECK ERRORS
2514 006677 005737 000706 TST SERFL :SEE IF ERROR
2515 006677 001737 BEQ WRTSB1 :IF NOT: BR
2516 006677 000137 007036 JMP WRTSB0
```

```
2518 ;BAD TAPE OVERFLOW SUBROUTINE*****
2519
2520 007160 005037 000712 BTOV: CLR RTYFL ;CLEAR RETRY FLAG
2521 007164 013737 000001 000724 MOV #1,BTFLG ;SET BAD TAPE OVERFLOW FLAG
2522 007172 005726 TST (SP)+ ;++G ADJUST STACK
2523 007174 000137 004424 JMP REOT ;GO REWIND AND REMOVE FROM TESTING
2524 007200 013701 000730 BTOV0: MOV BTPT,R1 ;SET TABLE POINTER
2525 007204 005721 TST (R1)+
2526 007206 005000 CLR R0
2527 007210 010003 BTOV1: MOV R0,R3
2528 007212 000241 CLC
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: TST 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 007324 012704 025405
2561 007330 104000
2562 007332 013704 000674
2563 007336 016437 001030 000730
2564 007344 017703 171360
2565 007350 000641
2566 007352 006003
2567 007354 104002
2568 007356 012704 027234
2569 007362 104000
2570 007364 005777 171340
2571 007370 001001
2572 007372 000207
2573 007374 000137 007200
2574
2575
2576
2577 007400 004737 022736
2578 007404 012704 027047
2579 007410 104000
2580 007412 000207
2581

;BAD TAPE STATISTIC PRINT*****
BTPRT: MOV #MSG28,R4
        TTOUTT ;DO CR/LF
        MOV UNP,R4
        MOV BTAADR(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
  
```

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

2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598 007414 012737 000002 000562 RSEQ: MOV #2,RDCMD
2599 007422 017704 171160 MOV @SWR,R4 :READ SWITCHES
2600 007426 0-2704 177763 BIC #177763,R4 :MASK READ BITS
2601 007432 005704 TST R4 :SEE IF BOTH READS
2602 007434 001004 BNE RSR :IF NOT: BR
2603 007436 032777 000002 171142 BIT #2,@SWR :SEE IF READ REVERSE FIRST
2604 007444 001051 BNE RSFR :IF NOT: BR
2605 007446 032777 000004 171132 P-R: BIT #4,@SWR :SEE IF SHOULD READ REVERSE
2606 007454 001005 BNE RSF :IF NOT: BR
2607 007456 012737 010000 000562 MOV #10000,RDCMD :LOAD READ REVERSE COMMAND
2608 007464 004737 007732 JSR PC,READ :GO READ REVERSE
2609 007470 032777 000010 171110 RSF: BIT #10,@SWR :SEE IF SHOULD READ FORWARD
2610 007476 001026 BNE RSEX :IF NOT: BR
2611 007500 032737 010000 000562 BIT #10000,RDCMD :SEE IF HAVE READ REVERSE
2612 007506 001407 BEQ RSF0 :IF NOT: BR
2613 007510 013737 000576 000666 MOV TSTAL,STAL
2614 007516 004737 012304 JSR PC,STALL :DO READ STALL
2615 007522 000137 007542 JMP RSF1
2616 007526 032777 000001 171052 RSF0: BIT #1,@SWR :SEE IF WRITE
2617 007534 001002 BNE RSF1 :IF NOT: BR
2618 007536 004737 012056 JSR PC,BKSP :GO BACKSPACE
2619 007542 012737 000002 000562 RSF1: MOV #2,RDCMD :LOAD READ FORWARD COMMAND
2620 007546 004737 007732 JSR PC,READ :GO READ
2621 007554 005737 000660 RSEX: TST EOTREC :++G BRANCH IF NOT AT EOT
2622 007560 100002 BPL 1$ :++G
2623 007562 000137 004424 JMP REOT :++G ELSE GO REWIND
2624 007564 000207 1$: RTS PC :++G EXIT
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637 007570 012737 010000 000562 RSFR: MOV #10000,RDCMD
2638 007574 032777 000010 171002 BIT #10,@SWR :SEE IF SHOULD READ FORWARD
2639 007604 001013 BNE RSFR1 :IF NOT: BR
2640 007606 032777 000001 170772 BIT #1,@SWR :SEE IF WRITE
2641 007614 001002 BNE RSFR0 :IF NOT: BR
2642 007616 004737 012056 JSR PC,BKSP :GO BACKSPACE TO START
2643 007622 012737 000002 000562 RSFR0: MOV #2,RDCMD :LOAD READ FORWARD COMMAND
2644 007626 004737 007732 JSR PC,READ :GO READ FORWARD
2645 007634 032777 000004 170744 RSFR1: BIT #4,@SWR :SEE IF SHOULD READ REVERSE
2646 007642 001344 BNE RSEX :IF NOT: BR
2647 007644 032737 010000 000562 BIT #10000,RDCMD
2648 007652 001005 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	:SEE 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		RSFRX:	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 001406
010014 005737 000564
010020 001403
010022 005237 000676
010024 005200
010026 013777 000556 170460
010028 012777 033472 170450
010030 012737 010000 000562
010032 001417
010034 013703 000556
010036 005103
010038 012737 000020 000552
010040 001402
010042 000251
010044 000203
010046 000377 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 #100000,EOTREC ;CLEAR FLAG
        MOV EOTREC,R3    ;GET MODIFIED RECORD COUNT
        SUB R3,R0        ;SET RECORD AT
        INC R0           ;SET TO PROPER NUMBER OF RECORDS
RDA:    MOV #MSG6,EMADDR ;SET ERROR MSG ADDRESS
        CLR TMFLG
        BIT #10000,RDCMD
        BEQ 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,0FC     ;LOAD CHAR CNTR
        MOV #RDATA,0BA   ;LOAD DATA ADDR
        BIT #10000,RDCMD ;SEE IF READ REVERSE
        BEQ RDA          ;IF NOT: BR
        MOV FMCNT,R3
        COM R3
        BIT #20,0DES      ;SEE IF CORE DUMP
        BEQ RD1          ;IF NOT: BR
        CLC
        ROR R3           ;R3 = FC/2
        ADD R3,0BA        ;SET REVERSE BUS ADDRESS
        MOV #76,MT01      ;SET READ REVERSE
        BR RD1B
```

2706	010112	012737	000070	000672	RD1A:	MOV	#70, MTC1	:SET READ FORWARD
2707	010120	012737	010132	000662	RD1B:	MOV	#RD2, RTRN	:SET INTERRUPT RETURN ADDRESS
2708	010126	000137	021216		RD1D:	JMP	TAPG	:GO EXECUTE TAPE COMMAND
2709	010132	032737	010000	000562	RD2:	BIT	#10000, RDCMD	:SEE IF READ REVERSE
2710	010140	001024				BNE	RD3	:IF SO: BR
2711	010146	032777	000020	170352		BIT	#20, ADS	
2712	010150	001404				BEQ	RD2B	:AWAIT SWDN
2713	010156	032777	000020	170342	RD2A:	BIT	#20, ADS	
2714	010160	001574				BNE	RD2A	:AWAIT TUR
2715	010166	032777	002000	170332	RD2B:	BIT	#2000, ADS	:SEE IF EOT
2716	010170	001410				BEQ	RD3	:IF NOT: BR
2717	010176	005737	000676			IST	TMFLG	:SEE IF TM
2718	010182	001005				BNE	RD3	:IF SO: BR
2719	010188	010037	000660			MOV	R0, EOTREC	
2720	010194	032737	100000	000660		BIS	#100000, EOTREC	:SET EOT FLAG
2721	010200	032737	000002	170302	RD3:	BIT	#2, ADS	:SEE IF AT LOAD POINT
2722	010206	001410				BEQ	RD4	:IF NOT: BR
2723	010212	004737	022736			JSR	PC, PAPRT	:PRINT CYCLE NUMBER
2724	010218	012704	025166			MOV	#MSG22, R4	
2725	010224	104000				TTOUTT		:PRINT BOT ERROR
2726	010230	104006				STOPP		
2727	010236	000137	003160			JMP	STARTA	:RESTART
2728	010242	032777	004000	170336	RD4:	BIT	#4000, ASWR	:SEE IF SHOULD CHECK ERRORS
2729	010248	001121				BNE	RD5	:IF NOT: BR
2730	010254	005737	000676			YST	TMFLG	
2731	010260	001472				BEQ	RD4B	:IF NO TM EXPT: BR
2732	010266	032777	000004	170234		BIT	#4, ADS	
2733	010272	001024				BNE	RD4A	:IF TM RECVD: BR
2734	010278	012737	033472	021132		MOV	#RDATA, CADER	:SAVE EXPT BUS ADDRESS
2735	010284	012737	000002	021140		MOV	#2, DRIVER	:SET TM STATUS ERROR FLAG
2736	010290	004737	020234			JSR	PC, ERPT	:GO PRINT TM ERROR
2737	010296	013704	000674			MOV	UNP, R4	
2738	010302	032737	010000	000562		BIT	#10000, RDCMD	:SEE IF READ REVERSE
2739	010308	001403				BEQ	1\$	:IF NOT: BR
2740	010314	005264	001150			INC	RDERR1(R4)	:BUMP READ REVERSE ERROR
2741	010320	000502				BR	RD6	
2742	010326	005264	001110		1\$:	INC	RDERR1(R4)	:BUMP READ FORWARD ERROR
2743	010332	000477				BR	RD6	
2744	010338	012703	033472		RD4A:	MOV	#RDATA, R3	
2745	010344	032737	010000	000562		BIT	#10000, RDCMD	:SEE IF READ REVERSE
2746	010350	001007				BNE	RD4A0	:IF SO: BR
2747	010356	032737	002000	000552		BIT	#2000, UDES	:SEE IF IN PE
2748	010362	001025				BNE	RD4A2	:IF SO: BR
2749	010368	062703	000002			ADD	#2, R3	
2750	010374	004422				BR	RD4A2	
2751	010380	013704	000556		RD4A0:	MOV	FMCNT, R4	
2752	010386	005737				COM	R4	
2753	010392	032737	000020	000552		BIT	#20, UDES	:SEE IF CORE DUMP
2754	010398	001402				BEQ	RD4A1	:IF NOT: BR
2755	010404	000241				CLC		
2756	010410	006004				ROR	R4	:SET TO FC/2
2757	010416	060403			RD4A1:	ADD	R4, R3	:SET EXPT BUS ADDRESS
2758	010422	042703	000001			BIC	#1, R3	:MAKE EXPT ADDRESS EVEN
2759	010428	032737	002000	000552		BIT	#2000, UDES	:SEE IF IN PE
2760	010434	001002				BNE	RD4A2	:IF SO: BR
2761	010440	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	SEFL	
2766	010454	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	005264	001110			INC	RDER1(R4)	:BUMP READ FORWARD ERROR
2771	010476	000402				BR	RD4E	
2772	010480	005264	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,BSWR	:SEE IF SHOULD DO DATA CHECK
2776	010498	001003				BNE	RD6	:IF NOT: BR
2777	010502	005737	000676			TST	TMFLG	
2778	010506	001002				BNE	RD6	
2779	010510	004737	015544			JSR	PC,DCHK	:GO CHECK DATA
2780	010514	005037	000706		RD6:	CLR	SEFL	:CLEAR STATUS ERROR FLAG
2781	010518	004737	014352			JSR	PC,DS3	:CLEAR BUFFER
2782	010522	032777	000040	170032		BIT	#40,BSWR	:SEE IF SHOULD YOZZLE
2783	010526	001402				BEQ	RD7	:IF NOT: BR
2784	010530	004737	011274			JSR	PC,YOZ	:ELSE GO YOZZLE
2785	010534	013737	000572	000666	RD7:	MOV	RSIAL,STAL	:SET DELAY
2786	010538	004737	012304			JSR	PC,STALL	:STALL
2787	010542	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	R0	
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	R0	: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 010706 032777 000020 167672 RDRTY: BIT #20,ASWR ;SEE IF RETRY INHIBITED
2822 010714 001001 BNE RDRT0 ;IF NOT: BR
2823 010716 000207 RTS PC ;ELSE RETURN
2824 010720 013703 000722 RDRT0: MOV ERSAY,R3
2825 010724 042703 102700 BIC #102700,R3 ;MARK NON-RECOVERABLE ERROR BITS
2826 010730 001410 BEQ RDRT1 ;IF NOT: BR
2827 010732 004737 022736 JSR PC,PAPRT ;PRINT HEADER
2828 010736 012704 026612 MOV #MSG79,R4
2829 010742 104000 TTOUTT ;PRINT NON-RECOVERABLE MESSAGE
2830 010744 004737 011260 JSR PC,NRTP ;PRINT ER FOR NON-RETRYABLE ERROR
2831 010750 000207 RDRT1A: RTS PC ;RETURN
2832 010752 032777 002000 167626 RDRT1: BIT #2000,ASWR ;SEE IF PRINT INHIBITED
2833 010760 001003 BNE RDRT1B ;IF SO: BR
2834 010762 012704 026265 MOV #MSG64,R4
2835 010766 104000 TTOUTT ;PRINT ORIGINAL ERROR TAG
2836 010770 005037 000702 RDRT1B: CLR RTCNT ;CLEAR RETRY COUNTER
2837 010774 005037 000706 RDRTG: CLR SERFL ;CLEAR STATUS ERROR FLAG
2838 011000 012737 000002 000712 MOV #2,RTYFL ;SET READ RETRY FLAG
2839 011006 004737 011274 JSR PC,YOZ ;GO TO YOZZLE TO RETRY READ
2840 011012 005737 000706 TST SERFL ;SEE IF RETRY ERROR
2841 011016 001031 BNE RDRT5 ;IF SO: BR
2842 011020 032777 002000 167560 BIT #2000,ASWR
2843 011026 001011 BNE RDRT2
2844 011030 012704 026744 MOV #MSG105,R4
2845 011034 104000 TTOUTT ;PRINT RECOVERED MESSAGE
2846 011036 012704 026307 MOV #MSG65,R4
2847 011042 104000 TTOUTT ;PRINT RETRY TAG
2848 011044 013703 000702 MOV RTCNT,R3
2849 011050 104002 OCTPP ;PRINT RETRY NUMBER
2850 011052 013704 000674 RDRT2: MOV UNP,R4
2851 011056 032737 010000 000562 BIT #10000,RDCMD ;SEE IF READ REVERSE
2852 011064 001003 BNE RDRT3 ;IF SO: BR
2853 011066 005264 002670 INC RFSOFT(R4) ;ELSO BUMP FORWARD SOFT ERROR COUNTER
2854 011072 000402 BR RDRT4
2855 011074 005264 002710 RDRT3: INC RRSOFT(R4) ;BUMP ERRORS SOFT CNTR
2856 011100 000207 RDRT4: RTS PC ;RETURN
2857 011102 005037 000646 RDRT5: CLR TEMP3 ;++G CLEAR RECOVERABLE ERROR INDICATOR
2858 011106 013703 000722 MOV ERSAY,R3 ;GET ER
2859 011112 042703 102700 BIC #102700,R3 ;MASK RECOVERABLE BITS
2860 011116 001413 BEQ RDRT5A ;IF RECOVERABLE: BR
2861 011120 004737 022736 JSR PC,PAPRT ;PRINT HEADER
2862 011124 012704 026612 MOV #MSG79,R4
2863 011130 104000 TTOUTT ;PRINT NON-RECOVERABLE MSG
2864 011132 004737 011260 JSR PC,NRTP ;PRINT ER
2865 011136 012737 000001 000646 MOV #1,TEMP3 ;SET FLAG
```

2866	C11144	000404			BR	RDRT5B	
2867	011146	032777	002000	167432	RDRT5A: BIT	#2000, @SWR	:SEE IF PRINT INHIBITED
2868	011154	001014			BNE	RDRT6	:IF SO: BR
2869	011156	012704	026307		RDRT5B: MOV	#MSG65, R4	
2870	011162	104000			TTOUTT		:PRINT RETRY TAG
2871	011164	013703	000702		MOV	RTCNT, R3	
2872	011170	104003			OCTPP		:PRINT PTRY NUMBER
2873	011172	005737	000646		TST	TEMP3	:SEE IF SID NON-RECOVERABLE
2874	011176	001403			BEQ	RDRT6	:IF NOT: BR
2875	011200	005037	000646		CLR	TEMP3	:CLEAR FLAG
2876	011204	000207			RTS	PC	:EXIT
2877	011206	005237	000702		RDRT6: INC	RTCNT	
2878	011212	023737	000702	000602	CMP	RTCNT, RETRY	:SEE IF DONE 8 RETRIES
2879	011220	001265			BNE	RDRT6	:IF NOT: BR
2880	011222	012704	027277		MOV	#MSG115, R4	
2881	011226	104000			TTOUTT		:PRINT HARD ERROR MESSAGE
2882	011230	013704	000674		MOV	UNP, R4	
2883	011234	032737	010000	000562	BIT	#10000, RDCMD	:SEE IF READ REVERSE
2884	011242	001003			BNE	RDRT7	:IF SO: BR
2885	011244	005264	002730		INC	RFHARD(R4)	:BUMP FORWARD HARD ERROR CNTR
2886	011250	000402			BR	RDRTX	
2887	011252	005264	002750		RDRT7: INC	RRHARD(R4)	:BUMP REVERSE HARD ERROR CNTR
2888	011256	000207			RDRTX: RTS	PC	:RETURN
2889							
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		
011454	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		

```

YOZ:  MOV #1,ATKS      ;SET TTY ENABLE
      MOV YSTAL,STAL
      JSR PC,STALL      ;DO YOZZLE STALL
YOZO:  MOV #1,ATC       ;SET TO 1 RECORD SPACING
      BIT #10000,RDCMD  ;SEE IF READ REVERSE
      BEQ YOZA          ;IF NOT: BR
      MOVB #30,MTCL     ;SET TO SPACE FORWARD
      BR YOZB
YOZA:  MOVB #32,MTCL     ;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,ABA     ;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,ABA        ;SET REVERSE BUS ADDRESS
      MOV #76,MTCL       ;SET READ REVERSE
      BR YOZC2
YOZC1: MOV #70,MTCL      ;SET READ FORWARD
YOZC2: MOV FMCNT,ATC     ;SET CHARACTER COUNT
      MOV #YOZD,RTRN     ;SET RETURN ADDRESS
      JMP TAPG           ;GO READ
YOZD:  BIT #4000,ASWR    ;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			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		JSR	PC, ER2	:GO CHECK ERRORS
2960	011622	000430			BR	YOZF	
2961	011624	012703	033472		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		JSR	PC, ER2	:GO CHECK ERRORS
2966	011646	000416			BR	YOZF	
2967	011650	004737	017406		JSR	PC, ERCHK	:ELSE GO CHECK ERRORS
2968	011654	032777	020000	166724	BIT	#20000, @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		JSR	PC, DS3	:GO CLEAR DATA AREA
2976	011710	105777	166674		TSTB	@TKS	:SEE IF HAVE NEW STALL VALUE
2977	011714	100032			YOZF0:		:IF NOT: BR
2978	011716	122777	000203	166666	BPL	#203, @TKB	:SEE IF CONT C
2979	011724	001026			CMPB	YOZG	:IF NOT: BR
2980	011726	012704	025767		MOV	#MSG44, R4	
2981	011732	104000			TTOUTT		:PRINT YSTALL REQUEST
2982	011734	013703	000600		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	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	012022	005077	166564		CLR	@TKB	:CLEAR CNTL G OUT OF BUFFER
2997	012026	004737	024452		JSR	PC, CNTG	:GO CHANGE SWREG
2998	012032	032777	000040	166546	BIT	#40, @SWR	:SEE IF SHOULD CONTINUE YOZZLE
2999	012040	001402			BEQ	YOZH	:IF NOT: BR
3000	012042	000137	011314		JMP	YOZO	
3001	012046	012777	000100	166534	MOV	#100, @TKS	:SET TTY INTERRUPT ENABLE
3002	012054	000207			RTS	PC	:EXIT

3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055

```

012056 013737 000576 000666 BKSP: MOV TSTAL STAL
012064 004737 012304 JSR PC,STALL ;DO TURN AROUND STALL
012070 012737 025017 000652 MOV #MSG10,EMADDR
012076 012777 033472 16641C P / #RDATA,0BA
012104 005737 000564 TST TMEX ;SEE IF TM
012110 001440 BEQ B0 ;IF NOT: BR
012112 012777 177777 166376 MOV #-1,BFC
012120 012737 000032 000672 MOV #32,NTC1
012126 012737 012140 000662 MOV #BKTM,RTRN
012134 000137 021216 JMP TAPG ;SPACE TO TM
012140 032777 010000 166440 BKTM: BIT #10000,BSWR ;SEE IF SHOULD CHECK ERROR
012146 001021 BNE B0 ;IF NOT: BR
012150 012737 026201 000652 MOV #MSG55,EMADDR
012156 032777 000004 166336 BIT #4,BDS ;SEE IF TM
012164 001006 BNE BKTM ;IF SO: BR
012174 004737 033472 021132 MOV #RDATA,CADER
012200 000404 JSR PC,ERPT ;PRINT ERROR
012202 012703 BR B0
012206 004737 033472 BKTM0: MOV #RDATA,R3
012212 013700 JSR PC,ER2
012216 005100 B0: MOV RCNT,R0
012220 005200 COM R0 ;BUILD SPACE AMOUNT
012230 005200 INC R0
012236 012737 025017 000652 MOV #MSG10,EMADDR ;SET ERROR MESG ADDRESS
012242 010077 166262 RO,BFC
012248 012737 000032 000672 BKRT: MOV #32,NTC1 ;SET SPACE REVERSE
012250 012737 012260 000662 MOV #B1,RTRN ;SET RETURN ADDRESS
012256 010037 000666 MOV RO,STAL ;SET INTERRUPT TIME MULTIPLIER
012260 030137 021216 JMP TAPG ;GO DO SPACE
012266 012703 033472 B1: MOV #RDATA,R3
012270 004737 017504 JSR PC,ER2
012276 013737 000576 000666 B2: MOV TSTAL STAL ;DO STALL
012302 004737 012304 JSR PC,STALL ;STALL
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 200(8) OR MAY BE MODIFIED
:AT ANY TIME BY ENTERING CNTRL C ON THE TTY AND
:INSERTING NEW VALUES IN RESPONSE TO THE REQUEST.
:THE READ STALL AND THE WRITE STALL ARE DELAYS
:EXECUTED BETWEEN EACH RECORD OF THE DATA BLOCK.
:THE TURN AROUND STALL IS EXECUTED EACH TIME
:THE DIRECTION OF TAPE MOVEMENT IS CHANGED AND
:ALSO EACH TIME THE TAPE OPERATION CHANGES FROM
:WRITE TO READ OR READ TO WRITE. THE YOZZLE
:STALL IS EXECUTED ONLY DURING THE YOZZLE ROUTINE.
*****
  
```

012304 005337 000666
 012310 001375
 012312 000207

STALL: DEC STAL
 ONE 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,RANSAV
MOV RANSAV,FMCT ;SET CHAR COUNT
MOV #1,PAIS     ;PRESET DATA PATTERN
RTS PC          ;EXIT
  
```

 :RANDOM RECORD COUNT GENERATOR:

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

RCNTR:

```

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

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 ALSO CHECK THE MASSBUS
:FOR THE PRESENCE OF THE REQUESTED DRIVE. IF IT IS NOT FOUND,
:A NON-EXIST DRIVE MESSAGE WILL BE PRINTED AND ANOTHER DRIVE
:REQUEST MADE. WHEN THE DRIVE IS FOUND, THE RESPONSE IS STORED
:AND CONTROL PASSED TO THE SLAVE SELECT ROUTINE.
:THE SLAVE SELECT ROUTINE ALSO CHECKS FOR THE PRESENCE OF THE
:SLAVE. IF IT IS NOT PRESENT, A MESSAGE IS PRINTED AND ANOTHER
:REQUEST IS ISSUED. WHEN THE SELECTED SLAVE IS FOUND TO BE
:PRESENT, A MESSAGE IS PRINTED IF IT IS A 7 CHANNEL DRIVE
:TO ASSIST IN SELECTING DENSITY, PARITY, AND FORMAT.
:UPON COMPLETION OF THE DEVICE TABLE, REQUESTS
:ARE PRINTED FOR ENTRY OF THE NUMBER OF CHARACTERS
:PER RECORD AND THE NUMBER OF RECORDS PER BLOCK. THE
:NEXT REQUEST IS FOR A PATTERN NUMBER TO BE USED
:FOR WRITING AND CHECKING OF READ DATA.
:FOLLOWING THE PATTERN REQUEST IS THE TAPE MARK OPTION.
:RESPONDING TO THE REQUEST (TM=) WITH A ONE (1)
:WILL CAUSE THE PROGRAM TO WRITE A TM AT THE
:END OF EACH DATA BLOCK AND TO EXPECT THE
:TM TO BE DETECTED IN EITHER READ FORWARD AND REVERSE
:OR DURING SPACE OPERATION. A RESPONSE OF ZERO (TM=0)
:DISALLOWS WRITING OF THE TM AND CAUSES THE READ
:AND SPACE ROUTINES TO EXPECT NO TM TO BE PRESENT.
:THE LAST REQUESTS ARE FOR ENTRY OF THE DESIRED
:WRITE, READ, AND TURN AROUND STALLS.

012400 005737 000634
012404 001001
012406 000207
012410 005037 000674
012414 005037 005156
012420 012700 000010
012424 012701 000746
012430 005021
012434 005300
012438 001375TINP: 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		TOUTT		: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		TOUTT		:REQUEST STARTING REGISTER ADDRESS
3181	012472	013703	000544	MOV	REGS,R3	
3182	012476	104002		OCTPP		:PRINT CURRENT REG START
3183	012500	012703	000544	MOV	#REGS,R3	:SAVE ADDRESS LOCATION
3184	012504	012701	000006	MOV	#6,R1	:SET SIZE OF ENTRY
3185	012510	012702	176400	MOV	#176400,R2	:SET UPPER LIMIT
3186	012514	012703	172300	MOV	#172300,R3	:SET LOWER LIMIT
3187	012520	004737	023354	JSR	PC,TTR	:GO GET RESPONSE
3188	012524	012704	026514	MOV	#MSG75,R4	
3189	012530	104000		TOUTT		:GO REQUEST VECTOR ADDRESS
3190	012534	013703	000546	MOV	VECT,R3	
3191	012538	104002		OCTPP		:PRINT CURRENT VECTOR
3192	012542	012703	000546	MOV	#VECT,R5	:SET SAVE LOCATION
3193	012546	012701	000003	MOV	#3,R1	:SET SIZE OF ENTRY
3194	012550	012702	000224	MOV	#224,R2	:SET UPPER LIMIT
3195	012554	012703	000150	MOV	#150,R3	:SET LOWER LIMIT
3196	012560	004737	023354	JSR	PC,TTR	:GO GET RESPONSE
3197	012564	013700	000546	MOV	VECT,R0	:GET VECTOR ADDRESS
3198	012570	012720	021714	MOV	#MTINT,(R0)+	:LOAD VECTOR WITH HANDLER ADDRESS
3199	012574	012710	000340	MOV	#340,(R0)	:LOAD PRIORITY LEVEL
3200	012578	013700	000544	MOV	REGS,R0	:GET STARTING REGISTER ADDRESS
3201	012604	012701	000016	MOV	#16,R1	:SET NUMBER OF REGISTERS
3202	012610	012702	000510	MOV	#C1,R2	:GET FIRST ADDRESS LOCATION
3203	012614	010022		TINPB0: MOV	R0,(R2)+	:BUILD TABLE OF ADDRESSES
3204	012616	062700	000002	ADD	#2,R0	:BUMP ADDRESS
3205	012622	001301		DEC	R1	:SEE IF DONE
3206	012626	001373		BNE	TINPB0	:IF NOT: BR
3207	012632	005737	000734	TST	ASEQF	:SEE IF AUTO SEQ
3208	012636	001403		BEQ	TINPC	:IF NOT: BR
3209	012642	005726		TST	(SP)+	:RESET STACK POINTER
3210	012646	000137	021732	JMP	ASEQ	:GO TO AUTO SEQUENCE
3211	012652	012777	000040	165650 TINPC: MOV	#40,@CS	:INITIALIZE
3212	012656	012704	026135	MOV	#MSG52,R4	
3213	012662	104000		TOUTT		:REQUEST DRIVE NUMBER
3214	012666	012705	000550	MOV	#DVN,R5	:GET ADDRESS
3215	012672	012701	000001	MOV	#1,R1	:SET SIZE OF RESPONSE
3216	012676	012702	000007	MOV	#7,R2	:SET UPPER LIMIT
3217	012682	012703	000000	MOV	#0,R3	:SET LOWER LIMIT
3218	012686	004737	023354	JSR	PC,TTR	:GO GET DRIVE NUMBER
3219	012692	013777	000550	165610 MOV	DVN,@CS	
3220	012696	005777	165574	TST	@C1	:ACCESS DRIVE
3221	012702	005777	010000	165576 BIT	#10000,@CS	:SEE IF NED
3222	012706	001411		BEQ	TINP0	:IF NOT: BR
3223	012712	012704	026426	MOV	#MSG71,R4	
3224	012716	104000		TOUTT		:PRINT NED
3225	012722	013704	000510	MOV	C1,R4	
3226	012726	005204		INC	R4	
3227	012730	152714	000100	BISB	#100,(R4)	:CLEAR TRE

3228	012744	000736			BR	TINPC		;++6 RETRY DVN
3229	012746	012704	025523		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	012770	012702	000007		MOV	#7,R2		:SET MAXIMUM LIMIT
3235	012772	012703	000000		MOV	#0,R3		:SET MINIMUM LIMIT
3236	013000	004737	023354		JSR	PC,TTR		:GO GET UNIT NUMBER
3237	013004	005737	000642		TST	TEMP1		:SEE IF HAVE NEW PARAMETER
3238	013010	001013			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	000732			BR	TINPO		:++6 ELSE RETRY
3242	013022	013700	000674		MOV	UNP,R0		
3243	013026	012760	177777	000746	MOV	#-1,UN1(R0)		:SET END UNIT TABLE
3244	013034	000137	013424		JMP	TINP2C		:GO GET RECORD COUNT
3245	013040	013700	000674		MOV	UNP,R0		
3246	013044	042760	000007	000746	BIC	#7,UN1(R0)		:CLEAR UNIT NUMBER
3247	013052	004737	014164		JSR	PC,TPOS1		:GO LOAD UNIT NUMBER TO PROPER POSITION
3248	013056	012777	000040	165434	MOV	#40,ACS		
3249	013064	013777	000550	165426	MOV	#N,ACS		
3250	013072	016077	000746	165442	MOV	{R0},ACS		
3251	013100	012777	002000	165430	TINPOC:	BIT	2000,ADT	:LOAD UNIT NUMBER
3252	013106	001005			BNE	TINPOD		:SEE IF SLAVE PRESENT
3253	013110	012704	026214		MOV	#MSG57,R4		:IF SO: BR
3254	013114	104000			TTOUTT			:PRINT NON-EXIST SLAVE
3255	013116	000137	012746		JMP	TINPO		:REDO
3256	013122	022777	142011	165406	TINPOD:	CMP	#142011,ADT	:++6 SEE IF 9TRK TM02,TU16/TE16
3257	013130	001406			BEQ	TINPOE		:IF SO: BR
3258	013136	012704	026110		MOV	#MSG50,R4		:ILLEGAL DRIVE TYPE
3259	013140	104000			TTOUTT			:GO PRINT
3260	013146	017703	165372		MOV	ADT,R3		
3261	013150	104002			OCTPP			:PRINT DRIVE TYPE REGISTER
3262	013156	012704	025011		MOV	#MSG9,R4		
3263	013162	104000			TTOUTT			:PRINT SERIAL NUMBER TAG
3264	013166	017703	165360		MOV	ASN,R3		
3265	013170	004737	024300		JSR	PC,SNPT		:PRINT SERIAL NUMBER
3266	013176	012704	025544		MOV	#MSG33,R4		
3267	013180	104000			TTOUTT			:PRINT DENSITY REQUEST
3268	013186	005037	000644		CLR	TEMP2		:CLEAR BUFFER
3269	013190	012701	000001		MOV	#1,R1		:SET NUMBER OF CHARACTERS TO INPUT
3270	013196	012702	000007		MOV	#7,R2		:SET MAXIMUM LIMIT
3271	013200	012703	000000		MOV	#0,R3		:SET MINIMUM LIMIT
3272	013206	004737	023354		JSR	PC,TTR		:GO GET DENSITY
3273	013210	005737	000642		TST	TEMP1		:SEE IF HAVE NEW PARAMETER
3274	013216	001407			BEQ	TINP2		:IF NOT: BR
3275	013222	022737	003400	000552	BIC	#3400,UDES		:ELSE CLEAR OLD PARAMETER
3276	013226	012703	000010		MOV	#10,R3		:SET POSITION FACTOR
3277	013230	004737	014152		JSR	PC,TPOS		:GO LOAD DENSITY INTO PROPER POSITION
3278	013236	012704	025560		MOV	#MSG34,R4		
3279	013240	104000			TTOUTT			:PRINT PARITY REQUEST
3280	013246	005037	000644		CLR	TEMP2		:CLR BUFFER
3281	013250	012701	000001		MOV	#1,R1		:SET NUMBER OF CHARACTERS TO INPUT
3282	013256	012702	000001		MOV	#1,R2		:SET MAXIMUM LIMIT
3283	013260	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			BEO	TINP2A	:IF NOT: BR
3287	013302	001737	000010	000552	BIC	#10, UDES	:ELSE CLEAR OLD PARAMETER
3288	013303	012703	000003		MOV	#3, R3	:SET POSITION FACTOR
3289	013310	004737	014155		JSR	PC,TPOS	:CO LOAD PARITY TO PROPER POSITION
3290	013320	012704	026157		TINP2A: MOV	#MSG53, R4	
3291	013324	104000			TTOUTT		:REQUEST FORMAT
3292	013326	005037	000644		CLR	TEMP2	
3293	013327	012701	000002		MOV	#2, R1	
3294	013328	012702	000016		MOV	#16, R2	
3295	013329	012703	000014		MOV	#14, R3	
3296	013346	004737	023354		JSR	PC,TTR	:GO GET FORMAT
3297	013352	005737	000642		TST	TEMP1	:SEE IF NEW PARAMETER
3298	013356	001407			BEO	TINP2B	:IF NOT: BR
3299	013360	042737	000170	000552	BIC	#170, UDES	
3300	013368	012703	000004		MOV	#4, R3	
3301	013376	004737	014152		JSR	PC,TPOS	
3302	013377	005237	005156		TINP2B: INC	REOTC	:BUMP EOT UNIT COUNTER
3303	013402	022737	000016	000674	CMP	#16, UNP	:SEE IF DONE UNITS
3304	013410	001405			BEO	TINP2C	:IF SO: BR
3305	013412	062737	000002	000674	ADD	#2, UNP	:POINT TO NEXT UNIT
3306	013420	000137	012746		JMP	TINP0	:ELSE LOOK FOR NEXT UNIT
3307	013422	005037	000674		TINP2C: CLR	UNP	:CLEAR UNIT POINTER
3308	013423	013700	005156		MOV	REOTC, R0	
3309	013424	000337	005156		SWAB	REOTC	
3310	013425	110037	005156		MCVB	R0, REOTC	:SET UNIT EOT COUNTER
3311	013426	012704	025573		TINP3: MOV	#MSG35, R4	
3312	013427	104000			TTOUTT		:PRINT RECORD COUNT REQUEST
3313	013428	013703	000554		MOV	RCNT, R3	
3314	013429	104000			OCTPP		:PRINT RECORD COUNT
3315	013430	012703	000554		MOV	#RCNT, R5	:SET RECORD COUNT ADDRESS
3316	013431	012701	000006		MOV	#6, R1	:SET NUMBER OF CHARACTERS TO INPUT
3317	013432	012702	177777		MOV	#-1, R2	:SET MAXIMUM LIMIT
3318	013433	012703	000001		MOV	#1, R3	:SET MINIMUM LIMIT
3319	013434	004737	023354		JSR	PC,TTR	:GO GET RECORD COUNT
3320	013437	013737	000554	000630	MOV	RCNT, RCSAV	:SAVE RECORD COUNT
3321	013438	012704	025614		MOV	#MSG36, R4	
3322	013439	104000			TTOUTT		:PRINT CHARACTER COUNT REQUEST
3323	013440	005437	000556		NEG	FMcnt	
3324	013441	013703	000556		MOV	FMcnt, R3	
3325	013442	104000			OCTPP		:PRINT CHAR COUNT
3326	013443	012703	000556		MOV	#FMcnt, R5	:SET CHARACTER COUNT ADDRESS
3327	013444	012701	000006		MOV	#6, R1	:SET NUMBER OF CHARACTERS TO INPUT
3328	013445	012702	004000		MOV	#4000, R2	:SET MAXIMUM LIMIT
3329	013446	012703	000004		MOV	#4, R3	:SET MINIMUM LIMIT
3330	013447	004737	023354		JSR	PC,TTR	:GO GET CHARACTER COUNT
3331	013448	005437	000556		NEG	FMcnt	:SET TO TWO'S COMPLIMENT
3332	013449	013737	000556	000632	MOV	FMcnt, FCSAV	:SAVE FRAME COUNT
3333	013450	012704	025640		MOV	#MSG37, R4	:PRINT PATTERN NUMBER REQUEST
3334	013451	104000			TTOUTT		
3335	013452	013703	000560		MOV	PATRN, R3	
3336	013453	104000			OCTPP		:PRINT PATTERN
3337	013454	005037	014554		CLR	DOFL	:CLEAR EXTERNAL DATA FLAG
3338	013455	012705	000560		MOV	#PATRN, R5	:SET PATTERN NUMBER ADDRESS
3339	013456	012701	000002		MOV	#2, R1	:SET NUMBER OF CHARACTERS TO INPUT

3340 013620 012702 000015
3341 013624 012703 000000
3342 013630 004737 023354
3343 013634 012704 023354
3344 013640 104000
3345 013644 013703 000564
3346 013648 104002
3347 013650 012705 000564
3348 013654 012701 000001
3349 013658 012702 000001
3350 013662 012703 000000
3351 013666 004737 023354
3352 013670 012704 025141
3353 013674 104000
3354 013678 013703 000566
3355 013682 104002
3356 013686 012705 000566
3357 013690 012701 000001
3358 013694 012702 000001
3359 013698 012703 000000
3360 013702 004737 023354
3361 013706 012704 025663
3362 013710 104000
3363 013714 013703 000570
3364 013718 104002
3365 013722 012705 000570
3366 013726 012701 000001
3367 013730 012702 000001
3368 013734 012703 000000
3369 013738 004737 023354
3370 013742 005737 014150
3371 013746 001060
3372 013750 012704 025703
3373 013754 104000
3374 013758 013703 000572
3375 013762 104002
3376 013766 012705 000572
3377 013770 012701 000006
3378 013774 012702 177777
3379 013778 012703 000001
3380 013782 004737 023354
3381 013786 012704 025731
3382 013790 104000
3383 013794 013703 000574
3384 013798 104002
3385 013802 012705 000574
3386 013806 012701 000006
3387 013810 012702 177777
3388 013814 012703 000001
3389 013818 004737 023354
3390 013822 012704 025743
3391 013826 104000
3392 013830 013703 000576
3393 013834 104002
3394 013838 012705 000576
3395 013842 012701 000006

TINP4:

MOV #15,R2 ;SET MAXIMUM LIMIT
MOV #0,R3 ;SET MINIMUM LIMIT
JSR PC,TTR ;GO GET PATTERN NUMBER
MOV #MSG69,R4
TTOUTT ;REQUEST TM
MOV TMEX,R3
OCTPP ;PRINT CURRENT TM FLAG SETTING
MOV #TMEX,R5 ;GET TM FLAG ADDRESS
MOV #1,R1 ;SET SIZE OF RESPONSE
MOV #1,R2 ;SET UPPER LIMIT
MOV #0,R3 ;SET LOWER LIMIT
JSR PC,TTR ;TM 1=YES
MOV #MSG21,R4
TTOUTT ;REQUEST INTERCHANGE READ
MOV INTRF,R3
OCTPP ;PRINT CURRENT SETTING
MOV #INTRF,R5 ;GET FLAG ADDRESS
MOV #1,R1 ;SET SIZE OF RESPONSE
MOV #1,R2 ;SET UPPER LIMIT
MOV #0,R3 ;SET LOWER LIMIT
JSR PC,TTR ;GO GET RESPONSE
MOV #MSG38,R4
TTOUTT ;REQUEST SINGLE PASS
MOV SPFLG,R3
OCTPP ;PRINT CURRENT SETTING
MOV #SPFLG,R5 ;SET ADDRESS OF FLAG
MOV #1,R1 ;SET SIZE OF RESPONSE
MOV #1,R2 ;SET UPPER LIMIT
MOV #0,R3 ;SET LOWER LIMIT
JSR PC,TTR ;GO GET RESPONSE
TST SCVFL ;SEE IF SHORT CONVERSATION
BNE TINPX ;IF SO: BR
MOV #MSG40,R4
TTOUTT ;PRINT READ STALL REQUEST
MOV RSTAL,R3
OCTPP ;PRINT READ STALL
MOV #RSTAL,R5 ;SET READ STALL ADDRESS
MOV #6,R1 ;SET NUMBER OF CHARACTERS TO INPUT
MOV #-1,R2 ;SET MAXIMUM LIMIT
MOV #1,R3 ;SET MINIMUM LIMIT
JSR PC,TTR ;GO GET READ STALL
MOV #MSG41,R4
TTOUTT ;PRINT WRITE STALL REQUEST
MOV WSTAL,R3
OCTPP ;PRINT READ STALL
MOV #WSTAL,R5 ;SET WRITE STALL ADDRESS
MOV #6,R1 ;SET NUMBER OF CHARACTERS TO INPUT
MOV #-1,R2 ;SET MAXIMUM LIMIT
MOV #1,R3 ;SET MINIMUM LIMIT
JSR PC,TTR ;GO GET WRITE STALL
MOV #MSG42,R4
TTOUTT ;PRINT TURN AROUND STALL REQUEST
MOV TSTAL,R3
OCTPP ;PRINT TA STALL
MOV #TSTAL,R5 ;SET TURN AROUND STALL ADDRESS
MOV #6,R1 ;SET NUMBER OF CHARACTERS TO INPUT

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							:UNIT DESCRIPTION POSITIONING SUBROUTINE*****
3405	014152	000241		TPOS:	CLC		
3406	014154	006137	000644		ROL	TEMP2	: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		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
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
014324	001410	
014326	012702	002002
014332	012701	027464
014336	042721	140300
014340	005737	
014344	001374	
014346	004737	015152
014352	012702	002000
014356	012701	033472
014362	0057021	
014364	005737	
014366	001374	
014370	013737	000552
014376	042737	177767

```

DSUP:  TST      RDFL      ;SEE IF DID RANDOM DATA
        BNE      DS1      ;++G F SO BRANCH
DS0:    TST      ASEQF     ;SEE IF AUTO SEQ
        BEQ      DSOC     ;IF NOT: BR
        TST      PATRN    ;SEE IF AUTO RANDOM
        BPL      DSOC     ;IF NOT: BR
        JSR      PC,DATR  ;ELSE GO GENERATE RANDOM DATA
        RTS      PC       ;RETURN
DSOC:   CMP      PATRN,PAT ;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      DS0B     ;IF SO: BR
        MOV      R3,PARS   ;SAVE PARITY
        JSR      PC,CRLRC  ;GO GENERATE EXPT CRC/LRC
DS0B:   RTS      PC
DSOA:   MOV      #WDATA,R3 ;R3 = ADDRS OF WRITE BUFFER
        MOV      PATRN,R1  ;R1 = PATTERN SELECTOR
        MOV      R1,PAT5
        ADD      #1,R1     ;BUMP POINTER
        ASL      R1        ;++G MAKE PATTERN SELECTOR EVEN
        JSR      PC,@DATBL(R1) ;GO GENERATE PATTERN
        BIT      #10000,@DT ;SEE IF 7 CH
        BEQ      DS2A     ;IF NOT: BR
        MOV      #2002,R2  ;SET BUFFER SIZE
        MOV      #WDATA,R1 ;SET START OF BUFFER
        BIC      #140300,(R1)+ ;MASK FOR 7 CH
        DEC      R2        ;SEE IF DONE
        BNE      DS2      ;IF NOT: BR
        JSR      PC,CRLRC  ;GO GENERATE EXPT CRC/LRC
DS2A:   MOV      #2000,R2  ;R2=BUFFER SIZE
DS3:    MOV      #RDATA,R1 ;R1=READ DATA START
        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
DS1:    BIT      #10000,@DT ;SEE IF 7 CH
        BEQ      DS2A     ;IF NOT: BR
        MOV      #2002,R2  ;SET BUFFER SIZE
        MOV      #WDATA,R1 ;SET START OF BUFFER
        BIC      #140300,(R1)+ ;MASK FOR 7 CH
        DEC      R2        ;SEE IF DONE
        BNE      DS2      ;IF NOT: BR
        JSR      PC,CRLRC  ;GO GENERATE EXPT CRC/LRC
DS2:    MOV      #2000,R2  ;R2=BUFFER SIZE
DS3:    MOV      #RDATA,R1 ;R1=READ DATA START
        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
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
PARS: -1
0

:EXIT
:PATTERN NUMBER SAVE

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

:++G BRANCH IF EXTERNAL INPUT
:++G
:++G RETURN
:SET EXTERNAL FLAG
:CLR PRB
:CLR PRS
:CLR PRS
:CLR TEMP1
:CLR FOR USE AS CHARACTER FLAG
:START READER
:++G SEE IF DONE
:++B
:CLR SAVE LOCATION
:SAVE CHARACTER
:SEE IF HAVE FOUND START CHARACTER
:IF SO : BR
:SEE IF CHARACTER IS 0
:IF SO : BR
:ELSE SET CHARACTER FOUND FLAG
:SAVE DATA SIZE
:SAVE DATA SIZE
:++G GO GET FIRST DATA CHAR
:LOAD BUFFER
:SEE IF READ ALL
:IF NOT : BR
:R1 = START OF WRITE BUFFER
:R2 = SIZE OF DATA FIELD
:REPEAT LOAD OF DATA FIELD
:SEE IF DONE
:IF NOT : BR
:++G RETURN
:SEE IF AT END OF DATA FIELD
:IF NOT : BR
:++G ELSE RESTART FILL
: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 005001 DAT2: CLR R1 ;R1=DATA
3521 014600 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: MOVB 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 014636 012701 052525 DAT5: MOV #52525,R1 ;R1=DATA
3543 014642 000747 BR DAT1A ;++G LOAD BUFFER
3544
3545          ;ALTERNATING ZERO/ONE*****
3546
3547 014644 012701 125252 DAT6: MOV #125252,R1 ;R1=DATA
3548 014650 000744 BR DAT1A ;++G LOAD BUFFER
3549
3550          ;ONE/ZERO IN ALTERNATING WORDS*****
3551
3552 014652 012701 125252 DAT7: MOV #125252,R1 ;SET WORD 1
3553 014656 012702 052525 MOV #52525,R2 ;SET WORD 2
3554 014662 012704 001002 MOV #1002,R4 ;SET NUMBER OF ENTRIES
3555 014666 010123 DAT7A: MOV R1,(R3)+ ;LOAD WORD 1
3556 014670 010223 MOV R2,(R3)+ ;LOAD WORD 2
3557 014672 005304 DEC R4 ;SEE IF DONE
3558 014674 001374 BNE DAT7A ;IF NOT: BR
3559 014676 000207 RTS PC ;++G RETURN
3560
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  DAT11: CLR R1          ;R1=STARTING DATA
3577 014732 012702 004004  MOV #4004,R2      ;R2=CHARACTER COUNT+4
3578 014736 110123  DAT11A: MOVB R1,(R3)+    ;LOAD BUFFER
3579 014740 105201  INCB R1          ;BUMP DATA
3580 014742 005302  DEC R2          ;SEE IF DONE
3581 014744 001374  BNE DAT11A    ;IF NOT: BR
3582 014746 000207  RTS PC        ;++G RETURN
3583
3584                                     ;ALL BITS 377-0*****
3585
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  DEC 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 015032 012702 000200
 3614 015036 012701 015062
 3615 015042 012704 000010
 3616 015046 012123
 3617 015050 005307
 3618 015052 001375
 3619 015054 005302
 3620 015056 001367
 3621 015060 000207
 3622 015062 000000
 3623 015064 177400
 3624 015066 000377
 3625 015070 000000
 3626 015072 177777
 3627 015074 000377
 3628 015076 177400
 3629 015100 177777

: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 BUFR
 BNE DAT15A ;IF NOT: BR
 RTS PC ;++G RETURN
 APATS: 0
 177400
 377
 0
 -1
 377
 177400
 -1

3630
 3631
 3632 015102 013704 000556
 3633 015106 012703 027464
 3634 015112 012701 177777
 3635 015116 005002
 3636 015120 004737 023322
 3637 015124 013723 000626
 3638 015130 005204
 3639 015132 001372
 3640 015134 004737 014316
 3641 015140 012737 000001 015150
 3642 015146 000207
 3643 015150 000000
 3644

: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

```

3645
3646
3647
3648
3649
3650
3651
3652
3653
3654 015152 013700 000556 CRCLRC: MOV FMCNT,R0 ;SET RECORD SIZE
3655 015156 005400 NEG R0
3656 015160 012701 027464 MOV #WDATA,R1 ;SET START OF BUFFER
3657 015164 005037 015534 CLR XORS
3658 015170 111104 CL0: MOVB (R1),R4 ;GET CHARACTER
3659 015172 004737 015362 JSR PC,CLP ;GO GET PARITY OF CHARACTER
3660 015176 004737 015510 JSR PC,XOR ;XOR CHARACTER
3661 015202 000241 CLC
3662 015204 006004 ROR R4 ;ROTATE 1 RIGHT
3663 015206 103014 BCC CL2 ;IF NO CARRY: BR
3664 015210 052704 000400 BIS #400,R4 ;SET BIT NINE
3665 015214 000241 CLC
3666 015216 010405 CL1: MOV R4,R5 ;SAVE CHARACTER
3667 015220 042705 177703 BIC #177703,R5
3668 015224 005105 COM R5
3669 015226 042705 177703 BIC #177703,R5
3670 015232 042704 000074 BIC #74,R4
3671 015236 050504 BIS R5,R4 ;COMPLIMENT BITS 2,3,4,5
3672 015240 010437 CL2: MOV R4,XORS
3673 015244 005300 DEC R0
3674 015246 001401 BEQ CLLAST ;IF LAST CHARACTER: BR
3675 015250 000747 BR ;++G GET NEXT
3676 015252 013704 015534 CLLAST: MOV XORS,R4
3677 015256 005137 015534 COM XORS
3678 015260 042737 177050 015534 BIC #177050,XORS
3679 015270 042704 177727 BIC #177727,R4 ;COMPLIMENT ALL BUT BITS 3&5
3680 015274 050437 015534 BIS R4,XORS
3681 015300 013737 015534 015536 MOV XORS,EXCRC ;SAVE EXPECTED CRC
3682 015306 013700 000556 MOV FMCNT,R0
3683 015312 005400 NEG R0
3684 015316 012701 027464 MOV #WDATA,R1 ;DO EXPT LRC
3685 015320 005037 015534 CLR XORS
3686 015324 111104 CL3: MOVB (R1),R4
3687 015326 004737 015362 JSR PC,CLP ;GET PARITY
3688 015332 004737 015510 JSR PC,XOR ;XOR CHARACTER
3689 015336 005300 DEC R0
3690 015340 001371 BNE CL3 ;DO ALL FOR LRC
3691 015370 013704 015536 MOV EXCRC,R4
3692 015374 004737 015510 JSR PC,XOR ;XOR CRC TO DATA
3693 015376 013737 015534 015540 MOV XORS,EXLRC ;SAVE EXPT LRC
3694 015380 000207 RTS PC ;RETURN
3695 015382 005704 CLP: TST R4 ;SEE IF 0 CHAR
3696 015384 001010 BNE CLPE ;IF NOT: BR
3697 015386 032737 000010 000552 BIT #10,UDES ;SEE IF EVEN PARITY
3698 015374 001404 BEQ PE ;IF NOT: BR
3699 015376 012704 000420 MOV #420,R4 ;SET 0 CHAR EVEN PARITY
3700 015402 005201 INC R1 ;BUMP POINTER

```

3701	015404	000207			RTS	PC	:RETURN
3702	015406	005037	015542		CLPE:	CLR	:CLEAR BIT COUNTER
3703	015412	012703	000010		MOV	#10,R3	:SET NUMBER OF BITS
3704	015416	032703	000001		CLP0:	BIT	:SEE IF ONE BIT
3705	015422	001402			BEQ	CLP1	:IF NOT: BR
3706	015424	005237	015542		INC	PARCNT	:BUMP COUNTER
3707	015430	000241			CLP1:	CLC	
3708	015432	006004			ROR	R4	:ROTATE TO NEXT BIT
3709	015434	005303			DEC	R3	
3710	015436	001367			BNE	CLP0	:CONTINUE FOR ALL BITS
3711	015440	112104			MOVB	(R1)+,R4	
3712	015442	042704	177400		BIC	#177400,R4	
3713	015446	032737	000001	015542	BIT	#1,PARCNT	:SEE IF ODD NUMBER OF ONE BITS
3714	015454	001005			BNE	CLP2	:IF SO: BR
3715	015456	032737	000010	000552	BIT	#10,UDES	:SEE IF SHOULD BE EVEN PARITY
3716	015464	001406			BEQ	CLP3	:IF NOT: BR
3717	015466	000207			RTS	PC	:ELSE EXIT
3718	015470	032737	000010	000552	CLP2:	BIT	:SEE IF SHOULD BE ODD PARITY
3719	015476	001001			BNE	CLP3	:IF NOT: BR
3720	015500	000207			RTS	PC	:ELSE EXIT
3721	015502	052704	000400		CLP3:	BIS	:SET PARITY BIT
3722	015506	000207			RTS	PC	
3723	015510	010446			XOR:	MOV	R4, -(SP)
3724	015512	043716	015534		BIC	XORS, (SP)	
3725	015516	040437	015534		BIC	R4, XORS	:XOR SUBROUTINE: R4 WITH XORS
3726	015522	052637	015534		BIS	(SP)+, XORS	
3727	015526	013704	015534		MOV	XORS, R4	
3728	015532	000207			RTS	PC	
3729							
3730	015534	000000			XORS:	0	:XOR SAVE
3731	015536	000000			EXCRC:	0	:EXPECTED CRC
3732	015540	000000			EXLRC:	0	:EXPECTED LRC
3733	015542	000000			PARCNT:	0	:PARITY COUNTER
3734							

```
3735  
3736  
3737  
3738  
3739  
3740  
3741  
3742  
3743  
3744  
3745  
3746  
3747  
3748  
3749  
3750 015544 005037 000656 DCHK: CLR BBC :CLEAR BAD RECORD CNTR  
3751 015550 005037 000704 CLR DERFL :CLEAR DATA ERROR FLAG  
3752 015554 013705 000556 MOV FMCNT,R5 :LOAD CHAR COUNT  
3753 015560 032737 000020 000552 BIT #20,UDES :SEE IF CORE DUMP  
3754 015566 001402 BEQ DCHK0 :IF NOT: BR  
3755 015570 000261 SEC  
3756 015572 006005 ROR R5 :R5 = FC/2  
3757 015574 012701 027464 DCHK0: MOV #WDATA,R1 :SET WRITE DATA ADDR  
3758 015600 012702 033472 MOV #RDATA,R2 :SET READ DATA ADDR  
3759 015604 032737 000010 000552 BIT #10,UDES :SEE IF EVEN PARITY  
3760 015612 001430 BEQ DFOC0 :IF NOT: BR  
3761 015614 032737 000020 000552 BIT #20,UDES :SEE IF CORE DUMP PARITY  
3762 015622 001024 BNE DFOC0 :IF SO: BR  
3763 015624 032737 002000 000552 BIT #2000,UDES :SEE IF PE MODE  
3764 015632 001020 BNE DFOC0 :IF SO: BR  
3765 015634 105711 DFGF: TSTB (R1) :SEE IF 0 CHAR  
3766 015636 001404 BEQ DFOD :IF SO: BR  
3767 015640 005201 INC R1 :BUMP POINTER  
3768 015642 005205 DFOE: INC R5 :SEE IF DONE  
3769 015644 001373 BNE DFOF :IF NOT: BR  
3770 015646 000406 BR DFOC :ELSE CONTINUE  
3771 015650 112721 000020 DFOD: MOVB #20,(R1)+ :SET 20 IN PLACE OF 0  
3772 015654 012737 177777 014406 MOV #-1,PATS :SET PATTERN GENERATE FLAG  
3773 015662 000767 BR DFOE  
3774 015664 013705 000556 DFOC: MOV FMCNT,R5 :RESET CHAR CNT  
3775 015670 012701 027464 MOV #WDATA,R1 :RESET DATA ADDRESS  
3776 015674 032737 010000 000562 DFOC0: BIT #10000,RDCMD :SEE IF READ REVERSE  
3777 015676 001462 BEQ DFO :IF NOT: BR  
3778 015678 013704 000556 DFOB: MOV FMCNT,R4 :GET FRAME COUNT  
3779 015680 005404 NEG R4 :SET TO WHOLE NUMBER  
3780 015682 032737 000020 000552 BIT #20,UDES :SEE IF CORE DUMP  
3781 015684 001402 BEQ DFOB0 :IF NOT: BR  
3782 015686 000241 CLC  
3783 015688 006004 ROR R4 :SET TO FC/2  
3784 015690 040401 DFOB0: ADD R4,R1 :POINT TO START OF WRITE DATA  
3785 015692 040402 ADD R4,R2 :POINT TO START OF READ DATA  
3786 015694 032737 000001 000556 BIT #1,FMCNT :SEE IF ODD FRAME COUNT  
3787 015696 001401 BEQ DFOA :IF NOT: BR  
3788 015698 105722 TSTB (R2)+ :BUMP POINTER  
3789 015700 032737 000020 000552 DFOA: BIT #20,UDES :SEE IF CORE DUMP  
3790 015752 001431 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				:POSITION BITS FOR REVERSE CORE DUMP
3802	016006	132712	000001	CLC		
3803	016012	001401		BITB	#1, (R2)	
3804	016014	000261		BEQ	DF0A2	
3805	016016	106012		SEC		
3806	016020	000241		DF0A2:	RORB	(R2)
3807	016022	132712	000001	CLC		
3808	016026	001401		BITB	#1, (R2)	
3809	016030	000261		BEQ	DF0A3	
3810	016032	106012		SEC		
3811	016034	005202		DF0A3:	RORB	(R2)
3812	016036	124142		INC	R2	:RESET POINTER
3813	016040	001010		DF0A4:	CMPB	-(R1), -(R2)
3814	016042	105037	000656	BNE	DF1	:TEST DATA CHARACTER
3815	016046	000411		CLRB	BBC	:IF NOT GOOD: BR
3816	016050	122122		BR	DF2	:CLEAR BAD RECORD COUNTER
3817	016052	001003		DF0:	CMPB	(R1)+, (R2)+
3818	016054	105037	000656	BNE	DF1	:CHECK DATA
3819	016060	000404		CLRB	BBC	:IF BAD: BR
3820	016062	004737	016666	BR	DF2	:CLEAR BAD RECORD CNTR
3821	016066	004737	016160	DF1:	JSR	PC, DRPKF
3822	016072	005205		JSR	PC, DERR	:GO GET DROPS AND PICKS
3823	016074	001405		DF2:	INC	R5
3824	016076	032737	010000	BEQ	DF3	:GO DO PRINT
3825	016104	001761	000562	BIT	#10000, RDCMD	:BUMP CHAR CNTR
3826	016106	000716		BEQ	DF0	:IF DONE ALL: BR
3827	016110	005037	000664	SR	DF0A	:SEE IF READ REVERSE
3828	016114	005737	000704	DF3:	CLR	HDRFL
3829	016120	001416		TST	DERFL	:ELSE CONTINUE READ REV
3830	016122	005737	000706	BEQ	DFX	:CLEAR HEADER FLAG
3831	016126	001013		TST	SERFL	:SEE IF HAD DATA ERROR
3832	016130	013704	000674	BNE	DFX	:IF NOT: BR
3833	016134	032737	010000	MOV	UNP, R4	IF NOT DATA ERROR ONLY: BR
3834	016142	001003		BIT	#10000, RDCMD	:SEE IF READ REVERSE
3835	016144	005264	001130	BNE	DF4	:IF SO: BR
3836	016150	000402		INC	DATER1(R4)	:BUMP DATA ERROR FORWARD COUNTER
3837	016152	005264	001170	BR	DFX	
3838	016156	000207		DF4:	INC	DEREV1(R4)
3839				DFX:	RTS	PC

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 001067
 016170 005237 000670
 016174 005737 000664
 016200 001007
 016202 004737 022736
 016206 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,BSWR :SEE IF SHOULD PRINT ERRORS
 BNE DERR4 :++G BRANCH IF NOT
 DERR0: INC PFLG :SET PRINT FLAG
 TST HDRFL :SEE IF HAVE PRINTED HEADER
 BNE DERR0A :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
 DERR0A: MOV #MSG4,R4
 TTOUTT :PRINT CHAR NO. HEADER
 MOV R2,R3 :POINT TO CHAR
 SUB #RDATA,R3
 DEC R3
 BIT #10000,RDCMD :SEE IF READ REVERSE
 BEQ DERR0B :IF NOT: BR
 MOV R5,R3 :GET CHAR NUMBER
 COM R3
 DERR0B: OCTPP :PRINT CHAR NUMBER
 MOV #MSG2,R4
 TTOUTT :PRINT EXPECTED DATA
 BIT #10000,RDCMD :SEE IF READ REVERSE
 BEQ DERR0C :IF NOT: BR
 MOVB (R1),R3 :GET CHAR
 BR DERR0D
 DERR0C: MOVB -(R1),R3 :LOAD EXPECTED DATA
 DERR0D: 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	016322	000401			BR	DERR2	
3901	016326	114203			MOVB	-(R2),R3	
3902	016330	004737	024166		JSR	PC,DOUT	:PRINT BAD CHAR
3903	016334	032737	010000	000562	BIT	#10000,RDCMD	:SEE IF READ REVERSE
3904	016342	001001			BNE	DERR4	:++G BRANCH IF NOT
3905	016344	122122			DERR3:	(R1)+,(R2)+	:PESET POINTERS
3906	016346	103237	000656		DERR4:	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		CLRB	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	016434	101470			BLOS	DERR6	:IF SO: BR
3921	016436	012705	177767		MOV	#177767,R5	:SET CHAR CNTR TO 8
3922	016442	032737	010000	000562	BIT	#10000,RDCMD	:SEE IF READ REVERSE
3923	016450	001416			BEQ	DERR4A	:IF NOT: BR
3924	016452	012701	027464		MOV	#WDATA,R1	:GET START OF BUFFER
3925	016456	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	016472	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:	FM CNT,TEMP1	:LOAD CHAR COUNT
3933	016514	005437	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	016536	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	:REPOSITION BBC
3941	016554	062705	000024		ADD	#24,R5	:SKIP 20 CHARS
3942	016560	103416			BCS	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	: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	:SET TO EOR

1.7

```

:++G BRANCH IF CONTINUE ON ERROR
:++G

:SEE IF PRINTED
:IF SO: BR
:SEE IF SHOULD PRINT
:IF NOT: BR
:ELSE PRINT
:CLEAR FLAG
:BUMP DATA ERROR FLAG
:RETURN

```

3952	016622	005777	161760
3953	016626	100012	
3954	016630	104006	
3955	016632	005737	000670
3956	016636	001006	
3957	016640	032777	002000
3958	016646	001002	
3959	016650	000137	01617
3960	016654	005037	000670
3961	016660	005237	000704
3962	016664	000207	
3963			

161740

```

DEREX:      TST      @SWR
            BPL      DEREX1
            STOPP
            TST      PFLG
            BNE      DEREX1
            BIT      #2000, @SWR
            BNE      DEREX1
            JMP      DERRO
DEREX1:     CLR      PFLG
            INC      DERFL
            RTS      PC

```

```

3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982 016656 005037 000642 DRPKF: CLR TEMP1
3983 016672 005037 000644 CLR TEMP2
3984 016676 005037 000646 CLR TEMP3
3985 016702 111137 000642 MOV (R1),TEMP1 ;LOAD GOOD CHAR
3986 016706 111237 000644 MOV (R2),TEMP2 ;LOAD BAD CHAR
3987 016712 013704 000674 MOV UNP,R4
3988 016716 016437 000770 MOV PIK1(R4),BPKP
3989 016724 016437 001010 MOV DRP1(R4),BDPP
3990 016732 032737 010000 000720 000716 000562 BIT #10000,R0CMD ;SEE IF READ REVERSE
3991 016740 001005 BNE DRPK ;IF SO: BR
3992 016742 124142 CMPB -(R1),-(R2) ;POINT TO CHAR
3993 016744 112137 000642 MOV (R1)+,TEMP1 ;LOAD GOOD CHAR
3994 016750 112237 000644 MOV (R2)+,TEMP2 ;LOAD BAD CHAR
3995 016754 004737 016766 DRPK: JSR PC,DROP ;GET DROPS
3996 016760 004737 017204 JSR PC,PICK ;GET PICKS
3997 016764 000207 RTS PC ;EXIT
3998 016766 113703 000642 DROP: MOV TEMP1,R3 ;R3 = GOOD CHAR
3999 016772 113704 000644 MOV TEMP2,R4 ;R4 = BAD CHAR
4000 016776 140403 DPC: BICB R4,R3 ;GET DROPS/PICKS
4001 017000 001001 BNE DPCG ;IF SOME: BR
4002 017002 000207 RTS PC ;RETURN
4003 017004 012737 000010 000710 DPCG: MOV #10,BCNT ;SET NUMBER TO CHECK
4004 017012 132703 000001 DPC0: BITB #1,R3 ;SEE IF DROPPED OR PICKED THIS BIT
4005 017016 001453 BEQ DPC2 ;IF NOT: BR
4006 017020 105737 000646 TSTB TEMP3 ;SEE IF ON PICKS
4007 017024 001016 BNE DPC1 ;IF SO: BR
4008 017026 005277 161664 INC @BDPP ;BUMP DROP CNTR
4009 017032 005777 161660 TST @BDPP
4010 017036 100045 BPL DPC2 ;IF NO OVERFLOW: BR
4011 017040 032777 002000 161540 BIT #2000,@SWR ;SEE IF HAVE PRINTED DATA
4012 017046 001402 BEQ DPC0A ;IF SO: BR
4013 017050 004737 022736 JSR PC,PAPRT ;PRINT CYCLE NUMBER
4014 017054 004737 017250 JSR PC,DPPRT ;PRINT DROPS AND PICKS
4015 017060 000415 BR DPC2A
4016 017062 005277 161632 DPC1: INC @BPKP ;BUMP PICK CNTR
4017 017066 005777 161626 TST @BPKP ;SEE IF OVERFLOW
4018 017072 100027 BPL DPC2 ;IF NOT: BR
4019 017074 032777 002000 161504 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		JSR	PC,DPPRT	:PRINT DROPS AND PICKS
4023	017114	013704	000674		MOV	UNP,R4	
4024	017120	016403	001010		MOV	DRP1(R4),R3	:SET DROP POINTER
4025	017124	016403	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	005023			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
 017432 001402
 017434 000241
 017436 006003
 017440 032737 000010 000672
 017446 001414
 017450 032737 010000 000562
 017456 001405
 017460 012703 033472
 017464 162703 000002
 017470 000405
 017472 062703 033472
 017476 000402
 017500 062703 027464
 017504 010337 021132
 017510 012704 000007
 017514 012701 021134
 017520 005021
 017522 005304
 017524 001375
 017526 020377 160762
 017532 001402
 017534 005237 021134
 017540 032737 000010 000672
 017546 001006
 017550 005777 160742

```

ERCHK: MOV    FMCNT,R3      ;GET FRAME COUNT
        BIT    #1,R3        ;SEE IF ODD
        BEQ    ERO          ;IF NOT: BR
        DEC    R3           ;BUMP COUNT
        NEG    R3
        BIT    #20,UDES     ;SEE IF CORE DUMP
        BEQ    EROB        ;IF NOT: BR
        CLC
        ROR    R3           ;SET TO FC/2
        BIT    #10,MTCl     ;SEE IF WRITE OP
        BEQ    ER1          ;IF SO: BR
        BIT    #10000,RDCMD
        BEQ    EROA
        MOV    #RDATA,R3
        SUB    #2,R3        ;SET POINTER
        BR     ER2
        EROA: ADD    #RDATA,R3 ;BUILD EXPT READ ADDRESS
        BR     ER2
        ER1:  ADD    #WDATA,R3 ;BUILD EXPT WRITE ADDRESS
        ER2:  MOV    R3,CADEN  ;SAVE ADDRESS
        MOV    #7,R4
        MOV    #BAER,R1
        ER2A0: CLR    (R1)+    ;CLEAR FLAGS
        DEC    R4
        BNE    ER2A0
        CMP    R3,#BA
        BEQ    ER2A1
        INC    BAER
        ER2A1: BIT    #10,MTCl ;SEE IF WRITE OPER
        BNE    ER2B        ;IF NOT: BR
        ER2A: TST    @FC     ;SEE IF FC=0
  
```

```

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

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	: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 RECORD 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	:SEE IF PE
4142	017632	001346		BNE	ER2A	:IF SO: BR
4143	017634	032737	010000	000562	BIT	: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	000757		BR	ER2C	:GO CHECK FC FOR TM
4149	017660	032777	160000	160622	ER3: BIT	: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	: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	: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	: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	: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	
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	001065			BNE	ERPT	:IF NOT: BR (ALLOW READ OF UNKNOWN TAPES)
4186	020062	032737	000040	000672	BIT	#40,MTCT	: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	013737	015536	021154	MOV	EXCRC,CRCV	:SAVE CRC
4191	020106	013737	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	017703	160366		MOV	@CC,R3	:GET CRC CHARACTER
4197	020142	042703	177000		BIC	#177000,R3	
4198	020146	023703	015536		C1P	EXCRC,R3	
4199	020152	001402			BEQ	ER7	:IF CRC GOOD: BR
4200	020154	005237	021146		INC	CRCER	:SET ERROR FLAG
4201	020160	017703	160350		MOV	@MR,R3	:GET LRC
4202	020164	000303			SWAB	R3	
4203	020166	005703			TST	R3	
4204	020170	100000			BPL	ER10	
4205	020172	052703	000400		BIS	#400,R3	
4206	020176	042703	177000		BIC	#177000,R3	
4207	020202	023703	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	012703	000006		MOV	#6,R3	
4215	020236	005037	000706		CLR	SERFL	:CLEAR ERROR FLAG
4216	020240	005037	000722		CLR	ERSAV	
4217	020250	012704	021134		MOV	#BAER,R4	
4218	020254	005724			TST	(R4)+	:SEE IF ANY ERROR
4219	020256	001004			BNE	ERPTG	:IF SO: BR
4220	020260	005303			DEC	R3	
4221	020262	001374			BNE	ERPTT	
4222	020264	000013	021076		JMP	ERPTT	
4223	020266	005237	000706		INC	SERFL	:SET ERROR FLAG
4224	020270	017737	160224	000722	MOV	@ER,ERSAV	:SAVE ERROR REGISTER
4225	020274	032777	002000	160276	BIT	#2000,BSWR	:SEE IF PRINT
4226	020310	001420			BEQ	ERPT0	:IF SO: BR
4227	020312	022737	000002	000712	CMP	#2,RTYFL	:SEE IF READ RETRY
4228	020314	001006			BNE	ERPTG1	:IF NOT: BR
4229	020316	013703	000702		MOV	RTCNT,R3	
4230	020318	005237			INC	R3	:BUMP RETRY COUNT
4231	020320	020337	000602		CMP	R3,RETRY	:SEE IF LAST RETRY
4232	020322	001406			BEQ	ERPT0	:IF SO: BR
4233	020324	022737	000002	021140	CMP	#2,DRVER	:SEE IF TM STATUS ERROR
4234	020326	001402			BEQ	ERPT0	:IF SO: BR
4235	020328	000137	021000		JMP	ERPT0	
4236	020330	005237	000670		INC	PFLG	
4237	020332	004737	022736		JSR	PC,PAPRT	:PRINT HEADER
4238	020334	013704	000652		MOV	EMADDR,R4	
4239	020336	104000			TOUTT		:PRINT ERROR TYPE

4240	020370	004737	021156		JSR	PC,FRPRT	;PRINT F OR R
4241	020374	005737	000676		TST	TMFLG	
4242	020400	001407			BEQ	ERPT1	
4243	020402	022737	026172	000652	CMP	#MSG54,EMADDR	
4244	020410	001403			BEQ	ERPT1	
4245	020412	012704	026210		MOV	#MSG56,R4	;PRINT TM
4246	020416	104000			TTOUTT		
4247	020420	005737	021136	ERPT1:	TST	CONER	
4248	020424	001414			BEQ	ERPT2	;IF NO CONT ERROR: BR
4249	020426	012704	025213		MOV	#MSG23,R4	
4250	020432	104000			TTOUTT		;PRINT C1 TAG
4251	020434	017703	160050		MOV	@C1,R3	
4252	020440	104002			OCTPP		;PRINT CONTROL 1
4253	020442	012704	025240		MOV	#MSG23D,R4	;PRINT CS TAC
4254	020446	104000			TTOUTT		
4255	020450	017703	160044		MOV	@CS,R3	
4256	020454	104002			OCTPP		;PRINT CONT STATUS
4257	020456	005737	021140	ERPT2:	TST	DRVER	
4258	020462	001414			BEQ	ERPT3	;IF SO DRIVE ERROR: BR
4259	020464	012704	025246		MOV	#MSG23E,R4	
4260	020470	104000			TTOUTT		;PRINT DS TAG
4261	020472	017703	160024		MOV	@DS,R3	
4262	020476	104002			OCTPP		;PRINT DRIVE STATUS
4263	020500	012704	025253		MOV	#MSG23F,R4	
4264	020504	104000			TTOUTT		;PRINT ER TAG
4265	020506	017703	160012		MOV	@ER,R3	
4266	020512	104002			OCTPP		;PRINT DRIVE ERROR
4267	020514	005737	021134	ERPT3:	TST	BAER	
4268	020520	001416			BEQ	ERPT4	;IF NO BA ERROR: BR
4269	020522	012704	025226		MOV	#MSG23B,R4	
4270	020526	104000			TTOUTT		;PRINT BA TAG
4271	020530	017703	157760		MOV	@BA,R3	
4272	020534	104002			OCTPP		;PRINT BUS ADDRESS
4273	020536	012704	000255	000636	MOV	#255,TOB	
4274	020540	004737	023670		JSR	PC,TOG	;PRINT /
4275	020550	013703	021132		MOV	CADER,R3	
4276	020554	104002			OCTPP		;PRINT EXPT BUS ADDRESS
4277	020556	005737	021142	ERPT4:	TST	FCER	
4278	020562	001406			BEQ	ERPT5	;IF NO FC ERROR: BR
4279	020564	012704	025233		MOV	#MSG23C,R4	
4280	020570	104000			TTOUTT		;PRINT FC TAG
4281	020572	017703	157720		MOV	@FC,R3	
4282	020576	104002			OCTPP		;PRINT FRAME COUNT
4283	020600	012704	025221	ERPT5:	MOV	#MSG23A,R4	
4284	020604	104000			TTOUTT		;PRINT WC TAG
4285	020606	017703	157700		MOV	@WC,R3	
4286	020612	104002			OCTPP		;PRINT WORD COUNT
4287	020614	005737	021146		TST	CRCER	
4288	020620	001420			BEQ	ERPT5A	;IF NO CRC ERROR: BR
4289	020622	012704	026235		MOV	#MSG58,R4	
4290	020626	104000			TTOUTT		;PRINT CRC TAG
4291	020630	017703	157674		MOV	@CC,R3	
4292	020634	042703	177000		BIC	#177000,R3	
4293	020640	104002			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		;PRINT EXPECTED CRC
4298	020662	005737	021144	ERPT5A:	TST	LRCR	
4299	020666	001416			BEQ	ERPT6	;IF NO LRC ERROR: BR
4300	020670	012704	026243		MOV	#MSG59,R4	
4301	020674	104000			TTOUTT		;PRINT LRC TAG
4302	020676	013703	021150		MOV	ACTLRC,R3	
4303	020678	104002			OCTPP		;PRINT ACTUAL LRC
4304	020704	012737	000255	000636	MOV	#255,TOB	
4305	020712	004737	023670		JSR	PC,TOG	
4306	020716	013703	015540		MOV	EXLRC,R3	
4307	020722	104002			OCTPP		;PRINT EXPECTED LRC
4308	020724	005737	021140	ERPT6:	TST	DRVER	
4309	020726	001416			BEQ	ERPT7	;IF NO DRIVE ERROR: BR
4310	020732	032737	002000	000552	BIT	#2000,UGES	
4311	020740	001416			BEQ	ERPT7	;IF NO PE: BR
4312	020742	017704	157556		MOV	ER,R4	
4313	020746	042704	075477		BIC	#75477,R4	;MASK OUT ALL BUT BITS 15,10,7,6
4314	020752	005737			TST	R4	
4315	020754	001416			BEQ	ERPT7	;IF NO CONDITIONALS SET: BR
4316	020756	012704	025265		MOV	#MSG23H,R4	
4317	020762	104000			TTOUTT		;PRINT CC TAG
4318	020764	017703	157540		MOV	@CC,R3	
4319	020770	042703	177000		BIC	#177000,R3	;MASK CC
4320	020774	104002			OCTPP		;PRINT CHECK CHARACTERS
4321	020776	000255			NOP		
4322	020778	032737	100000	157600	ERPT7:		
4323	020780	001416			ERPX0:	BIT	#100000,@SWR
4324	020782	104002			BEQ	EPX	;SEE IF STOP ON ERROR
4325	020784	005737			STOPP		;IF NOT: BR
4326	020786	001416	000670		TST	PFLG	;SEE IF HAVE PRINTED
4327	020788	001006			BNE	ERPX	;IF SO: BR
4328	020790	032737	002000	157560	BIT	#2000,@SWR	;SEE IF SHOULD PRINT
4329	020792	001006			BNE	ERPX	;IF NOT: BR
4330	020794	000137	020352		JMP	ERPT0	;PRINT ERROR
4331	020796	005037	000670		CLR	PFLG	
4332	020798	012777	000011	157442	ERPX:	MOV	#11,@C1
4333	020800	017704	157454		MOV	@AS,R4	;DRIVE CLEAR
4334	020802	010477	157450		MOV	R4,@AS	;CLEAR AS
4335	020804	013704	000510		MOV	C1,R4	
4336	020806	005204			INC	R4	
4337	020808	152714	000100		BISB	#100,(R4)	;RESET TRE
4338	020810	013777	000552	157444	MOV	UGES,@C2	;RESET TC
4339	020812	032737	000040	000672	ERPX1:	BIT	#40,MTIC1
4340	020814	001411			BEQ	ERPX2	;IF NOT READ/WRITE OP: BR
4341	020816	005737	000676		TST	TMFLG	
4342	020818	001406			BEQ	ERPX2	;IF NOT TM TIME: BR
4343	020820	013737	021154	015536	MOV	CRCSV,EXCRC	;RESTORE CRC
4344	020822	013737	021152	015540	MOV	LRCV,EXLRC	;RESTORE LRC
4345	020824	000207			PC		;EXIT
4346	020826	000000			ERPX2:		;EXPT ADDRESS SAVE
4347	020828	000000			CADER:	0	
4348	020830	000000			BAER:	0	
4349	020832	000000			CONER:	0	
4350	020834	000000			DRVER:	0	
4351	020836	000000			FCER:	0	
	020838	000000			LRCR:	0	
	020840	000000			CRCR:	0	

4352 021150 000000
 4353 021152 000000
 4354 021154 000000

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.
 :*****

4355	021156	032737	000010	000672	FRPRT:	BIT	#10,MTC1	:SEE IF WRITE COMMAND
4356	021164	001413				BEQ	FREX	:IF SO: BR
4357	021166	032737	000002	000672		BIT	#2,MTC1	:SEE IF REVERSE
4358	021174	001404				BEQ	FR0	:IF NOT: BR
4359	021176	012704	025123			MOV	#MSG17,R4	
4370	021202	104000				TTOUTT		:PRINT R
4371	021204	000403				BR	FREX	
4372	021206	012704	025120		FR0:	MOV	#MSG16,R4	
4373	021212	104000				TTOUTT		:PRINT F
4374	021214	000207			FREX:	RTS	PC	:EXIT
4375								

4376
 4377
 4378
 4379
 4380
 4381
 4382
 4383
 4384
 4385
 4386
 4387
 4388
 4389
 4390
 4391
 4392
 4393
 4394
 4395
 4396
 4397
 4398
 4399
 4400
 4401
 4402
 4403
 4404
 4405
 4406
 4407
 4408
 4409
 4410
 4411
 4412
 4413
 4414
 4415
 4416
 4417
 4418
 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 000026 000642
 021240 005237
 021244 001371
 021246 004737 022736
 021252 032737 000010 000672
 021260 001004
 021262 012704 024762
 021266 104000
 021270 000403
 021272 012704 024767
 021276 104000
 021280 004737 021156
 021284 012704 025343
 021290 104000
 021294 104000
 021298 032777 020000 157200
 021302 001411
 021306 004737 022736
 021310 012704 027322
 021314 104000
 021318 032777 020000 157156 18:
 021322 001374
 021326 022737 000026 000672
 021330 001003

TAPG: CLR TEMP1
 MOV DVN,DCS
 BIT #10000,ADS :SET DRIVE NO.
 B' E TAPG3 :SEE IF HAVE MOL
 I TEMP1 :IF SO: BR
 BNE TAPG0 :SEE IF TIMED OUT
 JSR PC,PAPRT :WAIT FOR READY
 BIT #10,NTC1 :PRINT CYCLE NUMBER
 BNE TAPG1 :SEE IF WRITE OP
 MOV #MSG5,R4 :IF NOT: BR
 TTOUTT :PRINT WRITE ERR
 BR TAPG2
 MOV #MSG6,R4 :PRINT READ ERR
 TTOUTT :PRINT F OR R
 JSR PC,FRPRT
 TAPG2: MOV #MSG25,R4 :PRINT NO MOL ERR
 TTOUTT
 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
 BIT #20000,ADS
 BNE 18:
 CMP #26,NTC1 :AWAIT PIP RESET
 BNE TAPG3A :SEE IF WRITE TM
 :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	032737	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,BWC	:SET WORD COUNT
4445	021424	012777	000011	157056	MOV	#11,BC1	:DRIVE CLEAR
4446	021432	017777	157060	157056	MOV	BFC,BFC	: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,MTCT	:SEE IF READ OP
4450	021454	001403			BEQ	TAPG3D	:IF NOT: BR
4451	021456	012777	000003	157050	MOV	#3,BMR	:SET INTERCHANGE READ MAINT. MODE
4452	021464	013704	000672		TAPG3D: MOV	MTCT,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,MTCT	:SET INTERRUPT ENABLE AND GO
4458	021516	000240			NOP		
4459	021520	013777	000672	156762	MOV	MTCT,BC1	:EXECUTE COMMAND
4460	021526	005077	157052		CLR	BPSW	: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,BPSW	:RESET PRIORITY
4467	021560	032777	002000	157020	BIT	#2000,BSW	: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	BWSR	:++C BRANCH IF CONTINUE ON ERROR
4476	021620	100001			BPL	TAPG7	
4477	021622	104000			STOPP		
4478	021624	000137	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 021646 122716 000003 CMPB #3, (SP) ;++G SEE IF CONT C
4487 021652 001412 BEQ TTINT0 ;IF SO: BR
4488 021654 122716 000007 CMPB #7, (SP) ;++G CHECK *OR CNTL G
4489 021660 001013 BNE RETURN ;IS SOFTWARE SWITCH REGISTER USED
4490 021662 022737 000176 000606 CMP #SWREG, SWR ;NO, GET OUT
4491 021670 001007 BNE RETURN ;GO CHANGE SWREG
4492 021672 004737 024452 JSR PC, CNTG ;++G GO TO EXIT
4493 021676 000404 BR RETURN
4494
4495 021700 010046 TTINT0: MOV R0, -(SP) ;++G SAVE R0(REC CNTR)
4496 021702 004737 013774 JSR PC, TINT4 ;GO GET STALL VALUES
4497 021706 012600 MOV (SP)+, R0 ;++G RESTORE R0(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 012704 027132
021736 104000
021740 012703 000650
021744 012701 000001
021750 012702 000001
021754 012703 000000
021760 004737 023354
021764 012704 026727
021770 104000
021774 012703 000742
021778 012701 000001
022003 012702 00C 01
022007 012703 000000
022012 004737 023354
022016 005037 000736
022022 004737 022162
022026 005737 000042
022030 001405
022034 012737 000001 000742
022038 000414
022042 012704 026673 18:
022046 104000
022050 012704 026707
022054 104000
022058 013703 000736
022062 104002
022066 012704 026716
022070 104000
022074 012700 000746 28:
022078 005710 ASFQ2:
022082 104003
022086 012003
022090 104002
022094 000737
022098 004737 022366
022102 004737 022562
022106 022737 000007 000736
022110 001405
022114 005237
022118 005237 000736
022122 005237
022126 005737 000742
022130 004737
022134 005100
022138 005737 000734
022142 001317
022146 000000

```

*****
: AUTO SEQUENCE
: THIS ROUTINE ENTERED VIA STARTING ADDRESS 240
: WILL EXERCISE ALL AVAILABLE SLAVES ON ALL AVAILABLE
: DRIVES IN BOTH PE AND NRZ ACCORDING TO THE PRESELECTED
: TEST PLAN. IF NRZ ONLY, PE TESTING WILL NOT BE ATTEMPTED.
*****

ASEQ:  MOV    #MSG108,R4
      TTOUTT
      MOV    #NRZOF,R5      :PRINT NRZ ONLY REQUEST
      MOV    #1,R1          :SET ADDRESS OF FLAG
      MOV    #1,P2          :SET SIZE OF ENTRY
      MOV    #0,R3          :SET UPPER LIMIT
      JSR    PC,TTR         :SET LOWER LIMIT
      MOV    #MSG104,R4     :GO GET RESPONSE
      TTOUTT
      MOV    #ASEQCF,R5    :REQUEST CONT OR NOT
      MOV    #1,R1          :SET ADDRESS OF ENTRY
      MOV    #1,R2          :SET SIZE OF ENTRY
      MOV    #0,R3          :SET UPPER LIMIT
      JSR    PC,TTR         :SET LOWER LIMIT
      CLR    ADRVN          :GO GET INPUT
      JSR    PC,HRDS        :CLEAR DRV NUM
      TST    B#42          :GO SELECT HARDWARE CONFIGURATION
      BEQ    18             :AUTO MODE? ++ C.W
      MOV    #1,ASEQCF      :BRANCH - IF NO ++ C.W
      BR     28             :SET AUTO SEQ FLAG ++ C.W
      MOV    #MSG101,R4    :DO AUTO SEQ TESTS ++ C.W
      TTOUTT
      MOV    #MSG102,R4    :PRINT DIVIDER
      OCTPP
      MOV    ADRVN,R3      :PRINT TM02 NUMBER
      MOV    #MSG103,R4    :PRINT TMU2
      TTOUTT
      MOV    #UN1,R0       :PRINT SLAVE HDR
      TST    (R0)          :POINT TO START OF SLAVE TABLE
      BAI    ASEQ3         :SEE IF END
      MOV    (R0)+,R3      :IF SO: BR
      OCTPP
      BR     ASEQ2         :PRINT SLAVE TABLE
      JSR    PC,AMOD1      :DO ALL
      JSR    PC,AMOD2      :GO DO MODE 1(NRZ)
      CMP    #7,ADRVN     :GO DO MODE 2(PE)
      BEQ    ASEQX         :SEE IF DONE ALL DRIVES
      INC    ADRVN        :IF SO: BR
      BR     ASEQ1         :BUMP DRIVE NUMBER
      TST    ASEQCF        :CONTINUE
      BEQ    18            :CONTINUOUS AUTO SEQUENCE? ++ C.W
      JSR    PC,TEND       :BRANCH - IF NO ++ C.W
      TST    ASEQF         :GO DO ACT END OF PASS
      BNE    ASEQO        :CONTINUE
      HALT
18:

```

```
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
```

022162 005037 005156
022163 005037 000642
022164 012777 000640 156320
022165 012777 000736 156312
022166 012701 156322
022167 012777 010000 156300
022168 001403
022169 005729
022170 000137 022122
022171 012701 002007
022172 012701 140010
022173 001370
022174 005000
022175 012701 000746
022176 005737 003040
022177 001410
022178 122737 000006 000041
022179 001004
022180 005737 000736
022181 001001
022182 005200
022276 010077 156240
022277 032777 010000 156212
022278 001403
022279 005237 000642
022280 010021
022281 022700 000007
022282 001402
022283 005200
022284 000762
022285 005737 000642
022286 001731
022287 013737 000642 005156
022288 000337 000642
022289 053737 000642 005156
022290 012711 177777
022364 000207

HRDS: CLR REOTC
CLR TEMP1
MOV #40,BCS
MOV ADRVN,BCS
MOV BDT,R1
BIT #10000,BCS
BEQ HRDS1
HRDS0: TST (SP)+
JMP ASEQ4
HRDS1: BIC #2007,R1
CMP #140010,R1
BNE HRDS0
CLR R0
MOV #UN1,R1
TST CHNFLG
BEQ HRDS2
CMPB #6,#41
BNE HRDS2
TST ADRVN
BNE HRDS2
INC R0
HRDS2: MOV R0,BC2
BIT #10000,BCS
BEQ HRDS3
INC TEMP1
MOV R0,(R1)+
HRDS3: CMP #7,R0
BEQ HRDS4
INC R0
BR HRDS2
HRDS4: TST TEMP1
BEQ HRDS0
MOV TEMP1,REOTC
SWAB TEMP1
BIS TEMP1,REOTC
MOV #1,(R1)
RTS PC

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

```

4603
4604
4605      ;SUBROUTINE TO SELECT NR? AUTO TEST MODE*****
4606      022366 005037 000654      AMOD1: CLR      BLCNTR      ;ASSURE BLOCK COUNTER IS 0
4607      022372 012701 000746      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              COM      (R1)
4610      022404 001402              BEQ      AMOD1B      ;IF FILLED ALL SLAVES: BR
4611      022406 005111              COM      (R1)
4612      022410 000772              BR      AMOD1A      ;ELSE DO ALL
4613      022412 005111      AMOD1B: COM      (R1)
4614      022414 004737 005172      JSR      PC,RWMDA      ;GO REWIND ALL AVAIL SLAVES
4615      022420 012737 000006 000740      MOV      #6,ABLCNT      ;SET NUMBER OF BLOCKS FOR MODE 1
4616      022426 012737 174000 000556      MOV      #-4000,FCNTR      ;SET FC = 4000
4617      022432 012737 000100 000554      MOV      #100,RCNTR      ;SET REC CNTR = 100
4618      022438 012737 000736 000550      MOV      ADRVN,DVN      ;SELECT DRIVE
4619      022450 012737 000001 000560      MOV      #1,PATRN      ;SELECT PATTERN 1
4620      022456 005037 000564      CLR      TMEX      ;ASSURE NO TMK
4621      022462 005037 000566      CLR      INTRF      ;ASSURE NORMAL READ
4622      022468 004737 000414      JSR      PC,STAUTO      ;GO DO AUTO MODE 1
4623      022472 012737 000010 000560      MOV      #10,PATRN      ;SELECT PATTERN 10
4624      022500 004737 003414      JSR      PC,STAUTO      ;GO DO PATTERN 10
4625      022504 012737 000014 000560      MOV      #14,PATRN      ;SELECT PATTERN 14
4626      022512 004737 003414      JSR      PC,STAUTO
4627      022516 005737 000650      TST      NRZOF      ;SEE IF NRZ ONLY
4628      022522 001411      BEQ      AMOD1C      ;IF NOT: BR
4629      022524 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      PC      ;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
        CLR BLCNTR ;CLEAR BLOCK CNTR
AMOD2A: MOV #UN1, R1 ;SET START OF SLAVE TABLE
        BIC #700, (R1) ;CLEAR NRZ
        BIS #2300, (R1)+ ;SET TO PE NORM, ODD
        COM (R1) ;SEE IF END OF TABLE
        BEQ AMOD2B ;IF SO: BR
        COM (R1)
        BR AMOD2A ;CONTINUE
AMOD2B: COM (R1)
        JSR PC, RWDA ;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 PC ;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
4722022736 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
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,ASLR ;SEE IF RANDOM DATA
BEQ PAPRTB ;IF NOT: BR
PAPRTA: MOV #122,TOB
JSR PC,TOB ;PRINT R
BR PAPRTD
PAPRTB: TST ASEOF ;SEE IF AUTO SEQ

4723	023132	001403		BEQ	PAPRTC		:IF NOT: BR
4724	023134	005737	000560	TST	PATRN		:SEE IF AUTO RANDOM
4725	023140	100764		BMI	PAPRTA		:IF SO: BR
4726	023142	013703	000560	PAPRTC: MOV	PATRN,R3		
4727	023146	104002		OCTPP			:PRINT PATRN NUMBER
4728	023150	012704	025054	PAPRTD: MOV	#MSG13,R4		
4729	023154	104000		TTOUTT			:PRINT BLOCK NO. HEADER
4730	023156	013703	000654	MOV	BLCNTR,R3		
4731	023162	104002		OCTPP			:PRINT NUMBER
4732	023164	012704	025062	MOV	#MSG14,R4		
4733	023170	104000		TTOUTT			:PRINT REC NO. HEADER
4734	023172	010003		MOV	R0,R3		
4735	023174	032737	000010 000672	BIT	#10,MTCT		:SEE IF WRITE OPERATION
4736	023202	001404		BEQ	PAPRT1		:IF SO: BR
4737	023204	032737	010000 000562	BIT	#10000,RDCMD		:SEE IF READ REVERSE
4738	023212	001016		BNE	PAPRT3		:IF SO: BR
4739	023214	013703	000554	PAPRT1: MOV	RCNT,R3		
4740	023220	005737	000676	TST	TMFLG		:SEE IF TAPE MARK TIME
4741	023224	001010		BNE	PAPRT2		:IF SO: BR
4742	023226	022737	012260 000662	CMP	#B1,RTRN		
4743	023234	001003		BNE	PAPRTY		:IF NOT BACK SPACE: BR
4744	023236	005737	000712	TST	RTYFL		
4745	023242	001402		BEQ	PAPRT3		:IF NOT RETRY: BP
4746	023244	160003		PAPRTY: SUB	R0,R3		:GET RECORD NUMBER
4747	023246	005203		PAPRT2: INC	R3		
4748	023250	104002		PAPRT3: OCTPP			:PRINT RECORD NUMBER
4749	023252	012737	000055 000636	MOV	#55,TOB		:LOAD DASH (-)
4750	023256	004737	023670	JSR	PC,TOG		:PRINT DASH (-)
4751	023262	013703	000554	MOV	RCNT,R3		
4752	023270	104002		OCTPP			:PRINT RECORD COUNT
4753	023272	012704	024774	MOV	#MSG7,R4		
4754	023276	104000		TTOUTT			:PRINT RECORD SIZE HEADER
4755	023300	013703	000556	MOV	FMCNT,R3		:GET CHARACTER COUNT
4756	023304	005303		DEC	R3		
4757	023306	005103		COM	R3		:REMOVE TWOS COMPLEMENT
4758	023310	104002		OCTPP			:PRINT RECORD SIZE
4759	023312	012737	000001 000664	MOV	#1,HDRFL		:SET HEADER FLAG
4760	023320	000207		RTS	PC		:RETURN
4761							

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 023354 005037 000642 TTR: CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG  
4797 023360 005000 CLR RO  
4798 023362 104010 TTR0: TTINN ;GO READ CHARACTER  
4799 023364 122737 000015 000640 CMPB #15,TIB ;++G SEE IF CR  
4800 023372 001004 BNE TTR1 ;IF NOT: BR  
4801 023374 005737 000642 TST TEMP1 ;SEE IF FIRST CHARACTER  
4802 023400 001436 BEQ TTR5 ;IF SO: BR  
4803 023402 000426 BR TTR2 ;++G ELSE GO LOAD VALUE  
4804 023404 122737 000060 000640 TTR1: CMPB #60,TIB ;++G SEE IF CHAR IS LESS THAN 0  
4805 023412 101401 BLOS TTR1A ;IF NOT: BR  
4806 023414 000431 BR TTR1A ;++G ELSE GO TO ERROR  
4807 023416 122737 000070 000640 TTR1A: CMPB #70,TIB ;++G SEE IF CHAR IS GREATER THAN 7  
4808 023424 101001 BHI TTR1B ;IF NOT: BR  
4809 023426 000424 BR TTR1B ;++G ELSE GO TO ERROR  
4810 023430 005237 000642 TTR1B: INC TEMP1 ;SET FIRST CHARACTER FLAG  
4811 023434 006300 ASL RO  
4812 023436 006300 ASL RO ;SHIFT 3 LEFT  
4813 023440 006300 ASL RO  
4814 023442 042737 177770 000640 BIC #177770,TIB ;STRIP ASCII  
4815 023450 053700 000640 BIS TIB,RO ;LOAD CHARACTER  
4816 023454 005301 DEC R1 ;SEE IF DONE  
4817 023456 001341 BNE TTR0 ;IF NOT: BR  
4818 023460 020002 TTR2: CMP RO,R2 ;SEE IF EXCEEDED MAXIMUM LIMIT  
4819 023462 101401 BLOS TTR3 ;IF NOT: BR  
4820 023464 000405 BR TTR3 ;++G ELSE GO TO ERROR  
4821 023466 020300 TTR3: CMP R3,RO ;SEE IF BELOW MINIMUM LIMIT  
4822 023470 101401 BLOS TTR4 ;IF NOT: BR  
4823 023472 000402 BR TTR4 ;++G ELSE GO TO ERROR  
4824 023474 010015 TTR4: MOV RO,(R5) ;LOAD VALUE  
4825 023476 000207 TTR5: RTS PC ;EXIT  
4826 023500 012704 025763 TTR5: MOV #MSG43,R4  
4827 023504 104000 TTR5: TTOUTT ;PRINT?  
4828 023506 162716 000020 SUB #20,(SP) ;RESET SP TO START OF VALUE ROUTINE  
4829 023512 000207 RTS PC ;REDO VALUE ENTRY
```

```

4830
4831
4832
4833
4834 023514 005277 155070
4835 023520 105777 155064
4836 023524 100375
4837 023526 017737 155060 000640
4838 023528 042737 000200 000640
4839 023532 105777 155046
4840 023536 100375
4841 023550 113777 000640 155040
4842 023556 000207
4843
4844
4845
4846 023560 112437 000636
4847 023564 122737 000043 000636
4848 023572 001444
4849 023574 122737 000045 000636
4850 023602 001407
4851 023604 122737 000041 000636
4852 023612 001435
4853 023614 004737 023670
4854 023620 000757
4855 023622 112737 000015 000636
4856 023630 004737 023670
4857 023634 012703 000006
4858 023640 005037 000636
4859 023644 004737 023670
4860 023650 005303
4861 023652 001372
4862 023654 112737 000012 000636
4863 023662 004737 023670
4864 023666 000734
4865 023670 105777 154720
4866 023674 100375
4867 023676 113777 000636 154712
4868 023704 000207
4869 023706 012703 000002
4870 023712 012737 000007 000636
4871 023720 004737 023670
4872 023724 005303
4873 023726 001371
4874 023730 000713
4875
4876
;TTY READ SUBROUTINE*****
TTIN: INC @TKS
TTIN1: TSTB @TKS
BPL TTIN1
MOV @TKB,TIB
BIC #200,TIB ;++G STRIP PARITY BIT
TTIN2: TSTB @TPS
BPL TTIN2
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
PC
TEX: RTS
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      023732 000403
4881      023732 012737 000001 024164  OCTPE:  BR    OCTPE1
4882      023732 010304      OCTPE1:  MOV  #1,OFL
4883      023732 001006      OCTPE1:  MOV  R3,R4          ;SEE IF NUMBER IS ZERO
4884      023732 005737 024164      OCTPE1:  BNE  OCTP0          ;IF NOT ZERO: BR
4885      023732 001006      OCTPE1:  TST  OFL
4886      023732 004737 024144      OCTPE1:  BNE  OCTP0
4887      023732 000450      OCTPE1:  JSR  PC,OCTPG1      ;ELSE PRINT ZERO
4888      023732 032704 100000      OCTPE1:  BR    OCTP3          ;++G SPACE AND EXIT
4889      023732 001406      OCTPE1:  BIT  #100000,R4      ;SEE IF MSD = 1
4890      023732 012704 000001      OCTPE1:  BEQ  OCTP1          ;IF NOT: BR
4891      024000 004737 024122      OCTPE1:  MOV  #1,R4
4892      024000 000137 024016      OCTPE1:  JSR  PC,OCTPG
4893      024010 005004      OCTPE1:  JMP  OCTP2          ;PRINT 1
4894      024012 004737 024122      OCTP1:  CLR  R4
4895      024016 010304      OCTP2:  JSR  PC,OCTPG
4896      024020 006004      OCTP2:  MOV  R3,R4          ;PRINT 0
4897      024022 006004      OCTP2:  ROR  R4
4898      024024 006004      OCTP2:  ROR  R4
4899      024026 006004      OCTP2:  ROR  R4          ;POSITION DIGIT
4900      024030 000304      OCTP2:  ROR  R4
4901      024032 004737 024122      OCTP2:  SWAB R4
4902      024036 010304      OCTP2:  JSR  PC,OCTPG
4903      024040 006004      OCTP2:  MOV  R3,R4          ;PRINT DIGIT 2
4904      024042 000304      OCTP2:  ROR  R4
4905      024044 004737 024122      OCTP2:  SWAB R4
4906      024046 010304      OCTP2:  JSR  PC,OCTPG
4907      024050 006104      OCTP2:  MOV  R3,R4          ;PRINT DIGIT 3
4908      024052 006104      OCTP2:  ROL  R4
4909      024054 000304      OCTP2:  ROL  R4
4910      024056 004737 024122      OCTP2:  SWAB R4
4911      024060 010304      OCTP2:  JSR  PC,OCTPG
4912      024064 006004      OCTP2:  MOV  R3,R4          ;PRINT DIGIT 4
4913      024066 006004      OCTP2:  ROR  R4
4914      024070 006004      OCTP2:  ROR  R4
4915      024072 006004      OCTP2:  ROR  R4
4916      024074 004737 024122      OCTP2:  JSR  PC,OCTPG
4917      024100 010304      OCTP2:  MOV  R3,R4
4918      024102 004737 024122      OCTP2:  JSR  PC,OCTPG
4919      024106 012737 000240 000636  OCTP3:  MOV  #240,TOB      ;PRINT DIGIT 5
4920      024114 004737 023670      OCTP3:  JSR  PC,TOG
4921      024120 004737 177770      OCTP3:  RTS  PC          ;PRINT SPACE
4922      024122 042704      OCTP3:  BIC  #177770,R4      ;EXIT
4923      024126 001004      OCTP3:  BNE  OCTPG0
4924      024130 005737 024164      OCTP3:  TST  OFL
4925      024134 001001      OCTP3:  BNE  OCTPG0
4926      024136 000207      OCTP3:  RTS  PC
4927      024140 005737 024164      OCTPG0:  INC  OFL
4928      024144 052704 000260      OCTPG1:  BIS  #260,R4
4929      024150 010437 000636      OCTPG1:  MOV  R4,TOB
4930      024154 004737 023670      OCTPG1:  JSR  PC,TOG
4931      024160 010304      OCTPG1:  MOV  R3,R4
4932      024162 000207      OCTPG1:  RTS  PC
4932      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	
024224	012777	000061 154364
024232	000403	
024234	012777	000060 154354
024242	006137	000636
024246	005304	
024250	001354	
024252	000207	
024254	013703	000646
024260	000303	
024262	004737	024166
024266	013703	000646
024272	004737	024166
024276	000207	

;ATA CHARACTER OUTPUT SUBROUTINE*****

```

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

```

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

```

SNPT: MOV    R3,R4
      SWAB   R4
      ROR    R4
      ROR    R4
      ROR    R4
      ROR    R4
      ROR    R4
      JSR    PC,SNPG      ;PRINT FIRST DIGIT
      MOV    R3,R4
      SWAB   R4
      JSR    PC,SNPG      ;PRINT SECOND DIGIT
      MOV    R3,R4
      ROR    R4
      ROR    R4
      ROR    R4
      ROR    R4
      JSR    PC,SNPG      ;PRINT THIRD DIGIT
      MOV    R3,R4
      JSR    PC,SNPG      ;PRINT FOURTH DIGIT
      RTS    PC
      SNPG: MOV    #260,TOB
      BIC    #177760,R4
      BIS    R4,TOB
      JSR    PC,TOG
      RTS    PC

```

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

```
4989          024402 000000          :LOCATIONS USED:
4990          024404 000000          TEMPST: .WORD 0
4991          024406 000000          COUNT: .WORD 0
4992          024410 022737 000176 000606 CKSWR: RDSW: .WORD 0
4993          024416 001123          CMP #SWREG,SWR          :SOFTWARE SWITCH REG PRESENT
4994          024420 105777 154164          BNE OUT          :NO GET OUT
4995          024424 100120          TSTB #TKS          :YES WAIT FOR
4996          024426 017737 154160 000640          BPL OUT          :READY GET CHARACTER
4997          024434 042737 177600 000640          MOV #TKB,TIB          :AND STRIP OFF
4998          024442 022737 000007 000640          BIC #177600,TIB          :THE GARBAGE
4999          024450 001106          CMP #7,TIB          :IS IT A '<^G>'
5000          024452 012704 027436          BNE OUT
5001          024456 104000          CNTG: MOV #SCNTG,R4
5002          024460 012704 027442          TOUTT
5003          024464 104000          CNTLU: MOV #SMSWR,R4
5004          024466 017703 154114          TOUTT
5005          024472 004737 023740          MOV #SI'R,R3
5006          024476 012704 027451          JSR PC,OCPE
5007          024502 104000          MOV #SMNEW,R4
5008          024510 005037 024402          TOUTT
5009          024516 012737 000007 024404          SREAD: CLR TEMPST
5010          024520 104010          1$: MOV #7,COUNT
5011          024524 122737 000025 000640          TTINN
5012          024528 001001          CMPB #25,TIB          :GO READ A CHARACTER
5013          024532 000737          BNE 2$          :IS IT A ^U?
5014          024536 122737 000015 000640          BR CNTLU          :BRANCH IF NOT
5015          024540 001013          3$: CMPB #15,TIB          :START OVER
5016          024544 012737 000200 024406          2$: BNE 4$          :IS IT A <CR>?
5017          024548 012704 027461          MOV #200,RDSW          :BRANCH IF NOT
5018          024552 104000          MOV #MCRLF,R4
5019          024556 022737 000007 024404          TOUTT
5020          024560 001037          CMP #7,COUNT          :WAS IT FIRST CHARACTER
5021          024564 000437          BNE 7$          :CHANGE SWR IF NOT FIRST ONE
5022          024568 122737 000060 000640          BR OUT          :GET OUT
5023          024572 003004          4$: CMPB #60,TIB
5024          024576 122737 000067 000640          BGT 5$
5025          024600 002004          CMPB #67,TIB
5026          024604 012704 025763          BGE 6$
5027          024608 104000          5$: MOV #MSG43,R4
5028          024612 000744          TOUTT
5029          024616 006337          BR 3$          :START OVER IF NOT LEGAL CHARACTER
5030          024620 006337          6$: ASL TEMPST
5031          024624 006337          ASL TEMPST
5032          024630 006337          ASL TEMPST
5033          024634 142737 000060 000640          BICB #60,TIB          :GET NITTY-GRITTY
5034          024638 153737 000640 024402          BISB TIB,TEMPST
5035          024642 005337          DEC COUNT          :ONLY WANT 6 DIGITS
5036          024646 001737          BEQ 1$
5037          024650 000717          BR 1$
5038          024654 013777 024402 153720          7$: MOV TEMPST,@SWR
5039          024658 000207          OUT: RTS PC          :CHANGE SWITCH REGISTER CONTENTS
5040          :HALT HANDLER*****          :RETURN TO BODY OF PROGRAM
5041          024670 000000          STOP: HALT
5042          024672 104004          CKSWR
5043          024674 000207          RTS PC          :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
024704	011666	000002
024710	162716	000002
024714	013646	
024716	063716	120724
024722	013607	
024724	023560	
024726	023732	
024730	024410	
024732	024670	
024734	023514	
	104000	
	104002	
	104004	
	104006	
	104010	

;TRAP HANDLER*****

153700	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		
	TTOUTT=	104000		
	OCTPP=	104002		
	CKSWRR=	104004		
	STOPP=	104006		
	TTINN=	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 /X*BN #/
5096
5097 025062 051052 020116 043 MSG14: .ASCII /*RN #/
5098
5099 025067 045 020041 020040 MSG15: .ASCII /X! BAD RECORDX#/
5100 025074 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
  
```

025213	045	051503	020061	MSG23: .ASCII /XCS1 #/
025220	043			
025221	045	041527	021440	MSG23A: .ASCII /XWC #/
025226	041045	020101	043	MSG23B: .ASCII /XBA #/
025233	045	041506	021440	MSG23C: .ASCII /XFC #/
025240	041445	031123	021440	MSG23D: .ASCII /XCS2 #/
025246	042045	020123	043	MSG23E: .ASCII /XDS #/
025253	045	051105	021440	MSG23F: .ASCII /XER #/
025260	040445	020123	043	MSG23G: .ASCII /XAS #/
025265	045	045503	021440	MSG23H: .ASCII /XCK #/
025272	042045	020102	043	MSG23I: .ASCII /XDB #/
025277	045	051115	021440	MSG23J: .ASCII /XMR #/
025304	042045	020124	043	MSG23K: .ASCII /XDT #/
025311	045	041524	021440	MSG23L: .ASCII /XTC #/
025316	051445	020116	043	MSG23M: .ASCII /XSN #/
025323	045	047041	020117	MSG24: .ASCII /X!NO INTERRUPTX#/
025330	047111	042524	051122	
025336	050125	022524	043	
025343	045	047041	020117	MSG25: .ASCII /X!NO MOL: HALTX#/
025350	047515	035114	044040	
025356	046101	022524	043	
025363	045	051104	050117	MSG26: .ASCII /XGROPS: #/
025370	035123	021440		
025374	050045	041511	051513	MSG27: .ASCII /XPICKS: #/
025402	020072	043		
025405	045	043		MSG28: .ASCII /X#/
025407	045	052045	030115	MSG30: .ASCII 'XXTM02-TU16/TE16 AU"O SEQUENCE (CZTUA10)X#'
025414	026462	052524	033061	
025421	052057	050503	020066	
025428	052501	047524	051440	
025435	052503	042524	041516	
025442	020105	041450	052132	
025449	040525	030111	022451	
025456	043			
025461	045	041445	052132	MSG31: .ASCII 'XXCZTUA10 TM02-TU16/TE16 RELIABX#'
025466	040525	030111	052040	
025474	030115	026462	052524	

5178	025502	033061	052057	030505	
5179	025510	020066	042522	042512	
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	025663	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%#/ 043
5235	026050	052117	047440	020116	
5236	026056	042522	044527	042116	
5237	026064	020072	040510	052114	
5238	026072	021445			
5239					
5240	026074	047040	052117	040440	MSG49: .ASCII / NOT AVAIL #/ 043
5241	026102	040526	046111	021440	
5242	026110	044440	046114	043505	MSG50: .ASCII / ILLEGAL DRIVE TYPE #/ 043
5243	026116	046101	042040	044522	
5244	026124	042526	052040	050131	
5245	026132	020105	043		
5246	026138	045	042045	044522	MSG52: .ASCII /%XDRIVE NUMBER = #/ 043
5247	026142	042526	047040	046525	
5248	026150	042502	020122	020075	
5249	026156	043			
5250					
5251	026157	045	047506	046522	MSG53: .ASCII /%FORMAT = #/ 043
5252	026164	052101	036440	021440	
5253					
5254	026172	053452	020105	046524	MSG54: .ASCII /*WE TM#/ 043
5255	026200	043			
5256					
5257	026201	052	042523	052040	MSG55: .ASCII /*SE TM#/ 043
5258	026206	021515			
5259					
5260	026210	052040	021515		MSG56: .ASCII / TM#/ 043
5261					
5262	026214	047040	047117	042455	MSG57: .ASCII / NON-EXIST SLAVEN#/ 043
5263	026222	044530	052123	051440	
5264	026230	040514	042526	043	
5265	026235	045	051103	020103	MSG58: .ASCII /%CRC #/ 043
5266	026242	043			
5267	026248	043	051114	020103	MSG59: .ASCII /%LRC #/ 043
5268	026250	043			
5269	026251	052	020104	043	MSG60: .ASCII /*D #/ 043
5270	026255	052	020120	043	MSG61: .ASCII /*P #/ 043
5271	026261	052	020106	043	MSG62: .ASCII /*F #/ 043
5272					
5273	026265	045	047452	044522	MSG64: .ASCII /%ORIGINAL ERROR*#/ 043
5274	026272	044507	040516	020114	
5275	026300	051105	047522	025122	
5276	026306	043			
5277					
5278	026307	045	042522	051124	MSG65: .ASCII /%RETRY: #/ 043
5279	026314	035131	021440		
5280					
5281	026320	020452	042523	051040	MSG66: .ASCII /*!SE RTRY #/ 043
5282	026326	051124	020131	043	
5283					
5284	026333	052	042441	040522	MSG67: .ASCII /*!ERASE#/ 043
5285	026340	042523	043		
5286					
5287	026343	045	042522	042522	MSG68: .ASCII /%REREV: #/ 043
5288	026350	035126	021440		
5289	026354	052045	050101	020105	MSG69: .ASCII /%TAPE MARK = #/ 043

5290	026362	040515	045522	036440	
5291	026370	021440			
5292					
5293	026372	020445	047516	042040	MSG70: .ASCII /%!NO DRY FROM REWIND: HALT%#/
5294	026377	032522	043040	047522	
5295	026400	020115	042522	044527	
5296	026406	042116	020072	040510	
5297	026414	052114	021445		
5298	026422	047040	047117	042455	MSG71: .ASCII / NON-EXIST DRIVE#/
5299	026426	044530	052123	042040	
5300	026434	044530	042522	043	
5301	026442	044522	042522	053506	MSG72: .ASCII /%REFWD: #/
5302	026447	043	021440		
5303	026454	035104	042522	051122	MSG73: .ASCII /%WTERR: #/
5304	026460	053445	042522		
5305	026466	020072	043		
5306	026471	045	042522	044507	MSG74: .ASCII /%REG STER START = #/
5307	026476	052123	051105	051440	
5308	026504	040524	052122	036440	
5309	026512	021440			
5310	026514	053045	041505	047524	MSG75: .ASCII /%VECTOR = #/
5311	026522	020122	020075	043	
5312	026527	045	042504	042522	MSG76: .ASCII /%DEREV: #/
5313	026534	035126	021440		
5314	026540	042045	043105	042127	MSG77: .ASCII /%DEFWD: #/
5315	026546	020072	043		
5316	026551	045	047041	047117	MSG78: .ASCII /%!NON-RETRYABLE WRITE ERROR: ER #/
5317	026556	051055	052105	054522	
5318	026564	041101	042514	053440	
5319	026572	044522	042524	042440	
5320	026600	051122	051117	020072	
5321	026606	051105	021440		
5322	026612	020445	047516	026516	MSG79: .ASCII /%!NON-RETRYABLE READ ERROR: ER #/
5323	026620	042522	051124	040531	
5324	026626	046102	020105	042522	
5325	026634	042101	042440	051122	
5326	026642	051117	020072	051105	
5327	026650	021440			
5328	026652	020445	042441	042116	MSG100: .ASCII /%!!END OF PASS %#/
5329	026660	047440	020106	040520	
5330	026666	051522	022440	043	
5331	026673	045	025052	025052	MSG101: .ASCII /%*****%#/
5332	026700	025052	025052	022452	
5333	026707	052	046524	031060	MSG102: .ASCII /*TM02 #/
5334	026714	021440			
5335	026716	051452	040514	042526	MSG103: .ASCII /*SLAVES #/
5336	026724	020122	043		
5337	026727	045	052501	047524	MSG104: .ASCII /%AUTO CONT: #/
5338	026734	041440	047117	035124	
5339	026742	021440			
5340	026744	051045	041505	053117	MSG105: .ASCII /%RECOVERED#/
5341	026752	051105	042105	043	
5342	026757	052	020441	040502	MSG106: .ASCII /*!!BAD TAPE OVERFLOW#/
5343	026764	020104	040524	042520	
5344	026772	047440	042526	043122	
5345	027000	047514	021527		

CZTUA10 TM02-TU16/TE16 RELIAB
CZTUA1.P11 16-JUN-82 10:46

MACY11 30(1046) 16-JUN-82 11:10 N 9
PAGE 117

SEQ 0117

```
5346 027004 051045 053505 047111 MSG16A: .ASCII /%REWIND TAPE; RESTART AT BLOCK ONE#/
5347 027012 020104 040524 042520
5348 027020 020073 042522 052123
5349 027026 021101 020124 052101
5350 027032 041040 047514 045503
5351 027038 047440 042516 042503
5352 027044 042516 020441 047125
5353 027050 042516 047503 042526
5354 027056 040522 046102 020105
5355 027102 040522 020104 050123
5356 027108 040522 042101 051040
5357 027114 040522 051117 020104
5358 027120 040522 052106 047440
5359 027126 040522 040524 042520
5360 027132 040522 055122 047440
5361 027138 040522 055131 021440
5362 027144 040522 050041 051517
5363 027150 040522 047511 020116
5364 027156 040522 052123 044440
5365 027202 040522 042522 051124
5366 027208 040522 051525 042520
5367 027214 040522 041040 042101
5368 027220 040522 050101 021505
5369 027226 040522 050105 040505
5370 027232 040522 021440
5371 027238 040522 042101 052040
5372 027244 040522 020105 050123
5373 027250 040522 022523 043
5374 027256 040522 051440 043117
5375 027262 035124 021440
5376 027268 020045 040510 042122
5377 027274 020072 043
5378 027280 040510 040510
5379 027286 040510 040505
5380 027292 040510 047522
5381 027298 040510 047125
5382 027304 040510 052111
5383 027310 040510 047111
5384 027316 040510 020072
5385 027322 040510 042522
5386 027328 040510 042522
5387 027334 040510 042522
5388 027340 040510 042522
5389 027346 040510 042522
5390 027352 040510 042522
5391 027358 040510 042522
5392 027404 040510 042522
5393 027410 040510 042522
5394 027416 040510 042522
5395 027422 040510 042522
5396 027428 040510 042522
5397 027434 040510 042522
5398 027440 040510 042522
5399 027446 040510 042522
5400 027452 040510 042522
5401 027458 040510 042522
```

MSG107: .ASCII /%!!UNRECOVERABLE BAD SPOT/
.ASCII /%BAD RECORD LEFT ON TAPE%#/
MSG108: .ASCII /%NRZ ONLY: #/
MSG109: .ASCII /%!!POSITION LOST IN RETRY#/
MSG110: .ASCII /%SUSPECT BAD TAPE#/
MSG111: .ASCII /%REPEAT: #/
MSG112: .ASCII /%BAD TAPE SPOTS%#/
MSG113: .ASCII /%SOFT: #/
MSG114: .ASCII /%HARD: #/
MSG115: .ASCII /%!!HARD READ ERROR#/
MSG116: .ASCII /%UNIT REWINDING: TEST WILL START AT BOT#/
MSG120: .ASCII /%REMOVE TM02 FROM UNIT UNDER TEST%#/
SCNT6: .ASCII /%G#/

CZTUAIO TMO2-TU16/TE16 RELIAB MACY11 30(1046) 16-JUN-82 11:10 B 10
CZTUA1.P11 16-JUN-82 10:46 PAGE 118

SEQ 0118

```
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
```

MACY11 30(1046) 16-JUN-82 11:10 C 10
PAGE 120
CROSS REFERENCE TABLE -- USER SYMBOLS

C 10
PAGE 120

[illegible]

MACY11 30(1046) 16-JUN-82 11:10 PAGE 121
CROSS REFERENCE TABLE -- USER SYMBOLS

D 10
PAGE 121

[illegible]

CZTUA10 TM02-TU16/TE16 RELIAB
CZTUA1.P11 16-JUN-82 10:46

MACY11 30(1046) 16-JUN-82 11:10 E 10
CROSS REFERENCE TABLE -- USER SYMBOLS PAGE 122

SEQ 0121

DATA4	003002	1834#							
DATA5	003004	1835#							
DATA6	003006	1836#							
DATA7	003010	1837#							
DATBL	002770	1829#	3453						
DATER1	001130	1685#	2040	3835*					
DATR	015102	1971#	3237	3633#					
DATRO	015120	3637#	3640						
DATO	014412	1830#	3475#						
DATOA	014444	3482#	3490	3494	3497				
DATOB	014452	3483#	3484						
DATOC	014516	3488#	3493#						
DATOD	014524	3498#	3506#						
DATOE	014534	3500#	3508#						
DATOF	C:4546	3502#	3504#						
DAT1	014556	1831#	3511#						
DAT1A	014562	3512#	3521#	3544	3549	3597			
DAT1B	014566	3513#	3521#						
DAT10	014700	1833#	3526#						
DAT10A	014712	3567#	3571#						
DAT11	014730	1839#	3576#						
DAT11A	014736	3578#	3581#						
DAT12	014750	1840#	3586#						
DAT12A	014760	3588#	3591#						
DAT13	014772	1841#	3596#						
DAT14	015002	1842#	3601#						
DAT14A	015014	3604#	3608#						
DAT15	015032	1843#	3613#						
DAT15A	015036	3614#	3620#						
DAT15B	015046	3616#	3618#						
DAT2	014576	1832#	3520#						
DAT3	014602	1833#	3525#						
DAT3A	014610	3527#	3538#						
DAT3B	014614	3528#	3531#						
DAT4	014626	1834#	3536#						
DAT5	014636	1835#	3543#						
DAT6	014644	1836#	3548#						
DAT7	014652	1837#	3553#						
DAT7A	014666	3556#	3559#						
DB	000532	1531#							
DCHK	015544	2779	2974	3750#					
DCHKO	015574	3754	3757#						
DEREV1	001170	1707#	2040	3837*					
DEREX	016622	3908	3929	3931	3939	3947	3950	3952#	
DEREX1	016654	3953	3956	3958	3960#				
DERFL	000704	1590#	3751*	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#					

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

[illegible]

SEQ 0123

DSUP	014200	2711	2713	2715	2721	2732	3034	4171	4261	4407	4423	4428	4588	
DSO	014206	1997	3433#											
DSOA	014270	3440	3448#											
DSOB	014266	3444	3447#											
DSOC	014230	3434	3436	3439#										
DS1	014316	3432	3454#	3641										
DS2	014336	3458#	3460											
DS2A	014346	3455	3461#											
DS3	014352	2781	2975	3462#										
DS4	014362	3464#	3466											
DT	000536	1533#	3251	3256	3260	3454	4569							
DVN	000550	1541#	1978	3214	3219	3249	4406	4618*	4685					
DOFL	014554	3337*	3475	3478*	3507#									
EMADDR	000652	1577#	2136*	2232*	2302*	2354*	2491*	2506*	2685*	3023*	3033*	3044*	4238	4243
		4470												
EOPB1	000744	1606#	1911*	1927	1929*	2197*								
EOTCO	002650	1773#	2124*	2125										
EOTREC	000660	1580#	2108	2110*	2214*	2310	2312*	2313*	2314*	2398	2621	2642	2644*	2645*
		2646*	2677	2681*	2682	2719*	2720*	2791	2799					
ER	000524	1528#	4164	4175	4224	4265	4312							
ERCHK	017406	2318	2764	2967	4097#									
ERPT	020234	2366	2736	3037	4183	4185	4187	4195	4208	4212	4214#			
ERPTG	020270	4219	4223#											
ERPTG1	020336	4228	4233#											
ERPTT	020254	4218#	4221											
ERPTO	020352	4226	4232	4234	4236#	4329								
ERPT1	020420	4242	4244	4247#										
ERPT2	020456	4248	4257#											
ERPT3	020514	4258	4267#											
ERPT4	020556	4268	4277#											
ERPT5	020600	4278	4283#											
ERPT5A	020662	4288	4298#											
ERPT6	020724	4299	4308#											
ERPT7	020776	4309	4311	4315	4321#									
ERPX	021034	4323	4326	4328	4330#									
ERPX0	021000	4333	4332#											
ERPX1	021076	4336	4338#											
ERPX2	021130	4336	4341	4344#										
ERSAV	000722	1597#	2331	2379	2433	2824	2858	2890	4216*	4224*				
ERTFL	000732	1601#	2482*	2500	2503*									
ERO	017422	4099	4101#											
EROA	017472	4109	4113#											
EROB	017440	4103	4106#											
ER1	017500	4107	4115#											
ER10	020176	4204	4206#											
ER2	017504	2369	2513	2762	2959	2965	3040	3051	4112	4114	4116#			
ER2A	017550	4127#	4132	4142										
ER2A0	017520	4119#	4121											
ER2A1	017540	4123	4125#											
ER2B	017564	4126	4131#											
ER2C	017610	4137#	4146	4148										
ER2D	017624	4134	4141#											
ER2E	017652	4144	4147#											
ER3	017660	4128	4130	4138	4140	4149#								
ER3A	017722	4154	4160#											

SEQ 0124

ER3A1	017760	4166	4168#																
EP3B	017766	4156	4159	4163	4170#														
ER4	017772	4150	4161	4169	4171#														
ER4A	020036	4174	4181#																
ER4A1	020030	4177	4179#																
ER6	020042	4172	4180	4182#															
ER6A	020126	4189	4194#																
ERI	020160	4199	4201#																
EXCRC	015536	3681*	3691	3731#	4190	4192*	4198	4296	4342*										
EXLRC	015540	3693*	3732#	4191	4193*	4207	4306	4343*											
FC	000516	1525#	2303*	2356*	2492*	2507*	2693*	2910*	2938*	3027*	3045*	4127	4137	4281					
		4446*																	
FCER	021142	4129*	4139*	4277	4349#														
FCSAV	000632	1566#	2086	3332*															
FMCNT	000556	1544#	2086*	3303	3693	3697	3751	3928	3938	3948	3995*	3323*	3324	3326					
		3331*	3332	3633	3654	3682	3752	3774	3778	3786	3928	3932	4097	4135					
		4434	4616*	4653*	4755														
FREX	021214	4366	4371	4374#															
FRPRT	021156	2892	3876	4240	4365#	4419	4472												
FRO	021206	4368	4372#																
HDRFL	000664	1582#	3827*	3871	4759*														
HERE	005126	2185	2191#																
HRDS	022162	4531	4565#																
HRDSO	022222	4572#	4576	4597															
HRDS1	022230	4571	4574#																
HRDS2	022276	4580	4582	4584	4587#	4595													
HRDS3	022320	4589	4592#																
HRDS4	022332	4593	4596#																
INTRF	000566	1548#	3356	3356	4447	4621*													
LRCER	021144	4210*	4213*	4248	4350#														
LRCSV	021152	4191*	4213*	4353#															
MCRLF	027461	5018#	4240																
MR	000534	1532#	4201	4451*	4504*														
MSG1	024736	3874	4061																
MSG10	025017	3874	4061	5087#															
MSG100	026652	211023	2110																

CZTUA10 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	025067	3211	5099#	
MSG16	025120	4372	5105#	
MSG16A	025700	2127	5346#	
MSG17	025123	4369	5107#	
MSG20	025126	3887	5071#	
MSG21	025141	2121	5109#	
MSG22	025166	2722	5113#	
MSG23	025213	4249	5118#	
MSG23A	025221	4283	5123#	
MSG23B	025226	4269	5126#	
MSG23C	025236	4279	5128#	
MSG23D	025240	4253	5130#	
MSG23E	025251	4263	5132#	
MSG23F	025260	5138#	5134#	
MSG23G	025263	4316	5140#	
MSG23H	025272	5142#		
MSG23I	025277	5144#		
MSG23J	025304	5146#		
MSG23K	025311	5148#		
MSG23L	025316	5150#		
MSG23M	025336	4473	5152#	
MSG23N	025343	4420	5156#	
MSG23O	025363	4048	5160#	
MSG23P	025374	4063	5163#	
MSG23Q	025405	2545	5260	5166#
MSG23R	025430	3895	5073#	
MSG23S	025407	3175	5167#	
MSG23T	025461	3172	5175#	
MSG23U	025523	3229	5182#	
MSG23V	025544	3266	5186#	
MSG23W	025560	3278	5189#	
MSG23X	025573	3311	5192#	
MSG23Y	025614	3321	5196#	
MSG23Z	025640	3333	5201#	
MSG24	025630	3361	5205#	
MSG240	025703	3377	5207#	
MSG241	025731	3388	5210#	
MSG242	025743	3390	5211#	
MSG243	025763	4326	5216#	5220#
MSG244	025767	4326	5222#	
MSG245	0258016	3227	5227#	
MSG246	0258030	3227	5227#	
MSG247	0258035	3227	5227#	
MSG248	0258042	2161	2225	5234#
MSG249	0258074	1985	4075	5240#
MSG25	025862	2302	4214	5077#
MSG250	025810	3338	5242#	
MSG251	025835	3338	5244#	
MSG252	025872	3338	5246#	
MSG253	025872	3338	5247#	5254#
MSG254	025872	3338	5247#	
MSG255	025872	3338	5247#	
MSG256	025872	3338	5247#	

SEQ 0126

MSG57	026214	3253	5262#																	
MSG58	026235	4289	5263#																	
MSG59	026243	4300	5267#																	
MSG6	024767	2136	2232	2685	4417	5079#														
MSG60	026231	2138	4692	5269#																
MSG61	026235	4698	5270#																	
MSG62	026261	4705	5271#																	
MSG64	024265	2343	5393	2834	5273#															
MSG65	026307	2012	5427	2446	2846	2869	5278#													
MSG66	026320	2491	5281#																	
MSG67	026333	2506	5284#																	
MSG68	026343	2042	5287#																	
MSG69	026354	3343	5289#																	
MSG7	024774	4753	5081#																	
MSG70	026372	2234	5293#																	
MSG71	026426	3223	5298#																	
MSG72	026447	2022	5301#																	
MSG73	026460	2017	5303#																	
MSG74	026471	3179	5305#																	
MSG75	026514	3188	5309#																	
MSG76	026527	2057	5311#																	
MSG77	026540	2037	5313#																	
MSG78	026551	2335	2383	2437	5315#															
MSG79	026612	2828	2862	5321#																
MSG8	025001	4715	5083#																	
MSG9	025011	3262	5085#																	
MTC1	000672	1585#	2305*	2355*	2508*	2704*	2706*	2913*	2915*	2935*	2937*	3028*	3046*	4106						
		4125	4131	4186	4338	4365	4367	4412	4430	4449	4452	4457*	4459*	4735						
		1505	3198	4503#																
MTINT	021714	4478	4504#																	
MTINTA	021716	1461#																		
NOP =	C^0240	2337	2385	2439	2830	2864	2890#													
NRTP	011260	1576#	4518	4627	4640															
NRZOF	000650	4879#	5056																	
OCTP	023732	4831#	5006																	
OCTPE	023740	4880	48																	

SEQ 0127

[illegible]

HACY11 30(1046) 16-JUN-82 11:10 L 10
CROSS REFERENCE TABLE -- USER SYMBOLS PAGE 129

L. 10

PAGE 129
USED - 00000

USER SYMBOLS

RDR13	0111074	2852	2855#
RDR14	0111100	2854	2856#
RDR15	0111102	2841	2857#
RDR15A	0111146	2860	2867#
RDR15B	0111156	2866	2869#
RDR16	0112006	2868	2874
RDR17	011252	2884	2887#
RDSW	024406	4992#	5017*
RDX	010704	2809#	
RDO	010030	2688	2690
RD1	010076	2700	2703#
RD1A	010112	2696	2706#
RD1B	010120	2705	2707#
RD1D	010126	2708#	
RD1O	010624	2790	2792
RD11	010634	2795	2797#
RD2	010132	2707	2709#
RD2A	010152	2713#	
RD2B	010162	2712	2715#
RD3	010212	2710	2716
RD4	010242	2722	2728#
RD4A	010340	2733	2744#
RD4AO	010372	2746	2751#
RD4AI	010414	2754	2757#
RD4A2	010436	2748	2750
RD4B	010444	2731	2764#
RD4C	010450	2763	2765#
RD4D	010500	2769	2772#
RD4E	010504	2771	2773#
RD5	010514	2729	2766
RD6	010536	2741	2743
RD7	010562	2783	2785#
RD7A	010612	2788	2791#
READ	007732	2608	2620
REGS	000544	1539#	3181
REOT	004424	2101#	2404
REOTC	005156	1944	1946*
		3302*	3308
REOTX	005054	2173	2177#
REOTXX	005140	2181	2195#
REOT1A	004472	2107	2110#
REOT1B	004520	2113	2116#
REOT1C	004540	2117	2121#
REOT1D	004544	2122#	
REOT1E	004564	2120	2127#
REOT1F	004534	2115	2119#
REOT2	004606	2132#	2135
REOT2A	004644	2133	2141#
REOT3	004672	2142	2147#
REOT4	004722	2153#	2169
REOT5	004736	2155#	2156
REOT6	005002	2158	2165#
REOT7	005032	2170#	
RETRY	000602	1554#	2878
RETURN	021710	4489	4491
RFHARD	002730	1807#	2055
		2877#	
		2693#	2796 2807
		2794#	
		2718	2721#
		2760	2762#
		2775#	
		2776	2778 2780#
		2633	2641 2676#
		3183	3200
		2499	2523
		2141*	2623 2647
		3309*	3310* 2148 2149* 2150* 2199# 2215* 2216 2217* 2218* 3166*
			4565* 4598* 4600*
		4231	
		4493	4498#
		2885*	

SEQ 0129

RFSOFT	002670	1784#	2030	2853*
RPCNT	000700	1588#	2346*	2387*
RRHARD	002750	1818#	2055	2887*
RRSOFT	002710	1795#	2050	2655*
RSEQ	007414	2002	2598#	
RSEX	007554	2610	2621#	2635
RSF	007470	2606	2609#	
RSFR	007570	2604	2626#	
RSFRX	007730	2643	2648#	
RSFR0	007622	2630	2632#	
RSFR1	007634	2628	2634#	
RSFR2	007666	2637	2640#	
RSFO	007526	2612	2616#	
RSF1	007542	2615	2617	2619#
RSR	007446	2602	2605#	
RSTAL	000572	1550#	2785	3374
RTCNT	000702	1589#	2345*	2390*
		4229	2345*	2429
				2448
				2458
				2477*
				2478
				2836*
				2848
				2871
				2877*
				2878
RTRN	000662	1581#	2306*	2358*
RTYFL	000712	1593#	2321	2348*
RTY1	001050	1652#	1905	2015
RTY2	001052	1653#		2340*
RTY3	001054	1654#		2389*
RTY4	001056	1655#		2481*
RTY5	001060	1656#		
RTY6	001062	1657#		
RTY7	001064	1658#		
RTY8	001066	1659#		
RWND	005160	1998	2209#	
RWDA	005172	2210	2212#	4614
RWDX	005472	2243	2259#	4651
RWDO	005230	2219#	2239	
RWND1	005302	2228#	2231	
RWND1A	005340	2223	2229	2237#
RWND2	005356	2221	2240#	
RWND3	005362	2241#	2252	
RWND4	005412	2246#	2247	
RWND5	005432	2250#	2258	
RWND6	005450	2249	2253#	
SCVFL	014150	1516*	3177	3370
SERFL	000706	1591#	2324	3399*
		2330	2328*	3373
		2330	2332*	3401#
			2332*	2418
			2332*	2488*
			2332*	2495
			2332*	2505*
			2332*	2514
			2332*	2765
			2332*	2780*
			2332*	2837*
			2332*	2840
SN	000540	1534#	3264	
SNPG	024356	4966	4969	4975
SNPT	024300	3265	4960#	4977
SPFLG	000570	1549#	2180	4979#
STAL	000666	1583#	1962*	
		2917*	2921*	3363
		2001	2004	3365
STALL	012304	1493	1517	2072*
START	003026	1500	1897#	2134*
STARTA	003160	1890	1898#	2230*
STARTB	003164	1496	1893#	2319*
STARTC	003152	1894	1913#	2489*
STARTD	003250	1912#	2198	2613*
STARTE	003242	1907#	1909	2638*
STARTF	003222			2785*
				2908*
				3021*
				3021*
				3048*
				3052*
				3075*
				4456*
				4464*
				2924
				3022
				3053
				3075#
				3076

SEQ 0130

[illegible]

SEO 0131

[illegible]

SE7 0132

[illegible]

MACY11 30(1046) 16-JUN-82 11:10 PAGE 134
CROSS REFERENCE TABLE -- USER SYMBOLS

D 11

PAGE 134

SER SYMBOLS

[illegible]

E 11
PAGE

135

1487# 1492# 1495# 1498# 1504# 1510#
1728# 1730# 1732# 1734# 1736# 1738#
1759# 1761# 1763# 1765# 1767# 1769#

CZTUA10 TM02-TU16/TE16 RELIAB MACY11 30(1046) 16-JUN-82 11:10 F 11
CZTUA1.P11 16-JUN-82 10:46 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: 4 9 1 SECONDS

RUN-TIME RATIO: 59/15=3.7

CORE USED: 15K (30 PAGES)