

TM02/TU16

RELIABILITY
CZTUAH0

AH-9448H-MC

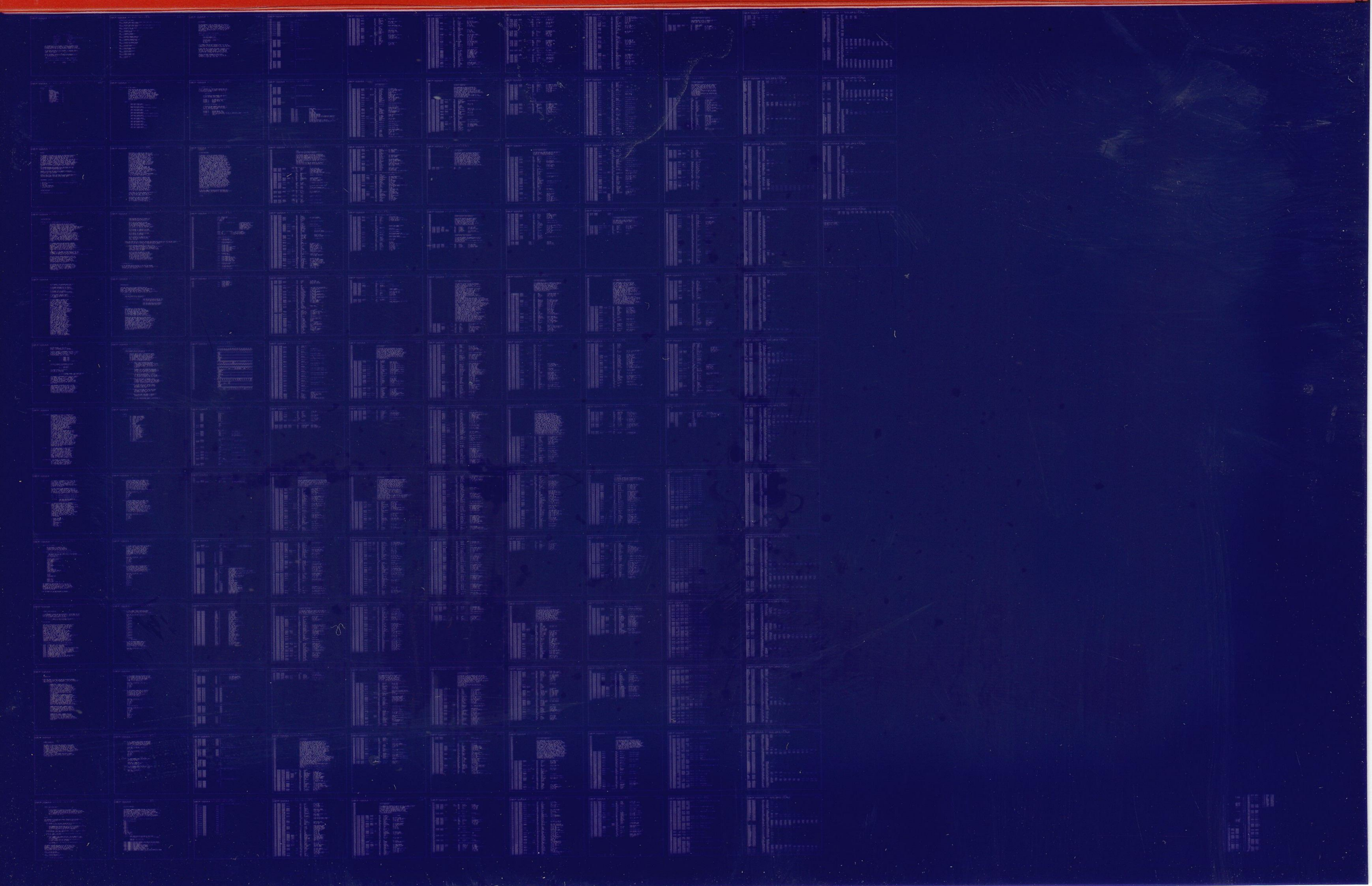
COPYRIGHT © 74-78

FICHE 1 OF 1

JUL 1978

digital

MADE IN USA



.REM 2

IDENTIFICATION

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

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

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

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

COPYRIGHT (C) 1974,1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

TABLE OF CONTENTS

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

1. ABSTRACT

THIS PROGRAM IS DESIGNED TO BE USED BY AN EXPERIENCED ENGINEER /TECHNICIAN FOR EVALUATION AND DEBUGGING OF MAG TAPE DRIVES. THE PROGRAM IS CAPABLE OF EXERCISING ANY TAPE DRIVE THAT CAN BE OPERATED ON A MASSBUS THROUGH THE TM02 MAG TAPE CONTROLLER. ANY TYPE OF TAPE DRIVE; NRZI, PE, 7 OR 9 TRACK MAY BE USED. ANY NUMBER OF DRIVES, SINGLE OR MULTIDRIVE SYSTEMS, UP TO EIGHT (8), MAY BE TESTED BY A SINGLE EXECUTION OF THE PROGRAM. THIS FLEXIBILITY IS POSSIBLE BECAUSE THE PROGRAM HAS NO FIXED PARAMETERS OR TESTING SEQUENCE. THE ENTIRE TEST PLAN, INCLUDING PARAMETERS AND OPERATING SEQUENCE, IS DETERMINED BY THE OPERATOR THROUGH RESPONSES TO TELETYPE REQUESTS AND SETTING OF CONSOLE SWITCHES.

THE PROGRAM PROVIDES FOR TESTING OF ALL TAPE DRIVE FUNCTIONS SUCH AS WRITING, READING, REWINDING, TAPE POSITIONING, EOT - BOT SENSING AND ASSUMES A GOOD RH AND TM02.

HOWEVER; THE RH AND TM02 ARE TESTED SOMEWHAT INTRINSICALLY DURING THE TEST CYCLE IN ORDER TO PROVIDE FULL INFORMATION ABOUT ANY ERROR CONDITIONS DETECTED.

DURING A TEST CYCLE, CHECKS ARE MADE FOR STATUS ERRORS, DATA ERRORS, POSITION ERRORS, WORD COUNT AND CURRENT MEMORY ADDRESS ERRORS WHEREVER APPLICABLE AS DETECTED BY THE RH OR TM02.

2. REQUIREMENTS (HARDWARE)

- A. ANY PDP-11 PROCESSOR - WITH OR WITHOUT HARDWARE SWITCH REGISTER
- B. 8K OF CORE
- C. TELETYPE
- D. TM02 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 NRZ ONLY. SEE ALSO SECTION 11 FOR DETAILS.
- E. 300(8): THIS ADDRESS IS TO BE USED AS A RESTART ONLY AND WILL PERFORM JUST AS IN 200(8) EXCEPT THAT THE PARAMETER INPUT LIST IS SHORTENED. THE SHORT PARAMETER LIST CONSISTS OF DRIVE NUMBER, SLAVE NUMBER, DENSITY, PARITY, FORMAT, RECORD COUNT, CHARACTER COUNT, PATTERN, TAPE MARK, AND INTERCHANGE READ.

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

- REGISTER START: THE RESPONSE REQUIRED FOR THIS REQUEST
IS TO ENTER THE ADDRESS OF THE FIRST RH
REGISTER (CS1) AS A SIX DIGIT UNIBUS ADDRESS.
- VECTOR ADDRESS: THE RESPONSE FOR THIS REQUEST
IS TO ENTER THE INTERRUPT VECTOR ADDRESS
USED BY THE RH AS A THREE (3) DIGIT ADDRESS.
- DRIVE NUMBER: THE DRIVE NUMBER (MASSBUS ADDRESS
OF THE TMO2) IS ENTERED AS ONE (1)
OCTAL CHARACTER AND MUST BE WITHIN THE LIMITS
OF 0 THROUGH 7.
- SLAVE NUMBER: THE SLAVE NUMBER IS ENTERED AS ONE
(1) OCTAL CHARACTER AND MUST BE
WITHIN THE LIMITS OF 0 THROUGH 7.
WHEN THE SLAVE NUMBER HAS BEEN
ENTERED AND IS LEGAL, THE PROGRAM TESTS
FOR THE PRESENCE OF A SLAVE OF THAT
NUMBER. IF THE SLAVE IS AVAILABLE
A PRINTOUT OF 7 CHANNEL, IF APPLICABLE,
AND ITS SERIAL NUMBER (IN BCD)
WILL BE MADE TO ASSIST THE OPERATOR
IN SETTING OF DENSITY, PARITY, AND FORMAT.
A CHECK IS MADE FOR THE PROPER SETTING
OF THE DRIVE TYPE REGISTER; IF WRONG, A
MESSAGE IS PRINTED FOR INFORMATION ONLY.
IF THE SLAVE IS NOT AVAILABLE,
A MESSAGE STATING SO WILL BE
PRINTED AND A NEW SLAVE NUMBER
REQUEST WILL BE ISSUED. WHEN A
GOOD SLAVE NUMBER HAS BEEN ENTERED,
REQUESTS FOR OPERATING DENSITY
PARITY AND FORMAT ARE MADE FOR THAT
SLAVE AND SHOULD BE RESPONDED TO
ACCORDING TO THAT PARTICULAR SLAVE'S
NEEDS. AS MANY AS EIGHT (8) SLAVE
NUMBER REQUESTS MAY BE USED, HOW-
EVER, AT LEAST ONE MUST BE USED.
THE SLAVE NUMBERS AND THEIR RESPECTIVE
DENSITY, PARITY AND FORMAT MAY BE ENTERED
IN ANY ORDER. THE INFORMATION FOR
EACH SLAVE ENTERED IS LOADED INTO A
TABLE FOR REFERENCE IN TESTING.
IF LESS THAN EIGHT(8) SLAVES ARE
REQUIRED, THEN RESPONDING TO THE
SLAVE NUMBER REQUEST WITH A CARRIAGE
RETURN WILL TERMINATE THE SLAVE
ENTRIES AND CONTINUE TO THE NEXT
PARAMETER. IT SHOULD BE REMEMBERED
THAT AT LEAST ONE SLAVE NUMBER REQUEST

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

DENSITY:

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

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

PARITY:

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

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

FORMAT:

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

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

RECORD COUNT:

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

CHARACTER COUNT:

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

PATTERN NUMBER: THIS RESPONSE IS A TWO (2) CHARACTER OCTAL NUMBER WITHIN THE LIMITS OF 0 THRU 15(8). THE NUMBER ENTERED WILL CAUSE A SPECIFIC DATA PATTERN TO BE USED FOR ALL READING AND WRITING. THIS DATA PATTERN IS NOT CHANGED UNLESS RANDOM DATA IS REQUESTED BY SETTING CONSOLE SWITCH EIGHT (8) TO A ONE. 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) CHARATERS 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 RESETING 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. RESETING 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. RESETING SWITCH 6 HOLDS LAST VALUE
GENERATED.

7. DYNAMIC PARAMETERS:

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

8. CONSOLE SWITCH SETTINGS

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

CONTROL:

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

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

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

SW15: 1=STOP ON ERROR
(100000)0=CONTINUE ON ERROR

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

SW13: 1=DO NOT CHECK DATA ERRORS

(020000)0=CHECK DATA ERRORS

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

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

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

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

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

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

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

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

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

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

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

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

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

SWITCH EXPLANATION AND EXAMPLES:

SW0-3: THESE SWITCHES ARE USED TO CONTROL THE SEQUENCE OF MAG TAPE OPERATIONS PERFORMED ON EACH AVAILABLE UNIT. THE BLOCK OF DATA DESCRIBED THROUGH THE RESPONSES TO TELETYPE REQUESTS AT INITIAL START WILL BE EITHER WRITTEN OR READ FROM EACH AVAILABLE UNIT IN THE ORDER THAT THEY WERE ENTERED. THE SEQUENCE OF OPERATIONS IS CALLED A CYCLE, AND WILL BE PERFORMED CONTINUOUSLY UNTIL STOPPED BY THE OPERATOR. WHEN END OF TAPE IS REACHED, THE UNIT WILL BE REWOUND AND FLAGGED AS UNAVAILABLE FOR TEST UNTIL ALL UNITS HAVE REACH EOT, AT WHICH TIME TESTING IS RESUMED ON ALL AVAILABLE UNITS.

EXAMPLES. 0-3

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

- SW4: SWITCH FOUR (4), WHEN SET TO A ONE (1), WILL CAUSE ANY DATA RELATED ERROR TO BE RETRIED. THE WRITE RETRY SCHEME CONSISTS OF REWRITING THE RECORD IN THE SAME SPOT ON TAPE FOUR (4) TIMES. IF ALL FOUR (4) REPEATS ARE SUCCESSFUL, THE RECORD IS CONSIDERED AS RECOVERED, AND A TAPE WRITE ERROR IS LOGGED. IF ANY OF THE FOUR (4) REPEATS IS UNSUCCESSFUL, A SKIP ERASE IS DONE, A SUSPECTED BAD TAPE SPOT IS LOGGED AT THIS BLOCK AND RECORD NUMBER, AND A SECOND RETRY OF FOUR REPEATS IS DONE. IF AFTER FOUR (4) RETRIES, THE RECORD CANNOT BE RECOVERED A NOTIFICATION IS PRINTED, AND TESTING IS RESUMED ON THE NEXT RECORD. IF 20(8) BAD TAPE SPOTS ARE FOUND, THE SLAVE WILL BE REWOUND AND REMOVED FROM TESTING WITH AN APPROPRIATE MESSAGE PRINTED. THE READ RETRY SCHEME CONSISTS OF REREADING THE RECORD UP TO EIGHT TIMES. IF ALL EIGHT REREADS ARE BAD, IT IS A HARD ERROR. IF ANY REREAD IS SUCCESSFUL, THIS IS A SOFT ERROR. IF THE ORIGINAL ERROR IS OF THE NON-RETRYABLE TYPE (IE: ILF,RMR,ILR,NEF,CBUSPE), THE RETRY SCHEME IS NOT ENTERED AND A MESSAGE IS PRINTED.
- SW5: SWITCH FIVE (5) WHEN SET DURING A READ FORWARD OR REVERSE WILL CAUSE THE TAPE TO CONTINUOUSLY READ THE CURRENT RECORD BY SPACING EITHER FORWARD OR REVERSE AND REREADING THAT RECORD. THIS TAPE MOVEMENT IS CALLED YOZZLING. THERE IS A SOFTWARE DELAY EXECUTED BETWEEN E.CH SPACE/READ OF THE RECORD AND IT MAY BE VARIED BY TYPING CONTROL C ON THE TELETYPE DURING THE EXECUTION OF THE YOZZLE AND RESPONDING TO THE PRINTED REQUEST WITH A SIX (6) DIGIT VALUE. THE YOZZLE STALL IS PRESET TO A VALUE OF 3000 IN THE PROGRAM TO PREVENT EXCESSIVE TAPE WEAR, BUT MAY BE SET TO ANY VALUE THROUGH THE TELETYPE.
- SW6-8: THESE THREE (3) SWITCHES CONTROL THE RANDOMIZATION OF DATA AND BLOCK SIZE AND MAY BE SET AND RESET AT ANY TIME. THE ACTUAL CHANGE WILL TAKE PLACE BETWEEN BLOCK CYCLES.
- SW9: SWITCH NINE (9) WHEN SET WILL CAUSE ALL AVAILABLE TAPE UNITS TO BE REWOUND AT THE END OF THE CURRENT BLOCK CYCLE. TESTING WILL BE RESUMED AT A BLOCK COUNT OF ONE (1) WHEN ALL UNITS HAVE REACHED BOT.

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

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

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

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

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

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

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

9. ERROR PRINTOUTS

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

A. OPERATION ERRORS:

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

1. READ/WRITE STATUS ERRORS: THESE ARE DETECTED BY EITHER THE TMO2 ITSELF OR BY THE MASSBUS CONTROLLER. ALL STATUS ERRORS WILL BE REPORTED.

2. TAPE POSITION ERRORS: THESE ARE INDICATED BY AN INCORRECT SPACE OR REWIND OPERATION IN WHICH TAPE POSITION BECOMES UNRELIABLE.

B. DATA ERRORS:

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

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

C. CONDITION ERRORS: (CATASTROPHIC)

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

1. EOT: WHEN EOT (END OF TAPE) IS ENCOUNTERED DURING
EITHER A READ OR WRITE, THE CYCLE IS COMPLETED
ON THE SHORTENED BLOCK AFTER WHICH THE SLAVE
WILL BE REWOUND AND FLAGGED AS UNAVAILABLE
FOR TESTING UNTIL ALL SLAVES HAVE REACHED EOT AND
ARE REWOUND. WHEN THE LAST AVAILABLE SLAVE
HAS REACHED EOT AND BEEN REWOUND TO BOT,
TESTING WILL BE RESUMED ON ALL SLAVES.
2. ILLEGAL BOT: WHEN A SLAVE ENCOUNTERS BOT DURING
A READ, WRITE, OR SPACE OPERATION, AN ERROR
IS PRINTED AND THE PROGRAM HALTED. THIS IS
A CATASTROPHIC ERROR. TESTING MAY BE RESUMED
BY PRESSING CONTINUE; BUT A RESTART IS
SUGGESTED.
3. NO INTERRUPT RETURNED: EACH TAPE OPERATION SHOULD BE
TERMINATED BY THE SETTING OF AN INTERRUPT IN
THE CPU. IF NO INTERRUPT IS RETURNED WITHIN
THE APPROPRIATE TIME, AN ERROR IS PRINTED.
4. NO MEDIUM ON-LINE: BEFORE AN OPERATION IS ATTEMPTED,
THE TMO2 IS CHECKED FOR MOL. IF IT IS NOT
SET, AN ERROR IS PRINTED, AND THE PROGRAM STOPPED.
TESTING MAY BE RESUMED BY PRESSING CONTINUE.
5. NO BOT ON REWIND: AS EACH SLAVE IS REWOUND A CHECK
IS MADE TO ASSURE THAT PROPER POSITION AT BOT
IS ESTABLISHED. IF BOT IS NOT SET UPON COMPLETION OF
A REWIND, AN ERROR IS PRINTED AND THE PROGRAM
WILL HALT. PRESS CONTINUE TO RESUME TESTING.
6. POSITION ERROR: IF POSITION IS LOST DURING A RETRY,
A MESSAGE IS PRINTED, THE TAPE REWOUND,
AND REMOVED FROM TESTING UNTIL ALL ARE
RESTARTED AT BLOCK ONE.
7. BAD TAPE OVERFLOW: IF 20(8) BAD TAPE SPOTS ARE FOUND,
A MESSAGE IS PRINTED, THE TAPE REWOUND,
AND REMOVED FROM TESTING UNTIL ALL ARE
RESTARTED AT BLOCK ONE.
8. HARD READ ERROR: IF ANY HARD READ ERROR IS ENCOUNTERED
DURING A RETRY, A MESSAGE IS PRINTED
REGARDLESS OF THE SETTING OF SW10.
9. NON-RETRYABLE: IF ANY NON-RETRYABLE ERROR IS ENCOUNTERED, A
MESSAGE IS PRINTED REGARDLESS OF THE SETTING OF SW10.

D. EXAMPLES:

GLOSSARY:

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

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

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

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

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

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

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

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

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

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

CZTL
CZTL

15
15
15
15
15
15
15
15
15
15
15
15
15
15
15
15
15
15
15
15
15
15
15
15
15
16
16
16
16
16
16
16
16
16
16
16

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

[illegible]

STATISTICS PRINTOUT

```

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
WTRR: NUMBER OF WRITE ERRORS NOT ASSOCIATED WITH BAD TAPE
REFWD: NUMBER OF READ FORWARD STATUS ERRORS
REREV: NUMBER OF READ REVERSE STATUS ERRORS
  SOFT: NUMBER OF RECOVERED READ ERRORS
  HARD: NUMBER OF UNRECOVERED READ ERRORS
DEFWD: NUMBER OF FORWARD DATA ERRORS WITH NO ASSOCIATED STATUS ERROR
DEREV: NUMBER OF REVERSE DATA ERRORS WITH NO ASSOCIATED STATUS ERROR

```

CZTU
CZTU

[illegible]

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 THOUGH RESPONSES TO TELETYPE REQUESTS FOR PARAMETERS AND CONSOLE SWITCH SETTINGS FOR OPERATION. THE OPERATION SELECTED WILL BE EXECUTED WITH THE PARAMETERS ENTERED CONTINUOUSLY ON EACH AVAILABLE UNIT, ONE BLOCK AT A TIME, UNTIL STOPPED BY THE OPERATOR. THE OPERATION MAY BE CHANGED DYNAMICALLY BY CHANGING THE CONSOLE SWITCHES AT ANY TIME. THE PROGRAM WILL ATTEMPT TO PERFORM ANY OPERATION SET AND THEREFORE CAUTION SHOULD BE TAKEN TO ASSURE THAT THE UNIT IS CAPABLE OF PERFORMING AS REQUESTED. FOR INSTANCE, ONE SHOULD NOT ATTEMPT TO PERFORM READ OPERATIONS ON A TAPE WHICH HAS NOT BEEN WRITTEN AS THE DATA, IF ANY, IS UNPREDICTABLE. HOWEVER, IF A TAPE HAS BEEN WRITTEN WITH THIS PROGRAM, IT CAN BE READ AS OFTEN AS DESIRED WITHOUT BEING REWRITTEN. THIS IS A GOOD PROCEDURE TO USE FOR TESTING TAPE COMPATABILITY. SCOPING OF TAPE UNITS BECOMES SIMPLE; BY SETTING THE DESIRED OPERATION AND ITS PARAMETER, A UNIT MAY BE CONTINUOUSLY EXERCISED IN ANY MANNER DESIRED. BY USING THE VARIOUS ERROR CONTROL SWITCHES AND ENTERING THE NEEDED STALL, ANY FUNCTION CAN BE SCOPED RATHER EASILY. RELIABILITY TESTING CAN BE PERFORMED BY USE OF THE RANDOMIZATION CAPABILITY. PERHAPS A CYCLE OF RANDOM TESTING MIGHT BE SET UP AND ALLOWED TO RUN FOR SOME PERIOD OF TIME, THE STATISTICAL COLLECTION OF DROPS AND PICKS IS THEN SIGNIFICANT. INTERMITTANT PROBLEMS CAN BE FOUND BY SETTING THE DESIRED OPERATION IN MOTION AND DISALLOWING ERROR PRINTOUTS WHILE ALLOWING A HALT ON ERROR. THE ERROR THAT CAUSED THE HALT CAN BE PRINTED BY RESETTING CONSOLE SWITCH TEN AND PRESSING CONTINUE. IF SOME PARTICULAR DATA PATTERN SHOULD BE CAUSING DATA ERROR, USE OF THE YOZZLE SWITCH AND ITS ASSOCIATED STALL WILL ALLOW SCOPING OF THIS PARTICULAR RECORD.

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

TRY IT, YOU'LL LIKE IT.

13. LISTING

x

```

1347 .LIST BIN,LOC,SEQ
1348 .TITLE CZTUARO TM02-TU16/TE16 RELIAB
1349 ;ZZ - CZTUARO
1350 ;21 APRIL 76
1351 ;R. BARNES
1352
1353 ;REVISED (++) J.G.ADAMS MAY 1977
1354 ;++G
1355 ;++G
1356 ;++G
1357 ;++G
1358 ;++G
1359 ;++G
1360 ;++G
1361
1362 .MCALL .SACT11,.$EOP,$CHAIN ;++G ACT11 HOOKS
1363 .NLIST MC ;++G DO NOT LIST MACRO CALLS
1364 .LIST ME ;++G LIST MACRO EXPANSIONS
1365 .ENABLE ABS,AMA ;++G ENABLE ABS AND MODE '37'
1366
1367
1368 ;CONSOLE SWITCHES*****
1369
1370 ;SW15: 1=STOP ON ERROR
1371 ; 0=CONTINUE ON ERROR
1372
1373 ;SW14: 1=PRINT READ/WRITE STATS
1374 ; 0=DO NOT PRINT STATS
1375
1376 ;SW13: 1=DO NOT CHECK DATA
1377 ; 0=CHECK DATA
1378 ;SW12: 1=DO NOT CHECK WRITE ERRORS
1379 ; 0=CHECK WRITE ERRORS
1380 ;SW11: 1=DO NOT CHECK READ ERRORS
1381 ; 0=CHECK READ ERRORS
1382 ;SW10: 1=DO NOT PRINT ERRORS
1383 ; 0=PRINT ERRORS
1384
1385 ;SW9: 1=REWIND TAPE
1386 ; 0=DO NOT REWIND
1387
1388 ;SW8: 1=USE RANDOM DATA
1389 ; 0=USE FIXED DATA PATTERN
1390 ;SW7: 1=USE RANDOM CHARACTER COUNT
1391 ; 0=USE FIXED CHAR COUNT
1392 ;SW6: 1=USE RANDOM RECORD COUNT
1393 ; 0=USE FIXED RECORD COUNT
1394
1395 ;SW5: 1=YOZZLE ON CURRENT RECORD
1396 ; 0=DO NOT YOZZLE
1397
1398 ;SW4: 1=DO BOTH READ AND WRITE RETRIES
1399 ; 0=INHIBIT RETRIES
1400
1401 ;SW3: 1=DO NOT READ FORWARD

```

1402	:	0=READ FORWARD
1403	;SW2:	1=DO NOT READ REVERSE
1404	:	0=READ REVERSE
1405	;SW1:	1=READ FORWARD FIRST
1406	:	0=READ REVERSE FIRST
1407	;SW0:	1=DO NOT WRITE
1408	:	0=WRITE

[illegible]

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


1517						;SHORT CONVERSATION RESTART*****
1518						
1519		000300				.=300
1520	000300	005237	014150	INC	SCVFL	;SET SHORT CONVERSATION FLAG
1521	000304	000137	003026	JMP	START	;ENTER SHORT PARAMETER LIST
1522						
1523		000510				.=510
1524						;TU16/TE16 REGISTER EQUIVS*****
1525						
1526	000510	172440		C1:	172440	
1527	000512	172442		WC:	172442	
1528	000514	172444		BA:	172444	
1529	000516	172446		FC:	172446	
1530	000520	172450		CS:	172450	
1531	000522	172452		DS:	172452	
1532	000524	172454		ER:	172454	
1533	000526	172456		AS:	172456	
1534	000530	172460		CC:	172460	
1535	000532	172462		DB:	172462	
1536	000534	172464		MR:	172464	
1537	000536	172466		DT:	172466	
1538	000540	172470		SN:	172470	
1539	000542	172472		C2:	172472	
1540						
1541						;CONSTANTS*****
1542						
1543	000544	172440		REGS:	172440	;STARTING REGISTER ADDRESS (CS1)
1544	000546	000224		VECT:	224	;VECTOR ADDRESS (RH INTERRUPT)
1545	000550	000000		DVN:	0	;DRIVE NUMBER
1546	000552	000000		UDES:	0	;UNIT DESCRIPTION (PARITY,DENSITY,UNIT,FORMAT)
1547	000554	000100		RCNT:	100	;RECORD COUNTER
1548	000556	177600		FMCNT:	177600	;NUMBER OF CHAR (4 - 4000) OCTAL IN TWOS COMPLEMENT
1549	000560	000001		PATRN:	1	;DATA PATTERN SELECTOR (0 - 15) OCTAL
1550	000562	000002		RDCMD:	2	;READ COMMAND
1551	000564	000000		TMEX:	0	;TAPE MARK FLAG: 1=TM 0=NO TM
1552	000566	000000		INTRF:	0	;INTERCHANGE READ 1=YES 0=NO
1553	000570	000000		SPFLG:	0	;SINGLE PASS 1=YES 0=NO
1554	000572	000001		RSTAL:	1	;READ STALL
1555	000574	000001		WSTAL:	1	;WRITE STALL
1556	000576	000001		TSTAL:	1	;TURN AROUND STAL
1557	000600	002000		YSTAL:	2000	;YOZZLE STAL
1558	000602	000010		RETRY:	10	;READ RETRY NUMBER
1559	000604	177776		PSW:	177776	;PROCESSOR STATUS
1560	000606	177570		SWR:	177570	;CONSOLE SWITCHES
1561	000610	177560		TKS:	177560	;TTY READ STATUS REGISTER
1562	000612	177562		TKB:	177562	;TTY READ BUFFER
1563	000614	177564		TPS:	177564	;TTY PUNCH STATUS REGISTER
1564	000616	177566		TPB:	177566	;TTY PUNCH OUTPUT REGISTER
1565	000620	177550		PRS:	177550	;H/S READER STATUS REGISTER
1566	000622	177552		PRB:	177552	;H/S READER BUFFER
1567	000624	153624		RANBAS:	153624	;RANDOM NUMBER GENERATOR BASE
1568	000626	032561		RANSAB:	032561	;RANDOM NUMBER BUFFER
1569	000630	000000		RCSAB:	0	;RECORD COUNT SAVE
1570	000632	000000		FCSAB:	0	;FRAME COUNT SAVE

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

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

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

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

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

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

22
22
22
22
22
22
22
22
22
22

[illegible]

1719					;DROPS + PICKS PER CHANNEL PER UNIT*****
1720					
1721	001210	000000	BP00:	0	
1722		001230		.=.+16	
1723	001230	000000	BP10:	0	
1724		001250		.=.+16	
1725	001250	000000	BP20:	0	
1726		001270		.=.+16	
1727	001270	000000	BP30:	0	
1728		001310		.=.+16	
1729	001310	000000	BP40:	0	
1730		001330		.=.+16	
1731	001330	000000	BP50:	0	
1732		001350		.=.+16	
1733	001350	000000	BP60:	0	
1734		001370		.=.+16	
1735	001370	000000	BP70:	0	
1736		001410		.=.+16	
1737	001410	000000	BD00:	0	
1738		001430		.=.+16	
1739	001430	000000	BD10:	0	
1740		001450		.=.+16	
1741	001450	000000	BD20:	0	
1742		001470		.=.+16	
1743	001470	000000	BD30:	0	
1744		001510		.=.+16	
1745	001510	000000	BD40:	0	
1746		001530		.=.+16	
1747	001530	000000	BD50:	0	
1748		001550		.=.+16	
1749	001550	000000	BD60:	0	
1750		001570		.=.+16	
1751	001570	000000	BD70:	0	
1752		001610		.=.+16	
1753					
1754					

[illegible]

```

1755
1756                ;UNIT BAD TAPE COUNTER:16 PER SLAVE*****
1757
1758 001610 000000    BT00: 0
1759                .=. +102
1760 001714 000000    BT01: 0
1761                .=. +102
1762 002020 000000    BT02: 0
1763                .=. +102
1764 002124 000000    BT03: 0
1765                .=. +102
1766 002230 000000    BT04: 0
1767                .=. +102
1768 002334 000000    BT05: 0
1769                .=. +102
1770 002440 000000    BT06: 0
1771                .=. +102
1772 002544 000000    BT07: 0
1773                .=. +102
1774
1775                ;UNIT END OF TAPE COUNTERS 1 PER SLAVE*****
1776
1777 002650 000000    EOTC0: 0
1778 002652 000000        0
1779 002654 000000        0
1780 002656 000000        0
1781 002660 000000        0
1782 002662 000000        0
1783 002664 000000        0
1784 002666 000000        0
1785
1786                ;UNIT READ FORWARD SOFT ERROR*****
1787
1788 002670 000000    RFSOFT: 0
1789 002672 000000        0
1790 002674 000000        0
1791 002676 000000        0
1792 002700 000000        0
1793 002702 000000        0
1794 002704 000000        0
1795 002706 000000        0
1796
1797                ;UNIT READ REVERSE SOFT ERROR*****
1798
1799 002710 000000    RRSOFT: 0
1800 002712 000000        0
1801 002714 000000        0
1802 002716 000000        0
1803 002720 000000        0
1804 002722 000000        0
1805 002724 000000        0
1806 002726 000000        0
1807
  
```

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

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

```

      .EVEN
      ;*****
      ;PROGRAM START AND SEQUENCE FORMATTER:
      ;
      ;THIS ROUTINE IS USED TO PERFORM ALL HOUSEKEEPING,
      ;DECIDE WHICH TRANSPORT TO TEST AND ITS AVAILABILITY,
      ;LOAD THE WRITE BUFFER WITH THE SELECTED DATA PATTERN,
      ;GENERATE ANY RANDOM NUMBER AND THEN EXECUTE
      ;THE TEST CYCLE REQUESTED BY THE SWITCH SETTING.
      ;AT THE END OF THE TEST CYCLE THE NEXT UNIT IS SELECTED
      ;AND CHECKED FOR AVAILABILITY AND THE TEST CYCLE IS
      ;EXECUTED ON IT.
      ;THE READ WRITE STATS MAY BE PRINTED AT THE END OF
      ;EACH TEST CYCLE VIA CONSOLE SWITCH FOURTEEN (14).
      ;*****

      ;START 200 & 300 *****
START:  MOV    #500,SP      ;++G SET STACK PTR
      CLR     ASEQF        ;CLEAR AUTO SEQUENCE FLAG
      CLR     (PC)+        ;CLEAR CHAIN INDICATOR
CHNFLG: .WORD    0         ;CHAIN MODE INDICATOR
      ;1/0 = CHAIN/NOT CHAIN MODE
      ;BRANCH IF LOADED VIA ACT11 CHAIN MODE
      CMP     #SENDAD,@#42
      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$:   CMPB    #6,@#41     ;++G BRANCH IF NOT LOADED VIA TMDP
      BNE     STAUT
      MOV     #MSG120,R4
      TTOUTT
      BR      STAUT
3$:    INC     ASEQF        ;++G
      JMP     SUSWR        ;++G SET AUTO SEQUENCE FLAG
      ;CHECK AND SET UP HRD/SOFT SWITCH REG ++ C.W

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

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

      ;START 210 *****
1901:  STARTA: CLR     TINF   ;CLEAR TTY ENTRY FLAG
1902:  STARTB: MOV     #TOB,R0
1903:  MOV     #37,R1
1904:  STARTD: CLR     (R0)+  ;CLEAR FLAGS AND COUNTERS
```

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

```
1961 003526 001005      BNE      STAR1C      ;IF NOT: BR
1962 003530 005037 000654 CLR      BLCNTR      ;RESET BLOCK CNTR
1963 003534 005037 000674 CLR      UNP        ;RESET UNIT POINTER
1964 003540 000207      RTS      PC        ;RETURN TO AUTO SEQ
1965
1966 003542 005037 000674      STAR1C: CLR      UNP
1967 003546 005000      CLR      R0
1968 003550 016037 000746 000552 MOV     UN1(R0),UDES ;LOAD FIRST UNIT DESCRIPTION
1969 003556 105777 175024      TSTB     @SWR      ;++G BRANCH IF NOT RANDOM RECORD
1970 003562 100003      BPL      START2      ;++G SIZE REQUESTED.
1971 003564 001402      BEQ      START2      ;IF NOT: BR
1972 003566 004737 012314      JSR      PC,CCNTR    ;GO GENERATE RANDOM RECORD SIZE
1973 003572 032777 000400 175006 START2: BIT     #400,@SWR ;SEE IF RANDOM DATA
1974 003600 001402      BEQ      START3      ;IF NOT: BR
1975 003602 004737 015102      JSR      PC,DATR     ;GO GENERATE RANDOM DATA
1976 003606 032777 000100 174772 START3: BIT     #100,@SWR ;SEE IF RANDOM RECORD COUNT
1977 003614 001402      BEQ      START4      ;IF NOT: BR
1978 003616 004737 012354      JSR      PC,RCNTR    ;GO GENERATE RANDOM RECORD COUNT
1979 003622 005760 000746      START4: TST     UN1(R0) ;++G BRANCH IF NOT AT EOT
1980 003626 100002      BPL      STAR40      ;IF NOT: BR
1981 003630 000137 004360      JMP      START7      ;ELSE GO TO NEXT UNIT
1982 003634 013777 000550 174656 STAR40: MOV     DVN,@CS ;SET DRIVE NUMBER
1983 003642 013777 000552 174672 MOV     UDES,@C2 ;SET UNIT NUMBER
1984 003650 105777 174646      TSTB     @DS        ;++G BRANCH IF UNIT AVAIL
1985 003654 100412      BMI      STAR4A
1986 003656 005337 000666      DEC      STAL
1987 003662 001357      BNE      START4
1988 003664 004737 022730      JSR      PC,PAPRT    ;PRINT HEADER
1989 003670 012704 026066      MOV     #MSG49,R4
1990 003674 104000      TTOUTT    ;PRINT NOT AVAIL
1991 003676 104006      STOPP
1992 003700 000750      BR         ;STOP
1993 003702 013746 000552      STAR4A: MOV     UDES,-(SP) ;++G RETRY
1994 003706 042716 175400      BIC      #175400,(SP) ;GET UNIT DESCRIPTION
1995      ;CMP     #1700,(SP)+ ;++G CLEAR ALL BUT FORMAT BITS
1996 003712 032726 002000      BIT      #2000,(SP)+ ;++G BRANCH IF NRZ
1997 003716 001406      BEQ      1$          ;++G BRANCH IF NZR
1998 003720 032777 000040 174574 BIT      #40,@DS ;++G BRANCH IF SLAVE IN PE FORMAT
1999 003726 001002      BNE      1$          ;++G
2000 003730 000137 004360      JMP      START7      ;++G GO TO NEXT UNIT
2001 003734 004737 014200      1$: JSR      PC,DSUP    ;GO SET UP WRITE DATA
2002 003740 004737 005160      JSR      PC,RWND    ;REWIND
2003 003744 004737 005522      JSR      PC,WRITE   ;WRITE
2004 003750 013737 000576 000666 MOV     TSTAL,STAL ;SET TURN AROUND DELAY
2005 003756 004737 012304      JSR      PC,STALL ;DELAY
2006 003762 004737 007414      JSR      PC,RSEQ    ;GO TO READ SEQUENCER
2007 003766 013737 000576 000666 MOV     TSTAL,STAL ;SET TURN AROUND DELAY
2008 003774 004737 012304      JSR      PC,STALL ;DELAY
2009 004000 032777 040000 174600 BIT      #40000,@SWR ;SEE IF SHOULD PRINT STATISTICS
2010 004006 001541      BEQ      START5      ;IF NOT: BR
2011 004010 012700 000001      MOV     #1,R0      ;SET RECORD COUNTER TO 1
2012 004014 004737 022730      JSR      PC,PAPRT    ;PRINT CYCLE NUMBER
2013 004020 004737 004030      JSR      PC,STP     ;GO PRINT STATS
2014 004024 000137 004276      JMP      STPX
2015 004030 004737 017250      STP: JSR      PC,DPPRT   ;PRINT DROPS AND PICKS
2016 004034 012704 026301      MOV     #MSG65,R4
```

```
2017 004040 104000 TTOUTT ;PRINT RETRY TOTAL
2018 004042 013704 000674 MOV UNP,R4
2019 004046 016403 001050 MOV RTY1(R4),R3
2020 004052 104002 OCTPP ;PRINT RETRIES
2021 004054 012704 026452 MOV #MSG73,R4
2022 004060 104000 TTOUTT ;PRINT WRITE ERROR TAG
2023 004062 013704 000674 MOV UNP,R4
2024 004066 016403 001070 MOV WTER1(R4),R3
2025 004072 104002 OCTPP ;PRINT WRITE ERRORS
2026 004074 012704 026441 MOV #MSG72,R4
2027 004100 104000 TTOUTT ;PRINT READ FORWARD ERROR TAG
2028 004102 013704 000674 MOV UNP,R4
2029 004106 016403 001110 MOV RDER1(R4),R3
2030 004112 104002 OCTPP ;PRINT READ FORWARD ERRORS
2031 004114 012704 027247 MOV #MSG113,R4
2032 004120 104000 TTOUTT ;PRINT SOFT TAG
2033 004122 013704 000674 MOV UNP,R4
2034 004126 016403 002670 MOV RFSOFT(R4),R3
2035 004132 104002 OCTPP ;PRINT FORWARD SOFT ERRORS
2036 004134 012704 027260 MOV #MSG114,R4
2037 004140 104000 TTOUTT ;PRINT HARD TAG
2038 004142 013704 000674 MOV UNP,R4
2039 004146 016403 002730 MOV RFHARD(R4),R3
2040 004152 104002 OCTPP ;PRINT HARD FORWARE ERRORS
2041 004154 012704 026532 MOV #MSG77,R4
2042 004160 104000 TTOUTT ;PRINT DATA ERROR FORWARD TAG
2043 004162 013704 000674 MOV UNP,R4
2044 004166 016403 001130 MOV DATER1(R4),R3
2045 004172 104002 OCTPP ;PRINT DATA ERROR FORWARD NUMBER
2046 004174 012704 026335 MOV #MSG68,R4
2047 004200 104000 TTOUTT ;PRINT READ ERROR REVERSE TAG
2048 004202 013704 000674 MOV UNP,R4
2049 004206 016403 001150 MOV RDERR1(R4),R3
2050 004212 104002 OCTPP ;PRINT REVESE ERROR NUMBER
2051 004214 012704 027247 MOV #MSG113,R4
2052 004220 104000 TTOUTT ;PRINT SOFT TAG
2053 004222 013704 000674 MOV UNP,R4
2054 004226 016403 002710 MOV RRSOFT(R4),R3
2055 004232 104002 OCTPP ;PRINT REVERSE SOFT ERROR
2056 004234 012704 027260 MOV #MSG114,R4
2057 004240 104000 TTOUTT ;PRINT HARD TAG
2058 004242 013704 000674 MOV UNP,R4
2059 004246 016403 002750 MOV RRHARD(R4),R3
2060 004252 104002 OCTPP
2061 004254 012704 026521 MOV #MSG76,R4
2062 004260 104000 TTOUTT ;PRINT DATA ERROR REVERSE TAG
2063 004262 013704 000674 MOV UNP,R4
2064 004266 016403 001170 MOV DEREV1(R4),R3
2065 004272 104002 OCTPP ;PRINT DATA REVERSE ERROR NUMBER
2066 004274 000207 RTS ;RETURN
2067 004276 005237 000726 STPX: INC PC ;SET STAT ONLY PRINT
2068 004302 004737 007324 JSR PC,BTPRT ;PRINT BAD TAPE STATS
2069 004306 005037 000726 CLR BTSTF ;CLEAR FLAG
2070 004312 017700 174270 START5: MOV @SWR,R0 ;LOAD SWR
2071 004316 042700 177762 BIC #177762,R0 ;MASK READ/WRITE SWITCHES
2072 004322 022700 000015 CMP #15,R0 ;SEE IF HAVE READ OR WRITE
```

```

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

```

```
2093 ;*****
2094 ;REWIND FROM EOT:
2095 ;
2096 ;WHEN ANY TRANSPORT BEING TESTED REACHES END OF TAPE
2097 ;DURING A READ OR WRITE OPERATION, IT WILL BE REWOUND
2098 ;AND FLAGGED AS UNAVAILABLE UNTIL ALL AVAILABLE UNITS
2099 ;HAVE REACHED EOT AT WHICH TIME ALL TESTING WILL BE RESUMED
2100 ;AT A BLOCK COUNT OF ONE (1). A MESSAGE WILL BE
2101 ;PRINTED ON THE SUPERVISORS CONSOLE AS EACH UNIT REACHES
2102 ;EOT AND IS REWOUND.
2103 ;*****
2104
2105 004424 013777 000552 174110 REOT: MOV UDES,AC2 ;LOAD COMMAND REGISTER
2106 004432 012777 000011 174050 MOV #11,AC1 ;DRIVE CLEAR
2107 004440 105777 174056 1$: TSTB ADS ;++G WAIT FOR DRIVE READY
2108 004444 100375 BPL 1$ ;AWAIT DRY
2109 004446 012777 000007 174034 MOV #7,AC1 ;START REWIND
2110 004454 005737 000724 TST BTFLG ;SEE IF BAD TAPE OVERFLOW REWIND
2111 004460 001004 BNE REOT1A ;IF SO: BR
2112 004462 013700 000660 MOV EOTREC,R0
2113 004466 042700 100000 BIC #100000,R0 ;SET RECORD NUMBER OF EOT
2114 004472 005037 000660 REOT1A: CLR EOTREC ;++G CLEAR EOT IND & REC CTR
2115 004476 004737 022730 JSR PC,PAPRT ;PRINT HEADER
2116 004502 022737 000002 000724 CMP #2,BTFLG ;SEE IF POSITION ERROR
2117 004510 001003 BNE REOT1B ;IF NOT: BR
2118 004512 012704 027140 MOV #MSG109,R4 ;SET POSITION ERROR MSG
2119 004516 000406 BR REOT1F
2120 004520 022737 000001 000724 REOT1B: CMP #1,BTFLG ;SEE IF BAD TAPE OVERFLOW
2121 004526 001004 BNE REOT1C ;IF NOT: BR
2122 004530 012704 026751 MOV #MSG106,R4 ;SET BAD TAPE OVERFLOW MSG
2123 004534 104000 REOT1F: TTOUTT ;PRINT REWIND REASON
2124 004536 000412 BR REOT1E
2125 004540 012704 025120 REOT1C: MOV #MSG20,R4 ;SET EOT MSG
2126 004544 104000 REOT1D: TTOUTT ;PRINT MSG
2127 004546 013704 000674 MOV UNP,R4
2128 004552 005264 002650 INC EOTC0(R4) ;BUMP CNTR
2129 004556 016403 002650 MOV EOTC0(R4),R3
2130 004562 104002 OCTPP ;PRINT EOT CNTR
2131 004564 012704 026776 REOT1E: MOV #MSG16A,R4
2132 004570 104000 TTOUTT ;PRINT RESTART MSG
2133 004572 005037 000724 CLR BTFLG ;CLEAR BAD TAPE FLAG
2134 004576 004737 004030 JSR PC,STP ;PRINT STATS
2135 004602 004737 007324 JSR PC,BTPRT ;PRINT BAD TAPE STATS
2136 004606 105777 173710 REOT2: TSTB ADS ;++G BRANCH IF UNIT IS READY
2137 004612 100414 BMI REOT2A
2138 004614 005337 000666 DEC STAL
2139 004620 001372 BNE REOT2 ;WAIT DRY
2140 004622 012737 024761 000652 MOV #MSG6,EMADDR
2141 004630 004737 022730 JSR PC,PAPRT ;PRINT HEADER
2142 004634 012704 026243 MOV #MSG60,R4
2143 004640 104000 TTOUTT ;PRINT NO DRIVE READY
2144 004642 104006 STOPP
2145 004644 105337 005156 REOT2A: DECB REOTC ;SEE IF LAST UNIT TO REACH EOT
2146 004650 001410 BEQ REOT3 ;IF SO: BR
2147 004652 013700 000674 MOV UNP,R0
2148 004656 052760 100000 000746 BIS #100000,UN1(R0) ;SET EOT FLAG
```

```
2149 004664 005726          TST      (SP)+      ;RESET STACK POINTER
2150 004666 000137 004360    JMP      START7    ;GO TO NEXT UNIT
2151 004672 000337 005156    REOT3: SWAB     REOTC      ;
2152 004676 013700 005156    MOV      REOTC,R0
2153 004702 000337 005156    SWAB     REOTC
2154 004706 110037 005156    MOV      RO,REOTC    ;RESTORE EOT UNIT COUNTER
2155 004712 005037 000674    CLR      UNP
2156 004716 013700 000674    MOV      UNP,R0      ;POINT TO FIRST UNIT
2157 004722 016037 000746 000552 REOT4: MOV      UN1(R0),UDES ;LOAD UNIT DESCRIPTION
2158 004730 013777 000552 173604 MOV      UDES,@C2     ;LOAD COMMAND REGISTER
2159 004736 032777 020000 173556 REOT5: BIT      #20000,@DS
2160 004744 001374          BNE      REOT5      ;AWAIT PIP RESET
2161 004746 032777 000002 173546 BIT      #2,@DS      ;SEE IF HAVE BOT
2162 004754 001012          BNE      REOT6      ;IF SO: BR
2163 004756 012700 000001    MOV      #1,R0
2164 004762 004737 022730    JSR      PC,PAPRT    ;PRINT HEADER
2165 004766 012704 026034    MOV      #MSG48,R4
2166 004772 104000          TTOUTT          ;PRINT BOT ERROR
2167 004774 104006          STOPP
2168 004776 013700 000674    MOV      UNP,R0
2169 005002 042760 100000 000746 REOT6: BIC      #100000,UN1(R0) ;CLEAR EOT FLAG
2170 005010 062737 000002 000674 ADD      #2,UNP
2171 005016 013700 000674    MOV      UNP,R0      ;POINT TO NEXT UNIT
2172 005022 022760 177777 000746 CMP      #-1,UN1(R0) ;++G BRANCH IF NOT LAST UNIT
2173 005030 001334          BNE      REOT4      ;++
2174 005032 005037 000674    REOT7: CLR      UNP      ;CLEAR UNIT POINTER
2175 005036 005037 000634    CLR      TINF     ;CLEAR TTY INPUT FLAG
2176 005042 005737 000734    TST      ASEQF     ;SEE IF AUTO SEQ
2177 005046 001402          BEQ      REOTX      ;IF NOT: BR
2178 005050 005726          TST      (SP)+      ;RESET STACK POINTER
2179 005052 000412          BR       TEND       ;GO DO END OF PASS ++ C.W
2180
2181 005054 004737 004372    REOTX: JSR      PC,RANSET    ;GO RESET RANDOM BASE
2182 005060 012737 177777 014406 MOV      #-1,PATS    ;PRESET PATTERN
2183 005066 005037 015150    CLR      RDFL     ;CLEAR RANDOM FLAG
2184 005072 005737 000570    TST      SPFLG     ;SEE IF SINGLE PASS
2185 005076 001420          BEQ      REOTXX     ;IF NOT: BR
2186 005100 012704 026644    TEND:  MOV      #MSG100,R4
2187 005104 104000          TTOUTT          ;PRINT END OF PASS
2188 005106 013700 000042    MOV      @#42,R0    ;GET ACT11 RETURN ADDRESS
2189 005112 001405          BEQ      HERE      ;BRANCH IF NOT ACT11
2190 005114 000005          RESET
2191 005116 004710    SENDAD: JSR      PC,(R0)
2192 005120 000240          NOP
2193 005122 000240          NOP
2194 005124 000240          NOP
2195 005126 005737 000734    HERE:  TST      ASEQF     ;AUTO MODE? ++ C.W
2196 005132 001401          BEQ      1$      ;BRANCH - IF NO ++ C.W
2197 005134 000207          RTS      PC        ;RETURN TO AUTO SEQ ++ C.W
2198 005136 104006          1$:  STOPP
2199 005140 012704 026644    REOTXX: MOV      #MSG100,R4    ;GET END OF PASS MESSAGE ++ C.W
2200 005144 104000          TTOUTT          ;PRINT MESSAGE ++ C.W
2201 005146 005237 000744    INC      EOPB1
2202 005152 000137 003242    JMP      STARTE    ;RESTART AT BLOCK NUMBER ONE
2203 005156 000000    REOTC:  0          ;EOT UNIT COUNTER
```

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

2260	005464	104000				TTOUTT		;PRINT NO BOT
2261	005466	104006				STOPP		
2262	005470	000760				BR	RWND5	;++G DO NEXT UNIT
2263	005472	013737	000714	000674	RWNDX:	MOV	UPS,UNP	;RESTORE UNIT POINTER
2264	005500	013700	000674			MOV	UNP,RO	
2265	005504	016037	000746	000552		MOV	UN1(RO),UDES	;RESET UNIT DESCRIPTION
2266	005512	013777	000552	173022		MOV	UDES,@C2	
2267	005520	000207				RTS	PC	;RETURN TO TEST
2268								

[illegible]

```

2269 ;*****
2270 ;WRITE ROUTINE:
2271 ;
2272 ;THIS ROUTINE IS USED TO WRITE ONTO TAPE THE BLOCK
2273 ;OF DATA DESCRIBED BY THE OPERATOR AND SET UP
2274 ;IN THE SEQUENCE FORMATTER. THE TAPE UNIT TO BE USED
2275 ;HAS BEEN ASSIGNED BY THE SEQUENCE FORMATTER AND
2276 ;ITS PARAMETERS SET IN A UNIT DESCRIPTION WORD.
2277 ;AS EACH RECORD OF THE BLOCK IS WRITTEN, IT IS CHECKED
2278 ;FOR STATUS ERRORS, WORD COUNT ZERO, AND CORRECT CURRENT
2279 ;MEMORY ADDRESS. IF THE WRITE OPERATION RESULTS IN
2280 ;ANY ERROR CONDITION, A WRITE RETRY OF THAT OPERATION
2281 ;MAY BE DONE BY SETTING SWITCH FOUR (4) TO A ONE (1).
2282 ;THE RETRY CONSISTS OF A BACKSPACE, ERASE FORWARD, AND
2283 ;REWRITE OF THE RECORD. (SEE WRITE RETRY SUBROUTINE)
2284 ;AFTER ALL DATA RECORDS IN THE BLOCK HAVE BEEN
2285 ;WRITTEN, THE WRITE ROUTINE WILL EXECUTE A WRITE
2286 ;TAPE MARK COMMAND IF THE TTY RESPONSE TM=1 WAS
2287 ;MADE AT INITIAL START. THE TM IS COUNTED AS TOTAL
2288 ;DATA RECORDS PLUS ONE (IE: IF 100 DATA RECORDS; TM=RECORD 101)
2289 ;IF THE WRITE OPERATION (DATA OR TM) CAUSES THE SELECTED SLAVE
2290 ;TO REACH END OF TAPE (EOT) AND THERE IS TO BE NO READING DONE,
2291 ;(SW2 AND SW3 SET TO A 1) THEN THE SLAVE IS REWOUND AND
2292 ;FLAGGED AS UNAVAILABLE FOR TESTING UNTIL ALL SLAVES HAVE
2293 ;REACHED EOT AND BEEN REWOUND AT WHICH TIME TESTING IS
2294 ;RESUMED ON ALL AVAILABLE SLAVES.
2295 ;WRITE RETRY MAY BE ALLOWED VIA CONSOLE SWITCH FOUR (4).
2296 ;ERROR CHECKING MAY BE DISALLOWED VIA CONSOLE SWITCH
2297 ;TWELVE (12).
2298 ;WRITING TO TAPE MAY BE DISALLOWED VIA CONSOLE SWITCH
2299 ;ZERO (0).
2300 ;*****
2301
2302 005522 032777 000001 173056 WRITE: BIT #1,@SWR ;SEE IF SHOULD WRITE
2303 005530 001402 BEQ WRTE
2304 005532 000137 006320 JMP WEX ;IF NOT: BR
2305 005536 013700 000554 WRTE: MOV RCNT,R0 ;R0=RECORD COUNT
2306 005542 012737 024754 000652 W0: MOV #MSG5,EMADDR ;SET ERROR MSG ADDRESS
2307 005550 013777 000556 172740 MOV FMCNT,@FC ;LOAD CHAR COUNT
2308 005556 012777 027456 172730 MOV #WDATA,@BA ;SET DATA ADDR
2309 005564 112737 000060 000672 MOVVB #60,MTCT ;SET WRITE OP COMMAND
2310 005572 012737 005604 000662 MOV #W1,RTRN ;SET RETURN ADDRESS
2311 005600 000137 021216 JMP TAPG ;GO EXECUTE COMMAND
2312 005604 032777 002000 172710 W1: BIT #2000,@DS ;SEE IF EOT
2313 005612 001414 BEQ W2 ;IF NOT AT EOT: BR
2314 005614 005737 000660 TST EOTREC ;++G BRANCH IF WRITTEN PAST EOT
2315 005620 100411 BMI W2 ;++G
2316 005622 010037 000660 MOV R0,EOTREC ;SAVE EOT RECORD NUMBER
2317 005626 052737 100000 000660 BIS #100000,EOTREC ;++G SET EOT FLAG
2318 005634 005337 000660 DEC EOTREC ;++G ADJUST RECORD COUNT
2319 005640 012700 000002 MOV #2,R0 ;++G SET R0 TO WRITE 1 MORE RECORD
2320 005644 032777 010000 172734 W2: BIT #10000,@SWR ;SEE IF SHOULD CHECK ERRORS
2321 005652 001002 BNE W3 ;IF NOT: BR
2322 005654 004737 017406 JSR PC,ERCHK ;GO CHECK ERRORS
2323 005660 013737 000574 000666 W3: MOV WSTAL,STAL ;SET DELAY
2324 005666 004737 012304 JSR PC,STALL ;DELAY

```

CZTU
CZTU[illegible]

2325	005672	005737	000712		TST	RTYFL	;SEE IF RETRY TIME
2326	005676	001401			BEQ	W3A	;IF NOT: BR
2327	005700	000207			RTS	PC	;ELSE RETURN
2328	005702	005737	000706	W3A:	TST	SERFL	;SEE IF WRITE ERROR
2329	005706	001450			BEQ	W5	;IF NOT: BR
2330	005710	013704	000674		MOV	UNP,R4	
2331	005714	005264	001070		INC	WTER1(R4)	;BUMP WRITE ERROR
2332	005720	005037	000706		CLR	SERFL	;CLEAR STATUS ERROR FLAG
2333	005724	032777	000020	172654	BIT	#20,@SWR	;SEE IF RETRY
2334	005732	001436			BEQ	W5	;IF NOT: BR
2335	005734	013703	000722		MOV	ERSAV,R3	
2336	005740	042703	102700		BIC	#102700,R3	;MASK UNRECOVERABLE ERROR
2337	005744	001410			BEQ	W4	;IF SO: BR
2338	005746	004737	022730		JSR	PC,PAPRT	;PRINT HEADER
2339	005752	012704	026543		MOV	#MSG78,R4	
2340	005756	104000			TTOUTT		;PRINT NON-RETRYABLE ERROR TAG
2341	005760	004737	011260		JSR	PC,NRTP	;PRINT ER FOR NON-RETRYABLE
2342	005764	000421			BR	W5	
2343	005766	013704	000674	W4:	MOV	UNP,R4	
2344	005772	005264	001050		INC	RTY1(R4)	;BUMP RETRY CNTR
2345	005776	032777	002000	172602	BIT	#2000,@SWR	;SEE IF PRINT ERRORS
2346	006004	001003			BNE	W4A	;IF NOT: BR
2347	006006	012704	026257		MOV	#MSG64,R4	
2348	006012	104000			TTOUTT		;PRINT ORIGINAL ERROR TAG
2349	006014	005037	000702	W4A:	CLR	RTCNT	;CLEAR RETRY NUMBER
2350	006020	005037	000700		CLR	RPCNT	;CLEAR REPEAT COUNTER
2351	006024	004737	006362		JSR	PC,WRTY	;GO RETRY WRITE ERROR
2352	006030	005037	000712	W5:	CLR	RTYFL	;CLEAR RETRY COUNTER
2353	006034	005300			DEC	RO	;SEE IF DONE ALL
2354	006036	001241			BNE	W0	;IF NOT: BR
2355	006040	005737	000564	W6:	TST	TMEX	;SEE IF TM
2356	006044	001525			BEQ	WEX	;IF NOT: BR
2357	006046	005237	000676		INC	TMFLG	;SET TM FLAG
2358	006052	012737	026164	000652	MOV	#MSG54,EMADDR	;POINT TO TM ERROR MSG
2359	006060	012737	000026	000672	MOV	#26,MTCT	;SET TM OP CODE
2360	006066	012777	000000	172422	MOV	#0,@FC	;LOAD FRAME COUNTER
2361	006074	012777	027456	172412	MOV	#WDATA,@BA	;LOAD BUS ADDRESS
2362	006102	012737	006114	000662	MOV	#WTMO,RTRN	;SAVE RETURN ADDRESS
2363	006110	000137	021216		JMP	TAPG	;WRITE TM
2364	006114	032777	010000	172464	WTMO:	BIT	#10000,@SWR
2365	006122	001076			BNE	WEX	;SEE IF SHOULD CHECK ERRORS
2366	006124	032777	000004	172370	BIT	#4,@DS	;SEE IF TM STATUS
2367	006132	001011			BNE	WTM1	;IF SO: BR
2368	006134	012737	027456	021132	MOV	#WDATA,CADER	;SET EXPT BUS ADDRESS
2369	006142	012737	000001	021140	MOV	#1,DRVR	;INDICATE ERROR
2370	006150	004737	020234		JSR	PC,ERPT	;PRINT TM ERROR
2371	006154	000404			BR	WTM2	
2372	006156	012703	027456		WTM1:	MOV	#WDATA,R3
2373	006162	004737	017504		JSR	PC,ER2	;GO CHECK FOR OTHER ERRORS
2374	006166	005737	000712		WTM2:	TST	RTYFL
2375	006172	001401			BEQ	WTM3	;SEE IF RETRY
2376	006174	000207			RTS	PC	;IF NOT: BR
2377	006176	005737	000706		WTM3:	TST	SERFL
2378	006202	001446			BEQ	WEX	;ELSE RETURN TO RETRY ROUTINE
2379	006204	013704	000674		MOV	UNP,R4	;SEE IF WRITE ERROR
2380	006210	005264	001070		INC	WTER1(R4)	;IF NOT: BR
							;BUMP WRITE ERROR

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

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

```

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

```

CLR      RTYFL      ;CLEAR RETRY FLAG
MOV      #1,BTFLG   ;SET BAD TAPE OVERFLOW FLAG
TST      (SP)+      ;++G ADJUST STACK
JMP      REOT       ;GO REWIND AND REMOVE FROM TESTING
MOV      BTPT,R1    ;SET TABLE POINTER
TST      (R1)+
CLR      R0
MOV      R0,R3
CLC
ROR      R3         ;R3=R3/2 FOR CORRECT NUMBER
OCTPP    ;PRINT ENTRY NUMBER
MOV      #MSG13,R4
TSTB     (R4)+      ;SKIP CR/LF
TTOUTT   ;PRINT BLOCK NUMBER TAG
MOV      (R1),R3
OCTPP    ;PRINT BLOCK NUMBER
MOV      #MSG14,R4
TTOUTT   ;PRINT RECORD NUMBER TAG
ADD      #40,R1     ;SET POINTER OFFSET FOR RECOED NUMBER
MOV      (R1)+,R3
OCTPP    ;PRINT RECORD NUMBER
SUB      #40,R1     ;RESET POINTER FOR BLOCK NUMBER
TST      (R0)+
CMP      R0,BTPT    ;SEE IF DONE
BEQ      BTOV2      ;IF SO: BR
MOV      #MSG28,R4
TTOUTT   ;DO CR/LF
BR      BTOV1       ;CONTINUE
TST      BTSTF      ;SEE IF STAT ONLY PRINT
BNE      BTOVX      ;IF SO: BR
MOV      #41,R3     ;SET SIZE OF TABLE
MOV      BTPT,R4    ;SET POINTER
CLR      (R4)+      ;CLEAR TABLE
DEC      R3         ;SEE IF DONE
BNE      BTOV3      ;IF NOT: BR
RTS      PC         ;RETURN

```

```

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

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

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

```
2586                                     :*****
2587                                     :READ SEQUENCER:
2588                                     :
2589                                     :THIS ROUTINE IS USED TO DETERMINE THE SEQUENCE
2590                                     :IN WHICH READ TAPE OPERATIONS ARE TO BE PERFORMED.
2591                                     :THIS IS NECESSARY WHEN THE UNIT BEING TESTED IS
2592                                     :CAPABLE OF READING DATA IN BOTH THE FORWARD AND
2593                                     :REVERSE DIRECTIONS.  CONSOLE SWITCHES ONE (1), TWO (2),
2594                                     :AND THREE (3) ARE USED TO DETERMINE THE READ SEQUENCE.
2595                                     :CONSOLE SWITCH ONE (1) DETERMINES WHETHER TO READ
2596                                     :THE BLOCK OF DATA FORWARD FIRST OR REVERSE FIRST.
2597                                     :SWITCH TWO (2) DISALLOWS READING IN THE REVERSE
2598                                     :DIRECTION AND SWITCH THREE (3) DISALLOWS READING IN
2599                                     :THE FORWARD DIRECTION.
2600                                     :*****
2601
2602 007414 012737 000002 000562 RSEQ: MOV    #2,RDCMD
2603 007422 017704 171160          MOV    @SWR,R4          ;READ SWITCHES
2604 007426 042704 177763          BIC    #177763,R4        ;MASK READ BITS
2605 007432 005704          TST     R4          ;SEE IF BOTH READS
2606 007434 001004          BNE     RSR          ;IF NOT: BR
2607 007436 032777 000002 171142  BIT    #2,@SWR          ;SEE IF READ REVERSE FIRST
2608 007444 001051          BNE     RSFR          ;IF NOT: BR
2609 007446 032777 000004 171132  RSFR: BIT    #4,@SWR          ;SEE IF SHOULD READ REVERSE
2610 007454 001005          BNE     RSF          ;IF NOT: BR
2611 007456 012737 010000 000562  MOV    #10000,RDCMD        ;LOAD READ REVERSE COMMAND
2612 007464 004737 007732          JSR     PC,READ          ;GO READ REVERSE
2613 007470 032777 000010 171110  RSF:  BIT    #10,@SWR         ;SEE IF SHOULD READ FORWARD
2614 007476 001026          BNE     RSEX          ;IF NOT: BR
2615 007500 032737 010000 000562  BIT    #10000,RDCMD        ;SEE IF HAVE READ REVERSE
2616 007506 001407          BEQ     RSFO          ;IF NOT: BR
2617 007510 013737 000576 000666  MOV    TSTAL,STAL
2618 007516 004737 012304          JSR     PC,STALL          ;DO READ STALL
2619 007522 000137 007542          JMP     RSF1
2620 007526 032777 000001 171052  RSFO: BIT    #1,@SWR          ;SEE IF WRITE
2621 007534 001002          BNE     RSF1          ;IF NOT: BR
2622 007536 004737 012056          JSR     PC,BKSP          ;GO BACKSPACE
2623 007542 012737 000002 000562  RSF1: MOV    #2,RDCMD        ;LOAD READ FORWARD COMMAND
2624 007550 004737 007732          JSR     PC,READ          ;GO READ
2625 007554 005737 000660          RSEX: TST     EOTREC        ;++G BRANCH IF NOT AT EOT
2626 007560 100002          BPL     1$          ;++G
2627 007562 000137 004424          JMP     REOT          ;++G ELSE GO REWIND
2628 007566 000207          1$:    RTS     PC          ;++G EXIT
2629
2630 007570 012737 010000 000562  RSFR: MOV    #10000,RDCMD
2631 007576 032777 000010 171002  BIT    #10,@SWR          ;SEE IF SHOULD READ FORWARD
2632 007604 001013          BNE     RSFR1          ;IF NOT: BR
2633 007606 032777 000001 170772  BIT    #1,@SWR          ;SEE IF WRITE
2634 007614 001002          BNE     RSFR0          ;IF NOT: BR
2635 007616 004737 012056          JSR     PC,BKSP          ;GO BACKSPACE TO START
2636 007622 012737 000002 000562  RSFR0: MOV    #2,RDCMD        ;LOAD READ FORWARD COMMAND
2637 007630 004737 007732          JSR     PC,READ          ;GO READ FORWARD
2638 007634 032777 000004 170744  RSFR1: BIT    #4,@SWR          ;SEE IF SHOULD READ REVERSE
2639 007642 001344          BNE     RSEX          ;IF NOT: BR
2640 007644 032737 010000 000562  BIT    #10000,RDCMD
2641 007652 001005          BNE     RSFR2          ;IF READ REVERSE: BR
```



```
2654 ;*****
2655 ;READ ROUTINE:
2656 ;
2657 ;THIS ROUTINE PERFORMS THE READ OPERATION DETERMINED
2658 ;BY THE READ SEQUENCE ROUTINE ONE RECORD AT A TIME.
2659 ;AT THE END OF EACH READ OPERATION THE STATUS REGISTER
2660 ;IS SCANNED FOR EITHER END OF TAPE OR BEGINNING OF TAPE.
2661 ;IF EOT WAS REACHED, CONTROL WILL BE PASSED TO
2662 ;THE EOT SUBROUTINE TO REWIND THE UNIT AND FLAG IT
2663 ;UNAVAILABLE UNTIL ALL UNITS HAVE REACHED EOT.
2664 ;IF BOT WAS REACHED AN ERROR IS PRINTED AND THE
2665 ;PROGRAM WILL HALT. TESTING MAY BE RESUMED BY PRESSING
2666 ;THE CONTINUE SWITCH.
2667 ;IF A TAPE MARK IS EXPECTED (TM=1) THEN THE
2668 ;READ ROUTINE EXPECTS THE FIRST RECORD OF A
2669 ;READ REVERSE TO BE A TM, AND THE LAST RECORD
2670 ;OF A READ FORWARD TO BE A TM. REMEMBER
2671 ;THAT THE TM ADDS ONE (1) TO THE TOTAL NUMBER
2672 ;OF RECORDS IN A BLOCK.
2673 ;CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13) DETERMINE WHETHER
2674 ;OR NOT TO CHECK FOR STATUS ERRORS (11) OR DATA ERRORS (13),
2675 ;CONSOLE SWITCH FIVE (5) IS USED TO CAUSE A CONTINUOUS
2676 ;READ AND SPACE (FORWARD OR REVERSE) OF THE CURRENT
2677 ;RECORD ON TAPE (YOZZLE).
2678 ;*****
2679
2680 007732 013700 000554 READ: MOV RCNT,R0 ;LOAD REC CNTR
2681 007736 005737 000660 TST EOTREC ;SEE IF EOT
2682 007742 100013 BPL RDA ;IF NOT: BR
2683 007744 032737 010000 000562 BIT #10000,RDCMD ;SEE IF READ FORWARD
2684 007752 001407 BEQ RDA ;IF SO: BR
2685 007754 042737 100000 000660 BIC #100000,EOTREC ;CLEAR FLAG
2686 007762 013703 000660 MOV EOTREC,R3 ;GET MODIFIED RECORD COUNT
2687 007766 160300 SUB R3,R0 ;SET RECORD AT
2688 007770 005200 INC R0 ;SET TO PROPER NUMBER OF RECORDS
2689 007772 012737 024761 000652 RDA: MOV #MSG6,EMADDR ;SET ERROR MSG ADDRESS
2690 010000 005037 000676 CLR TMFLG
2691 010004 032737 010000 000562 BIT #10000,RDCMD
2692 010012 001406 BEQ RDO ;IF READ FORWARD: BR
2693 010014 005737 000564 TST TMEX ;SEE IF TM
2694 010020 001403 BEQ RDO ;IF NOT: BR
2695 010022 005237 000676 INC TMFLG ;SET TM FLAG
2696 010026 005200 INC R0
2697 010030 013777 000556 170460 RDO: MOV FMCNT,@FC ;LOAD CHAR CNTR
2698 010036 012777 033464 170450 MOV #RDATA,@BA ;LOAD DATA ADDR
2699 010044 032737 010000 000562 BIT #10000,RDCMD ;SEE IF READ REVERSE
2700 010052 001417 BEQ RD1A ;IF NOT: BR
2701 010054 013703 000556 MOV FMCNT,R3
2702 010060 005103 COM R3
2703 010062 032737 000020 000552 BIT #20,UDES ;SEE IF CORE DUMP
2704 010070 001402 BEQ RD1 ;IF NOT: BR
2705 010072 000241 CLC
2706 010074 006003 ROR R3 ;R3 = FC/2
2707 010076 060377 170412 RD1: ADD R3,@BA ;SET REVERSE BUS ADDRESS
2708 010102 012737 000076 000672 MOV #76,MTCT ;SET READ REVERSE
2709 010110 000403 BR RD1B
```

Address	Hex	Dec	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op41
---------	-----	-----	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	------

CZTU
CZTU

[illegible]

Address	Hex	Dec	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op41
---------	-----	-----	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	------

[illegible]

```
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825 010706 032777 000020 167672 RDRTY: BIT #20,@SWR ;SEE IF RETRY INHIBITED
2826 010714 001001 BNE RDRT0 ;IF NOT: BR
2827 010716 000207 RTS PC ;ELSE RETURN
2828 010720 013703 000722 RDRT0: MOV ERSAV,R3
2829 010724 042703 102700 BIC #102700,R3 ;MARK NON-RECOVERABLE ERROR BITS
2830 010730 001410 BEQ RDRT1 ;IF NOT: BR
2831 010732 004737 022730 JSR PC,PAPRT ;PRINT HEADER
2832 010736 012704 026604 MOV #MSG79,R4
2833 010742 104000 TTOUTT ;PRINT NON-RECOVERABLE MESSAGE
2834 010744 004737 011260 JSR PC,NRTP ;PRINT ER FOR NON-RETRYABLE ERROR
2835 010750 000207 RDRT1A: RTS PC ;RETURN
2836 010752 032777 002000 167626 RDRT1: BIT #2000,@SWR ;SEE IF PRINT INHIBITED
2837 010760 001003 BNE RDRT1B ;IF SO: BR
2838 010762 012704 026257 MOV #MSG64,R4
2839 010766 104000 TTOUTT ;PRINT ORIGINAL ERROR TAG
2840 010770 005037 000702 RDRT1B: CLR RTCNT ;CLEAR RETRY COUNTER
2841 010774 005037 000706 RDRTG: CLR SERFL ;CLEAR STATUS ERROR FLAG
2842 011000 012737 000002 000712 MOV #2,RTYFL ;SET READ RETRY FLAG
2843 011006 004737 011274 JSR PC,YOZ ;GO TO YOZZLE TO RETRY READ
2844 011012 005737 000706 TST SERFL ;SEE IF RETRY ERROR
2845 011016 001031 BNE RDRT5 ;IF SO: BR
2846 011020 032777 002000 167560 BIT #2000,@SWR
2847 011026 001011 BNE RDRT2
2848 011030 012704 026736 MOV #MSG105,R4
2849 011034 104000 TTOUTT ;PRINT RECOVERED MESSAGE
2850 011036 012704 026301 MOV #MSG65,R4
2851 011042 104000 TTOUTT ;PRINT RETRY TAG
2852 011044 013703 000702 MOV RTCNT,R3
2853 011050 104002 OCTPP ;PRINT RETRY NUMBER
2854 011052 013704 000674 RDRT2: MOV UNP,R4
2855 011056 032737 010000 000562 BIT #10000,RDCMD ;SEE IF READ REVERSE
2856 011064 001003 BNE RDRT3 ;IF SO: BR
2857 011066 005264 002670 INC RFSOFT(R4) ;ELSO BUMP FORWARD SOFT ERROR COUNTER
2858 011072 000402 BR RDRT4
2859 011074 005264 002710 RDRT3: INC RRSOFT(R4) ;BUMP ERRORS SOFT CNTR
2860 011100 000207 RDRT4: RTS PC ;RETURN
2861 011102 005037 000646 RDRT5: CLR TEMP3 ;++G CLEAR RECOVERABLE ERROR INDICATOR
2862 011106 013703 000722 MOV ERSAV,R3 ;GET ER
2863 011112 042703 102700 BIC #102700,R3 ;MASK RECOVERABLE BITS
2864 011116 001413 BEQ RDRT5A ;IF RECOVERABLE: BR
2865 011120 004737 022730 JSR PC,PAPRT ;PRINT HEADER
2866 011124 012704 026604 MOV #MSG79,R4
2867 011130 104000 TTOUTT ;PRINT NON-RECOVERABLE MSG
2868 011132 004737 011260 JSR PC,NRTP ;PRINT ER
2869 011136 012737 000001 000646 MOV #1,TEMP3 ;SET FLAG
```

2870	011144	000404			BR	RDRT5B		
2871	011146	032777	002000	167432	RDRT5A:	BIT	#2000,@SWR	;SEE IF PRINT INHIBITED
2872	011154	001014				BNE	RDRT6	;IF SO: BR
2873	011156	012704	026301		RDRT5B:	MOV	#MSG65,R4	
2874	011162	104000				TTOUIT		;PRINT RETRY TAG
2875	011164	013703	000702			MOV	RTCNT,R3	
2876	011170	104002				OCTPP		;PRINT RETRY NUMBER
2877	011172	005737	000646			TST	TEMP3	;SEE IF DID NON-RECOVERABLE
2878	011176	001403				BEQ	RDRT6	;IF NOT: BR
2879	011200	005037	000646			CLR	TEMP3	;CLEAR FLAG
2880	011204	000207				RTS	PC	;EXIT
2881	011206	005237	000702		RDRT6:	INC	RTCNT	
2882	011212	023737	000702	000602		CMP	RTCNT,RETRY	;SEE IF DONE 8 RETRIES
2883	011220	001265				BNE	RDRTG	;IF NOT: BR
2884	011222	012704	027271			MOV	#MSG115,R4	
2885	011226	104000				TTOUIT		;PRINT HARD ERROR MESSAGE
2886	011230	013704	000674			MOV	UNP,R4	
2887	011234	032737	010000	000562		BIT	#10000,RDCMD	;SEE IF READ REVERSE
2888	011242	001003				BNE	RDRT7	;IF SO: BR
2889	011244	005264	002730			INC	RFHARD(R4)	;BUMP FORWARD HARD ERROR CNTR
2890	011250	000402				BR	RDRTX	
2891	011252	005264	002750		RDRT7:	INC	RRHARD(R4)	;BUMP REVERSE HARD ERROR CNTR
2892	011256	000207			RDRTX:	RTS	PC	;RETURN
2893								
2894	011260	013703	000722		NRTP:	MOV	ERSAV,R3	;GET ER REGISTER
2895	011264	104002				OCTPP		;PRINT ER
2896	011266	004737	021156			JSR	PC,FRPRT	;PRINT F OR R
2897	011272	000207				RTS	PC	;RETURN

[illegible]

2954	011562	032737	000020	000552	BIT	#20,UDES	;SEE IF CORE DUMP
2955	011570	001402			BEQ	YOZD4	;IF NOT: BR
2956	011572	000241			CLC		
2957	011574	006004			ROR	R4	;SET TO FC/2
2958	011576	060403			YOZD4:	ADD R4,R3	;SET EXPT BUS ADDRESS
2959	011600	042703	000001		BIC	#1,R3	;MAKE EXPT ADDRESS EVEN
2960	011604	032737	002000	000552	BIT	#2000,UDES	;SEE IF PE
2961	011612	001001			BNE	YOZD2	;IF SO: BR
2962	011614	005743			TST	-(R3)	;SET EXPT BA
2963	011616	004737	017504		YOZD2:	JSR PC,ER2	;GO CHECK ERRORS
2964	011622	000430			BR	YOZF	
2965	011624	012703	033464		YOZD0:	MOV #RDATA,R3	
2966	011630	032737	002000	000552	BIT	#2000,UDES	;SEE IF PE
2967	011636	001001			BNE	YOZD3	;IF SO: BR
2968	011640	005723			TST	(R3)+	;SET EXPT BA
2969	011642	004737	017504		YOZD3:	JSR PC,ER2	;GO CHECK ERRORS
2970	011646	000416			BR	YOZF	
2971	011650	004737	017406		YOZD1:	JSR PC,ERCHK	;ELSE GO CHECK ERRORS
2972	011654	032777	020000	166724	YOZE:	BIT #20000,@SWR	;SEE IF SHOULD CHECK DATA
2973	011662	001010			BNE	YOZF	;IF NOT: BR
2974	011664	005737	000676		TST	TMFLG	;SEE IF TAPE MARK
2975	011670	001005			BNE	YOZF	;IF SO: BR
2976	011672	005737	000712		TST	RTYFL	;SEE IF RETRY
2977	011676	001004			BNE	YOZF0	;IF SO: BR
2978	011700	004737	015544		JSR	PC,DCHK	;ELSE GO CHECK DATA
2979	011704	004737	014352		YOZF:	JSR PC,DS3	;GO CLEAR DATA AREA
2980	011710	105777	166674		YOZF0:	TSTB @TKS	;SEE IF HAVE NEW STALL VALUE
2981	011714	100032			BPL	YOZG	;IF NOT: BR
2982	011716	122777	000203	166666	CMPB	#203,@TKB	;SEE IF CONT C
2983	011724	001026			BNE	YOZG	;IF NOT: BR
2984	011726	012704	025761		MOV	#MSG44,R4	
2985	011732	104000			TTOUTT		;PRINT YSTALL REQUEST
2986	011734	013703	000600		MOV	YSTAL,R3	
2987	011740	104002			OCTPP		;PRINT PRESENT STALL
2988	011742	010037	000646		MOV	RO,TEMP3	;SAVE RO(REC CNT)
2989	011746	012705	000600		MOV	#YSTAL,R5	;SET ADDRESS OF YSTL
2990	011752	012701	000006		MOV	#6,R1	;SET NUMBER OF CHAR TO INPUT
2991	011756	012702	177777		MOV	#-1,R2	;SET MAXIMUM LIMIT
2992	011762	012703	002000		MOV	#2000,R3	;SET MINIMUM LIMIT
2993	011766	004737	023346		JSR	PC,TTR	;GO GET VALUE
2994	011772	013700	000646		MOV	TEMP3,RO	;RESTORE RO(REC CNTR)
2995	011776	000137	011274		JMP	YOZ	;RESTART YOZZLE
2996	012002	122777	000207	166602	YOZG:	CMPB #207,@TKB	;CHECK FOR CNTL G
2997	012010	001010			BNE	YOZI	
2998	012012	022737	000176	000606	CMP	#SWREG,SWR	;IS SWREG SELECTED
2999	012020	001004			BNE	YOZI	
3000	012022	005077	166564		CLR	@TKB	;CLEAR CNTL G OUT OF BUFFER
3001	012026	004737	024444		JSR	PC,CNTG	;GO CHANGE SWREG
3002	012032	032777	000040	166546	YOZI:	BIT #40,@SWR	;SEE IF SHOULD CONTINUE YOZZLE
3003	012040	001402			BEQ	YOZH	;IF NOT: BR
3004	012042	000137	011314		JMP	YOZO	
3005	012046	012777	000100	166534	YOZH:	MOV #100,@TKS	;SET TTY INTERRUPT ENABLE
3006	012054	000207			RTS	PC	;EXIT
3007							

```
3008
3009
3010 ;*****
3011 ;BACKSPACE SUBROUTINE:
3012 ;
3013 ;THIS SUBROUTINE IS USED TO PERFORM THE
3014 ;BACKSPACE OPERATION REQUIRED BY THE READ
3015 ;ROUTINE FOR READ FORWARD AFTER WRITING.
3016 ;IF A TAPE MARK IS EXPECTED (TM=1) THEN THE SPACE
3017 ;ROUTINE ASSUMES THAT THE TM WILL BE FIRST WHEN
3018 ;BACKSPACING. THEREFORE TWO OPERATIONS ARE REQUIRED
3019 ;TO SPACE OVER A BLOCK. FIRST SPACE OVER THE TM, THEN
3020 ;SPACE OVER THE DATA RECORDS.
3021 ;A CHECK FOR RECORD COUNT ZERO IS MADE AT THE
3022 ;END OF THE SPACE OPERATION TO ASSURE THAT PROPER
3023 ;TAPE POSITIONING WAS DONE.
3024 ;*****
3025 012056 013737 000576 000666 BKSP: MOV TSTAL,STAL
3026 012064 004737 012304 JSR PC,STALL ;DO TURN AROUND STALL
3027 012070 012737 025011 000652 MOV #MSG10,EMADDR
3028 012076 012777 033464 166410 MOV #RDATA,@BA
3029 012104 005737 000564 TST TMEX ;SEE IF TM
3030 012110 001440 BEQ B0 ;IF NOT: BR
3031 012112 012777 177777 166376 MOV #-1,@FC
3032 012120 012737 000032 000672 MOV #32,MTC1
3033 012126 012737 012140 000662 MOV #BKTM,RTRN
3034 012134 000137 021216 JMP TAPG ;SPACE TO TM
3035 012140 032777 010000 166440 BKTM: BIT #10000,@SWR ;SEE IF SHOULD CHECK ERROR
3036 012146 001021 BNE B0 ;IF NOT: BR
3037 012150 012737 026173 000652 MOV #MSG55,EMADDR
3038 012156 032777 000004 166336 BIT #4,@DS ;SEE IF TM
3039 012164 001006 BNE BKTM0 ;IF SO: BR
3040 012166 012737 033464 021132 MOV #RDATA,CADER
3041 012174 004737 020234 JSR PC,ERPT ;PRINT ERROR
3042 012200 000404 BR B0
3043 012202 012703 033464 BKTM0: MOV #RDATA,R3
3044 012206 004737 017504 JSR PC,ER2
3045 012212 013700 000554 B0: MOV RCNT,R0
3046 012216 005100 COM R0 ;BUILD SPACE AMOUNT
3047 012220 005200 INC R0
3048 012222 012737 025011 000652 MOV #MSG10,EMADDR ;SET ERROR MESG ADDRESS
3049 012230 010077 166262 MOV R0,@FC
3050 012234 012737 000032 000672 BKRT: MOV #32,MTC1 ;SET SPACE REVERSE
3051 012242 012737 012260 000662 MOV #B1,RTRN ;SET RETURN ADDRESS
3052 012250 010037 000666 MOV R0,STAL ;SET INTERRUPT TIME MULTIPLIER
3053 012254 000137 021216 JMP TAPG ;GO DO SPACE
3054 012260 012703 033464 B1: MOV #RDATA,R3
3055 012264 004737 017504 JSR PC,ER2
3056 012270 013737 000576 000666 B2: MOV TSTAL,STAL ;DO STALL
3057 012276 004737 012304 JSR PC,STALL ;STALL
3058 012302 000207 RTS PC ;EXIT
3059
```

3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079 012304 005337 000666
3080 012310 001375
3081 012312 000207

;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.

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

```

3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095 012314 012701 177760
3096 012320 012702 174000
3097 012324 004737 023314
3098 012330 042737 000001 000626
3099 012336 013737 000626 000556
3100 012344 012737 177777 014406
3101 012352 000207
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113 012354 012702 000001
3114 012360 012701 000500
3115 012364 004737 023314
3116 012370 013737 000626 000554
3117 012376 000207
3118
3119

```

```

;*****
;RANDOM CHARACTER COUNT GENERATOR:
;
;THIS ROUTINE ENTERED VIA CONSOLE SWITCH
;SEVEN (7) IS USED TO GENERATE A RANDOM
;CHARACTER COUNT FOR EACH DATA BLOCK.
;ALL RECORDS WITHIN A GIVEN BLOCK WILL BE
;THE SAME, BUT EACH BLOCK WILL VARY.
;THE LIMITS ARE TWENTY (20) TO FOUR THOUSAND
;(4000) OCTAL CHARACTERS PER RECORD.
;*****
CCNTR: MOV    #-20,R1      ;SET HIGH LIMIT
        MOV    #-4000,R2  ;SET LOW LIMIT
        JSR    PC,RANG    ;GO GENERATE NUMBER
        BIC    #1,RANSAV
        MOV    RANSAV,FMCNT ;SET CHAR COUNT
        MOV    #-1,PATS    ;PRESET DATA PATTERN
        RTS    PC         ;EXIT
;*****
;RANDOM RECORD COUNT GENERATOR:
;
;THIS ROUTINE ENTERED VIA CONSOLE SWITCH SIX (6)
;IS USED TO GENERATE A RANDOM NUMBER OF RECORDS
;FOR EACH BLOCK OF DATA.
;THE LIMITS ARE ONE (1) TO FIVE HUNDRED (500) OCTAL
;RECORDS PER BLOCK.
;*****
RCNTR:  MOV    #1,R2      ;SET LOW LIMIT
        MOV    #500,R1    ;SET HIGH LIMIT
        JSR    PC,RANG    ;GO GENERATE NUMBER
        MOV    RANSAV,RCNT ;SET RECORD COUNT
        RTS    PC         ;EXIT

```

```
3120
3121
3122 ;*****
3123 ;TEST CONDITION ENTRY ROUTINE:
3124 ;
3125 ;THIS ROUTINE IS USED TO ALLOW THE OPERATOR
3126 ;TO ENTER, AT THE TTY, THE NECESSARY PARAMETERS
3127 ;TO RUN THE PROGRAM AS HE WISHES. THE
3128 ;ROUTINE IS ONLY ENTERED UPON INITIAL STARTING
3129 ;FROM LOCATION 200(8).
3130 ;THE MAIN PURPOSE OF THIS ROUTINE IS TO ESTABLISH
3131 ;A TABLE OF DEVICES TO BE TESTED. THIS TABLE
3132 ;CONSISTS OF AN ENTRY FOR EACH OF ONE (1) TO
3133 ;EIGHT (8) DEVICES. EACH ENTRY CONTAINS THE
3134 ;SLAVE NUMBER, DENSITY, PARITY, AND
3135 ;FORMAT. THE INFORMATION IS ENTERED
3136 ;IN RESPONSE TO PRINTED REQUESTS AT THE TTY.
3137 ;SLAVES MAY BE ENTERED IN ANY ORDER. EACH
3138 ;PARAMETER IS CHECKED FOR LEGALITY BEFORE BEING
3139 ;SET INTO THE TABLE.
3140 ;THE DRIVE NUMBER REQUEST WILL ALSO CHECK THE MASSBUS
3141 ;FOR THE PRESENCE OF THE REQUESTED DRIVE. IF IT IS NOT FOUND,
3142 ;A NON-EXIST DRIVE MESSAGE WILL BE PRINTED AND ANOTHER DRIVE
3143 ;REQUEST MADE. WHEN THE DRIVE IS FOUND, THE RESPONSE IS STORED
3144 ;AND CONTROL PASSED TO THE SLAVE SELECT ROUTINE.
3145 ;THE SLAVE SELECT ROUTINE ALSO CHECKS FOR THE PRESENCE OF THE
3146 ;SLAVE. IF IT IS NOT PRESENT, A MESSAGE IS PRINTED AND ANOTHER
3147 ;REQUEST IS ISSUED. WHEN THE SELECTED SLAVE IS FOUND TO BE
3148 ;PRESENT, A MESSAGE IS PRINTED IF IT IS A 7 CHANNEL DRIVE
3149 ;TO ASSIST IN SELECTING DENSITY, PARITY, AND FORMAT.
3150 ;UPON COMPLETION OF THE DEVICE TABLE, REQUESTS
3151 ;ARE PRINTED FOR ENTRY OF THE NUMBER OF CHARACTERS
3152 ;PER RECORD AND THE NUMBER OF RECORDS PER BLOCK. THE
3153 ;NEXT REQUEST IS FOR A PATTERN NUMBER TO BE USED
3154 ;FOR WRITING AND CHECKING OF READ DATA.
3155 ;FOLLOWING THE PATTERN REQUEST IS THE TAPE MARK OPTION.
3156 ;RESPONDING TO THE REQUEST (TM=) WITH A ONE (1)
3157 ;WILL CAUSE THE PROGRAM TO WRITE A TM AT THE
3158 ;END OF EACH DATA BLOCK AND TO EXPECT THE
3159 ;TM TO BE DETECTED IN EITHER READ FORWARD AND REVERSE
3160 ;OR DURING SPACE OPERATION. A RESPONSE OF ZERO (TM=0)
3161 ;DISALLOWS WRITING OF THE TM AND CAUSES THE READ
3162 ;AND SPACE ROUTINES TO EXPECT NO TM TO BE PRESENT.
3163 ;THE LAST REQUESTS ARE FOR ENTRY OF THE DESIRED
3164 ;WRITE, READ, AND TURN AROUND STALLS.
3165 ;*****
3166 012400 005737 000634 TINP: TST TINF ;SEE IF SHOULD INPUT FROM TTY
3167 012404 001001 BNE TINPA ;IF SO: BR
3168 012406 000207 RTS PC ;EXIT
3169 012410 005037 000674 TINPA: CLR UNP ;CLEAR TABLE POINTER
3170 012414 005037 005156 CLR REOTC ;CLEAR EOT UNIT COUNTER
3171 012420 012700 000010 MOV #10,R0 ;SET SIZE OF TABLE
3172 012424 012701 000746 MOV #UN1,R1 ;SET START OF TABLE
3173 012430 005021 TINPB: CLR (R1)+ ;CLEAR TABLE
3174 012432 005300 DEC R0 ;SEE IF DONE
3175 012434 001375 BNE TINPB ;IF NOT: BR
```

```
3176 012436 012704 025453      MOV      #MSG31,R4
3177 012442 005737 000734      TST      ASEQF      ;SEE IF AUTO SEQ
3178 012446 001402              BEQ      TINPB1      ;IF NOT: BR
3179 012450 012704 025401      MOV      #MSG30,R4      ;SET AUTO SEQ HDR
3180 012454 104000      TINPB1: TTOUTT      ;PRINT PROGRAM NAME
3181 012456 005737 014150      TST      SCVFL      ;SEE IF SHORT CONVERSATION
3182 012462 001067              BNE      TINPC      ;IF SO: BR
3183 012464 012704 026463      MOV      #MSG74,R4
3184 012470 104000      TTOUTT      ;REQUEST STARTING REGISTER ADDRESS
3185 012472 013703 000544      MOV      REGS,R3
3186 012476 104002      OCTPP      ;PRINT CURRENT REG START
3187 012500 012705 000544      MOV      #REGS,R5      ;SAVE ADDRESS LOCATION
3188 012504 012701 000006      MOV      #6,R1      ;SET SIZE OF ENTRY
3189 012510 012702 176400      MOV      #176400,R2      ;SET UPPER LIMIT
3190 012514 012703 172300      MOV      #172300,R3      ;SET LOWER LIMIT
3191 012520 004737 023346      JSR      PC,TTR      ;GO GET RESPONSE
3192 012524 012704 026506      MOV      #MSG75,R4
3193 012530 104000      TTOUTT      ;GO REQUEST VECTOR ADDRESS
3194 012532 013703 000546      MOV      VECT,R3
3195 012536 104002      OCTPP      ;PRINT CURRENT VECTOR
3196 012540 012705 000546      MOV      #VECT,R5      ;SET SAVE LOCATION
3197 012544 012701 000003      MOV      #3,R1      ;SET SIZE OF ENTRY
3198 012550 012702 000224      MOV      #224,R2      ;SET UPPER LIMIT
3199 012554 012703 000150      MOV      #150,R3      ;SET LOWER LIMIT
3200 012560 004737 023346      JSR      PC,TTR      ;GO GET RESPONSE
3201 012564 013700 000546      MOV      VECT,R0      ;GET VECTOR ADDRESS
3202 012570 012720 021714      MOV      #MTINT,(R0)+      ;LOAD VECTOR WITH HANDLER ADDRESS
3203 012574 012710 000340      MOV      #340,(R0)      ;LOAD PRIORITY LEVEL
3204 012600 013700 000544      MOV      REGS,R0      ;GET STARTING REGISTER ADDRESS
3205 012604 012701 000016      MOV      #16,R1      ;SET NUMBER OF REGISTERS
3206 012610 012702 000510      MOV      #C1,R2      ;GET FIRST ADDRESS LOCATION
3207 012614 010022      TINPB0: MOV      R0,(R2)+      ;BUILD TABLE OF ADDRESSES
3208 012616 062700 000002      ADD      #2,R0      ;BUMP ADDRESS
3209 012622 005301      DEC      R1      ;SEE IF DONE
3210 012624 001373      BNE      TINPB0      ;IF NOT: BR
3211 012626 005737 000734      TST      ASEQF      ;SEE IF AUTO SEQ
3212 012632 001403      BEQ      TINPC      ;IF NOT: BR
3213 012634 005726      TST      (SP)+      ;RESET STACK POINTER
3214 012636 000137 021732      JMP      ASEQ      ;GO TO AUTO SEQUENCE
3215 012642 012777 000040      165650 TINPC: MOV      #40,@CS      ;INITIALIZE
3216 012650 012704 026127      MOV      #MSG52,R4
3217 012654 104000      TTOUTT      ;REQUEST DRIVE NUMBER
3218 012656 012705 000550      MOV      #DVN,R5      ;GET ADDRESS
3219 012662 012701 000001      MOV      #1,R1      ;SET SIZE OF RESPONSE
3220 012666 012702 000007      MOV      #7,R2      ;SET UPPER LIMIT
3221 012672 012703 000000      MOV      #0,R3      ;SET LOWER LIMIT
3222 012676 004737 023346      JSR      PC,TTR      ;GO GET DRIVE NUMBER
3223 012702 013777 000550      165610 MOV      DVN,@CS
3224 012710 005777 165574      TST      @C1
3225 012714 032777 010000      165576 BIT      #10000,@CS      ;ACCESS DRIVE
3226 012722 001411      BEQ      TINP0      ;SEE IF NED
3227 012724 012704 026420      MOV      #MSG71,R4      ;IF NOT: BR
3228 012730 104000      TTOUTT      ;PRINT NED
3229 012732 013704 000510      MOV      C1,R4
3230 012736 005204      INC      R4
3231 012740 152714 000100      BISB      #100,(R4)      ;CLEAR TRE
```

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

3288	013270	004737	023346		JSR	PC,TTR	;GO INPUT PARITY
3289	013274	005737	000642		TST	TEMP1	;SEE IF HAVE NEW PARAMETER
3290	013300	001407			BEQ	TINP2A	;IF NOT: BR
3291	013302	042737	000010	000552	BIC	#10,UDES	;ELSE CLEAR OLD PARAMETER
3292	013310	012703	000003		MOV	#3,R3	;SET POSITION FACTOR
3293	013314	004737	014152		JSR	PC,TPOS	;GO LOAD PARITY TO PROPER POSITION
3294	013320	012704	026151		TINP2A: MOV	#MSG53,R4	
3295	013324	104000			TTOUTT		;REQUEST FORMAT
3296	013326	005037	000644		CLR	TEMP2	
3297	013332	012701	000002		MOV	#2,R1	
3298	013336	012702	000016		MOV	#16,R2	
3299	013342	012703	000014		MOV	#14,R3	
3300	013346	004737	023346		JSR	PC,TTR	;GO GET FORMAT
3301	013352	005737	000642		TST	TEMP1	;SEE IF NEW PARAMETER
3302	013356	001407			BEQ	TINP2B	;IF NOT: BR
3303	013360	042737	000170	000552	BIC	#170,UDES	
3304	013366	012703	000004		MOV	#4,R3	
3305	013372	004737	014152		JSR	PC,TPOS	
3306	013376	005237	005156		TINP2B: INC	REOTC	;BUMP EOT UNIT COUNTER
3307	013402	022737	000016	000674	CMP	#16,UNP	;SEE IF DONE UNITS
3308	013410	001405			BEQ	TINP2C	;IF SO: BR
3309	013412	062737	000002	000674	ADD	#2,UNP	;POINT TO NEXT UNIT
3310	013420	000137	012746		JMP	TINP0	;ELSE LOOK FOR NEXT UNIT
3311	013424	005037	000674		TINP2C: CLR	UNP	;CLEAR UNIT POINTER
3312	013430	013700	005156		MOV	REOTC,R0	
3313	013434	000337	005156		SWAB	REOTC	
3314	013440	110037	005156		MOVB	R0,REOTC	;SET UNIT EOT COUNTER
3315	013444	012704	025565		TINP3: MOV	#MSG35,R4	
3316	013450	104000			TTOUTT		;PRINT RECORD COUNT REQUEST
3317	013452	013703	000554		MOV	RCNT,R3	
3318	013456	104002			OCTPP		;PRINT RECORD COUNT
3319	013460	012705	000554		MOV	#RCNT,R5	;SET RECORD COUNT ADDRESS
3320	013464	012701	000006		MOV	#6,R1	;SET NUMBER OF CHARACTERS TO INPUT
3321	013470	012702	177777		MOV	#-1,R2	;SET MAXIMUM LIMIT
3322	013474	012703	000001		MOV	#1,R3	;SET MINIMUM LIMIT
3323	013500	004737	023346		JSR	PC,TTR	;GO GET RECORD COUNT
3324	013504	013737	000554	000630	MOV	RCNT,RCSAV	;SAVE RECORD COUNT
3325	013512	012704	025606		MOV	#MSG36,R4	
3326	013516	104000			TTOUTT		;PRINT CHARACTER COUNT REQUEST
3327	013520	005437	000556		NEG	FMCNT	
3328	013524	013703	000556		MOV	FMCNT,R3	
3329	013530	104002			OCTPP		;PRINT CHAR COUNT
3330	013532	012705	000556		MOV	#FMCNT,R5	;SET CHARACTER COUNT ADDRESS
3331	013536	012701	000006		MOV	#6,R1	;SET NUMBER OF CHARACTERS TO INPUT
3332	013542	012702	004000		MOV	#4000,R2	;SET MAXIMUM LIMIT
3333	013546	012703	000004		MOV	#4,R3	;SET MINIMUM LIMIT
3334	013552	004737	023346		JSR	PC,TTR	;GO GET CHARACTER COUNT
3335	013556	005437	000556		NEG	FMCNT	;SET TO TWO'S COMPLIMENT
3336	013562	013737	000556	000632	MOV	FMCNT,FCSAV	;SAVE FRAME COUNT
3337	013570	012704	025632		MOV	#MSG37,R4	;PRINT PATTERN NUMBER REQUEST
3338	013574	104000			TTOUTT		
3339	013576	013703	000560		MOV	PATRN,R3	
3340	013602	104002			OCTPP		;PRINT PATTERN
3341	013604	005037	014554		CLR	DOFL	;CLEAR EXTERNAL DATA FLAG
3342	013610	012705	000560		MOV	#PATRN,R5	;SET PATTERN NUMBER ADDRESS
3343	013614	012701	000002		MOV	#2,R1	;SET NUMBER OF CHARACTERS TO INPUT

Address	Hex	Dec	Label	Instruction	Comment
3344	013620	012702	000015	MOV	#15,R2 ;SET MAXIMUM LIMIT
3345	013624	012703	000000	MOV	#0,R3 ;SET MINIMUM LIMIT
3346	013630	004737	023346	JSR	PC,TTR ;GO GET PATTERN NUMBER
3347	013634	012704	026346	MOV	#MSG69,R4
3348	013640	104000		TTOUTT	;REQUEST TM
3349	013642	013703	000564	MOV	TMEX,R3
3350	013646	104002		OCTPP	;PRINT CURRENT TM FLAG SETTING
3351	013650	012705	000564	MOV	#TMEX,R5 ;GET TM FLAG ADDRESS
3352	013654	012701	000001	MOV	#1,R1 ;SET SIZE OF RESPONSE
3353	013660	012702	000001	MOV	#1,R2 ;SET UPPER LIMIT
3354	013664	012703	000000	MOV	#0,R3 ;SET LOWER LIMIT
3355	013670	004737	023346	JSR	PC,TTR ;TM 1=YES
3356	013674	012704	025133	MOV	#MSG21,R4
3357	013700	104000		TTOUTT	;REQUEST INTERCHANGE READ
3358	013702	013703	000566	MOV	INTRF,R3
3359	013706	104002		OCTPP	;PRINT CURRENT SETTING
3360	013710	012705	000566	MOV	#INTRF,R5 ;GET FLAG ADDRESS
3361	013714	012701	000001	MOV	#1,R1 ;SET SIZE OF RESPONSE
3362	013720	012702	000001	MOV	#1,R2 ;SET UPPER LIMIT
3363	013724	012703	000000	MOV	#0,R3 ;SET LOWER LIMIT
3364	013730	004737	023346	JSR	PC,TTR ;GO GET RESPONSE
3365	013734	012704	025655	MOV	#MSG38,R4
3366	013740	104000		TTOUTT	;REQUEST SINGLE PASS
3367	013742	013703	000570	MOV	SPFLG,R3
3368	013746	104002		OCTPP	;PRINT CURRENT SETTING
3369	013750	012705	000570	MOV	#SPFLG,R5 ;SET ADDRESS OF FLAG
3370	013754	012701	000001	MOV	#1,R1 ;SET SIZE OF RESPONSE
3371	013760	012702	000001	MOV	#1,R2 ;SET UPPER LIMIT
3372	013764	012703	000000	MOV	#0,R3 ;SET LOWER LIMIT
3373	013770	004737	023346	JSR	PC,TTR ;GO GET RESPONSE
3374	013774	005737	014150	TINP4: TST	SCVFL ;SEE IF SHORT CONVERSATION
3375	014000	001060		BNE	TINPX ;IF SO: BR
3376	014002	012704	025675	MOV	#MSG40,R4
3377	014006	104000		TTOUTT	;PRINT READ STALL REQUEST
3378	014010	013703	000572	MOV	RSTAL,R3
3379	014014	104002		OCTPP	;PRINT READ STALL
3380	014016	012705	000572	MOV	#RSTAL,R5 ;SET READ STALL ADDRESS
3381	014022	012701	000006	MOV	#6,R1 ;SET NUMBER OF CHARACTERS TO INPUT
3382	014026	012702	177777	MOV	#-1,R2 ;SET MAXIMUM LIMIT
3383	014032	012703	000001	MOV	#1,R3 ;SET MINIMUM LIMIT
3384	014036	004737	023346	JSR	PC,TTR ;GO GET READ STALL
3385	014042	012704	025723	MOV	#MSG41,R4
3386	014046	104000		TTOUTT	;PRINT WRITE STALL REQUEST
3387	014050	013703	000574	MOV	WSTAL,R3
3388	014054	104002		OCTPP	;PRINT READ STALL
3389	014056	012705	000574	MOV	#WSTAL,R5 ;SET WRITE STALL ADDRESS
3390	014062	012701	000006	MOV	#6,R1 ;SET NUMBER OF CHARACTERS TO INPUT
3391	014066	012702	177777	MOV	#-1,R2 ;SET MAXIMUM LIMIT
3392	014072	012703	000001	MOV	#1,R3 ;SET MINIMUM LIMIT
3393	014076	004737	023346	JSR	PC,TTR ;GO GET WRITE STALL
3394	014102	012704	025735	MOV	#MSG42,R4
3395	014106	104000		TTOUTT	;PRINT TURN AROUND STALL REQUEST
3396	014110	013703	000576	MOV	TSTAL,R3
3397	014114	104002		OCTPP	;PRINT TA STALL
3398	014116	012705	000576	MOV	#TSTAL,R5 ;SET TURN AROUND STALL ADDRESS
3399	014122	012701	000006	MOV	#6,R1 ;SET NUMBER OF CHARACTERS TO INPUT

```

;UNIT DESCRIPTION POSITIONING SUBROUTINE*****

```

[illegible]

```
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435 014200 005737 015150          DSUP: TST      RDFL          ;SEE IF DID RANDOM DATA
3436 014204 001044                      BNE      DS1          ;++G F SO BRANCH
3437 014206 005737 000734          DS0:  TST      ASEQF         ;SEE IF AUTO SEQ
3438 014212 001406                      BEQ      DSOC          ;IF NOT: BR
3439 014214 005737 000560          TST      PATRN          ;SEE IF AUTO RANDOM
3440 014220 100003                      BPL      DSOC          ;IF NOT: BR
3441 014222 004737 015102          JSR      PC,DATR         ;ELSE GO GENERATE RANDOM DATA
3442 014226 000207                      RTS      PC          ;RETURN
3443 014230 023737 000560 014406 DSOC:  CMP      PATRN,PATS       ;SEE IF NEW PATTERN
3444 014236 001014                      BNE      DSOA          ;IF SO: BR
3445 014240 013703 000552          MOV      UDES,R3         ;GET UNIT DESCRIPTION
3446 014244 042703 177767          BIC      #177767,R3       ;MASK EVEN PARITY
3447 014250 023703 014410          CMP      PARS,R3         ;SEE IF SAME AS LAST TIME
3448 014254 001404                      BEQ      DS0B          ;IF SO: BR
3449 014256 010337 014410          MOV      R3,PARS        ;SAVE PARITY
3450 014262 004737 015152          JSR      PC,CRCLRC        ;GO GENERATE EXPT CRC/LRC
3451 014266 000207          DS0B:  RTS      PC
3452 014270 012703 027456          DS0A:  MOV      #WDATA,R3       ;R3 = ADDRS OF WRITE BUFFER
3453 014274 013701 000560          MOV      PATRN,R1        ;R1 = PATTERN SELECTOR
3454 014300 010137 014406          MOV      R1,PATS
3455 014304 062701 000001          ADD      #1,R1
3456 014310 006301          ASL      R1
3457 014312 004771 002770          JSR      PC,@DATBL(R1) ;GO GENERATE PATTERN
3458 014316 032777 010000 164212 DS1:  BIT      #10000,@DT       ;SEE IF 7 CH
3459 014324 001410                      BEQ      DS2A          ;IF NOT: BR
3460 014326 012702 002002          MOV      #2002,R2       ;SET BUFFER SIZE
3461 014332 012701 027456          MOV      #WDATA,R1        ;SET START OF BUFFER
3462 014336 042721 140300          DS2:  BIC      #140300,(R1)+      ;MASK FOR 7 CH
3463 014342 005302                      DEC      R2
3464 014344 001374                      BNE      DS2
3465 014346 004737 015152          DS2A:  JSR      PC,CRCLRC        ;GO GENERATE EXPT CRC/LRC
3466 014352 012702 002000          DS3:  MOV      #2000,R2       ;R2=BUFFER SIZE
3467 014356 012701 033464          MOV      #RDATA,R1        ;R1=READ DATA START
3468 014362 005021          DS4:  CLR      (R1)+
3469 014364 005302                      DEC      R2
3470 014366 001375                      BNE      DS4
3471 014370 013737 000552 014410          MOV      UDES,PARS       ;GET UNIT DESCRIPTION
3472 014376 042737 177767 014410          BIC      #177767,PARS     ;MASK PARITY
```

[illegible]

```

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

```

CZTUAM
 CZTUAM
 4132
 4132
 4134
 4135
 4136
 4137
 4138
 4139
 4140
 4141
 4142
 4143
 4144
 4145
 4146
 4147
 4148
 4149
 4150
 4151
 4152
 4153
 4154
 4155
 4156
 4157
 4158
 4159
 4160
 4161
 4162
 4163
 4164
 4165
 4166
 4167
 4168
 4169
 4170
 4171
 4172
 4173
 4174
 4175
 4176
 4177
 4178
 4179
 4180
 4181
 4182
 4183
 4184
 4185
 4186
 4187
 4188
 4189
 4190

```

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

```

```

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

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

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

CRCLRC: MOV FMCNT,R0 ;SET RECORD SIZE
        NEG R0
        MOV #WDATA,R1 ;SET START OF BUFFER
        CLR XORS
CL0:     MOVB (R1),R4 ;GET CHARACTER
        JSR PC,CLP ;GO GET PARITY OF CHARACTER
        JSR PC,XOR ;XOR CHARACTER
        CLC
        ROR R4 ;ROTATE 1 RIGHT
        BCC CL2 ;IF NO CARRY: BR
        BIS #400,R4 ;SET BIT NINE
        CLC
CL1:     MOV R4,R5 ;SAVE CHARACTER
        BIC #177703,R5
        COM R5
        BIC #177703,R5
        BIC #74,R4
        BIS R5,R4 ;COMPLIMENT BITS 2,3,4,5
CL2:     MOV R4,XORS
        DEC R0
        BEQ CLLAST ;IF LAST CHARACTER: BR
        BR CLO ;++G GET NEXT
CLLAST: MOV XORS,R4
        COM XORS
        BIC #177050,XORS
        BIC #177727,R4 ;COMPLIMENT ALL BUT BITS 3&5
        BIS R4,XORS
        MOV XORS,EXCRC ;SAVE EXPECTED CRC
        MOV FMCNT,R0
        NEG R0
        MOV #WDATA,R1 ;DO EXPT LRC
        CLR XORS
CL3:     MOVB (R1),R4
        JSR PC,CLP ;GET PARITY
        JSR PC,XOR ;XOR CHARACTER
        DEC R0
        BNE CL3 ;DO ALL FOR LRC
        MOV EXCRC,R4
        JSR PC,XOR ;XOR CRC TO DATA
        MOV XORS,EXLRC ;SAVE EXPT LRC
        RTS PC ;RETURN
CLP:     TST R4 ;SEE IF 0 CHAR
        BNE CLPE ;IF NOT: BR
        BIT #10,UDES ;SEE IF EVEN PARITY
        BEQ CLPE ;IF NOT: BR
        MOV #420,R4 ;SET 0 CHAR EVEN PARITY
        INC R1 ;BUMP POINTER
```

```

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

```

```
3739
3740
3741      ;*****
3742      ;DATA CHECK SUBROUTINE:
3743      ;
3744      ;THIS SUBROUTINE IS USED TO COMPARE EACH CHARACTER
3745      ;OF DATA READ FROM TAPE WITH THE EXPECTED CHARACTER.
3746      ;ANY ERROR DETECTED WILL CAUSE CONTROL TO BE
3747      ;PASSED TO AN ERROR PRINT SUBROUTINE AND A
3748      ;SUBROUTINE TO ACCUMULATE THE NUMBER OF BITS
3749      ;DROPPED AND PICKED UP FROM EACH CHARACTER.
3750      ;THE NUMBER OF READ ERRORS IS ALSO ACCUMULATED.
3751      ;DATA CHECKING MAY BE TERMINATED BY USE OF
3752      ;CONSOLE SWITCH THIRTEEN (13).
3753      ;*****
3754 015544 005037 000656      DCHK: CLR      BBC      ;CLEAR BAD RECORD CNTR
3755 015550 005037 000704      CLR      DERFL     ;CLEAR DATA ERROR FLAG
3756 015554 013705 000556      MOV      FMCNT,R5    ;LOAD CHAR COUNT
3757 015560 032737 000020 000552 BIT      #20,UDES  ;SEE IF CORE DUMP
3758 015566 001402      BEQ      DCHK0      ;IF NOT: BR
3759 015570 000261      SEC
3760 015572 006005      ROR      R5          ;R5 = FC/2
3761 015574 012701 027456      DCHK0: MOV      #WDATA,R1 ;SET WRITE DATA ADDR
3762 015600 012702 033464      MOV      #RDATA,R2    ;SET READ DATA ADDR
3763 015604 032737 000010 000552 BIT      #10,UDES  ;SEE IF EVEN PARITY
3764 015612 001430      BEQ      DFOC0      ;IF NOT: BR
3765 015614 032737 000020 000552 BIT      #20,UDES  ;SEE IF CORE DUMP PARITY
3766 015622 001024      BNE      DFOC0      ;IF SO: BR
3767 015624 032737 002000 000552 BIT      #2000,UDES ;SEE IF PE MODE
3768 015632 001020      BNE      DFOC0      ;IF SO: BR
3769 015634 105711      DFOF:  TSTB      (R1)      ;SEE IF 0 CHAR
3770 015636 001404      BEQ      DFOD      ;IF SO: BR
3771 015640 005201      INC      R1          ;BUMP POINTER
3772 015642 005205      DFOE:  INC      R5          ;SEE IF DONE
3773 015644 001373      BNE      DFOF      ;IF NOT: BR
3774 015646 000406      BR       DFOC      ;ELSE CONTINUE
3775 015650 112721 000020      DFOD:  MOVB     #20,(R1)+ ;SET 20 IN PLACE OF 0
3776 015654 012737 177777 014406 MOV      #-1,PATS ;SET PATTERN GENERATE FLAG
3777 015662 000767      BR       DFOE
3778 015664 013705 000556      DFOC:  MOV      FMCNT,R5 ;RESET CHAR CNT
3779 015670 012701 027456      MOV      #WDATA,R1 ;RESET DATA ADDRESS
3780 015674 032737 010000 000562 DFOC0: BIT      #10000,RDCMD ;SEE IF READ REVERSE
3781 015702 001462      BEQ      DFO      ;IF NOT: BR
3782 015704 013704 000556      DFOB:  MOV      FMCNT,R4 ;GET FRAME COUNT
3783 015710 005404      NEG      R4          ;SET TO WHOLE NUMBER
3784 015712 032737 000020 000552 BIT      #20,UDES  ;SEE IF CORE DUMP
3785 015720 001402      BEQ      DFOB0      ;IF NOT: BR
3786 015722 000241      CLC
3787 015724 006004      ROR      R4          ;SET TO FC/2
3788 015726 060401      DFOB0: ADD      R4,R1      ;POINT TO START OF WRITE DATA
3789 015730 060402      ADD      R4,R2      ;POINT TO START OF READ DATA
3790 015732 032737 000001 000556 BIT      #1,FMCNT ;SEE IF ODD FRAME COUNT
3791 015740 001401      BEQ      DFOA      ;IF NOT: BR
3792 015742 105722      TSTB      (R2)+      ;BUMP POINTER
3793 015744 032737 000020 000552 DFOA: BIT      #20,UDES  ;SEE IF CORE DUMP
3794 015752 001431      BEQ      DFOA4      ;IF NOT: BR
```

Address	Hex	Dec	Label	Instruction	Comments
3795	015754	000241		CLC	
3796	015756	132742	000001	BITB #1,-(R2)	;SEE IF BIT 0 = 1
3797	015762	001401		BEQ DFOA0	;IF NOT: BR
3798	015764	000261		SEC	
3799	015766	106012	DFOA0:	RORB (R2)	
3800	015770	000241		CLC	
3801	015772	132712	000001	BITB #1,(R2)	
3802	015776	001401		BEQ DFOA1	
3803	016000	000261		SEC	
3804	016002	106012	DFOA1:	RORB (R2)	;POSITION BITS FOR REVERSE CORE DUMP
3805	016004	000241		CLC	
3806	016006	132712	000001	BITB #1,(R2)	
3807	016012	001401		BEQ DFOA2	
3808	016014	000261		SEC	
3809	016016	106012	DFOA2:	RORB (R2)	
3810	016020	000241		CLC	
3811	016022	132712	000001	BITB #1,(R2)	
3812	016026	001401		BEQ DFOA3	
3813	016030	000261		SEC	
3814	016032	106012	DFOA3:	RORB (R2)	
3815	016034	005202		INC R2	;RESET POINTER
3816	016036	124142	DFOA4:	CMPB -(R1),-(R2)	;TEST DATA CHARACTER
3817	016040	001010		BNE DF1	;IF NOT GOOD: BR
3818	016042	105037	000656	CLRB BBC	;CLEAR BAD RECORD COUNTER
3819	016046	000411		BR DF2	
3820	016050	122122	DF0:	CMPB (R1)+,(R2)+	;CHECK DATA
3821	016052	001003		BNE DF1	;IF BAD: BR
3822	016054	105037	000656	CLRB BBC	;CLEAR BAD RECORD CNTR
3823	016060	000404		BR DF2	
3824	016062	004737	DF1:	JSR PC,DRPKF	;GO GET DROPS AND PICKS
3825	016066	004737	016160	JSR PC,DERR	;GO DO PRINT
3826	016072	005205	DF2:	INC R5	;BUMP CHAR CNTR
3827	016074	001405		BEQ DF3	;IF DONE ALL: BR
3828	016076	032737	010000 000562	BIT #10000,RDCMD	;SEE IF READ REVERSE
3829	016104	001761		BEQ DFO	;IF NOT: BR
3830	016106	000716		BR DFOA	;ELSE CONTINUE READ REV
3831	016110	005037	000664	DF3:	CLR HDRFL
3832	016114	005737	000704	TST DERFL	;CLEAR HEADER FLAG
3833	016120	001416		BEQ DFX	;SEE IF HAD DATA ERROR
3834	016122	005737	000706	TST SERFL	;IF NOT: BR
3835	016126	001013		BNE DFX	;IF NOT DATA ERROR ONLY: BR
3836	016130	013704	000674	MOV UNP,R4	
3837	016134	032737	010000 000562	BIT #10000,RDCMD	;SEE IF READ REVERSE
3838	016142	001003		BNE DF4	;IF SO: BR
3839	016144	005264	001130	INC DATER1(R4)	;BUMP DATA ERROR FORWARD COUNTER
3840	016150	000402		BR DFX	
3841	016152	005264	001170	DF4:	INC DEREV1(R4)
3842	016156	000207		DFX:	RTS
3843					PC

```
3844
3845
3846      ;*****
3847      ;DATA ERROR SUBROUTINE:
3848      ;
3849      ;THIS SUBROUTINE IS USED TO PRINT OUT ANY
3850      ;ERRORS FOUND DURING THE DATA CHECK.
3851      ;EACH CHARACTER FOUND BAD WILL BE PRINTED
3852      ;IN BIT FORMAT ALONG WITH ITS EXPECTED CHARACTER.
3853      ;AN ERROR HEADER CONSISTING OF THE UNIT NUMBER,
3854      ;BLOCK NUMBER, RECORD NUMBER, SIZE OF RECORD, AND
3855      ;ERROR TYPE (READ FORWARD, READ REVERSE, WRITE, ETC)
3856      ;IS PRINTED ONLY ONCE FOR EACH RECORD FOUND BAD.
3857      ;A COUNT IS MADE OF THE NUMBER OF SUCCESSIVE BAD
3858      ;CHARACTERS, AND IF TEN (10) SUCCESSIVE BAD CHARACTERS
3859      ;ARE FOUND IN A SINGLE RECORD, A MESSAGE INDICATING
3860      ;A BAD RECORD CONDITION IS PRINTED AND THE NEXT
3861      ;TWENTY (20) CHARACTERS ARE SKIPPED BEFORE CHECKING
3862      ;IS RESUMED. IF THE BAD RECORD CONDITION IS FOUND
3863      ;THREE TIMES IN A RECORD, ALL REMAINING DATA IS
3864      ;SKIPPED EXCEPT THE FINAL TEN (10) CHARACTERS.
3865      ;THIS SKIPPING IS OF COURSE ONLY POSSIBLE IN
3866      ;RECORDS WHICH CONTAIN A SUFFICIENT NUMBER OF CHARACTERS.
3867      ;PRINTING OF ERRORS MAY BE DISALLOWED AT ANY TIME
3868      ;BY SETTING CONSOLE SWITCH TEN (10) TO A ONE.
3869      ;THE OPERATOR MAY CAUSE THE PROGRAM TO HALT ON ANY ERROR
3870      ;BY SETTING CONSOLE SWITCH FIFTEEN (15) TO A ONE.
3871      ;*****
3872 016160 032777 002000 162420 DERR: BIT #2000,ASWR ;SEE IF SHOULD PRINT ERRORS
3873 016166 001067 BNE DERR4 ;++G BRANCH IF NOT
3874 016170 005237 000670 DERR0: INC PFLG ;SET PRINT FLAG
3875 016174 005737 000664 TST HDRFL ;SEE IF HAVE PRINTED HEADER
3876 016200 001007 BNE DERROA ;IF SO: BR
3877 016202 004737 022730 JSR PC,PAPRT ;PRINT CYCLE NUMBER
3878 016206 012704 024730 MOV #MSG1,R4 ;LOAD ERROR MSG ADDR
3879 016212 104000 TTOUTT ;PRINT ERROR
3880 016214 004737 021156 JSR PC,FRPRT ;PRINT F OR R
3881 016220 012704 024747 DERROA: MOV #MSG4,R4
3882 016224 104000 TTOUTT ;PRINT CHAR NO. HEADER
3883 016226 010203 MOV R2,R3
3884 016230 162703 033464 SUB #RDATA,R3 ;POINT TO CHAR
3885 016234 005303 DEC R3
3886 016236 032737 010000 000562 BIT #10000,RDCMD ;SEE IF READ REVERSE
3887 016244 001402 BEQ DERROB ;IF NOT: BR
3888 016246 010503 MOV R5,R3 ;GET CHAR NUMBER
3889 016250 005103 COM R3
3890 016252 104002 DERROB: OCTPP ;PRINT CHAR NUMBER
3891 016254 012704 024735 MOV #MSG2,R4
3892 016260 104000 TTOUTT ;PRINT EXPECTED DATA
3893 016262 032737 010000 000562 BIT #10000,RDCMD ;SEE IF READ REVERSE
3894 016270 001402 BEQ DERROC ;IF NOT: BR
3895 016272 111103 MOVB (R1),R3 ;GET CHAR
3896 016274 000401 BR DERROD
3897 016276 114103 DERROC: MOVB -(R1),R3 ;LOAD EXPECTED DATA
3898 016300 004737 024160 DERROD: JSR PC,DOUT ;GO PRINT CHAR
3899 016304 012704 024742 MOV #MSG3,R4
```

Address	Hex	Dec	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op41
---------	-----	-----	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	------


```

3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986 016666 005037 000642      DRPKF: CLR      TEMP1
3987 016672 005037 000644      CLR      TEMP2
3988 016676 005037 000646      CLR      TEMP3
3989 016702 111137 000642      MOV      (R1),TEMP1      ;LOAD GOOD CHAR
3990 016706 111237 000644      MOV      (R2),TEMP2      ;LOAD BAD CHAR
3991 016712 013704 000674      MOV      UNP,R4
3992 016716 016437 000770 000720  MOV      PIK1(R4),BPKP
3993 016724 016437 001010 000716  MOV      DRP1(R4),BDPP
3994 016732 032737 010000 000562  BIT      #10000,RDCMD      ;SEE IF READ REVERSE
3995 016740 001005      BNE      DRPK      ;IF SO: BR
3996 016742 124142      CMP      -(R1),-(R2)      ;POINT TO CHAR
3997 016744 112137 000642      MOV      (R1)+,TEMP1      ;LOAD GOOD CHAR
3998 016750 112237 000644      MOV      (R2)+,TEMP2      ;LOAD BAD CHAR
3999 016754 004737 016766      DRPK: JSR      PC,DROP      ;GET DROPS
4000 016760 004737 017204      JSR      PC,PICK      ;GET PICKS
4001 016764 000207      RTS      PC      ;EXIT
4002 016766 113703 000642      DROP: MOV      TEMP1,R3      ;R3 = GOOD CHAR
4003 016772 113704 000644      MOV      TEMP2,R4      ;R4 = BAD CHAR
4004 016776 140403      DPC: BIC      R4,R3      ;GET DROPS/PICKS
4005 017000 001001      BNE      DPCG      ;IF SOME: BR
4006 017002 000207      RTS      PC      ;RETURN
4007 017004 012737 000010 000710  DPCG: MOV      #10,BCNT      ;SET NUMBER TO CHECK
4008 017012 132703 000001      DPC0: BIT      #1,R3      ;SEE IF DROPPED OR PICKED THIS BIT
4009 017016 001455      BEQ      DPC2      ;IF NOT: BR
4010 017020 105737 000646      TST      TEMP3      ;SEE IF ON PICKS
4011 017024 001016      BNE      DPC1      ;IF SO: BR
4012 017026 005277 161664      INC      @BDPP      ;BUMP DROP CNTR
4013 017032 005777 161660      TST      @BDPP
4014 017036 100045      BPL      DPC2      ;IF NO OVERFLOW: BR
4015 017040 032777 002000 161540  BIT      #2000,@SWR      ;SEE IF HAVE PRINTED DATA
4016 017046 001402      BEQ      DPC0A      ;IF SO: BR
4017 017050 004737 022730      JSR      PC,PAPRT      ;PRINT CYCLE NUMBER
4018 017054 004737 017250      DPC0A: JSR      PC,DPPRT      ;PRINT DROPS AND PICKS
4019 017060 000415      BR      DPC2A
4020 017062 005277 161632      DPC1: INC      @BPKP      ;BUMP PICK CNTR
4021 017066 005777 161626      TST      @BPKP      ;SEE IF OVERFLOW
4022 017072 100027      BPL      DPC2      ;IF NOT: BR
4023 017074 032777 002000 161504  BIT      #2000,@SWR      ;SEE IF HAVE PRINTED DATA

```

Address	Hex	Dec	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	
---------	-----	-----	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--

```
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 017406 013703 000556      ERCHK: MOV      FMCNT,R3      ;GET FRAME COUNT
4102 017412 032703 000001      BIT      #1,R3      ;SEE IF ODD
4103 017416 001401              BEQ      ERO      ;IF NOT: BR
4104 017420 005303              DEC      R3      ;BUMP COUNT
4105 017422 005403              NEG      R3
4106 017424 032737 000020 000552 ERO:  BIT      #20,UDES      ;SEE IF CORE DUMP
4107 017432 001402              BEQ      EROB      ;IF NOT: BR
4108 017434 000241              CLC
4109 017436 006003              ROR      R3      ;SET TO FC/2
4110 017440 032737 000010 000672 EROB: BIT      #10,MTC1      ;SEE IF WRITE OP
4111 017446 001414              BEQ      ER1      ;IF SO: BR
4112 017450 032737 010000 000562      BIT      #10000,RDCMD
4113 017456 001405              BEQ      EROA
4114 017460 012703 033464      MOV      #RDATA,R3
4115 017464 162703 000002      SUB      #2,R3      ;SET POINTER
4116 017470 000405              BR      ER2
4117 017472 062703 033464      EROA:  ADD      #RDATA,R3      ;BUILD EXPT READ ADDRESS
4118 017476 000402              BR      ER2
4119 017500 062703 027456      ER1:  ADD      #WDATA,R3      ;BUILD EXPT WRITE ADDRESS
4120 017504 010337 021132      ER2:  MOV      R3,CADER      ;SAVE ADDRESS
4121 017510 012704 000007      MOV      #7,R4
4122 017514 012701 021134      MOV      #BAER,R1
4123 017520 005021      ER2A0: CLR      (R1)+      ;CLEAR FLAGS
4124 017522 005304              DEC      R4
4125 017524 001375              BNE      ER2A0
4126 017526 020377 160762      CMP      R3,@BA      ;SEE IF ADDRESS OK
4127 017532 001402              BEQ      ER2A1      ;IF SO: BR
4128 017534 005237 021134      INC      BAER      ;SET BUS ADDRESS ERROR
4129 017540 032737 000010 000672 ER2A1: BIT      #10,MTC1      ;SEE IF WRITE OPER
4130 017546 001006              BNE      ER2B      ;IF NOT: BR
4131 017550 005777 160742      ER2A: TST      @FC      ;SEE IF FC=0
```

4132	017554	001441				BEQ	ER3	:IF SO: BR
4133	017556	005237	021142			INC	FCER	:SET FC ERROR
4134	017562	000436				BR	ER3	:++G
4135	017564	032737	000040	000672	ER2B:	BIT	#40,MTC1	:SEE IF SPACE OPER
4136	017572	001766				BEQ	ER2A	:IF SO: BR
4137	017574	005737	000676			TST	TMFLG	:SEE IF TM TIME
4138	017600	001011				BNE	ER2D	:IF SO: BR
4139	017602	013703	000556			MOV	FMCNT,R3	
4140	017606	005403				NEG	R3	:R3 = EXPT RECORD SIZE
4141	017610	020377	160702		ER2C:	CMP	R3,@FC	:SEE IF FC = EXPT
4142	017614	001421				BEQ	ER3	:IF SO: BR
4143	017616	005237	021142			INC	FCER	:SET FC ERROR FLAG
4144	017622	000416				BR	ER3	:++G
4145	017624	032737	002000	000552	ER2D:	BIT	#2000,UDES	:SEE IF PE
4146	017632	001346				BNE	ER2A	:IF SO: BR
4147	017634	032737	010000	000562		BIT	#10000,RDCMD	:SEE IF READ REVERSE
4148	017642	001003				BNE	ER2E	:IF SO: BR
4149	017644	012703	000002			MOV	#2,R3	
4150	017650	000757				BR	ER2C	:LOOK FOR EXPT = 2
4151	017652	012703	000001		ER2E:	MOV	#1,R3	
4152	017656	000754				BR	ER2C	:GO CHECK FC FOR TM
4153	017660	032777	160000	160622	ER3:	BIT	#160000,@C1	:SEE IF COUNT ERROR
4154	017666	001441				BEQ	ER4	
4155	017670	017703	160624			MOV	@CS,R3	:GET CONT STATUS REG
4156	017674	042703	000307			BIC	#307,R3	:MASK OUT IR,OR,UNIT NO.
4157	017700	005703				TST	R3	:SEE IF ANY OTHER ERRORS
4158	017702	001407				BEQ	ER3A	:IF NOT: BR
4159	017704	005737	000676			TST	TMFLG	:SEE IF TAPE MARK TIME
4160	017710	001426				BEQ	ER3B	:IF NOT: BR
4161	017712	042703	001000			BIC	#1000,R3	:MASK MISSED TRANS
4162	017716	005703				TST	R3	:SEE IF ANY OTHER ERRORS
4163	017720	001022				BNE	ER3B	:IF SO: BR
4164	017722	032777	060000	160560	ER3A:	BIT	#60000,@C1	:SEE IF EITHER TRE OR MCPE
4165	017730	001420				BEQ	ER4	:IF NOT: BR
4166	017732	005737	000676			TST	TMFLG	:SEE IF TM TIME
4167	017736	001413				BEQ	ER3B	:IF NOT: BR
4168	017740	017703	160560			MOV	@ER,R3	:GET ERROR REGISTER
4169	017744	032737	000010	000552		BIT	#10,UDES	:SEE IF EVEN PARITY
4170	017752	001402				BEQ	ER3A1	:IF NOT: BR
4171	017754	042703	000100			BIC	#100,R3	:MASK PAR
4172	017760	042703	001000		ER3A1:	BIC	#1000,R3	:MASK FCE
4173	017764	001402				BEQ	ER4	:IF NO ERRORS EXCEPT FCE: BR
4174	017766	005237	021136		ER3B:	INC	CONER	:SET CONT ERROR FLAG
4175	017772	032777	040000	160522	ER4:	BIT	#40000,@DS	:SEE IF DRIVE ERROR
4176	020000	001420				BEQ	ER6	:IF NOT: BR
4177	020002	005737	000676			TST	TMFLG	:SEE IF TAPE MARK TIME
4178	020006	001413				BEQ	ER4A	:IF NOT: BR
4179	020010	017703	160510			MOV	@ER,R3	:GET ER
4180	020014	032737	000010	000552		BIT	#10,UDES	:SEE IF EVEN PARITY
4181	020022	001402				BEQ	ER4A1	:IF NOT: BR
4182	020024	042703	000100			BIC	#100,R3	:MASK PAR
4183	020030	042703	001000		ER4A1:	BIC	#1000,R3	:MASK OUT FCE
4184	020034	001402				BEQ	ER6	:++G &

[illegible]

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

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

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

4356 021150 000000
4357 021152 000000
4358 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.
: *****

4368
4369 021156 032737 000010 000672 FRPRT: BIT #10,MTC1 ;SEE IF WRITE COMMAND
4370 021164 001413 BEQ FREX ;IF SO: BR
4371 021166 032737 000002 000672 BIT #2,MTC1 ;SEE IF REVERSE
4372 021174 001404 BEQ FRO ;IF NOT: BR
4373 021176 012704 025115 MOV #MSG17,R4
4374 021202 104000 TTOUTT ;PRINT R
4375 021204 000403 BR FREX
4376 021206 012704 025112 FRO: MOV #MSG16,R4
4377 021212 104000 TTOUTT ;PRINT F
4378 021214 000207 FREX: RTS PC ;EXIT
4379

```
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409 021216 005037 000642
4410 021222 013777 000550 157270
4411 021230 032777 010000 157264
4412 021236 001026
4413 021240 005237 000642
4414 021244 001371
4415 021246 004737 022730
4416 021252 032737 000010 000672
4417 021260 001004
4418 021262 012704 024754
4419 021266 104000
4420 021270 000405
4421 021272 012704 024761
4422 021276 104000
4423 021300 004737 021156
4424 021304 012704 025335
4425 021310 104000
4426 021312 104006
4427 021314 032777 020000 157200
4428 021322 001411
4429 021324 004737 022730
4430 021330 012704 027314
4431 021334 104000
4432 021336 032777 020000 157156 1$:
4433 021344 001374
4434 021346 022737 000026 000672
4435 021354 001003

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

TAPG: CLR TEMP1
      MOV DVN,@CS ;SET DRIVE NO.
TAPG0: BIT #10000,@DS ;SEE IF HAVE MOL
      BNE TAPG3 ;IF SO: BR
      INC TEMP1 ;SEE IF TIMED OUT
      BNE TAPG0 ;WAIT FOR READY
      JSR PC,PAPRT ;PRINT CYCLE NUMBER
      BIT #10,MTCL ;SEE IF WRITE OP
      BNE TAPG1 ;IF NOT: BR
      MOV #MSG5,R4
      TTOUTT ;PRINT WRITE ERR
      BR TAPG2
TAPG1: MOV #MSG6,R4
      TTOUTT ;PRINT READ ERR
      JSR PC,FRPRT ;PRINT F OR R
TAPG2: MOV #MSG25,R4
      TTOUTT ;PRINT NO MOL ERR
      STOPP
TAPG3: BIT #20000,@DS ;SEE IF PIP RESET
      BEQ TAPG3F ;IF SO: BR
      JSR PC,PAPRT ;PRINT HEADER
      MOV #MSG116,R4
      TTOUTT ;PRINT REWINDING MESSAGE
1$: BIT #20000,@DS
      BNE 1$ ;AWAIT PIP RESET
TAPG3F: CMP #26,MTCL ;SEE IF WRITE TM
      BNE TAPG3A ;IF NOT: BR
```

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

```
4484
4485
4486 ;TTY INTERRUPT HANDLER*****
4487 021630 012777 000340 156746 TTINT: MOV #340,@PSW ;RESET PSW
4488 021636 017746 156750 MOV @TKB,-(SP) ;++G GET CHARACTER
4489 021642 042716 000200 BIC #200,(SP) ;++G STRIP PARITY BIT
4490 021646 122716 000003 CMPB #3,(SP) ;++G SEE IF CONT C
4491 021652 001412 BEQ TTINTO ;IF SO: BR
4492 021654 122716 000007 CMPB #7,(SP) ;++G CHECK FOR CNTL G
4493 021660 001013 BNE RETURN
4494 021662 022737 000176 000606 CMP #SWREG,SWR ;IS SOFTWARE SWITCH REGISTER USED
4495 021670 001007 BNE RETURN ;NO, GET OUT
4496 021672 004737 024444 JSR PC,CNTG ;GO CHANGE SWREG
4497 021676 000404 BR RETURN ;+;G GO TO EXIT
4498
4499 021700 010046 TTINTO: MOV R0,-(SP) ;++G SAVE R0(REC CNTR)
4500 021702 004737 013774 JSR PC,TINP4 ;GO GET STALL VALUES
4501 021706 012600 MOV (SP)+,R0 ;++G RESTORE R0(REC CNTR)
4502 021710 005726 RETURN: TST (SP)+ ;++G POP CHAR OFF STACK
4503 021712 000002 RTI ;RETURN
4504
4505 ;MAG TAPE INTERRUPT HANDLER*****
4506
4507 021714 000240 MTINT: NOP
4508 021716 042777 000037 156610 MTINTA: BIC #37,@MR ;CLEAR MAINT MODE
4509 021724 013716 000662 MOV RTRN,(SP) ;++G GET RETURN ADDRESS
4510 021730 000002 RTI ;++G RETURN
```

```
4511 ;*****
4512 ;AUTO SEQUENCE
4513 ;
4514 ;THIS ROUTINE ,ENTERED VIA STARTING ADDRESS 240
4515 ;WILL EXERCISE ALL AVAILABLE SLAVES ON ALL AVAILABLE
4516 ;DRIVES IN BOTH PE AND NRZ ACCORDING TO THE PRESELECTED
4517 ;TEST PLAN. IF NRZ ONLY, PE TESTING WILL NOT BE ATTEMPTED.
4518 ;*****
4519
4520 021732 012704 027124      ASEQ:  MOV    #MSG108,R4
4521 021736 104000              TTOUTT
4522 021740 012705 000650      MOV    #NRZOF,R5      ;PRINT NRZ ONLY REQUEST
4523 021744 012701 000001      MOV    #1,R1          ;SET ADDRESS OF FLAG
4524 021750 012702 000001      MOV    #1,R2          ;SET SIZE OF ENTRY
4525 021754 012703 000000      MOV    #0,R3          ;SET UPPER LIMIT
4526 021760 004737 023346      MOV    #0,R3          ;SET LOWER LIMIT
4527 021764 012704 026721      JSR     PC,TTR         ;GO GET RESPONSE
4528 021770 104000              MOV    #MSG104,R4
4529 021772 012705 000742      TTOUTT
4530 021776 012701 000001      MOV    #ASEQCF,R5      ;REQUEST CONT OR NOT
4531 022002 012702 000001      MOV    #1,R1          ;SET ADDRESS OF ENTRY
4532 022006 012703 000000      MOV    #1,R2          ;SET SIZE OF ENTRY
4533 022012 004737 023346      MOV    #0,R3          ;SET UPPER LIMIT
4534 022016 005037 000736      JSR     PC,TTR         ;SET LOWER LIMIT
4535 022022 004737 022154      ASEQ0: CLR    ADRVN      ;GO GET INPUT
4536 022026 005737 000042      ASEQ1: JSR     PC,HRDS    ;CLEAR DRV NUM
4537 022032 001404              TST     @#42          ;GO SELECT HARDWARE CONFIGURATION
4538 022034 012737 000001 000742 BEQ     1$          ;AUTO MODE? ++ C.W
4539 022042 000414              BEQ     1$          ;BRANCH - IF NO ++ C.W
4540 022044 012704 026665      1$:  MOV    #1,ASEQCF  ;SET AUTO SEQ FLAG ++ C.W
4541 022050 104000              BR      2$          ;DO AUTO SEQ TESTS ++ C.W
4542 022052 012704 026701      MOV    #MSG101,R4
4543 022056 104000              TTOUTT
4544 022060 013703 000736      MOV    #MSG102,R4      ;PRINT DIVIDER
4545 022064 104002              TTOUTT
4546 022066 012704 026710      MOV    ADRVN,R3      ;PRINT TM02 NUMBER
4547 022072 104000              OCTPP
4548 022074 012700 000746      MOV    #MSG103,R4      ;PRINT TM02
4549 022100 005710              TTOUTT
4550 022102 100403              MOV    #UN1,R0      ;PRINT SLAVE HDR
4551 022104 012003              ASEQ2: TST     (R0)      ;POINT TO START OF SLAVE TABLE
4552 022106 104002              BMI     ASEQ3      ;SEE IF END
4553 022110 000773              MOV    (R0)+,R3    ;IF SO: BR
4554 022112 004737 022360      OCTPP
4555 022116 004737 022554      BR      ASEQ2      ;PRINT SLAVE TABLE
4556 022122 022737 000007 000736 ASEQ3: JSR     PC,AMOD1  ;DO ALL
4557 022130 001403              JSR     PC,AMOD2  ;GO DO MODE 1(NRZ)
4558 022132 005237 000736      ASEQ4: CMP     #7,ADRVN  ;GO DO MODE 2(PE)
4559 022136 000731              BEQ     ASEQX      ;SEE IF DONE ALL DRIVES
4560 022140 005737 000742      INC     ADRVN      ;IF SO: BR
4561 022144 001402              BR      ASEQ1      ;BUMP DRIVE NUMBER
4562 022146 000137 005100      ASEQX: TST     ASEQCF  ;CONTINUE
4563 022152 000000              BEQ     1$          ;CONTINUOUS AUTO SEQUENCE? ++ C.W
                                JMP     TEND        ;BRANCH - IF NO ++ C.W
                                1$:  HALT          ;DO ACT END OF PASS ++ C.W
```

```

4564
4565
4566
4567 022154 005037 005156 HRDS: CLR REOTC ;CLEAR EOT UNIT CNTR
4568 022160 005037 000642 CLR TEMP1
4569 022164 012777 000040 156326 MOV #40,ACS ;INIT
4570 022172 013777 000736 156320 MOV ADRVN,ACS ;SET DRIVE
4571 022200 017701 156332 MOV @DT,R1 ;READ DRIVE TYPE
4572 022204 032777 010000 156306 BIT #10000,ACS ;TEST FOR NON-EXISTANT DRIVE
4573 022212 001403 BEQ HRDS1 ;IF DRIVE AVAIL: BR
4574 022214 005726 HRDS0: TST (SP)+ ;RESET STACK POINTER
4575 022216 000137 022122 JMP ASEQ4 ;GO SEE IF TRIED ALL DRIVES
4576 022222 042701 002007 HRDS1: BIC #2007,R1 ;MASK SLAVE TYPE
4577 022226 022701 140010 CMP #140010,R1 ;++G SEE IF TU16/TE16 TAPE
4578 022232 001370 BNE HRDS0 ;IF NOT: BR
4579 022234 005000 CLR R0
4580 022236 012701 000746 MOV #UN1,R1 ;SET START OF SLAVE TABLE
4581 022242 005737 003040 TST CHNFLG ;++G BRANCH IF NOT IN CHAIN MODE
4582 022246 001410 BEQ HRDS2
4583 022250 122737 000006 000041 CMPB #6,@#41 ;++G BRANCH IF NOT LOADED VIA TMDP
4584 022256 001004 BNE HRDS2
4585 022260 005737 000736 TST ADRVN ;++G BRANCH IF NOT DRIVE 0
4586 022264 001001 BNE HRDS2 ;++G
4587 022266 005200 INC R0 ;++G DO NOT TEST DRIVE 0 SLAVE 0
4588 ;++G IF TMDP CHAIN
4589 022270 010077 156246 HRDS2: MOV R0,AC2 ;SELECT SLAVE
4590 022274 032777 010000 156220 BIT #10000,ADS ;SEE IF SLAVE AVAIL FOR TEST(MOL)
4591 022302 001403 BEQ HRDS3 ;IF NOT: BR
4592 022304 005237 000642 INC TEMP1 ;SET SLAVE FOUND FLAG
4593 022310 010021 MOV R0,(R1)+ ;LOAD SLAVE TABLE
4594 022312 022700 000007 HRDS3: CMP #7,R0 ;SEE IF DONE ALL SLAVES
4595 022316 001402 BEQ HRDS4 ;IF SO: BR
4596 022320 005200 INC R0 ;ELSE BUMP SLAVE NUMBER
4597 022322 000762 BR HRDS2 ;CONTINUE SELECTION
4598 022324 005737 000642 HRDS4: TST TEMP1 ;SEE IF FOUND ANY SLAVES
4599 022330 001731 BEQ HRDS0 ;IF NOT: BR
4600 022332 013737 000642 005156 MOV TEMP1,REOTC ;SET NUMBER OF UNITS
4601 022340 000337 000642 SWAB TEMP1
4602 022344 053737 000642 005156 BIS TEMP1,REOTC ;SET EOT CNTR
4603 022352 012711 177777 MOV #-1,(R1) ;TERMINATE SLAVE TABLE
4604 022356 000207 RTS PC ;RETURN TO SEQ

```

```
4605
4606
4607      ;SUBROUTINE TO SELECT NRZ AUTO TEST MODE*****
4608 022360 005037 000654      AMOD1: CLR      BLCNTR      ;ASSURE BLOCK COUNTER IS 0
4609 022364 012701 000746      MOV      #UN1,R1      ;GET START OF SLAVE TABLE
4610 022370 052721 001700      AMOD1A: BIS      #1700,(R1)+ ;SET ALL SLAVE TO NRZ,NORM,ODD
4611 022374 005111              COM      (R1)
4612 022376 001402              BEQ      AMOD1B      ;IF FILLED ALL SLAVES: BR
4613 022400 005111              COM      (R1)
4614 022402 000772              BR      AMOD1A      ;ELSE DO ALL
4615 022404 005111      AMOD1B: COM      (R1)
4616 022406 004737 005172      JSR      PC,RWDA      ;GO REWIND ALL AVAIL SLAVES
4617 022412 012737 000006 000740      MOV      #6,ABLCNT ;SET NUMBER OF BLOCKS FOR MODE 1
4618 022420 012737 174000 000556      MOV      #-4000,FMCNT ;SET FC = 4000
4619 022426 012737 000100 000554      MOV      #100,RCNT ;SET REC CNTR = 100
4620 022434 013737 000736 000550      MOV      ADRVN,DVN ;SELECT DRIVE
4621 022442 012737 000001 000560      MOV      #1,PATRN ;SELECT PATTERN 1
4622 022450 005037 000564      CLR      TMEX      ;ASSURE NO TMK
4623 022454 005037 000566      CLR      INTRF     ;ASSURE NORMAL READ
4624 022460 004737 003414      JSR      PC,STAUTO ;GO DO AUTO MODE 1
4625 022464 012737 000010 000560      MOV      #10,PATRN ;SELECT PATTERN 10
4626 022472 004737 003414      JSR      PC,STAUTO ;GO DO PATTERN 10
4627 022476 012737 000014 000560      MOV      #14,PATRN ;SELECT PATTERN 14
4628 022504 004737 003414      JSR      PC,STAUTO
4629 022510 005737 000650      TST      NRZOF      ;SEE IF NRZ ONLY
4630 022514 001411      BEQ      AMOD1C      ;IF NOT: BR
4631 022516 012737 177777 000740      MOV      #-1,ABLCNT ;FORCE TO EOT
4632 022524 012737 153624 000624      MOV      #153624,RANBAS
4633 022532 012737 032561 000626      MOV      #32561,RANSAV ;RESET RANDOM DATA BASE
4634 022540 012737 177777 000560      AMOD1C: MOV     #-1,PATRN ;SELECT AUTO RANDOM DATA
4635 022546 004737 003414      JSR      PC,STAUTO
4636 022552 000207      RTS      PC      ;RETURN TO SEQ
```

```

4637
4638
4639
4640 022554 005737 003040          AMOD2: TST      CHNFLG      ;++G BRANCH IF IN CHAIN MODE
4641 022560 001003                    BNE      1$          ;++G
4642 022562 005737 000650          TST      NRZOF      ;SEE IF NRZ ONLY
4643 022566 001057                    BNE      AMOD2X      ;IF SO: BR
4644 022570 005037 000654          1$: CLR      BLCNTR     ;CLEAR BLOCK CNTR
4645 022574 012701 000746          MOV      #UN1,R1    ;SET START OF SLAVE TABLE
4646 022600 042711 001700          AMOD2A: BIC      #1700,(R1) ;CLEAR NRZ
4647 022604 052721 002300          BIS      #2300,(R1)+ ;SET TO PE NORM, ODD
4648 022610 005111                    COM      (R1)      ;SEE IF END OF TABLE
4649 022612 001402                    BEQ      AMOD2B      ;IF SO: BR
4650 022614 005111                    COM      (R1)
4651 022616 000770                    BR       AMOD2A      ;CONTINUE
4652 022620 005111          AMOD2B: COM      (R1)
4653 022622 004737 005172          JSR      PC,RWMDA    ;REWIND ALL SLAVES
4654 022626 012737 000006 000740  MOV      #6,ABLCNT   ;SET AUTO BLOCK COUNT
4655 022634 012737 174000 000556  MOV      #-4000,FMCNT ;SET FC = 4000
4656 022642 012737 000100 000554  MOV      #100,RCNT   ;SET REC CNTR TO 100
4657 022650 012737 000010 000560  MOV      #10,PATRN   ;SELECT PATTERN 10
4658 022656 004737 003414          JSR      PC,STAUTO   ;GO DO AUTO SEQ
4659 022662 012737 000014 000560  MOV      #14,PATRN   ;SELECT PATTERN 14
4660 022670 004737 003414          JSR      PC,STAUTO
4661 022674 012737 000015 000560  MOV      #15,PATRN   ;SELECT PATTERN 15
4662 022702 004737 003414          JSR      PC,STAUTO
4663 022706 012737 177777 000740  MOV      #-1,ABLCNT   ;FORCE TO END OF TAPE
4664 022714 012737 177777 000560  MOV      #-1,PATRN   ;SELECT AUTO RANDOM DATA
4665 022722 004737 003414          JSR      PC,STAUTO
4666 022726 000207          AMOD2X: RTS      PC          ;RETURN TO SEQ
4667
4668

```

CZTU
CZTU

[illegible]

```

4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685 022730 012704 025032 PAPRT: MOV #MSG12,R4
4686 022734 104000 TTOUTT ;PRINT DRIVE HEADER
4687 022736 013703 000550 MOV DVN,R3
4688 022742 104002 OCTPP ;PRINT DRIVE NUMBER
4689 022744 012704 025016 MOV #MSG11,R4
4690 022750 104000 TTOUTT ;PRINT UNIT HEADER
4691 022752 013703 000552 MOV UDES,R3
4692 022756 042703 177770 BIC #177770,R3
4693 022762 104002 OCTPP ;PRINT UNIT NUMBER
4694 022764 012704 026243 MOV #MSG60,R4
4695 022770 104000 TTOUTT ;PRINT DENSITY TAG
4696 022772 013703 000552 MOV UDES,R3
4697 022776 000303 SWAB R3
4698 023000 042703 177770 BIC #177770,R3
4699 023004 104002 OCTPP ;PRINT DENSITY
4700 023006 012704 026247 MOV #MSG61,R4
4701 023012 104000 TTOUTT ;PRINT PARITY TAG
4702 023014 005003 CLR R3
4703 023016 032737 000010 000552 BIT #10,UDES
4704 023024 001402 BEQ PAPRT0
4705 023026 012703 000001 MOV #1,R3
4706 023032 104002 PAPRT0: OCTPP ;PRINT PARITY
4707 023034 012704 026253 MOV #MSG62,R4
4708 023040 104000 TTOUTT ;PRINT FORMAT TAG
4709 023042 013703 000552 MOV UDES,R3
4710 023046 000241 CLC
4711 023050 006003 ROR R3
4712 023052 006003 ROR R3
4713 023054 006003 ROR R3 ;PONTION FORMAT
4714 023056 006003 ROR R3
4715 023060 042703 177760 BIC #177760,R3
4716 023064 104002 OCTPP ;PRINT FORMAT
4717 023066 012704 024773 MOV #MSG8,R4
4718 023072 104000 TTOUTT ;PRINT PATRN TAG
4719 023074 032777 000400 155504 BIT #400,@SWR ;SEE IF RANDOM DATA
4720 023102 001406 BEQ PAPRTB ;IF NOT: BR
4721 023104 012737 000122 000636 PAPRTA: MOV #122,TOB
4722 023112 004737 023662 JSR PC,TOG ;PRINT R
4723 023116 000411 BR PAPRTD
4724 023120 005737 000734 PAPRTB: TST ASEQF ;SEE IF AUTO SEQ

```

Line	Address	Instruction	Comments
4725	023124	001403	
4726	023126	005737	000560
4727	023132	100764	
4728	023134	013703	000560
4729	023140	104002	
4730	023142	012704	025046
4731	023146	104000	
4732	023150	013703	000654
4733	023154	104002	
4734	023156	012704	025054
4735	023162	104000	
4736	023164	010003	
4737	023166	032737	000010 000672
4738	023174	001404	
4739	023176	032737	010000 000562
4740	023204	001016	
4741	023206	013703	000554
4742	023212	005737	000676
4743	023216	001010	
4744	023220	022737	012260 000662
4745	023226	001003	
4746	023230	005737	000712
4747	023234	001402	
4748	023236	160003	
4749	023240	005203	
4750	023242	104002	
4751	023244	012737	000055 000636
4752	023252	004737	023662
4753	023256	013703	000554
4754	023262	104002	
4755	023264	012704	024766
4756	023270	104000	
4757	023272	013703	000556
4758	023276	005303	
4759	023300	005103	
4760	023302	104002	
4761	023304	012737	000001 000664
4762	023312	000207	
4763			

[illegible]

```
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773 023314 063737 000626 000624 RANG: ADD RANSV,RANBAS
4774 023322 063737 000624 000626 ADD RANBAS,RANSV ;GET NEW NUMBER
4775 023330 023701 000626 CMP RANSV,R1 ;SEE IF NUMBER TOO BIG
4776 023334 101367 BHI RANG ;IF SO: BR
4777 023336 020237 000626 CMP R2,RANSV ;SEE IF NUMBER TOO SMALL
4778 023342 101364 BHI RANG ;IF SO: BR
4779 023344 000207 RTS PC ;EXIT
4780
```

```
4781 ;*****
4782 ;TTY ENTRY SUBROUTINE:
4783 ;
4784 ;THIS SUBROUTINE IS USED BY THE TEST CONDITION
4785 ;ENTRY ROUTINE TO READ THE RESPONSE ENTERED
4786 ;AT THE TTY AND CHECK THEM FOR LEGALITY AND
4787 ;LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL
4788 ; (0-7) AND MUST FALL WITHIN THE LIMITS SET BY
4789 ;THE CALLING ROUTINE.
4790 ;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
4791 ;A QUESTION MARK IS TYPED (?) AND THE RESPONSE
4792 ;MAY BE REENTERED.
4793 ;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
4794 ;MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
4795 ;CARRIAGE RETURN
4796 ;*****
4797
4798 023346 005037 000642 TTR: CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
4799 023352 005000 CLR RO
4800 023354 104010 TTR0: TTINN ;GO READ CHARACTER
4801 023356 122737 000015 000640 CMPB #15,TIB ;++G SEE IF CR
4802 023364 001004 BNE TTR1 ;IF NOT: BR
4803 023366 005737 000642 TST TEMP1 ;SEE IF FIRST CHARACTER
4804 023372 001436 BEQ TTR5 ;IF SO: BR
4805 023374 000426 BR TTR2 ;++G ELSE GO LOAD VALUE
4806 023376 122737 000060 000640 TTR1: CMPB #60,TIB ;++G SEE IF CHAR IS LESS THAN 0
4807 023404 101401 BLOS TTR1A ;IF NOT: BR
4808 023406 000431 BR T1NER ;++G ELSE GO TO ERROR
4809 023410 122737 000070 000640 TTR1A: CMPB #70,TIB ;++G SEE IF CHAR IS GREATER THAN 7
4810 023416 101001 BHI TTR1B ;IF NOT: BR
4811 023420 000424 BR T1NER ;++G ELSE GO TO ERROR
4812 023422 005237 000642 TTR1B: INC TEMP1 ;SET FIRST CHARACTER FLAG
4813 023426 006300 ASL RO
4814 023430 006300 ASL RO ;SHIFT 3 LEFT
4815 023432 006300 ASL RO
4816 023434 042737 177770 000640 BIC #177770,TIB ;STRIP ASCII
4817 023442 053700 000640 BIS TIB,RO ;LOAD CHARACTER
4818 023446 005301 DEC R1 ;SEE IF DONE
4819 023450 001341 BNE TTR0 ;IF NOT: BR
4820 023452 020002 TTR2: CMP RO,R2 ;SEE IF EXCEEDED MAXIMUM LIMIT
4821 023454 101401 BLOS TTR3 ;IF NOT: BR
4822 023456 000405 BR T1NER ;++G ELSE GO TO ERROR
4823 023460 020300 TTR3: CMP R3,RO ;SEE IF BELOW MINIMUM LIMIT
4824 023462 101401 BLOS TTR4 ;IF NOT: BR
4825 023464 000402 BR T1NER ;++G ELSE GO TO ERROR
4826 023466 010015 TTR4: MOV RO,(R5) ;LOAD VALUE
4827 023470 000207 TTR5: RTS PC ;EXIT
4828 023472 012704 025755 T1NER: MOV #MSG43,R4
4829 023476 104000 TTOUTT ;PRINT?
4830 023500 162716 000020 SUB #20,(SP) ;RESET SP TO START OF VALUE ROUTINE
4831 023504 000207 RTS PC ;REDO VALUE ENTRY
```

```

4832
4833                                     ;TTY READ SUBROUTINE*****
4834
4835 023506 005277 155076          TTIN:  INC      @TKS
4836 023512 105777 155072          TTIN1: TSTB     @TKS
4837 023516 100375                BPL      TTIN1
4838 023520 017737 155066 000640    MOV      @TKB,TIB
4839 023526 042737 000200 000640    BIC      #200,TIB      ;++G STRIP PARITY BIT
4840 023534 105777 155054          TTIN2: TSTB     @TPS
4841 023540 100375                BPL      TTIN2
4842 023542 113777 000640 155046    MOVB     TIB,@TPB
4843 023550 000207                RTS      PC
4844
4845                                     ;TTY OUTPUT SUBROUTINE*****
4846
4847 023552 112437 000636          TTOUT:  MOVB     (R4)+,TOB
4848 023556 122737 000043 000636    CMPB     #43,TOB
4849 023564 001444                BEQ      TEX
4850 023566 122737 000045 000636    CMPB     #45,TOB
4851 023574 001407                BEQ      TCRLF
4852 023576 122737 000041 000636    CMPB     #41,TOB
4853 023604 001435                BEQ      TBELL      ;DO BELL
4854 023606 004737 023662          JSR      PC,TOG
4855 023612 000757                BR       TTOUT
4856 023614 112737 000015 000636    TCRLF:  MOVB     #15,TOB
4857 023622 004737 023662          JSR      PC,TOG
4858 023626 012703 000006          MOV      #6,R3
4859 023632 005037 000636          TCRLFA: CLR      TOB
4860 023636 004737 023662          JSR      PC,TOG
4861 023642 005303                DEC      R3
4862 023644 001372                BNE      TCRLFA      ;DO FILLERS
4863 023646 112737 000012 000636    MOVB     #12,TOB
4864 023654 004737 023662          JSR      PC,TOG
4865 023660 000734                BR       TTOUT
4866
4867 023662 105777 154726          TOG:    TSTB     @TPS
4868 023666 100375                BPL      TOG
4869 023670 113777 000636 154720    MOVB     TOB,@TPB
4870 023676 000207                RTS      PC
4871 023700 012703 000002          TEX:    MOV      #2,R3
4872 023704 012737 000007 000636    TBELL:  MOV      #7,TOB
4873 023712 004737 023662          TBELA:  JSR      PC,TOG
4874 023716 005303                DEC      R3
4875 023720 001371                BNE      TBELA
4876 023722 000713                BR       TTOUT
4877
4878

```

:OCTAL OUTPUT SUBROUTINE*****

```

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

```

4935
4936
4937
4938 024160 005037 000636
4939 024164 012704 000010
4940 024170 110337 000636
4941 024174 105777 154414
4942 024200 100375
4943 024202 132737 000200 000636
4944 024210 105737 000636
4945 024214 100004
4946 024216 012777 000061 154372
4947 024224 000403
4948 024226 012777 000060 154362
4949 024234 006137 000636
4950 024240 005304
4951 024242 001354
4952 024244 000207
4953 024246 013703 000646
4954 024252 000303
4955 024254 004737 024160
4956 024260 013703 000646
4957 024264 004737 024160
4958 024270 000207
4959
4960
4961
4962 024272 010304
4963 024274 000304
4964 024276 006004
4965 024300 006004
4966 024302 006004
4967 024304 006004
4968 024306 004737 024350
4969 024312 010304
4970 024314 000304
4971 024316 004737 024350
4972 024322 010304
4973 024324 006004
4974 024326 006004
4975 024330 006004
4976 024332 006004
4977 024334 004737 024350
4978 024340 010304
4979 024342 004737 024350
4980 024346 000207
4981 024350 012737 000260 000636
4982 024356 042704 177760
4983 024362 050437 000636
4984 024366 004737 023662
4985 024372 000207
4986
4987
4988
4989
4990

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

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

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

```

```
4991                                     ;LOCATIONS USED:
4992 024374 000000 TEMPST: .WORD 0
4993 024376 000000 COUNT: .WORD 0
4994 024400 000000 RDSW: .WORD 0
4995 024402 022737 000176 000606 CKSWR: CMP #SWREG,SWR ;SOFTWARE SWITCH REG PRESENT
4996 024410 001123 BNE OUT ;NO, GET OUT
4997 024412 105777 154172 TSTB @TKS ;YES, WAIT FOR
4998 024416 100120 BPL OUT ;READY, GET CHARACTER
4999 024420 017737 154166 000640 MOV @TKB,TIB ;AND STRIP OFF
5000 024426 042737 177600 000640 BIC #177600,TIB ;THE GARBAGE
5001 024434 022737 000007 000640 CMP #7,TIB ;IS IT A < G>
5002 024442 001106 BNE OUT
5003 024444 012704 027430 CNTG: MOV #SCNTG,R4
5004 024450 104000 TTOUTT
5005 024452 012704 027434 CNTLU: MOV #MSWR,R4
5006 024456 104000 TTOUTT
5007 024460 017703 154122 MOV @SWR,R3
5008 024464 004737 023732 JSR PC,OCTPE
5009 024470 012704 027443 MOV #SMNEW,R4
5010 024474 104000 TTOUTT
5011 024476 005037 024374 $READ: CLR TEMPST
5012 024502 012737 000007 024376 MOV #7,COUNT
5013 024510 104010 1$: TTINN ;GO READ A CHARACTER
5014 024512 122737 000025 000640 CMPB #25,TIB ;IS IT A U?
5015 024520 001001 BNE 2$ ;BRANCH IF NOT
5016 024522 000753 3$: BR CNTLU ;START OVER
5017 024524 122737 000015 000640 2$: CMPB #15,TIB ;IS IT A <CR>?
5018 024532 001013 BNE 4$ ;BRANCH IF NOT
5019 024534 012737 000200 024400 MOV #200,RDSW
5020 024542 012704 027453 MOV #MCRLF,R4
5021 024546 104000 TTOUTT
5022 024550 022737 000007 024376 CMP #7,COUNT ;WAS IT FIRST CHARACTER
5023 024556 001035 BNE 7$ ;CHANGE SWR IF NOT FIRST ONE
5024 024560 000437 BR OUT ;GET OUT
5025 024562 122737 000060 000640 4$: CMPB #60,TIB
5026 024570 003004 BGT 5$
5027 024572 122737 000067 000640 CMPB #67,TIB
5028 024600 002004 BGE 6$
5029 024602 012704 025755 5$: MOV #MSG43,R4
5030 024606 104000 TTOUTT
5031 024610 000744 BR 3$ ;START OVER IF NOT LEGAL CHARACTER
5032 024612 006337 024374 6$: ASL TEMPST
5033 024616 006337 024374 ASL TEMPST
5034 024622 006337 024374 ASL TEMPST
5035 024626 142737 000060 000640 BICB #60,TIB ;GET NITTY-GRITTY
5036 024634 153737 000640 024374 BISB TIB,TEMPST
5037 024642 005337 024376 DEC COUNT ;ONLY WANT 6 DIGITS
5038 024646 001755 BEQ 5$
5039 024650 000717 BR 1$
5040 024652 013777 024374 153726 7$: MOV TEMPST,@SWR ;CHANGE SWITCH REGISTER CONTENTS
5041 024660 000207 OUT: RTS PC ;RETURN TO BODY OF PROGRAM
5042                                     ;HALT HANDLER*****
5043
5044 024662 000000 STOP: HALT
5045 024664 104004 CKSWRR ;CHECK FOR CONTROL G
5046 024666 000207 RTS PC
```

```
5047
5048
5049
5050 ;TRAP HANDLER*****
5051 024670 016677 000002 153706 TRAP30: MOV 2(6),@PSW ;ADJUST PSW
5052 024676 011666 000002 MOV @SP,2(6) ;PLACE RETURN ADDRESS OVER PSW
5053 024702 162716 000002 SUB #2,@SP ;SUB. 2 FROM RETURN ADDRESS
5054 024706 013646 MOV @6)+,-(6)
5055 024710 062716 120716 ADD #TABLE-104000,@SP ;GET SUBROUTINE STARTING ADDRESS
5056 024714 013607 MOV @SP)+,PC ;GO TO SUBROUTINE
5057 024716 023552 TABLE: TTOUT
5058 024720 023724 OCTP
5059 024722 024402 CKSWR
5060 024724 024662 STOP
5061 024726 023506 TTIN
5062 104000 TTOUTT= 104000
5063 104002 OCTPP= 104002
5064 104004 CKSWRR= 104004
5065 104006 STOPP= 104006
5066 104010 TTINN= 104010
5067
```

Line	Address	Offset	Length	Value	Label	Comment
5068						
5069						
5070						:ERROR MESSAGES*****
5071	024730	042052	020105	043	MSG1:	.ASCII /*DE #/
5072						
5073	024735	045	035507	021440	MSG2:	.ASCII /%G; #/
5074						
5075	024742	041045	020073	043	MSG3:	.ASCII /%B; #/
5076						
5077	024747	045	047103	021440	MSG4:	.ASCII /%CN #/
5078						
5079	024754	053452	020105	043	MSG5:	.ASCII /*WE #/
5080						
5081	024761	052	042522	021440	MSG6:	.ASCII /*RE #/
5082						
5083	024766	051052	020123	043	MSG7:	.ASCII /*RS #/
5084						
5085	024773	052	040520	051124	MSG8:	.ASCII /*PATRN #/
5086	025000	020116	043			
5087	025003	040	047123	020072	MSG9:	.ASCII / SN: #/
5088	025010	043				
5089	025011	052	042523	021440	MSG10:	.ASCII /*SE #/
5090						
5091	025016	051452	040514	042526	MSG11:	.ASCII /*SLAVE NO. #/
5092	025024	047040	027117	021440		
5093						
5094	025032	042045	044522	042526	MSG12:	.ASCII /%DRIVE NO. #/
5095	025040	047040	027117	021440		
5096						
5097	025046	025045	047102	021440	MSG13:	.ASCII /%*BN #/
5098						
5099	025054	051052	020116	043	MSG14:	.ASCII /*RN #/
5100						
5101	025061	045	020041	020040	MSG15:	.ASCII /%! BAD RECORD%%#/
5102	025066	020040	020040	020040		
5103	025074	041040	042101	051040		
5104	025102	041505	051117	022504		
5105	025110	021445				
5106						
5107	025112	043040	043		MSG16:	.ASCII / F#/
5108						
5109	025115	040	021522		MSG17:	.ASCII / R#/
5110						
5111	025120	020041	047505	020124	MSG20:	.ASCII /! EOT NO: #/
5112	025126	047516	020072	043		
5113						
5114						
5115	025133	045	047111	042524	MSG21:	.ASCII /%INTERCHANGE READ = #/
5116	025140	041522	040510	043516		
5117	025146	020105	042522	042101		
5118	025154	036440	021440			
5119						
5120	025160	020445	046111	042514	MSG22:	.ASCII /%!ILLEGAL BOT: HALT%%#/
5121	025166	040507	020114	047502		
5122	025174	035124	044040	046101		
5123	025202	022524	043			

SEQ 0113

5124									
5125	025205	045	051503	020061	MSG23:	.ASCII	/XCS1 #/		
5126	025212	043							
5127									
5128	025213	045	041527	021440	MSG23A:	.ASCII	/XWC #/		
5129									
5130	025220	041045	020101	043	MSG23B:	.ASCII	/XBA #/		
5131									
5132	025225	045	041506	021440	MSG23C:	.ASCII	/XFC #/		
5133									
5134	025232	041445	031123	021440	MSG23D:	.ASCII	/XCS2 #/		
5135									
5136	025240	042045	020123	043	MSG23E:	.ASCII	/XDS #/		
5137									
5138	025245	045	051105	021440	MSG23F:	.ASCII	/XER #/		
5139									
5140	025252	040445	020123	043	MSG23G:	.ASCII	/XAS #/		
5141									
5142	025257	045	045503	021440	MSG23H:	.ASCII	/XCK #/		
5143									
5144	025264	042045	020102	043	MSG23I:	.ASCII	/XDB #/		
5145									
5146	025271	045	051115	021440	MSG23J:	.ASCII	/XMR #/		
5147									
5148	025276	042045	020124	043	MSG23K:	.ASCII	/XDT #/		
5149									
5150	025303	045	041524	021440	MSG23L:	.ASCII	/XTC #/		
5151									
5152	025310	051445	020116	043	MSG23M:	.ASCII	/XSN #/		
5153									
5154	025315	045	047041	020117	MSG24:	.ASCII	/X!NO INTERRUPTX#/		
5155	025322	047111	042524	051122					
5156	025330	050125	022524	043					
5157									
5158	025335	045	047041	020117	MSG25:	.ASCII	/X!NO MOL: HALTX#/		
5159	025342	047515	035114	044040					
5160	025350	046101	022524	043					
5161									
5162	025355	045	051104	050117	MSG26:	.ASCII	/XDROPS: #/		
5163	025362	035123	021440						
5164									
5165	025366	050045	041511	051513	MSG27:	.ASCII	/XPICKS: #/		
5166	025374	020072	043						
5167									
5168	025377	045	043		MSG28:	.ASCII	/X#/		
5169	025401	045	052045	030115	MSG30:	.ASCII	'XX%TM02-TU16/TE16 AUTO SEQUENCE (CZTUAHO)X#'	; ++G	
5170	025406	026462	052524	033061					
5171	025414	052057	030505	020066					
5172	025422	052501	047524	051440					
5173	025430	050505	042525	041516					
5174	025436	020105	041450	052132					
5175	025444	040525	030110	022451					
5176	025452	043							
5177	025453	045	041445	052132	MSG31:	.ASCII	'XXCZTUAHO TM02-TU16/TE16 RELIABXX#'		
5178	025460	040525	030110	052040					
5179	025466	030115	026462	052524					

[illegible]

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

5236	026034	020445	047516	041040	MSG48:	.ASCII	/%.NO BOT ON REWIND: HALT%#/
5237	026042	052117	047440	020116			
5238	026050	042522	044527	042116			
5239	026056	020072	040510	052114			
5240	026064	021445					
5241							
5242	026066	047040	052117	040440	MSG49:	.ASCII	/ NOT AVAIL #/
5243	026074	040526	046111	021440			
5244	026102	044440	046114	043505	MSG50:	.ASCII	/ ILLEGAL DRIVE TYPE #/
5245	026110	046101	042040	044522			
5246	026116	042526	052040	050131			
5247	026124	020105	043				
5248	026127	045	042045	044522	MSG52:	.ASCII	/%%DRIVE NUMBER - #/
5249	026134	042526	047040	046525			
5250	026142	042502	020122	020075			
5251	026150	043					
5252							
5253	026151	045	047506	046522	MSG53:	.ASCII	/%FORMAT = #/
5254	026156	052101	036440	021440			
5255							
5256	026164	053452	020105	046524	MSG54:	.ASCII	/*WE TM#/
5257	026172	043					
5258							
5259	026173	052	042523	052040	MSG55:	.ASCII	/*SE TM#/
5260	026200	021515					
5261							
5262	026202	052040	021515		MSG56:	.ASCII	/ TM#/
5263							
5264	026206	047040	047117	042455	MSG57:	.ASCII	/ NON-EXIST SLAVE#/
5265	026214	044530	052123	051440			
5266	026222	040514	042526	043			
5267	026227	045	051103	020103	MSG58:	.ASCII	/%CRC #/
5268	026234	043					
5269	026235	045	051114	020103	MSG59:	.ASCII	/%LRC #/
5270	026242	043					
5271	026243	052	020104	043	MSG60:	.ASCII	/*D #/
5272	026247	052	020120	043	MSG61:	.ASCII	/*P #/
5273	026253	052	020106	043	MSG62:	.ASCII	/*F #/
5274							
5275	026257	045	047452	044522	MSG64:	.ASCII	/%*ORIGINAL ERROR*#/
5276	026264	044507	040516	020114			
5277	026272	051105	047522	025122			
5278	026300	043					
5279							
5280	026301	045	042522	051124	MSG65:	.ASCII	/%RETRY: #/
5281	026306	035131	021440				
5282							
5283	026312	020452	042523	051040	MSG66:	.ASCII	/*!SE RTRY #/
5284	026320	051124	020131	043			
5285							
5286	026325	052	042441	040522	MSG67:	.ASCII	/*!ERASE#/
5287	026332	042523	043				
5288							
5289	026335	045	042522	042522	MSG68:	.ASCII	/%RREV: #/
5290	026342	035126	021440				
5291	026346	052045	050101	020105	MSG69:	.ASCII	/%TAPE MARK = #/

CZTU
CZTU

RFSO
RPCN
RRHA
RRSO
RSEQ
RSEX
RSF
RSFR
RSFR
RSFR
RSFR
RSFR
RSFO
RSF1
RSR
RSTA
RTCN

RTRN
RTYF
RTY1
RTY2
RTY3
RTY4
RTY5
RTY6
RTY7
RTY8
RWND
RWND
RWND
RWND
RWND
RWND
RWND
RWND
RWND
RWND
SCVF
SERF

SN
SNPG
SNPT
SPFL
STAL

STAL
STAR
STAR
STAR
STAR
STAR
STAR
STAR
STAR

SEQ 0117

5348	026776	051045	053505	047111	MSG16A: .ASCII	/XREWIND TAPE; RESTART AT BLOCK ONE#/
5349	027004	020104	040524	042520		
5350	027012	020073	042522	052123		
5351	027020	051101	020124	052101		
5352	027026	041040	047514	045503		
5353	027034	047440	042516	043		
5354	027041	045	020441	047125	MSG107: .ASCII	/X!!UNRECOVERABLE BAD SPOT/
5355	027046	042522	047503	042526		
5356	027054	040522	046102	020105		
5357	027062	040502	020104	050123		
5358	027070	052117				
5359	027072	041045	042101	051040	.ASCII	/XBAD RECORD LEFT ON TAPEX#/
5360	027100	041505	051117	020104		
5361	027106	042514	052106	047440		
5362	027114	020116	040524	042520		
5363	027122	021445				
5364	027124	047045	055122	047440	MSG108: .ASCII	/XNRZ ONLY: #/
5365	027132	046116	035131	021440		
5366	027140	020452	050041	051517	MSG109: .ASCII	/X*!!POSITION LOST IN RETRY#/
5367	027146	052111	047511	020116		
5368	027154	047514	052123	044440		
5369	027162	020116	042522	051124		
5370	027170	021531				
5371	027172	051445	051525	042520	MSG110: .ASCII	/XSUSPECT BAD TAPE#/
5372	027200	052103	041040	042101		
5373	027206	052040	050101	021505		
5374	027214	051045	050105	040505	MSG111: .ASCII	/XREPEAT: #/
5375	027222	035124	021440			
5376	027226	041040	042101	052040	MSG112: .ASCII	/ BAD TAPE SPOTSX#/
5377	027234	050101	020105	050123		
5378	027242	052117	022523	043		
5379						
5380	027247	045	051440	043117	MSG113: .ASCII	/X SOFT: #/
5381	027254	035124	021440			
5382						
5383	027260	020045	040510	042122	MSG114: .ASCII	/X HARD: #/
5384	027266	020072	043			
5385						
5386	027271	045	020441	040510	MSG115: .ASCII	/X!!HARD READ ERROR#/
5387	027276	042122	051040	040505		
5388	027304	020104	051105	047522		
5389	027312	021522				
5390	027314	020445	047125	052111	MSG116: .ASCII	/X'UNIT REWINDING: TEST WILL START AT BOT#/
5391	027322	051040	053505	047111		
5392	027330	044504	043516	020072		
5393	027336	042524	052123	053440		
5394	027344	046111	020114	052123		
5395	027352	051101	020124	052101		
5396	027360	041040	052117	043		
5397	027365	045	042522	047515	MSG120: .ASCII	/XREMOVE TMDP FROM UNIT UNDER TESTX#/
5398	027372	042526	052040	042115		
5399	027400	020120	051106	046517		
5400	027406	052440	044516	020124		
5401	027414	047125	042504	020122		
5402	027422	042524	052123	021445		
5403	027430	057045	021507		\$CNTG: .ASCII	/X G#/

CZTU
CZTU

STAR
STAR
STAR
STAR
STAR
STAR
STAR
STAR
STAR
STAR
STAR
STAR
STAR
STAR
STAU
STAU
STOP
STOP
STP
STPX
SUSW
SWR

SWRE
TABL
TAPG
TAPG
TAPG
TAPG
TAPG
TAPG
TAPG
TAPG
TAPG
TAPG
TAPG
TAPG
TAPG
TAPG
TBEL
TBEL
TCRL
TCRL
TEMP
TEMP

TEMP

TEMP

TEND

CZTUAMO TMO2-TU16/TE16 RELIAB
CZTUAM.P11 26-APR-78 09:51

MACY11 30A(1052) 16-MAY-78 09:48 PAGE 118

SEQ 0118

5404	027434	051445	051127	020075	SMSWR: .ASCII /%SWR= #/	
5405	027442	043				
5406	027443	040	047040	053505	SMNEW: .ASCII / NEW= #/	
5407	027450	020075	043			
5408	027453	045	043		MCRLF: .ASCII /%#/	
5409						
5410		027456			.EVEN	
5411	027456	000000			WDATA: 0	;WRITE BUFFER
5412						
5413		033464			.=.+4004	
5414	033464	000000			RDATA: 0	;READ BUFFER
5415						
5416		000001			.END	

ABLCMT	000740	1608#	1960	4617*	4631*	4654*	4663*							
ACTLRC	021150	4213*	4306	4350#										
ADRVN	000736	1607#	4534*	4544	4556	4558*	4570	4585	4620					
AMOD1	022360	4554	4608#											
AMOD1A	022370	4610#	4614											
AMOD1B	022404	4612	4615#											
AMOD1C	022540	4630	4634#											
AMOD2	022554	4555	4640#											
AMOD2A	022600	4646#	4651											
AMOD2B	022620	4649	4552#											
AMOD2X	022726	4643	4666#											
APATS	015062	3618	3626#											
AS	000526	1533#	4336	4337*										
ASEQ	021732	3214	4520#											
ASEQCF	000742	1609#	4529	4538*	4560									
ASEQF	000734	1515*	1606#	1869*	1888*	1937	1958	2176	2195	3177	3211	3437	4724	
ASEQX	022140	4557	4560#											
ASEQO	022016	1939	4534#											
ASEQ1	022022	4535#	4559											
ASEQ2	022100	4549#	4553											
ASEQ3	022112	4550	4554#											
ASEQ4	022122	4556#	4575											
BA	000514	1528#	2308*	2361*	2497*	2513*	2698*	2707*	2929*	2938*	3028*	4126	4275	
BAER	021134	4122	4128*	4221	4271	4350#								
BBC	000656	1583#	3754*	3818*	3822*	3910*	3911	3917*	3918*	3919*	3920	3922*	3944*	
BCNT	000710	1596#	4007*	4030*	4033*	4038*	4059*	4062*	4066*	4071*				
BDPP	000716	1599#	3993*	4012*	4013	4041*	4046*	4055*	4057*	4060	4064*			
BD00	001410	1634	1737#											
BD10	001430	1635	1739#											
BD20	001450	1636	1741#											
BD30	001470	1637	1743#											
BD40	001510	1638	1745#											
BD50	001530	1639	1747#											
BD60	001550	1640	1749#											
BD70	001570	1641	1751#											
BKRT	012234	2498	3050#											
BKSP	012056	2622	2635	3025#										
BKTM	012140	3033	3035#											
BKTMO	012202	3039	3043#											
BLCNTR	000654	1582#	1916*	1957*	1960	1962*	2473	4608*	4644*	4732				
BPKP	000720	1600#	3992*	4020*	4021	4040*	4045*	4056*	4058*	4069	4073*			
BP00	001210	1626	1721#											
BP10	001230	1627	1723#											
BP20	001250	1628	1725#											
BP30	001270	1629	1727#											
BP40	001310	1630	1729#											
BP50	001330	1631	1731#											
BP60	001350	1632	1733#											
BP70	001370	1633	1735#											
BTADDR	001030	1645#	2467	2567										
BTFLG	000724	1602#	2110	2116	2120	2133*	2501*	2525*						
BT0V	007160	2480	2524#											
BT0VX	007322	2553	2559#											
BT0V0	007200	2528#	2577											
BT0V1	007210	2531#	2551											
BT0V2	007276	2548	2552#											

[illegible]

DATA4	003002	1838#						
DATA5	003004	1839#						
DATA6	003006	1840#						
DATA7	003010	1841#						
DATBL	002770	1833#	3457					
DATER1	001130	1689#	2044	3839*				
DATR	015102	1975	3441	3637#				
DATRO	015120	3641#	3644					
DATO	014412	1834	3479#					
DATOA	014444	3486#	3494	3498	3501			
DATOB	014452	3487#	3488					
DATOC	014516	3492	3499#					
DATOD	014524	3502#	3510					
DATOE	014534	3504#	3509					
DATOF	014546	3506	3508#					
DAT1	014556	1835	3515#					
DAT1A	014562	3516#	3525	3548	3553	3601		
DAT1B	014566	3517#	3519					
DAT10	014700	1842	3568#					
DAT10A	014712	3571#	3575					
DAT11	014730	1843	3580#					
DAT11A	014736	3582#	3585					
DAT12	014750	1844	3590#					
DAT12A	014760	3592#	3595					
DAT13	014772	1845	3600#					
DAT14	015002	1846	3605#					
DAT14A	015014	3608#	3612					
DAT15	015032	1847	3617#					
DAT15A	015036	3618#	3624					
DAT15B	015046	3620#	3622					
DAT2	014576	1836	3524#					
DAT3	014602	1837	3529#					
DAT3A	014610	3531#	3542					
DAT3B	014614	3532#	3535					
DAT4	014626	1838	3540#					
DAT5	014636	1839	3547#					
DAT6	014644	1840	3552#					
DAT7	014652	1841	3557#					
DAT7A	014666	3560#	3563					
DB	000532	1535#						
DCHK	015544	2783	2978	3754#				
DCHK0	015574	3758	3761#					
DEREV1	001170	1711#	2064	3841*				
DEREX	016622	3912	3933	3935	3943	3951	3954	3956#
DEREX1	016654	3957	3960	3962	3964#			
DERFL	000704	1594#	3755*	3832	3965*			
DERR	016160	3825	3872#					
DERR0	016170	3874#	3963					
DERR0A	016220	3876	3881#					
DERR0B	016252	3887	3890#					
DERR0C	016276	3894	3897#					
DERR0D	016300	3896	3898#					
DERR1	016326	3902	3905#					
DERR2	016330	3904	3906#					
DERR3	016344	3909#						
DERR4	016346	3873	3908	3910#				

DERR4A	016506	3927	3936#																
DERR4B	016550	3921	3944#																
DERR5	016604	3948	3952#																
DERR6	016616	3924	3946	3955#															
DFX	016156	3833	3835	3840	3842#														
DF0	016050	3781	3820#	3829															
DF0A	015744	3791	3793#	3830															
DF0A0	015766	3797	3799#																
DF0A1	016002	3802	3804#																
DF0A2	016016	3807	3809#																
DF0A3	016032	3812	3814#																
DF0A4	016036	3794	3816#																
DF0B	015704	3782#																	
DF0B0	015726	3785	3788#																
DF0C	015664	3774	3778#																
DF0C0	015674	3764	3766	3768	3780#														
DF0D	015650	3770	3775#																
DF0E	015642	3772#	3777																
DF0F	015634	3769#	3773																
DF1	016062	3817	3821	3824#															
DF2	016072	3819	3823	3826#															
DF3	016110	3827	3831#																
DF4	016152	3838	3841#																
DOUT	024160	3898	3906	4938#	4955	4957													
DOUTD	024246	4953#																	
DOUT1	024174	4941#	4942	4951															
DOUT2	024226	4945	4948#																
DOUT3	024234	4947	4949#																
DPC	016776	4004#	4050																
DPCG	017004	4005	4007#																
DPC0	017012	4008#	4042																
DPC0A	017054	4016	4018#																
DPC1	017062	4011	4020#																
DPC1A	017110	4024	4026#																
DPC2	017152	4009	4014	4022	4036#														
DPC2A	017114	4019	4027#																
DPC2B	017136	4031#	4034																
DPC3	017202	4039	4043#																
DPPRT	017250	2015	4018	4026	4052#														
DPPRTX	017404	4072	4075#																
DPPRT0	017320	4060#	4065																
DPPRT1	017344	4063	4066#																
DPPRT2	017360	4069#	4074																
DROP	016766	3999	4002#																
DRPK	016754	3995	3999#																
DRPKF	016666	3824	3986#																
DRP1	001010	1634#	3993	4028	4046	4055													
DRP2	001012	1635#																	
DRP3	001014	1636#																	
DRP4	001016	1637#																	
DRP5	001020	1638#																	
DRP6	001022	1639#																	
DRP7	001024	1640#																	
DRP8	001026	1641#																	
DRVER	021140	2369*	2739*	4185*	4237	4261	4312	4352#											
DS	000522	1531#	1984	1998	2074	2107	2136	2159	2161	2232	2250	2252	2312	2366					

[illegible]

[illegible]

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

[illegible]

[illegible]

[illegible]

[illegible]

START0	003174	1904#	1906															
START1	003454	1951#	2083															
START2	003572	1970	1971	1973#														
START3	003606	1974	1976#															
START4	003622	1956	1977	1979#	1987	1992												
START5	004312	2010	2070#															
START6	004330	2074#	2077															
START7	004360	1981	2000	2075	2082#	2150												
START8	004366	2073	2083#															
STAR1A	003466	1953#																
STAR1B	003506	1954	1957#															
STAR1C	003542	1959	1961	1966#														
STAR4A	003702	1985	1993#															
STAR40	003634	1980	1982#															
STAUT	003136	1516	1884	1887	1892#													
STAUTO	003414	1942#	4624	4626	4628	4635	4658	4660	4662	4665								
STOP	024662	5044#	5060															
STOPP =	104006	1991	2081	2144	2167	2198	2240	2261	2730	3958	4328	4426	4481	5065#				
STP	004030	2013	2015#	2134														
STPX	004276	2014	2067#															
SUSWR	003262	1889	1919#															
SWR	000606	1560#	1878*	1879*	1922	1926*	1929	1969	1973	1976	2009	2070	2213	2302				
		2320	2333	2345	2364	2381	2395	2404	2427	2446	2603	2607	2609	2613				
		2620	2631	2633	2638	2732	2779	2786	2825	2836	2846	2871	2945	2972				
		2998	3002	3035	3872	3913	3956	3961	4015	4023	4188	4229	4326	4331				
		4471	4479	4494	4719	4995	5007	5040*										
SWREG	000176	1492#	1878	1926	1929	2998	4494	4995										
TABLE	024716	5055	5057#															
TAPG	021216	2311	2363	2515	2712	2922	2944	3034	3053	4409#								
TAPG0	021230	4411#	4414															
TAPG1	021272	4417	4421#															
TAPG2	021304	4420	4424#															
TAPG3	021314	4412	4427#															
TAPG3A	021364	4435	4438#															
TAPG3B	021400	4437	4440	4442#														
TAPG3C	021420	4445	4448#															
TAPG3D	021464	4452	4454	4456#														
TAPG3E	021510	4459	4461#															
TAPG3F	021346	4428	4434#															
TAPG4	021536	4466#	4467	4469														
TAPG5	021552	4470#																
TAPG6	021614	4472	4479#															
TAPG7	021624	4480	4482#															
TBELA	023704	4872#	4875															
TBELL	023700	4853	4871#															
TCRLF	023614	4851	4856#															
TCRLFA	023632	4859#	4862															
TEMPST	024374	4992#	5011*	5032*	5033*	5034*	5036*	5040										
TEMP1	000642	1577#	3241	3277	3289	3301	3485*	3491	3495*	3936*	3937*	3938*	3939	3941				
		3986*	3989*	3997*	4002	4047	4409*	4413*	4465*	4466*	4568*	4592*	4598	4600				
		4601*	4602	4798*	4803	4812*												
TEMP2	000644	1578#	3235*	3236	3272*	3284*	3296*	3410*	3414	3496*	3503	3987*	3990*	3998*				
		4003	4048															
TEMP3	000646	1579#	2436*	2444*	2458	2460*	2861*	2869*	2877	2879*	2988*	2994	3988*	4010				
		4049*	4953	4956														
TEND	005100	2179	2186#	4562														

TEX	023676	4849	4870#											
TIB	000640	1576#	4801	4806	4809	4816*	4817	4838*	4839*	4842	4999*	5000*	5001	5014
		5017	5025	5027	5035*	5036								
TINER	023472	4808	4811	4822	4825	4828#								
TINF	000634	1574#	1892*	1897*	1901*	2175*	3166							
TINP	012400	1940	3166#											
TINPA	012410	3167	3169#											
TINPB	012430	3173#	3175											
TINPB0	012614	3207#	3210											
TINPB1	012454	3178	3180#											
TINPC	012642	3182	3212	3215#	3232									
TINPX	014142	3375	3403#											
TINPO	012746	3226	3233#	3245	3259	3310								
TINPOA	013022	3244	3246#											
TINPOB	013040	3242	3249#											
TINPOC	013100	3255#												
TINPOD	013122	3256	3260#											
TINPOE	013146	3261	3266#											
TINP1	013164	3270#												
TINP2	013242	3278	3282#											
TINP2A	013320	3290	3294#											
TINP2B	013376	3302	3306#											
TINP2C	013424	3248	3308	3311#										
TINP3	013444	3315#												
TINP4	013774	3374#	4500											
TKB	000612	1562#	2982	2996	3000*	4488	4838	4999						
TKS	000610	1561#	1951*	2911*	2980	3005*	4835*	4836	4997					
TMEX	000564	1551#	2355	2693	2805	3029	3349	3351	4622*					
TMFLG	000676	1591#	2357*	2401*	2417	2690*	2695*	2721	2734	2781	2793*	2807	2809*	2812*
		2923	2947	2974	4137	4159	4166	4177	4192	4245	4344	4742		
TOB	000636	1575#	1902	4277*	4298*	4308*	4721*	4751*	4847*	4848	4850	4852	4856*	4859*
		4863*	4869	4872*	4920*	4930*	4938*	4940*	4943	4944	4949*	4981*	4983*	
TOG	023662	4278	4299	4309	4722	4752	4854	4857	4860	4864	4867#	4868	4873	4921
		4931	4984											
TPB	000616	1564#	4842*	4869*	4946*	4948*								
TPOS	014152	3281	3293	3305	3409#	3412								
TPOS1	014164	3251	3413#											
TPS	000614	1563#	4840	4867	4941									
TRAP30	024670	1469	5051#											
TSTAL	000576	1556#	2004	2007	2493	2617	2642	3025	3056	3396	3398			
TTIN	023506	4835#	5061											
TTINN =	104010	4800	5013	5066#										
TTINT	021630	1485	4487#											
TTINT0	021700	4491	4499#											
TTIN1	023512	4836#	4837											
TTIN2	023534	4840#	4841											
TTOUT	023552	4847#	4855	4865	4876	5057								
TTOUTT=	104000	1886	1990	2017	2022	2027	2032	2037	2042	2047	2052	2057	2062	2080
		2123	2126	2132	2143	2166	2187	2200	2239	2260	2340	2348	2388	2398
		2430	2432	2442	2449	2451	2455	2537	2541	2550	2565	2573	2583	2729
		2833	2839	2849	2851	2867	2874	2885	2985	3180	3184	3193	3217	3228
		3234	3258	3263	3267	3271	3283	3295	3316	3326	3338	3348	3357	3366
		3377	3386	3395	3879	3882	3892	3900	3916	4053	4068	4243	4250	4254
		4258	4264	4268	4274	4284	4288	4294	4305	4321	4374	4377	4419	4422
		4425	4431	4475	4478	4521	4528	4541	4543	4547	4686	4690	4695	4701
		4708	4718	4731	4735	4756	4829	5004	5006	5010	5021	5030	5062#	

TTR	023346	2993 3373	3191 3384	3200 3393	3222 3402	3240 4526	3276 4533	3288 4798#	3300	3323	3334	3346	3355	3364
TTR0	023354	4800#	4819											
TTR1	023376	4802	4806#											
TTR1A	023410	4807	4809#											
TTR1B	023422	4810	4812#											
TTR2	023452	4805	4820#											
TTR3	023460	4821	4823#											
TTR4	023466	4824	4826#											
TTR5	023470	4804	4827#											
UDES	000552	1546# 2266 3471 4180	1955* 2703 3701 4186	1968* 2751 3719 4198	1983 2757 3722 4314	1993 2763 3757 4341	2105 2934 3763 4444	2157* 2954 3765 4691	2158 2960 3767 4696	2228* 2966 3784 4703	2229 3279* 3793 4709	2248* 3291* 4106	2249 3303* 4145	2265* 3445 4169
UNP	000674	1590# 2063 2242* 2566 3836	1952 2082* 2244* 2741 3991	1963* 2127 2245 2771 4027	1966* 2147 2254* 2854 4044	2018 2155* 2263* 2886 4054	2023 2156 2264 3169*	2028 2168 2330 3243	2033 2170* 2343 3246	2038 2171 2379 3249	2043 2174* 2392 3307	2048 2216 2464 3309*	2053 2217* 2466 3311*	2058 2223 2484 3413
UNIX	000766	1622#												
UN1	000746	1614# 2228 4645	1943 2241*	1945* 2246	1953 2248	1955 2265	1968 3172	1979 3247*	2148* 3250*	2157 3254	2169* 3414*	2172 4548	2224 4580	2226 4609
UN2	000750	1615#												
UN3	000752	1616#												
UN4	000754	1617#												
UN5	000756	1618#												
UN6	000760	1619#												
UN7	000762	1620#												
UN8	000764	1621#												
UPS	000714	1598#	2216*	2263										
VECT	000546	1544#	3194	3196	3201									
WC	000512	1527#	4289	4448*										
WDATA	027456	2308 3779	2361 3928	2368 3940	2372 4119	2513 5411#	2516 2390	3452 2400#	3461	3502	3638	3660	3688	3761
WEX	006320	2304	2356	2365	2378	2382	2390	2400#						
WRITE	005522	2003	2302#											
WRTE	005536	2303	2305#											
WRTSB	006764	2416	2492#											
WRTSB0	007036	2501#	2520											
WRTSB1	007052	2500	2504#	2519										
WRTSB2	007062	2505	2507#											
WRTSB3	007136	2514	2516#											
WRTY	006362	2351	2399	2415#										
WRTYR	006414	2420	2422#											
WRTYTM	006410	2418	2421#											
WRTY0	006370	2416#	2426	2487										
WRTY1	006470	2428	2435#											
WRTY2	006472	2423	2436#											
WRTY2A	006536	2439	2446#											
WRTY2B	006554	2445	2450#											
WRTY3	006620	2447	2459	2462#										
WRTY3A	006636	2463	2466#											
WRTY4	006724	2479	2481#											
WRTY5	006760	2483	2488#											
WRW	006336	2404#												

[illegible]

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

. ABS. 033466 000

ERRORS DETECTED: 0

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

RUN-TIME: 6 12 2 SECONDS

RUN-TIME RATIO: 131/21=6.1

CORE USED: 13K (25 PAGES)

DOCUMENT PAGES: 134