

TM03/TU45

DATA RELIABILITY PROGRAM
CZTURA0

AH-E497A-MC

COPYRIGHT © 75-78
FICHE 1 OF 1

JUL 1978

digital
MADE IN USA

.REM 2

IDENTIFICATION

PRODUCT CODE: AC-E496A-MC
PRODUCT TITLE: CZTURA0 TM03-TU45 DATA RELIABILITY PROGRAM
DATE CREATED: 25 MAY 1978
MAINTAINER: COMPUTER SPECIAL SYSTEMS
AUTHOR: CSS DIAGNOSTICS

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 MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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

COPYRIGHT (©) 1975, 1978 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

PARAGRAPH	SUBJECT	PAGE
1.	ABSTRACT	3
2.	REQUIREMENTS	3
3.	LOADING PROCEDURE	3
4.	STARTING PROCEDURE	4
5.	DATA PATTERNS	11
6.	RANDOMIZATION	12
7.	DYNAMIC PARAMETERS	13
8.	CONSOLE SWITCH	14
9.	ERROR PRINTOUTS	19
10.	STATISTICS PRINTOUT	27
11.	AUTO SEQUENCE	28
12.	TESTING PROCEDURES	30
13.	LISTING	32

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 THE TU45 MAGNETIC ON A MASSBUS THROUGH THE TM03 MAG TAPE CONTROLLER. ANY COMBINATION OF TM03'S & TU45'S UP TO A MAXIMUM OF 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 TM03.

HOWEVER; THE RH AND TM03 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 TM03.

2. REQUIREMENTS (HARDWARE)

- A. ANY PDP-11 PROCESSOR
- B. 8K OF CORE
- C. TELETYPE
- D. TM03 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;
200(8),204(8),210(8),AND 240(8):

- 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(TM03 ADDRESS), SLAVE NUMBER, DENSITY, PARITY, FORMAT, RECORD COUNT, CHARACTER COUNT, PATTERN NUMBER, TAPE MARK AND STALL FOR READ, WRITE, AND TURNAROUND. ALL REPOSSES SHOULD BE MADE IN OCTAL AND WITHIN THE LIMITS OF THE PARAMETER. A QUESTION MARK (?) WILL BE TYPED IF ANY CHARACTER ENTERED IS NOT BETWEEN 0 THRU 7 (OCTAL). THE CHARACTER MAY BE RETYPED FOLLOWING THE QUESTION MARK. IF THE RESPONSE IS NOT WITHIN ITS LIMITS. A QUESTION MARK (?) IS TYPED AND THE ENTIRE RESPONSE MAY BE RENTERED. SOME RESPONSES REQUIRE MORE THAN ONE (1) CHARACTER, BUT NONE REQUIRES MORE THAN SIX (6). RESPONSES OF MORE THAN ONE CHARACTER NEED NOT HAVE LEADING ZEROS AND SHOULD BE TERMINATED BY A CARRIAGE RETURN IF LESS THAN THE MAXIMUM NUMBER OF CHARACTERS IS INPUT.
- B. 204(8): THIS ADDRESS SHOULD BE USED ANYTIME A RESTART OF THE PROGRAM IS NECESSARY AND THE PARAMETERS ENTERED AT THE INITIAL START OF 200(8) NEED NOT BE CHANGED. ALSO NOTE THAT ANY DATA PATTERN WHICH HAD BEEN GENERATED BY SETTING THE RANDOM DATA SWITCH (CONSOLE SWITCH EIGHT) WILL NOT BE OVERWRITTEN AND THEREFORE IS HELD IN CORE FOR USE UNTIL CONSOLE SWITCH EIGHT(8) IS AGAIN SET AND THAT ALL STATISTICS WILL BE RETAINED.
- C. 210(8): THIS ADDRESS IS THE SAME AS USING 204(8) IN THAT THE PREVIOUSLY SET PARAMETERS ARE USED; HOWEVER, THE DATA PATTERN IS RETURNED TO THE FIXED PATTERN ORIGINALLY CALLED FOR AT THE 200(8) START AND ALL STATISTICS ARE CLEARED TO ZERO.
- D. 240(8): THIS IS A SPECIAL ADDRESS WHICH WILL CAUSE THE PROGRAM TO EXECUTE A PREDETERMINED TEST PLAN ON ALL AVAILABLE DRIVES AND SLAVES. THE ONLY INPUT REQUIRED BY THE OPERATOR IS A RESPONSE TO REQUESTS FOR THE RH ADDRESS, VECTOR ADDRESS, CONTINUOUS OPERATION OF THE SEQUENCE, AND NRZ ONLY.
- 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.

**NOTE SEE ALSO SECTION 8-CONSOLE SWITCH SETTINGS

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 TM03) 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.

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. BELOW;
THE SOFTWARE SWR IS INVOKED WITH A SWITCH SETTING OF 100000 (HALT
ON ERROR) IF LOADED VIA ACT11. NO OPERATOR INTERVENTION IS REQUIRED.

**EXCEPTION: IF THIS PROGRAM IS LOADED VIA TMDP CHAIN MODE THE
PROGRAM WILL TEST ALL SLAVES ON THE FIRST AVAILABLE
DRIVE EXCEPT SLAVE 0.

DENSITY: THE DESITY 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. 3 = 800BPI, NRZI
- B. 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 NC MAL (TWO FRAMES PER WORD)
- B. 15 = CORE DUMP (FOUR FRAMES PER WORD)
- C. 16 = PDP-15 OR IBM COMPATABLE (TWO FRAMES PER WORD)
(DATA IS BYTE SWAPPED ON TAPE)

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 FIX⁵ 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. (CZTUTAO) 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. 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 = 200
- B. CHARACTER COUNT = 4000
- C. PATTERN NUMBER 1
- D. TM=1
- E. INTERCHANGE READ = 0
- F. SINGLE PASS = 0
- G. CRC CORRECTION = 0
- H. READ STALL = 10
- I. WRITE STALL = 10
- J. TURN AROUND STALL = 10

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:

TU45 TAPE DRIVE TEST

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) 9 CHAN SN: 0022
DENSITY=(3)
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).

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 CZTUTAO)

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 RESETTNG 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. RESETTNG 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. RESETTNG 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 B 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 CONTROL B WHILE DOING A YOZZLE. A YOZZLE STALL REQUEST WILL BE PRINTED AND SHOULD BE RESPONDED TO WITH THE DESIRED VALUE.

8. CONSOLE SWITCH SETTINGS

CONTROL:

- 1) CONTROL G < G>;
SELECTS SOFTWARE SWR AND ALLOWS USER TO SELECT NEW SWITCHES.
THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW=
WHERE: XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWR.
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 THE SOFTWARE SWR
B) IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWR
CONTENTS WILL NOT BE CHANGED.
- 2) CONTROL A < A>;
ALTERNATES USAGE OF THE SWR BETWEEN THE HARDWARE SWR & SOFTWARE SWR.
- 3) CONTROL B < B>;
SEE SECTION 7 DYNAMIC PARAMETERS
- 4) CONTROL U < U>;
DELETES ALL CHARACTERS TYPED IN RESPONSE TO A REQUEST.

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
0=CONTINUE ON ERROR
- SW14: 1=PRINT READ/WRITE STATISTICS
0=DO NOT PRINT STATS
- SW13: 1=DO NOT CHECK DATA ERRORS
0=CHECK DATA ERRORS
- SW12: 1=DO NOT CHECK WRITE STATUS ERRORS (NOR CLEAR THEM IF THEY DO OCCUR)
0=CHECK WRITE STATUS ERRORS
- SW11: 1=DO NOT CHECK READ STATUS ERRORS (NOR CLEAR THEM IF THEY DO OCCUR)
0=CHECK READ STATUS ERRORS
- SW10: 1=DO NOT PRINT ANY ERRORS (EXCEPT CATASTROPHIC ERRORS)
0=PRINT ALL ERRORS
- SW9: 1=REWIND ALL AVAILABLE TAPES
0=DO NOT REWIND
- SW8: 1=GENERATE RANDOM DATA
0=USED FIXED DATA

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

SW6: 1=GENERATE RANDOM RECORD COUNT
0=USED FIXED RECORD COUNT

SW5: 1=YOZZLE ON CURRENT RECORD
0=DO NOT YOZZLE ON RECORD

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

SW3: 1=DO NOT READ FORWARD
0=READ FORWARD

SW2: 1=DO NOT READ REVERSE
0=READ REVERSE

SW1: 1=READ FORWARD FIRST
0=READ REVERSE FIRST

SW0: 1=DO NOT WRITE
0=WRITE

SWITCH EXPLANATION AND EXAMPLES:

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

EXAMPLES: 0-3

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

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

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

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

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

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

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

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 TM03 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 TM03 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/TU45 CONTROL REGISTER
- WC = RH WORD COUNT
- BA = RH BUS ADDRESS
- FC = TU45 FRAME COUNT
- CS2 = RH CONTROLLER STATUS
- DS = TU45 DRIVE STATUS
- ER = TU45 ERROR REGISTER
- AS = ATTENTION SUMMARY
- CK = TU45 CHECK CHARACTER
- DB = RH DATA BUFFER
- MR = TU45 MAINTENANCE REGISTER
- DT = TU45 DRIVE TYPE
- SN = TU45 SERIAL NUMBER
- TC = TU45 TEST CONTROL
- *F = DATA FORMAT
- *P = PARITY
- *D = DENSITY
- *PATRN = DATA PATTERN NUMBER (R = RANDOM)

EXAMPLE 1: IN THIS EXAMPLE SLAVE 1 ON TM03 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 TM03 1 WAS OPERATING AT 800 BPI IN EVEN PARITY USING THE NINE CHANNEL NORMAL DATA FORMAT. A READ STATUS ERROR WAS DETECTED DURING THE REVERSE READ OF THE TENTH (10) RECORD OF THE 25 RECORDS IN THIS BLOCK (12). THE SIZE OF THE RECORD IS TWENTY (20) FRAMES. THE PRINTOUT INDICATES THE DETECTION OF A VERTICAL PARITY ERROR (VPE: BIT 6 OF ER) AND A CYCLIC REDUNDENCY ERROR (CRC: BIT 15 OF ER). THE CRC CHARACTER, AS RECEIVED, IS NOT AS EXPECTED AND IS PRINTED SHOWING BOTH THE ACTUAL (FIRST) AND THE EXPECTED (LAST).

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

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

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

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

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

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

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

CN 0

G 11111111

B 00000000

CN 1

G 11111111

B 00000000

CN 2

G 11111111

B 00000000

CN 3

G 11111111

B 00000000

CN 4

G 11111111

B 00000000

CN 5

G 11111111

B 00000000

CN 6

G 11111111

B 00000000

CN 7

G 11111111

B 00000000

BAD RECORD

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

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

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

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

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

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

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

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

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

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

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

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

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

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

10. STATISTICS PRINTOUT

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

STATISTICS PRINT EXAMPLE (A HEADER WILL PRECEED THE STATS)

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

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

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

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

11. AUTO SEQUENCE

THE AUTO SEQUENCE (START AT ADDRESS 240) WILL EXECUTE A
PREDETERMINED TEST PLAN ON ALL AVAILABLE SLAVES ON EACH
AVAILABLE TM03. 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:

TU45 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 TM03 AND ITS SLAVES ARE FOUND, A DIVIDER LINE OF
ASTERICKS WILL BE PRINTED FOLLOWED BY A PRINTOUT OF THE
TM03 AND ITS SLAVES BEING TESTED. AS EACH TM03 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 = 0) OR RESTART WITH
THE FIRST AVAILABLE UNIT (AUTO CONT = 1).

AUTO SEQUENCE TEST PLAN:

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

1. NRZ CYCLE:

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

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

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

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

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

12. TESTING PROCEDURES

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

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

TRY IT, YOU'LL LIKE IT.

1310
1311
1312
1313
1314
1315
1316
1317

.LIST BIN,LOC,SEQ
.TITLE TM03/TU45 DATA RELIABILITY PROGRAM
;+B CZTURA0
;21 FEB 1977
;R. BARNES
;REVISED (++) J.G.ADAMS MAY 1977
;+B
;+B

1)INCORRECT RECORD COUNT
STORED WHEN EOT REACHED ON WRITE
2)ADJUST STACK PTR ON BAD TAPE OVFLW

```

1318
1319      .MCALL  .SACT11,.$EOP,$SAVE,$RESTORE,$CHAIN
1320      .NLIST  MC
1321      .LIST   ME
1322      .ENABLE ABS,AMA
1323
1324      ;CONSOLE SWITCHES*****
1325
1326      ;SW15: 1=STOP ON ERROR
1327      :        0=CONTINUE ON ERROR
1328      ;SW14: 1=PRINT READ/WRITE STATS
1329      :        0=DO NOT PRINT STATS
1330      ;SW13: 1=DO NOT CHECK DATA
1331      :        0=CHECK DATA
1332      ;SW12: 1=DO NOT CHECK WRITE ERRORS
1333      :        0=CHECK WRITE ERRORS
1334      ;SW11: 1=DO NOT CHECK READ ERRORS
1335      :        0=CHECK READ ERRORS
1336      ;SW10: 1=DO NOT PRINT ERRORS
1337      :        0=PRINT ERRORS
1338      ;SW9:  1=REWIND TAPE
1339      :        0=DO NOT REWIND
1340      ;SW8:  1=USE RANDOM DATA
1341      :        0=USE FIXED DATA PATTERN
1342      ;SW7:  1=USE RANDOM CHARACTER COUNT
1343      :        0=USE FIXED CHAR COUNT
1344      ;SW6:  1=USE RANDOM RECORD COUNT
1345      :        0=USE FIXED RECORD COUNT
1346      ;SW5:  1=YOZZLE ON CURRENT RECORD
1347      :        0=DO NOT YOZZLE
1348      ;SW4:  1=DO BOTH READ AND WRITE RETRIES
1349      :        0=INHIBIT RETRIES
1350      ;SW3:  1=DO NOT READ FORWARD
1351      :        0=READ FORWARD
1352      ;SW2:  1=DO NOT READ REVERSE
1353      :        0=READ REVERSE
1354      ;SW1:  1=READ FORWARD FIRST
1355      :        0=READ REVERSE FIRST
1356      ;SW0:  1=DO NOT WRITE
1357      :        0=WRITE
1358      ;IF SWR <15::00> = 177777 OR NOT AVAILABLE USE SOFTWARE SWITCH REGISTER

```

[illegible]

```

1405                                ;REGISTER EQUIVS*****
1406
1407                                R0=X0
1408                                R1=X1
1409                                R2=X2
1410                                R3=X3
1411                                R4=X4
1412                                R5=X5
1413                                SP=X6
1414                                PC=X7
1415                                NOP=240
1416
1417                                ;TRAP CATCHERS*****
1418
1419                                .=20
1420 000020 023646                .WORD TTOUT                ;SET IOT TRAP TO TTOUT ROUTINE
1421 000022 000340                .WORD 340                  ;PRIORITY LEVEL 7
1422
1423                                TYPE=IOT                    ;EQUATE TYPE TO AN IOT INSTRUCTION
1424                                .=34
1425 000034 024020                .WORD OCTP                  ;SET TRAP TRAP TO OCTP ROUTINE
1426 000036 000340                .WORD 340                  ;EQUATE TYPOCT TO TRAP INSTRUCTION
1427                                TYPOCT=TRAP
1428
1429                                ;ACT11 HOOK *****
1430                                $SVPC=.                      ;SAVE CURRENT LOCATION CTR
1431                                .=46
1432 000046 005022                .WORD $ENDAD                ;SET LOCATION 46
1433                                .=52
1434 000052 000000                .WORD 0                    ;SET LOCATION 52 = 0
1435                                .=$SVPC                    ;RESTORE LOCATION CTR
1436
1437                                ;TTY INTERRUPT VECTOR*****
1438                                .=60
1439 000060 021504                .WORD TTINT                ;TTY INTERRUPT HANDLER ADDRESS
1440 000062 000340                .WORD 340                  ;PRIORITY LEVEL 7
1441
1442                                ;SOFTWARE SWITCH REGISTER*****
1443                                ;INVOKED IF SWR <15::00> = 177777 OR NOT AVAILABLE
1444                                .=176
1445 000176 000000                SWREG: .WORD 0
1446
1447                                ;START ADDRESS*****
1448                                .=200
1449 000200 000137 003026        JMP START                ;ENTER PARAMETERS VIA TTY
1450
1451                                .=204
1452 000204 000137 003152        JMP STARTC                ;USE FIXED PARAMETERS; HOLD DATA
1453
1454                                .=210
1455 000210 005037 015030        CLR RDFL
1456 000214 000137 003160        JMP STARTA                ;USE FIXED PARAMETERS; NEW DATA
1457
1458                                ;MAG TAPE INTERRUPT VECTOR*****
1459
1460                                .=224

```

TM03/TU45 DATA RELIABILITY PROGRAM
CZTURA.P11 13-JUN-78 10:30

I 3
MACY11 30(1046) 22-JUN-78 09:14 PAGE 34

SEQ 0034

1461	000224	021734	MTINT		;MAG TAPE INTERRUPT HANDLER ADDRESS
1462	000226	000340	340		
1463					
1464					
1465					;AUTO SEQUENCE START*****
1466		000240	. =240		
1467	000240	005237	000736	INC	ASEQF ;SET AUTO SEQUENCE FLAG
1468	000244	000137	003136	JMP	STAUT ;GO TO START OF AUTO SEQUENCE

```

1469                                ;SHORT CONVERSATION RESTART*****
1470
1471                                .=300
1472 000300 005237 014070          INC      SCVFL          ;SET SHORT CONVERSATION FLAG
1473 000304 000137 003026          JMP      START        ;ENTER SHORT PARAMETER LIST
1474
1475                                .=510
1476                                ;TU45 REGISTER EQUIVS*****
1477
1478 000510 172440          C1:      172440
1479 000512 172442          WC:      172442
1480 000514 172444          BA:      172444
1481 000516 172446          FC:      172446
1482 000520 172450          CS:      172450
1483 000522 172452          DS:      172452
1484 000524 172454          ER:      172454
1485 000526 172456          AS:      172456
1486 000530 172460          CC:      172460
1487 000532 172462          DB:      172462
1488 000534 172464          MR:      172464
1489 000536 172466          DT:      172466
1490 000540 172470          SN:      172470
1491 000542 172472          TC:      172472
1492
1493                                ;CONSTANTS*****
1494
1495 000544 172440          REGS:     172440          ;STARTING REGISTER ADDRESS (CS1)
1496 000546 000224          VECT:     224           ;VECTOR ADDRESS (RH INTERRUPT)
1497 000550 000000          DVM:       0           ;DRIVE NUMBER
1498 000552 000000          UDES:     0           ;UNIT DESCRIPTION (PARITY,DENSITY,UNIT,FORMAT)
1499 000554 000200          RCNT:     200          ;RECORD COUNTER
1500 000556 174000          FMCNT:    174000       ;NUMBER OF CHAR (4 - 4000) OCTAL IN TWOS COMPLEMENT
1501 000560 000001          PATRN:     1           ;DATA PATTERN SELECTOR (0 - 15) OCTAL
1502 000562 000002          RDCMD:     2           ;READ COMMAND
1503 000564 000001          TMEX:      1           ;TAPE MARK FLAG: 1=TM 0=NO TM
1504 000566 000000          CRCC:      0           ;CRC CORRECTION FLAG (YES=1,NO=0)
1505 000570 000000          INTRF:     0           ;INTERCHANGE READ 1=YES 0=NO
1506 000572 000000          SPFLG:     0           ;SINGLE PASS 1=YES 0=NO
1507 000574 000010          RSTAL:     10          ;READ STALL
1508 000576 000010          WSTAL:     10          ;WRITE STALL
1509 000600 000010          TSTAL:     10          ;TURN AROUND STAL
1510 000602 002000          YSTAL:     2000        ;YOZZLE STAL
1511 000604 000010          RETRY:     10          ;READ RETRY NUMBER
1512 000606 177776          PSW:      177776       ;PROCESSOR STATUS
1513 000610 177570          SWR:      177570       ;CONSOLE SWITCHES
1514 000612 177560          TKS:      177560       ;TTY READ STATUS REGISTER
1515 000614 177562          TKB:      177562       ;TTY READ BUFFER
1516 000616 177564          TPS:      177564       ;TTY PUNCH STATUS REGISTER
1517 000620 177566          TPB:      177566       ;TTY PUNCH OUTPUT REGISTER
1518 000622 177550          PRS:      177550       ;H/S READER STATUS REGISTER
1519 000624 177552          PRB:      177552       ;H/S READER BUFFER
1520 000626 153624          RANBAS:    153624      ;RANDOM NUMBER GENERATOR BASE
1521 000630 032561          RANSAB:    032561      ;RANDOM NUMBER BUFFER
1522 000632 000200          RCSAV:     200         ;RECORD COUNT SAVE
1523 000634 174000          FCSAV:    174000       ;FRAME COUNT SAVE

```

```

1524
1525
1526
1527 000636 000000      TINF: 0          ;TTY ENTRY FLAG
1528 000640      STFLG:          ;
1529 000640 000000      TOB: 0          ;TTY OUTPUT BUFFER
1530 000642 000000      TIB: 0          ;TTY INPUT BUFFER
1531 000644 000000      TEMP1: 0        ;TEMP STORAGE
1532 000646 000000      TEMP2: 0        ;TEMP STORAGE
1533 000650 000000      TEMP3: 0        ;TEMP STORAGE
1534 000652 000000      NRZOF: 0        ;NRZ ONLY FLAG
1535 000654 000000      EMADDR: 0       ;ERROR MSG ADDRESS STORAGE
1536 000656 000000      BLCNTR: 0       ;BLOCK COUNTER
1537 000660 000000      BBC: 0          ;BAD RECORD COUNTER
1538 000662 000000      EOTREC: 0       ;EOT FLAG
1539 000664 000000      RTRN: 0         ;INTERRUPT RETURN STORAGE
1540 000666 000000      HDRFL: 0        ;HEADER FLAG
1541 000670 000000      STAL: 0         ;DELAY STORAGE
1542 000672 000000      PFLG: 0        ;PRINT FLAG
1543 000674 000000      MTC1: 0         ;MAG TAPE CONT REGISTER BUFFER
1544 000676 000000      UNP: 0          ;UNIT TABLE POINTER
1545 000700 000000      TMFLG: 0        ;TAPE MARK FLAG
1546 000702 000000      RPCNT: 0        ;REPEAT COUNTER
1547 000704 000000      RTCNT: 0        ;RETRY COUNTER
1548 000706 000000      DERFL: 0        ;DATA ERROR FLAG
1549 000710 000000      SERFL: 0        ;STATUS ERROR FLAG
1550 000712 000000      BCNT: 0         ;BIT COUNTER
1551 000714 000000      RTYFL: 0        ;RETRY FLAG
1552 000716 000000      UPS: 0          ;UNIT POINTER SAVE
1553 000720 000000      BDPP: 0         ;BITS DROPPED POINTER
1554 000722 000000      BPKP: 0         ;BITS PICKED POINTER
1555 000724 000000      ERSV: 0         ;ERROR SAVE LOC
1556 000726 000000      BTFLG: 0        ;BAD TAPE FLAG
1557 000730 000000      BTSTF: 0        ;STATISTIC PRINT FLAG
1558 000732 000000      BTPT: 0         ;BAD TAPE POINTER
1559 000734 000000      ERTFL: 0        ;ERASE FLAG
1560 000736      ENDFLG:          ;
1561 000736 000000      ASEQF: 0         ;AUTO SEQ FLAG
1562 000740 000000      ADRVN: 0         ;UTO SEQ DRIVE NUMBER
1563 000742 000000      ABLCNT: 0       ;AUTO BLOCK COUNTER
1564 000744 000001      ASEQCF: 1       ;AUTO SEQ CONTINUOUS FLAG

```

```

1565
1566                                     ;UNIT ORDER AND DESCRIPTION TABLE *****
1567
1568 000746 000000 UN1: 0 ;THIS TABLE IS LOADED
1569 000750 000000 UN2: 0 ;WITH UNIT NUMBERS AND
1570 000752 000000 UN3: 0 ;THEIR DESCRIPTIONS IN
1571 000754 000000 UN4: 0 ;THE ORDER THAT THEY
1572 000756 000000 UN5: 0 ;WILL BE TESTED
1573 000760 000000 UN6: 0
1574 000762 000000 UN7: 0
1575 000764 000000 UN8: 0
1576 000766 177777 UNX: -1
1577
1578                                     ;UNIT DROPS AND PICKS POINTERS*****
1579
1580 000770 001210 PIK1: BP00
1581 000772 001230 PIK2: BP10
1582 000774 001250 PIK3: BP20
1583 000776 001270 PIK4: BP30
1584 001000 001310 PIK5: BP40
1585 001002 001330 PIK6: BP50
1586 001004 001350 PIK7: BP60
1587 001006 001370 PIK8: BP70
1588 001010 001410 DRP1: BD00
1589 001012 001430 DRP2: BD10
1590 001014 001450 DRP3: BD20
1591 001016 001470 DRP4: BD30
1592 001020 001510 DRP5: BD40
1593 001022 001530 DRP6: BD50
1594 001024 001550 DRP7: BD60
1595 001026 001570 DRP8: BD70
1596
1597                                     ;UNIT BAD TAPE POINTERS*****
1598
1599 001030 001610 BTADDR: BT00
1600 001032 001714 BT01
1601 001034 002020 BT02
1602 001036 002124 BT03
1603 001040 002230 BT04
1604 001042 002334 BT05
1605 001044 002440 BT06
1606 001046 002544 BT07
1607
1608                                     ;UNIT WRITE RETRY COUNTER*****
1609
1610 ;SET START OF STATISTICS TABLE
1611 001050 STTBL:
1612 001050 000000 RTY1: 0
1613 001052 000000 RTY2: 0
1614 001054 000000 RTY3: 0
1615 001056 000000 RTY4: 0
1616 001060 000000 RTY5: 0
1617 001062 000000 RTY6: 0
1618 001064 000000 RTY7: 0
1619 001066 000000 RTY8: 0
1620

```

```

1621                                     ;UNIT WRITE ERRORS*****
1622
1623 001070 000000 WTER1: 0
1624 001072 000000 WTER2: 0
1625 001074 000000 WTER3: 0
1626 001076 000000 WTER4: 0
1627 001100 000000 WTER5: 0
1628 001102 000000 WTER6: 0
1629 001104 000000 WTER7: 0
1630 001106 000000 WTER8: 0
1631
1632                                     ;UNIT READ FORWARD ERRORS*****
1633
1634 001110 000000 RDER1: 0
1635 001112 000000 RDER2: 0
1636 001114 000000 RDER3: 0
1637 001116 000000 RDER4: 0
1638 001120 000000 RDER5: 0
1639 001122 000000 RDER6: 0
1640 001124 000000 RDER7: 0
1641 001126 000000 RDER8: 0
1642
1643                                     ;UNIT DATA ERRORS FORWARD*****
1644
1645 001130 000000 DATER1: 0
1646 001132 000000 0
1647 001134 000000 0
1648 001136 000000 0
1649 001140 000000 0
1650 001142 000000 0
1651 001144 000000 0
1652 001146 000000 0
1653
1654                                     ;UNIT READ REVERSE ERRORS*****
1655
1656 001150 000000 RDERR1: 0
1657 001152 000000 0
1658 001154 000000 0
1659 001156 000000 0
1660 001160 000000 0
1661 001162 000000 0
1662 001164 000000 0
1663 001166 000000 0
1664
1665                                     ;UNIT DATA ERRORS REVERSE*****
1666
1667 001170 000000 DEREV1: 0
1668 001172 000000 0
1669 001174 000000 0
1670 001176 000000 0
1671 001200 000000 0
1672 001202 000000 0
1673 001204 000000 0
1674 001206 000000 0

```

```
1675 ;DROPS + PICKS PER CHANNEL PER UNIT*****
1676
1677 001210 000000 BP00: 0
1678 001230 000000 .=.+16
1679 001230 000000 BP10: 0
1680 001250 000000 .=.+16
1681 001250 000000 BP20: 0
1682 001270 000000 .=.+16
1683 001270 000000 BP30: 0
1684 001310 000000 .=.+16
1685 001310 000000 BP40: 0
1686 001330 000000 .=.+16
1687 001330 000000 BP50: 0
1688 001350 000000 .=.+16
1689 001350 000000 BP60: 0
1690 001370 000000 .=.+16
1691 001370 000000 BP70: 0
1692 001410 000000 .=.+16
1693 001410 000000 BD00: 0
1694 001430 000000 .=.+16
1695 001430 000000 BD10: 0
1696 001450 000000 .=.+16
1697 001450 000000 BD20: 0
1698 001470 000000 .=.+16
1699 001470 000000 BD30: 0
1700 001510 000000 .=.+16
1701 001510 000000 BD40: 0
1702 001530 000000 .=.+16
1703 001530 000000 BD50: 0
1704 001550 000000 .=.+16
1705 001550 000000 BD60: 0
1706 001570 000000 .=.+16
1707 001570 000000 BD70: 0
1708 001610 .=.+16
1709
1710
```

```

1711
1712                ;UNIT BAD TAPE COUNTER:16 PER SLAVE*****
1713
1714 001610 000000    BT00: 0
1715                .=. +102
1716 001714 000000    BT01: 0
1717                .=. +102
1718 002020 000000    BT02: 0
1719                .=. +102
1720 002124 000000    BT03: 0
1721                .=. +102
1722 002230 000000    BT04: 0
1723                .=. +102
1724 002334 000000    BT05: 0
1725                .=. +102
1726 002440 000000    BT06: 0
1727                .=. +102
1728 002544 000000    BT07: 0
1729                .=. +102
1730
1731                ;UNIT END OF TAPE COUNTERS 1 PER SLAVE*****
1732
1733 002650 000000    EOTCO: 0
1734 002652 000000        0
1735 002654 000000        0
1736 002656 000000        0
1737 002660 000000        0
1738 002662 000000        0
1739 002664 000000        0
1740 002666 000000        0
1741
1742                ;UNIT READ FORWARD SOFT ERROR*****
1743
1744 002670 000000    RFSOFT: 0
1745 002672 000000        0
1746 002674 000000        0
1747 002676 000000        0
1748 002700 000000        0
1749 002702 000000        0
1750 002704 000000        0
1751 002706 000000        0
1752
1753                ;UNIT READ REVERSE SOFT ERROR*****
1754
1755 002710 000000    RRSOFT: 0
1756 002712 000000        0
1757 002714 000000        0
1758 002716 000000        0
1759 002720 000000        0
1760 002722 000000        0
1761 002724 000000        0
1762 002726 000000        0
1763

```

```

1764
1765           ;UNIT READ FORWARD HARD ERROR*****
1766
1767 002730 000000 RFHARD: 0
1768 002732 000000      0
1769 002734 000000      0
1770 002736 000000      0
1771 002740 000000      0
1772 002742 000000      0
1773 002744 000000      0
1774 002746 000000      0
1775
1776           ;UNIT READ REVERSE HARD ERROR*****
1777
1778 002750 000000 RRHARD: 0
1779 002752 000000      0
1780 002754 000000      0
1781 002756 000000      0
1782 002760 000000      0
1783 002762 000000      0
1784 002764 000000      0
1785 002766 000000      0
1786           ;SET END OF STATISTICS TABLE
1787 002770 ENDTBL:
1788
1789           ;DATA PATTERN GENERATORS*****
1790
1791 002770 002770 DATBL: .           ;ENTRY TABLE
1792 002772 014302 DATA0: DAT0       ;EXTERNAL INPUT FROM H/S READER(SEE CZTUTAO)
1793 002774 014442 DATA1: DAT1       ;ALL ONES
1794 002776 014462 DATA2: DAT2       ;ALL ZEROS
1795 003000 014466 DATA3: DAT3       ;WALKING ONE
1796 003002 014512 DATA4: DAT4       ;WALKING ZERO
1797 003004 014522 DATA5: DAT5       ;ALTERNATING ONE/ZERO
1798 003006 014530 DATA6: DAT6       ;ALTERNATING ZERO/ONE
1799 003010 014536 DATA7: DAT7       ;ALTERNATING ONE/ZERO IN ALTERNATING CHARACTERS
1800 003012 014564 DATA10: DAT10      ;WALKING ONE/ALL ONE IN ALTERNATING CHARACTERS
1801 003014 014614 DATA11: DAT11      ;ALL BITS 0-377
1802 003016 014634 DATA12: DAT12      ;ALL BITS 377-0
1803 003020 014656 DATA13: DAT13      ;ALTERNATING CHARACTERS 0 AND 377
1804 003022 014666 DATA14: DAT14      ;WALKING ZERO/ALL ZERO IN ALTERNATING CHARACTERS
1805 003024 014716 DATA15: DAT15      ;AUTO SEQUENCE PATTERN 0,0,-1,-1,-1,0,0
1806

```

```

1807          .EVEN
1808          ;*****
1809          ;PROGRAM START AND SEQUENCE FORMATTER:
1810          ;
1811          ;THIS ROUTINE IS USED TO PERFORM ALL HOUSEKEEPING,
1812          ;DECIDE WHICH TRANSPORT TO TEST AND ITS AVAILABILITY,
1813          ;LOAD THE WRITE BUFFER WITH THE SELECTED DATA PATTERN,
1814          ;GENERATE ANY RANDOM NUMBER AND THEN EXECUTE
1815          ;THE TEST CYCLE REQUESTED BY THE SWITCH SETTING.
1816          ;AT THE END OF THE TEST CYCLE THE NEXT UNIT IS SELECTED
1817          ;AND CHECKED FOR AVAILABILITY AND THE TEST CYCLE IS
1818          ;EXECUTED ON IT.
1819          ;THE READ WRITE STATS MAY BE PRINTED AT THE END OF
1820          ;EACH TEST CYCLE VIA CONSOLE SWITCH FOURTEEN (14).
1821          ;*****
1822
1823
1824          ;START 200, & 300*****
1825          003026 012706 000500          START:  MOV    #500,SP          ;SET STACK PTR
1826          003032 005037 000736          CLR     ASEQF          ;CLEAR AUTO SEQUENCE FLAG
1827          003036 005027          CLR     (PC)+          ;CLEAR CHAIN INDICATOR
1828          003040 000000          CHNFLG: .WORD 0          ;CHAIN MODE INDICATOR
1829          ;1/0 = CHAIN/NOT CHAIN MODE
1830          003042 022737 005022 000042          CMP     #SENDAD,a#42          ;BRANCH IF LOADED VIA ACT11 CHAIN MODE
1831          003050 001404          BEQ     50$
1832          003052 005737 000042          TST     a#42          ;BRANCH IF IN DUMP MODE
1833          003056 001413          BEQ     52$
1834          003060 000406          BR      51$
1835          003062 012737 000176 000610 50$:  MOV     #SWREG,SWR          ;INVOKE SOFTWARE SWR
1836          003070 012777 100000 175512          MOV     #100000,aSWR          ;WITH HALT ON ERROR SET
1837          003076 005237 003040          51$:  INC     CHNFLG          ;SET CHNFLG = CHAIN MODE
1838          003102 000137 003126          JMP     3$          ;GO TO CHAIN ADDRESS
1839          003106          52$:
1840          003106 122737 000006 000041          CMPB    #6,a#41          ;BRANCH IF LOADED VIA TMDP
1841          003114 001010          BNE     STAUT
1842          003116 012704 027251          MOV     #MSG120,R4          ;ADVISE USER TO REMOVE TMDP FROM SLAVE
1843          003122 000004          TYPE
1844          003124 000404          BR      STAUT
1845          003126 005237 000736          3$:  INC     ASEQF          ;SET AUTO SEQUENCE FLAG
1846          003132 000137 022004          JMP     ASEQ0          ;GO TO AUTO SEQUENCER
1847
1848
1849          003136 012737 000001 000636          ;START 240*****
1850          003144 005037 015030          STAUT:  MOV     #1,TINF          ;SET TTY ENTRY FLAG
1851          003150 000405          CLR     RDFL          ;CLEAR RANDOM DATA FLAG
1852          BR      STARTB
1853
1854          003152 005037 000636          ;START 204*****
1855          003156 000432          STARTC: CLR     TINF          ;CLEAR TTY INPUT FLAG
1856          BR      STARTD
1857
1858          003160 005037 000636          ;START 210*****
1859          003164 012700 000640          STARTA: CLR     TINF          ;CLEAR TTY ENTRY FLAG
1860          003170 012701 000076          STARTB: MOV     #STFLG,R0          ;GET STARTING ADDRESS OF FLAGS
1861          003174 105020          1$:  MOV     #ENDFLG-STFLG,R1
1862          003176 005301          CLR     (R0)+          ;CLEAR FLAGS AND COUNTERS
          DEC     R1

```

```

1863 003200 001375          BNE      1$
1864 003202 012706 000500    MOV      #500,SP          ;SET STACK POINTER
1865 003206 004737 004276    JSR      PC,RANSET        ;GO RESET RANDOM BASE
1866 003212 012700 001050    MOV      #STTBL,R0        ;GET STARTING ADDRESS OF STAT TABLE
1867 003216 012701 001720    MOV      #ENDTBL-STTBL,R1      ;AND # OF BYTES IN TABLE
1868 003222 105020          2$:    CLRB      (R0)+          ;CLEAR STATISTIC COUNTERS
1869 003224 005301          DEC      R1
1870 003226 001375          BNE      2$
1871 003230 012737 177777 014276    MOV      #-1,PATS          ;PRESET PATTERN
1872 003236 012737 000001 000656    STARTE: MOV      #1,BLCNTR          ;PRESET BLOCK COUNTER
1873 003244 013746 000004          STARTD: MOV      @#4,-(SP)          ;SAVE ERROR TRAP VECTOR
1874 003250 013746 000006          MOV      @#6,-(SP)
1875 003254 022737 000176 000610    CMP      #SWREG,SWR          ;BRANCH IF SOFTWARE SWR
1876 003262 001413          BEQ      2$          ;ALREADY SELECTED
1877 003264 012737 003310 000004    MOV      #1$,@#4          ;SET TIMEOUT TRAP TO 1$ BELOW
1878 003272 005037 000006          CLR      @#6
1879 003276 022777 177777 175304    CMP      #177777,@SWR          ;BRANCH IF SWR = 177777 TRAP
1880 003304 001402          BEQ      2$          ;IF NOT AVAIL (1$) OTHERWISE
1881 003306 000404          BR       3$          ;GO TO 3$
1882 003310 022626          1$:    CMP      (SP)+,(SP)+          ;RESET STACK
1883 003312 012737 000176 000610    2$:    MOV      #SWREG,SWR          ;SET SWR = SOFTWARE SWR
1884 003320 012637 000006          3$:    MOV      (SP)+,@#6          ;RESTORE ERROR TRAP
1885 003324 012637 000004          MOV      (SP)+,@#4
1886 003330 012706 000500          MOV      #500,SP
1887 003334 004737 012220          JSR      PC,TINP          ;GO GET PARAMETERS FROM TTY
1888 003340 012777 000040 175152    MOV      #40,@CS          ;INITIALIZE
1889 003346 005000          STAUTO: CLR      R0          ;POINT TO FIRST ENTRY
1890 003350 022760 177777 000746    1$:    CMP      #-1,UN1(R0)          ;BRANCH IF LAST ENTRY
1891 003356 001406          BEQ      2$
1892 003360 042760 100000 000746    BIC      #100000,UN1(R0)      ;CLEAR EOT FLAG
1893 003366 062700 000002          ADD      #2,R0          ;POINT TO NEXT UNIT ENTRY
1894 003372 000766          BR       1$          ;CONTINUE CLEARING
1895 003374 013703 005054          2$:    MOV      REOTC,R3
1896 003400 000303          SWAB      R3
1897 003402 110337 005054          MOV      R3,REOTC          ;RESTORE EOT CNTR
1898 003406 012777 000100 175176    START1: MOV      #100,@TKS          ;SET KEYBOARD IE BIT
1899 003414 013700 000676          MOV      UNP,R0          ;R0 = UNIT TABLE POINTER
1900 003420 022760 177777 000746    STAR1A: CMP      #-1,UN1(R0)          ;BRANCH IF LAST ENTRY
1901 003426 001404          BEQ      STAR1B
1902 003430 016037 000746 000552    MOV      UN1(R0),UDES          ;LOAD NEXT UNIT DESCRIPTION
1903 003436 000446          BR       START4
1904 003440 005237 000656          STAR1B: INC      BLCNTR          ;BUMP BLOCK COUNTER
1905 003444 005737 000736          TST      ASEQF          ;SEE IF AUTO SEQ
1906 003450 001411          BEQ      STAR1C          ;IF NOT: BR
1907 003452 023737 000656 000742    CMP      BLCNTR,ABLCNT          ;SEE IF DONE SEQ
1908 003460 001005          BNE      STAR1C          ;IF NOT: BR
1909 003462 005037 000656          CLR      BLCNTR          ;RESET BLOCK CNTR
1910 003466 005037 000676          CLR      UNP          ;RESET UNIT POINTER
1911 003472 000207          RTS      PC          ;RETURN TO AUTO SEQ
1912 003474 005037 000676          STAR1C: CLR      UNP
1913 003500 005000          CLR      R0
1914 003502 016037 000746 000552    MOV      UN1(R0),UDES          ;LOAD FIRST UNIT DESCRIPTION
1915 003510 032777 000200 175072    BIT      #200,@SWR          ;SEE IF RANDOM RECORD SIZE
1916 003516 001402          BEQ      START2          ;IF NOT: BR
1917 003520 004737 012134          JSR      PC,CCNTR          ;GO GENERATE RANDOM RECORD SIZE
1918 003524 032777 000400 175056    START2: BIT      #400,@SWR          ;SEE IF RANDOM DATA

```

```

1919 003532 001402          BEQ      START3      ; IF NOT: BR
1920 003534 004737 014766    JSR      PC,DATR      ; GO GENERATE RANDOM DATA
1921 003540 032777 000100 175042 START3: BIT      #100,@SWR ; SEE IF RANDOM RECORD COUNT
1922 003546 001402          BEQ      START4      ; IF NOT: BR
1923 003550 004737 012174    JSR      PC,RCNTR     ; GO GENERATE RANDOM RECORD COUNT
1924 003554 005760 000746    START4: TST      UN1(R0) ; SEE IF REACHED EOT
1925 003560 100002          BPL      STAR40      ; IF NOT: BR
1926 003562 000137 004264    JMP      START7      ; ELSE GO TO NEXT UNIT
1927 003566 013777 000550 174724 STAR40: MOV     DVN,@CS ; SET DRIVE NUMBER
1928 003574 013777 000552 174740 MOV     UDES,@TC ; SET UNIT NUMBER
1929 003602 105777 174714    TSTB     @DS ; SEE IF UNIT AVAIL
1930 003606 100412          BMI      STAR4A      ; IF SO: BR
1931 003610 005337 000670    DEC      STAL
1932 003614 001357          BNE      START4      ; AWAIT TUR
1933 003616 004737 022602    JSR      PC,PAPRT    ; PRINT HEADER
1934 003622 012704 025750    MOV      #MSG49,R4
1935 003626 000004          TYPE
1936 003630 000000          HALT
1937 003632 000750          BR      START4
1938 003634 004737 014116    STAR4A: JSR      PC,DSUP    ; GO SET UP WRITE DATA
1939 003640 004737 005426    JSR      PC,INIT    ; INIT SLAVE
1940 003644 004737 005056    JSR      PC,RWND    ; REWIND
1941 003650 004737 005534    JSR      PC,WRITE   ; WRITE
1942 003654 013737 000600 000670 MOV      TSTAL,STAL ; SET TURN AROUND DELAY
1943 003662 004737 012124    JSR      PC,STALL   ; DELAY
1944 003666 004737 007424    JSR      PC,RSEQ    ; GO TO READ SEQUENCER
1945 003672 013737 000600 000670 MOV      TSTAL,STAL ; SET TURN AROUND DELAY
1946 003700 004737 012124    JSR      PC,STALL   ; DELAY
1947 003704 032777 040000 174676 BIT      #40000,@SWR ; SEE IF SHOULD PRINT STATISTICS
1948 003712 001541          BEQ      START5      ; IF NOT: BR
1949 003714 012700 000001    MOV      #1,R0 ; SET RECORD COUNTER TO 1
1950 003720 004737 022602    JSR      PC,PAPRT    ; PRINT CYCLE NUMBER
1951 003724 004737 003734    JSR      PC,STP
1952 003730 000137 004202    JMP      STPX
1953 003734 004737 017106    STP:    JSR      PC,DPPRT    ; PRINT DROPS AND PICKS
1954 003740 012704 026162    MOV      #MSG65,R4
1955 003744 000004          TYPE
1956 003746 013704 000676    MOV      UNP,R4 ; TYPE MSG
1957 003752 016403 001050    MOV      RTY1(R4),R3
1958 003756 104400          TYPOCT
1959 003760 012704 026333    MOV      #MSG73,R4 ; PRINT RETRIES
1960 003764 000004          TYPE
1961 003766 013704 000676    MOV      UNP,R4 ; TYPE MSG
1962 003772 016403 001070    MOV      WTER1(R4),R3
1963 003776 104400          TYPOCT
1964 004000 012704 026322    MOV      #MSG72,R4 ; PRINT WRITE ERRORS
1965 004004 000004          TYPE
1966 004006 013704 000676    MOV      UNP,R4 ; TYPE MSG
1967 004012 016403 001110    MOV      RDER1(R4),R3
1968 004016 104400          TYPOCT
1969 004020 012704 027126    MOV      #MSG113,R4 ; PRINT READ FORWARD ERRORS
1970 004024 000004          TYPE
1971 004026 013704 000676    MOV      UNP,R4 ; TYPE MSG
1972 004032 016403 002670    MOV      RFSOFT(R4),R3
1973 004036 104400          TYPOCT
1974 004040 012704 027137    MOV      #MSG114,R4 ; PRINT FORWARD SOFT ERRORS

```

```

1975 004044 000004          TYPE          ;TYPE MSG
1976 004046 013704 000676    MOV      UNP,R4
1977 004052 016403 002730    MOV      RFHARD(R4),R3
1978 004056 104400          TYP OCT      ;PRINT HARD FORWARE ERRORS
1979 004060 012704 026413    MOV      #MSG77,R4
1980 004064 000004          TYPE          ;TYPE MSG
1981 004066 013704 000676    MOV      UNP,R4
1982 004072 016403 001130    MOV      DATER1(R4),R3
1983 004076 104400          TYP OCT      ;PRINT DATA ERROR FORWARD NUMBER
1984 004100 012704 026216    MOV      #MSG68,R4
1985 004104 000004          TYPE          ;TYPE MSG
1986 004106 013704 000676    MOV      UNP,R4
1987 004112 016403 001150    MOV      RDERR1(R4),R3
1988 004116 104400          TYP OCT      ;PRINT REVESE ERROR NUMBER
1989 004120 012704 027126    MOV      #MSG113,R4
1990 004124 000004          TYPE          ;TYPE MSG
1991 004126 013704 000676    MOV      UNP,R4
1992 004132 016403 002710    MOV      RRSOFT(R4),R3
1993 004136 104400          TYP OCT      ;PRINT REVERSE SOFT ERROR
1994 004140 012704 027137    MOV      #MSG114,R4
1995 004144 000004          TYPE          ;TYPE MSG
1996 004146 013704 000676    MOV      UNP,R4
1997 004152 016403 002750    MOV      RRHARD(R4),R3
1998 004156 104400          TYP OCT
1999 004160 012704 026402    MOV      #MSG76,R4
2000 004164 000004          TYPE          ;TYPE MSG
2001 004166 013704 000676    MOV      UNP,R4
2002 004172 016403 001170    MOV      DEREV1(R4),R3
2003 004176 104400          TYP OCT      ;PRINT DATA REVERSE ERROR NUMBER
2004 004200 000207          RTS      PC      ;RETURN
2005 004202 005237 000730    STPX:  INC      BTSTF      ;SET STAT ONLY PRINT
2006 004206 004737 007334    JSR      PC,BTPRT      ;PRINT BAD TAPE STATS
2007 004212 005037 000730    CLR      BTSTF      ;CLEAR FLAG
2008 004216 017700 174366    START5: MOV      @SWR,R0      ;LOAD SWR
2009 004222 042700 177762    BIC      #177762,R0      ;MASK READ/WRITE SWITCHES
2010 004226 022700 000015    CMP      #15,R0      ;SEE IF HAVE READ OR WRITE
2011 004232 001417          BEQ      START8      ;IF NOT: BR
2012 004234 105777 174262    START6: TSTB      @DS      ;SEE IF HAVE UNIT READY
2013 004240 100411          BMI      START7      ;IF SO: BR
2014 004242 005337 000670    DEC      STAL
2015 004246 001372          BNE      START6      ;DELAY FOR TUR
2016 004250 004737 022602    JSR      PC,PAPRT      ;PRINT HEADER
2017 004254 012704 025750    MOV      #MSG49,R4
2018 004260 000004          TYPE          ;TYPE MSG
2019 004262 000000          HALT          ;STOP
2020 004264 062737 000002 000676    START7: ADD      #2,UNP      ;POINT TO NEXT UNIT
2021 004272 000137 003406    START8: JMP      START1      ;CONTINUE
2022
2023          ;RANDOM BASE RESET*****
2024
2025 004276 012737 153624 000626    RANSET: MOV      #153624,RANBAS ;RESET BASE
2026 004304 012737 032561 000630    MOV      #32561,RANSAV      ;RESET BUFFER
2027 004312 013737 000632 000554    MOV      RCSAV,RCNT      ;RESET RECORD COUNT
2028 004320 013737 000634 000556    MOV      FCSAV,FMCNT      ;RESET FRAME COUNT
2029 004326 000207          RTS      PC
2030

```

```

2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043 004330 013777 000552 174204 REOT: MOV UDES,ATC ;LOAD TAPE CONTROL REGISTER
2044 004336 012777 000011 174144 MOV #11,AC1 ;DRIVE CLEAR
2045 004344 105777 174152 1S: TSTB ADS ;WAIT FOR DRY
2046 004350 100375 BPL 1S
2047 004352 012777 000007 174130 MOV #7,AC1 ;START REWIND
2048 004360 005737 000726 TST BTFLG ;SEE IF BAD TAPE OVERFLOW REWIND
2049 004364 001004 BNE REOT1A ;IF SO: BR
2050 004366 013700 000662 MOV EOTREC,R0
2051 004372 042700 100000 BIC #100000,R0 ;SET RECORD NUMBER OF EOT
2052 004376 005037 000662 REOT1A: CLR EOTREC ;CLEAR EOT INDICATOR & REC COUNT
2053 004402 004737 022602 JSR PC,PAPRT ;PRINT HEADER
2054 004406 022737 000002 000726 CMP #2,BTFLG ;SEE IF POSITION ERROR
2055 004414 001003 BNE REOT1B ;IF NOT: BR
2056 004416 012704 027017 MOV #MSG109,R4 ;SET POSITION ERROR MSG
2057 004422 000406 BR REOT1F
2058 004424 022737 000001 000726 REOT1B: CMP #1,BTFLG ;SEE IF BAD TAPE OVERFLOW
2059 004432 001004 BNE REOT1C ;IF NOT: BR
2060 004434 012704 026645 MOV #MSG106,R4 ;SET BAD TAPE OVERFLOW MSG
2061 004440 000004 REOT1F: TYPE ;TYPE MSG
2062 004442 000412 BR REOT1E
2063 004444 012704 024650 REOT1C: MOV #MSG20,R4 ;SET EOT MSG
2064 004450 000004 TYPE ;TYPE MSG
2065 004452 013704 000676 MOV UNP,R4
2066 004456 005264 002650 INC EOTC0(R4) ;BUMP CNTR
2067 004462 016403 002650 MOV EOTC0(R4),R3
2068 004466 104400 TYPOCT ;PRINT EOT CNTR
2069 004470 012704 026672 REOT1E: MOV #MSG16A,R4
2070 004474 000004 TYPE ;TYPE MSG
2071 004476 005037 000726 CLR BTFLG ;CLEAR BAD TAPE FLAG
2072 004502 004737 003734 JSR PC,STP ;PRINT STATS
2073 004506 004737 007334 JSR PC,BTPRT ;PRINT BAD TAPE STATS
2074 004512 105777 174004 REOT2: TSTB ADS ;BRANCH IF DRY SET
2075 004516 100414 BMI REOT2A
2076 004520 005337 000670 DEC STAL
2077 004524 001372 BNE REOT2 ;WAIT DRY
2078 004526 012737 024507 000654 MOV #MSG6,EMADDR
2079 004534 004737 022602 JSR PC,PAPRT ;PRINT HEADER
2080 004540 012704 026124 MOV #MSG60,R4
2081 004544 000004 TYPE ;TYPE MSG
2082 004546 000000 HALT
2083 004550 105337 005054 REOT2A: DECB REOTC ;SEE IF LAST UNIT TO REACH EOT
2084 004554 001410 BEQ REOT3 ;IF SO: BR
2085 004556 013700 000676 MOV UNP,R0
2086 004562 052760 100000 000746 BIS #100000,UN1(R0) ;SET EOT FLAG

```

2087	004570	005726			TST	(SP)+		;RESET STACK POINTER
2088	004572	000137	004264		JMP	START7		;GO TO NEXT UNIT
2089	004576	000337	005054		REOT3:	SWAB	REOTC	
2090	004602	013700	005054			MOV	REOTC,R0	
2091	004606	000337	005054			SWAB	REOTC	
2092	004612	110037	005054			MOVB	R0,REOTC	;RESTORE EOT UNIT COUNTER
2093	004616	005037	000676			CLR	UNP	
2094	004622	013700	000676			MOV	UNP,R0	;POINT TO FIRST UNIT
2095	004626	016037	000746	000552	REOT4:	MOV	UN1(R0),UDES	;LOAD UNIT DESCRIPTION
2096	004634	013777	000552	173700		MOV	UDES,@TC	;LOAD COMMAND REGISTER
2097	004642	032777	020000	173652	REOT5:	BIT	#20000,@DS	
2098	004650	001374				BNE	REOT5	;AWAIT PIP RESET
2099	004652	032777	000002	173642		BIT	#2,@DS	;SEE IF HAVE BOT
2100	004660	001012				BNE	REOT6	;IF SO: BR
2101	004662	012700	000001			MOV	#1,R0	
2102	004666	004737	022602			JSR	PC,PAPRT	;PRINT HEADER
2103	004672	012704	025715			MOV	#MSG48,R4	
2104	004676	000004				TYPE		;TYPE MSG
2105	004700	000000				HALT		
2106	004702	013700	000676			MOV	UNP,R0	
2107	004706	042760	100000	000746	REOT6:	BIC	#100000,UN1(R0)	;CLEAR EOT FLAG
2108	004714	062737	000002	000676		ADD	#2,UNP	
2109	004722	013700	000676			MOV	UNP,R0	;POINT TO NEXT UNIT
2110	004726	022760	177777	000746		CMP	#-1,UN1(R0)	;BRANCH IF NOT LAST UNIT
2111	004734	001334				BNE	REOT4	
2112	004736	005037	000676		REOT7:	CLR	UNP	;CLEAR UNIT POINTER
2113	004742	005037	000636			CLR	TINF	;CLEAR TTY INPUT FLAG
2114	004746	005737	000736			TST	ASEQF	;SEE IF AUTO SEQ
2115	004752	001402				BEQ	REOTX	;IF NOT: BR
2116	004754	005726				TST	(SP)+	;RESET STACK POINTER
2117	004756	000207				RTS	PC	;RETURN TO AUTO SEQ
2118	004760	004737	004276		REOTX:	JSR	PC,RANSET	;GO RESET RANDOM BASE
2119	004764	012737	177777	014276		MOV	#-1,PATS	;PRESET PATTERN
2120	004772	005037	015030			CLR	RDFL	;CLEAR RANDOM FLAG
2121	004776	005737	000572			TST	SPFLG	;SEE IF SINGLE PASS
2122	005002	001422				BEQ	REOTXX	;IF NOT: BR
2123	005004	012704	026525		TEND:	MOV	#MSG100,R4	
2124	005010	000004				TYPE		;TYPE MSG
2125	005012	013700	000042			MOV	@#42,R0	;GET ACT11 RETURN ADDRESS
2126	005016	001405				BEQ	HERE	;BRANCH IF NOT ACT11
2127	005020	000005				RESET		
2128	005022	004710			\$ENDAD:	JSR	PC,(R0)	
2129	005024	000240				NOP		
2130	005026	000240				NOP		
2131	005030	000240				NOP		
2132	005032	000240			HERE:	NOP		
2133	005034	005737	003040			TST	CHNFLG	;BRANCH IF NOT CHAIN MODE
2134	005040	001402				BEQ	1\$	
2135	005042	000137	022004			JMP	ASEQ0	;RETURN TO AUTO SEQUENCER
2136	005046	000000			1\$:	HALT		
2137	005050	000137	003236		REOTXX:	JMP	STARTE	;RESTART AT BLOCK NUMBER ONE
2138	005054	000000			REOTC:	0		;EOT UNIT COUNTER

```

2139
2140
2141
2142
2143
2144
2145
2146
2147
2148 005056 032777 001000 173524 RWND: BIT #1000,@SWR ;SEE IF SHOULD REWIND
2149 005064 001001 BNE RWNDA ;IF SO: BR
2150 005066 000207 RTS PC ;ELSE EXIT
2151 005070 013737 000676 000716 RWNDA: MOV UNP,UPS ;SAVE UNIT POINTER
2152 005076 005037 000676 CLR UNP ;CLEAR POINTER
2153 005102 005037 000662 CLR EOTREC ;CLEAR EDT FLAG
2154 005106 000337 005054 SWAB REOTC
2155 005112 013700 005054 MOV REOTC,R0
2156 005116 000337 005054 SWAB REOTC
2157 005122 110037 005054 MOV RO,REOTC ;RESTORE EOT UNIT COUNTER
2158 005126 013700 000676 RWND0: MOV UNP,R0 ;POINT TO UNIT ENTRY
2159 005132 022760 177777 000746 CMP #-1,UN1(R0) ;BRANCH IF LAST ENTRY
2160 005140 001445 BEQ RWND2
2161 005142 005760 000746 TST UN1(R0) ;SEE IF ALREADY REWINDING
2162 005146 100433 BMI RWND1A ;IF SO: BR
2163 005150 016037 000746 000552 MOV UN1(R0),UCES ;SET UNIT DESCRIPTION
2164 005156 013777 000552 173356 MOV UCES,@TC ;LOAD COMMAND REGISTER
2165 005164 012777 000011 173316 MOV #11,@C1 ;DRIVE CLEAR
2166 005172 012777 000007 173310 MOV #7,@C1 ;START REWIND
2167 005200 105777 173316 1$: TSTB @DS
2168 005204 100414 BMI RWND1A ;IF DRY: BR
2169 005206 005337 000670 DEC STAL
2170 005212 001372 BNE 1$ ;AWAIT DRY
2171 005214 012737 024507 000654 MOV #MSG6,EMADDR
2172 005222 004737 022602 JSR PC,PAPRT ;PRINT HEADER
2173 005226 012704 026245 MOV #MSG70,R4
2174 005232 000004 TYPE
2175 005234 000000 HALT ;TYPE MSG
2176 005236 042760 100000 000746 RWND1A: BIC #100000,UN1(R0) ;CLEAR EOT FLAG
2177 005244 062737 000002 000676 ADD #2,UNP ;BUMP POINTER
2178 005252 000725 BR RWND0 ;DO NEXT UNIT
2179 005254 005037 000676 RWND2: CLR UNP ;CLEAR POINTER
2180 005260 013700 000676 RWND3: MOV UNP,R0 ;POINT TO UNIT ENTRY
2181 005264 022760 177777 000746 CMP #-1,UN1(R0) ;BRANCH IF LAST ENTRY
2182 005272 001441 BEQ RWNDX
2183 005274 016037 000746 000552 MOV UN1(R0),UCES ;SET UNIT DESCRIPTION
2184 005302 013777 000552 173232 MOV UCES,@TC ;LOAD COMMAND REGISTER
2185 005310 032777 020000 173204 1$: BIT #20000,@DS
2186 005316 001374 BNE 1$ ;AWAIT PIP RESET
2187 005320 013777 000552 173214 MOV UCES,@TC ;LOAD UNIT DESCRIPTION
2188 005326 032777 000002 173166 BIT #2,@DS ;SEE IF HAVE BOT
2189 005334 001407 BEQ RWND6 ;IF NOT: BR
2190 005336 062737 000002 000676 RWND5: ADD #2,UNP ;BUMP POINTER
2191 005344 012777 000011 173136 MOV #11,@C1 ;DRIVE CLEAR
2192 005352 000742 BR RWND3 ;DO NEXT UNIT
2193 005354 012700 000001 RWND6: MOV #1,R0
2194 005360 004737 022602 JSR PC,PAPRT ;PRINT HEADER

```

```

2195 005364 012704 025715      MOV      #MSG48,R4
2196 005370 000004      TYPE
2197 005372 000000      HALT      ;TYPE MSG
2198 005374 000760      BR        RWND5      ;DO NEXT UNIT
2199 005376 013737 000716 000676 RWNDX: MOV      UPS,UNP      ;RESTORE UNIT POINTER
2200 005404 013700 000676      MOV      UNP,R0
2201 005410 016037 000746 000552      MOV      UN1(R0),UDES ;RESET UNIT DESCRIPTION
2202 005416 013777 000552 173116      MOV      UDES,@TC
2203 005424 000207      RTS        PC      ;RETURN TO TEST
2204
2205
2206      ;*****
2207      ;INITIALIZE SELECTED SALVE
2208      ;THIS ROUTINE REWINDS AND SETS THE PROPER DENSITY IF
2209      ;THE DENSITY REQUIRED FOR THE TEST IS DIFFERENT FROM
2210      ;THE DENSITY AT WHICH THE SLAVE IS SELECTED.
2211      ;*****
2212
2213 005426 013746 000552      INIT:  MOV      UDES,-(SP)      ;GET UNIT DESCRIPTION
2214 005432 013777 000550 173060      MOV      DVN,@CS      ;LOAD DRIVE #
2215 005440 011677 173076      MOV      (SP),@TC      ;LOAD SLAVE # & SLAVE DESCRIPTION
2216 005444 042716 174377      BIC      #174377,(SP) ;CLEAR ALL BUT DENSITY BITS
2217 005450 022726 001400      CMP      #1400,(SP)+ ;BRANCH IF NOT NRZ
2218 005454 001005      BNE      1$
2219 005456 032777 000040 173036      BIT      #40,@DS      ;BRANCH IF SLAVE IS IN PE MODE
2220 005464 001422      BEQ      4$      ;PES = 0
2221 005466 000404      BR        2$
2222 005470 032777 000040 173024 1$:  BIT      #40,@DS      ;BRANCH IF SLAVE IS IN PE MODE
2223 005476 001015      BNE      4$      ;PES = 1
2224 005500 012777 000007 173002 2$:  MOV      #7,@C1      ;LOAD REWIND COMMAND
2225 005506 105777 173010 20$:  TSTB     @DS      ;WAIT FOR READY
2226 005512 100375      BPL      20$
2227 005514 032777 020000 173000 3$:  BIT      #20000,@DS ;WAIT FOR PIP = 0
2228 005522 001374      BNE      3$
2229 005524 012777 000011 172756      MOV      #11,@C1      ;CLEAR DRIVE
2230 005532 000207      4$:  RTS        PC

```

```

2231 ;*****
2232 ;WRITE ROUTINE:
2233 ;
2234 ;THIS ROUTINE IS USED TO WRITE ONTO TAPE THE BLOCK
2235 ;OF DATA DESCRIBED BY THE OPERATOR AND SET UP
2236 ;IN THE SEQUENCE FORMATTER. THE TAPE UNIT TO BE USED
2237 ;HAS BEEN ASSIGNED BY THE SEQUENCE FORMATTER AND
2238 ;ITS PARAMETERS SET IN A UNIT DESCRIPTION WORD.
2239 ;AS EACH RECORD OF THE BLOCK IS WRITTEN, IT IS CHECKED
2240 ;FOR STATUS ERRORS, WORD COUNT ZERO, AND CORRECT CURRENT
2241 ;MEMORY ADDRESS. IF THE WRITE OPERATION RESULTS IN
2242 ;ANY ERROR CONDITION, A WRITE RETRY OF THAT OPERATION
2243 ;MAY BE DONE BY SETTING SWITCH FOUR (4) TO A ONE (1).
2244 ;THE RETRY CONSISTS OF A BACKSPACE, ERASE FORWARD, AND
2245 ;REWRITE OF THE RECORD. (SEE WRITE RETRY SUBROUTINE)
2246 ;AFTER ALL DATA RECORDS IN THE BLOCK HAVE BEEN
2247 ;WRITTEN, THE WRITE ROUTINE WILL EXECUTE A WRITE
2248 ;TAPE MARK COMMAND IF THE TTY RESPONSE TM=1 WAS
2249 ;MADE AT INITIAL START. THE TM IS COUNTED AS TOTAL
2250 ;DATA RECORDS PLUS ONE (IE: IF 100 DATA RECORDS; TM=RECORD 101)
2251 ;IF THE WRITE OPERATION (DATA OR TM) CAUSES THE SELECTED SLAVE
2252 ;TO REACH END OF TAPE (EOT) AND THERE IS TO BE NO READING DONE,
2253 ;(SW2 AND SW3 SET TO A 1) THEN THE SLAVE IS REWOUND AND
2254 ;FLAGGED AS UNAVAILABLE FOR TESTING UNTIL ALL SLAVES HAVE
2255 ;REACHED EOT AND BEEN REWOUND AT WHICH TIME TESTING IS
2256 ;RESUMED ON ALL AVAILABLE SLAVES.
2257 ;WRITE RETRY MAY BE ALLOWED VIA CONSOLE SWITCH FOUR (4).
2258 ;ERROR CHECKING MAY BE DISALLOWED VIA CONSOLE SWITCH
2259 ;TWELVE (12).
2260 ;WRITING TO TAPE MAY BE DISALLOWED VIA CONSOLE SWITCH
2261 ;ZERO (0).
2262 ;*****
2263
2264 005534 032777 000001 173046 WRITE: BIT #1,@SWR ;SEE IF SHOULD WRITE
2265 005542 001402 BEQ WRT ;
2266 005544 000137 006332 JMP WEX ;IF NOT: BR
2267 005550 013700 000554 WRT: MOV RCNT,RO ;RO=RECORD COUNT
2268 005554 012737 024502 000654 W0: MOV #MSG5,EMADDR ;SET ERROR MSG ADDRESS
2269 005562 013777 000556 172726 MOV FMCNT,@FC ;LOAD CHAR COUNT
2270 005570 012777 027346 172716 MOV #WDATA,@BA ;SET DATA ADDR
2271 005576 112737 000060 000674 MOV #60,MTC1 ;SET WRITE OP COMMAND
2272 005604 012737 005616 000664 MOV #W1,RTRN ;SET RETURN ADDRESS
2273 005612 000137 021072 JMP TAPG ;GO EXECUTE COMMAND
2274 005616 032777 002000 172676 W1: BIT #2000,@DS ;SEE IF EOT
2275 005624 001414 BEQ W2 ;IF NOT AT EOT: BR
2276 005626 005737 000662 TST EOTREC ;BRANCH IF WRITTEN PAST EOT
2277 005632 100411 BMI W2
2278 005634 010037 000662 MOV RO,EOTREC ;SAVE RECORD COUNT
2279 005640 052737 100000 000662 BIS #100000,EOTREC ;++B SET EOT INDICATOR
2280 005646 005337 000662 DEC EOTREC ;++B ADJUST RECORD COUNT
2281 005652 012700 000002 MOV #2,RO ;SET TO WRITE 1 LAST RECORD
2282 005656 032777 010000 172724 W2: BIT #10000,@SWR ;SEE IF SHOULD CHECK ERRORS
2283 005664 001002 BNE W3 ;IF NOT: BR
2284 005666 004737 017244 JSR PC,ERCHK ;GO CHECK ERRORS
2285 005672 013737 000576 000670 W3: MOV WSTAL,STAL ;SET DELAY
2286 005700 004737 012124 JSR PC,STALL ;DELAY

```

2287	005704	005737	000714		TST	RTYFL		;SEE IF RETRY TIME
2288	005710	001401			BEQ	W3A		;IF NOT: BR
2289	005712	000207			RTS	PC		;ELSE RETURN
2290	005714	005737	000710		W3A: TST	SERFL		;SEE IF WRITE ERROR
2291	005720	001450			BEQ	W5		;IF NOT: BR
2292	005722	013704	000676		MOV	UNP,R4		
2293	005726	005264	001070		INC	WTER1(R4)		;BUMP WRITE ERROR
2294	005732	005037	000710		CLR	SERFL		;CLEAR STATUS ERROR FLAG
2295	005736	032777	000020	172644	BIT	#20,@SWR		;SEE IF RETRY
2296	005744	001436			BEQ	W5		;IF NOT: BR
2297	005746	013703	000724		MOV	ERSAV,R3		
2298	005752	042703	102700		BIC	#102700,R3		;MASK UNRECOVERABLE ERROR
2299	005756	001410			BEQ	W4		;IF SO: BR
2300	005760	004737	022602		JSR	PC,PAPRT		;PRINT HEADER
2301	005764	012704	026424		MOV	#MSG78,R4		
2302	005770	000004			TYPE			;TYPE MSG
2303	005772	004737	011244		JSR	PC,NRTP		;PRINT ER FOR NON-RETRYABLE
2304	005776	000421			BR	W5		
2305	006000	013704	000676		W4: MOV	UNP,R4		
2306	006004	005264	001050		INC	RTY1(R4)		;BUMP RETRY CNTR
2307	006010	032777	002000	172572	BIT	#2000,@SWR		;SEE IF PRINT ERRORS
2308	006016	001003			BNE	W4A		;IF NOT: BR
2309	006020	012704	026140		MOV	#MSG64,R4		
2310	006024	000004			TYPE			;TYPE MSG
2311	006026	005037	000704		W4A: CLR	RTCNT		;CLEAR RETRY NUMBER
2312	006032	005037	000702		CLR	RPCNT		;CLEAR REPEAT COUNTER
2313	006036	004737	006374		JSR	PC,WRTY		;GO RETRY WRITE ERROR
2314	006042	005037	000714		W5: CLR	RTYFL		;CLEAR RETRY COUNTER
2315	006046	005300			DEC	RO		;SEE IF DONE ALL
2316	006050	001241			BNE	W0		;IF NOT: BR
2317	006052	005737	000564		W6: TST	TMEX		;SEE IF TM
2318	006056	001525			BEQ	WEX		;IF NOT: BR
2319	006060	005237	000700		INC	TMFLG		;SET TM FLAG
2320	006064	012737	026045	000654	WTM: MOV	#MSG54,EMADDR		;POINT TO TM ERROR MSG
2321	006072	012737	000026	000674	MOV	#26,MTCT		;SET TM OP CODE
2322	006100	012777	000000	172410	MOV	#0,@FC		;LOAD FRAME COUNTER
2323	006106	012777	027346	172400	MOV	#WDATA,@BA		;LOAD BUS ADDRESS
2324	006114	012737	006126	000664	MOV	#WTMO,RTN		;SAVE RETURN ADDRESS
2325	006122	000137	021072		JMP	TAPG		;WRITE TM
2326	006126	032777	010000	172454	WTMO: BIT	#10000,@SWR		;SEE IF SHOULD CHECK ERRORS
2327	006134	001076			BNE	WEX		
2328	006136	032777	000004	172356	BIT	#4,@DS		;SEE IF TM STATUS
2329	006144	001011			BNE	WTM1		;IF SO: BR
2330	006146	012737	027346	021012	MOV	#WDATA,CADER		;SET EXPT BUS ADDRESS
2331	006154	012737	000001	021020	MOV	#1,DRVER		;INDICATE ERROR
2332	006162	004737	020064		JSR	PC,ERPT		;PRINT TM ERROR
2333	006166	000404			BR	WTM2		
2334	006170	012703	027346		WTM1: MOV	#WDATA,R3		;SET EXPT ADDRESS
2335	006174	004737	017340		JSR	PC,ER2		;GO CHECK FOR OTHER ERRORS
2336	006200	005737	000714		WTM2: TST	RTYFL		;SEE IF RETRY
2337	006204	001401			BEQ	WTM3		;IF NOT: BR
2338	006206	000207			RTS	PC		;ELSE RETURN TO RETRY ROUTINE
2339	006210	005737	000710		WTM3: TST	SERFL		;SEE IF WRITE ERROR
2340	006214	001446			BEQ	WEX		;IF NOT: BR
2341	006216	013704	000676		MOV	UNP,R4		
2342	006222	005264	001070		INC	WTER1(R4)		;BUMP WRITE ERROR

2343	006226	032777	000020	172354	BIT	#20,@SWR	;SEE IF SHOULD RETRY
2344	006234	001436			BEQ	WEX	;IF NOT: BR
2345	006236	013703	000724		MOV	ERSAV,R3	
2346	006242	042703	102700		BIC	#102700,R3	;MASK UNRECOVERABLE ERROR
2347	006246	001410			BEQ	WTM4	;IF SO: BR
2348	006250	004737	022602		JSR	PC,PAPRT	;PRINT HEADER
2349	006254	012704	026424		MOV	#MSG78,R4	
2350	006260	000004			TYPE		;TYPE MSG
2351	006262	004737	011244		JSR	PC,NRTP	;PRINT ER FOR NON-RETRYABLE
2352	006266	000421			BR	WEX	
2353	006270	005037	000702		WTM4: CLR	RPCNT	;CLEAR REPEAT CNTR
2354	006274	013704	000676		MOV	UNP,R4	
2355	006300	005264	001050		INC	RTY1(R4)	;BUMP RETRY CNTR
2356	006304	005037	000704		CLR	RTCNT	;CLEAR RETRY CNTR
2357	006310	032777	002000	172272	BIT	#2000,@SWR	;SEE IF PRINT ERRORS
2358	006316	001003			BNE	WTM4A	;IF NOT: BR
2359	006320	012704	026140		MOV	#MSG64,R4	
2360	006324	000004			TYPE		;TYPE MSG
2361	006326	004737	006374		WTM4A: JSR	PC,WRTY	;GO DO RETRY
2362	006332	005037	000714		WEX: CLR	RTYFL	;CLEAR RETRY FLAG
2363	006336	005037	000700		CLR	TMFLG	;CLEAR TAPE MARK FLAG
2364	006342	005737	000662		TST	EOTREC	;BRANCH IF NOT AT EOT
2365	006346	100011			BPL	WRWX	
2366	006350	017703	172234		WRW: MOV	@SWR,R3	
2367	006354	042703	177763		BIC	#177763,R3	
2368	006360	022703	000014		CMP	#14,R3	;SEE IF WRITE ONLY
2369	006364	001002			BNE	WRWX	;IF NOT: BR
2370	006366	000137	004330		JMP	REOT	;ELSE REWIND
2371	006372	000207			WRWX: RTS	PC	;EXIT

```

2372                                     ;*****
2373                                     ;WRITE ERROR RETRY
2374                                     ;
2375                                     ;*****
2376
2377 006374 012737 000001 000714 WRTY:  MOV  #1,RTYFL      ;SET RETRY FLAG
2378 006402 004737 006776 WRTY0: JSR   PC,WRTSB      ;GO SPACE REVERSE FOR REPEAT
2379 006406 005737 000700      TST   TMFLG          ;SEE IF TAPE MARK TIME
2380 006412 001003      BNE   WRTYTM          ;IF SO: BR
2381 006414 004737 005554      JSR   PC,W0          ;REWRITE RECORD
2382 006420 000402      BR    WRTYR          ;GO ON
2383 006422 004737 006064 WRTYTM: JSR   PC,WTM          ;GO WRITE TAPE MARK AGAIN
2384 006426 005737 000710 WRTYR:  TST   SERFL          ;REWRITE GOOD
2385 006432 001024      BNE   WRTY2          ;IF NOT: BR
2386 006434 005237 000702      INC   RPCNT          ;BUMP REPEAT COUNTER
2387 006440 022737 000004 000702      CMP   #4,RPCNT      ;SEE IF FOUR GOOD REPEATS
2388 006446 001355      BNE   WRTY0          ;IF NOT: REPEAT
2389 006450 032777 002000 172132      BIT   #2000,@SWR      ;SEE IF PRINT
2390 006456 001011      BNE   WRTY1          ;IF NOT: BR
2391 006460 012704 026632      MOV   #MSG105,R4
2392 006464 000004      TYPE          ;TYPE MSG
2393 006466 012704 026162      MOV   #MSG65,R4
2394 006472 000004      TYPE          ;TYPE MSG
2395 006474 013703 000704      MOV   RTCNT,R3
2396 006500 104400      TYPOCT          ;PRINT RETRY NUMBER
2397 006502 000207 WRTY1:  RTS   PC          ;RESUME TESTING
2398 006504 013703 000724 WRTY2:  MOV   ERSAR,R3      ;GET ER
2399 006510 005037 000650      CLR   TEMP3          ;CLEAR RECOVERABLE ERROR INDICATOR
2400 006514 042703 102700      BIC   #102700,R3      ;MASK RECOVERABLE BITS
2401 006520 001413      BEQ   WRTY2A          ;IF RECOVERABLE: BR
2402 006522 004737 022602      JSR   PC,PAPRT        ;PRINT HEADER
2403 006526 012704 026424      MOV   #MSG78,R4
2404 006532 000004      TYPE          ;TYPE MSG
2405 006534 004737 011244      JSR   PC,NRTP          ;PRINT ER
2406 006540 012737 000001 000650      MOV   #1,TEMP3      ;SET FLAG
2407 006546 000407      BR    WRTY2B
2408 006550 032777 002000 172032 WRTY2A: BIT   #2000,@SWR      ;SEE IF PRINT
2409 006556 001025      BNE   WRTY3          ;IF NOT: BR
2410 006560 012704 027051      MOV   #MSG110,R4
2411 006564 000004      TYPE          ;TYPE MSG
2412 006566 012704 026162 WRTY2B: MOV   #MSG65,R4
2413 006572 000004      TYPE          ;TYPE MSG
2414 006574 013703 000704      MOV   RTCNT,R3
2415 006600 104400      TYPOCT          ;PRINT RETRY NUMBER
2416 006602 012704 027073      MOV   #MSG111,R4
2417 006606 000004      TYPE          ;TYPE MSG
2418 006610 013703 000702      MOV   RPCNT,R3
2419 006614 104400      TYPOCT          ;PRINT REPEAT NUMBER
2420 006616 005737 000650      TST   TEMP3          ;SEE IF DID NON-RECOVERABLE
2421 006622 001403      BEQ   WRTY3          ;IF NOT: BR
2422 006624 005037 000650      CLR   TEMP3          ;CLEAR FLAG
2423 006630 000207 WRTY3:  RTS   PC          ;EXIT
2424 006632 005737 000704      TST   RTCNT          ;SEE IF FIRST RETRY
2425 006636 001004      BNE   WRTY3A          ;IF NOT: BR
2426 006640 013704 000676      MOV   UNP,R4
2427 006644 005364 001070      DEC   WTER1(R4)      ;DECREMENT WRITE ERROR CNTR

```

```

2428 006650 013704 000676      WRTY3A: MOV      UNP,R4      ;GET UNIT NUMBER
2429 006654 016437 001030 000732  MOV      BTADDR(R4),BTPT ;GET ADDRESS OF UNIT BAD TAPE CNTR
2430 006662 017704 172044      MOV      @BTPT,R4      ;GET COUNTER
2431 006666 005724      TST      (R4)+      ;SET POINTER OFFSET
2432 006670 010477 172036      MOV      R4,@BTPT
2433 006674 013703 000732      MOV      BTPT,R3
2434 006700 060304      ADD      R3,R4      ;SET ABSOLUTE POINTER
2435 006702 013714 000656      MOV      BLCNTR,(R4) ;SET BLOCK NUMBER
2436 006706 062704 000040      ADD      #40,R4      ;ADD RCNT OFFSET
2437 006712 013714 000554      MOV      RCNT,(R4)
2438 006716 160014      SUB      R0,(R4)      ;SET RECORD NUMBER
2439 006720 005214      INC      (R4)      ;CORRECT RECORD NUMBER
2440 006722 022777 000040 172002  CMP      #40,@BTPT    ;SEE IF TOO MANY BAD SPOTS
2441 006730 001002      BNE      WRTY4      ;IF NOT: BR
2442 006732 000137 007172      JMP      BTOV      ;ELSE GO TO BAD TAPE OVERFLOW
2443 006736 005237 000704      WRTY4: INC      RTCNT ;BUMP RETRY COUNTER
2444 006742 022737 000004 000704  CMP      #4,RTCNT    ;SEE IF DONE 4 RETRIES
2445 006750 001410      BEQ      WRTY5      ;IF SO: BR
2446 006752 013704 000676      MOV      UNP,R4
2447 006756 005264 001050      INC      RTY1(R4)    ;BUMP RETRY COUNTER
2448 006762 005237 000734      INC      ERTFL      ;SET ERASE FLAG
2449 006766 000137 006402      JMP      WRTY0      ;DO NEXT RETRY
2450 006772 000137 007410      WRTY5: JMP      BTUR    ;ELSE GO TO BAD TAPE UNRECOVERABLE
2451
2452      ;WRITE RETRY BACKSPACE-ERASE SUBROUTINE*****
2453
2454 006776 005037 000710      WRTSB: CLR      SERFL    ;CLEAR FLAG
2455 007002 013737 000600 000670  MOV      TSTAL,STAL
2456 007010 004737 012124      JSR      PC,STALL    ;DO TURN AROUND DELAY
2457 007014 012737 026173 000654  MOV      #MSG66,EMADDR ;SET ERROR CODE
2458 007022 012777 177777 171466  MOV      #-1,@FC      ;SET TO BACKSPACE 1 RECORD
2459 007030 012777 033354 171456  MOV      #RDATA,@BA   ;SET BA
2460 007036 004737 012054      JSR      PC,BKRT      ;GO BACKSPACE
2461 007042 005737 000710      TST      SERFL      ;SEE IF ERROR
2462 007046 001406      BEQ      WRTSB1      ;IF NOT: BR
2463 007050 012737 000002 000726  WRTSB0: MOV      #2,BTFLG ;SET FLAG
2464 007056 022626      CMP      (SP)+,(SP)+ ;RESET STACK
2465 007060 000137 004330      JMP      REOT        ;GO REWIND AND REMOVE FROM TESTING
2466 007064 005737 000734      WRTSB1: TST      ERTFL ;SEE IF SHOULD ERASE
2467 007070 001001      BNE      WRTSB2      ;IF SO: BR
2468 007072 000207      RTS      PC          ;RETURN
2469 007074 005037 000734      WRTSB2: CLR      ERTFL ;CLEAR ERASE FLAG
2470 007100 005037 000702      CLR      RPCNT      ;CLEAR REPEAT CNTR
2471 007104 005037 000710      CLR      SERFL      ;CLEAR FLAG
2472 007110 012737 026206 000654  MOV      #MSG67,EMADDR ;SET ERROR CODE
2473 007116 005077 171374      CLR      @FC        ;CLEAR FRAME COUNT
2474 007122 012737 000024 000674  MOV      #24,MTCT1    ;SET ERASE OP-CODE
2475 007130 012777 027346 171356  MOV      #WDATA,@BA   ;SET BA
2476 007136 012737 007150 000664  MOV      #WRTSB3,RTRN ;SET RETURN ADDRESS
2477 007144 000137 021072      JMP      TAPG        ;GO ERASE
2478 007150 012703 027346      WRTSB3: MOV      #WDATA,R3 ;SET EXPT BA
2479 007154 004737 017340      JSR      PC,ER2      ;GO CHECK ERRORS
2480 007160 005737 000710      TST      SERFL      ;SEE IF ERROR
2481 007164 001737      BEQ      WRTSB1      ;IF NOT: BR
2482 007166 000137 007050      JMP      WRTSB0
2483

```

```

2484                                     ;BAD TAPE OVERFLOW SUBROUTINE*****
2485
2486 007172 005037 000714 BTOV: CLR RTYFL ;CLEAR RETRY FLAG
2487 007176 012737 000001 000726 MOV #1,BTFLG ;SET BAD TAPE OVERFLOW FLAG
2488 007204 005726 TST (SP)+ ;++B ADJUST STACK PTR
2489 007206 000137 004330 JMP REOT ;GO REWIND AND REMOVE FROM TESTING
2490 007212 013701 000732 BTOV0: MOV BTPT,R1 ;SET TABLE POINTER
2491 007216 005721 TST (R1)+
2492 007220 005000 CLR R0
2493 007222 010003 BTOV1: MOV R0,R3
2494 007224 000241 CLC
2495 007226 006003 ROR R3 ;R3=R3/2 FOR CORRECT NUMBER
2496 007230 104400 TYPOCT ;PRINT ENTRY NUMBER
2497 007232 012704 024577 MOV #MSG13+1,R4
2498 007236 000004 TYPE ;TYPE MSG
2499 007240 011103 MOV (R1),R3
2500 007242 104400 TYPOCT ;PRINT BLOCK NUMBER
2501 007244 012704 024604 MOV #MSG14,R4
2502 007250 000004 TYPE ;TYPE MSG
2503 007252 062701 000040 ADD #40,R1 ;SET POINTER OFFSET FOR RECOED NUMBER
2504 007256 012103 MOV (R1)+,R3
2505 007260 104400 TYPOCT ;PRINT RECORD NUMBER
2506 007262 162701 000040 SUB #40,R1 ;RESET POINTER FOR BLOCK NUMBER
2507 007266 005720 TST (R0)+
2508 007270 020077 171436 CMP R0,BTPT ;SEE IF DONE
2509 007274 001404 BEQ BTOV2 ;IF SO: BR
2510 007276 012704 025131 MOV #MSG28,R4
2511 007302 000004 TYPE ;TYPE MSG
2512 007304 000746 BR BTOV1 ;CONTINUE
2513 007306 005737 000730 BTOV2: TST BTSTF ;SEE IF STAT ONLY PRINT
2514 007312 001007 BNE BTOVX ;IF SO: BR
2515 007314 012703 000041 MOV #41,R3 ;SET SIZE OF TABLE
2516 007320 013704 000732 MOV BTPT,R4 ;SET POINTER
2517 007324 005024 BTOV3: CLR (R4)+ ;CLEAR TABLE
2518 007326 005303 DEC R3 ;SEE IF DONE
2519 007330 001375 BNE BTOV3 ;IF NOT: BR
2520 007332 000207 BTOVX: RTS PC ;RETURN
2521

```

```

2522
2523                ;BAD TAPE STATISTIC PRINT*****
2524
2525 007334 012704 025131      BTPRT:  MOV    #MSG28,R4
2526 007340 000004                TYPE    ;TYPE MSG
2527 007342 013704 000676      MOV    UNP,R4
2528 007346 016437 001030 000732  MOV    BTADDR(R4),BTPT ;SET TABLE POINTER
2529 007354 017703 171352      MOV    @BTPT,R3
2530 007360 000241      CLC
2531 007362 006003      ROR    R3                ;CORRECT NUMBER
2532 007364 104400      TYPOCT                ;PRINT NUMBER OF BAD SPOTS
2533 007366 012704 027105      MOV    #MSG112,R4
2534 007372 000004                TYPE    ;TYPE MSG
2535 007374 005777 171332      TST    @BTPT                ;SEE IF ANY BAD SPOTS
2536 007400 001001      BNE    BTPRT1                ;IF SO: BR
2537 007402 000207      RTS    PC                ;ELSE RETURN
2538 007404 000137 007212      BTPRT1: JMP    BTOVO                ;PRINT STATS
2539
2540                ;BAD TAPE UNRECOVERABLE SUBROUTINE*****
2541
2542 007410 004737 022602      BTUR:   JSR    PC,PAPRT                ;PRINT HEADER
2543 007414 012704 026733      MOV    #MSG107,R4
2544 007420 000004                TYPE    ;TYPE MSG
2545 007422 000207      RTS    PC                ;RESUME TESTING
2546

```

```

2547                                     :*****
2548                                     :READ SEQUENCER:
2549                                     :
2550                                     :THIS ROUTINE IS USED TO DETERMINE THE SEQUENCE
2551                                     :IN WHICH READ TAPE OPERATIONS ARE TO BE PERFORMED.
2552                                     :THIS IS NECESSARY WHEN THE UNIT BEING TESTED IS
2553                                     :CAPABLE OF READING DATA IN BOTH THE FORWARD AND
2554                                     :REVERSE DIRECTIONS.  CONSOLE SWITCHES ONE (1), TWO (2),
2555                                     :AND THREE (3) ARE USED TO DETERMINE THE READ SEQUENCE.
2556                                     :CONSOLE SWITCH ONE (1) DETERMINES WHETHER TO READ
2557                                     :THE BLOCK OF DATA FORWARD FIRST OR REVERSE FIRST.
2558                                     :SWITCH TWO (2) DISALLOWS READING IN THE REVERSE
2559                                     :DIRECTION AND SWITCH THREE (3) DISALLOWS READING IN
2560                                     :THE FORWARD DIRECTION.
2561                                     :*****
2562
2563 007424 012737 000002 000562 RSEQ:  MOV    #2,RDCMD
2564 007432 017704 171152          MOV    @SWR,R4          ;READ SWITCHES
2565 007436 042704 177763          BIC    #177763,R4        ;MASK READ BITS & SEE IF BOTH READS
2566 007442 001004          BNE    RSR                      ;IF NOT: BR
2567 007444 032777 000002 171136  BIT    #2,@SWR          ;SEE IF READ REVERSE FIRST
2568 007452 001050          BNE    RSFR                    ;IF NOT: BR
2569 007454 032777 000004 171126  RSR:  BIT    #4,@SWR          ;SEE IF SHOULD READ REVERSE
2570 007462 001005          BNE    RSF                      ;IF NOT: BR
2571 007464 012737 010000 000562  MOV    #10000,RDCMD        ;LOAD READ REVERSE COMMAND
2572 007472 004737 007736          JSR    PC,READ          ;GO READ REVERSE
2573 007476 032777 000010 171104  RSF:  BIT    #10,@SWR        ;SEE IF SHOULD READ FORWARD
2574 007504 001025          BNE    RSEX                    ;IF NOT: BR
2575 007506 032737 010000 000562  BIT    #10000,RDCMD        ;SEE IF HAVE READ REVERSE
2576 007514 001406          BEQ    RSFO                    ;IF NOT: BR
2577 007516 013737 000600 000670  MOV    TSTAL,STAL
2578 007524 004737 012124          JSR    PC,STALL          ;DO READ STALL
2579 007530 000406          BR     RSF1
2580 007532 032777 000001 171050  RSFO: BIT    #1,@SWR          ;SEE IF WRITE
2581 007540 001002          BNE    RSF1                    ;IF NOT: BR
2582 007542 004737 011700          JSR    PC,BKSP          ;GO BACKSPACE
2583 007546 012737 000002 000562  RSF1: MOV    #2,RDCMD        ;LOAD READ FORWARD COMMAND
2584 007554 004737 007736          JSR    PC,READ          ;GO READ
2585 007560 005737 000662          RSEX: TST    EOTREC        ;BRANCH IF NOT AT EOT
2586 007564 100002          BPL    1$
2587 007566 000137 004330          JMP    REOT              ;ELSE GO TO REWIND
2588 007572 000207          1$:   RTS    PC                  ;EXIT
2589
2590 007574 012737 010000 000562  RSFR:  MOV    #10000,RDCMD
2591 007602 032777 000010 171000  BIT    #10,@SWR          ;SEE IF SHOULD READ FORWARD
2592 007610 001013          BNE    RSFR1                    ;IF NOT: BR
2593 007612 032777 000001 170770  BIT    #1,@SWR          ;SEE IF WRITE
2594 007620 001002          BNE    RSFR0                    ;IF NOT: BR
2595 007622 004737 011700          JSR    PC,BKSP          ;GO BACKSPACE TO START
2596 007626 012737 000002 000562  RSFR0: MOV    #2,RDCMD        ;LOAD READ FORWARD COMMAND
2597 007634 004737 007736          JSR    PC,READ          ;GO READ FORWARD
2598 007640 032777 000004 170742  RSFR1: BIT    #4,@SWR          ;SEE IF SHOULD READ REVERSE
2599 007646 001344          BNE    RSEX                    ;IF NOT: BR
2600 007650 032737 010000 000562  BIT    #10000,RDCMD
2601 007656 001005          BNE    RSFR2                    ;IF READ REVERSE: BR
2602 007660 013737 000600 000670  MOV    TSTAL,STAL          ;DO READ STALL

```

TM03/TU45 DATA RELIABILITY PROGRAM
CZTURA.P11 13-JUN-78 10:30

MACY11 30(1046) 22-JUN-78 09:14 PAGE 58

SEQ 0058

```
2603 007666 004737 012124
2604 007672 012737 010000 000562 RSFR2: MOV PC,STALL
2605 007700 004737 007736 JSR #10000,RDCMD ;LOAD READ REVERSE
2606 007704 005737 000662 TST PC,READ ;GO READ REVERSE
2607 007710 100011 BPL EOTREC ;SEE IF AT END OF TAPE
2608 007712 163737 000554 000662 RSFRX ;IF NOT: BR
2609 007720 005437 000662 SUB RCNT,EOTREC
2610 007724 005237 000662 NEG EOTREC ;SET TO PROPER RECORD NUMBER
2611 007730 000137 004330 INC EOTREC
2612 007734 000207 RSFRX: JMP REOT ;ELSE GO TO REWIND
2613 RTS PC ;EXIT
```

```

2614                                     ;*****
2615                                     ;READ ROUTINE:
2616                                     ;
2617                                     ;THIS ROUTINE PERFORMS THE READ OPERATION DETERMINED
2618                                     ;BY THE READ SEQUENCE ROUTINE ONE RECORD AT A TIME.
2619                                     ;AT THE END OF EACH READ OPERATION THE STATUS REGISTER
2620                                     ;IS SCANNED FOR EITHER END OF TAPE OR BEGINNING OF TAPE.
2621                                     ;IF EOT WAS REACHED, CONTROL WILL BE PASSED TO
2622                                     ;THE EOT SUBROUTINE TO REWIND THE UNIT AND FLAG IT
2623                                     ;UNAVAILABLE UNTIL ALL UNITS HAVE REACHED EOT.
2624                                     ;IF BOT WAS REACHED AN ERROR IS PRINTED AND THE
2625                                     ;PROGRAM WILL HALT. TESTING MAY BE RESUMED BY PRESSING
2626                                     ;THE CONTINUE SWITCH.
2627                                     ;IF A TAPE MARK IS EXPECTED (TM=1) THEN THE
2628                                     ;READ ROUTINE EXPECTS THE FIRST RECORD OF A
2629                                     ;READ REVERSE TO BE A TM, AND THE LAST RECORD
2630                                     ;OF A READ FORWARD TO BE A TM. REMEMBER
2631                                     ;THAT THE TM ADDS ONE (1) TO THE TOTAL NUMBER
2632                                     ;OF RECORDS IN A BLOCK.
2633                                     ;CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13) DETERMINE WHETHER
2634                                     ;OR NOT TO CHECK FOR STATUS ERRORS (11) OR DATA ERRORS (13),
2635                                     ;CONSOLE SWITCH FIVE (5) IS USED TO CAUSE A CONTINUOUS
2636                                     ;READ AND SPACE (FORWARD OR REVERSE) OF THE CURRENT
2637                                     ;RECORD ON TAPE (YOZZLE).
2638                                     ;*****
2639
2640 007736 013700 000554      READ:  MOV     RCNT,R0      ;LOAD REC CNTR
2641 007742 005737 000662      TST     EOTREC      ;SEE IF EOT
2642 007746 100013      BPL     RDA      ;IF NOT: BR
2643 007750 032737 010000 000562  BIT     #10000,RDCMD  ;SEE IF READ FORWARD
2644 007756 001407      BEQ     RDA      ;IF SO: BR
2645 007760 042737 100000 000662  BIC     #100000,EOTREC ;CLEAR FLAG
2646 007766 013703 000662      MOV     EOTREC,R3     ;GET MODIFIED RECORD COUNT
2647 007772 160300      SUB     R3,R0      ;SET RECORD AT
2648 007774 005200      INC     R0      ;SET TO PROPER NUMBER OF RECORDS
2649 007776 012737 024507 000654  RDA:  MOV     #MSG6,EMADDR ;SET ERROR MSG ADDRESS
2650 010004 005037 000700      CLR     TMFLG
2651 010010 032737 010000 000562  BIT     #10000,RDCMD
2652 010016 001406      BEQ     R00      ;IF READ FORWARD: BR
2653 010020 005737 000564      TST     TMEX      ;SEE IF TM
2654 010024 001403      BEQ     R00      ;IF NOT: BR
2655 010026 005237 000700      INC     TMFLG      ;SET TM FLAG
2656 010032 005200      INC     R0
2657 010034 013777 000556 170454  R00:  MOV     FMCNT,@FC      ;LOAD CHAR CNTR
2658 010042 012777 033354 170444  MOV     #RDATA,@BA      ;LOAD DATA ADDR
2659 010050 032737 010000 000562  BIT     #10000,RDCMD  ;SEE IF READ REVERSE
2660 010056 001417      BEQ     RD1A      ;IF NOT: BR
2661 010060 013703 000556      MOV     FMCNT,R3
2662 010064 005103      COM     R3
2663 010066 032737 000020 000552  BIT     #20,UDES      ;SEE IF CORE DUMP
2664 010074 001402      BEQ     RD1      ;IF NOT: BR
2665 010076 000241      CLC
2666 010100 006003      ROR     R3      ;R3 = FC/2
2667 010102 060377 170406      RD1:  ADD     R3,@BA      ;SET REVERSE BUS ADDRESS
2668 010106 012737 000076 000674  MOV     #76,MT1      ;SET READ REVERSE
2669 010114 000403      BR      RD1B

```

2670	010116	012737	000070	000674	RD1A:	MOV	#70, MTC1	;SET READ FORWARD
2671	010124	012737	010136	000664	RD1B:	MOV	#RD2, RTRN	;SET INTERRUPT RETURN ADDRESS
2672	010132	000137	021072		RD1D:	JMP	TAPG	;GO EXECUTE TAPE COMMAND
2673	010136	032737	010000	000562	RD2:	BIT	#10000, RDCMD	;SEE IF READ REVERSE
2674	010144	001014				BNE	RD3	;IF SO: BR
2675	010146	032777	002000	170346		BIT	#2000, ADS	;SEE IF EOT
2676	010154	001410				BEQ	RD3	;IF NOT: BR
2677	010156	005737	000700			TST	TMFLG	;SEE IF TM
2678	010162	001005				BNE	RD3	;IF SO: BR
2679	010164	010037	000662			MOV	R0, EOTREC	
2680	010170	052737	100000	000662		BIS	#100000, EOTREC	;SET EOT FLAG
2681	010176	032777	000002	170316	RD3:	BIT	#2, ADS	;SEE IF AT LOAD POINT
2682	010204	001410				BEQ	RD4	;IF NOT: BR
2683	010206	004737	022602			JSR	PC, PAPRT	;PRINT CYCLE NUMBER
2684	010212	012704	024710			MOV	#MSG22, R4	
2685	010216	000004				TYPE		;TYPE MSG
2686	010220	000000				HALT		
2687	010222	000137	003160			JMP	STARTA	;RESTART
2688	010226	032777	004000	170354	RD4:	BIT	#4000, ASWR	;SEE IF SHOULD CHECK ERRORS
2689	010234	001121				BNE	RD5	;IF NOT: BR
2690	010236	005737	000700			TST	TMFLG	
2691	010242	001472				BEQ	RD4B	;IF NO TM EXPT: BR
2692	010244	032777	000004	170250		BIT	#4, ADS	
2693	010252	001024				BNE	RD4A	;IF TM RECVD: BR
2694	010254	012737	033354	021012		MOV	#RDATA, CADER	;SAVE EXPT BUS ADDRESS
2695	010262	012737	000002	021020		MOV	#2, DRIVER	;SET TM STATUS ERROR FLAG
2696	010270	004737	020064			JSR	PC, ERPT	;GO PRINT TM ERROR
2697	010274	013704	000676			MOV	UNP, R4	
2698	010300	032737	010000	000562		BIT	#10000, RDCMD	;SEE IF READ REVERSE
2699	010306	001403				BEQ	1\$	;IF NOT: BR
2700	010310	005264	001150			INC	RDERR1(R4)	;BUMP READ REVERSE ERROR
2701	010314	000502				BR	RD6	
2702	010316	005264	001110		1\$:	INC	RDER1(R4)	;BUMP READ FORWARD ERROR
2703	010322	000477				BR	RD6	
2704	010324	012703	033354		RD4A:	MOV	#RDATA, R3	
2705	010330	032737	010000	000562		BIT	#10000, RDCMD	;SEE IF READ REVERSE
2706	010336	001007				BNE	RD4A0	;IF SO: BR
2707	010340	032737	002000	000552		BIT	#2000, UDES	;SEE IF IN PE
2708	010346	001025				BNE	RD4A2	;IF SO: BR
2709	010350	062703	000002			ADD	#2, R3	
2710	010354	000422				BR	RD4A2	
2711	010356	013704	000556		RD4A0:	MOV	FM CNT, R4	
2712	010362	005104				COM	R4	
2713	010364	032737	000020	000552		BIT	#20, UDES	;SEE IF CORE DUMP
2714	010372	001402				BEQ	RD4A1	;IF NOT: BR
2715	010374	000241				CLC		
2716	010376	006004				ROR	R4	;SET TO FC/2
2717	010400	060403			RD4A1:	ADD	R4, R3	;SET EXPT BUS ADDRESS
2718	010402	042703	000001			BIC	#1, R3	;MAKE EXPT ADDRESS EVEN
2719	010406	032737	002000	000552		BIT	#2000, UDES	;SEE IF IN PE
2720	010414	001002				BNE	RD4A2	;IF SO: BR
2721	010416	162703	000002			SUB	#2, R3	
2722	010422	004737	017340		RD4A2:	JSR	PC, ER2	
2723	010426	000402				BR	RD4C	
2724	010430	004737	017244		RD4B:	JSR	PC, ERCHK	;GO CHECK ERRORS
2725	010434	005737	000710		RD4C:	TST	SERFL	

2726	010440	001417			BEQ	RD5		;IF NO ERROR: BR
2727	010442	013704	000676		MOV	UNP,R4		
2728	010446	032737	010000	000562	BIT	#10000,RDCMD		;SEE IF READ REVERSE
2729	010454	001003			BNE	RD4D		;IF SO: BR
2730	010456	005264	001110		INC	RDER1(R4)		;BUMP READ FORWARD ERROR
2731	010462	000402			BR	RD4E		
2732	010464	005264	001150		RD4D: INC	RDERR1(R4)		;BUMP READ REVERSE ERROR
2733	010470	004737	010672		RD4E: JSR	PC,RDRTY		;GO RETRY
2734	010474	005037	000714		CLR	RTYFL		;CLEAR RETRY FLAG
2735	010500	032777	020000	170102	RD5: BIT	#20000,@SWR		;SEE IF SHOULD DO DATA CHECK
2736	010506	001005			BNE	RD6		;IF NOT: BR
2737	010510	005737	000700		TST	TMFLG		
2738	010514	001002			BNE	RD6		
2739	010516	004737	015374		JSR	PC,DCHK		;GO CHECK DATA
2740	010522	005037	000710		RD6: CLR	SERFL		;CLEAR STATUS ERROR FLAG
2741	010526	004737	014240		JSR	PC,DS3		;CLEAR BUFFER
2742	010532	032777	000040	170050	BIT	#40,@SWR		;SEE IF SHOULD YOZZLE
2743	010540	001402			BEQ	RD7		;IF NOT: BR
2744	010542	004737	011260		JSR	PC,YOZ		;ELSE GO YOZZLE
2745	010546	013737	000574	000670	RD7: MOV	RSTAL,STAL		;SET DELAY
2746	010554	004737	012124		JSR	PC,STALL		;STALL
2747	010560	032737	010000	000562	BIT	#10000,RDCMD		;SEE IF READ REVERSE
2748	010566	001403			BEQ	RD7A		;IF NOT: BR
2749	010570	005037	000700		CLR	TMFLG		;CLEAR TAPE MARK FLAG
2750	010574	000405			BR	RD10		
2751	010576	005737	000662		RD7A: TST	EOTREC		;SEE IF EOT FOUND
2752	010602	100002			BPL	RD10		;IF NOT: BR
2753	010604	012700	000001		MOV	#1,R0		;SET TO EOT
2754	010610	005300			RD10: DEC	R0		
2755	010612	001402			BEQ	RD11		;IF DONE ALL: BR
2756	010614	000137	010034		JMP	RDO		
2757	010620	032737	010000	000562	RD11: BIT	#10000,RDCMD		;SEE IF READ REVERSE
2758	010626	001016			BNE	RDEX		;IF SO: BR
2759	010630	005737	000662		TST	EOTREC		;SEE IF FOUND EOT
2760	010634	100413			BMI	RDEX		;IF SO: BR
2761	010636	005737	000564		TST	TMEX		;SEE IF TM EXPECTED
2762	010642	001410			BEQ	RDEX		;IF NOT: BR
2763	010644	005737	000700		TST	TMFLG		;SEE IF TM FOUND
2764	010650	001005			BNE	RDEX		;IF SO: BR
2765	010652	005237	000700		INC	TMFLG		;ELSE SET FLAG
2766	010656	005200			INC	R0		;SET RECORD COUNT TO ONE
2767	010660	000137	010034		JMP	RDO		;GO READ TM
2768	010664	005037	000700		RDEX: CLR	TMFLG		
2769	010670	000207			RDX: RTS	PC		;EXIT

```

2770 ;*****
2771 ;READ ERROR RETRY SUBROUTINE:
2772 ;
2773 ;THIS SUBROUTINE WILL RETRY ALL DATA RELATED
2774 ;READ ERRORS UP TO EIGHT (8) TIMES. IF ALL
2775 ;FOUR RETRIES ARE BAD, IT IS CONSIDERED
2776 ;A HARD ERROR. IF ANY ARE GOOD, IT IS A
2777 ;SOFT ERROR. RETRIES MAY BE INHIBITED
2778 ;VIA SWITCH FOUR (SW4=0: INHIBIT RETRIES)
2779 ;*****
2780
2781 010672 032777 000020 167710 RDRTY: BIT #20,@SWR ;SEE IF RETRY INHIBITED
2782 010700 001001 BNE RDRT0 ;IF NOT: BR
2783 010702 000207 RTS PC ;ELSE RETURN
2784 010704 013703 000724 RDRT0: MOV ERSAV,R3
2785 010710 042703 102700 BIC #102700,R3 ;MARK NON-RECOVERABLE ERROR BITS
2786 010714 001410 BEQ RDRT1 ;IF NOT: BR
2787 010716 004737 022602 JSR PC,PAPRT ;PRINT HEADER
2788 010722 012704 026465 MOV #MSG79,R4
2789 010726 000004 TYPE ;TYPE MSG
2790 010730 004737 011244 JSR PC,NRTP ;PRINT ER FOR NON-RETRYABLE ERROR
2791 010734 000207 RDRT1A: RTS PC ;RETURN
2792 010736 032777 002000 167644 RDRT1: BIT #2000,@SWR ;SEE IF PRINT INHIBITED
2793 010744 001003 BNE RDRT1B ;IF SO: BR
2794 010746 012704 026140 MOV #MSG64,R4
2795 010752 000004 TYPE ;TYPE MSG
2796 010754 005037 000704 RDRT1B: CLR RTCNT ;CLEAR RETRY COUNTER
2797 010760 005037 000710 RDRTG: CLR SERFL ;CLEAR STATUS ERROR FLAG
2798 010764 012737 000002 000714 MOV #2,RTYFL ;SET READ RETRY FLAG
2799 010772 004737 011260 JSR PC,YOZ ;GO TO YOZZLE TO RETRY READ
2800 010776 005737 000710 TST SERFL ;SEE IF RETRY ERROR
2801 011002 001031 BNE RDRT5 ;IF SO: BR
2802 011004 032777 002000 167576 BIT #2000,@SWR
2803 011012 001011 BNE RDRT2
2804 011014 012704 026632 MOV #MSG105,R4
2805 011020 000004 TYPE ;TYPE MSG
2806 011022 012704 026162 MOV #MSG65,R4
2807 011026 000004 TYPE ;TYPE MSG
2808 011030 013703 000704 MOV RTCNT,R3
2809 011034 104400 TYPOCT ;PRINT RETRY NUMBER
2810 011036 013704 000676 RDRT2: MOV UNP,R4
2811 011042 032737 010000 000562 BIT #10000,RDCMD ;SEE IF READ REVERSE
2812 011050 001003 BNE RDRT3 ;IF SO: BR
2813 011052 005264 002670 INC RFSOFT(R4) ;ELSO BUMP FORWARD SOFT ERROR COUNTER
2814 011056 000402 BR RDRT4
2815 011060 005264 002710 RDRT3: INC RRSOFT(R4) ;BUMP ERRORS SOFT CNTR
2816 011064 000207 RDRT4: RTS PC ;RETURN
2817 011066 013703 000724 RDRT5: MOV ERSAV,R3 ;GET ER
2818 011072 005037 000650 CLR TEMP3 ;CLEAR RECOVERABLE ERROR INDICATOR
2819 011076 042703 102700 BIC #102700,R3 ;MASK RECOVERABLE BITS
2820 011102 001413 BEQ RDRT5A ;IF RECOVERABLE: BR
2821 011104 004737 022602 JSR PC,PAPRT ;PRINT HEADER
2822 011110 012704 026465 MOV #MSG79,R4
2823 011114 000004 TYPE ;TYPE MSG
2824 011116 004737 011244 JSR PC,NRTP ;PRINT ER
2825 011122 012737 000001 000650 MOV #1,TEMP3 ;SET FLAG

```

```

2826 011130 000404
2827 011132 032777 002000 167450 RDRT5A: BR RDRT5B
2828 011140 001014 BNE #2000,ASWR ;SEE IF PRINT INHIBITED
2829 011142 012704 026162 RDRT5B: MOV RDRT6 ;IF SO: BR
2830 011146 000004 TYPE #MSG65,R4 ;TYPE MSG
2831 011150 013703 000704 MOV RTCNT,R3
2832 011154 104400 TYPOCT ;PRINT RETRY NUMBER
2833 011156 005737 000650 TST TEMP3 ;SEE IF DID NON-RECOVERABLE
2834 011162 001403 BEQ RDRT6 ;IF NOT: BR
2835 011164 005037 000650 CLR TEMP3 ;CLEAR FLAG
2836 011170 000207 RTS PC ;EXIT
2837 011172 005237 000704 RDRT6: INC RTCNT
2838 011176 023737 000704 000604 CMP RTCNT,RETRY ;SEE IF DONE 8 RETRIES
2839 011204 001265 BNE RDRTG ;IF NOT: BR
2840 011206 012704 027150 MOV #MSG115,R4
2841 011212 000004 TYPE ;TYPE MSG
2842 011214 013704 000676 MOV UNP,R4
2843 011220 032737 010000 000562 BIT #10000,RDCMD ;SEE IF READ REVERSE
2844 011226 001003 BNE RDRT7 ;IF SO: BR
2845 011230 005264 002730 INC RFHARD(R4) ;BUMP FORWARD HARD ERROR CNTR
2846 011234 000402 BR RDRTX
2847 011236 005264 002750 RDRT7: INC RRHARD(R4) ;BUMP REVERSE HARD ERROR CNTR
2848 011242 000207 RDRTX: RTS PC ;RETURN
2849
2850 011244 013703 000724 NRTP: MOV ERSAR,R3 ;GET ER REGISTER
2851 011250 104400 TYPOCT ;PRINT ER
2852 011252 004737 021036 JSR PC,FRPRT ;PRINT F OR R
2853 011256 000207 RTS PC ;RETURN

```

```

2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867 011260 013737 000602 000670 YOZ:  MOV YSTAL,STAL
2868 011266 004737 012124          JSR  PC,STALL      ;DO YOZZLE STALL
2869 011272 012777 177777 167216 YOZO:  MOV  #-1,@FC      ;SET TO 1 RECORD SPACING
2870 011300 032737 010000 000562        BIT  #10000,RDCMD  ;SEE IF READ REVERSE
2871 011306 001404          BEQ  YOZA      ;IF NOT: BR
2872 011310 112737 000030 000674        MOVB #30,MTC1    ;SET TO SPACE FORWARD
2873 011316 000403          BR    YOZB
2874 011320 112737 000032 000674 YOZA:  MOVB #32,MTC1    ;SET TO SPACE REVERSE
2875 011326 012737 011346 000664 YOZB:  MOV  #YOZC,RTRN  ;SET RETURN ADDRESS
2876 011334 012737 177775 000670        MOV  #177775,STAL ;SET TIME MULTIPLIER
2877 011342 000137 021072          JMP  TAPG      ;GO YOZZLE
2878 011346 005737 000700          YOZC:  TST  TMFLG      ;SEE IF TM
2879 011352 001404          BEQ  1$        ;IF NOT: BR
2880 011354 012737 040000 000670        MOV  #40000,STAL ;SET TM STALL
2881 011362 000403          BR    2$
2882 011364 013737 000602 000670 1$:  MOV  YSTAL,STAL
2883 011372 004737 012124          2$:  JSR  PC,STALL  ;DO YOZZLE STALL
2884 011376 012777 033354 167110        MOV  #RDATA,@BA  ;SET BUS ADDRESS
2885 011404 032737 010000 000562        BIT  #10000,RDCMD  ;SEE IF READ REVERSE
2886 011412 001416          BEQ  YOZC1    ;IF NOT: BR
2887 011414 013703 000556          MOV  FMCNT,R3
2888 011420 005103          COM  R3
2889 011422 032737 000020 000552        BIT  #20,UDES    ;SEE IF CORE DUMP
2890 011430 001401          BEQ  YOZC0    ;IF NOT: BR
2891 011432 006203          ASR  R3        ;R3 = FC/2
2892 011434 060377 167054          YOZC0: ADD  R3,@BA    ;SET REVERSE BUS ADDRESS
2893 011440 012737 000076 000674        MOV  #76,MTC1    ;SET READ REVERSE
2894 011446 000403          BR    YOZC2
2895 011450 012737 000070 000674 YOZC1: MOV  #70,MTC1    ;SET READ FORWARD
2896 011456 013777 000556 167032 YOZC2: MOV  FMCNT,@FC  ;SET CHARACTER COUNT
2897 011464 012737 011476 000664        MOV  #YOZD,RTRN  ;SET RETURN ADDRESS
2898 011472 000137 021072          JMP  TAPG      ;GO READ
2899 011476 032777 004000 167104 YOZD:  BIT  #4000,@SWR  ;SEE IF SHOULD CHECK ERRORS
2900 011504 001050          BNE  YOZE      ;IF NOT: BR
2901 011506 005737 000700          TST  TMFLG      ;SEE IF TAPE MARK TIME
2902 011512 001443          BEQ  YOZD1    ;IF NOT: BR
2903 011514 032737 010000 000562        BIT  #10000,RDCMD  ;SEE IF READ REVERSE
2904 011522 001425          BEQ  YOZD0    ;IF NOT: BR
2905 011524 012703 033354          MOV  #RDATA,R3
2906 011530 013704 000556          MOV  FMCNT,R4
2907 011534 005104          COM  R4
2908 011536 032737 000020 000552        BIT  #20,UDES    ;SEE IF CORE DUMP
2909 011544 001401          BEQ  YOZD4    ;IF NOT: BR

```

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
---------	-----	-----	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	------

```

2938
2939
2940      ;*****
2941      ;BACKSPACE SUBROUTINE:
2942      ;
2943      ;THIS SUBROUTINE IS USED TO PERFORM THE
2944      ;BACKSPACE OPERATION REQUIRED BY THE READ
2945      ;ROUTINE FOR READ FORWARD AFTER WRITING.
2946      ;IF A TAPE MARK IS EXPECTED (TM=1) THEN THE SPACE
2947      ;ROUTINE ASSUMES THAT THE TM WILL BE FIRST WHEN
2948      ;BACKSPACING. THEREFORE TWO OPERATIONS ARE REQUIRED
2949      ;TO SPACE OVER A BLOCK. FIRST SPACE OVER THE TM, THEN
2950      ;SPACE OVER THE DATA RECORDS.
2951      ;A CHECK FOR RECORD COUNT ZERO IS MADE AT THE
2952      ;END OF THE SPACE OPERATION TO ASSURE THAT PROPER
2953      ;TAPE POSITIONING WAS DONE.
2954      ;*****
2955 011700 013737 000600 000670 BKSP: MOV TSTAL,STAL
2956 011706 004737 012124 JSR PC,STALL ;DO TURN AROUND STALL
2957 011712 012737 024537 000654 MOV #MSG10,EMADDR
2958 011720 012777 033354 166566 MOV #RDATA,@BA
2959 011726 005737 000564 TST TMEX ;SEE IF TM
2960 011732 001440 BEQ B0 ;IF NOT: BR
2961 011734 012777 177777 166554 MOV #-1,@FC
2962 011742 012737 000032 000674 MOV #32,MTC1
2963 011750 012737 011762 000664 MOV #BKTM,RTRN
2964 011756 000137 021072 JMP TAPG ;SPACE TO TM
2965 011762 032777 010000 166620 BKTM: BIT #10000,@SWR ;SEE IF SHOULD CHECK ERROR
2966 011770 001021 BNE B0 ;IF NOT: BR
2967 011772 012737 026054 000654 MOV #MSG55,EMADDR
2968 012000 032777 000004 166514 BIT #4,@DS ;SEE IF TM
2969 012006 001006 BNE BKTM0 ;IF SO: BR
2970 012010 012737 033354 021012 MOV #RDATA,CADER
2971 012016 004737 020064 JSR PC,ERPT ;PRINT ERROR
2972 012022 000404 BR B0
2973 012024 012703 033354 BKTM0: MOV #RDATA,R3
2974 012030 004737 017340 JSR PC,ER2
2975 012034 013700 000554 B0: MOV RCNT,R0
2976 012040 005400 NEG R0 ;BUILD SPACE AMOUNT
2977 012042 012737 024537 000654 MOV #MSG10,EMADDR ;SET ERROR MESSG ADDRESS
2978 012050 010077 166442 MOV R0,@FC
2979 012054 012737 000032 000674 BKRT: MOV #32,MTC1 ;SET SPACE REVERSE
2980 012062 012737 012100 000664 MOV #B1,RTRN ;SET RETURN ADDRESS
2981 012070 010037 000670 MOV R0,STAL ;SET INTERRUPT TIME MULTIPLIER
2982 012074 000137 021072 JMP TAPG ;GO DO SPACE
2983 012100 012703 033354 B1: MOV #RDATA,R3
2984 012104 004737 017340 JSR PC,ER2
2985 012110 013737 000600 000670 B2: MOV TSTAL,STAL ;DO STALL
2986 012116 004737 012124 JSR PC,STALL ;STALL
2987 012122 000207 RTS PC ;EXIT
2988

```

2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007

```
*****  
;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.  
*****
```

3008 012124 005337 000670
3009 012130 001375
3010 012132 000207

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

```

3011
3012
3013 ;*****
3014 ;RANDOM CHARACTER COUNT GENERATOR:
3015 ;
3016 ;THIS ROUTINE ENTERED VIA CONSOLE SWITCH
3017 ;SEVEN (7) IS USED TO GENERATE A RANDOM
3018 ;CHARACTER COUNT FOR EACH DATA BLOCK.
3019 ;ALL RECORDS WITHIN A GIVEN BLOCK WILL BE
3020 ;THE SAME, BUT EACH BLOCK WILL VARY.
3021 ;THE LIMITS ARE TWENTY (20) TO FOUR THOUSAND
3022 ; (4000) OCTAL CHARACTERS PER RECORD.
3023 ;*****
3024 012134 012701 177760      CCNTR: MOV    #-20,R1      ;SET HIGH LIMIT
3025 012140 012702 175000      MOV    #-3000,R2    ;SET LOW LIMIT
3026 012144 004737 023164      JSR    PC,RANG     ;GO GENERATE NUMBER
3027 012150 042737 000001 000630 BIC    #1,RANSAV
3028 012156 013737 000630 000556 MOV    RANSAV,FMCNT ;SET CHAR COUNT
3029 012164 012737 177777 014276 MOV    #-1,PATS    ;PRESET DATA PATTERN
3030 012172 000207              RTS    PC          ;EXIT
3031
3032 ;*****
3033 ;RANDOM RECORD COUNT GENERATOR:
3034 ;
3035 ;THIS ROUTINE ENTERED VIA CONSOLE SWITCH SIX (6)
3036 ;IS USED TO GENERATE A RANDOM NUMBER OF RECORDS
3037 ;FOR EACH BLOCK OF DATA.
3038 ;THE LIMITS ARE ONE (1) TO FIVE HUNDRED (500) OCTAL
3039 ;RECORDS PER BLOCK.
3040 ;*****
3041
3042 012174 012702 000001      RCNTR: MOV    #1,R2      ;SET LOW LIMIT
3043 012200 012701 000500      MOV    #500,R1     ;SET HIGH LIMIT
3044 012204 004737 023164      JSR    PC,RANG     ;GO GENERATE NUMBER
3045 012210 013737 000630 000554 MOV    RANSAV,RCNT  ;SET RECORD COUNT
3046 012216 000207              RTS    PC          ;EXIT
3047
3048

```

```

3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095 012220 005737 000636
3096 012224 001002
3097 012226 000137 013706
3098 012232 005037 000676
3099 012236 005037 005054
3100 012242 012700 000010
3101 012246 012701 000746
3102 012252 005021
3103 012254 005300
3104 012256 001375

:*****
:TEST CONDITION ENTRY ROUTINE:
:
:THIS ROUTINE IS USED TO ALLOW THE OPERATOR
:TO ENTER, AT THE TTY, THE NECESSARY PARAMETERS
:TO RUN THE PROGRAM AS HE WISHES. THE
:ROUTINE IS ONLY ENTERED UPON INITIAL STARTING
:FROM LOCATION 200(8).
:THE MAIN PURPOSE OF THIS ROUTINE IS TO ESTABLISH
:A TABLE OF DEVICES TO BE TESTED. THIS TABLE
:CONSISTS OF AN ENTRY FOR EACH OF ONE (1) TO
:EIGHT (8) DEVICES. EACH ENTRY CONTAINS THE
:SLAVE NUMBER, DENSITY, PARITY, AND
:FORMAT. THE INFORMATION IS ENTERED
:IN RESPONSE TO PRINTED REQUESTS AT THE TTY.
:SLAVES MAY BE ENTERED IN ANY ORDER. EACH
:PARAMETER IS CHECKED FOR LEGALITY BEFORE BEING
:SET INTO THE TABLE.
:THE DRIVE NUMBER REQUEST WILL ALSO CHECK THE MASSBUS
:FOR THE PRESENCE OF THE REQUESTED DRIVE. IF IT IS NOT FOUND,
:A NON-EXIST DRIVE MESSAGE WILL BE PRINTED AND ANOTHER DRIVE
:REQUEST MADE. WHEN THE DRIVE IS FOUND, THE RESPONSE IS STORED
:AND CONTROL PASSED TO THE SLAVE SELECT ROUTINE.
:THE SLAVE SELECT ROUTINE ALSO CHECKS FOR THE PRESENCE OF THE
:SLAVE. IF IT IS NOT PRESENT, A MESSAGE IS PRINTED AND ANOTHER
:REQUEST IS ISSUED. WHEN THE SELECTED SLAVE IS FOUND TO BE
:PRESENT, A MESSAGE IS PRINTED IF IT IS A 7 CHANNEL DRIVE
:TO ASSIST IN SELECTING DENSITY, PARITY, AND FORMAT.
:UPON COMPLETION OF THE DEVICE TABLE, REQUESTS
:ARE PRINTED FOR ENTRY OF THE NUMBER OF CHARACTERS
:PER RECORD AND THE NUMBER OF RECORDS PER BLOCK. THE
:NEXT REQUEST IS FOR A PATTERN NUMBER TO BE USED
:FOR WRITING AND CHECKING OF READ DATA.
:FOLLOWING THE PATTERN REQUEST IS THE TAPE MARK OPTION.
:RESPONDING TO THE REQUEST (TM=) WITH A ONE (1)
:WILL CAUSE THE PROGRAM TO WRITE A TM AT THE
:END OF EACH DATA BLOCK AND TO EXPECT THE
:TM TO BE DETECTED IN EITHER READ FORWARD AND REVERSE
:OR DURING SPACE OPERATION. A RESPONSE OF ZERO (TM=0)
:DISALLOWS WRITTING OF THE TM AND CAUSES THE READ
:AND SPACE ROUTINES TO EXPECT NO TM TO BE PRESENT.
:THE LAST REQUESTS ARE FOR ENTRY OF THE DESIRED
:WRITE, READ, AND TURN AROUND STALLS.
:*****

TINP: TST TINF ;SEE IF SHOULD INPUT FROM TTY
      BNE 1$ ;IF SO: BR
      JMP TINP4 ;GET SWITCHES
1$: CLR UNP ;CLEAR TABLE POINTER
   CLR REOTC ;CLEAR EOT UNIT COUNTER
   MOV #10,R0 ;SET SIZE OF TABLE
   MOV #UN1,R1 ;SET START OF TABLE
3$: CLR (R1)+ ;CLEAR TABLE
   DEC R0 ;SEE IF DONE
   BNE 3$ ;IF NOT: BR

```

3105	012260	012704	025200		MOV	#MSG31,R4	
3106	012264	005737	000736		TST	ASEQF	;SEE IF AUTO SEQ
3107	012270	001402			BEQ	4\$	;IF NOT: BR
3108	012272	012704	025133		MOV	#MSG30,R4	;SET AUTO SEQ HDR
3109	012276	010446		4\$:	MOV	R4,-(SP)	;SAVE ADDRESS OF MESSAGE
3110	012300	000004			TYPE		;TYPE MSG
3111	012302	105036			CLRB	@(SP)+	;DO NOT TYPE TITLE ON RESTART
3112	012304	012704	025255		MOV	#MSG31A,R4	;TYPE INSTRUCTION
3113	012310	000004			TYPE		
3114	012312	105037	025255		CLRB	MSG31A	;DO NOT TYPE STARTUP INSTRUCTIONS ON RESTART
3115	012316	005737	014070		TST	SCVFL	;SEE IF SHORT CONVERSATION
3116	012322	001067			BNE	6\$	;IF SO: BR
3117	012324	012704	026344		MOV	#MSG74,R4	
3118	012330	000004			TYPE		;TYPE MSG
3119	012332	013703	000544		MOV	REGS,R3	
3120	012336	104400			TYPOCT		;PRINT CURRENT REG START
3121	012340	012705	000544		MOV	#REGS,R5	;SAVE ADDRESS LOCATION
3122	012344	012701	000007		MOV	#7,R1	;SET SIZE OF ENTRY
3123	012350	012702	176400		MOV	#176400,R2	;SET UPPER LIMIT
3124	012354	012703	172300		MOV	#172300,R3	;SET LOWER LIMIT
3125	012360	004737	023352		JSR	PC,TTR	;GO GET RESPONSE
3126	012364	012704	026367		MOV	#MSG75,R4	
3127	012370	000004			TYPE		;TYPE MSG
3128	012372	013703	000546		MOV	VECT,R3	
3129	012376	104400			TYPOCT		;PRINT CURRENT VECTOR
3130	012400	012705	000546		MOV	#VECT,R5	;SET SAVE LOCATION
3131	012404	012701	000004		MOV	#4,R1	;SET SIZE OF ENTRY
3132	012410	012702	000224		MOV	#224,R2	;SET UPPER LIMIT
3133	012414	012703	000150		MOV	#150,R3	;SET LOWER LIMIT
3134	012420	004737	023352		JSR	PC,TTR	;GO GET RESPONSE
3135	012424	013700	000546		MOV	VECT,R0	;GET VECTOR ADDRESS
3136	012430	012720	021734		MOV	#MTINT,(R0)+	;LOAD VECTOR WITH HANDLER ADDRESS
3137	012434	012710	000340		MOV	#340,(R0)	;LOAD PRIORITY LEVEL
3138	012440	013700	000544		MOV	REGS,R0	;GET STARTING REGISTER ADDRESS
3139	012444	012701	000016		MOV	#16,R1	;SET NUMBER OF REGISTERS
3140	012450	012702	000510		MOV	#C1,R2	;GET FIRST ADDRESS LOCATION
3141	012454	010022		5\$:	MOV	R0,(R2)+	;BUILD TABLE OF ADDRESSES
3142	012456	062700	000002		ADD	#2,R0	;BUMP ADDRESS
3143	012462	005301			DEC	R1	;SEE IF DONE
3144	012464	001373			BNE	5\$	;IF NOT: BR
3145	012466	005737	000736		TST	ASEQF	;SEE IF AUTO SEQ
3146	012472	001403			BEQ	6\$	;IF NOT: BR
3147	012474	005726			TST	(SP)+	;RESET STACK POINTER
3148	012476	000137	021752		JMP	ASEQ	;GO TO AUTO SEQUENCE
3149	012502	012777	000040	166010	6\$:	MOV	#40,@CS
3150	012510	012704	026011		MOV	#MSG52,R4	
3151	012514	000004			TYPE		;TYPE MSG
3152	012516	012705	000550		MOV	#DVN,R5	;GET ADDRESS
3153	012522	012701	000002		MOV	#2,R1	;SET SIZE OF RESPONSE
3154	012526	012702	000007		MOV	#7,R2	;SET UPPER LIMIT
3155	012532	012703	000000		MOV	#0,R3	;SET LOWER LIMIT
3156	012536	004737	023352		JSR	PC,TTR	;GO GET DRIVE NUMBER
3157	012542	013777	000550	165750	MOV	DVN,@CS	
3158	012550	005777	165734		TST	@C1	;ACCESS DRIVE
3159	012554	032777	010000	165736	BIT	#10000,@CS	;SEE IF NED
3160	012562	001411			BEQ	TINPO	;IF NOT: BR

3161	012564	012704	026301		MOV	#MSG71,R4	
3162	012570	000004			TYPE		;TYPE MSG
3163	012572	013704	000510		MOV	C1,R4	
3164	012576	005204			INC	R4	
3165	012600	152714	000100		BISB	#100,(R4)	;CLEAR TRE
3166	012604	000736			BR	6\$	;RETRY DVN
3167	012606	012704	025342	TINPO:	MOV	#MSG32,R4	
3168	012612	000004			TYPE		;TYPE MSG
3169	012614	005037	000646		CLR	TEMP2	;CLEAR BUFFER
3170	012620	012705	000646		MOV	#TEMP2,R5	;SET UNIT DESCRIPTION BUFFER ADDRESS
3171	012624	012701	000002		MOV	#2,R1	;SET NUMBER OF CHARACTERS TO INPUT
3172	012630	012702	000007		MOV	#7,R2	;SET MAXIMUM LIMIT
3173	012634	012703	000000		MOV	#0,R3	;SET MINIMUM LIMIT
3174	012640	004737	023352		JSR	PC,TTR	;GO GET UNIT NUMBER
3175	012644	005737	000644		TST	TEMP1	;SEE IF HAVE NEW PARAMETER
3176	012650	001012			BNE	TINPOB	;IF SO: BR
3177	012652	005737	000676		TST	UNP	;SEE IF FIRST ENTRY
3178	012656	001753			BEQ	TINPO	
3179	012660	013700	000676		MOV	UNP,R0	
3180	012664	012760	177777	000746	MOV	#-1,UN1(R0)	;SET END UNIT TABLE
3181	012672	000137	013272		JMP	TINP2C	;GO GET RECORD COUNT
3182	012676	013700	000676		TINPOB: MOV	UNP,R0	
3183	012702	042760	000007	000746	BIC	#7,UN1(R0)	;CLEAR UNIT NUMBER
3184	012710	004737	014102		JSR	PC,TPOS1	;GO LOAD UNIT NUMBER TO PROPER POSITION
3185	012714	012777	000040	165576	MOV	#40,ACS	
3186	012722	013777	000550	165570	MOV	DVN,ACS	
3187	012730	016077	000746	165604	MOV	UN1(R0),@TC	;LOAD UNIT NUMBER
3188	012736	032777	002000	165572	TINPOC: BIT	#2000,@DT	;SEE IF SLAVE PRESENT
3189	012744	001004			BNE	TINPOD	;IF SO: BR
3190	012746	012704	026067		MOV	#MSG57,R4	
3191	012752	000004			TYPE		;TYPE MSG
3192	012754	000714			BR	TINPO	;REDO
3193	012756	017703	165554		TINPOD: MOV	@DT,R3	;GET CONTENTS OF DT REG
3194	012762	042703	000007		BIC	#7,R3	;CLEAR DRIVE TYPE #
3195	012766	022703	142050		CMP	#142050,R3	;SEE IF 9TRK TM03,TU45
3196	012772	001410			BEQ	TINPOE	;IF SO: BR
3197	012774	012704	025764		MOV	#MSG50,R4	;ILLEGAL DRIVE TYPE
3198	013000	000004			TYPE		;TYPE MSG
3199	013002	017703	165530		MOV	@DT,R3	
3200	013006	042703	000007		BIC	#7,R3	;CLEAR SLAVE #
3201	013012	104400			TYPOCT		;PRINT DRIVE TYPE REGISTER
3202	013014	012704	024531		TINPOE: MOV	#MSG9,R4	
3203	013020	000004			TYPE		;TYPE MSG
3204	013022	017703	165512		MOV	@SN,R3	
3205	013026	004737	024334		JSR	PC,SNPT	;PRINT SERIAL NUMBER
3206	013032	012704	025363		TINP1: MOV	#MSG33,R4	
3207	013036	000004			TYPE		;TYPE MSG
3208	013040	005037	000646		CLR	TEMP2	;CLEAR BUFFER
3209	013044	012701	000002		MOV	#2,R1	;SET NUMBER OF CHARACTERS TO INPUT
3210	013050	012702	000004		MOV	#4,R2	;SET MAXIMUM LIMIT
3211	013054	012703	000003		MOV	#3,R3	;SET MINIMUM LIMIT
3212	013060	004737	023352		JSR	PC,TTR	;GO GET DENSITY
3213	013064	005737	000644		TST	TEMP1	;SEE IF HAVE NEW PARAMETER
3214	013070	001407			BEQ	TINP2	;IF NOT: BR
3215	013072	042737	003400	000552	BIC	#3400,UES	;ELSE CLEAR OLD PARAMETER
3216	013100	012703	000010		MOV	#10,R3	;SET POSITION FACTOR

3217	013104	004737	014072		JSR	PC,TPOS	:GO LOAD DENSITY INTO PROPER POSITION
3218	013110	012704	025377		TINP2: MOV	#MSG34,R4	
3219	013114	000004			TYPE		:TYPE MSG
3220	013116	005037	000646		CLR	TEMP2	:CLR BUFFER
3221	013122	012701	000002		MOV	#2,R1	:SET NUMBER OF CHARACTERS TO INPUT
3222	013126	012702	000001		MOV	#1,R2	:SET MAXIMUM LIMIT
3223	013132	012703	000000		MOV	#0,R3	:SET MINIMUM LIMIT
3224	013136	004737	023352		JSR	PC,TTR	:GO INPUT PARITY
3225	013142	005737	000644		TST	TEMP1	:SEE IF HAVE NEW PARAMETER
3226	013146	001407			BEQ	TINP2A	:IF NOT: BR
3227	013150	042737	000010	000552	BIC	#10,UDES	:ELSE CLEAR OLD PARAMETER
3228	013156	012703	000003		MOV	#3,R3	:SET POSITION FACTOR
3229	013162	004737	014072		JSR	PC,TPOS	:GO LOAD PARITY TO PROPER POSITION
3230	013166	012704	026032		TINP2A: MOV	#MSG53,R4	
3231	013172	000004			TYPE		:TYPE MSG
3232	013174	005037	000646		CLR	TEMP2	
3233	013200	012701	000003		MOV	#3,R1	
3234	013204	012702	000017		MOV	#17,R2	
3235	013210	012703	000000		MOV	#0,R3	
3236	013214	004737	023352		JSR	PC,TTR	:GO GET FORMAT
3237	013220	005737	000644		TST	TEMP1	:SEE IF NEW PARAMETER
3238	013224	001407			BEQ	TINP2B	:IF NOT: BR
3239	013226	042737	000170	000552	BIC	#170,UDES	
3240	013234	012703	000004		MOV	#4,R3	
3241	013240	004737	014072		JSR	PC,TPOS	
3242	013244	005237	005054		TINP2B: INC	REOTC	:BUMP EOT UNIT COUNTER
3243	013250	022737	000016	000676	CMP	#16,UNP	:SEE IF DONE UNITS
3244	013256	001405			BEQ	TINP2C	:IF SO: BR
3245	013260	062737	000002	000676	ADD	#2,UNP	:POINT TO NEXT UNIT
3246	013266	000137	012606		JMP	TINP0	:ELSE LOOK FOR NEXT UNIT
3247	013272	005037	000676		TINP2C: CLR	UNP	:CLEAR UNIT POINTER
3248	013276	013700	005054		MOV	REOTC,R0	
3249	013302	000337	005054		SWAB	REOTC	
3250	013306	110037	005054		MOVB	R0,REOTC	:SET UNIT EOT COUNTER
3251	013312	012704	025412		TINP3: MOV	#MSG35,R4	
3252	013316	000004			TYPE		:TYPE MSG
3253	013320	013703	000554		MOV	RCNT,R3	
3254	013324	104400			TYPOCT		:PRINT RECORD COUNT
3255	013326	012705	000554		MOV	#RCNT,R5	:SET RECORD COUNT ADDRESS
3256	013332	012701	000007		MOV	#7,R1	:SET NUMBER OF CHARACTERS TO INPUT
3257	013336	012702	177777		MOV	#-1,R2	:SET MAXIMUM LIMIT
3258	013342	012703	000001		MOV	#1,R3	:SET MINIMUM LIMIT
3259	013346	004737	023352		JSR	PC,TTR	:GO GET RECORD COUNT
3260	013352	013737	000554	000632	MOV	RCNT,RCSAV	:SAVE RECORD COUNT
3261	013360	012704	025433		MOV	#MSG36,R4	
3262	013364	000004			TYPE		:TYPE MSG
3263	013366	005437	000556		NEG	FMCNT	
3264	013372	013703	000556		MOV	FMCNT,R3	
3265	013376	104400			TYPOCT		:PRINT CHAR COUNT
3266	013400	012705	000556		MOV	#FMCNT,R5	:SET CHARACTER COUNT ADDRESS
3267	013404	012701	000007		MOV	#7,R1	:SET NUMBER OF CHARACTERS TO INPUT
3268	013410	012702	004000		MOV	#4000,R2	:SET MAXIMUM LIMIT
3269	013414	012703	000004		MOV	#4,R3	:SET MINIMUM LIMIT
3270	013420	004737	023352		JSR	PC,TTR	:GO GET CHARACTER COUNT
3271	013424	005437	000556		NEG	FMCNT	:SET TO TWO'S COMPLIMENT
3272	013430	013737	000556	000634	MOV	FMCNT,FCSAV	:SAVE FRAME COUNT

3273	013436	012704	025452	MOV	#MSG37,R4	;PRINT PATTERN NUMBER REQUEST
3274	013442	000004		TYPE		;TYPE MSG
3275	013444	013703	000560	MOV	PATRN,R3	
3276	013450	104400		TYPOCT		;PRINT PATTERN
3277	013452	005037	014440	CLR	DOFL	;CLEAR EXTERNAL DATA FLAG
3278	013456	012705	000560	MOV	#PATRN,R5	;SET PATTERN NUMBER ADDRESS
3279	013462	012701	000003	MOV	#3,R1	;SET NUMBER OF CHARACTERS TO INPUT
3280	013466	012702	000015	MOV	#15,R2	;SET MAXIMUM LIMIT
3281	013472	012703	000000	MOV	#0,R3	;SET MINIMUM LIMIT
3282	013476	004737	023352	JSR	PC,TTR	;GO GET PATTERN NUMBER
3283	013502	012704	026227	MOV	#MSG69,R4	
3284	013506	000004		TYPE		;TYPE MSG
3285	013510	013703	000564	MOV	TMEX,R3	
3286	013514	104400		TYPOCT		;PRINT CURRENT TM FLAG SETTING
3287	013516	012705	000564	MOV	#TMEX,R5	;GET TM FLAG ADDRESS
3288	013522	012701	000002	MOV	#2,R1	;SET SIZE OF RESPONSE
3289	013526	012702	000001	MOV	#1,R2	;SET UPPER LIMIT
3290	013532	012703	000000	MOV	#0,R3	;SET LOWER LIMIT
3291	013536	004737	023352	JSR	PC,TTR	;TM 1=YES
3292	013542	012704	024663	MOV	#MSG21,R4	
3293	013546	000004		TYPE		;TYPE MSG
3294	013550	013703	000570	MOV	INTRF,R3	
3295	013554	104400		TYPOCT		;PRINT CURRENT SETTING
3296	013556	012705	000570	MOV	#INTRF,R5	;GET FLAG ADDRESS
3297	013562	012701	000002	MOV	#2,R1	;SET SIZE OF RESPONSE
3298	013566	012702	000001	MOV	#1,R2	;SET UPPER LIMIT
3299	013572	012703	000000	MOV	#0,R3	;SET LOWER LIMIT
3300	013576	004737	023352	JSR	PC,TTR	;GO GET RESPONSE
3301	013602	012704	025475	MOV	#MSG38,R4	
3302	013606	000004		TYPE		;TYPE MSG
3303	013610	013703	000572	MOV	SPFLG,R3	
3304	013614	104400		TYPOCT		;PRINT CURRENT SETTING
3305	013616	012705	000572	MOV	#SPFLG,R5	;SET ADDRESS OF FLAG
3306	013622	012701	000002	MOV	#2,R1	;SET SIZE OF RESPONSE
3307	013626	012702	000001	MOV	#1,R2	;SET UPPER LIMIT
3308	013632	012703	000000	MOV	#0,R3	;SET LOWER LIMIT
3309	013636	004737	023352	JSR	PC,TTR	;GO GET RESPONSE
3310	013642	012704	025515	MOV	#MSG39,R4	
3311	013646	000004		TYPE		;TYPE MSG
3312	013650	013703	000566	MOV	CRCC,R3	
3313	013654	104400		TYPOCT		
3314	013656	012705	000566	MOV	#CRCC,R5	
3315	013662	012701	000002	MOV	#2,R1	
3316	013666	012702	000001	MOV	#1,R2	
3317	013672	012703	000000	MOV	#0,R3	
3318	013676	004737	023352	JSR	PC,TTR	
3319	013702	004737	023216	JSR	PC,GTSWR	;GET SWITCHES
3320	013706	005737	014070	TST	SCVFL	;BRANCH IF SHORT CONVERSATION
3321	013712	001063		BNE	TINPX	
3322	013714	005737	000636	TST	TINF	;BRANCH IF NO TTY INPUT
3323	013720	001460		BEQ	TINPX	
3324	013722	012704	025555	MOV	#MSG40,R4	
3325	013726	000004		TYPE		;TYPE MSG
3326	013730	013703	000574	MOV	RSTAL,R3	
3327	013734	104400		TYPOCT		;PRINT READ STALL
3328	013736	012705	000574	MOV	#RSTAL,R5	;SET READ STALL ADDRESS

TINP3A:

TINP4:

1\$:

```

3329 013742 012701 000007      MOV      #7,R1      ;SET NUMBER OF CHARACTERS TO INPUT
3330 013746 012702 177777      MOV      #-1,R2     ;SET MAXIMUM LIMIT
3331 013752 012703 000001      MOV      #1,R3      ;SET MINIMUM LIMIT
3332 013756 004737 023352      JSR      PC,TTR     ;GO GET READ STALL
3333 013762 012704 025604      MOV      #MSG41,R4
3334 013766 000004      TYPE      ;TYPE MSG
3335 013770 013703 000576      MOV      WSTAL,R3
3336 013774 104400      TYPOCT     ;PRINT READ STALL
3337 013776 012705 000576      MOV      #WSTAL,R5  ;SET WRITE STALL ADDRESS
3338 014002 012701 000007      MOV      #7,R1      ;SET NUMBER OF CHARACTERS TO INPUT
3339 014006 012702 177777      MOV      #-1,R2     ;SET MAXIMUM LIMIT
3340 014012 012703 000001      MOV      #1,R3      ;SET MINIMUM LIMIT
3341 014016 004737 023352      JSR      PC,TTR     ;GO GET WRITE STALL
3342 014022 012704 025616      MOV      #MSG42,R4
3343 014026 000004      TYPE      ;TYPE MSG
3344 014030 013703 000600      MOV      TSTAL,R3
3345 014034 104400      TYPOCT     ;PRINT TA STALL
3346 014036 012705 000600      MOV      #TSTAL,R5  ;SET TURN AROUND STALL ADDRESS
3347 014042 012701 000007      MOV      #7,R1      ;SET NUMBER OF CHARACTERS TO INPUT
3348 014046 012702 177777      MOV      #-1,R2     ;SET MAXIMUM LIMIT
3349 014052 012703 000001      MOV      #1,R3      ;SET MINIMUM LIMIT
3350 014056 004737 023352      JSR      PC,TTR     ;GO GET TURN AROUND STALL
3351 014062 005037 014070      TINPX:  CLR      SCVFL  ;CLEAR SHORT CONVERSATION FLAG
3352 014066 000207      RTS      PC      ;EXIT
3353 014070 000000      SCVFL:  0      ;SHORT CONVERSATION FLAG
3354
3355      ;UNIT DESCRIPTION POSITIONING SUBROUTINE*****
3356
3357 014072 006337 000646      TPOS:   ASL      TEMP2   ;POSITION CHARACTER
3358 014076 005303      DEC      R3      ;SEE IF DONE
3359 014100 001374      BNE      TPOS    ;IF NOT: BR
3360 014102 013700 000676      TPOS1:  MOV      UNP,R0   ;LOAD UNIT POINTER
3361 014106 053760 000646      BIS      TEMP2,UN1(R0) ;LOAD CHARACTER INTO UN1(R0)
3362 014114 000207      RTS      PC      ;EXIT
3363

```

```

3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382 014116 005737 015030      DSUP: TST      RDFL      ;SEE IF DID RANDOM DATA
3383 014122 001044              BNE      DS2A      ;IF NOT: BR
3384 014124 005737 000736      DS0:  TST      ASEQF     ;SEE IF AUTO SEQ
3385 014130 001406              BEQ      DSOC      ;IF NOT: BR
3386 014132 005737 000560      TST      PATRN     ;SEE IF AUTO RANDOM
3387 014136 100003              BPL      DSOC      ;IF NOT: BR
3388 014140 004737 014766      JSR      PC,DATR   ;ELSE GO GENERATE RANDOM DATA
3389 014144 000207              RTS      PC        ;RETURN
3390 014146 023737 000560 014276 DSOC: CMP      PATRN,PATS ;SEE IF NEW PATTERN
3391 014154 001014              BNE      DSOA      ;IF SO: BR
3392 014156 013703 000552      MOV      UDES,R3    ;GET UNIT DESCRIPTION
3393 014162 042703 177767      BIC      #177767,R3 ;MASK EVEN PARITY
3394 014166 023703 014300      CMP      PARS,R3    ;SEE IF SAME AS LAST TIME
3395 014172 001404              BEQ      DSOB      ;IF SO: BR
3396 014174 010337 014300      MOV      R3,PARS   ;SAVE PARITY
3397 014200 004737 015032      JSR      PC,CRCLRC ;GO GENERATE EXPT CRC/LRC
3398 014204 000207              RTS      PC
3399 014206 012703 027346      DSOB: MOV      #WDATA,R3 ;R3 = ADDRS OF WRITE BUFFER
3400 014212 013701 000560      DSOA: MOV      PATRN,R1  ;R1 = PATTERN SELECTOR
3401 014216 010137 014276      MOV      R1,PATS
3402 014222 062701 000001      MOV      R1,PATS
3403 014226 006301              ADD      #1,R1      ;BUMP POINTER
3404 014230 004771 002770      ASL      R1        ;MAKE PATTERN SELECTOR EVEN
3405 014234 004737 015032      JSR      PC,@DATBL(R1) ;GO GENERATE PATTERN
3406 014240 013702 000556      DS2A: JSR      PC,CRCLRC ;GO GENERATE EXPT CRC/LRC
3407 014244 006202              DS3:  MOV      FMCNT,R2   ;R2=BUFFER SIZE
3408 014246 012701 033354      ASR      R2        ;R2=FRAME CMT/2
3409 014252 005021              MOV      #RDATA,R1  ;R1=READ DATA START
3410 014254 005202              DS4:  CLR      (R1)+    ;CLEAR BUFFER
3411 014256 001375              INC      R2        ;SEE IF DONE ALL
3412 014260 013737 000552 014300 BNE      DS4      ;IF NOT: BR
3413 014266 042737 177767 014300 MOV      UDES,PARS ;GET UNIT DESCRIPTION
3414 014274 000207              BIC      #177767,PARS ;MASK PARITY
3415 014276 177777              RTS      PC        ;EXIT
3416 014300 000000              PATS: -1        ;PATTERN NUMBER SAVE
3417
3418

```

```

3419
3420                                     ;EXTERNAL DATA INPUT FROM H/S READER (256 CHARACTER MAXIMUM)
3421
3422 014302 005737 014440      DATO:  TST      DOFL      ;SEE IF SHOULD DO EXTERNAL INPUT
3423 014306 001352              BNE      DS2A      ;IF NOT: BR
3424 014310 012737 000001 014440      MOV      #1,DOFL  ;SET EXTERNAL FLAG
3425 014316 005077 164300              CLR      @PRS   ;CLEAR READER STATUS
3426 014322 005037 000644              CLR      TEMP1  ;CLEAR FOR USE AS CHARACTER FLAG
3427 014326 052777 000001 164266      DATOA:  BIS      #1,@PRS ;START READER
3428 014334 105777 164262      DATOB:  TSTB     @PRS   ;SEE IF DONE
3429 014340 100375              BPL      DATOB      ;IF NOT : BR
3430 014342 005001              CLR      R1         ;CLEAR SAVE LOCATION
3431 014344 117701 164254              MOVB     @PRB,R1 ;SAVE CHARACTER
3432 014350 005737 000644              TST      TEMP1  ;SEE IF HAVE FOUND START CHARACTER
3433 014354 001011              BNE      DATOC      ;IF SO : BR
3434 014356 105701              TSTB     R1         ;SEE IF CHARACTER IS 0
3435 014360 001762              BEQ      DATOA      ;IF SO : BR
3436 014362 012737 000001 000644      MOV      #1,TEMP1 ;ELSE SET CHARACTER FOUND FLAG
3437 014370 010137 000646      MOV      R1,TEMP2    ;SAVE DATA SIZE
3438 014374 010102              MOV      R1,R2      ;SAVE DATA SIZE
3439 014376 000753              BR        DATOA     ;GO GET FIRST DATA CHAR
3440 014400 110123              DATOC:  MOVB     R1,(R3)+ ;LOAD BUFFER
3441 014402 005302              DEC      R2         ;SEE IF READ ALL
3442 014404 001350              BNE      DATOA      ;IF NOT : BR
3443 014406 012701 027346      DATOD:  MOV      #WDATA,R1 ;R1 = START OF WRITE BUFFER
3444 014412 013702 000646      MOV      TEMP2,R2    ;R2 = SIZE OF DATA FIELD
3445 014416 112123      DATOE:  MOVB     (R1)+,(R3)+ ;REPEAT LOAD OF DATA FIELD
3446 014420 022703 033354      CMP      #RDATA,R3  ;SEE IF DONE
3447 014424 003002              BGT      DATOF      ;IF NOT: BR
3448 014426 000137 014234      JMP      DS2A      ;EXIT
3449 014432 005302      DATOF:  DEC      R2         ;SEE IF AT END OF DATA FIELD
3450 014434 001370              BNE      DATOE      ;IF NOT : BR
3451 014436 000763              BR        DATOD     ;ELSE RESTART FILL
3452 014440 000000      DOFL:    0              ;EXTERNAL DATA FLAG=1 IF ALREADY DONE
3453

```

```

3454                                     ;ALL ONES*****
3455
3456 014442 012701 177777  DAT1:  MOV    #-1,R1          ;R1=DATA
3457 014446 012702 002002  DAT1A: MOV    #2002,R2        ;R2=WORD COUNT +2
3458 014452 010123          1$:  MOV    R1,(R3)+        ;LOAD BUFFER
3459 014454 005302          DEC    R2                ;SEE IF DONE
3460 014456 001375          BNE     1$                ;IF NOT: BR
3461 014460 000207          RTS     PC
3462
3463                                     ;ALL ZEROS*****
3464
3465 014462 005001  DAT2:  CLR     R1                ;R1=DATA
3466 014464 000770          BR      DAT1A            ;LOAD BUFFER
3467
3468                                     ;WALKING ONE*****
3469
3470 014466 012701 000001  DAT3:  MOV    #1,R1          ;R1=DATA
3471 014472 000241          CLC
3472 014474 012702 004004  DAT3A: MOV    #4004,R2        ;R2=CHARACTER COUNT+4
3473 014500 110123          1$:  MOVB   R1,(R3)+        ;LOAD BUFFER
3474 014502 106101          ROLB   R1                ;SET NEXT CHARACTER
3475 014504 005302          DEC    R2                ;SEE IF DONE
3476 014506 001374          BNE     1$                ;IF NOT: BR
3477 014510 000207          RTS     PC
3478
3479                                     ;WALKING ZERO*****
3480
3481 014512 012701 000376  DAT4:  MOV    #376,R1         ;R1=START OF DATA
3482 014516 000261          SEC
3483 014520 000765          BR      DAT3A            ;LOAD BUFFER
3484
3485                                     ;ALTERNATING ONE/ZERO*****
3486
3487
3488 014522 012701 052525  DAT5:  MOV    #52525,R1       ;R1=DATA
3489 014526 000747          BR      DAT1A            ;LOAD BUFFER
3490
3491                                     ;ALTERNATING ZERO/ONE*****
3492
3493 014530 012701 125252  DAT6:  MOV    #125252,R1      ;R1=DATA
3494 014534 000744          BR      DAT1A            ;LOAD BUFFER
3495
3496                                     ;ONE/ZERO IN ALTERNATING WORDS*****
3497
3498 014536 012701 125252  DAT7:  MOV    #125252,R1      ;SET WORD 1
3499 014542 012702 052525  MOV    #52525,R2          ;SET WORD 2
3500 014546 012704 001002  MOV    #1002,R4          ;SET NUMBER OF ENTRIES
3501 014552 010123          1$:  MOV    R1,(R3)+        ;LOAD WORD 1
3502 014554 010223          MOV    R2,(R3)+        ;LOAD WORD 2
3503 014556 005304          DEC    R4                ;SEE IF DONE
3504 014560 001374          BNE     1$                ;IF NOT: BR
3505 014562 000207          RTS     PC
3506

```

```

3507                                     ;WALKING ONE/ALL ONE IN ALTERNATING CHARS****
3508
3509 014564 012702 002002    DAT10: MOV    #2002,R2      ;SET BUFFER SIZE
3510 014570 012701 000001    MOV    #1,R1          ;SET WALK BASE
3511 014574 000241
3512 014576 012713 177400    1$:  MOV    #177400,(R3)    ;LOAD ALL ONE BYTE
3513 014602 050123          BIS    R1,(R3)+        ;LOAD WALK BYTE
3514 014604 106101          ROLB   R1              ;WALK ONE
3515 014606 005302          DEC    R2
3516 014610 001372          BNE    1$              ;DO FULL BUFFER
3517 014612 000207          RTS     PC
3518
3519                                     ;ALL BITS 0-377*****
3520
3521 014614 005001            DAT11: CLR    R1          ;R1=STARTING DATA
3522 014616 012702 004004    MOV    #4004,R2        ;R2=CHARACTER COUNT+4
3523 014622 110123          1$:  MOVB   R1,(R3)+      ;LOAD BUFFER
3524 014624 105201          INCB   R1              ;BUMP DATA
3525 014626 005302          DEC    R2              ;SEE IF DONE
3526 014630 001374          BNE    1$              ;IF NOT: BR
3527 014632 000207          RTS     PC              ;RETURN
3528
3529                                     ;ALL BITS 377-0*****
3530
3531 014634 012701 000377    DAT12: MOV    #377,R1      ;R1=STARTING DATA
3532 014640 012702 004004    MOV    #4004,R2        ;R2=CHARACTER COUNT+4
3533 014644 110123          1$:  MOVB   R1,(R3)+      ;LOAD BUFFER
3534 014646 105301          DECB   R1              ;BUMP DATA
3535 014650 005302          DEC    R2              ;SEE IF DONE
3536 014652 001374          BNE    1$              ;IF NOT: BR
3537 014654 000207          RTS     PC              ;RETURN
3538
3539                                     ;ALTERNATING CHARACTERS 0 AND 377*****
3540
3541 014656 012701 000377    DAT13: MOV    #377,R1      ;R1 = DATA
3542 014662 000137 014446    JMP     DAT1A          ;LOAD BUFFER
3543
3544                                     ;WALKING ZERO/ALL ZERO IN ALTERNATING CHARS*****
3545
3546 014666 012702 002002    DAT14: MOV    #2002,R2      ;SET BUFFER SIZE
3547 014672 012701 000376    MOV    #376,R1          ;SET WALK BASE
3548 014676 000261
3549 014700 010113          1$:  MOV    R1,(R3)        ;LOAD WALK BYTE
3550 014702 042723 177400    BIC    #177400,(R3)+    ;CLEAR HIGH BYTE
3551 014706 106101          ROLB   R1              ;WALK ZERO BIT
3552 014710 005302          DEC    R2
3553 014712 001372          BNE    1$              ;FILL BUFFER
3554 014714 000207          RTS     PC              ;RETURN
3555

```

```

3556                                     ;AUTO SEQUENCE PATTERN*****
3557
3558 014716 012702 000200      DAT15: MOV    #200,R2      ;SET NUMBER OF ENTRIES
3559 014722 012701 014746      1$:  MOV    #APATS,R1      ;SET START OF PATTERN
3560 014726 012704 000010      MOV    #10,R4      ;SET SIZE OF PATTERN
3561 014732 012123      2$:  MOV    (R1)+,(R3)+      ;FILL BUFFER
3562 014734 005304      DEC    R4      ;SEE IF DONE PATTERN
3563 014736 001375      BNE    2$      ;IF NOT: BR
3564 014740 005302      DEC    R2      ;SEE IF DONE BUFER
3565 014742 001367      BNE    1$      ;IF NOT: BR
3566 014744 000207      RTS    PC      ;RETURN
3567
3568 014746 000000      APATS:  0
3569 014750 177400      177400
3570 014752 000377      377
3571 014754 000000      0
3572 014756 177777      -1
3573 014760 000377      377
3574 014762 177400      177400
3575 014764 177777      -1
3576
3577                                     ;RANDOM DATA GENERATOR SUBROUTINE*****
3578
3579 014766 013704 000556      DATR:  MOV    FMCNT,R4      ;SET NUMBER OF FRAMES
3580 014772 012703 027346      MOV    #WDATA,R3      ;SET ADDRESS OF START OF BUFFER
3581 014776 012701 177777      MOV    #-1,R1      ;SET HIGH LIMIT
3582 015002 005002      CLR    R2      ;SET LOW LIMIT
3583 015004 004737 023164      1$:  JSR    PC,RANG      ;GO GENERATE NUMBER
3584 015010 013723 000630      MOV    RANSV,(R3)+      ;LOAD BUFFER
3585 015014 005204      INC    R4      ;SEE IF DONE WHOLE BUFFER
3586 015016 001372      BNE    1$      ;IF NOT: BR
3587 015020 012737 000001 015030  MOV    #1,RDFL      ;SET RANDOM DATA FLAG
3588 015026 000207      RTS    PC      ;EXIT
3589 015030 000000      RDFL:  0      ;RANDOM DATA SELECT FLAG

```

```

3590
3591
3592
3593
3594
3595
3596
3597
3598
3599 015032 013700 000556      CRCLRC: MOV    FMCNT,R0      ;SET RECORD SIZE
3600 015036 005400              NEG      R0
3601 015040 012701 027346      MOV      #WDATA,R1    ;SET START OF BUFFER
3602 015044 005037 015366      CLR      XORS
3603 015050 111104              CL0:  MOVB   (R1),R4      ;GET CHARACTER
3604 015052 004737 015240      JSR      PC,CLP      ;GO GET PARITY OF CHARACTER
3605 015056 004737 015342      JSR      PC,XOR      ;XOR CHARACTER
3606 015062 000241              CLC
3607 015064 006004              ROR      R4      ;ROTATE 1 RIGHT
3608 015066 103014              BCC      CL2      ;IF NO CARRY: BR
3609 015070 052704 000400      BIS      #400,R4      ;SET BIT NINE
3610 015074 000241              CLC
3611 015076 010405              CL1:  MOV      R4,R5      ;SAVE CHARACTER
3612 015100 042705 177703      BIC      #177703,R5
3613 015104 005105              COM      R5
3614 015106 042705 177703      BIC      #177703,R5
3615 015112 042704 000074      BIC      #74,R4
3616 015116 050504              BIS      R5,R4      ;COMPLIMENT BITS 2,3,4,5
3617 015120 010437 015366      CL2:  MOV      R4,XORS
3618 015124 005300              DEC      R0
3619 015126 001350              BNE      CL0      ;BRANCH IF NOT LAST CHAR
3620 015130 013704 015366      CLLAST: MOV    XORS,R4
3621 015134 005137 015366      COM      XORS
3622 015140 042737 177050 015366 BIC      #177050,XORS
3623 015146 042704 177727      BIC      #177727,R4      ;COMPLIMENT ALL BUT BITS 3&5
3624 015152 050437 015366      BIS      R4,XORS
3625 015156 013737 015366 015370 MOV    XORS,EXCRC      ;SAVE EXPECTED CRC
3626 015164 013700 000556      MOV      FMCNT,R0
3627 015170 005400              NEG      R0
3628 015172 012701 027346      MOV      #WDATA,R1    ;DO EXPT LRC
3629 015176 005037 015366      CLR      XORS
3630 015202 111104              CL3:  MOVB   (R1),R4      ;GET PARITY
3631 015204 004737 015240      JSR      PC,CLP      ;XOR CHARACTER
3632 015210 004737 015342      JSR      PC,XOR
3633 015214 005300              DEC      R0
3634 015216 001371              BNE      CL3      ;DO ALL FOR LRC
3635 015220 013704 015370      MOV      EXCRC,R4
3636 015224 004737 015342      JSR      PC,XOR      ;XOR CRC TO DATA
3637 015230 013737 015366 015372 MOV    XORS,EXLRC      ;SAVE EXPT LRC
3638 015236 000207              RTS      PC      ;RETURN
3639 015240 005704              CLP:  TST      R4      ;SEE IF 0 CHAR
3640 015242 001010              BNE      CLPE      ;IF NOT: BR
3641 015244 032737 000010 000552 BIT      #10,UDES      ;SEE IF EVEN PARITY
3642 015252 001404              BEQ      CLPE      ;IF NOT: BR
3643 015254 012704 000420      MOV      #420,R4      ;SET 0 CHAR EVEN PARITY
3644 015260 005201              INC      R1      ;BUMP POINTER
3645 015262 000207              RTS      PC      ;RETURN

```

```

3646 015264 005046          CLPE:  CLR      -(SP)          ;CLEAR WORKING SPACE ON STACK
3647 015266 106304          1$:   ASLB     R4              ;SHIFT DATA
3648 015270 005516          ADC      (SP)              ;ADDUP # OF 1 BITS
3649 015272 105704          TSTB     R4                ;BRANCH IF ALL 0'S LEFT
3650 015274 001374          BNE      1$
3651 015276 112104          MOVB     (R1)+,R4
3652 015300 042704 177400    BIC      #177400,R4
3653 015304 106026          RORB     (SP)+
3654 015306 103405          BCS      CLP2
3655 015310 032737 000010 000552 BIT      #10,UDES      ;SEE IF SHOULD BE EVEN PARITY
3656 015316 001406          BEQ      CLP3              ;IF NOT: BR
3657 015320 000207          RTS      PC                ;ELSE EXIT
3658 015322 032737 000010 000552 CLP2: BIT      #10,UDES      ;SEE IF SHOULD BE ODD PARITY
3659 015330 001001          BNE      CLP3              ;IF NOT: BR
3660 015332 000207          RTS      PC                ;ELSE EXIT
3661 015334 052704 000400    CLP3: BIS      #400,R4      ;SET PARITY BIT
3662 015340 000207          RTS      PC
3663
3664 015342 010446          XOR:   MOV      R4,-(SP)
3665 015344 043716 015366    BIC      XORS,(SP)
3666 015350 040437 015366    BIC      R4,XORS
3667 015354 052637 015366    BIS      (SP)+,XORS
3668 015360 013704 015366    MOV      XORS,R4
3669 015364 000207          RTS      PC
3670
3671 015366 000000          XORS:   0
3672 015370 000000          EXCRC:  0
3673 015372 000000          EXLRC:  0
3674

```

;XOR SUBROUTINE: R4 WITH XORS

;XOR SAVE
;EXPECTED CRC
;EXPECTED LRC

```

3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690 015374 005037 000660          DCHK: CLR      BBC          ;CLEAR BAD RECORD CNTR
3691 015400 005037 000706          CLR      DERFL        ;CLEAR DATA ERROR FLAG
3692 015404 013705 000556          MOV      FMCNT,R5      ;LOAD CHAR COUNT
3693 015410 032737 000020 000552  BIT      #20,UDES      ;SEE IF CORE DUMP
3694 015416 001401          BEQ      DCHK0        ;IF NOT: BR
3695 015420 006205          ASR      R5          ;R5 = FC/2
3696 015422 012701 027346          DCHK0: MOV      #WDATA,R1      ;SET WRITE DATA ADDR
3697 015426 012702 033354          MOV      #RDATA,R2      ;SET READ DATA ADDR
3698 015432 032737 000010 000552  BIT      #10,UDES      ;SEE IF EVEN PARITY
3699 015440 001430          BEQ      DFOCO        ;IF NOT: BR
3700 015442 032737 000020 000552  BIT      #20,UDES      ;SEE IF CORE DUMP PARITY
3701 015450 001024          BNE      DFOCO        ;IF SO: BR
3702 015452 032737 002000 000552  BIT      #2000,UDES      ;SEE IF PE MODE
3703 015460 001020          BNE      DFOCO        ;IF SO: BR
3704 015462 105711          DFOF:  TSTB      (R1)        ;SEE IF 0 CHAR
3705 015464 001404          BEQ      DFOD        ;IF SO: BR
3706 015466 005201          INC      R1          ;BUMP POINTER
3707 015470 005205          DFOE:  INC      R5          ;SEE IF DONE
3708 015472 001373          BNE      DFOF        ;IF NOT: BR
3709 015474 000406          BR       DFOC        ;ELSE CONTINUE
3710 015476 112721 000020          DFOD:  MOVB     #20,(R1)+      ;SET 20 IN PLACE OF 0
3711 015502 012737 177777 014276  MOV      #-1,PATS      ;SET PATTERN GENERATE FLAG
3712 015510 000767          BR       DFOE
3713 015512 013705 000556          DFOC:  MOV      FMCNT,R5      ;RESET CHAR CNT
3714 015516 012701 027346          MOV      #WDATA,R1      ;RESET DATA ADDRESS
3715 015522 032737 010000 000562  DFOCO:  BIT      #10000,RDCMD      ;SEE IF READ REVERSE
3716 015530 001462          BEQ      DFO        ;IF NOT: BR
3717 015532 013704 000556          DFOB:  MOV      FMCNT,R4      ;GET FRAME COUNT
3718 015536 005404          NEG      R4          ;SET TO WHOLE NUMBER
3719 015540 032737 000020 000552  BIT      #20,UDES      ;SEE IF CORE DUMP
3720 015546 001402          BEQ      DFOB0        ;IF NOT: BR
3721 015550 000241          CLC
3722 015552 006004          ROR      R4          ;SET TO FC/2
3723 015554 060401          DFOB0: ADD      R4,R1      ;POINT TO START OF WRITE DATA
3724 015556 060402          ADD      R4,R2      ;POINT TO START OF READ DATA
3725 015560 032737 000001 000556  BIT      #1,FMCNT      ;SEE IF ODD FRAME COUNT
3726 015566 001401          BEQ      DFOA        ;IF NOT: BR
3727 015570 105722          TSTB      (R2)+      ;BUMP POINTER
3728 015572 032737 000020 000552  DFOA:  BIT      #20,UDES      ;SEE IF CORE DUMP
3729 015600 001431          BEQ      DFOA4        ;IF NOT: BR
3730 015602 000241          CLC

```

```

3731 015604 132742 000001      BITB    #1,-(R2)      ;SEE IF BIT 0 = 1
3732 015610 001401      BEQ      DFOA0      ;IF NOT: BR
3733 015612 000261      SEC
3734 015614 106012      DFOA0:  RORB    (R2)
3735 015616 000241      CLC
3736 015620 132712 000001      BITB    #1,(R2)
3737 015624 001401      BEQ      DFOA1
3738 015626 000261      SEC
3739 015630 106012      DFOA1:  RORB    (R2)      ;POSITION BITS FOR REVERSE CORE DUMP
3740 015632 000241      CLC
3741 015634 132712 000001      BITB    #1,(R2)
3742 015640 001401      BEQ      DFOA2
3743 015642 000261      SEC
3744 015644 106012      DFOA2:  RORB    (R2)
3745 015646 000241      CLC
3746 015650 132712 000001      BITB    #1,(R2)
3747 015654 001401      BEQ      DFOA3
3748 015656 000261      SEC
3749 015660 106012      DFOA3:  RORB    (R2)
3750 015662 005202      INC      R2      ;RESET POINTER
3751 015664 124142      DFOA4:  CMPB    -(R1),-(R2) ;TEST DATA CHARACTER
3752 015666 001010      BNE      DF1      ;IF NOT GOOD: BR
3753 015670 105037 000660      CLRB    BBC      ;CLEAR BAL RECORD COUNTER
3754 015674 000411      BR       DF2
3755 015676 122122      DFO:    CMPB    (R1)+,(R2)+ ;CHECK DATA
3756 015700 001003      BNE      DF1      ;IF BAD: BR
3757 015702 105037 000660      CLRB    BBC      ;CLEAR BAD RECORD CNTR
3758 015706 000404      BR       DF2
3759 015710 004737 016522      DF1:    JSR     PC,DRPKF ;GO GET DROPS AND PICKS
3760 015714 004737 016006      JSR     PC,DERR ;GO DO PRINT
3761 015720 005205      DF2:    INC     R5      ;BUMP CHAR CNTR
3762 015722 001405      BEQ      DF3      ;IF DONE ALL: BR
3763 015724 032737 010000 000562      BIT     #10000,RDCMD ;SEE IF READ REVERSE
3764 015732 001761      BEQ      DFO      ;IF NOT: BR
3765 015734 000716      BR       DFOA      ;ELSE CONTINUE READ REV
3766 015736 005037 000666      DF3:    CLR     HDRFL ;CLEAR HEADER FLAG
3767 015742 005737 000706      TST     DERFL ;SEE IF HAD DATA ERROR
3768 015746 001416      BEQ      DFX      ;IF NOT: BR
3769 015750 005737 000710      TST     SERFL
3770 015754 001013      BNE      DFX      ;IF NOT DATA ERROR ONLY: BR
3771 015756 013704 000676      MOV     UNP,R4
3772 015762 032737 010000 000562      BIT     #10000,RDCMD ;SEE IF READ REVERSE
3773 015770 001003      BNE      DF4      ;IF SO: BR
3774 015772 005264 001130      INC     DATER1(R4) ;BUMP DATA ERROR FORWARD COUNTER
3775 015776 000402      BR       DFX
3776 016000 005264 001170      DF4:    INC     DEREV1(R4) ;BUMP REVERSE DATA ERROR
3777 016004 000207      DFX:    RTS     PC      ;EXIT
3778

```

```

3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807 016006 032777 002000 162574 DERR: BIT #2000,ASWR ;BRANCH IF NO ERROR
3808 016014 001067 DERR4 ;PRINTOUT DESIRED
3809 016016 005237 000672 DERR0: INC PFLG ;SET PRINT FLAG
3810 016022 005737 000666 TST HDRFL ;SEE IF HAVE PRINTED HEADER
3811 016026 001007 BNE DERR0A ;IF SO: BR
3812 016030 004737 022602 JSR PC,PAPRT ;PRINT CYCLE NUMBER
3813 016034 012704 024456 MOV #MSG1,R4 ;LOAD ERROR MSG ADDR
3814 016040 000004 TYPE ;TYPE MSG
3815 016042 004737 021036 JSR PC,FRPRT ;PRINT F OR R
3816 016046 012704 024475 DERR0A: MOV #MSG4,R4
3817 016052 000004 TYPE ;TYPE MSG
3818 016054 010203 MOV R2,R3
3819 016056 162703 033354 SUB #RDATA,R3 ;POINT TO CHAR
3820 016062 005303 DEC R3
3821 016064 032737 010000 000562 BIT #10000,RDCMD ;SEE IF READ REVERSE
3822 016072 001402 BEQ DERROB ;IF NOT: BR
3823 016074 010503 MOV R5,R3 ;GET CHAR NUMBER
3824 016076 005103 COM R3
3825 016100 104400 DERROB: TYPOCT ;PRINT CHAR NUMBER
3826 016102 012704 024463 MOV #MSG2,R4
3827 016106 000004 TYPE ;TYPE MSG
3828 016110 032737 010000 000562 BIT #10000,RDCMD ;SEE IF READ REVERSE
3829 016116 001402 BEQ DERROC ;IF NOT: BR
3830 016120 111103 MOVB (R1),R3 ;GET CHAR
3831 016122 000401 BR DERROD
3832 016124 114103 DERROC: MOVB -(R1),R3 ;LOAD EXPECTED DATA
3833 016126 004737 024234 DERROD: JSR PC,DOUT ;GO PRINT CHAR
3834 016132 012704 024470 MOV #MSG3,R4

```

```

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

```

3835	016136	000004			TYPE		:TYPE MSG
3836	016140	032737	010000	000562	BIT	#10000,RDCMD	:SEE IF READ REVERSE
3837	016146	001402			BEQ	DERR1	:IF NOT: BR
3838	016150	111203			MOVB	(R2),R3	:GET CHAR
3839	016152	000401			BR	DERR2	
3840	016154	114203			DERR1: MOVB	-(R2),R3	
3841	016156	004737	024234		DERR2: JSR	PC,DOUT	:PRINT BAD CHAR
3842	016162	032737	010000	000562	BIT	#10000,RDCMD	:BRANCH IF NOT READ
3843	016170	001001			BNE	DERR4	:REVERSE
3844	016172	122122			DERR3: CMPB	(R1)+,(R2)+	:RESET POINTERS
3845	016174	105237	000660		DERR4: INCB	BBC	:BUMP BAD RECORD CNTR
3846	016200	122737	000010	000660	CMPB	#10,BBC	:SEE IF BLD BTH
3847	016206	001123			BNE	DEREX	:IF NOT: BR
3848	016210	032777	002000	162372	BIT	#2000,ASWR	:SEE IF PRINT INHIBIT
3849	016216	001003			BNE	1\$	:IF SO: BR
3850	016220	012704	024611		MOV	#MSG15,R4	
3851	016224	000004			TYPE		:TYPE MSG
3852	016226	105037	000660		1\$: CLRB	BBC	:RESET BAD RECORD CNTR
3853	016232	000337	000660		SWAB	BBC	:POSITION BLD BTH AMOUNT
3854	016236	105237	000660		INCB	BBC	:BUMP AMOUNT
3855	016242	122737	000003	000660	CMPB	#3,BBC	:SEE IF HAD 3 BLD BTHS
3856	016250	101054			BHI	DERR4B	:IF NOT: BR
3857	016252	000337	000660		SWAB	BBC	:REPOSITION BBC
3858	016256	022705	177767		CMP	#177767,R5	:SEE IF ON LAST EIGHT CHARS
3859	016262	101473			BLOS	DERR6	:IF SO: BR
3860	016264	012705	177767		MOV	#177767,R5	:SET CHAR CNTR TO 8
3861	016270	032737	010000	000562	BIT	#10000,RDCMD	:SEE IF READ REVERSE
3862	016276	001416			BEQ	DERR4A	:IF NOT: BR
3863	016300	012701	027346		MOV	#WDATA,R1	:GET START OF BUFFER
3864	016304	012702	033354		MOV	#RDATA,R2	:GET START OF BUFFER
3865	016310	062701	000010		ADD	#10,R1	
3866	016314	062702	000010		ADD	#10,R2	:POINT TO START +10
3867	016320	032737	000001	000556	BIT	#1,FMCNT	:SEE IF ODD FRAME COUNT
3868	016326	001453			BEQ	DEREX	:IF NOT: BR
3869	016330	105722			TSTB	(R2)+	:BUMP POINTER
3870	016332	000451			BR	DEREX	
3871	016334	013737	000556	000644	DERR4A: MOV	FMCNT,TEMP1	:LOAD CHAR COUNT
3872	016342	005137	000644		COM	TEMP1	
3873	016346	005237	000644		INC	TEMP1	
3874	016352	162737	000010	000644	SUB	#10,TEMP1	:POINT TO BUFFER -8
3875	016360	013701	000644		MOV	TEMP1,R1	:POINT TO NEXT CHAR
3876	016364	062701	027346		ADD	#WDATA,R1	:POINT TO NEXT WRITE CHAR
3877	016370	013702	000644		MOV	TEMP1,R2	:POINT TO END OF READ DATA -8 FORWARD
3878	016374	062702	033354		ADD	#RDATA,R2	:POINT TO NEXT CHAR
3879	016400	000426			BR	DEREX	:EXIT
3880	016402	000337	000660		DERR4B: SWAB	BBC	:REPOSITION BBC
3881	016406	000241			CLC		
3882	016410	062705	000024		ADD	#24,R5	:SKIP 20 CHARS
3883	016414	103416			BCS	DERR6	:IF EXCEED RECORD SIZE: BR
3884	016416	032737	010000	000562	BIT	#10000,RDCMD	:SEE IF READ REVERSE
3885	016424	001405			BEQ	DERR5	:IF NOT: BR
3886	016426	162701	000024		SUB	#24,R1	
3887	016432	162702	000024		SUB	#24,R2	:RESET POINTERS
3888	016436	000407			BR	DEREX	
3889	016440	062701	000024		DERR5: ADD	#24,R1	:SKIP 20 CHARS
3890	016444	062702	000024		ADD	#24,R2	:SKIP FORWARD 20 CHARS

3891	016450	000402		BR	DEREX	
3892	016452	012705	177777	DERR6: MOV	#-1,R5	;SET TO EOR
3893	016456	005777	162126	DEREX: TST	@SWR	;BRANCH IF NOT HALT ON ERROR
3894	016462	100012		BPL	DEREX1	
3895	016464	000000		HALT		
3896	016466	005737	000672	TST	PFLG	;SEE IF PRINTED
3897	016472	001006		BNE	DEREX1	;IF SO: BR
3898	016474	032777	002000 162106	BIT	#2000,@SWR	;SEE IF SHOULD PRINT
3899	016502	001002		BNE	DEREX1	;IF NOT: BR
3900	016504	000137	016016	JMP	DERRO	;ELSE PRINT
3901	016510	005037	000672	DEREX1: CLR	PFLG	;CLEAR FLAG
3902	016514	005237	000706	INC	DERFL	;BUMP DATA ERROR FLAG
3903	016520	000207		RTS	PC	;RETURN
3904						

```

3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923 016522 005037 000644      DRPKF: CLR      TEMP1
3924 016526 005037 000646      CLR      TEMP2
3925 016532 005037 000650      CLR      TEMP3
3926 016536 111137 000644      MOV      (R1),TEMP1      ;LOAD GOOD CHAR
3927 016542 111237 000646      MOV      (R2),TEMP2      ;LOAD BAD CHAR
3928 016546 013704 000676      MOV      UNP,R4
3929 016552 016437 000770 000722 MOV      PIK1(R4),BPKP
3930 016560 016437 001010 000720 MOV      DRP1(R4),BDPP
3931 016566 032737 010000 000562 BIT      #10000,RDCMD      ;SEE IF READ REVERSE
3932 016574 001005      BNE      DRPK      ;IF SO: BR
3933 016576 124142      CMPB     -(R1),-(R2)      ;POINT TO CHAR
3934 016600 112137 000644      MOV      (R1)+,TEMP1      ;LOAD GOOD CHAR
3935 016604 112237 000646      MOV      (R2)+,TEMP2      ;LOAD BAD CHAR
3936 016610 004737 016622      DRPK: JSR      PC,DROP      ;GET DROPS
3937 016614 004737 017042      JSR      PC,PICK      ;GET PICKS
3938 016620 000207      RTS      PC      ;EXIT
3939 016622 113703 000644      DROP: MOV      TEMP1,R3      ;R3 = GOOD CHAR
3940 016626 113704 000646      MOV      TEMP2,R4      ;R4 = BAD CHAR
3941 016632 140403      DPC: BICB      R4,R3      ;GET DROPS/PICKS
3942 016634 001001      BNE      DPCG      ;IF SOME: BR
3943 016636 000207      RTS      PC      ;RETURN
3944 016640 012737 000010 000712 DPCG: MOV      #10,BCNT      ;SET NUMBER TO CHECK
3945 016646 132703 000001 DPC0: BITB      #1,R3      ;SEE IF DROPPED OR PICKED THIS BIT
3946 016652 001455      BEQ      DPC2      ;IF NOT: BR
3947 016654 105737 000650      TSTB     TEMP3      ;SEE IF ON PICKS
3948 016660 001016      BNE      DPC1      ;IF SO: BR
3949 016662 005277 162032      INC      @BDPP      ;BUMP DROP CNTR
3950 016666 005777 162026      TST      @BDPP
3951 016672 100045      BPL      DPC2      ;IF NO OVERFLOW: BR
3952 016674 032777 002000 161706 BIT      #2000,@SWR      ;SEE IF HAVE PRINTED DATA
3953 016702 001402      BEQ      DPC0A      ;IF SO: BR
3954 016704 004737 022602      JSR      PC,PAPRT      ;PRINT CYCLE NUMBER
3955 016710 004737 017106      DPC0A: JSR      PC,DPPRT      ;PRINT DROPS AND PICKS
3956 016714 000415      BR      DPC2A
3957 016716 005277 162000      DPC1: INC      @BPKP      ;BUMP PICK CNTR
3958 016722 005777 161774      TST      @BPKP      ;SEE IF OVERFLOW
3959 016726 100027      BPL      DPC2      ;IF NOT: BR
3960 016730 032777 002000 161652 BIT      #2000,@SWR      ;SEE IF HAVE PRINTED DATA

```

3961	016736	001402			BEQ	DPC1A	; IF SO: BR
3962	016740	004737	022602		JSR	PC,PAPRT	; PRINT CYCLE NUMBER
3963	016744	004737	017106		DPC1A: JSR	PC,DPPRT	; PRINT DROPS AND PICKS
3964	016750	013704	000676		DPC2A: MOV	UNP,R4	
3965	016754	016403	001010		MOV	DRP1(R4),R3	; SET DROP POINTER
3966	016760	016404	000770		MOV	PIK1(R4),R4	; SET PICK POINTER
3967	016764	012737	000010	000712	MOV	#10,BCNT	; SET NUMBER OF BITS
3968	016772	005023			DPC2B: CLR	(R3)+	; CLEAR DROPS
3969	016774	005024			CLR	(R4)+	; CLEAR PICK
3970	016776	005337	000712		DEC	BCNT	; SEE IF DONE
3971	017002	001373			BNE	DPC2B	; IF NOT: BR
3972	017004	000207			RTS	PC	; EXIT
3973	017006	000241			DPC2: CLC		
3974	017010	106003			RORB	R3	; GET NEXT BIT
3975	017012	005337	000712		DEC	BCNT	; SEE IF DONE
3976	017016	001410			BEQ	DPC3	
3977	017020	062737	000002	000722	ADD	#2,BPKP	
3978	017026	062737	000002	000720	ADD	#2,BDPP	
3979	017034	000137	016646		JMP	DPC0	; CONTINUE
3980	017040	000207			RTS	PC	; RETURN
3981	017042	013704	000676		PICK: MOV	UNP,R4	; GET UNIT POINTER
3982	017046	016437	000770	000722	MOV	PIK1(R4),BPKP	; SET PICK POINTER
3983	017054	016437	001010	000720	MOV	DRP1(R4),BDPP	; SET DROP POINTER
3984	017062	113704	000644		MOVB	TEMP1,R4	; R4 = GOOD CHAR
3985	017066	113703	000646		MOVB	TEMP2,R3	; R3 = BAD CHAR
3986	017072	112737	000001	000650	MOVB	#1,TEMP3	; SET PICK FLAG
3987	017100	004737	016632		JSR	PC,DPC	; GO CHECK PICKS
3988	017104	000207			RTS	PC	; EXIT
3989	017106	012704	025107		DPPRT: MOV	#MSG26,R4	
3990	017112	000004			TYPE		; TYPE MSG
3991	017114	013704	000676		MOV	UNP,R4	
3992	017120	016437	001010	000720	MOV	DRP1(R4),BDPP	; SET DROP POINTER
3993	017126	016437	000770	000722	MOV	PIK1(R4),BPKP	; SET PICK POINTER
3994	017134	062737	000016	000720	ADD	#16,BDPP	
3995	017142	062737	000016	000722	ADD	#16,BPKP	
3996	017150	012737	000010	000712	MOV	#10,BCNT	; SET NUMBER TO PRINT
3997	017156	017703	161536		DPPRT0: MOV	@BDPP,R3	
3998	017162	104400			TYPOCT		; PRINT DROPS
3999	017164	005337	000712		DEC	BCNT	; SEE IF DONE
4000	017170	001404			BEQ	DPPRT1	; IF NOT: BR
4001	017172	162737	000002	000720	SUB	#2,BDPP	; BUMP POINTER
4002	017200	000766			BR	DPPRT0	; CONTINUE FOR ALL 8 BITS
4003	017202	012737	000010	000712	DPPRT1: MOV	#10,BCNT	; SET NUMBER TO PRINT
4004	017210	012704	025120		MOV	#MSG27,R4	
4005	017214	000004			TYPE		; TYPE MSG
4006	017216	017703	161500		DPPRT2: MOV	@BPKP,R3	
4007	017222	104400			TYPOCT		; PRINT PICKS
4008	017224	005337	000712		DEC	BCNT	; SEE IF DONE
4009	017230	001404			BEQ	DPPRTX	; IF SO: BR
4010	017232	162737	000002	000722	SUB	#2,BPKP	; BUMP POINTER
4011	017240	000766			BR	DPPRT2	; CONTINUE FOR ALL 8 BITS
4012	017242	000207			DPPRTX: RTS	PC	; RETURN

```

4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038 017244 013703 000556      ERCHK: MOV    FMCNT,R3      ;GET FRAME COUNT
4039 017250 032703 000001      BIT     #1,R3      ;SEE IF ODD
4040 017254 001401              BEQ     ERO          ;IF NOT: BR
4041 017256 005303              DEC     R3          ;BUMP COUNT
4042 017260 005403      ERO:   NEG     R3
4043 017262 032737 000020 000552 BIT     #20,UDES      ;SEE IF CORE DUMP
4044 017270 001401              BEQ     EROB         ;IF NOT: BR
4045 017272 006203              ASR     R3          ;SET TO FC/2
4046 017274 032737 000010 000674 EROB:  BIT     #10,MTC1      ;SEE IF WRITE OP
4047 017302 001414              BEQ     ER1          ;IF SO: BR
4048 017304 032737 010000 000562      BIT     #10000,RDCMD
4049 017312 001405              BEQ     EROA
4050 017314 012703 033354      MOV     #RDATA,R3
4051 017320 162703 000002      SUB     #2,R3          ;SET POINTER
4052 017324 000405              BR      ER2
4053 017326 062703 033354      EROA:  ADD     #RDATA,R3      ;BUILD EXPT READ ADDRESS
4054 017332 000402              BR      ER2
4055 017334 062703 027346      ER1:   ADD     #WDATA,R3      ;BUILD EXPT WRITE ADDRESS
4056 017340 010337 021012      ER2:   MOV     R3,CADER      ;SAVE ADDRESS
4057 017344 012704 000007      MOV     #7,R4
4058 017350 012701 021014      MOV     #BAER,R1
4059 017354 005021      ER2A0:  CLR     (R1)+      ;CLEAR FLAGS
4060 017356 005304              DEC     R4
4061 017360 001375              BNE     ER2A0
4062 017362 020377 161126      CMP     R3,#BA      ;SEE IF ADDRESS OK
4063 017366 001402              BEQ     ER2A1      ;IF SO: BR
4064 017370 005237 021014      INC     BAER      ;SET BUS ADDRESS ERROR
4065 017374 032737 000010 000674 ER2A1: BIT     #10,MTC1      ;SEE IF WRITE OPER
4066 017402 001006              BNE     ER2B         ;IF NOT: BR
4067 017404 005777 161106      ER2A:  TST     @FC      ;SEE IF FC=0
4068 017410 001441              BEQ     ER3          ;IF SO: BR

```

```

4069 017412 005237 021022      INC      FCER      ;SET FC ERROR
4070 017416 000436      BR      ER3
4071 017420 032737 000040 000674 ER2B: BIT      #40,MTCL ;SEE IF SPACE OPER
4072 017426 001766      BEQ      ER2A      ;IF SO: BR
4073 017430 005737 000700      TST      TMFLG     ;SEE IF TM TIME
4074 017434 001011      BNE      ER2D      ;IF SO: BR
4075 017436 013703 000556      MOV      FMCNT,R3
4076 017442 005403      NEG      R3      ;R3 = EXPT RECORD SIZE
4077 017444 020377 161046      ER2C: CMP      R3,@FC     ;SEE IF FC = EXPT
4078 017450 001421      BEQ      ER3      ;IF SO: BR
4079 017452 005237 021022      INC      FCER      ;SET FC ERROR FLAG
4080 017456 000416      BR      ER3
4081 017460 032737 002000 000552 ER2D: BIT      #2000,UDS   ;SEE IF PE
4082 017466 001346      BNE      ER2A      ;IF SO: BR
4083 017470 032737 010000 000562      BIT      #10000,RDCMD ;SEE IF READ REVERSE
4084 017476 001003      BNE      ER2E      ;IF SO: BR
4085 017500 012703 000002      MOV      #2,R3
4086 017504 000757      BR      ER2C      ;LOOK FOR EXPT = 2
4087 017506 012703 000001      ER2E: MOV      #1,R3
4088 017512 000754      BR      ER2C      ;GO CHECK FC FOR TM
4089 017514 032777 160000 160766 ER3:  BIT      #160000,@C1  ;SEE IF COUNT ERROR
4090 017522 001437      BEQ      ER4
4091 017524 017703 160770      MOV      @CS,R3      ;GET CONT STATUS REG
4092 017530 042703 000307      BIC      #307,R3      ;MASK OUT IR,OR,UNIT NO. & SEE IF OTHER ERRORS
4093 017534 001406      BEQ      ER3A      ;IF NOT: BR
4094 017536 005737 000700      TST      TMFLG     ;SEE IF TAPE MARK TIME
4095 017542 001425      BEQ      ER3B      ;IF NOT: BR
4096 017544 042703 001000      BIC      #1000,R3      ;MASK MISSED TRANS & BR IF OTHER ERRORS
4097 017550 001022      BNE      ER3B
4098 017552 032777 060000 160730 ER3A: BIT      #60000,@C1  ;SEE IF EITHER TRE OR MCPE
4099 017560 001420      BEQ      ER4      ;IF NOT: BR
4100 017562 005737 000700      TST      TMFLG     ;SEE IF TM TIME
4101 017566 001413      BEQ      ER3B      ;IF NOT: BR
4102 017570 017703 160730      MOV      @ER,R3      ;GET ERROR REGISTER
4103 017574 032737 000010 000552      BIT      #10,UDS     ;SEE IF EVEN PARITY
4104 017602 001402      BEQ      ER3A1      ;IF NOT: BR
4105 017604 042703 000100      BIC      #100,R3      ;MASK PAR
4106 017610 042703 001000      ER3A1: BIC      #1000,R3     ;MASK FCE
4107 017614 001402      BEQ      ER4      ;IF NO ERRORS EXCEPT FCE: BR
4108 017616 005237 021016      ER3B: INC      CONER     ;SET CONT ERROR FLAG
4109 017622 032777 040000 160672 ER4:  BIT      #40000,@DS   ;SEE IF DRIVE ERROR
4110 017630 001420      BEQ      ER6      ;IF NOT: BR
4111 017632 005737 000700      TST      TMFLG     ;SEE IF TAPE MARK TIME
4112 017636 001413      BEQ      ER4A      ;IF NOT: BR
4113 017640 017703 160660      MOV      @ER,R3      ;GET ER
4114 017644 032737 000010 000552      BIT      #10,UDS     ;SEE IF EVEN PARITY
4115 017652 001402      BEQ      ER4A1      ;IF NOT: BR
4116 017654 042703 000100      BIC      #100,R3      ;MASK PAR
4117 017660 042703 001000      ER4A1: BIC      #1000,R3     ;MASK OUT FCE & BRANCH IF
4118 017664 001402      BEQ      ER6      ;NO OTHER ERRORS
4119 017666 005237 021020      ER4A: INC      DRIVER    ;SET DRIVER ERROR FLAG
4120 017672 032737 002000 000552 ER6:  BIT      #2000,UDS   ;IF IN PE MODE: BR
4121 017700 001071      BNE      ERPT      ;SEE IF NO DATA CHECK
4122 017702 032777 020000 160700      BIT      #20000,@SWR  ;IF NOT: BR (ALLOW READ OF UNKNOWN TAPES)
4123 017710 001065      BNE      ERPT
4124 017712 032737 000040 000674      BIT      #40,MTCL  ;SEE IF WRITE OR READ OP

```

4125	017720	001461			BEQ	ERPT	;IF NOT: BR
4126	017722	005737	000700		TST	TMFLG	;SEE IF TAPE MARK TIME
4127	017726	001413			BEQ	ER6A	;IF NOT: BR
4128	017730	013737	015370	021034	MOV	EXCRC,CRCSV	;SAVE CRC
4129	017736	013737	015372	021032	MOV	EXLRC,LRCV	;SAVE LRC
4130	017744	005037	015370		CLR	EXCRC	
4131	017750	012737	000023	015372	MOV	#23,EXLRC	;SET CRC/LRC FOR TM
4132	017756	032737	000060	000552	ER6A: BIT	#60,UDS	;SEE IF FORMAT 14
4133	017764	001037			BNE	ERPT	;IF NOT: BR
4134	017766	017703	160536		MOV	@CC,R3	;GET CRC CHARACTER
4135	017772	042703	177000		BIC	#177000,R3	
4136	017776	023703	015370		CMP	EXCRC,R3	
4137	020002	001402			BEQ	ER7	;IF CRC GOOD: BR
4138	020004	005237	021026		INC	CRCL	;SET ERROR FLAG
4139	020010	017703	160520		ER7: MOV	@MR,R3	;GET LRC
4140	020014	000303			SWAB	R3	
4141	020016	005703			TST	R3	
4142	020020	100002			BPL	ER10	
4143	020022	052703	000400		BIS	#400,R3	
4144	020026	042703	177000		ER10: BIC	#177000,R3	
4145	020032	023703	015372		CMP	EXLRC,R3	
4146	020036	001412			BEQ	ERPT	;IF LRC GOOD: BR
4147	020040	010337	021030		MOV	R3,ACTLRC	;SAVE ACTUAL LRC
4148	020044	005237	021024		INC	LRCR	;SET LRC ERROR FLAG
4149	020050	032737	010000	000562	BIT	#10000,RDCMD	;SEE IF READ REVERSE
4150	020056	001402			BEQ	ERPT	;IF NOT: BR
4151	020060	005037	021024		CLR	LRCR	;ELSE CLEAR LRC ERROR
4152	020064	012703	000006		ERPT: MOV	#6,R3	
4153	020070	005037	000710		CLR	SERFL	;CLEAR ERROR FLAG
4154	020074	005037	000724		CLR	ERSAV	
4155	020100	012704	021014		MOV	#BAER,R4	
4156	020104	005724			ERPTT: TST	(R4)+	;SEE IF ANY ERROR
4157	020106	001004			BNE	ERPTG	;IF SO: BR
4158	020110	005303			DEC	R3	
4159	020112	001374			BNE	ERPTT	
4160	020114	000137	020746		JMP	ERPX1	
4161	020120	005237	000710		ERPTG: INC	SERFL	;SET ERROR FLAG
4162	020124	017737	160374	000724	MOV	@ER,ERSAV	;SAVE ERROR REGISTER
4163	020132	032777	002000	160450	BIT	#2000,@SWR	;SEE IF PRINT
4164	020140	001420			BEQ	ERPT0	;IF SO: BR
4165	020142	022737	000002	000714	CMP	#2,RTYFL	;SEE IF READ RETRY
4166	020150	001006			BNE	ERPTG1	;IF NOT: BR
4167	020152	013703	000704		MOV	RTCNT,R3	
4168	020156	005203			INC	R3	;BUMP RETRY COUNT
4169	020160	020337	000604		CMP	R3,RETRY	;SEE IF LAST RETRY
4170	020164	001406			BEQ	ERPT0	;IF SO: BR
4171	020166	022737	000002	021020	ERPTG1: CMP	#2,DRVER	;SEE IF TM STATUS ERROR
4172	020174	001407			BEQ	ERPT0	;IF SO: BR
4173	020176	000137	020626		JMP	ERPX0	
4174	020202	005237	000672		ERPT0: INC	PFLG	
4175	020206	004737	022602		JSR	PC,PAPRT	;PRINT HEADER
4176	020212	013704	000654		MOV	EMADDR,R4	
4177	020216	000004			TYPE		;TYPE MSG
4178	020220	004737	021036		JSR	PC,FRPRT	;PRINT F OR R
4179	020224	005737	000700		TST	TMFLG	
4180	020230	001407			BEQ	ERPT1	

4181	020232	022737	026045	000654		CMP	#MSG54,EMADDR	
4182	020240	001403				BEQ	ERPT1	
4183	020242	012704	026063			MOV	#MSG56,R4	;PRINT TM
4184	020246	000004				TYPE		;TYPE MSG
4185	020250	005737	021016		ERPT1:	TST	CONER	
4186	020254	001414				BEQ	ERPT2	;IF NO CONT ERROR: BR
4187	020256	012704	024737			MOV	#MSG23,R4	
4188	020262	000004				TYPE		;TYPE MSG
4189	020264	017703	160220			MOV	@C1,R3	
4190	020270	104400				TYPOCT		;PRINT CONTROL '
4191	020272	012704	024764			MOV	#MSG23D,R4	;PRINT CS TAG
4192	020276	000004				TYPE		;TYPE MSG
4193	020300	017703	160214			MOV	@CS,R3	
4194	020304	104400				TYPOCT		;PRINT CONT STATUS
4195	020306	005737	021020		ERPT2:	TST	DRVER	
4196	020312	001414				BEQ	ERPT3	;IF SO DRIVE ERROR: BR
4197	020314	012704	024772			MOV	#MSG23E,R4	
4198	020320	000004				TYPE		;TYPE MSG
4199	020322	017703	160174			MOV	@DS,R3	
4200	020326	104400				TYPOCT		;PRINT DRIVE STATUS
4201	020330	012704	024777			MOV	#MSG23F,R4	
4202	020334	000004				TYPE		;TYPE MSG
4203	020336	017703	160162			MOV	@ER,R3	
4204	020342	104400				TYPOCT		;PRINT DRIVE ERROR
4205	020344	005737	021014		ERPT3:	TST	BAER	
4206	020350	001416				BEQ	ERPT4	;IF NO BA ERROR: BR
4207	020352	012704	024752			MOV	#MSG23B,R4	
4208	020356	000004				TYPE		;TYPE MSG
4209	020360	017703	160130			MOV	@BA,R3	
4210	020364	104400				TYPOCT		;PRINT BUS ADDRESS
4211	020366	012737	000255	000640		MOV	#255,TOB	
4212	020374	004737	023756			JSR	PC,TOG	;PRINT /
4213	020400	013703	021012			MOV	CADER,R3	
4214	020404	104400				TYPOCT		;PRINT EXPT BUS ADDRESS
4215	020406	005737	021022		ERPT4:	TST	FCER	
4216	020412	001406				BEQ	ERPT5	;IF NO FC ERROR: BR
4217	020414	012704	024757			MOV	#MSG23C,R4	
4218	020420	000004				TYPE		;TYPE MSG
4219	020422	017703	160070			MOV	@FC,R3	
4220	020426	104400				TYPOCT		;PRINT FRAME COUNT
4221	020430	012704	024745		ERPT5:	MOV	#MSG23A,R4	
4222	020434	000004				TYPE		;TYPE MSG
4223	020436	017703	160050			MOV	@WC,R3	
4224	020442	104400				TYPOCT		;PRINT WORD COUNT
4225	020444	005737	021026			TST	CRCER	
4226	020450	001420				BEQ	ERPT5A	;IF NO CRC ERROR: BR
4227	020452	012704	026110			MOV	#MSG58,R4	
4228	020456	000004				TYPE		;TYPE MSG
4229	020460	017703	160044			MOV	@CC,R3	
4230	020464	042703	177000			BIC	#177000,R3	
4231	020470	104400				TYPOCT		;PRINT ACTUAL CRC
4232	020472	012737	000255	000640		MOV	#255,TOB	
4233	020500	004737	023756			JSR	PC,TOG	
4234	020504	013703	015370			MOV	EXCRC,R3	
4235	020510	104400				TYPOCT		;PRINT EXPECTED CRC
4236	020512	005737	021024		ERPT5A:	TST	LRCER	

4237	020516	001416			BEQ	ERPT6	; IF NO LRC ERROR: BR
4238	020520	012704	026116		MOV	#MSG59,R4	
4239	020524	000004			TYPE		; TYPE MSG
4240	020526	013703	021030		MOV	ACTLRC,R3	
4241	020532	104400			TYPOCT		; PRINT ACTUAL LRC
4242	020534	012737	000255	000640	MOV	#255,TOB	
4243	020542	004737	023756		JSR	PC,TOG	
4244	020546	013703	015372		MOV	EXLRC,R3	
4245	020552	104400			TYPOCT		; PRINT EXPECTED LRC
4246	020554	005737	021020		ERPT6: TST	DRVER	
4247	020560	001421			BEQ	ERPT7	; IF NO DRIVE ERROR: BR
4248	020562	032737	002000	000552	BIT	#2000, UDES	
4249	020570	001415			BEQ	ERPT7	; IF NO PE: BR
4250	020572	017704	157726		MOV	@ER,R4	
4251	020576	042704	075477		BIC	#75477,R4	; MASK OUT ALL BUT BITS 15,10,7,6
4252	020602	001410			BEQ	ERPT7	; IF NO CONDITIONALS SET: BR
4253	020604	012704	025011		MOV	#MSG23H,R4	
4254	020610	000004			TYPE		; TYPE MSG
4255	020612	017703	157712		MOV	@CC,R3	
4256	020616	042703	177000		BIC	#177000,R3	; MASK CC
4257	020622	104400			TYPOCT		; PRINT CHECK CHARACTERS
4258	020624	000240			ERPT7: NOP		
4259	020626	005777	157756		ERPX0: TST	@SWR	; BRANCH IF NOT HALT ON ERROR
4260	020632	100012			BPL	ERPX	
4261	020634	000000			HALT		
4262	020636	005737	000672		TST	PFLG	; SEE IF HAVE PRINTED
4263	020642	001006			BNE	ERPX	; IF SO: BR
4264	020644	032777	002000	157736	BIT	#2000,@SWR	; SEE IF SHOULD PRINT
4265	020652	001002			BNE	ERPX	; IF NOT: BR
4266	020654	000137	020202		JMP	ERPT0	; PRINT ERROR
4267	020660	005037	000672		ERPX: CLR	PFLG	
4268	020664	005737	000566		TST	CRCC	; BRANCH IF CRC ERROR
4269	020670	001007			BNE	1\$	; CORRECTION DESIRED
4270	020672	012777	000040	157620	MOV	#40,@CS	; ELSE INIT
4271	020700	013777	000550	157612	MOV	DVN,@CS	; RESET DRIVE NUMBER
4272	020706	000414			BR	2\$	
4273	020710	012777	000011	157572	1\$: MOV	#11,@C1	; DRIVE CLEAR
4274	020716	017704	157604		MOV	@AS,R4	
4275	020722	010477	157600		MOV	R4,@AS	; CLEAR AS
4276	020726	013704	000510		MOV	C1,R4	
4277	020732	005204			INC	R4	
4278	020734	152714	000100		BISB	#100,(R4)	; RESET TRE
4279	020740	013777	000552	157574	2\$: MOV	UDES,@TC	; RESET TC
4280	020746	032737	000040	000674	ERPX1: BIT	#40, MTC1	
4281	020754	001415			BEQ	ERPX2	; IF NOT READ/WRITE OP: BR
4282	020756	005737	000700		TST	TMFLG	
4283	020762	001412			BEQ	ERPX2	; IF NOT TM TIME: BR
4284	020764	032737	002000	000552	BIT	#2000, UDES	; CHECK UDES
4285	020772	001006			BNE	ERPX2	; BR IF PE
4286	020774	013737	021034	015370	MOV	CRCSV,EXCRC	; RESTORE CRC
4287	021002	013737	021032	015372	MOV	LRCSV,EXLRC	; RESTORE LRC
4288	021010	000207			ERPX2: RTS	PC	; EXIT
4289	021012	000000			CADER:	0	; EXPT ADDRESS SAVE
4290	021014	000000			BAER:	0	
4291	021016	000000			CONER:	0	
4292	021020	000000			DRVER:	0	

4293 021022 000000
4294 021024 000000
4295 021026 000000
4296 021030 000000
4297 021032 000000
4298 021034 000000
4299

FCER: 0
LRCER: 0
CRCER: 0
ACTLRC: 0
LRCSV: 0
CRCSV: 0

4300
4301
4302
4303
4304
4305
4306
4307
4308

; 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.

4309 021036 032737 000010 000674 FRPRT:
4310 021044 001411
4311 021046 012704 024645
4312 021052 032737 000002 000674
4313 021060 001002
4314 021062 012704 024642
4315 021066 000004
4316 021070 000207
4317

BIT #10,MTC1 ;SEE IF WRITE COMMAND
BEQ 2\$;IF SO: BR
MOV #MSG17,R4 ;SET TO TYPE REVERSE MSG
BIT #2,MTC1 ;BRANCH IF REVERSE
BNE 1\$
MOV #MSG16,R4 ;SET FORWARD MESSAGE
1\$: TYPE ;TYPE MSG
2\$: RTS PC ;EXIT

```

4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347 021072 005037 000644 TAPG: CLR TEMP1
4348 021076 013777 000550 157414 MOV DVN,ACS ;SET DRIVE NO.
4349 021104 032777 010000 157410 TAPG0: BIT #10000,ADS ;SEE IF HAVE MOL
4350 021112 001026 BNE TAPG3 ;IF SO: BR
4351 021114 005237 000644 INC TEMP1 ;SEE IF TIMED OUT
4352 021120 001371 BNE TAPG0 ;WAIT FOR READY
4353 021122 004737 022602 JSR PC,PAPRT ;PRINT CYCLE NUMBER
4354 021126 032737 000010 000674 BIT #10,MTCL ;SEE IF WRITE OP
4355 021134 001004 BNE TAPG1 ;IF NOT: BR
4356 021136 012704 024502 MOV #MSG5,R4
4357 021142 000004 TYPE ;TYPE MSG
4358 021144 000405 BR TAPG2
4359 021146 012704 024507 TAPG1: MOV #MSG6,R4
4360 021152 000004 TYPE ;TYPE MSG
4361 021154 004737 021036 JSR PC,FRPRT ;PRINT F OR R
4362 021160 012704 025067 TAPG2: MOV #MSG25,R4
4363 021164 000004 TYPE ;TYPE MSG
4364 021166 000000 HALT
4365 021170 032777 020000 157324 TAPG3: BIT #20000,ADS ;SEE IF PIP RESET
4366 021176 001411 BEQ TAPG3F ;IF SO: BR
4367 021200 004737 022602 JSR PC,PAPRT ;PRINT HEADER
4368 021204 012704 027173 MOV #MSG116,R4
4369 021210 000004 TYPE ;TYPE MSG
4370 021212 032777 020000 157302 1$: BIT #20000,ADS
4371 021220 001374 BNE 1$ ;AWAIT PIP RESET
4372 021222 022737 000026 000674 TAPG3F: CMP #26,MTCL ;SEE IF WRITE TM
4373 021230 001003 BNE TAPG3A ;IF NOT: BR

```

4374	021232	012704	177777		MOV	#-1,R4	;ELSE SET FC FOR -1
4375	021236	000406			BR	TAPG3B	
4376	021240	013704	000556		TAPG3A: MOV	FMCNT,R4	
4377	021244	032704	000001		BIT	#1,R4	
4378	021250	001401			BEQ	TAPG3B	
4379	021252	005304			DEC	R4	
4380	021254	000261			TAPG3B: SEC		
4381	021256	006004			ROR	R4	;SET WC = FC/2 FOR NORMAL FORMAT
4382	021260	032737	000020	000552	BIT	#20,UDES	;SEE IF CORE DUMP FORMAT
4383	021266	001402			BEQ	TAPG3C	;IF NOT: BR
4384	021270	000261			SEC		
4385	021272	006004			ROR	R4	;SET WC = FC/4 FOR CORE DUMP
4386	021274	010477	157212		TAPG3C: MOV	R4,@WC	;SET WORD COUNT
4387	021300	012777	000011	157202	MOV	#11,@C1	;DRIVE CLEAR
4388	021306	017777	157204	157202	MOV	@FC,@FC	;RESET FC LOADED
4389	021314	005737	000570		TST	INTRF	;SEE IF INTERCHANGE READ
4390	021320	001407			BEQ	TAPG3D	;IF NOT: BR
4391	021322	032737	000040	000674	BIT	#40,MTCT	;SEE IF READ OP
4392	021330	001403			BEQ	TAPG3D	;IF NOT: BR
4393	021332	012777	000003	157174	MOV	#3,@MR	;SET INTERCHANGE READ MAINT. MODE
4394	021340	013704	000674		TAPG3D: MOV	MTCT,R4	;GET COMMAND
4395	021344	042704	177707		BIC	#177707,R4	;MASK OP CODE
4396	021350	022704	000030		CMP	#30,R4	;SEE IF SPACE OP CODE
4397	021354	001403			BEQ	TAPG3E	;IF SO: BR
4398	021356	012737	177740	000670	MOV	#-40,STAL	;SET INTERRUPT DELAY MULT TO 40
4399	021364	052737	000101	000674	TAPG3E: BIS	#101,MTCT	;SET INTERRUPT ENABLE AND GO
4400	021372	000240			NOP		
4401	021374	013777	000674	157106	MOV	MTCT,@C1	;EXECUTE COMMAND
4402	021402	005077	157200		CLR	@PSW	;CLEAR PRIORITY
4403	021406	005037	000644		CLR	TEMP1	
4404	021412	005237	000644		TAPG4: INC	TEMP1	;SEE IF HAVE TIMED OUT
4405	021416	001375			BNE	TAPG4	;IF NOT: BR
4406	021420	005237	000670		INC	STAL	
4407	021424	001372			BNE	TAPG4	;DO TIME DELAY MULTIPLIER
4408	021426	012777	000340	157152	TAPG5: MOV	#340,@PSW	;RESET PRIORITY
4409	021434	032777	002000	157146	BIT	#2000,@SWR	;SEE IF SHOULD PRINT ERRORS
4410	021442	001012			BNE	TAPG6	;IF NOT: BR
4411	021444	004737	022602		JSR	PC,PAPRT	;PRINT CYCLE NUMBER
4412	021450	013704	000654		MOV	EMADDR,R4	
4413	021454	000004			TYPE		;TYPE MSG
4414	021456	004737	021036		JSR	PC,FRPRT	;PRINT F OR R
4415	021462	012704	025047		MOV	#MSG24,R4	
4416	021466	000004			TYPE		;TYPE MSG
4417	021470	005777	157114		TAPG6: TST	@SWR	;BRANCH IF NOT HALT ON ERROR
4418	021474	100001			BPL	TAPG7	
4419	021476	000000			HALT		
4420	021500	000137	021736		TAPG7: JMP	MTINTA	;RETURN TO CALLING ROUTINE
4421							

```

4422
4423
4424 021504 017746 157104
4425 021510 042716 000200
4426 021514 122716 000003
4427 021520 001005
4428 021522 000005
4429 021524 005077 157056
4430 021530 000137 000200
4431 021534 122716 000001
4432 021540 001016
4433 021542 022737 000176 000610
4434 021550 001015
4435 021552 012737 177570 000610
4436 021560 004737 023306
4437 021564 012704 027317
4438 021570 000004
4439 021572 004737 023330
4440 021576 022716 000007
4441 021602 001005
4442 021604 012737 000176 000610
4443 021612 004737 023216
4444 021616 022716 000002
4445 021622 001042
4446 021624 004737 023306
4447 021630 005237 014070
4448 021634 004737 013642
4449 021640 032777 000040 156742
4450 021646 001426
4451 021650 012704 025642
4452 021654 000004
4453 021656 013703 000602
4454 021662 104400
4455 021664 012705 000602
4456 021670 012701 000007
4457 021674 012702 177777
4458 021700 012703 002000
4459 021704 004737 023352
4460 021710 004737 023330
4461 021714 005726
4462 021716 012716 011260
4463 021722 000002
4464 021724 004737 023330
4465 021730 005726
4466 021732 000002
4467
4468
4469 021734 000240
4470 021736 042777 000037 156570
4471 021744 013716 000664
4472 021750 000002

;TTY INTERRUPT HANDLER
TTINT: MOV @TKB,-(SP) ;GET CHARACTER
        BIC #200,(SP) ;STRIP PARITY BIT
        CMPB #3,(SP) ;BRANCH IF NOT C
        BNE 1$
        RESET ;RESET ALL I/O
        CLR @PSW ;CLEAR PSW
        JMP @#200 ;RESTART PROGRAM
1$: CMPB #1,(SP) ;BRANCH IF NOT A
    BNE 2$
    CMP #SWREG,SWR ;BRANCH IF HARDWARE SWR IS INVOKED
    BNE 3$
    MOV #177570,SWR ;INVOKE HARWARE SWR
    JSR PC,SAVE ;SAVE REGISTERS ON THE STACK
    MOV #MSG121,R4 ;TYPE 'HARDWARE SWR IN USE'
    TYPE
    JSR PC,RESTORE ;RESTORE REGISTERS
2$: CMP #7,(SP) ;BRANCH IF NOT G
    BNE 4$
3$: MOV #SWREG,SWR ;INVOKE SOFTWARE SWR
    JSR PC,GTSWR ;GET SWITCHES
4$: CMP #2,(SP) ;BRANCH IF NOT B
    BNE 6$
    JSR PC,SAVE ;SAVE REGISTERS ON THE STACK
    INC SCVFL ;SET FLAG
    JSR PC,TINP3A ;GO CHECK CRC CORRECTION
    BIT #40,@SWR ;BRANCH IF NOT YOZZLING
    BEQ 5$
    MOV #MSG44,R4 ;REQUEST NEW YOZZLE STALL
    TYPE ;TYPE MSG
    MOV YSTAL,R3
    TYPOCT ;PRINT PRESENT STALL
    MOV #YSTAL,R5 ;SET ADDRESS OF YSTL
    MOV #7,R1 ;SET NUMBER OF CHAR TO INPUT
    MOV #-1,R2 ;SET MAXIMUM LIMIT
    MOV #2000,R3 ;SET MINIMUM LIMIT
    JSR PC,TTR ;GO GET VALUE
    JSR PC,RESTORE ;RESTORE REGISTERS
    TST (SP)+ ;POP CHARACTER OF THE STACK
    MOV #YOZ,(SP) ;RETURN TO 'YOZ'
    RTI ;RETURN TO YOZ
5$: JSR PC,RESTORE
6$: TST (SP)+ ;POP CHARACTER OFF THE STACK
    RTI ;RETURN

;MAG TAPE INTERRUPT HANDLER
MTINT: NOP
MTINTA: BIC #37,@MR ;CLEAR MAINT MODE
        MOV RTRN,(SP) ;SET RETURN TO (RTRN)
        RTI ;RETURN

```

```

4473                                     :*****
4474                                     :AUTO SEQUENCE
4475                                     :
4476                                     :THIS ROUTINE ,ENTERED VIA STARTING ADDRESS 240
4477                                     :WILL EXERCISE ALL AVAILABLE SLAVES ON ALL AVAILABLE
4478                                     :DRIVES IN BOTH PE AND NRZ ACCORDING TO THE PRESELECTED
4479                                     :TEST PLAN. IF NRZ ONLY, PE TESTING WILL NOT BE ATTEMPTED.
4480                                     :*****
4481
4482 021752 012704 026615      ASEQ:  MOV      #MSG104,R4
4483 021756 000004                                     :TYPE MSG
4484 021760 012705 000744      MOV      #ASEQCF,R5      :SET ADDRESS OF ENTRY
4485 021764 012701 000002      MOV      #2,R1          :SET SIZE OF ENTRY
4486 021770 012702 000001      MOV      #1,R2          :SET UPPER LIMIT
4487 021774 012703 000000      MOV      #0,R3          :SET LOWER LIMIT
4488 022000 004737 023352      JSR      PC,TTR          :GO GET INPUT
4489 022004 005037 000740      ASEQ0:  CLR      ADRVN      :CLEAR DRV NUM
4490 022010 004737 022122      ASEQ1:  JSR      PC,HRDS     :GO SELECT HARDWARE CONFIGURATION
4491 022014 012704 026546      MOV      #MSG101,R4
4492 022020 000004                                     :TYPE MSG
4493 022022 012704 026575      MOV      #MSG102,R4
4494 022026 000004                                     :TYPE MSG
4495 022030 013703 000740      MOV      ADRVN,R3
4496 022034 104400                                     :PRINT TM03
4497 022036 012704 026604      MOV      #MSG103,R4
4498 022042 000004                                     :TYPE MSG
4499 022044 012700 000746      MOV      #UN1,R0          :POINT TO START OF SLAVE TABLE
4500 022050 005710      ASEQ2:  TST      (R0)            :SEE IF END
4501 022052 100403      BMI      ASEQ3                :IF SO: BR
4502 022054 012003      MOV      (R0)+,R3
4503 022056 104400      TYPOCT                                :PRINT SLAVE TABLE
4504 022060 000773      BR      ASEQ2                  :DO ALL
4505 022062 004737 022306      ASEQ3:  JSR      PC,AMOD1     :GO DO MODE 1(NRZ)
4506 022066 004737 022446      JSR      PC,AMOD2     :GO DO MODE 2(PE)
4507 022072 022737 000007 000740 ASEQ4:  CMP      #7,ADRVN   :SEE IF DONE ALL DRIVES
4508 022100 001403      BEQ      ASEQX                :IF SO: BR
4509 022102 005237 000740      INC      ADRVN          :BUMP DRIVE NUMBER
4510 022106 000740      BR      ASEQ1                  :CONTINUE
4511 022110 005737 000744      ASEQX:  TST      ASEQCF     :SEE IF CONTINUOUS AUTO SEQ
4512 022114 001333      BNE      ASEQ0                :++B CONTINUE TESTING
4513 022116 000137 005004      JMP      TEND

```

```

4514
4515
4516
4517 022122 005037 005054 HRDS: CLR REOTC ;CLEAR EOT UNIT CNTR
4518 022126 005037 000644 CLR TEMP1
4519 022132 012777 000040 156360 MOV #40,ACS ;INIT
4520 022140 013777 000740 156352 MOV ADRVN,ACS ;SET DRIVE
4521 022146 032777 010000 156344 BIT #10000,ACS ;TEST FOR NON-EXISTANT DRIVE
4522 022154 001403 BEQ 2$ ;IF DRIVE AVAIL: BR
4523 022156 005726 1$: TST (SP)+ ;RESET STACK POINTER
4524 022160 000137 022072 JMP ASEQ4 ;GO SEE IF TRIED ALL DRIVES
4525 022164 005000 2$: CLR RO
4526 022166 012701 000746 MOV #UN1,R1 ;SET START OF SLAVE TABLE
4527 022172 005737 003040 TST CHNFLG ;BRANCH IF NOT IN CHAIN MODE
4528 022176 001410 BEQ 3$
4529 022200 122737 000006 000041 CMPB #6,@#41 ;BRANCH IF NOT LOADED VIA TMDP
4530 022206 001004 BNE 3$
4531 022210 005737 000740 TST ADRVN ;BRANCH IF NOT DRIVE 0
4532 022214 001001 BNE 3$
4533 022216 005200 INC RO ;DO NOT TEST SLAVE 0
4534 022220 010077 156316 3$: MOV RO,ATC ;SELECT SLAVE
4535 022224 032777 010000 156270 BIT #10000,ADS ;SEE IF SLAVE AVAIL FOR TEST(MOL)
4536 022232 001403 BEQ 4$ ;IF NOT: BR
4537 022234 005237 000644 INC TEMP1 ;SET SLAVE FOUND FLAG
4538 022240 010021 MOV RO,(R1)+ ;LOAD SLAVE TABLE
4539 022242 005200 4$: INC RO ;STEP TO NEXT SLAVE
4540 022244 022700 000010 CMP #10,RO ;BRANCH IF ALL SLAVE NOT DONE
4541 022250 001363 BNE 3$
4542 022252 005737 000644 5$: TST TEMP1 ;SEE IF FOUND ANY SLAVES
4543 022256 001737 BEQ 1$ ;IF NOT: BR
4544 022260 013737 000644 005054 MOV TEMP1,REOTC ;SET NUMBER OF UNITS
4545 022266 000337 000644 SWAB TEMP1
4546 022272 053737 000644 005054 BIS TEMP1,REOTC ;SET EOT CNTR
4547 022300 012711 177777 MOV #-1,(R1) ;TERMINATE SLAVE TABLE
4548 022304 000207 RTS PC ;RETURN TO SEQ

```

```

4549
4550                                     ;SUBROUTINE TO SELECT NRZ AUTO TEST MODE*****
4551
4552 022306 005037 000656          AMOD1: CLR      BLCNTR      ;ASSURE BLOCK COUNTER IS 0
4553 022312 012701 000746          MOV      #UN1,R1      ;GET START OF SLAVE TABLE
4554 022316 052721 001700          1$:  BIS      #1700,(R1)+  ;SET ALL SLAVE TO NRZ,NORM,ODD
4555 022322 022711 177777          CMP      #-1,(R1)      ;LOOP UNTIL REACED END OF TABLE
4556 022326 001373                BNE      1$
4557 022330 004737 005070          JSR      PC,RWDA      ;GO REWIND ALL AVAIL SLAVES
4558 022334 012737 000050 000742  MOV      #50,ABLCNT  ;SET NUMBER OF BLOCKS FOR MODE 1
4559 022342 012737 174000 000556  MOV      #-4000,FMCNT ;SET FC = 4000
4560 022350 012737 000100 000554  MOV      #100,RCNT   ;SET REC CNTR = 100
4561 022356 013737 000740 000550  MOV      ADRVN,DVN   ;SELECT DRIVE
4562 022364 012737 000001 000560  MOV      #1,PATRN   ;SELECT PATTERN 1
4563 022372 005037 000564          CLR      TMEX        ;ASSURE NO TMK
4564 022376 005037 000570          CLR      INTRF       ;ASSURE NORMAL READ
4565 022402 004737 003346          JSR      PC,STAUTO    ;GO DO AUTO MODE 1
4566 022406 012737 000010 000560  MOV      #10,PATRN   ;SELECT PATTERN 10
4567 022414 004737 003346          JSR      PC,STAUTO    ;GO DO PATTERN 10
4568 022420 012737 000014 000560  MOV      #14,PATRN   ;SELECT PATTERN 14
4569 022426 004737 003346          JSR      PC,STAUTO
4570 022432 012737 177777 000560  3$:  MOV      #-1,PATRN ;SELECT AUTO RANDOM DATA
4571 022440 004737 003346          JSR      PC,STAUTO
4572 022444 000207                RTS      PC          ;RETURN TO SEQ

```

```

4573
4574
4575
4576 022446 005037 000656 AMOD2: CLR BLCNTR ;CLEAR BLOCK CNTR
4577 022452 012701 000746 MOV #UN1,R1 ;SET START OF SLAVE TABLE
4578 022456 042711 001700 1$: BIC #1700,(R1) ;CLEAR NRZ
4579 022462 052721 002300 BIS #2300,(R1)+ ;SET TO PE NORM, ODD
4580 022466 022711 177777 CMP #-1,(R1) ;LOOP UNTIL END OF TABLE
4581 022472 001371 BNE 1$
4582 022474 004737 005070 JSR PC,RWVDA ;REWIND ALL SLAVES
4583 022500 012737 000006 000742 MOV #6,ABLCNT ;SET AUTO BLOCK COUNT
4584 022506 012737 174000 000556 MOV #-4000,FMCNT ;SET FC = 4000
4585 022514 012737 000100 000554 MOV #100,RCNT ;SET REC CNTR TO 100
4586 022522 012737 000010 000560 MOV #10,PATRN ;SELECT PATTERN 10
4587 022530 004737 003346 JSR PC,STAUTO ;GO DO AUTO SEQ
4588 022534 012737 000014 000560 MOV #14,PATRN ;SELECT PATTERN 14
4589 022542 004737 003346 JSR PC,STAUTO
4590 022546 012737 000015 000560 MOV #15,PATRN ;SELECT PATTERN 15
4591 022554 004737 003346 JSR PC,STAUTO
4592 022560 012737 177777 000742 MOV #-1,ABLCNT ;FORCE TO END OF TAPE
4593 022566 012737 177777 000560 MOV #-1,PATRN ;SELECT AUTO RANDOM DATA
4594 022574 004737 003346 JSR PC,STAUTO
4595 022600 000207 3$: RTS PC ;RETURN TO SEQ
4596
4597

```

```

4598
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4610
4611
4612
4613
4614 022602 012704 024560
4615 022606 000004
4616 022610 013703 000550
4617 022614 104400
4618 022616 012704 024544
4619 022622 000004
4620 022624 013703 000552
4621 022630 042703 177770
4622 022634 104400
4623 022636 012704 026124
4624 022642 000004
4625 022644 013703 000552
4626 022650 000303
4627 022652 042703 177770
4628 022656 104400
4629 022660 012704 026130
4630 022664 000004
4631 022666 005003
4632 022670 032737 000010 000552
4633 022676 001402
4634 022700 012703 000001
4635 022704 104400
4636 022706 012704 026134
4637 022712 000004
4638 022714 013703 000552
4639 022720 000241
4640 022722 006003
4641 022724 006003
4642 022726 006003
4643 022730 006003
4644 022732 042703 177760
4645 022736 104400
4646 022740 012704 024521
4647 022744 000004
4648 022746 032777 000400 155634
4649 022754 001406
4650 022756 012737 000122 000640
4651 022764 004737 023756
4652 022770 000411
4653 022772 005737 000736

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

PAPRT: MOV #MSG12,R4
      TYPE ;TYPE MSG
      MOV DVN,R3
      TYPOCT ;PRINT DRIVE NUMBER
      MOV #MSG11,R4
      TYPE ;TYPE MSG
      MOV UDES,R3
      BIC #177770,R3
      TYPOCT ;PRINT UNIT NUMBER
      MOV #MSG60,R4
      TYPE ;TYPE MSG
      MOV UDES,R3
      SWAB R3
      BIC #177770,R3
      TYPOCT ;PRINT DENSITY
      MOV #MSG61,R4
      TYPE ;TYPE MSG
      CLR R3
      BIT #10,UDES
      BEQ PAPRTU
      MOV #1,R3
      TYPOCT ;PRINT PARITY
      MOV #MSG62,R4
      TYPE ;TYPE MSG
      MOV UDES,R3
      CLC
      ROR R3
      ROR R3
      ROR R3 ;PONTION FORMAT
      ROR R3
      BIC #177760,R3
      TYPOCT ;PRINT FORMAT
      MOV #MSG8,R4
      TYPE ;TYPE MSG
      BIT #400,@SWR ;SEE IF RANDOM DATA
      BEQ PAPRTB ;IF NOT: BR
      MOV #122,TOB
      JSR PC,TOG ;PRINT R
      BR PAPRTD
      TST ASEQF ;SEE IF AUTO SEQ

```

Address	Op Code	Op 1	Op 2	Op 3	Op 4	Op 5	Op 6	Op 7	Op 8	Op 9	Op 10	Op 11	Op 12	Op 13	Op 14	Op 15	Op 16	Op 17	Op 18	Op 19	Op 20	Op 21	Op 22	Op 23	Op 24	Op 25	Op 26	Op 27	Op 28	Op 29	Op 30	Op 31	Op 32	Op 33	Op 34	Op 35	Op 36	Op 37	Op 38	Op 39	Op 40	Op 41	Op 42	Op 43	Op 44	Op 45	Op 46	Op 47	Op 48	Op 49	Op 50	Op 51	Op 52	Op 53	Op 54	Op 55	Op 56	Op 57	Op 58	Op 59	Op 60	Op 61	Op 62	Op 63	Op 64	Op 65	Op 66	Op 67	Op 68	Op 69	Op 70	Op 71	Op 72	Op 73	Op 74	Op 75	Op 76	Op 77	Op 78	Op 79	Op 80	Op 81	Op 82	Op 83	Op 84	Op 85	Op 86	Op 87	Op 88	Op 89	Op 90	Op 91	Op 92	Op 93	Op 94	Op 95	Op 96	Op 97	Op 98	Op 99	Op 100	Op 101	Op 102	Op 103	Op 104	Op 105	Op 106	Op 107	Op 108	Op 109	Op 110	Op 111	Op 112	Op 113	Op 114	Op 115	Op 116	Op 117	Op 118	Op 119	Op 120	Op 121	Op 122	Op 123	Op 124	Op 125	Op 126	Op 127	Op 128	Op 129	Op 130	Op 131	Op 132	Op 133	Op 134	Op 135	Op 136	Op 137	Op 138	Op 139	Op 140	Op 141	Op 142	Op 143	Op 144	Op 145	Op 146	Op 147	Op 148	Op 149	Op 150	Op 151	Op 152	Op 153	Op 154	Op 155	Op 156	Op 157	Op 158	Op 159	Op 160	Op 161	Op 162	Op 163	Op 164	Op 165	Op 166	Op 167	Op 168	Op 169	Op 170	Op 171	Op 172	Op 173	Op 174	Op 175	Op 176	Op 177	Op 178	Op 179	Op 180	Op 181	Op 182	Op 183	Op 184	Op 185	Op 186	Op 187	Op 188	Op 189	Op 190	Op 191	Op 192	Op 193	Op 194	Op 195	Op 196	Op 197	Op 198	Op 199	Op 200	Op 201	Op 202	Op 203	Op 204	Op 205	Op 206	Op 207	Op 208	Op 209	Op 210	Op 211	Op 212	Op 213	Op 214	Op 215	Op 216	Op 217	Op 218	Op 219	Op 220	Op 221	Op 222	Op 223	Op 224	Op 225	Op 226	Op 227	Op 228	Op 229	Op 230	Op 231	Op 232	Op 233	Op 234	Op 235	Op 236	Op 237	Op 238	Op 239	Op 240	Op 241	Op 242	Op 243	Op 244	Op 245	Op 246	Op 247	Op 248	Op 249	Op 250	Op 251	Op 252	Op 253	Op 254	Op 255	Op 256	Op 257	Op 258	Op 259	Op 260	Op 261	Op 262	Op 263	Op 264	Op 265	Op 266	Op 267	Op 268	Op 269	Op 270	Op 271	Op 272	Op 273	Op 274	Op 275	Op 276	Op 277	Op 278	Op 279	Op 280	Op 281	Op 282	Op 283	Op 284	Op 285	Op 286	Op 287	Op 288	Op 289	Op 290	Op 291	Op 292	Op 293	Op 294	Op 295	Op 296	Op 297	Op 298	Op 299	Op 300	Op 301	Op 302	Op 303	Op 304	Op 305	Op 306	Op 307	Op 308	Op 309	Op 310	Op 311	Op 312	Op 313	Op 314	Op 315	Op 316	Op 317	Op 318	Op 319	Op 320	Op 321	Op 322	Op 323	Op 324	Op 325	Op 326	Op 327	Op 328	Op 329	Op 330	Op 331	Op 332	Op 333	Op 334	Op 335	Op 336	Op 337	Op 338	Op 339	Op 340	Op 341	Op 342	Op 343	Op 344	Op 345	Op 346	Op 347	Op 348	Op 349	Op 350	Op 351	Op 352	Op 353	Op 354	Op 355	Op 356	Op 357	Op 358	Op 359	Op 360	Op 361	Op 362	Op 363	Op 364	Op 365	Op 366	Op 367	Op 368	Op 369	Op 370	Op 371	Op 372	Op 373	Op 374	Op 375	Op 376	Op 377	Op 378	Op 379	Op 380</
---------	---------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	----------

```

4692
4693
4694
4695
4696
4697
4698
4699
4700
4701 023164 063737 000630 000626 RANG: ADD RANSAV,RANBAS
4702 023172 063737 000626 000630 ADD RANBAS,RANSAV ;GET NEW NUMBER
4703 023200 023701 000630 CMP RANSAV,R1 ;SEE IF NUMBER TOO BIG
4704 023204 101367 BHI RANG ;IF SO: BR
4705 023206 020237 000630 CMP R2,RANSAV ;SEE IF NUMBER TOO SMALL
4706 023212 101364 BHI RANG ;IF SO: BR
4707 023214 000207 RTS PC ;EXIT
4708
4709 ;SUBROUTINE TO GET NEW SOFTWARE SWR
4710
4711 023216 022737 000176 000610 GTSWR: CMP #SWREG,SWR ;BRANCH IF SOFTWARE SWR
4712 023224 001027 BNE 1$ ;NOT INVOKED
4713 023226 004737 023306 JSR PC,.SAVE ;SAVE REGISTERS ON THE STACK
4714 023232 012704 024436 MOV #SMSWR,R4 ;TYPE 'SWR = '
4715 023236 000004 TYPE ;TYPE MSG
4716 023240 017703 155344 MOV @SWR,R3 ;GET CURRENT SWR
4717 023244 104400 TYPOCT
4718 023246 012704 024446 MOV #SMNEW,R4 ;ASK FOR NEW SETTING
4719 023252 000004 TYPE ;TYPE MSG
4720 023254 013705 000610 MOV SWR,R5 ;TTR ROUTINE RETURNS VALUE TO (R5)
4721 023260 012701 000007 MOV #7,R1 ;LIMIT RESPONSE TO 7 CHARS
4722 023264 012702 177777 MOV #177777,R2 ;BETWEEN 0 AND 177777
4723 023270 012703 000000 MOV #0,R3
4724 023274 004737 023352 JSR PC,TTR ;GET RESPONSE
4725 023300 004737 023330 JSR PC,.RESTORE ;RESTORE REGISTERS
4726 023304 000207 1$: RTS PC ;RETURN
4727
4728 ;;ROUTINE TO SAVE REGISTERS ON THE STACK
4729 023306 010546 .SAVE: MOV X5,-(SP) ;;R5 IS SAVED AT 12(SP)
4730 023310 010446 MOV X4,-(SP) ;;R4 IS SAVED AT 10(SP)
4731 023312 010346 MOV X3,-(SP) ;;R3 IS SAVED AT 6(SP)
4732 023314 010246 MOV X2,-(SP) ;;R2 IS SAVED AT 4(SP)
4733 023316 010146 MOV X1,-(SP) ;;R1 IS SAVED AT 2(SP)
4734 023320 010046 MOV X0,-(SP) ;;R0 IS SAVED AT (SP)
4735 023322 016646 000014 MOV 14(SP),-(SP) ;;PUSH RETURN PC ON THE STACK
4736 023326 000207 RTS PC ;;RETURN TO CALLER
4737
4738 ;;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
4739 023330 012666 000014 .RESTORE:MOV (SP)+,14(SP) ;;STORE RETURN PC ON STACK
4740 023334 012600 MOV (SP)+,X0
4741 023336 012601 MOV (SP)+,X1
4742 023340 012602 MOV (SP)+,X2
4743 023342 012603 MOV (SP)+,X3
4744 023344 012604 MOV (SP)+,X4
4745 023346 012605 MOV (SP)+,X5
4746 023350 000207 RTS PC ;;RETURN
4747

```

```

4748 ;*****
4749 ;TTY ENTRY SUBROUTINE:
4750 ;
4751 ;THIS SUBROUTINE IS USED BY THE TEST CONDITION
4752 ;ENTRY ROUTINE TO READ THE RESPONSE ENTERED
4753 ;AT THE TTY AND CHECK THEM FOR LEGALITY AND
4754 ;LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL
4755 ; (0-7) AND MUST FALL WITHIN THE LIMITS SET BY
4756 ;THE CALLING ROUTINE.
4757 ;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
4758 ;A QUESTION MARK IS TYPED (?) AND THE RESPONSE
4759 ;MAY BE REENTERED.
4760 ;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
4761 ;MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
4762 ;CARRIAGE RETURN
4763 ;*****
4764
4765 023352 010146 TTR: MOV R1,-(SP) ;SAVE CHAR COUNT
4766 023354 011601 10$: MOV (SP),R1 ;RESTORE CHAR COUNT (FOR U)
4767 023356 005037 000644 CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
4768 023362 005000 CLR R0
4769 023364 004737 023604 1$: JSR PC,TTIN ;GO READ CHARACTER
4770 023370 122737 000003 000642 CMPB #3,TIB ;BRANCH IF NOT C
4771 023376 001003 BNE 11$
4772 023400 000005 RESET
4773 023402 000137 000200 JMP @#200 ;RESTART AT 200
4774 023406 122737 000015 000642 11$: CMPB #15,TIB ;SEE IF CR
4775 023414 001004 BNE 2$ ;IF NOT: BR
4776 023416 005737 000644 TST TEMP1 ;SEE IF FIRST CHARACTER
4777 023422 001457 BEQ 9$ ;IF SO: BR
4778 023424 000451 BR 6$ ;ELSE GO LOAD VALUE
4779 023426 122737 000025 000642 2$: CMPB #25,TIB ;BRANCH IF NOT CONTROL U
4780 023434 001004 BNE 21$
4781 023436 012704 025131 MOV #MSG28,R4 ;TYPE <CR><LF>
4782 023442 000004 TYPE ;TYPE MSG
4783 023444 000743 BR 10$
4784 023446 122737 000177 000642 21$: CMPB #177,TIB ;BRANCH IF NOT 'RUBOUT'
4785 023454 001011 BNE 3$
4786 023456 000241 CLC ;REMOVE LAST CHARACTER
4787 023460 006000 ROR R0
4788 023462 006200 ASR R0
4789 023464 006200 ASR R0
4790 023466 012704 027247 MOV #MSG118,R4 ;TYPE ' '
4791 023472 000004 TYPE ;TYPE MSG
4792 023474 005201 INC R1 ;DEC CHAR RECEIVED COUNT
4793 023476 000732 BR 1$ ;GET NEXT CHARACTER
4794 023500 122737 000060 000642 3$: CMPB #60,TIB ;SEE IF CHAR IS LESS THAN 0
4795 023506 101027 BHI TIB
4796 023510 122737 000070 000642 4$: CMPB #70,TIB ;SEE IF CHAR IS GREATER THAN 7
4797 023516 101423 BLOS TIB
4798 023520 005237 000644 5$: INC TEMP1 ;SET FIRST CHARACTER FLAG
4799 023524 006300 ASL R0
4800 023526 006300 ASL R0 ;SHIFT 3 LEFT
4801 023530 006300 ASL R0
4802 023532 042737 177770 000642 BIC #177770,TIB ;STRIP ASCII
4803 023540 053700 000642 BIS TIB,R0 ;LOAD CHARACTER

```

4804	023544	005301		DEC	R1	;SEE IF DONE
4805	023546	001306		BNE	1\$	;IF NOT: BR
4806	023550	020002	6\$:	CMP	R0,R2	;SEE IF EXCEEDED MAXIMUM LIMIT
4807	023552	101005		BHI	TINER	
4808	023554	020300	7\$:	CMP	R3,R0	;SEE IF BELOW MINIMUM LIMIT
4809	023556	101003		BHI	TINER	
4810	023560	010015	8\$:	MOV	R0,(R5)	;LOAD VALUE
4811	023562	005726	9\$:	TST	(SP)+	;POP CHAR COUNT OFF STACK
4812	023564	000207		RTS	PC	;EXIT
4813						
4814	023566	012704	025636	TINER:	MOV #MSG43,R4	
4815	023572	000004		TYPE		;TYPE MSG
4816	023574	005726		TST	(SP)+	;POP CHAR COUNT OFF STACK
4817	023576	162716	000020	SUB	#20,(SP)	;RESET SP TO START OF VALUE ROUTINE
4818	023602	000207		RTS	PC	;REDO VALUE ENTRY

```

4819
4820
4821
4822 023604 005277 155002
4823 023610 105777 154776
4824 023614 100375
4825 023616 017737 154772 000642
4826 023624 042737 000200 000642
4827 023632 013737 000642 000640
4828 023640 004737 023756
4829 023644 000207
4830
4831
4832
4833 023646 112437 000640
4834 023652 105737 000640
4835 023656 001436
4836 023660 122737 000045 000640
4837 023666 001407
4838 023670 122737 000041 000640
4839 023676 001436
4840 023700 004737 023756
4841 023704 000760
4842 023706 112737 000015 000640
4843 023714 004737 023756
4844 023720 012703 000006
4845 023724 005037 000640
4846 023730 004737 023756
4847 023734 005303
4848 023736 001372
4849 023740 112737 000012 000640
4850 023746 004737 023756
4851 023752 000735
4852 023754 000002
4853
4854 023756 105777 154634
4855 023762 100375
4856 023764 113777 000640 154626
4857 023772 000207
4858
4859 023774 012703 000002
4860 024000 012737 000007 000640
4861 024006 004737 023756
4862 024012 005303
4863 024014 001371
4864 024016 000713
4865
4866

;TTY READ SUBROUTINE*****
TTIN: INC @TKS
1$: TSTB @TKS
BPL 1$
MOV @TKB,TIB
BIC #200,TIB ;STRIP PARITY BIT
MOV TIB,TOB ;MOVE CHAR TO TTY OUTPUT BFR
JSR PC,TOG ;ECHO CHARACTER
RTS PC

;TTY OUTPUT SUBROUTINE*****
TTOUT: MOVB (R4)+,TOB
TSTB TOB
BEQ 3$
CMPB #45,TOB
BEQ 1$
CMPB #41,TOB
BEQ TBELL ;DO BELL
JSR PC,TOG
BR TTOUT
1$: MOVB #15,TOB
JSR PC,TOG
MOV #6,R3
2$: CLR TOB
JSR PC,TOG
DEC R3
BNE 2$ ;DO FILLERS
MOVB #12,TOB
JSR PC,TOG
BR TTOUT
3$: RTI ;RETURN

TOG: TSTB @TPS
BPL TOG
MOVB TOB,@TPB
RTS PC ;RETURN

TBELL: MOV #2,R3
1$: MOV #7,TOB
JSR PC,TOG
DEC R3
BNE 1$
BR TTOUT

```

```

4867                                     ;OCTAL OUTPUT SUBROUTINE*****
4868
4869 024020 005037 024232      OCTP:  CLR      OFL          ;CLEAR FLAG FOR LEADING ZERO
4870 024024 010304              MOV      R3,R4          ;SEE IF NUMBER IS ZERO
4871 024026 001003              BNE      OCTP0          ;IF NOT ZERO: BR
4872 024030 004737 024212      JSR      PC,OCTPG1        ;ELSE PRINT ZERO
4873 024034 000447              BR       OCTP3          ;SPACE AND EXIT
4874 024036 005704              OCTP0: TST      R4          ;BRANCH IF MSD = 0
4875 024040 100006              BPL      OCTP1
4876 024042 012704 000001      MOV      #1,R4
4877 024046 004737 024170      JSR      PC,OCTPG        ;PRINT 1
4878 024052 000137 024064      JMP      OCTP2
4879 024056 005004              OCTP1: CLR      R4
4880 024060 004737 024170      JSR      PC,OCTPG        ;PRINT 0
4881 024064 010304              OCTP2: MOV      R3,R4
4882 024066 006004              ROR      R4
4883 024070 006004              ROR      R4
4884 024072 006004              ROR      R4          ;POSITION DIGIT
4885 024074 006004              ROR      R4
4886 024076 000304              SWAB     R4
4887 024100 004737 024170      JSR      PC,OCTPG        ;PRINT DIGIT 2
4888 024104 010304              MOV      R3,R4
4889 024106 006004              ROR      R4
4890 024110 000304              SWAB     R4
4891 024112 004737 024170      JSR      PC,OCTPG        ;PRINT DIGIT 3
4892 024116 010304              MOV      R3,R4
4893 024120 006104              ROL      R4
4894 024122 006104              ROL      R4
4895 024124 000304              SWAB     R4
4896 024126 004737 024170      JSR      PC,OCTPG        ;PRINT DIGIT 4
4897 024132 010304              MOV      R3,R4
4898 024134 006004              ROR      R4
4899 024136 006004              ROR      R4
4900 024140 006004              ROR      R4
4901 024142 004737 024170      JSR      PC,OCTPG
4902 024146 010304              MOV      R3,R4
4903 024150 004737 024170      JSR      PC,OCTPG        ;PRINT DIGIT 5
4904 024154 012737 000240 000640 OCTP3: MOV      #240,TOB
4905 024162 004737 023756      JSR      PC,TOG          ;PRINT SPACE
4906 024166 000002              RTI              ;EXIT
4907 024170 042704 177770      OCTPG: BIC      #177770,R4
4908 024174 001004              BNE      OCTPG0
4909 024176 005737 024232      TST      OFL
4910 024202 001001              BNE      OCTPG0
4911 024204 000207              RTS      PC
4912
4913 024206 005237 024232      OCTPG0: INC      OFL
4914 024212 052704 000260      OCTPG1: BIS      #260,R4
4915 024216 010437 000640      MOV      R4,TOB
4916 024222 004737 023756      JSR      PC,TOG
4917 024226 010304              MOV      R3,R4
4918 024230 000207              RTS      PC
4919 024232 000000      OFL:      0          ;FIRST CHAR FLAG
4920

```

```

4921
4922
4923 ;DATA CHARACTER OUTPUT SUBROUTINE*****
4924 024234 012704 000010 DOUT: MOV #10,R4 ;SET NUMBER TO PRINT
4925 024240 110337 000640 MOV R3,TOB
4926 024244 105777 154346 1$: TSTB @TPS
4927 024250 100375 BPL 1$
4928 024252 105737 000640 TSTB TOB
4929 024256 100004 BPL 2$
4930 024260 012777 000061 154332 MOV #061,@TPB
4931 024266 000403 BR 3$
4932 024270 012777 000060 154322 2$: MOV #060,@TPB
4933 024276 006337 000640 3$: ASL TOB
4934 024302 005304 DEC R4
4935 024304 001357 BNE 1$
4936 024306 000207 RTS PC
4937
4938 024310 013703 000650 DOUTD: MOV TEMP3,R3
4939 024314 000303 SWAB R3
4940 024316 004737 024234 JSR PC,DOUT
4941 024322 013703 000650 MOV TEMP3,R3
4942 024326 004737 024234 JSR PC,DOUT
4943 024332 000207 RTS PC
4944
4945 ;TU45 SERIAL NUMBER PRINT SUBROUTINE*****
4946
4947 024334 010304 SNPT: MOV R3,R4
4948 024336 000304 SWAB R4
4949 024340 006004 ROR R4
4950 024342 006004 ROR R4
4951 024344 006004 ROR R4
4952 024346 006004 ROR R4
4953 024350 004737 024412 JSR PC,SNPG ;PRINT FIRST DIGIT
4954 024354 010304 MOV R3,R4
4955 024356 000304 SWAB R4
4956 024360 004737 024412 JSR PC,SNPG ;PRINT SECOND DIGIT
4957 024364 010304 MOV R3,R4
4958 024366 006004 ROR R4
4959 024370 006004 ROR R4
4960 024372 006004 ROR R4
4961 024374 006004 ROR R4
4962 024376 004737 024412 JSR PC,SNPG ;PRINT THIRD DIGIT
4963 024402 010304 MOV R3,R4
4964 024404 004737 024412 JSR PC,SNPG ;PRINT FOURTH DIGIT
4965 024410 000207 RTS PC ;EXIT
4966 024412 012737 000260 000640 SNPG: MOV #260,TOB ;SET NUMBER BASE
4967 024420 042704 177760 BIC #177760,R4 ;MASK NUMBER
4968 024424 050437 000640 BIS R4,TOB ;BUILD DIGIT
4969 024430 004737 023756 JSR PC,TOG ;GO TYPE
4970 024434 000207 RTS PC ;RETURN
4971

```

```

4972
4973                                     ;ERROR MESSAGES*****
4974
4975 024436 051445 051127 036440 $MSWR: .ASCIZ /%SWR = /
4976 024444 000040
4977 024446 047040 053505 036440 $MNEW: .ASCIZ / NEW = /
4978 024454 000040
4979 024456 042052 020105 000 MSG1: .ASCIZ /*DE /
4980 024463 045 035507 000040 MSG2: .ASCIZ /%G; /
4981 024470 041045 020073 000 MSG3: .ASCIZ /%B; /
4982 024475 045 047103 000040 MSG4: .ASCIZ /%CN /
4983 024502 053452 020105 000 MSG5: .ASCIZ /*WE /
4984 024507 052 042522 000040 MSG6: .ASCIZ /*RE /
4985 024514 051052 020123 000 MSG7: .ASCIZ /*RS /
4986 024521 052 040520 051124 MSG8: .ASCIZ /*PATRN /
4987 024526 020116 000
4988 024531 045 047123 020072 MSG9: .ASCIZ /%SN: /
4989 024536 000
4990 024537 052 042523 000040 MSG10: .ASCIZ /*SE /
4991 024544 051452 040514 042526 MSG11: .ASCIZ /*SLAVE NO. /
4992 024552 047040 027117 000040
4993 024560 022445 042045 044522 MSG12: .ASCIZ /%%DRIVE NO. /
4994 024566 042526 047040 027117
4995 024574 000040
4996 024576 025045 047102 000040 MSG13: .ASCIZ /%*BN /
4997 024604 051052 020116 000 MSG14: .ASCIZ /*RN /
4998 024611 045 020041 020040 MSG15: .ASCIZ /%! BAD RECORD%%/
4999 024616 020040 020040 020040
5000 024624 041040 042101 051040
5001 024632 041505 051117 022504
5002 024640 000045
5003 024642 043040 000 MSG16: .ASCIZ / F/
5004 024645 040 000122 MSG17: .ASCIZ / R/
5005 024650 020041 047505 020124 MSG20: .ASCIZ /! EOT NO: /
5006 024656 047516 020072 000
5007
5008 024663 045 047111 042524 MSG21: .ASCIZ /%INTERCHANGE READ = /
5009 024670 041522 040510 043516
5010 024676 020105 042522 042101
5011 024704 036440 000040
5012 024710 020445 046111 042514 MSG22: .ASCIZ /%!ILLEGAL BOT: HALT%%/
5013 024716 040507 020114 047502
5014 024724 035124 044040 046101
5015 024732 022524 022445 000
5016 024737 045 051503 020061 MSG23: .ASCIZ /%CS1 /
5017 024744 000
5018 024745 045 041527 000040 MSG23A: .ASCIZ /%WC /
5019 024752 041045 020101 000 MSG23B: .ASCIZ /%BA /
5020 024757 045 041506 000040 MSG23C: .ASCIZ /%FC /
5021 024764 041445 031123 000040 MSG23D: .ASCIZ /%CS2 /
5022 024772 042045 020123 000 MSG23E: .ASCIZ /%DS /
5023 024777 045 051105 000040 MSG23F: .ASCIZ /%ER /
5024 025004 040445 020123 000 MSG23G: .ASCIZ /%AS /
5025 025011 045 045503 000040 MSG23H: .ASCIZ /%CK /
5026 025016 042045 020102 000 MSG23I: .ASCIZ /%DB /
5027 025023 045 051115 000040 MSG23J: .ASCIZ /%MR /

```

5028	025030	042045	020124	000	MSG23K: .ASCIZ /%DT /
5029	025035	045	041524	000040	MSG23L: .ASCIZ /%TC /
5030	025042	051445	020116	000	MSG23M: .ASCIZ /%SN /
5031	025047	045	047041	020117	MSG24: .ASCIZ /%!NO INTERRUPT%/
5032	025054	047111	042524	051122	
5033	025062	050125	022524	000	
5034	025067	045	047041	020117	MSG25: .ASCIZ /%!NO MOL: HALT%/
5035	025074	047515	035114	044040	
5036	025102	046101	022524	000	
5037	025107	045	051104	050117	MSG26: .ASCIZ /%DROPS: /
5038	025114	035123	000040		
5039	025120	050045	041511	051513	MSG27: .ASCIZ /%PICKS: /
5040	025126	020072	000		
5041	025131	045	000		MSG28: .ASCIZ /%/
5042	025133	045	052045	047515	MSG30: .ASCIZ '%TM03-TU45 AUTO SEQUENCE (CZTURA0)%';++B
5043	025140	026463	052524	032464	
5044	025146	040440	052125	020117	
5045	025154	042523	052521	047105	
5046	025162	042503	024040	055103	
5047	025170	052524	040522	024460	
5048	025176	000045			
5049	025200	022445	046524	031460	MSG31: .ASCIZ '%TM03-TU45 DATA RELIABILITY TEST (CZTURA0)%';++B
5050	025206	052055	032125	020065	
5051	025214	040504	040524	051040	
5052	025222	046105	040511	044502	
5053	025230	044514	054524	052040	
5054	025236	051505	020124	041450	
5055	025244	052132	051125	030101	
5056	025252	022451	000		
5057	025255	124	050131	020105	MSG31A: .ASCIZ /%TYPE <CR> TO TERMINATE ALL REQUESTS & C TO RESTART%/
5058	025262	041474	037122	052040	
5059	025270	020117	042524	046522	
5060	025276	047111	052101	020105	
5061	025304	046101	020114	042522	
5062	025312	052521	051505	051524	
5063	025320	023040	057040	020103	
5064	025326	047524	051040	051505	
5065	025334	040524	052122	000045	
5066	025342	051445	040514	042526	MSG32: .ASCIZ /%SLAVE NUMBER = /
5067	025350	047040	046525	042502	
5068	025356	020122	020075	000	
5069	025363	045	042504	051516	MSG33: .ASCIZ /%DENSITY = /
5070	025370	052111	020131	020075	
5071	025376	000			
5072	025377	045	040520	044522	MSG34: .ASCIZ /%PARITY = /
5073	025404	054524	036440	000040	
5074	025412	051045	041505	051117	MSG35: .ASCIZ /%RECORD COUNT = /
5075	025420	020104	047503	047125	
5076	025426	020124	020075	000	
5077	025433	045	044103	051101	MSG36: .ASCIZ /%CHAR COUNT = /
5078	025440	041440	052517	052116	
5079	025446	036440	000040		
5080	025452	050045	052101	042524	MSG37: .ASCIZ /%PATTERN NUMBER = /
5081	025460	047122	047040	046525	
5082	025466	042502	020122	020075	
5083	025474	000			

5084	025475	045	044523	043516	MSG38:	.ASCIZ	/XSINGLE PASS = /
5085	025502	042514	050040	051501			
5086	025510	020123	020075	000			
5087	025515	045	051103	020103	MSG39:	.ASCIZ	/XCRC CORRECTION (YES=1,NO=0) = /
5088	025522	047503	051122	041505			
5089	025530	044524	047117	024040			
5090	025536	042531	036523	026061			
5091	025544	047516	030075	020051			
5092	025552	020075	000				
5093	025555	045	042445	052116	MSG40:	.ASCIZ	/XENTER STALLS%READ = /
5094	025562	051105	051440	040524			
5095	025570	046114	022523	042522			
5096	025576	042101	036440	000040			
5097	025604	053445	044522	042524	MSG41:	.ASCIZ	/XWRITE = /
5098	025612	036440	000040				
5099							
5100	025616	052045	051125	020116	MSG42:	.ASCIZ	/XTURN AROUND = /
5101	025624	051101	052517	042116			
5102	025632	036440	000040				
5103	025636	037445	000045		MSG43:	.ASCIZ	/X?X/
5104	025642	042445	052116	051105	MSG44:	.ASCIZ	/XENTER YOZZLE STALL = /
5105	025650	054440	055117	046132			
5106	025656	020105	052123	046101			
5107	025664	020114	020075	000			
5108	025671	045	051105	020122	MSG45:	.ASCIZ	/XERR AMT /
5109	025676	046501	020124	000			
5110	025703	045	041506	000040	MSG46:	.ASCIZ	/XFC /
5111	025710	041445	020101	000	MSG47:	.ASCIZ	/XCA /
5112	025715	045	047041	020117	MSG48:	.ASCIZ	/X!NO BOT ON REWIND: HALT%X/
5113	025722	047502	020124	047117			
5114	025730	051040	053505	047111			
5115	025736	035104	044040	046101			
5116	025744	022524	000045				
5117	025750	047045	052117	040440	MSG49:	.ASCIZ	/XNOT AVAIL /
5118	025756	040526	046111	000040			
5119	025764	044445	046114	043505	MSG50:	.ASCIZ	/XILLEGAL DRIVE TYPE /
5120	025772	046101	042040	044522			
5121	026000	042526	052040	050131			
5122	026006	020105	000				
5123	026011	045	051104	053111	MSG52:	.ASCIZ	/XDRIVE NUMBER = /
5124	026016	020105	052516	041115			
5125	026024	051105	036440	000040			
5126	026032	043045	051117	040515	MSG53:	.ASCIZ	/XFORMAT = /
5127	026040	020124	020075	000			
5128	026045	052	042527	052040	MSG54:	.ASCIZ	/XWE TM/
5129	026052	000115					
5130	026054	051452	020105	046524	MSG55:	.ASCIZ	/XSE TM/
5131	026062	000					
5132	026063	040	046524	000	MSG56:	.ASCIZ	/ TM/
5133	026067	045	047516	026516	MSG57:	.ASCIZ	/XNON-EXIST SLAVE/
5134	026074	054105	051511	020124			
5135	026102	046123	053101	000105			
5136	026110	041445	041522	000040	MSG58:	.ASCIZ	/XCRC /
5137	026116	046045	041522	000040	MSG59:	.ASCIZ	/XLRC /
5138	026124	042052	000040		MSG60:	.ASCIZ	/XD /
5139	026130	050052	000040		MSG61:	.ASCIZ	/XP /

5140	026134	043052	000040		MSG62: .ASCIZ /*F /
5141	026140	025045	051117	043511	MSG64: .ASCIZ /*ORIGINAL ERROR*/
5142	026146	047111	046101	042440	
5143	026154	051122	051117	000052	
5144	026162	051045	052105	054522	MSG65: .ASCIZ /*RETRY: /
5145	026170	020072	000		
5146	026173	052	051441	020105	MSG66: .ASCIZ /*!SE RTRY /
5147	026200	052122	054522	000040	
5148	026206	020452	051105	051501	MSG67: .ASCIZ /*!ERASE/
5149	026214	000105			
5150	026216	051045	051105	053105	MSG68: .ASCIZ /*REREV: /
5151	026224	020072	000		
5152	026227	045	040524	042520	MSG69: .ASCIZ /*TAPE MARK = /
5153	026234	046440	051101	020113	
5154	026242	020075	000		
5155	026245	045	047041	020117	MSG70: .ASCIZ /*!NO DRY FROM REWIND: HALT*/
5156	026252	051104	020131	051106	
5157	026260	046517	051040	053505	
5158	026266	047111	035104	044040	
5159	026274	046101	022524	000	
5160	026301	045	047516	026516	MSG71: .ASCIZ /*NON-EXIST DRIVE/
5161	026306	054105	051511	020124	
5162	026314	051104	053111	000105	
5163	026322	051045	043105	042127	MSG72: .ASCIZ /*REFWD: /
5164	026330	020072	000		
5165	026333	045	052127	051105	MSG73: .ASCIZ /*WTERR: /
5166	026340	035122	000040		
5167	026344	051045	043505	051511	MSG74: .ASCIZ /*REGISTER START = /
5168	026352	042524	020122	052123	
5169	026360	051101	020124	020075	
5170	026366	000			
5171	026367	045	042526	052103	MSG75: .ASCIZ /*VECTOR = /
5172	026374	051117	036440	000040	
5173	026402	042045	051105	053105	MSG76: .ASCIZ /*DEREV: /
5174	026410	020072	000		
5175	026413	045	042504	053506	MSG77: .ASCIZ /*DEFWD: /
5176	026420	035104	000040		
5177	026424	020445	047516	026516	MSG78: .ASCIZ /*!NON-RETRYABLE WRITE ERROR: ER /
5178	026432	042522	051124	040531	
5179	026440	046102	020105	051127	
5180	026446	052111	020105	051105	
5181	026454	047522	035122	042440	
5182	026462	020122	000		
5183	026465	045	047041	047117	MSG79: .ASCIZ /*!NON-RETRYABLE READ ERROR: ER /
5184	026472	051055	052105	054522	
5185	026500	041101	042514	051040	
5186	026506	040505	020104	051105	
5187	026514	047522	035122	042440	
5188	026522	020122	000		
5189	026525	045	020441	047105	MSG100: .ASCIZ /*!!END OF PASS */
5190	026532	020104	043117	050040	
5191	026540	051501	020123	000045	
5192	026546	022445	025052	025052	MSG101: .ASCIZ /******
5193	026554	025052	025052	025052	
5194	026562	025052	025052	025052	
5195	026570	025052	025052	000	

5196	026575	052	046524	031460	MSG102: .ASCIZ /*TM03 /
5197	026602	000040			
5198	026604	051452	040514	042526	MSG103: .ASCIZ /*SLAVES /
5199	026612	020123	000		
5200	026615	045	052501	047524	MSG104: .ASCIZ /*AUTO CONT: /
5201	026622	041440	047117	035124	
5202	026630	000040			
5203	026632	051045	041505	053117	MSG105: .ASCIZ /*RECOVERED/
5204	026640	051105	042105	000	
5205	026645	052	020441	040502	MSG106: .ASCIZ /*!!BAD TAPE OVERFLOW/
5206	026652	020104	040524	042520	
5207	026660	047440	042526	043122	
5208	026666	047514	000127		
5209	026672	051045	053505	047111	MSG16A: .ASCIZ /*REWIND TAPE; RESTART AT BLOCK 1/
5210	026700	020104	040524	042520	
5211	026706	020073	042522	052123	
5212	026714	051101	020124	052101	
5213	026722	041040	047514	045503	
5214	026730	030440	000		
5215	026733	045	020441	047125	MSG107: .ASCIZ /*!!UNRECOVERABLE BAD SPOT/
5216	026740	042522	047503	042526	
5217	026746	040522	046102	020105	
5218	026754	040502	020104	050123	
5219	026762	052117	000		
5220	026765	045	040502	020104	.ASCIZ /*BAD RECORD LEFT ON TAPE%/
5221	026772	042522	047503	042122	
5222	027000	046040	043105	020124	
5223	027006	047117	052040	050101	
5224	027014	022505	000		
5225	027017	052	020441	047520	MSG109: .ASCIZ /*!!POSITION LOST IN RETRY/
5226	027024	044523	044524	047117	
5227	027032	046040	051517	020124	
5228	027040	047111	051040	052105	
5229	027046	054522	000		
5230	027051	045	052523	050123	MSG110: .ASCIZ /*SUSPECT BAD TAPE/
5231	027056	041505	020124	040502	
5232	027064	020104	040524	042520	
5233	027072	000			
5234	027073	045	042522	042520	MSG111: .ASCIZ /*REPEAT: /
5235	027100	052101	020072	000	
5236	027105	040	040502	020104	MSG112: .ASCIZ / BAD TAPE SPOTS%/
5237	027112	040524	042520	051440	
5238	027120	047520	051524	000045	
5239					
5240	027126	020045	047523	052106	MSG113: .ASCIZ /* SOFT: /
5241	027134	020072	000		
5242					
5243	027137	045	044040	051101	MSG114: .ASCIZ /* HARD: /
5244	027144	035104	000040		
5245					
5246	027150	020445	044041	051101	MSG115: .ASCIZ /*!!HARD READ ERROR/
5247	027156	020104	042522	042101	
5248	027164	042440	051122	051117	
5249	027172	000			
5250	027173	045	052441	044516	MSG116: .ASCIZ /*!UNIT IS REWINDING: TEST WILL START AT BOT/
5251	027200	020124	051511	051040	

```

5252 027206 053505 047111 044504
5253 027214 043516 020072 042524
5254 027222 052123 053440 046111
5255 027230 020114 052123 051101
5256 027236 020124 052101 041040
5257 027244 052117 000
5258 027247 134 000
5259 027251 045 042522 047515
5260 027256 042526 052040 042115
5261 027264 020120 051106 046517
5262 027272 051440 040514 042526
5263 027300 052040 020117 042502
5264 027306 052040 051505 042524
5265 027314 022504 000
5266 027317 045 040510 042122
5267 027324 040527 042522 051440
5268 027332 051127 044440 020116
5269 027340 051525 022505 000
5270
5271 027346
5272 027346 000000
5273
5274 033354
5275 033354 000000
5276
5277 000001

```

```

MSG118: .ASCIZ / /
MSG120: .ASCIZ /%REMOVE TMDP FROM SLAVE TO BE TESTED%/

```

```

MSG121: .ASCIZ /%HARDWARE SWR IN USE%/

```

```

.WDATA: 0 .EVEN ;WRITE BUFFER
.RDATA: 0 .+.4004 ;READ BUFFER
.END

```

ABLCNT	000742	BTPT	000732	DATOB	014334	DFOD	015476	ER	000524
ACTLRC	021030	BTSTF	000730	DATOC	014400	DFOE	015470	ERCHK	017244
ADRVN	000740	BTUR	007410	DATOD	014406	DFOF	015462	ERPT	020064
AMOD1	022306	BT00	001610	DATOE	014416	DF1	015710	ERPTG	020120
AMOD2	022446	BT01	001714	DATOF	014432	DF2	015720	ERPTG1	020166
APATS	014746	BT02	002020	DAT1	014442	DF3	015736	ERPTT	020104
AS	000526	BT03	002124	DAT1A	014446	DF4	016000	ERPTO	020202
ASEQ	021752	BT04	002230	DAT10	014564	DOUT	024234	ERPT1	020250
ASEQCF	000744	BT05	002334	DAT11	014614	DOUTD	024310	ERPT2	020306
ASEQF	000736	BT06	002440	DAT12	014634	DPC	016632	ERPT3	020344
ASEQX	022110	BT07	002544	DAT13	014656	DPCG	016640	ERPT4	020406
ASEQ0	022004	B0	012034	DAT14	014666	DPC0	016646	ERPT5	020430
ASEQ1	022010	B1	012100	DAT15	014716	DPCOA	016710	ERPT5A	020512
ASEQ2	022050	B2	012110	DAT2	014462	DPC1	016716	ERPT6	020554
ASEQ3	022062	CADER	021012	DAT3	014466	DPC1A	016744	ERPT7	020624
ASEQ4	022072	CC	000530	DAT3A	014474	DPC2	017006	ERPX	020660
BA	000514	CCNTR	012134	DAT4	014512	DPC2A	016750	ERPX0	020626
BAER	021014	CHNFLG	003040	DAT5	014522	DPC2B	016772	ERPX1	020746
BBC	000660	CLLAST	015130	DAT6	014530	DPC3	017040	ERPX2	021010
BCNT	000712	CLP	015240	DAT7	014536	DPPRT	017106	ERSAV	000724
BDPP	000720	CLPE	015264	DB	000532	DPPRTX	017242	ERTFL	000734
BD00	001410	CLP2	015322	DCHK	015374	DPPRT0	017156	ERO	017260
BD10	001430	CLP3	015334	DCHK0	015422	DPPRT1	017202	EROA	017326
BD20	001450	CLO	015050	DEREV1	001170	DPPRT2	017216	EROB	017274
BD30	001470	CL1	015076	DEREX	016456	DROP	016622	ER1	017334
BD40	001510	CL2	015120	DEREX1	016510	DRPK	016610	ER10	020026
BD50	001530	CL3	015202	DERFL	000706	DRPKF	016522	ER2	017340
BD60	001550	CONER	021016	DERR	016006	DRP1	001010	ER2A	017404
BD70	001570	CRCC	000566	DERR0	016016	DRP2	001012	ER2A0	017354
BKRT	012054	CRCER	021026	DERR0A	016046	DRP3	001014	ER2A1	017374
BKSP	011700	CRCLRC	015032	DERR0B	016100	DRP4	001016	ER2B	017420
BKTM	011762	CRCSV	021034	DERR0C	016124	DRP5	001020	ER2C	017444
BKTM0	012024	CS	000520	DERR0D	016126	DRP6	001022	ER2D	017460
BLCNTR	000656	C1	000510	DERR1	016154	DRP7	001024	ER2E	017506
BPKP	000722	DATA0	002772	DERR2	016156	DRP8	001026	ER3	017514
BP00	001210	DATA1	002774	DERR3	016172	DRVER	021020	ER3A	017552
BP10	001230	DATA10	003012	DERR4	016174	DS	000522	ER3A1	017610
BP20	001250	DATA11	003014	DERR4A	016334	DSUP	014116	ER3B	017616
BP30	001270	DATA12	003016	DERR4B	016402	DS0	014124	ER4	017622
BP40	001310	DATA13	003020	DERR5	016440	DS0A	014206	ER4A	017666
BP50	001330	DATA14	003022	DERR6	016452	DS0B	014204	ER4A1	017660
BP60	001350	DATA15	003024	DFX	016004	DS0C	014146	ER6	017672
BP70	001370	DATA2	002776	DF0	015676	DS2A	014234	ER6A	017756
BTADDR	001030	DATA3	003000	DF0A	015572	DS3	014240	ER7	020010
BTFLG	000726	DATA4	003002	DF0A0	015614	DS4	014252	EXCRC	015370
BT0V	007172	DATA5	003004	DF0A1	015630	DT	000536	EXLRC	015372
BT0VX	007332	DATA6	003006	DF0A2	015644	DVN	000550	FC	000516
BT0V0	007212	DATA7	003010	DF0A3	015660	DOFL	014440	FCER	021022
BT0V1	007222	DATBL	002770	DF0A4	015664	EMADDR	000654	FCSAV	000634
BT0V2	007306	DATER1	001130	DF0B	015532	ENDFLG	000736	FMCNT	000556
BT0V3	007324	DATR	014766	DF0B0	015554	ENDTBL	002770	FRPRT	021036
BTPRT	007334	DAT0	014302	DF0C	015512	EOTC0	002650	GTSWR	023216
BTPRT1	007404	DAT0A	014326	DF0C0	015522	EOTREC	000662	HDRFL	000666

HERE	005032	MSG23M	025042	MSG73	026333	RANSET	004276	RD6	010522
HRDS	022122	MSG24	025047	MSG74	026344	RCNT	000554	RD7	010546
INIT	005426	MSG25	025067	MSG75	026367	RCNTR	012174	RD7A	010576
INTRF	000570	MSG26	025107	MSG76	026402	RCSAV	000632	READ	007736
LRCER	021024	MSG27	025120	MSG77	026413	RDA	007776	REGS	000544
LRCV	021032	MSG28	025131	MSG78	026424	RDATA	033354	REOT	004330
MR	000534	MSG3	024470	MSG79	026465	RDCMD	000562	REOTC	005054
MSG1	024456	MSG30	025133	MSG8	024521	RDERR1	001150	REOTX	004760
MSG10	024537	MSG31	025200	MSG9	024531	RDER1	001110	REOTXX	005050
MSG100	026525	MSG31A	025255	MTC1	000674	RDER2	001112	REOT1A	004376
MSG101	026546	MSG32	025342	MTINT	021734	RDER3	001114	REOT1B	004424
MSG102	026575	MSG33	025363	MTINTA	021736	RDER4	001116	REOT1C	004444
MSG103	026604	MSG34	025377	NOP =	000240	RDER5	001120	REOT1E	004470
MSG104	026615	MSG35	025412	NRTP	011244	RDER6	001122	REOT1F	004440
MSG105	026632	MSG36	025433	NRZOF	000652	RDER7	001124	REOT2	004512
MSG106	026645	MSG37	025452	OCTP	024020	RDER8	001126	REOT2A	004550
MSG107	026733	MSG38	025475	OCTPG	024170	RDEX	010664	REOT3	004576
MSG109	027017	MSG39	025515	OCTPG0	024206	RDFL	015030	REOT4	004626
MSG11	024544	MSG4	024475	OCTPG1	024212	RDRTG	010760	REOT5	004642
MSG110	027051	MSG40	025555	OCTP0	024036	RDRTX	011242	REOT6	004706
MSG111	027073	MSG41	025604	OCTP1	024056	RDRTY	010672	REOT7	004736
MSG112	027105	MSG42	025616	OCTP2	024064	RDRT0	010704	RETRY	000604
MSG113	027126	MSG43	025636	OCTP3	024154	RDRT1	010736	RFHARD	002730
MSG114	027137	MSG44	025642	OFL	024232	RDRT1A	010734	RFSOFT	002670
MSG115	027150	MSG45	025671	PAPRT	022602	RDRT1B	010754	RPCNT	000702
MSG116	027173	MSG46	025703	PAPRTA	022756	RDRT2	011036	RRHARD	002750
MSG118	027247	MSG47	025710	PAPRTB	022772	RDRT3	011060	RRSOFT	002710
MSG12	024560	MSG48	025715	PAPRTC	023006	RDRT4	011064	RSEQ	007424
MSG120	027251	MSG49	025750	PAPRTD	023014	RDRT5	011066	RSEX	007560
MSG121	027317	MSG5	024502	PAPRTY	023110	RDRT5A	011132	RSF	007476
MSG13	024576	MSG50	025764	PAPRT0	022704	RDRT5B	011142	RSFR	007574
MSG14	024604	MSG52	026011	PAPRT1	023060	RDRT6	011172	RSFRX	007734
MSG15	024611	MSG53	026032	PAPRT2	023112	RDRT7	011236	RSFR0	007626
MSG16	024642	MSG54	026045	PAPRT3	023114	RDX	010670	RSFR1	007640
MSG16A	026672	MSG55	026054	PARS	014300	RD0	010034	RSFR2	007672
MSG17	024645	MSG56	026063	PATRN	000560	RD1	010102	RSF0	007532
MSG2	024463	MSG57	026067	PATS	014276	RD1A	010116	RSF1	007546
MSG20	024650	MSG58	026110	PFLG	000672	RD1B	010124	RSR	007454
MSG21	024663	MSG59	026116	PICK	017042	RD1D	010132	RSTAL	000574
MSG22	024710	MSG6	024507	PIK1	000770	RD10	010610	RTCNT	000704
MSG23	024737	MSG60	026124	PIK2	000772	RD11	010620	RTRN	000664
MSG23A	024745	MSG61	026130	PIK3	000774	RD2	010136	RTYFL	000714
MSG23B	024752	MSG62	026134	PIK4	000776	RD3	010176	RTY1	001050
MSG23C	024757	MSG64	026140	PIK5	001000	RD4	010226	RTY2	001052
MSG23D	024764	MSG65	026162	PIK6	001002	RD4A	010324	RTY3	001054
MSG23E	024772	MSG66	026173	PIK7	001004	RD4A0	010356	RTY4	001056
MSG23F	024777	MSG67	026206	PIK8	001006	RD4A1	010400	RTY5	001060
MSG23G	025004	MSG68	026216	PRB	000624	RD4A2	010422	RTY6	001062
MSG23H	025011	MSG69	026227	PRS	000622	RD4B	010430	RTY7	001064
MSG23I	025016	MSG7	024514	PSW	000606	RD4C	010434	RTY8	001066
MSG23J	025023	MSG70	026245	RANBAS	000626	RD4D	010464	RWND	005056
MSG23K	025030	MSG71	026301	RANG	023164	RD4E	010470	RWDA	005070
MSG23L	025035	MSG72	026322	RANSAV	000630	RD5	010500	RWDX	005376

RWND0	005126	STTBL	001050	TINP2C	013272	WRTE	005550	W2	005656
RWND1A	005236	SWR	000610	TINP3	013312	WRTSB	006776	W3	005672
RWND2	005254	SWREG	000176	TINP3A	013642	WRTSB0	007050	W3A	005714
RWND3	005260	TAPG	021072	TINP4	013706	WRTSB1	007064	W4	006000
RWND5	005336	TAPG0	021104	TKB	000614	WRTSB2	007074	W4A	006026
RWND6	005354	TAPG1	021146	TKS	000612	WRTSB3	007150	W5	006042
SCVFL	014070	TAPG2	021160	TMEX	000564	WRTY	006374	W6	006052
SERFL	000710	TAPG3	021170	TMFLG	000700	WRTYR	006426	XOR	015342
SN	000540	TAPG3A	021240	TOB	000640	WRTYTM	006422	XORS	015366
SNPG	024412	TAPG3B	021254	TOG	023756	WRTY0	006402	YOZ	011260
SNPT	024334	TAPG3C	021274	TPB	000620	WRTY1	006502	YOZA	011320
SPFLG	000572	TAPG3D	021340	TPOS	014072	WRTY2	006504	YOZB	011326
STAL	000670	TAPG3E	021364	TPOS1	014102	WRTY2A	006550	YOZC	011346
STALL	012124	TAPG3F	021222	TPS	000616	WRTY2B	006566	YOZCO	011434
START	003026	TAPG4	021412	TSTAL	000600	WRTY3	006632	YOZC1	011450
STARTA	003160	TAPG5	021426	TTIN	023604	WRTY3A	006650	YOZC2	011456
STARTB	003164	TAPG6	021470	TTINT	021504	WRTY4	006736	YOZD	011476
STARTC	003152	TAPG7	021500	TTOUT	023646	WRTY5	006772	YOZD0	011576
STARTD	003244	TBELL	023774	TTR	023352	WRW	006350	YOZD1	011622
STARTE	003236	TC	000542	TYPE =	000004	WRWX	006372	YOZD2	011570
START1	003406	TEMP1	000644	TYPOCT=	104400	WSTAL	000576	YOZD3	011614
START2	003524	TEMP2	000646	UDES	000552	WTER1	001070	YOZD4	011550
START3	003540	TEMP3	000650	UNP	000676	WTER2	001072	YOZE	011626
START4	003554	TEND	005004	UNX	000766	WTER3	001074	YOZF	011656
START5	004216	TIB	000642	UN1	000746	WTER4	001076	YOZG	011662
START6	004234	TINER	023566	UN2	000750	WTER5	001100	YOZH	011676
START7	004264	TINF	000636	UN3	000752	WTER6	001102	YOZO	011272
START8	004272	TINP	012220	UN4	000754	WTER7	001104	YSTAL	000602
STAR1A	003420	TINPX	014062	UN5	000756	WTER8	001106	SENDAD	005022
STAR1B	003440	TINPO	012606	UN6	000760	WTM	006064	SMNEW	024446
STAR1C	003474	TINPOB	012676	UN7	000762	WTM0	006126	SMSWR	024436
STAR4A	003634	TINPOC	012736	UN8	000764	WTM1	006170	SSVPC =	000040
STAR40	003566	TINPOD	012756	UPS	000716	WTM2	006200	.	= 033356
STAUT	003136	TINPOE	013014	VECT	000546	WTM3	006210	.RESTO	023330
STAUTO	003346	TINP1	013032	WC	000512	WTM4	006270	.SAVE	023306
STFLG	000640	TINP2	013110	WDATA	027346	WTM4A	006326		
STP	003734	TINP2A	013166	WEX	006332	W0	005554		
STPX	004202	TINP2B	013244	WRITE	005534	W1	005616		

. ABS. 033356 000

ERRORS DETECTED: 0

.CZTURA.SEQ/SOL CZTURA.P11
RUN-TIME: 60 108 6 SECONDS
RUN-TIME RATIO: 894/175=5.0
CORE USED: 14K (28 PAGES)