```
                        .REM    *
```

IDENTIFICATION

PRODUCT CODE: AC-A800E-MC

PRODUCT NAME: C2TEDE0 TM03-TE16/TU77 DATA RELIBILITY PROGRAM

DATE CREATED: 20-FEBRUARY-1984

MAINTAINER: TAPE DIAGNOSTIC GROUP

AUTHOR: J. HITT

TABLE OF CONTENTS

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

                        SW5:            SWITCH FIVE (5) WHEN SET DURING A READ
                                        FORWARD OR REVERSE WILL CAUSE THE TAPE
                                        TO CONTINUOUSLY READ THE CURRENT RECORD
                                        BY SPACING EITHER FORWARD OR REVERSE AND
                                        REREADING THAT RECORD.  THIS TAPE MOVEMENT
                                        IS CALLED YOZZLING.  THERE IS A SOFTWARE
                                        DELAY EXECUTED BETWEEN EACH SPACE/READ
                                        OF THE RECORD AND IT MAY BE VARIED BY
                                        TYPING CONTROL C ON THE TELETYPE DURING
                                        THE EXECUTION OF THE YOZZLE AND RESPONDING
                                        TO THE PRINTED REQUEST WITH A SIX (6)
                                        DIGIT VALUE.  THE YOZZLE STALL IS PRESET
                                        TO A VALUE OF 3000 IN THE PROGRAM TO
                                        PREVENT EXCESSIVE TAPE WEAR, BUT MAY BE
                                        SET TO ANY VALUE THROUGH THE TELETYPE.

                        SW6-8:          THESE THREE (3) SWITCHES CONTROL THE RANDOMIZATION
                                        OF DATA AND BLOCK SIZE AND MAY BE SET
                                        AND RESET AT ANY TIME.  THE ACTUAL CHANGE
                                        WILL TAKE PLACE BETWEEN BLOCK CYCLES.

                        SW9:            SWITCH NINE (9) WHEN SET WILL CAUSE ALL
                                        AVAILABLE TAPE UNITS TO BE REWOUND AT
                                        THE END OF THE CURRENT BLOCK CYCLE.  TESTING
                                        WILL BE RESUMED AT A BLOCK COUNT OF
                                        ONE (1) WHEN ALL UNITS HAVE REACHED BOT.
```

```
744
745
746                                     SW10-13:        THESE SWITCHES ARE USED TO CONTROL THE
747                                                     ERROR HANDLING TO BE DONE ON THE TAPE
748                                                     OPERATION DESCRIBED BY SWITCHES 0-3.
749
750                                             A.      SWITCH TEN (10) WHEN SET TO A ONE
751                                                     WILL DISALLOW ANY ERROR PRINTOUTS MADE
752                                                     ON THE OPERATION IN PROGRESS.  CATASTROPHIC
753                                                     FAILURES AND INFORMATION PRINTOUTS WILL
754                                                     STILL OCCUR.  IE: UNIT NOT AVAILABLE, ILLEGAL
755                                                     BOT, DROP OR PICK OVERFLOW, AND EOT REWIND.
756
757                                             B.      SWITCH ELEVEN (11) WHEN SET TO A ONE
758                                                     WILL DISALLOW THE CHECKING FOR STATUS
759                                                     ERRORS ON READ (FORWARD OR REVERSE) OPERATIONS.
760
761                                             C.      SWITCH TWELVE (12) WHEN SET TO A ONE
762                                                     WILL DISALLOW THE CHECKING FOR STATUS
763                                                     ERRORS ON WRITE OPERATIONS.
764
765                                             D.      SWITCH THIRTEEN (13) WHEN SET TO A ONE
766                                                     WILL DISALLOW THE CHECKING OF READ
767                                                     DATA.  THIS SWITCH HAS NO EFFECT ON
768                                                     STATUS CHECKING.
769
770                                             **NOTE THAT WHEN SW11 OR 12 ARE SET, NOT ONLY ARE ERRORS NOT CHECKED, BU
771                                             ***THEREFOR USE CAUTION TO ASSURE THAT OPERATIONS ARE NOT UNEXECUTED DUE
772                                             ****DO NOT SET SW 11 OR 12 TO A ONE (1), DURING A RETRY SEQUENCE.
773
774                                     SW14:           SWITCH FOURTEEN (14)WHEN SET TO A ONE (1) WILL
775                                                     PRINT THE ACCUMULATED READ/WRITE STATISTICS FOR THE SELECTED
776                                                     SLAVE UNDER TEST AT THE END OF THE CURRENT BLOCK
777                                                     CYCLE.  THE STATISTICS PRINTED ARE THE NUMBER OF BITS
778                                                     DROPPED OR PICKED, THE NUMBER OF RETRIES, WRITE ERRORS,
779                                                     READ ERRORS, AND DATA ERRORS.
780
781                                     SW15:           SWITCH FIFTEEN (15) WHEN SET TO A ONE,
782                                                     WILL CAUSE THE PROGRAM TO HALT ON ANY
783                                                     ERROR DETECTED BY THE OPERATION IN PROGRESS.
784                                                     IF BOTH SWITCH TEN (10) AND FIFTEEN (15)
785                                                     ARE SET, THE ACTUAL ERROR DETECTED WILL
786                                                     NOT BE PRINTED BUT WILL CAUSE A HALT.
787                                                     IF SWITCH TEN (10) IS RESET BEFORE PRESSING
788                                                     CONTINUE, THE ERROR WHICH CAUSED THE HALT
789                                                     WILL BE PRINTED BEFORE TESTING IS RESUMED.
790
```

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 APPROPIATE 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 UNITLL 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-RETRTABLE: IF ANY NON RETRTABLE 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/TE16 CONTROL REGISTER
WC = RH WORD COUNT
BA = RH BUS ADDRESS
FC = TE16 FRAME COUNT
CS2 = RH CONTROLLER STATUS
DS = TE16 DRIVE STATUS
ER = TE16 ERROR REGISTER
AS = ATTENTION SUMMARY
CK = TE16 CHECK CHARACTER
DB = RH DATA BUFFER
MR = TE16 MAINTENENCE REGISTER
DT = TE16 DRIVE TYPE
SN = TE16 SERIAL NUMBER
TC = TE16 TEST CONTROL
*F = DATA FORMAT
*P = PARITY
*D = DENSITY
*PATRN = DATA PATTERN NUMBER (R = RANDOM)

```
933
934
935                                     EXAMPLE 1: IN THIS EXAMPLE SLAVE 1 ON TM03 0 WAS
936                                             OPERATING AT 1600 BPI IN ODD PARITY USING
937                                             THE NINE CHANNEL NORMAL DATA FORMAT.  A
938                                             WRITE STATUS ERROR WAS DETECTED.  THE BAD
939                                             STATUS INDICATES THAT AN UNCORRECTABLE DATA
940                                             ERROR (BIT 6 OF ER) AND A PE FORMAT ERROR
941                                             (BIT 7 OF ER) OCCURED DURING THE WRITE
942                                             OPERATION OF THE SIXTH (6) RECORD OF THE
943                                             FIFTY (50) RECORDS IN BLOCK (2).  THE
944                                             SIZE OF THE RECORD WAS TWO HUNDRED (200)
945                                             FRAMES.  THE CHECK CHARACTER REFLECTS THE BAD TRACK.
946
947                                             DRIVE NO. 0 *SLAVE NO. 1 *D 4 *P 0 *F 14 *PATRN 1
948                                             *BN 2 *RN 6-50 *RS = 200 *WE
949                                             CS1 144260
950                                             CS2 100
951                                             DS 150640
952                                             ER 300
953                                             WC 0
954                                             CK 4
955
956                                     EXAMPLE 2: IN THIS EXAMPLE SLAVE 3 ON TM03 1 WAS
957                                             OPERATING AT 800 BPI IN EVEN PARITY USING
958                                             THE NINE CHANNEL NORMAL DATA FORMAT.  A
959                                             READ STATUS ERROR WAS DETECTED DURING
960                                             THE REVERSE READ OF THE TENTH (10)
961                                             RECORD OF THE 25 RECORDS IN THIS BLOCK (12).
962                                             THE SIZE OF THE RECORD IS TWENTY (20) FRAMES.
963                                             THE PRINTOUT INDICATES THE DETECTION OF A
964                                             VERTICAL PARITY ERROR (VPE: BIT 6 OF ER) AND A
965                                             CYCLIC REDUNDENCY ERROR (CRC: BIT 15 OF ER).
966                                             THE CRC CHARACTER, AS RECEIVED, IS NOT
967                                             AS EXPECTED AND IS PRINTED SHOWING BOTH
968                                             THE ACTUAL (FIRST) AND THE EXPECTED (LAST).
969
970                                             DRIVE NO. 2 *SLAVE NO. 3 *D 3 *P 1 *F 14 *PATRN 3
971                                             *BN 12 *RN 10 25 *RS 20 *RE R
972                                             CS1 144276
973                                             CS2 100
974                                             DS 150600
975                                             ER 100100
976                                             WC 0
977                                             CRC 767-777
978
```

```
                        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
```

```
1073
1074
1075                    EXAMPLE 7: THIS EXAMPLE REFLECTS AN ERROR DETECTED
1076                            WHILE WRITING A TAPE MARK (TM) AT THE END
1077                            OF THE CURRENT DATA BLOCK PER OPTION RESPONSE
1078                            TM=1.  NOTE THAT THE TM RECORD NUMBER
1079                            IS ONE GREATER THAN THE TOTAL NUMBER OF DATA
1080                            RECORDS IN THE CURRENT BLOCK.
1081
1082                            DRIVE NO. 1 *SLAVE NO. 1 *D 2 *P 0 *F 14
1083                            *BN 67 *RN 101-100 *RS 36 *WE TM
1084                            CS1 144226
1085                            CS2 300
1086                            DS 150604
1087                            ER 1000
1088                            WC 0
1089
1090                    EXAMPLE 8: THIS EXAMPLE SHOWS TWO (2) PRINTOUTS
1091                            REFLECTING A WRITE RETRY WHICH WAS NOT
1092                            SUCCESSFUL THE FIRST TIME, BUT WHICH
1093                            DID RECOVER ON THE SECOND.
1094                            THE UNSUCCESSFUL RETRY IS LOGGED AS
1095                            A SUSPECTED BAD TAPE SPOT BY ITS
1096                            BLOCK AND RECORD NUMBER.
1097
1098
1099                            DRIVE NO. 0 *SLAVE NO. 2 *D 4 *P 0 *F 14 *PATRN 6
1100                            *BN 2 *RN 12-20 *RS 667 *WE
1101                            CS1 144260
1102                            CS2 100
1103                            DS 150640
1104                            ER 100
1105                            WC 0
1106                            ***ORIGINAL ERROR***
1107
1108                            DRIVE NO. 0 SLAVE NO. 2 *D 4 *P 0 *F 14 *PATRN 6
1109                            *BN 2 *RN 12-20 *RS 667 *WE
1110                            CS1 144260
1111                            CS2 100
1112                            DS 150640
1113                            ER 100
1114                            WC 0
1115                            SUSPECT BAD TAPE
1116                            RETRY: 0
1117                            REPT: 0
1118                            RECOVERED
1119                            RETRY: 1
1120
```

```
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

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

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 UNSUCESSFUL, A SKIP ERASE IS
                                DONE, A SUPECTED BAD TAPE SPOT IS LOGGED
                                AT THIS BLOCK AND RECORD NUMBER, AND A SECOND
                                RETRY OF FOUR REPEATS IS DONE. IF AFTER FOUR
                                (4) RETRIES, THE RECORD CANNOT BE RECOVERED
                                A NOTIFICATION IS PRINTED, AND TESTING IS
                                RESUMED ON THE NEXT RECORD.
                                IF 20(8) BAD TAPE SPOTS ARE FOUND, THE
                                SLAVE WILL BE REWOUND AND REMOVED FROM
                                TESTING WITH AN APPROPRIATE MESSAGE PRINTED.
                                THE READ RETRY SCHEME CONSISTS OF REREADING THE
                                RECORD UP TO EIGHT TIMES. IF ALL EIGHT REREADS ARE
                                BAD, IT IS A HARD ERROR. IF ANY REREAD IS
                                SUCCESSFUL, THIS IS A SOFT ERROR.
                                IF THE ORIGINAL ERROR IS OF THE NON-RETRYABLE
                                TYPE (IE; ILF,RMR,ILR,NEF,CBUSPE), THE RETRY SCHEME
                                IS NOT ENTERED AND A MESSAGE IS PRINTED.

                        SW5:    SWITCH FIVE (5) WHEN SET DURING A READ
                                FORWARD OR REVERSE WILL CAUSE THE TAPE
                                TO CONTINUOUSLY READ THE CURRENT RECORD
                                BY SPACING EITHER FORWARD OR REVERSE AND
                                REREADING THAT RECORD.  THIS TAPE MOVEMENT
                                IS CALLED YOZZLING.  THERE IS A SOFTWARE
                                DELAY EXECUTED BETWEEN EACH SPACE/READ
                                OF THE RECORD AND IT MAY BE VARIED BY
                                TYPING CONTROL C ON THE TELETYPE DURING
                                THE EXECUTION OF THE YOZZLE AND RESPONDING
                                TO THE PRINTED REQUEST WITH A SIX (6)
                                DIGIT VALUE.  THE YOZZLE STALL IS PRESET
                                TO A VALUE OF 3000 IN THE PROGRAM TO
                                PREVENT EXCESSIVE TAPE WEAR, BUT MAY BE
                                SET TO ANY VALUE THROUGH THE TELETYPE.

                        SW6-8:  THESE THREE (3) SWITCHES CONTROL THE RANDOMIZATION
                                OF DATA AND BLOCK SIZE AND MAY BE SET
                                AND RESET AT ANY TIME.  THE ACTUAL CHANGE
                                WILL TAKE PLACE BETWEEN BLOCK CYCLES.

                        SW9:    SWITCH NINE (9) WHEN SET WILL CAUSE ALL
                                AVAILABLE TAPE UNITS TO BE REWOUND AT
                                THE END OF THE CURRENT BLOCK CYCLE.  TESTING
                                WILL BE RESUMED AT A BLOCK COUNT OF
                                ONE (1) WHEN ALL UNITS HAVE REACHED BOT.
```

```
744
745
746                                     SW10-13:        THESE SWITCHES ARE USED TO CONTROL THE
747                                                     ERROR HANDLING TO BE DONE ON THE TAPE
748                                                     OPERATION DESCRIBED BY SWITCHES 0-3.
749
750                                             A.      SWITCH TEN (10) WHEN SET TO A ONE
751                                                     WILL DISALLOW ANY ERROR PRINTOUTS MADE
752                                                     ON THE OPERATION IN PROGRESS.  CATASTROPHIC
753                                                     FAILURES AND INFORMATION PRINTOUTS WILL
754                                                     STILL OCCUR.  IE: UNIT NOT AVAILABLE, ILLEGAL
755                                                     BOT, DROP OR PICK OVERFLOW, AND EOT REWIND.
756
757                                             B.      SWITCH ELEVEN (11) WHEN SET TO A ONE
758                                                     WILL DISALLOW THE CHECKING FOR STATUS
759                                                     ERRORS ON READ (FORWARD OR REVERSE) OPERATIONS.
760
761                                             C.      SWITCH TWELVE (12) WHEN SET TO A ONE
762                                                     WILL DISALLOW THE CHECKING FOR STATUS
763                                                     ERRORS ON WRITE OPERATIONS.
764
765                                             D.      SWITCH THIRTEEN (13) WHEN SET TO A ONE
766                                                     WILL DISALLOW THE CHECKING OF READ
767                                                     DATA.  THIS SWITCH HAS NO EFFECT ON
768                                                     STATUS CHECKING.
769
770                                             **NOTE THAT WHEN SW11 OR 12 ARE SET, NOT ONLY ARE ERRORS NOT CHECKED, BU
771                                             ***THEREFOR USE CAUTION TO ASSURE THAT OPERATIONS ARE NOT UNEXECUTED DUE
772                                             ****DO NOT SET SW 11 OR 12 TO A ONE (1), DURING A RETRY SEQUENCE.
773
774                                     SW14:           SWITCH FOURTEEN (14)WHEN SET TO A ONE (1) WILL
775                                                     PRINT THE ACCUMULATED READ/WRITE STATISTICS FOR THE SELECTED
776                                                     SLAVE UNDER TEST AT THE END OF THE CURRENT BLOCK
777                                                     CYCLE.  THE STATISTICS PRINTED ARE THE NUMBER OF BITS
778                                                     DROPPED OR PICKED, THE NUMBER OF RETRIES, WRITE ERRORS,
779                                                     READ ERRORS, AND DATA ERRORS.
780
781                                     SW15:           SWITCH FIFTEEN (15) WHEN SET TO A ONE,
782                                                     WILL CAUSE THE PROGRAM TO HALT ON ANY
783                                                     ERROR DETECTED BY THE OPERATION IN PROGRESS.
784                                                     IF BOTH SWITCH TEN (10) AND FIFTEEN (15)
785                                                     ARE SET, THE ACTUAL ERROR DETECTED WILL
786                                                     NOT BE PRINTED BUT WILL CAUSE A HALT.
787                                                     IF SWITCH TEN (10) IS RESET BEFORE PRESSING
788                                                     CONTINUE, THE ERROR WHICH CAUSED THE HALT
789                                                     WILL BE PRINTED BEFORE TESTING IS RESUMED.
790
```

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 APPROPIATE 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 UNITLL 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/TE16 CONTROL REGISTER
WC = RH WORD COUNT
BA = RH BUS ADDRESS
FC = TE16 FRAME COUNT
CS2 = RH CONTROLLER STATUS
DS = TE16 DRIVE STATUS
ER = TE16 ERROR REGISTER
AS = ATTENTION SUMMARY
CK = TE16 CHECK CHARACTER
DB = RH DATA BUFFER
MR = TE16 MAINTENENCE REGISTER
DT = TE16 DRIVE TYPE
SN = TE16 SERIAL NUMBER
TC = TE16 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
```

```
1027
1028
1029                    EXAMPLE 5: THIS EXAMPLE SHOWS A READ DATA ERROR
1030                            WHICH OCCURRED, WITHOUT AN ACCOMPANING
1031                            STATUS ERROR, WHICH RESULTED IN A BAD RECORD.
1032
1033                            DRIVE NO. 3 *SLAVE NO. 1 *D 4 *P 0*F 14 *PATRN R
1034                            *BN 100 *RN 66-200 *RS 2000 *DE F
1035                            CN 0
1036                            G 11111111
1037                            B 00000000
1038                            CN 1
1039                            G 11111111
1040                            B 00000000
1041                            CN 2
1042                            G 11111111
1043                            B 00000000
1044                            CN 3
1045                            G 11111111
1046                            B 00000000
1047                            CN 4
1048                            G 11111111
1049                            B 00000000
1050                            CN 5
1051                            G 11111111
1052                            B 00000000
1053                            CN 6
1054                            G 11111111
1055                            B 00000000
1056                            CN 7
1057                            G 11111111
1058                            B 00000000
1059                                    BAD RECORD
1060
1061                    EXAMPLE 6: THE FOLLOWING EXAMPLE SHOWS THE
1062                            RESULT OF A SPACE OPERATION THAT
1063                            SHOULD HAVE SPACED REVERSE OVER
1064                            AN ENTIRE 100 RECORD BLOCK BUT
1065                            WHICH TERMINATED AT THE END OF 40
1066                            RECORDS.  LEAVING A POSITION ERROR OF 40
1067
1068                            DRIVE NO. 2 *SLAVE NO. 6 *D 2 *P 0 *F 14
1069                            *BN 3 *RN 100-100 *RS 1000 *SE R
1070                            ERR AMT 40
1071
```

```
1073
1074
1075                    EXAMPLE 7: THIS EXAMPLE REFLECTS AN ERROR DETECTED
1076                            WHILE WRITING A TAPE MARK (TM) AT THE END
1077                            OF THE CURRENT DATA BLOCK PER OPTION RESPONSE
1078                            TM=1.  NOTE THAT THE TM RECORD NUMBER
1079                            IS ONE GREATER THAN THE TOTAL NUMBER OF DATA
1080                            RECORDS IN THE CURRENT BLOCK.
1081
1082                            DRIVE NO. 1 *SLAVE NO. 1 *D 2 *P 0 *F 14
1083                            *BN 67 *RN 101-100 *RS 36 *WE TM
1084                            CS1 144226
1085                            CS2 300
1086                            DS 150604
1087                            ER 1000
1088                            WC 0
1089
1090                    EXAMPLE 8: THIS EXAMPLE SHOWS TWO (2) PRINTOUTS
1091                            REFLECTING A WRITE RETRY WHICH WAS NOT
1092                            SUCCESSFUL THE FIRST TIME, BUT WHICH
1093                            DID RECOVER ON THE SECOND.
1094                            THE UNSUCCESSFUL RETRY IS LOGGED AS
1095                            A SUSPECTED BAD TAPE SPOT BY ITS
1096                            BLOCK AND RECORD NUMBER.
1097
1098
1099                            DRIVE NO. 0 *SLAVE NO. 2 *D 4 *P 0 *F 14 *PATRN 6
1100                            *BN 2 *RN 12-20 *RS 667 *WE
1101                            CS1 144260
1102                            CS2 100
1103                            DS 150640
1104                            ER 100
1105                            WC 0
1106                            ***ORIGINAL ERROR***
1107
1108                            DRIVE NO. 0 SLAVE NO. 2 *D 4 *P 0 *F 14 *PATRN 6
1109                            *BN 2 *RN 12-20 *RS 667 *WE
1110                            CS1 144260
1111                            CS2 100
1112                            DS 150640
1113                            ER 100
1114                            WC 0
1115                            SUSPECT BAD TAPE
1116                            RETRY: 0
1117                            REPT: 0
1118                            RECOVERED
1119                            RETRY: 1
1120
```

```
1122
1123
1124                    EXAMPLE 9: IF , DURING A WRITE RETRY THE BACKSPACE
1125                            OR THE ERASE OPERATION RESULT IN AN ERROR,
1126                            THE ERROR WILL BE PRINTED AND THE PROGRAM
1127                            HALTED.  THIS EXAMPLE SHOWS THE ERROR PRINT
1128                            FOR A SPACE AND AN ERASE (2 EXAMPLES)
1129
1130                            DRIVE NO. 1 *SLAVE NO. 1 *D 3 *P 0 *F 14
1131                            BN 12 *RN 8-64 *RS 500 *SE RTRY
1132                            ERR AMT 1
1133
1134                            DRIVE NO. 1 *SLAVE NO. 1 *D 3 *P 0 *F 14
1135                            *BN 12 *RN 8-64 *RS 500 *ERASE
1136                            CS1 144224
1137                            CS2 100
1138                            DS 150600
1139                            ER 400
1140                            WC 0
1141
1142                    EXAMPLE 10: THIS EXAMPLE SHOWS THE PRINTOUT FROM
1143                            A REWIND OPERATION WHICH DOES NOT HAVE
1144                            BOT SET AT THE END.
1145
1146                            DRIVE NO. 2 *SLAVE NO. 3 *D 3 *P 0 *F 14
1147                            *BN 66 *RN 15-20 *RS 1000
1148                            NOT BOT ON REWIND: HALT
1149
1150                    EXAMPLE 11: THIS EXAMPLE SHOWS THE PRINTOUT MADE WHEN
1151                            THERE IS NO INTERRUPT RETURNED AT THE END
1152                            OF AN OPERATION.
1153
1154                            DRIVE NO. 7 *SLAVE NO. 7 *D 2 *P 1 *F 14
1155                            *BN 1 *RN 25-26 *RS 1200
1156                            NO INTERRUPT
```

```
1158
1159
1160                          10.     STATISTICS PRINTOUT
1161                                  ---------- ----
1162
1163                          THE PROGRAM, THROUGH ITS ERROR CHECKING, IS ABLE TO GATHER CERTAIN
1164                          STATISTICS ABOUT THE PERFORMANCE OF EACH UNIT UNDER TEST.
1165                          THIS INFORMATION IS PRINTED OUT WHENEVER A UNIT IS REWOUND
1166                          FROM END OF TAPE, OR BECAUSE IT IS TO BE REMOVED FROM TESTING
1167                          DUE TO SOME CATASTROPHIC ERROR. (POSITION LOST, BAD TAPE OVERFLOW)
1168                          THE STATISTICS MAY BE PRINTED AT ANY TIME BY SETTING
1169                          SWITCH 14 TO A ONE (1). THIS PRESENTS A PICTURE OF PERFORMANCE
1170                          UP TO THIS TIME. THE STATISTICS WILL BE CLEARED UPON REWIND OF THE
1171                          UNIT; BUT NOT BY SETTING SW 14.
1172
1173                          STATISTICS PRINT EXAMPLE (A HEADER WILL PRECEED THE STATS)
1174
1175                          DROPS: 0 3 0 0 0 6 45 0
1176                          PICKS: 1 0 0 0 0 0 0 2
1177                          RETRY: 1
1178                          WTERR: 2
1179                          REFWD: 3
1180                            SOFT: 2
1181                            HARD: 1
1182                          DEFWD: 0
1183                          REREV: 4
1184                            SOFT: 1
1185                            HARD: 3
1186                          DEREV: 0
1187                          2 BAD TAPE SPOTS
1188                          0 *BN 1 *RN 2
1189                          1 *BN 15 *RN 100
1190
1191                          ** NOTE ** DROPS AND PICKS REFLECT CORE BIT POSITIONS.
1192                                     THE FOLLOWING IS A TABLE OF CORE BITS TO TRACK NUMBER.
1193
1194                                     TRACK NO. 7 6 5 3 9 1 8 2
1195                                     CORE BIT  7 6 5 4 3 2 1 0
1196
1197                          DROPS: NUMBER OF DATA BITS DROPPED: PER CORE BIT(SEE NOTE ABOVE)
1198                          PICKS: NUMBER OF DATA BITS PICKED UP: PER CORE BIT(SEE NOTE ABOVE)
1199                          RETRY: NUMBER OF WRITE RETRIES
1200                          WTERR: NUMBER OF WRITE ERRORS NOT ASSOCIATED WITH BAD TAPE
1201                          REFWD: NUMBER OF READ FORWARD STATUS ERRORS
1202                          REREV: NUMBER OF READ REVERSE STATUS ERRORS
1203                            SOFT: NUMBER OF RECOVERED READ ERRORS
1204                            HARD: NUMBER OF UNRECOVERED READ ERRORS
1205                          DEFWD: NUMBER OF FORWARD DATA ERRORS WITH NO ASSOCIATED STATUS ERROR
1206                          DEREV: NUMBER OF REVERSE DATA ERRORS WITH NO ASSOCIATED STATUS ERROR
1207
```

11. AUTO SEQUENCE

HE 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:

```
        TE16 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
THEN COMPLIMENT PATTERN. REPEATED FOR A FULL BUFFER
RANDOM DATA: RANDOM

12. TESTING PROCEDURES

AS PREVIOUSLY STATED THIS PROGRAM CONTAINS NO FIXED TESTS. THE ENTIRE TEST CYCLE TO BE EXECUTED IS DESCRIBED BY THE OPERATOR THOUGH RESPONSES TO TELETYPE REQUESTS FOR PARAMETERS AND CONSOLE SWITCH SETTINGS FOR OPERATION. THE OPERATION SELECTED WILL BE EXECUTED WITH THE PARAMETERS ENTERED CONTINUOUSLY ON EACH AVAILABLE UNIT, ONE BLOCK AT A TIME, UNTIL STOPPED BY THE OPERATOR. THE OPERATION MAY BE CHANGED DYNAMICALLY BY CHANGING THE CONSOLE SWITCHES AT ANY TIME. THE PROGRAM WILL ATTEMPT TO PERFORM ANY OPERATION SET AND THEREFORE CAUTION SHOULD BE TAKEN TO ASSURE THAT THE UNIT IS CAPABLE OF PERFORMING AS REQUESTED. FOR INSTANCE, ONE SHOULD NOT ATTEMPT TO PERFORM READ OPERATIONS ON A TAPE WHICH HAS NOT BEEN WRITTEN AS THE DATA, IF ANY, IS UNPREDICTABLE. HOWEVER, IF A TAPE HAS BEEN WRITTEN WITH THIS PROGRAM, IT CAN BE READ AS OFTEN AS DESIRED WITHOUT BEING REWRITTEN. THIS IS A GOOD PROCEDURE TO USE FOR TESTING TAPE COMPATABILITY. SCOPING OF TAPE UNITS BECOMES SIMPLE; BY SETTING THE DESIRED OPERATION AND ITS PARAMETER, A UNIT MAY BE CONTINUOUSLY EXERCISED IN ANY MANNER DESIRED. BY USING THE VARIOUS ERROR CONTROL SWITCHES AND ENTERING THE NEEDED STALL, ANY FUNCTION CAN BE SCOPED RATHER EASILY. RELIABILITY TESTING CAN BE PREFORMED 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 TOZZLE 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.

```
                        .LIST   BIN,LOC,SEQ
                        .TITLE CZTEDEO TM03-TE16/TU77 DRT
                        ;DATA RELIABILITY TEST
                        ;AC-A800E-MC
                        ;21 FEB 1977
                        ;J.G.ADAMS

                        ;REVISED (++B) J.G.ADAMS MAY 1978        1)INCORRECT RECORD COUNT
                        ;++B                                     STORED WHEN EOT REACHED ON WRITE
                        ;++B                                     2)ADJUST STACK PTR ON BAD TAPE OVFLW
                        ;++B                                     3)ADDED TU77 TEST CAPABILITY
                        ;++B                                     4)DOES NOT GENERATE LRC/CRC ON FIRST
                        ;++B                                     RECORD IN AUTO ACCEPT MODE

                        ;(++C)  M.PAGE  FEB  79
                        ;++C                                     RECORD NUMBERING SYSTEM NOT CONSISTENT
                        ;                                        BETWEEN FORWARD AND REVERSE TAPE MOVEMEN
                        ;                                        FORMAT ERROR (BIT 4) MADE RETRYABLE

                        ;REVISED JAN 1984 BY J.A.C.HITT          FIX SO THAT RECORD SIZE CAN BE
                        ;(CZTEDD)                                A MAXIMUM OF 10000 OCTAL BYTES
                        ;                                        LONG.  THIS IS CONDITIONAL IN
                        ;                                        THAT THERE MUST BE ENOUGH MEMORY
                        ;                                        FOR THE BUFFER, OTHERWISE RECORD
                        ;                                        SIZE WILL BE 4000 OCTAL.  THIS
                        ;                                        CORRECTS AID REPORT #CC0001450

                        ;REVISED FEB 1984 BY J. HITT             ADD XON/XOFF FUNCTIONALITY FOR
                        ;(CZTEDE)                                PRINTOUTS.

                        .MCALL  .$ACT11,.$EOP,$SAVE,$RESTORE,$CHAIN
                        .NLIST  MC
                        .LIST   ME
                        .ENABLE ABS,AMA

                        ;CONSOLE SWITCHES**********

                        ;SW15:  1=STOP ON ERROR
                        ;       0=CONTINUE ON ERROR
                        ;SW14:  1=PRINT READ/WRITE STATS
                        ;       0=DO NOT PRINT STATS
                        ;SW13:  1=DO NOT CHECK DATA
                        ;       0=CHECK DATA
                        ;SW12:  1=DO NOT CHECK WRITE ERRORS
                        ;       0=CHECK WRITE ERRORS
                        ;SW11:  1=DO NOT CHECK READ ERRORS
                        ;       0=CHECK READ ERRORS
                        ;SW10:  1=DO NOT PRINT ERRORS
                        ;       0=PRINT ERRORS
                        ;SW9:   1=REWIND TAPE
                        ;       0=DO NOT REWIND
                        ;SW8:   1=USE RANDOM DATA
                        ;       0=USE FIXED DATA PATTERN
                        ;SW7:   1=USE RANDOM CHARACTER COUNT
                        ;       0=USE FIXED CHAR COUNT
                        ;SW6:   1=USE RANDOM RECORD COUNT
```

```
1391                                    ;       0=USE FIXED RECORD COUNT
1392                                    ;SW5:  1=YOZZLE ON CURRENT RECORD
1393                                    ;       0=DO NOT YOZZLE
1394                                    ;SW4:  1=DO BOTH READ AND WRITE RETRIES
1395                                    ;       0=INHIBIT RETRIES
1396                                    ;SW3:  1=DO NOT READ FORWARD
1397                                    ;       0=READ FORWARD
1398                                    ;SW2:  1=DO NOT READ REVERSE
1399                                    ;       0=READ REVERSE
1400                                    ;SW1:  1=READ FORWARD FIRST
1401                                    ;       0=READ REVERSE FIRST
1402                                    ;SW0:  1=DO NOT WRITE
1403                                    ;       0=WRITE
1404                            ;IF SWR <15::00> = 177777 OR NOT AVAILABLE USE SOFTWARE SWITCH REGISTER
```

```
                        ;TM03-TE16,TU77 REGISTER BITS**********

                        ;15 ;14 ;13 ;12 ;11 ;10 ;09 ;08 ;07 ;06 ;05 ;04 ;03 ;02 ;01 ;00 ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;CS1:   ;SC ;TRE;MCP;SPR;DNA;PSL;A17;A16;RDY;IE ;FUN;FUN;FUN;FUN;FUN;GO ;
                        ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;WC:    ; WORD                                                          ;
                        ; COUNT                                                         ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;BA:    ; BUS                                                           ;
                        ; ADDRESS                                                       ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;FC:    ; FRAME                                                         ;
                        ; COUNT                                                         ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;CS2:   ;DLT;WCE;UPE;NED;NEM;PGE;MXF;MDP;OR  ;IR  ;CLR;PAT;BAI;U2 ;U1 ;U0 ;
                        ;   ;   ;   ;   ;   ;   ;   ;ERR;   ;   ;   ;   ;   ;   ;   ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;DS:    ;ATA;ERR;PIP;MOL;WRL;EOT;SPR;DPR;DRY;SSC;PES;SDN;IDB;EOF;BOT;SLA;
                        ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;ER:    ;CDE;UNS;OPI;DTE;NEF;CS ;FCE;NSG;PEF;INC;LAT;FMT;CNT;RMR;ILR;ILF;
                        ;CRC;   ;   ;   ;ITM;   ;   ;LRC;VPE;3PE;   ;BPE;   ;   ;   ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;ATS:   ; ATTENTION                                                     ;
                        ; SUMMARY                                                       ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;CC:    ; CHECK                                                         ;
                        ; CHARACTER                                                     ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;DB:    ; DATA                                                          ;
                        ; BUFFER                                                        ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;MR:    ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;WRT;MM ;OP ;OP ;OP ;OP ;MM ;
                        ; 7 ; 6 ; 5 ; 4 ; 3 ; 2 ; 1 ; 0 ; P ;CLK;CLK; 4 ; 3 ; 2 ; 1 ;GO ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;DT:    ; 1 ; 1 ;   ; 0 ;   ;SPR;   ;   ;   ;   ;   ; 1 ; 0 ; 1 ;125; 75; 45;
                        ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;IPS;IPS;IPS;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;SN:    ; SERIAL                                                        ;
                        ; NUMBER                                                        ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                ;TC:    ;ACL;FCS;TCW;ENA;SPR;DEN;DEN;DEN;FMT;FMT;FMT;FMT;EVN;SSN;SSN;SSN;
                        ;   ;   ;   ;DT ;   ; 4 ; 2 ; 1 ; 8 ; 4 ; 2 ; 1 ;PAR; 4 ; 2 ; 1 ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
1466
1467                                    ;TRAP CATCHERS**********
1468
1476            000020                  .=20
1477   000020   023110                  .WORD   TTOUT           ;SET IOT TRAP TO TTOUT ROUTINE
1478   000022   000340                  .WORD   340             ;PRIORITY LEVEL 7
1479
1480            000004                  TYPE=IOT                ;EQUATE TYPE TO AN IOT INSTRUCTION
1481            000034                  .=34
1482   000034   023326                  .WORD   OCTP            ;SET TRAP TRAP TO OCTP ROUTINE
1483   000036   000340                  .WORD   340
1484            104400                  TYPOCT=TRAP             ;EQUATE TYPOCT TO TRAP INSTRUCTION
1485
 (1)                                ;ACT11 HOOK **********
 (1)            000040                  $SVPC=.                 ;SAVE CURRENT LOCATION CTR
 (1)            000042                  .=42
 (1)   000042   000000                  .WORD   0
 (1)            000046                  .=46
 (1)   000046   005010                  .WORD   $ENDAD          ;SET LOCATION 46
 (1)            000052                  .=52
 (1)   000052   000000                  .WORD   0               ;SET LOCATION 52 = 0
 (1)            000040                  .=$SVPC                 ;RESTORE LOCATION CTR
 (1)
1486                                    ;TTY INTERRUPT VECTOR**********
1487            000060                  .=60
1488   000060   021050                  .WORD   TTINT           ;TTY INTERRUPT HANDLER ADDRESS
1489   000062   000340                  .WORD   340             ;PRIORITY LEVEL 7
1490
1491                                    ;SOFTWARE SWITCH REGISTER*********
1492                                    ;INVOKED IF SWR <15::00> = 177777 OR NOT AVAILABLE
1493            000176                  .=176
1494   000176   000000          SWREG:  .WORD   0
1495
1496                                    ;START ADDRESS**********
1497            000200                  .=200
1498   000200   000137   003032         JMP     START           ;ENTER PARAMETERS VIA TTY
1499
1500            000204                  .=204
1501   000204   000137   003250         JMP     STARTC          ;USE FIXED PARAMETERS; HOLD DATA
1502
1503            000210                  .=210
1504   000210   005037   014520         CLR     RDFL
1505   000214   000137   003256         JMP     STARTA          ;USE FIXED PARAMETERS; NEW DATA
1506
1507                                    ;MAG TAPE INTERRUPT VECTOR**********
1508
1509            000224                  .=224
1510   000224   021274                  MTINT                   ;MAG TAPE INTERRUPT HANDLER ADDRESS
1511   000226   000340                  340
1512
1513                                    ;AUTO SEQUENCE START**********
1514
1515            000240                  .=240
1516   000240   005237   000742         INC     ASEQF           ;SET AUTO SEQUENCE FLAG
1517   000244   000137   003234         JMP     STAUT           ;GO TO START OF AUTO SEQUENCE
```

```
                                        ;SHORT CONVERSATION RESTART********

                    000300              .=300
000300  005237  013560                  INC     SCVFL                   ;SET SHORT CONVERSATION FLAG
000304  000137  003032                  JMP     START                   ;ENTER SHORT PARAMETER LIST

                    000510              .=510
                                        ;TU16 REGISTER EQUIVS**********

000510  172440                  C1:     172440
000512  172442                  WC:     172442
000514  172444                  BA:     172444
000516  172446                  FC:     172446
000520  172450                  CS:     172450
000522  172452                  DS:     172452
000524  172454                  ER:     172454
000526  172456                  AS:     172456
000530  172460                  CC:     172460
000532  172462                  DB:     172462
000534  172464                  MR:     172464
000536  172466                  DT:     172466
000540  172470                  SN:     172470
000542  172472                  TC:     172472

                                        ;CONSTANTS**********

000544  172440          REGS:   172440          ;STARTING REGISTER ADDRESS (CS1)
000546  000224          VECT:   224             ;VECTOR ADDRESS (RH INTERRUPT)
000550  000000          DVN:    0               ;DRIVE NUMBER
000552  000000          UDES:   0               ;UNIT DESCRIPTION (PARITY,DENSITY,UNIT,FORMAT)
000554  000100          RCNT:   100             ;RECORD COUNTER
000556  174000          FMCNT:  174000          ;NUMBER OF CHAR (4000) OCTAL IN TWOS COMPLEMENT
000560  174000          BUFMAX: 174000          ;MAXIMUM BUFFER SIZE
000562  026544          WDATA:  BUFBEG          ;START OF WRITE BUFFER
000564  032544          RDATA:  BUFBEG+4000     ;START OF READ BUFFER
000566  000001          PATRN:  1               ;DATA PATTERN SELECTOR (0 - 15) OCTAL
000570  000000          RDCMD:  0               ;READ COMMAND
000572  000001          TMEX:   1               ;TAPE MARK FLAG: 1=TM 0=NO TM
000574  000000          CRCC:   0               ;CRC CORRECTION FLAG (YES=1,NO=0)
000576  000000          INTRF:  0               ;INTERCHANGE READ 1=YES 0=NO
000600  000000          SPFLG:  0               ;SINGLE PASS 1=YES 0=NO
000602  000001          RSTAL:  1               ;READ STALL
000604  000001          WSTAL:  1               ;WRITE STALL
000606  000001          TSTAL:  1               ;TURN AROUND STAL
000610  002000          YSTAL:  2000            ;YOZZLE STAL
000612  000010          RETRY:  10              ;READ RETRY NUMBER
000614  177776          PSW:    177776          ;PROCESSOR STATUS
000616  177570          SWR:    177570          ;CONSOLE SWITCHES
000620  177560          TKS:    177560          ;TTY READ STATUS REGISTER
000622  177562          TKB:    177562          ;TTY READ BUFFER
000624  177564          TPS:    177564          ;TTY PUNCH STATUS REGISTER
000626  177566          TPB:    177566          ;TTY PUNCH OUTPUT REGISTER
000630  177550          PRS:    177550          ;H/S READER STATUS REGISTER
000632  177552          PRB:    177552          ;H/S READER BUFFER
000634  153624          RANBAS: 153624          ;RANDOM NUMBER GENERATOR BASE
000636  032561          RANSAV: 032561          ;RANDOM NUMBER BUFFER
```

```
1575  000640  000100                RCSAV:  100             ;RECORD COUNT SAVE
1576  000642  174000                FCSAV:  174000          ;FRAME COUNT SAVE
1577
1578                                      ;FLAGS AND COUNTERS**********
1579
1580  000644  000000                TINF:   0               ;TTY ENTRY FLAG
1581  000646                        STFLG:
1582  000646  000000                TOB:    0               ;TTY OUTPUT BUFFER
1583  000650  000000                TIB:    0               ;TTY INPUT BUFFER
1584  000652  000000                TEMP1:  0               ;TEMP STORAGE
1585  000654  000000                TEMP2:  0               ;TEMP STORAGE
1586  000656  000000                TEMP3:  0               ;TEMP STORAGE
1587  000660  000000                EMADDR: 0               ;ERROR MSG ADDRESS STORAGE
1588  000662  000000                BLCNTR: 0               ;BLOCK COUNTER
1589  000664  000000                BBC:    0               ;BAD RECORD COUNTER
1590  000666  000000                EOTREC: 0               ;EOT FLAG
1591  000670  000000                RTRN:   0               ;INTERRUUPT RETURN STORAGE
1592  000672  000000                HDRFL:  0               ;HEADER FLAG
1593  000674  000000                STAL:   0               ;DELAY STORAGE
1594  000676  000000                PFLG:   0               ;PRINT FLAG
1595  000700  000000                MTC1:   0               ;MAG TAPE CONT REGISTER BUFFER
1596  000702  000000                UNP:    0               ;UNIT TABLE POINTER
1597  000704  000000                TMFLG:  0               ;TAPE MARK FLAG
1598  000706  000000                RPCNT:  0               ;REPEAT COUNTER
1599  000710  000000                RTCNT:  0               ;RETRY COUNTER
1600  000712  000000                DERFL:  0               ;DATA ERROR FLAG
1601  000714  000000                SERFL:  0               ;STATUS ERROR FLAG
1602  000716  000000                BCNT:   0               ;BIT COUNTER
1603  000720  000000                RTYFL:  0               ;RETRY FLAG
1604  000722  000000                UPS:    0               ;UNIT POINTER SAVE
1605  000724  000000                BDPP:   0               ;BITS DROPPED POINTER
1606  000726  000000                BPKP:   0               ;BITS PICKED POINTER
1607  000730  000000                ERSAV:  0               ;ERROR SAVE LOC
1608  000732  000000                BTFLG:  0               ;BAD TAPE FLAG
1609  000734  000000                BTSTF:  0               ;STATISTIC PRINT FLAG
1610  000736  000000                BTPT:   0               ;BAD TAPE POINTER
1611  000740  000000                ERTFL:  0               ;ERASE FLAG
1612  000742                        ENDFLG:
1613  000742  000000                ASEQF:  0               ;AUTO SEQ FLAG
1614  000744  000000                ABLCNT: 0               ;AUTO BLOCK COUNTER
1615  000746  000000                ASEQCF: 0               ;AUTO SEQ CONTINUOUS FLAG
1616  000750  000000                $CNTRLS: 0              ;XON/XOFF FLAG
```

```
1618
1619                                    ;UNIT ORDER AND DESCRIPTION TABLE **********
1620
1621  000752  000000            UN1:    0               ;THIS TABLE IS LOADED
1622  000754  000000            UN2:    0               ;WITH UNIT NUMBERS AND
1623  000756  000000            UN3:    0               ;THEIR DESCRIPTIONS IN
1624  000760  000000            UN4:    0               ;THE ORDER THAT THEY
1625  000762  000000            UN5:    0               ;WILL BE TESTED
1626  000764  000000            UN6:    0
1627  000766  000000            UN7:    0
1628  000770  000000            UN8:    0
1629  000772  177777            UNX:    -1
1630
1631                                    ;UNIT DROPS AND PICKS POINTERS*********
1632
1633  000774  001214            PIK1:   BP00
1634  000776  001234            PIK2:   BP10
1635  001000  001254            PIK3:   BP20
1636  001002  001274            PIK4:   BP30
1637  001004  001314            PIK5.   BP40
1638  001006  001334            PIK6:   BP50
1639  001010  001354            PIK7    BP60
1640  001012  001374            PIK8:   BP70
1641  001014  001414            DRP1:   BD00
1642  001016  001434            DRP2:   BD10
1643  001020  001454            DRP3:   BD20
1644  001022  001474            DRP4:   BD30
1645  001024  001514            DRP5:   BD40
1646  001026  001534            DRP6:   BD50
1647  001030  001554            DRP7:   BD60
1648  001032  001574            DRP8:   BD70
1649
1650                                    ;UNIT BAD TAPE POINTERS**********
1651
1652  001034  001614            BTADDR: BT00
1653  001036  001720                    BT01
1654  001040  002024                    BT02
1655  001042  002130                    BT03
1656  001044  002234                    BT04
1657  001046  002340                    BT05
1658  001050  002444                    BT06
1659  001052  002550                    BT07
1660
1661                                    ;UNIT WRITE RETRY COUNTER**********
1662
1663                                ;SET START OF STATISTICS TABLE
1664  001054                        STTBL:
1665  001054  000000            RTY1:   0
1666  001056  000000            RTY2:   0
1667  001060  000000            RTY3:   0
1668  001062  000000            RTY4:   0
1669  001064  000000            RTY5:   0
1670  001066  000000            RTY6:   0
1671  001070  000000            RTY7:   0
1672  001072  000000            RTY8:   0
1673
```

```
1674                                        ;UNIT WRITE ERRORS**********
1675
1676  001074  000000                WTER1:  0
1677  001076  000000                WTER2:  0
1678  001100  000000                WTER3:  0
1679  001102  000000                WTER4:  0
1680  001104  000000                WTER5:  0
1681  001106  000000                WTER6:  0
1682  001110  000000                WTER7:  0
1683  001112  000000                WTER8:  0
1684
1685                                        ;UNIT READ FORWARD ERRORS**********
1686
1687  001114  000000                RDER1:  0
1688  001116  000000                RDER2:  0
1689  001120  000000                RDER3:  0
1690  001122  000000                RDER4:  0
1691  001124  000000                RDER5:  0
1692  001126  000000                RDER6:  0
1693  001130  000000                RDER7:  0
1694  001132  000000                RDER8:  0
1695
1696                                        ;UNIT DATA ERRORS FORWARD**********
1697
1698  001134  000000                DATER1: 0
1699  001136  000000                        0
1700  001140  000000                        0
1701  001142  000000                        0
1702  001144  000000                        0
1703  001146  000000                        0
1704  001150  000000                        0
1705  001152  000000                        0
1706
1707                                        ;UNIT READ REVERSE ERRORS**********
1708
1709  001154  000000                RDERR1: 0
1710  001156  000000                        0
1711  001160  000000                        0
1712  001162  000000                        0
1713  001164  000000                        0
1714  001166  000000                        0
1715  001170  000000                        0
1716  001172  000000                        0
1717
1718                                        ;UNIT DATA ERRORS REVERSE**********
1719
1720  001174  000000                DEREV1: 0
1721  001176  000000                        0
1722  001200  000000                        0
1723  001202  000000                        0
1724  001204  000000                        0
1725  001206  000000                        0
1726  001210  000000                        0
1727  001212  000000                        0
```

M3


```
1729                                    ;DROPS + PICKS PER CHANNEL PER UNIT**********
1730
1731  001214  000000            BP00:   0
1732          001234                    .=.+16
1733  001234  000000            BP10:   0
1734          001254                    .=.+16
1735  001254  000000            BP20:   0
1736          001274                    .=.+16
1737  001274  000000            BP30:   0
1738          001314                    .=.+16
1739  001314  000000            BP40:   0
1740          001334                    .=.+16
1741  001334  000000            BP50:   0
1742          001354                    .=.+16
1743  001354  000000            BP60:   0
1744          001374                    .=.+16
1745  001374  000000            BP70:   0
1746          001414                    .=.+16
1747  001414  000000            BD00:   0
1748          001434                    .=.+16
1749  001434  000000            BD10:   0
1750          001454                    .=.+16
1751  001454  000000            BD20:   0
1752          001474                    .=.+16
1753  001474  000000            BD30:   0
1754          001514                    .=.+16
1755  001514  000000            BD40:   0
1756          001534                    .=.+16
1757  001534  000000            BD50:   0
1758          001554                    .=.+16
1759  001554  000000            BD60:   0
1760          001574                    .=.+16
1761  001574  000000            BD70:   0
1762          001614                    .=.+16
1763
1764
```

```
1766
1767                                    ;UNIT BAD TAPE COUNTER:16 PER SLAVE**********
1768
1769  001614  000000            BT00:   0
1770          001720                    .=.+102
1771  001720  000000            BT01:   0
1772          002024                    .=.+102
1773  002024  000000            BT02:   0
1774          002130                    .=.+102
1775  002130  000000            BT03:   0
1776          002234                    .=.+102
1777  002234  000000            BT04:   0
1778          002340                    .=.+102
1779  002340  000000            BT05:   0
1780          002444                    .=.+102
1781  002444  000000            BT06:   0
1782          002550                    .=.+102
1783  002550  000000            BT07:   0
1784          002654                    .=.+102
1785
1786                                    ;UNIT END OF TAPE COUNTERS 1 PER SLAVE*******
1787
1788  002654  000000            EOTCO:  0
1789  002656  000000                    0
1790  002660  000000                    0
1791  002662  000000                    0
1792  002664  000000                    0
1793  002666  000000                    0
1794  002670  000000                    0
1795  002672  000000                    0
1796
1797                                    ;UNIT READ FORWARD SOFT ERROR*******
1798
1799  002674  000000            RFSOFT: 0
1800  002676  000000                    0
1801  002700  000000                    0
1802  002702  000000                    0
1803  002704  000000                    0
1804  002706  000000                    0
1805  002710  000000                    0
1806  002712  000000                    0
1807
1808                                    ;UNIT READ REVERSE SOFT ERROR*******
1809
1810  002714  000000            RRSOFT: 0
1811  002716  000000                    0
1812  002720  000000                    0
1813  002722  000000                    0
1814  002724  000000                    0
1815  002726  000000                    0
1816  002730  000000                    0
1817  002732  000000                    0
1818
```

```
1820
1821                                    ;UNIT READ FORWARD HARD ERROR*******
1822
1823  002734  000000            RFHARD: 0
1824  002736  000000                    0
1825  002740  000000                    0
1826  002742  000000                    0
1827  002744  000000                    0
1828  002746  000000                    0
1829  002750  000000                    0
1830  002752  000000                    0
1831
1832                                    ;UNIT READ REVERSE HARD ERROR*******
1833
1834  002754  000000            RRHARD: 0
1835  002756  000000                    0
1836  002760  000000                    0
1837  002762  000000                    0
1838  002764  000000                    0
1839  002766  000000                    0
1840  002770  000000                    0
1841  002772  000000                    0
1842                                    ;SET END OF STATISTICS TABLE
1843  002774                    ENDTBL:
1844
1845                                    ;DATA PATTERN GENERATORS***********
1846
1847  002774  002774            DATBL:  .                       ;ENTRY TABLE
1848  002776  013772            DATA0:  DAT0                    ;EXTERNAL INPUT FROM H/S READER(SEE MAINDEC-11-D2TUF)
1849  003000  014132            DATA1:  DAT1                    ;ALL ONES
1850  003002  014152            DATA2:  DAT2                    ;ALL ZEROS
1851  003004  014156            DATA3:  DAT3                    ;WALKING ONE
1852  003006  014202            DATA4:  DAT4                    ;WALKING ZERO
1853  003010  014212            DATA5:  DAT5                    ;ALTERNATING ONE/ZERO
1854  003012  014220            DATA6:  DAT6                    ;ALTERNATING ZERO/ONE
1855  003014  014226            DATA7:  DAT7                    ;ALTERNATING ONE/ZERO IN ALTERNATING CHARACTERS
1856  003016  014254            DATA10: DAT10                   ;WALKING ONE/ALL ONE IN ALTERNATING CHARACTERS
1857  003020  014304            DATA11: DAT11                   ;ALL BITS 0-377
1858  003022  014324            DATA12: DAT12                   ;ALL BITS 377-0
1859  003024  014346            DATA13: DAT13                   ;ALTERNATING CHARACTERS  0 AND 377
1860  003026  014356            DATA14: DAT14                   ;WALKING ZERO/ALL ZERO IN ALTERNATING CHARACTERS
1861  003030  014406            DATA15: DAT15                   ;AUTO SEQUENCE PATTERN 0,0,-1,-1,-1,0,0
1862
```

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

                                ;START 200, & 300**********
003030  012706  000500        START:  MOV     #500,SP         ;SET STACK PTR
003036  005037  000742                CLR     ASEQF           ;CLEAR AUTO SEQUENCE FLAG
                                        ;**JACH>>>(1/84)
                                        ;THIS NEXT SECTION IS USED TO TEST IF THERE IS MORE THAT 16K
                                        ;OF MEMORY.  IF SO, THEN ALLOW LARGE RECORD LENGTHS (10000).
                                        ;OTHERWISE DEFAULT TO 4000 OCTAL.
                                        ;
003042  013746  000004                MOV     @#4,-(SP)       ;SAVE CONTENTS OF TRAP
003046  012737  003076  000004        MOV     #AA1$,@#4       ;SET UP NEW TRAP
003054  005000                        CLR     R0              ;CLEAR "TRAP FLAG"
003056  005737  040000                TST     @#40000         ;TEST AT THE 16K BOUNDARY
003062  005700                        TST     R0              ;CHECK R0 AND SEE IF STILL ZERO
003064  001012                        BNE     BB1$            ;IF SO, THEN NO TRAP OCCURRED
003066  012737  170000  000560        MOV     #170000,BUFMAX  ;IF NO TRAP, THEN ALLOW 10000 BYTE RECORDS
003074  000406                        BR      BB1$            ;AND CONTINUE
                                ;
003076  012737  174000  000560  AA1$:  MOV     #174000,BUFMAX  ;TRAP   THEN ALLOW ONLY 4000 BYTE RECORDS
003104  012700  177777                MOV     #177777,R0      ;SET THAT TRAP OCCURRED
003110  000002                        RTI                     ;AND RETURN FROM TRAP
                                ;
003112  013700  000560        BB1$:   MOV     BUFMAX,R0       ;PUT MAX RECORD LENGTH IN R0 AND CONVERT
003116  005400                        NEG     R0              ;IT TO A POSITIVE NUMBER
003120  012737  026544  000562        MOV     #BUFBEG,WDATA   ;SET STARTING ADDRESS OF WRITE BUFFER
003126  012737  026544  000564        MOV     #BUFBEG,RDATA   ;
003134  060037  000564                ADD     R0,RDATA        ;SET STARTING ADDRESS OF READ BUFFER
003140  012637  000004                MOV     (SP)+,@#4       ;RESTORE
                                        ;**JACH<<<
                                ;
003144  005027                        CLR     (PC)+           ;;CLEAR CHAIN INDICATOR
003146  000000                CHNFLG: .WORD   0               ;;CHAIN MODE INDICATOR
                                        ;;1=0 = CHAIN/NOT CHAIN MODE
005150  005737  000042                TST     @#42            ;;BRANCH IF  IN DUMP MODE
005154  001407                        BEQ     50$
005156  012737  000176  000616        MOV     #SWREG,SWR      ;;INVOKE SOFTWARE SWR
005164  005237  003146                INC     CHNFLG          ;;SET CHNFLG = CHAIN MODE
003170  000137  003174                JMP     3$              ;;GO TO CHAIN ADDRESS
003174                          50$:
003174  122737  000006  000041  3$:   CMPB    @6,@#41         ;BRANCH IF LOADED VIA TMDP
```

```
C2TEDE0.TMO3 TE16.TOC2 DRT     MACY11 30(1046)  07-MAR-84  14:21  PAGE 33-1
C2TEDE.P11    07-MAR-84 14:04                                          SEQ 0042

 1912  003202  001003                        BNE     4$
 1913  003204  000004  026314                TYPE,MSG120             ;ADVISE USER TO REMOVE TMDP FROM SLAVE
 1914  003210  000000                        HALT
 1915  003212  005737  003146          4$:   TST     CHNFLG          ;SEE IF IN CHAIN MODE
 1916  003216  001406                        BEQ     STAUT
 1917  003220  005237  000742                INC     ASEQF           ;SET AUTO SEQUENCE FLAG
 1918  003224  000004  024357                TYPE,MSG30              ;TYPE TITLE
 1919  003230  000137  021342                JMP     ASEQ0           ;GO TO AUTO SEQUENCER
 1920
 1921                                        ;START 240**********
 1922  003234  012737  000001  000644  STAUT: MOV     #1,TINF         ;SET TTY ENTRY FLAG
 1923  003242  005037  014520                CLR     RDFL            ;CLEAR RANDOM DATA FLAG
 1924  003246  000405                        BR      STARTB
 1925
 1926                                        ;START 204**********
 1927  003250  005037  000644          STARTC: CLR     TINF            ;CLEAR TTY INPUT FLAG
 1928  003254  000442                        BR      STARTD
 1929
 1930                                        ;START 210**********
 1931  003256  005037  000644          STARTA: CLR     TINF            ;CLEAR TTY ENTRY FLAG
 1932  003262  012700  000646          STARTB: MOV     #STFLG,R0       ;GET STARTING ADDRESS OF FLAGS
 1933  003266  012701  000074                  MOV     #ENDFLG-STFLG,R1
 1934  003272  105020                  1$:     CLRB    (R0)+           ;CLEAR FLAGS AND COUNTERS
 1935  003274  005301                        DEC     R1
 1936  003276  001375                        BNE     1$
 1937  003300  012706  000500                MOV     #500,SP         ;SET STACK POINTER
 1938  003304  004737  004234                JSR     PC,RANSET       ;GO RESET RANDOM BASE
 1939  003310  012700  001054                MOV     #STTBL,R0       ;GET STARTING ADDRESS OF STAT TABLE
 1940  003314  012701  001720                MOV     #ENDTBL-STTBL,R1        ;AND # OF BYTES IN TABLE
 1941  003320  105020                  2$:     CLRB    (R0)+           ;CLEAR STATISTIC COUNTERS
 1942  003322  005301                        DEC     R1
 1943  003324  001375                        BNE     2$
 1944  003326  012700  000752                MOV     #UN1,R0         ;SET ALL SLAVES ON LINE
 1945  003332  022710  177777          3$:   CMP     #-1,(R0)        ;BRANCH IF AT END OF TABLE
 1946  003336  001403                        BEQ     4$
 1947  003340  042720  040000                BIC     #040000,(R0)+   ;MARK SLAVE ON-LINE
 1948  003344  000772                        BR      3$
 1949  003346  012737  177777  013766  4$:   MOV     #-1,PATS        ;PRESET PATTERN
 1950  003354  012737  000001  000662  STARTE: MOV     #1,BLCNTR       ;PRESET BLOCK COUNTER
 1951  003362  013746  000004          STARTD: MOV     @#4,-(SP)       ;SAVE ERROR TRAP VECTOR
 1952  003366  013746  000006                  MOV     @#6,-(SP)
 1953  003372  022737  000176  000616        CMP     #SWREG,SWR      ;BRANCH IF SOFTWARE SWR
 1954  003400  001413                        BEQ     2$              ;ALREADY SELECTED
 1955  003402  012737  003426  000004        MOV     #1$,@#4         ;SET TIMEOUT TRAP TO 1$ BELOW
 1956  003410  005037  000006                CLR     @#6
 1957  003414  022777  177777  175174        CMP     #177777,@#SWR   ;BRANCH IF SWR = 177777 TRAP
 1958  003422  001402                        BEQ     3$              ;IF NOT AVAIL (1$) OTHERWISE
 1959  003424  000404                        BR      3$              ;GO TO 3$
 1960  003426  022626                  1$:   CMP     (SP)+,(SP)+     ;RESET STACK
 1961  003430  012737  000176  000616  2$:   MOV     #SWREG,SWR      ;SET SWR = SOFTWARE SWR
 1962  003436  012637  000006          3$:   MOV     (SP)+,@#6       ;RESTORE ERROR TRAP
 1963  003442  012637  000004                MOV     (SP)+,@#4
 1964  003446  012706  000500                MOV     #500,SP
 1965  003452  004737  012062                JSR     PC,TINP         ;GO GET PARAMETERS FROM TTY
 1966  003456  012777  000040  175034        MOV     #40,@CS         ;INITIALIZE
 1967  003464  005000                  STAUTO: CLR     R0              ;POINT TO FIRST ENTRY
```

```
1968  003466  022760  177777  000752  1$:     CMP     #-1,UN1(R0)       ;BRANCH IF LAST ENTRY
1969  003474  001406                          BEQ     2$
1970  003476  042760  100000  000752          BIC     #100000,UN1(R0)   ;CLEAR EOT FLAG
1971  003504  062700  000002                  ADD     #2,R0             ;POINT TO NEXT UNIT ENTRY
1972  003510  000766                          BR      1$                ;CONTINUE CLEARING
1973  003512  113737  005043  005042  2$:     MOVB    REOTC+1,REOTC     ;RESTORE EOT COUNTER
1974  003520  012777  000100  175072  START1: MOV     #100,@TKS         ;SET KEYBOARD IE BIT
1975  003526  013700  000702                  MOV     UNP,R0            ;R0 = UNIT TABLE POINTER
1976  003532  022760  177777  000752  STAR1A: CMP     #-1,UN1(R0)       ;BRANCH IF LAST ENTRY
1977  003540  001404                          BEQ     STAR1B
1978  003542  016037  000752  000552          MOV     UN1(R0),UDES      ;LOAD NEXT UNIT DESCRIPTION
1979  003550  000445                          BR      START4
1980  003552  005237  000662          STAR1B: INC     BLCNTR            ;BUMP BLOCK COUNTER
1981  003556  005737  000742                  TST     ASEQF             ;SEE IF AUTO SEQ
1982  003562  001411                          BEQ     STAR1C            ;IF NOT: BR
1983  003564  023737  000662  000744          CMP     BLCNTR,ABLCNT     ;SEE IF DONE SEQ
1984  003572  001005                          BNE     STAR1C            ;IF NOT: BR
1985  003574  005037  000662                  CLR     BLCNTR            ;RESET BLOCK CNTR
1986  003600  005037  000702                  CLR     UNP               ;RESET UNIT POINTER
1987  003604  000207                          RTS     PC                ;RETURN TO AUTO SEQ
1988  003606  005037  000702          STAR1C: CLR     UNP
1989  003612  005000                          CLR     R0
1990  003614  016037  000752  000552          MOV     UN1(R0),UDES      ;LOAD FIRST UNIT DESCRIPTION
1991  003622  105777  174770                  TSTB    @SWR              ;SEE IF RANDOM RECORD SIZE
1992  003626  100002                          BPL     START2            ;IF NOT: BR
1993  003630  004737  011776                  JSR     PC,CCNTR          ;GO GENERATE RANDOM RECORD SIZE
1994  003634  032777  000400  174754  START2: BIT     #400,@SWR         ;SEE IF RANDOM DATA
1995  003642  001402                          BEQ     START3            ;IF NOT: BR
1996  003644  004737  014456                  JSR     PC,DATR           ;GO GENERATE RANDOM DATA
1997  003650  032777  000100  174740  START3: BIT     #100,@SWR         ;SEE IF RANDOM RECORD COUNT
1998  003656  001402                          BEQ     START4            ;IF NOT: BR
1999  003660  004737  012036                  JSR     PC,RCNTR          ;GO GENERATE RANDOM RECORD COUNT
2000  003664  032760  140000  000752  START4: BIT     #140000,UN1(R0)   ;BRANCH IF UNIT AT EOT
2001  003672  001065                          BNE     START7            ;OR MARKED OFF-LINE
2002  003674  012777  000040  174616          MOV     #40,@CS           ;DO A MASSBUS CLEAR
2003  003702  013777  000550  174610          MOV     DVN,@CS           ;SET DRIVE NUMBER
2004  003710  013777  000552  174624          MOV     UDES,@TC          ;SET SLAVE NUMBER
2005  003716  105777  174600          1$:     TSTB    @DS               ;SEE IF SLAVE AVAIL
2006  003722  100405                          BMI     2$                ;IF SO: BR
2007  003724  005337  000674                  DEC     STAL
2008  003730  001372                          BNE     1$                ;AWAIT TUR
2009  003732  000137  020426                  JMP     OFFLINE           ;GO MARK DRIVE OFF LINE
2010  003736  004737  013606          2$:     JSR     PC,DSUP           ;GO SET UP WRITE DATA
2011  003742  004737  005350                  JSR     PC,INIT           ;INIT SLAVE
2012  003746  004737  005044                  JSR     PC,RWND           ;REWIND
2013  003752  004737  005464                  JSR     PC,WRITE          ;WRITE
2014  003756  013737  000606  000674          MOV     TSTAL,STAL        ;SET TURN AROUND DELAY
2015  003764  004737  011766                  JSR     PC,STALL                  ;DELAY
2016  003770  004737  007322                  JSR     PC,RSEQ           ;GO TO READ SEQUENCER
2017  003774  013737  000606  000674          MOV     TSTAL,STAL        ;SET TURN AROUND DELAY
2018  004002  004737  011766                  JSR     PC,STALL          ;DELAY
2019  004006  032777  040000  174602          BIT     #40000,@SWR       ;SEE IF SHOULD PRINT STATISTICS
2020  004014  001414                          BEQ     START7            ;IF NOT: BR
2021  004016  012700  000001                  MOV     #1,R0             ;SET RECORD COUNTER TO 1
2022  004022  004737  022126                  JSR     PC,PAPRT          ;PRINT CYCLE NUMBER
2023  004026  004737  004056                  JSR     PC,STP            ;GO PRINT STATS
```

```
 2024  004032  005237  000734                  INC     BTSTF           ;SET STAT ONLY PRINT
 2025  004036  004737  007240                  JSR     PC,BTPRT        ;PRINT BAD TAPE STATS
 2026  004042  005037  000734                  CLR     BTSTF           ;CLEAR FLAG
 2027  004046  062737  000002  000702  START7: ADD     #2,UNP          ;POINT TO NEXT UNIT
 2028  004054  000621                  START8: BR      START1          ;CONTINUE
```

```
                        ;********** SUBROUTINE TO PRINT STATISTICS **********
                        
004056  004737  016504  STP:    JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
004062  000004  025323           TYPE,MSG65              ;TYPE MSG
004066  013700  000702           MOV     UNP,R0
004072  016003  001054           MOV     RTY1(R0),R3
004076  104400                   TYPOCT                  ;PRINT RETRIES
004100  000004  025434           TYPE,MSG73              ;TYPE MSG
004104  016003  001074           MOV     WTER1(R0),R3
004110  104400                   TYPOCT                  ;PRINT WRITE ERRORS
004112  000004  025423           TYPE,MSG72              ;TYPE MSG
004116  016003  001114           MOV     RDER1(R0),R3
004122  104400                   TYPOCT                  ;PRINT READ FORWARD ERRORS
004124  000004  026201           TYPE,MSG113             ;TYPE MSG
004130  016003  002674           MOV     RFSOFT(R0),R3
004134  104400                   TYPOCT                  ;PRINT FORWARD SOFT ERRORS
004136  000004  026212           TYPE,MSG114             ;TYPE MSG
004142  016003  002734           MOV     RFHARD(R0),R3
004146  104400                   TYPOCT                  ;PRINT HARD FORWARE ERRORS
004150  000004  025520           TYPE,MSG77              ;TYPE MSG
004154  016003  001134           MOV     DATER1(R0),R3
004160  104400                   TYPOCT                  ;PRINT DATA ERROR FORWARD NUMBER
004162  000004  025355           TYPE,MSG68              ;TYPE MSG
004166  016003  001154           MOV     RDERR1(R0),R3
004172  104400                   TYPOCT                  ;PRINT REVESE ERROR NUMBER
004174  000004  026201           TYPE,MSG113             ;TYPE MSG
004200  016003  002714           MOV     RRSOFT(R0),R3
004204  104400                   TYPOCT                  ;PRINT REVERSE SOFT ERROR
004206  000004  026212           TYPE,MSG114             ;TYPE MSG
004212  016003  002754           MOV     RRHARD(R0),R3
004216  104400                   TYPOCT
004220  000004  025507           TYPE,MSG76              ;TYPE MSG
004224  016003  001174           MOV     DEREV1(R0),R3
004230  104400                   TYPOCT                  ;PRINT DATA REVERSE ERROR NUMBER
004232  000207                   RTS     PC              ;RETURN
                        
                                 ;RANDOM BASE RESET**********
                        
004234  012737  153624  000634  RANSET: MOV     #153624,RANBAS  ;RESET BASE
004242  012737  032561  000636          MOV     #32561,RANSAV   ;RESET BUFFER
004250  013737  000640  000554          MOV     RCSAV,RCNT      ;RESET RECORD COUNT
004256  013737  000642  000556          MOV     FCSAV,FMCNT     ;RESET FRAME COUNT
004264  000207                   RTS     PC
```

```
                                        ;*************************************************************
                                        ;REWIND FROM EOT:
                                        ;
                                        ;WHEN ANY TRANSPORT BEING TESTED REACHES END OF TAPE
                                        ;DURING A READ OR WRITE OPERATION, IT WILL BE REWOUND
                                        ;AND FLAGGED AS UNAVAILABLE UNTIL ALL AVAILABLE UNITS
                                        ;HAVE REACHED EOT AT WHICH TIME ALL TESTING WILL BE RESUMED
                                        ;AT A BLOCK COUNT OF ONE (1).  A MESSAGE WILL BE
                                        ;PRINTED ON THE SUPERVISORS CONSOLE AS EACH UNIT REACHES
                                        ;EOT AND IS REWOUND.
                                        ;*************************************************************

004266  013777  000552  174246  REOT:   MOV     UDES,@TC        ;LOAD TAPE CONTROL REGISTER
004274  013700  000702            REOT1:  MOV     UNP,R0          ;GET UNIT POINTER
004300  032760  040000  000750            BIT     #40000,UN1(R0)  ;BRANCH IF UNIT MARKED OFF-LINE
004306  001014                            BNE     2$
004310  012777  000011  174172            MOV     #11,@C1         ;DRIVE CLEAR
004316  105777  174200            1$:     TSTB    @DS             ;WAIT FOR DRY
004322  100375                            BPL     1$
004324  012777  000007  174156            MOV     #7,@C1          ;START REWIND
004332  005737  000732                    TST     BTFLG           ;SEE IF BAD TAPE OVERFLOW REWIND
004336  001004                            BNE     3$              ;IF SO: BR
004340  013700  000666            2$:     MOV     EOTREC,R0
004344  042700  100000                    BIC     #100000,R0      ;SET RECORD NUMBER OF EOT
004350  005037  000666            3$:     CLR     EOTREC          ;CLEAR EOT INDICATOR & REC COUNT
004354  004737  022126                    JSR     PC,PAPRT        ;PRINT HEADER
004360  022737  000002  000732            CMP     #2,BTFLG        ;SEE IF POSITION ERROR
004366  001004                            BNE     4$              ;IF NOT: BR
004370  012737  026074  004420            MOV     #MSG109,6$      ;SET POSITION ERROR MSG
004376  000407                            BR      5$
004400  022737  000001  000732  4$:       CMP     #1,BTFLG        ;SEE IF BAD TAPE OVERFLOW
004406  001006                            BNE     REOT1C          ;IF NOT: BR
004410  012737  025727  004420            MOV     #MSG106,6$      ;SET BAD TAPE OVERFLOW MSG
004416  000004                    5$:     TYPE                    ;TYPE MSG
004420  000000                    6$:     .WORD   0               ;WILL CONTAIN MESSAGE ADDRESS
004422  000411                            BR      REOT1E
004424  000004  024060            REOT1C: TYPE,MSG20              ;TYPE EOT MSG
004430  013704  000702                    MOV     UNP,R4
004434  005264  002654                    INC     EOTCO(R4)       ;BUMP CNTR
004440  016403  002654                    MOV     EOTCO(R4),R3
004444  104400                            TYPOCT                  ;PRINT EOT CNTR
004446  000004  025752            REOT1E: TYPE,MSG16A             ;TYPE MSG
004452  005037  000732                    CLR     BTFLG           ;CLEAR BAD TAPE FLAG
004456  004737  004056                    JSR     PC,STP          ;PRINT STATS
004462  004737  007240                    JSR     PC,BTPRT        ;PRINT BAD TAPE STATS
004466  013700  000702            REOT2:  MOV     UNP,R0          ;GET UNIT POINTER
004472  032760  040000  000752            BIT     #40000,UN1(R0)  ;BRANCH IF UNIT MARKED OFF-LINE
004500  001010                            BNE     REOT2A
004502  105777  174014                    TSTB    @DS             ;BRANCH IF DRY SET
004506  100405                            BMI     REOT2A
004510  005337  000674                    DEC     STAL
004514  001364                            BNE     REOT2           ;WAIT DRY
004516  000137  004426                    JMP     OFFLINE         ;GO MARK SLAVE OFFLINE

004522  105337  005042            REOT2A: DECB    REOTC           ;SEE IF LAST UNIT TO REACH EOT
004526  001410                            BEQ     REOT3           ;IF SO: BR
```

```
2131  004530  013700  000702                    MOV     UNP,R0
2132  004534  052760  100000  000752            BIS     #100000,UN1(R0) ;SET EOT FLAG
2133  004542  005726                            TST     (SP)+           ;RESET STACK POINTER
2134  004544  000137  004046                    JMP     START7          ;GO TO NEXT UNIT
2135  004550  113737  005043  005042  REOT3:    MOVB    REOTC+1,REOTC   ;RESTORE UNITS EOT COUNTER
2136  004556  005037  000702                    CLR     UNP
2137  004562  005000                            CLR     R0              ;POINT TO FIRST UNIT
2138  004564  016037  000752  000552  REOT4:    MOV     UN1(R0),UDES    ;LOAD UNIT DESCRIPTION
2139  004572  013777  000552  173742            MOV     UDES,@TC        ;SELECT SLAVE
2140  004600  032760  040000  000752            BIT     #40000,UN1(R0)  ;BRANCH IF UNIT NOT MARKED OFF-LINE
2141  004606  001412                            BEQ     1$
2142  004610  032777  010000  173704            BIT     #10000,@DS      ;BRANCH IF MEDIUM NOT ON LINE
2143  004616  001427                            BEQ     10$
2144  004620  062737  000401  005042            ADD     #401,REOTC      ;INCREMENT # OF UNITS UNDER TEST
2145  004626  042760  140000  000752            BIC     #140000,UN1(R0) ;MARK UNIT BACK ON-LINE
2146  004634  012777  000011  173646  1$:       MOV     #11,@C1         ;DRIVE CLEAR
2147  004642  105777  173654          2$:       TSTB    @DS             ;WAIT FOR DRIVE READY
2148  004646  100375                            BPL     2$
2149  004650  012777  000007  173632            MOV     #7,@C1          ;REWIND UNIT
2150  004656  032777  000002  173636  3$:       BIT     #2,@DS          ;WAIT FOR BOT TO SET
2151  004664  001774                            BEQ     3$
2152  004666  032777  020000  173626  4$:       BIT     #20000,@DS      ;WAIT FOR PIP TO CLEAR
2153  004674  001374                            BNE     4$              ;AWAIT PIP RESET
2154
2155  004676  042760  100000  000752  10$:      BIC     #100000,UN1(R0) ;CLEAR EOT FLAG
2156  004704  062737  000002  000702            ADD     #2,UNP
2157  004712  013700  000702                    MOV     UNP,R0          ;POINT TO NEXT UNIT
2158  004716  022760  177777  000[illegible]    CMP     #-1,UN1(R0)     ;BRANCH IF NOT LAST UNIT
2159  004724  001317                            BNE     REOT4
2160  004726  005037  000702          REOT7:    CLR     UNP             ;CLEAR UNIT POINTER
2161  004732  005037  000644                    CLR     TINF            ;CLEAR TTY INPUT FLAG
2162  004736  005737  000742                    TST     ASEQF           ;SEE IF AUTO SEQ
2163  004742  001402                            BEQ     REOTX           ;IF NOT; BR
2164  004744  005726                            TST     (SP)+           ;RESET STACK POINTER
2165  004746  000207                            RTS     PC              ;RETURN TO AUTO SEQ
2166  004750  004737  004234          REOTX:    JSR     PC,RANSET       ;GO RESET RANDOM BASE
2167  004754  012737  177777  013766            MOV     #-1,PATS        ;PRESET PATTERN
2168  004762  005037  014520                    CLR     RDFL            ;CLEAR RANDOM FLAG
2169  004766  005737  000600                    TST     SPFLG           ;SEE IF SINGLE PASS
2170  004772  001421                            BEQ     REOTXX          ;IF NOT; BR
2171  004774  000004  025630          TEND:     TYPE,MSG100             ;TYPE MSG
2172  005000  013700  000042                    MOV     @#42,R0         ;GET ACT11 RETURN ADDRESS
(1)   005004  001405                            BEQ     HERE            ;BRANCH IF NOT ACT11
(1)   005006  000005                            RESET
(1)   005010  004710          $ENDAD:   JSR     PC,(R0)
(1)   005012  000240                            NOP
(1)   005014  000240                            NOP
(1)   005016  000240                            NOP
(1)   005020  000240          HERE:     NOP
2173  005022  005737  003146                    TST     CHNFLG          ;BRANCH IF NOT CHAIN MODE
2174  005026  001402                            BEQ     1$
2175  005030  000137  003542                    JMP     ASEQ0           ;RETURN TO AUTO SEQUENCER
2176  005034  000000          1$:       HALT
2177  005036  000137  003354          REOTXX:   JMP     START1          ;RESTART AT BLOCK NUMBER ONE
2178  005042  000000          REOTC:    0               ;EOT UNIT COUNTER
```

```
                                        ;*********************************************************
                                        ;REWIND ALL AVAIL TAPES:
                                        ;
                                        ;THIS ROUTINE; ENTERED VIA CONSOLE SWITCH NINE (9),
                                        ;WILL REWIND ALL AVAILABLE TAPES TO BOT NO MATTER
                                        ;WHERE THEY ARE CURRENTLY POSITIONED AND RESUME TESTING
                                        ;ON THE CURRENTLY SELECTED UNIT.
                                        ;*********************************************************

005044  032777  001000  173544  RWND:   BIT     #1000,@SWR      ;SEE IF SHOULD REWIND
005052  001001                          BNE     RWNDA           ;IF SO: BR
005054  000207                          RTS     PC              ;ELSE EXIT
005056  013737  000702  000722  RWNDA:  MOV     UNP,UPS         ;SAVE UNIT POINTER
005064  005037  000702                  CLR     UNP             ;CLEAR POINTER
005070  005037  000666                  CLR     EOTREC          ;CLEAR EOT FLAG
005074  113737  005043  005042          MOVB    REOTC+1,REOTC   ;++B RESTORE UNIT CTR
005102  013700  000702          RWND0:  MOV     UNP,R0          ;POINT TO UNIT ENTRY
00510   022760  177777  000752          CMP     #-1,UN1(R0)     ;BRANCH IF LAST ENTRY
005114  001437                          BEQ     RWND2
005116  032760  140000  000752          BIT     #140000,UN1(R0) ;BRANCH IF ALREADY REWINDING
005124  001024                          BNE     RWND1A          ;OR MARKED OFF LINE
005126  016037  000752  000552          MOV     UN1(R0),UDES    ;SET UNIT DESCRIPTION
005134  013777  000552  173400          MOV     UDES,@TC        ;LOAD COMMAND REGISTER
005142  012777  000011  173340          MOV     #11,@C1         ;DRIVE CLEAR
005150  012777  000007  173332          MOV     #7,@C1          ;START REWIND
005156  105777  173340          1$:     TSTB    @DS
005162  100405                          BMI     RWND1A          ;IF DRY: BR
005164  005337  000674                  DEC     STAL
005170  001372                          BNE     1$              ;AWAIT DRY
005172  000137  020426                  JMP     OFFLINE         ;GO MARK UNIT OFF LINE
005176  042760  100000  000752  RWND1A: BIC     #100000,UN1(R0) ;CLEAR EOT FLAG
005204  062737  000002  000702          ADD     #2,UNP          ;BUMP POINTER
005212  000733                          BR      RWND0           ;DO NEXT UNIT
005214  005037  000702          RWND2:  CLR     UNP             ;CLEAR POINTER
005220  013700  000702          RWND3:  MOV     UNP,R0          ;POINT TO UNIT ENTRY
005224  022760  177777  000752          CMP     #-1,UN1(R0)     ;BRANCH IF LAST ENTRY
005232  001433                          BEQ     RWNDX
005234  016037  000752  000552          MOV     UN1(R0),UDES    ;SET UNIT DESCRIPTION
005242  032760  040000  000752          BIT     #40000,UN1(R0)  ;BRANCH IF UNIT MARKED OFF LINE
005250  001015                          BNE     RWND5
005252  013777  000552  173262          MOV     UDES,@TC        ;LOAD UNIT DESCRIPTION
005260  032777  020000  173234  1$:     BIT     #20000,@DS
005266  001374                          BNE     1$              ;AWAIT PIP RESET
005270  032777  000002  173224          BIT     #2,@DS          ;BRANCH IF SLAVE AT BOT
005276  001002                          BNE     RWND5
005300  000137  020426                  JMP     OFFLINE         ;PRINT OFFLINE MESSAGE
005304  062737  000002  000702  RWND5:  ADD     #2,UNP          ;BUMP POINTER
005312  012777  000011  173170          MOV     #11,@C1         ;DRIVE CLEAR
005320  000737                          BR      RWND3           ;DO NEXT UNIT

005322  013700  000722          RWNDX:  MOV     UPS,R0          ;RESTORE UNIT POINTER
005326  010037  000702                  MOV     R0,UNP
005332  016037  000752  000552          MOV     UN1(R0),UDES    ;RESET UNIT DESCRIPTION
005340  013777  000552  173174          MOV     UDES,@TC
005346  000207                          RTS     PC              ;RETURN TO TEST
```

```
2236
2237                                    ;*********************************************************
2238                                    ;INITIALIZE SELECTED SALVE
2239                                    ;THIS ROUTINE REWINDS AND SETS THE PROPER DENSITY IF
2240                                    ;THE DENSITY REQUIRED FOR THE TEST IS DIFFERENT FROM
2241                                    ;THE DENSITY AT WHICH THE SLAVE IS SELECTED.
2242                                    ;*********************************************************
2243
2244  005350  013746  000552      INIT:  MOV     UDES,-(SP)       ;GET UNIT DESCRIPTION
2245  005354  012777  000040  173136      MOV     #40,@CS          ;DO A MASS CLEAR
2246  005362  013777  000550  173130      MOV     DVN,@CS          ;LOAD DR1.
2247  005370  011677  173146              MOV     (SP),@TC         ;LOAD SLAVE # & SLAVE DESCRIPTION
2248  005374  042716  174377              BIC     #174377,(SP)     ;CLEAR ALL BUT DENSITY BITS
2249  005400  022726  001400              CMP     #1400,(SP)+      ;BRANCH IF NOT NRZ
2250  005404  001005                      BNE     1$
2251  005406  032777  000040  173106      BIT     #40,@DS          ;BRANCH IF SLAVE IS IN PE MODE
2252  005414  001422                      BEQ     4$               ;PES = 0
2253  005416  000404                      BR      2$
2254  005420  032777  000040  173074  1$: BIT     #40,@DS          ;BRANCH IF SLAVE IS IN PE MODE
2255  005426  001015                      BNE     4$               ;PES = 1
2256  005430  012777  000007  173052  2$: MOV     #7,@C1           ;LOAD REWIND COMMAND
2257  005436  105777  173060      20$: TSTB    @DS              ;WAIT FOR READY
2258  005442  100375                      BPL     20$
2259  005444  032777  020000  173050  3$: BIT     #20000,@DS       ;WAIT FOR PIP = 0
2260  005452  001374                      BNE     3$
2261  005454  012777  000011  173026      MOV     #11,@C1          ;CLEAR DRIVE
2262  005462  000207              4$: RTS     PC
```

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

005464  032777  000001  173124  WRITE:  BIT     #1,@SWR         ;SEE IF SHOULD WRITE
005472  001402                          BEQ     WRTE
005474  000137  006244                   JMP     WEX             ;IF NOT; BR
005500  013700  000554          WRTE:   MOV     RCNT,R0         ;R0=RECORD COUNT
005504  012737  023746  000660  W0:     MOV     #MSG5,EMADDR    ;SET ERROR MSG ADDRESS
005512  013777  000556  172776          MOV     FMCNT,@FC       ;LOAD CHAR COUNT
005520  013777  000562  172766          MOV     @#WDATA,@BA     ;SET DATA ADDR
005526  112737  000060  000700          MOVB    #60,MTC1        ;SET WRITE OP COMMAND
005534  012737  005546  000670          MOV     #W1,RTRN        ;SET RETURN ADDRESS
005542  000137  020506                  JMP     TAPG            ;GO EXECUTE COMMAND
005546  032777  002000  172746  W1:     BIT     #2000,@DS       ;SEE IF EOT
005554  001412                          BEQ     1$              ;IF NOT AT EOT; BR
005556  005737  000666                  TST     EOTREC          ;BRANCH IF WRITTEN PAST EOT
005562  100407                          BMI     1$
005564  005300                          DEC     R0              ;ADJUST # OF RECORDS WRITTEN
005566  052700  100000                  BIS     #100000,R0      ;SET EOT INDICATOR
005572  010037  000666                  MOV     R0,EOTREC       ;SAVE RECORD COUNT
005576  012700  000002                  MOV     #2,R0           ;SET TO WRITE 1 LAST RECORD
005602  032777  010000  173006  1$:     BIT     #10000,@SWR     ;SEE IF SHOULD CHECK ERRORS
005610  001002                          BNE     2$              ;IF NOT; BR
005612  004737  016636                  JSR     PC,ERCHK        ;GO CHECK ERRORS
005616  013737  000604  000674  2$:     MOV     WSTAL,STAL      ;SET DELAY
005624  004737  011766                  JSR     PC,STALL        ;DELAY
```

```
2320  005630  005737  000720              TST     RTYFL           ;SEE IF RETRY TIME
2321  005634  001401                      BEQ     3$              ;IF NOT: BR
2322  005636  000207                      RTS     PC              ;ELSE RETURN
2323  005640  005737  000714      3$:     TST     SERFL           ;SEE IF WRITE ERROR
2324  005644  001446                      BEQ     W5              ;IF NOT: BR
2325  005646  013704  000702              MOV     UNP,R4
2326  005652  005264  001074              INC     WTER1(R4)       ;BUMP WRITE ERROR
2327  005656  005037  000714              CLR     SERFL           ;CLEAR STATUS ERROR FLAG
2328  005662  032777  000020  172726      BIT     #20,@SWR        ;SEE IF RETRY
2329  005670  001434                      BEQ     W5              ;IF NOT: BR
2330  005672  013703  000730              MOV     ERSAV,R3
2331  005676  042703  102720              BIC     #102720,R3      ;MASK UNRECOVERABLE ERROR
2332  005702  001407                      BEQ     W4              ;IF SO: BR
2333  005704  004737  022126              JSR     PC,PAPRT        ;PRINT HEADER
2334  005710  000004  025531              TYPE,MSG78              ;TYPE MSG
2335  005714  004737  011066              JSR     PC,NRTP         ;PRINT ER FOR NON-RETRYABLE
2336  005720  000420                      BR      W5
2337  005722  013704  000702      W4:     MOV     UNP,R4
2338  005726  005264  001054              INC     RTY1(R4)        ;BUMP RETRY CNTR
2339  005732  032777  002000  172656      BIT     #2000,@SWR      ;SEE IF PRINT ERRORS
2340  005740  001002                      BNE     W4A             ;IF NOT: BR
2341  005742  000004  025301              TYPE,MSG64              ;TYPE MSG
2342  005746  005037  000710      W4A:    CLR     RTCNT           ;CLEAR RETRY NUMBER
2343  005752  005037  000706              CLR     RPCNT           ;CLEAR REPEAT COUNTER
2344  005756  004737  006300              JSR     PC,WRTY         ;GO RETRY WRITE ERROR
2345  005762  005037  000720      W5:     CLR     RTYFL           ;CLEAR RETRY COUNTER
2346  005766  005300                      DEC     R0              ;SEE IF DONE ALL
2347  005770  001245                      BNE     W0              ;IF NOT: BR
2348  005772  005737  000572      W6:     TST     TMEX            ;SEE IF TM
2349  005776  001522                      BEQ     WEX             ;IF NOT: BR
2350  006000  005237  000704              INC     TMFLG           ;SET TM FLAG
2351  006004  012737  025212  000660  WTM:  MOV     #MSG54,EMADDR   ;POINT TO TM ERROR MSG
2352  006012  012737  000026  000700      MOV     #26,MTC1        ;SET TM OP CODE
2353  006020  005077  172472              CLR     @FC             ;LOAD FRAME COUNTER
2354  006024  013777  000562  172462      MOV     @#WDATA,@BA     ;LOAD BUS ADDRESS
2355  006032  012737  006044  000670      MOV     #WTM0,RTRN      ;SAVE RETURN ADDRESS
2356  006040  000137  020506              JMP     TAPG            ;WRITE TM
2357  006044  032777  010000  172544  WTM0:  BIT     #10000,@SWR     ;SEE IF SHOULD CHECK ERRORS
2358  006052  001074                      BNE     WEX
2359  006054  032777  000004  172440      BIT     #4,@DS          ;SEE IF TM STATUS
2360  006062  001011                      BNE     WTM1            ;IF SO: BR
2361  006064  013737  000562  020340      MOV     @#WDATA,CADER   ;SET EXPT BUS ADDRESS
2362  006072  012737  000001  020346      MOV     #1,DRVER        ;INDICATE ERROR
2363  006100  004737  017466              JSR     PC,ERPT         ;PRINT TM ERROR
2364  006104  000404                      BR      WTM2
2365  006106  013703  000562      WTM1:   MOV     @#WDATA,R3      ;SET EXPT ADDRESS
2366  006112  004737  016730              JSR     PC,ER2          ;GO CHECK FOR OTHER ERRORS
2367  006116  005737  000720      WTM2:   TST     RTYFL           ;SEE IF RETRY
2368  006122  001401                      BEQ     WTM3            ;IF NOT: BR
2369  006124  000207                      RTS     PC              ;ELSE RETURN TO RETRY ROUTINE
2370  006126  005737  000714      WTM3:   TST     SERFL           ;SEE IF WRITE ERROR
2371  006132  001444                      BEQ     WEX             ;IF NOT: BR
2372  006134  013704  000702              MOV     UNP,R4
2373  006140  005264  001074              INC     WTER1(R4)       ;BUMP WRITE ERROR
2374  006144  032777  000020  172444      BIT     #20,@SWR        ;SEE IF SHOULD RETRY
2375  006152  001434                      BEQ     WEX             ;IF NOT: BR
```

```
2376  006154  013703  000730                  MOV     ERSAV,R3
2377  006160  042703  102720                  BIC     #102720,R3      ;MASK UNRECOVERABLE ERROR
2378  006164  001407                          BEQ     WTM4            ;IF SO: BR
2379  006166  004737  022126                  JSR     PC,PAPRT        ;PRINT HEADER
2380  006172  000004  025531                  TYPE,MSG78              ;TYPE MSG
2381  006176  004737  011066                  JSR     PC,NRTP         ;PRINT ER FOR NON-RETRYABLE
2382  006202  000420                          BR      WEX
2383  006204  005037  000706          WTM4:   CLR     RPCNT           ;CLEAR REPEAT CNTR
2384  006210  013704  000702                  MOV     UNP,R4
2385  006214  005264  001054                  INC     RTY1(R4)        ;BUMP RETRY CNTR
2386  006220  005037  000710                  CLR     RTCNT           ;CLEAR RETRY CNTR
2387  006224  032777  002000  172364          BIT     #2000,@SWR      ;SEE IF PRINT ERRORS
2388  006232  001002                          BNE     WTM4A           ;IF NOT: BR
2389  006234  000004  025301                  TYPE,MSG64              ;TYPE MSG
2390  006240  004737  006300          WTM4A:  JSR     PC,WRTY         ;GO DO RETRY
2391  006244  005037  000720          WEX:    CLR     RTYFL           ;CLEAR RETRY FLAG
2392  006250  005037  000704                  CLR     TMFLG           ;CLEAR TAPE MARK FLAG
2393  006254  005737  000666                  TST     EOTREC          ;BRANCH IF NOT AT EOT
2394  006260  100006                          BPL     WRWX
2395  006262  032777  000014  172326  WRW:    BIT     #14,@SWR        ;BRANCH IF EITHER READ ENABLED
2396  006270  001002                          BNE     WRWX
2397  006272  000137  004266                  JMP     REOT            ;ELSE REWIND
2398  006276  000207                  WRWX:   RTS     PC              ;EXIT
```

```
2400                                    ;*************************
2401                                    ;WRITE ERROR RETRY
2402                                    ;
2403                                    ;*************************
2404
2405  006300  012737  000001  000720  WRTY:   MOV     #1,RTYFL        ;SET RETRY FLAG
2406  006306  004737  006666          WRTY0:  JSR     PC,WRTSB        ;GO SPACE REVERSE FOR REPEAT
2407  006312  005737  000704                  TST     TMFLG           ;SEE IF TAPE MARK TIME
2408  006316  001003                          BNE     WRTYTM          ;IF SO: BR
2409  006320  004737  005504                  JSR     PC,WO           ;REWRITE RECORD
2410  006324  000402                          BR      WRTYR           ;GO ON
2411  006326  004737  006004          WRTYTM: JSR     PC,WTM          ;GO WRITE TAPE MARK AGAIN
2412  006332  005737  000714          WRTYR:  TST     SERFL           ;REWRITE GOOD
2413  006336  001022                          BNE     WRTY2           ;IF NOT: BR
2414  006340  005237  000706                  INC     RPCNT           ;BUMP REPEAT COUNTER
2415  006344  022737  000004  000706        CMP     #4,RPCNT        ;SEE IF FOUR GOOD REPEATS
2416  006352  001355                          BNE     WRTY0           ;IF NOT: REPEAT
2417  006354  032777  002000  172234        BIT     #2000,@SWR      ;SEE IF PRINT
2418  006362  001007                          BNE     WRTY1           ;IF NOT: BR
2419  006364  000004  025714                  TYPE,MSG105             ;TYPE MSG
2420  006370  000004  025323                  TYPE,MSG65              ;TYPE MSG
2421  006374  013703  000710                  MOV     RTCNT,R3
2422  006400  104400                          TYPOCT                  ;PRINT RETRY NUMBER
2423  006402  000207          WRTY1:  RTS     PC              ;RESUME TESTING
2424  006404  013703  000730          WRTY2:  MOV     ERSAV,R3        ;GET ER
2425  006410  005037  000656                  CLR     TEMP3           ;CLEAR RECOVERABLE ERROR INDICATOR
2426  006414  042703  102720                  BIC     #102720,R3      ;MASK RECOVERABLE BITS
2427  006420  001412                          BEQ     WRTY2A          ;IF RECOVERABLE: BR
2428  006422  004737  022126                  JSR     PC,PAPRT        ;PRINT HEADER
2429  006426  000004  025531                  TYPE,MSG78              ;TYPE MSG
2430  006432  004737  011066                  JSR     PC,NRIP         ;PRINT ER
2431  006436  012737  000001  000656        MOV     #1,TEMP3        ;SET FLAG
2432  006444  000406                          BR      WRTY2B
2433  006446  032777  002000  172142  WRTY2A: BIT     #2000,@SWR      ;SEE IF PRINT
2434  006454  001022                          BNE     WRTY3           ;IF NOT: BR
2435  006456  000004  026124                  TYPE,MSG110             ;TYPE MSG
2436  006462  000004  025323          WRTY2B: TYPE,MSG65              ;TYPE MSG
2437  006466  013703  000710                  MOV     RTCNT,R3
2438  006472  104400                          TYPOCT                  ;PRINT RETRY NUMBER
2439  006474  000004  026146                  TYPE,MSG111             ;TYPE MSG
2440  006500  013703  000706                  MOV     RPCNT,R3
2441  006504  104400                          TYPOCT                  ;PRINT REPEAT NUMBER
2442  006506  005737  000656                  TST     TEMP3           ;SEE IF DID NON RECOVERABLE
2443  006512  001403                          BEQ     WRTY3           ;IF NOT: BR
2444  006514  005037  000656                  CLR     TEMP3           ;CLEAR FLAG
2445  006520  000207                          RTS     PC              ;EXIT
2446  006522  005737  000710          WRTY3:  TST     RTCNT           ;SEE IF FIRST RETRY
2447  006526  001004                          BNE     WRTY3A          ;IF NOT: BR
2448  006530  013704  000702                  MOV     UNP,R4
2449  006534  005364  001074                  DEC     WTER1(R4)       ;DECREMENT WRITE ERROR CNTR
2450  006540  013704  000702          WRTY3A: MOV     UNP,R4          ;GET UNIT NUMBER
2451  006544  016437  001034  000736        MOV     BTADDR(R4),BTPT ;GET ADDRESS OF UNIT BAD TAPE CNTR
2452  006552  017704  172160                  MOV     @BTPT,R4        ;GET COUNTER
2453  006556  005724                          TST     (R4)+           ;SET POINTER OFFSET
2454  006560  010477  172150                  MOV     R4,@BTPT
2455  006564  013703  000736                  MOV     BTPT,R3
```

```
2456  006570  060304                        ADD     R3,R4           ;SET ABSOLUTE POINTER
2457  006572  013714  000662                MOV     BLCNTR,(R4)     ;SET BLOCK NUMBER
2458  006576  062704  000040                ADD     #40,R4          ;ADD RCNT OFFSET
2459  006602  013714  000554                MOV     RCNT,(R4)
2460  006606  160014                        SUB     R0,(R4)         ;SET RECORD NUMBER
2461  006610  005214                        INC     (R4)            ;CORRECT RECORD NUMBER
2462  006612  022777  000040  172116        CMP     #40,@BTPT       ;SEE IF TOO MANY BAD SPOTS
2463  006620  001002                        BNE     WRTY4           ;IF NOT: BR
2464  006622  000137  007104                JMP     BTOV            ;ELSE GO TO BAD TAPE OVERFLOW
2465  006626  005237  000710        WRTY4:  INC     RTCNT           ;BUMP RETRY COUNTER
2466  006632  022737  000004  000710        CMP     #4,RTCNT        ;SEE IF DONE 4 RETRIES
2467  006640  001410                        BEQ     WRTY5           ;IF SO: BR
2468  006642  013704  000702                MOV     UNP,R4
2469  006646  005264  001054                INC     RTY1(R4)        ;BUMP RETRY COUNTER
2470  006652  005237  000740                INC     ERTFL           ;SET ERASE FLAG
2471  006656  000137  006306                JMP     WRTY0           ;DO NEXT RETRY
2472  006662  000137  007310        WRTY5:  JMP     BTUR            ;ELSE GO TO BAD TAPE UNRECOVERABLE
2473
2474                                        ;WRITE RETRY BACKSPACE-ERASE SUBROUTINE**********
2475
2476  006666  005037  000714        WRTSB:  CLR     SERFL           ;CLEAR FLAG
2477  006672  013737  000606  000674        MOV     TSTAL,STAL
2478  006700  004737  011766                JSR     PC,STALL        ;DO TURN AROUND DELAY
2479  006704  012737  025334  000660        MOV     #MSG66,EMADDR   ;SET ERROR CODE
2480  006712  012777  177777  171576        MOV     #-1,@#C         ;SET TO BACKSPACE 1 RECORD
2481  006720  013703  000564                MOV     @#RDATA,R3      ;SET EXPECTED BA
2482  006724  010377  171564                MOV     R3,@#BA
2483  006730  012737  000032  000700        MOV     #32,MTC1        ;SET BACK SPACE OP CODE
2484  006736  012737  006750  000670        MOV     #1$,RTRN        ;SET RETURN PC
2485  006744  000137  020506                JMP     TAPG            ;EXECUTE BACKSPACE COMMAND
2486  006750  004737  016730        1$:     JSR     PC,ER2          ;CHECK ERRORS
2487  006754  004737  011766                JSR     PC,STALL        ;STALL
2488  006760  005737  000714                TST     SERFL           ;SEE IF ERROR
2489  006764  001406                        BEQ     WRTSB1          ;IF NOT: BR
2490  006766  012737  000002  000732  WRTSB0: MOV     #2,BTFLG        ;SET FLAG
2491  006774  022626                        CMP     (SP)+,(SP)+     ;RESET STACK
2492  006776  000137  004266                JMP     REOT            ;GO REWIND AND REMOVE FROM TESTING
2493  007002  005737  000740        WRTSB1: TST     ERTFL           ;SEE IF SHOULD ERASE
2494  007006  001001                        BNE     WRTSB2          ;IF SO: BR
2495  007010  000207                        RTS     PC              ;RETURN
2496  007012  005037  000740        WRTSB2: CLR     ERTFL           ;CLEAR ERASE FLAG
2497  007016  005037  000706                CLR     RPCNT           ;CLEAR REPEAT CNTR
2498  007022  005037  000714                CLR     SERFL           ;CLEAR FLAG
2499  007026  012737  025346  000660        MOV     #MSG67,EMADDR   ;SET ERROR CODE
2500  007034  005077  171456                CLR     @#C             ;CLEAR FRAME COUNT
2501  007040  012737  000024  000700        MOV     #24,MTC1        ;SET ERASE OP-CODE
2502  007046  013703  000562                MOV     @#WDATA,R3      ;SET EXPECTED BA
2503  007052  010377  171436                MOV     R3,@#BA
2504  007056  012737  007070  000670        MOV     #1$,RTRN        ;SET RETURN ADDRESS
2505  007064  000137  020506                JMP     TAPG            ;GO ERASE
2506  007070  004737  016730        1$:     JSR     PC,ER2          ;GO CHECK ERRORS
2507  007074  005737  000714                TST     SERFL           ;SEE IF ERROR
2508  007100  001740                        BEQ     WRTSB1          ;IF NOT: BR
2509  007102  000731                        BR      WRTSB0
2510
2511                                        ;BAD TAPE OVERFLOW SUBROUTINE**********
```

```
007104  005037  000720           BTOV:   CLR     RTYFL           ;CLEAR RETRY FLAG
007110  012737  000001  000732           MOV     #1,BTFLG        ;SET BAD TAPE OVERFLOW FLAG
007116  005726                           TST     (SP)+           ;++B ADJUST STACK PTR
007120  000137  004266                   JMP     REOT            ;GO REWIND AND REMOVE FROM TESTING
007124  013701  000736           BTOV0:  MOV     BTPT,R1         ;SET TABLE POINTER
007130  005721                           TST     (R1)+
007132  005000                           CLR     R0
007134  010003                   BTOV1:  MOV     R0,R3
007136  000241                           CLC
007140  006003                           ROR     R3              ;R3=R3/2 FOR CORRECT NUMBER
007142  104400                           TYPOCT                  ;PRINT ENTRY NUMBER
007144  000004  024010                   TYPE,MSG13+1
                                                                         ;TYPE MSG
007150  011103                           MOV     (R1),R3
007152  104400                           TYPOCT                  ;PRINT BLOCK NUMBER
007154  000004  024015                   TYPE,MSG14              ;TYPE MSG
007160  062701  000040                   ADD     #40,R1          ;SET POINTER OFFSET FOR RECOED NUMBER
007164  012103                           MOV     (R1)+,R3
007166  104400                           TYPOCT                  ;PRINT RECORD NUMBER
007170  162701  000040                   SUB     #40,R1          ;RESET POINTER FOR BLOCK NUMBER
007174  005720                           TST     (R0)+
007176  020077  171534                   CMP     R0,@BTPT        ;SEE IF DONE
007202  001403                           BEQ     BTOV2           ;IF SO: BR
007204  000004  024355                   TYPE,MSG28              ;TYPE '<CR><LF>'
007210  000751                           BR      BTOV1           ;CONTINUE
007212  005737  000734           BTOV2:  TST     BTSTF           ;SEE IF STAT ONLY PRINT
007216  001007                           BNE     BTOVX           ;IF SO: BR
007220  012703  000041                   MOV     #41,R3          ;SET SIZE OF TABLE
007224  013704  000736                   MOV     BTPT,R4         ;SET POINTER
007230  005024                   BTOV3:  CLR     (R4)+           ;CLEAR TABLE
007232  005303                           DEC     R3              ;SEE IF DONE
007234  001375                           BNE     BTOV3           ;IF NOT: BR
007236  000207                   BTOVX:  RTS     PC              ;RETURN
```

```
2547
2548                                        ;BAD TAPE STATISTIC PRINT********
2549
2550  007240  000004  024355        BTPRT:  TYPE,MSG28              ;TYPE '<CR><LF>'
2551  007244  013704  000702                MOV     UNP,R4
2552  007250  016437  001034  000736      MOV     BTADDR(R4),BTPT ;SET TABLE POINTER
2553  007256  017703  171454              MOV     @BTPT,R3
2554  007262  000241                       CLC
2555  007264  006003                       ROR     R3              ;CORRECT NUMBER
2556  007266  104400                       TYPOCT                  ;PRINT NUMBER OF BAD SPOTS
2557  007270  000004  026160              TYPE,MSG112             ;TYPE MSG
2558  007274  005777  171436              TST     @BTPT           ;SEE IF ANY BAD SPOTS
2559  007300  001001                       BNE     BTPRT1          ;IF SO: BR
2560  007302  000207                       RTS     PC              ;ELSE RETURN
2561  007304  000137  007124        BTPRT1: JMP     BTOVO           ;PRINT STATS
2562
2563                                        ;BAD TAPE UNRECOVERABLE SUBROUTINE**********
2564
2565  007310  004737  022126        BTUR:   JSR     PC,PAPRT        ;PRINT HEADER
2566  007314  000004  026013              TYPE,MSG107             ;TYPE MSG
2567  007320  000207                       RTS     PC              ;RESUME TESTING
2568
```

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

007322  005037  000570        RSEQ:   CLR     RDCMD
007326  017704  171264                MOV     @SWR,R4         ;READ SWITCHES
007332  042704  177763                BIC     #177763,R4      ;MASK READ BITS & SEE IF BOTH READS
007336  001004                        BNE     RSR             ;IF NOT: BR
007340  032777  000002  171250        BIT     #2,@SWR         ;SEE IF READ REVERSE FIRST
007346  001041                        BNE     RSFR            ;IF NOT: BR
007350  032777  000004  171240  RSR:  BIT     #4,@SWR         ;SEE IF SHOULD READ REVERSE
007356  001005                        BNE     RSF             ;IF NOT: BR
007360  012737  000001  000570        MOV     #1,RDCMD        ;LOAD READ REVERSE COMMAND
007366  004737  007576                JSR     PC,READ         ;GO READ REVERSE
007372  032777  000010  171216  RSF:  BIT     #10,@SWR        ;SEE IF SHOULD READ FORWARD
007400  001066                        BNE     RSEX            ;IF NOT: BR
007402  005737  000570                TST     RDCMD   ;SEE IF HAVE READ REVERSE
007406  001406                        BEQ     RSF0            ;IF NOT: BR
007410  013737  000606  000674        MOV     TSTAL,STAL
007416  004737  011766                JSR     PC,STALL        ;DO READ STALL
007422  000406                        BR      RSF1
007424  032777  000001  171164  RSF0: BIT     #1,@SWR         ;SEE IF WRITE
007432  001002                        BNE     RSF1            ;IF NOT: BR
007434  004737  011514                JSR     PC,BKSP         ;GO BACKSPACE
007440  005037  000570        RSF1:   CLR     RDCMD           ;LOAD READ FORWARD COMMAND
007444  004737  007576                JSR     PC,READ         ;GO READ
007450  000442                        BR      RSEX            ;GO TO EXIT

007452  012737  000001  000570  RSFR: MOV     #1,RDCMD
007460  032777  000010  171130        BIT     #10,@SWR        ;SEE IF SHOULD READ FORWARD
007466  001012                        BNE     RSFR1           ;IF NOT: BR
007470  032777  000001  171120        BIT     #1,@SWR         ;SEE IF WRITE
007476  001002                        BNE     RSFR0           ;IF NOT: BR
007500  004737  011514                JSR     PC,BKSP         ;GO BACKSPACE TO START
007504  005037  000570        RSFR0:  CLR     RDCMD           ;LOAD READ FORWARD COMMAND
007510  004737  007576                JSR     PC,READ         ;GO READ FORWARD
007514  032777  000004  171074  RSFR1: BIT     #4,@SWR         ;SEE IF SHOULD READ REVERSE
007522  001015                        BNE     RSEX            ;IF NOT: BR
007524  005737  000570                TST     RDCMD
007530  001005                        BNE     RSFR2           ;IF READ REVERSE: BR
007532  013737  000606  000674        MOV     TSTAL,STAL      ;DO READ STALL
007540  004737  011766                JSR     PC,STALL
007544  012737  000001  000570  RSFR2: MOV     #1,RDCMD        ;LOAD READ REVERSE
007552  004737  007576                JSR     PC,READ         ;GO READ REVERSE
```

SEQ 0058

```
2626  007556  005037  000570          RSEX:   CLR     RDCMD
2627  007562  005737  000666                  TST     EOTREC          ;BRANCH IF EOT NOT REACHED
2628  007566  001402                          BEQ     RSFRX
2629  007570  000137  004266                  JMP     REOT            ;REWIND AND REPORT STATS
2630  007574  000207          RSFRX:  RTS     PC              ;EXIT
2631
```

```
2633                                    ;*************************************************************
2634                                    ;READ ROUTINE:
2635                                    ;
2636                                    ;THIS ROUTINE PERFORMS THE READ OPERATION DETERMINED
2637                                    ;BY THE READ SEQUENCE ROUTINE ONE RECORD AT A TIME.
2638                                    ;AT THE END OF EACH READ OPERATION THE STATUS REGISTER
2639                                    ;IS SCANNED FOR EITHER END OF TAPE OR BEGINNING OF TAPE.
2640                                    ;IF EOT WAS REACHED, CONTROL WILL BE PASSED TO
2641                                    ;THE EOT SUBROUTINE TO REWIND THE UNIT AND FLAG IT
2642                                    ;UNAVAILABLE UNTIL ALL UNITS HAVE REACHED EOT.
2643                                    ;IF BOT WAS REACHED AN ERROR IS PRINTED AND THE
2644                                    ;PROGRAM WILL HALT.  TESTING MAY BE RESUMED BY PRESSING
2645                                    ;THE CONTINUE SWITCH.
2646                                    ;IF A TAPE MARK IS EXPECTED (TM=1) THEN THE
2647                                    ;READ ROUTINE EXPECTS THE FIRST RECORD OF A
2648                                    ;READ REVERSE TO BE A TM, AND THE LAST RECORD
2649                                    ;OF A READ FORWARD TO BE A TM.  REMEMBER
2650                                    ;THAT THE TM ADDS ONE (1) TO THE TOTAL NUMBER
2651                                    ;OF RECORDS IN A BLOCK.
2652                                    ;CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13) DETERMINE WHETHER
2653                                    ;OR NOT TO CHECK FOR STATUS ERRORS (11) OR DATA ERRORS (13).
2654                                    ;CONSOLE SWITCH FIVE (5) IS USED TO CAUSE A CONTINUOUS
2655                                    ;READ AND SPACE (FORWARD OR REVERSE) OF THE CURRENT
2656                                    ;RECORD ON TAPE (YOZZLE).
2657                                    ;*************************************************************
2658
2659  007576  013700  000554           READ:   MOV     RCNT,R0         ;LOAD REC CNTR
2660  007602  005737  000666                   TST     EOTREC          ;SEE IF EOT
2661  007606  100012                           BPL     RDA             ;IF NOT: BR
2662  007610  005737  000570                   TST     RDCMD           ;SEE IF READ FORWARD
2663  007614  001407                           BEQ     RDA             ;IF SO: BR
2664  007616  042737  100000  000666           BIC     #100000,EOTREC  ;CLEAR FLAG
2665  007624  013703  000666                   MOV     EOTREC,R3       ;GET MODIFIED RECORD COUNT
2666  007630  160300                           SUB     R3,R0           ;SET RECORD AT
2667  007632  005200                           INC     R0              ;SET TO PROPER NUMBER OF RECORDS
2668  007634  012737  023753  000660  RDA:    MOV     #MSG6,EMADDR    ;SET ERROR MSG ADDRESS
2669  007642  005037  000704                   CLR     TMFLG
2670  007646  005737  000570                   TST     RDCMD
2671  007652  001406                           BEQ     RD0             ;IF READ FORWARD: BR
2672  007654  005737  000572                   TST     TMEX            ;SEE IF TM
2673  007660  001403                           BEQ     RD0             ;IF NOT: BR
2674  007662  005237  000704                   INC     TMFLG           ;SET TM FLAG
2675  007666  005200                           INC     R0
2676  007670  013777  000556  170620  RD0:    MOV     FMCNT,@#C       ;LOAD CHAR CNTR
2677  007676  013777  000564  170610          MOV     @#RDATA,@BA     ;LOAD DATA ADDR
2678  007704  005737  000570                   TST     RDCMD           ;SEE IF READ REVERSE
2679  007710  001417                           BEQ     RD1A            ;IF NOT: BR
2680  007712  013703  000556                   MOV     FMCNT,R3
2681  007716  005103                           COM     R3
2682  007720  032737  000020  000552           BIT     #20,UDES        ;SEE IF CORE DUMP
2683  007726  001402                           BEQ     RD1             ;IF NOT: BR
2684  007730  000241                           CLC
2685  007732  006003                           ROR     R3              ;R3 = FC/2
2686  007734  060377  170554           RD1:    ADD     R3,@BA          ;SET REVERSE BUS ADDRESS
2687  007740  012737  000076  000700           MOV     #76,MTC1        ;SET READ REVERSE
2688  007746  000403                           BR      RD1B
```

```
2689  007750  012737  000070  000700  RD1A:   MOV     #70,MIC1        ;SET READ FORWARD
2690  007756  012737  007770  000670  RD1B:   MOV     #RD2,RTRN       ;SET INTERRUPT RETURN ADDRESS
2691  007764  000137  020506                  JMP     TAPG            ;GO EXECUTE TAPE COMMAND
2692  007770  005737  000570          RD2:    TST     RDCMD           ;IGNORE EOT IF READ REVERSE
2693  007774  001014                  BNE     RD3
2694  007776  032777  002000  170516          BIT     #2000,@DS       ;SEE IF EOT
2695  010004  001410                  BEQ     RD3             ;IF NOT: BR
2696  010006  005737  000704                  TST     TMFLG           ;SEE IF TM
2697  010012  001005                  BNE     RD3             ;IF SO: BR
2698  010014  010037  000666                  MOV     R0,EOTREC       ;GET # OF RECORDS LEFT IN BLOCK TO READ
2699  010020  052737  100000  000666          BIS     #100000,EOTREC  ;SET EOT FLAG
2700  010026  032777  000002  170466  RD3:    BIT     #2,@DS          ;SEE IF AT LOAD POINT
2701  010034  001407                  BEQ     RD4             ;IF NOT: BR
2702  010036  004737  022126                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
2703  010042  000004  024113                  TYPE,MSG22              ;TYPE MSG
2704  010046  000000                  HALT
2705  010050  000137  003256                  JMP     STARTA          ;RESTART
2706  010054  032777  004000  170534  RD4:    BIT     #4000,@SWR      ;SEE IF SHOULD CHECK ERRORS
2707  010062  001116                  BNE     RD5             ;IF NOT: BR
2708  010064  005737  000704                  TST     TMFLG
2709  010070  001470                  BEQ     RD4B            ;IF NO TM EXPT: BR
2710  010072  032777  000004  170422          BIT     #4,@DS
2711  010100  001023                  BNE     RD4A            ;IF TM RECVD: BR
2712  010102  013737  000564  020340          MOV     @#RDATA,CADER   ;SAVE EXPT BUS ADDRESS
2713  010110  012737  000002  020346          MOV     #2,DRVER        ;SET TM STATUS ERROR FLAG
2714  010116  004737  017466                  JSR     PC,ERPT         ;GO PRINT TM ERROR
2715  010122  013704  000702                  MOV     UNP,R4
2716  010126  005737  000570                  TST     RDCMD           ;SEE IF READ REVERSE
2717  010132  001403                  BEQ     1$              ;IF NOT: BR
2718  010134  005264  001154                  INC     RDERR1(R4)      ;BUMP READ REVERSE ERROR
2719  010140  000500                  BR      RD6
2720  010142  005264  001114          1$:     INC     RDER1(R4)       ;BUMP READ FORWARD ERROR
2721  010146  000475                  BR      RD6
2722  010150  013703  000564          RD4A:   MOV     @#RDATA,R3
2723  010154  005737  000570                  TST     RDCMD           ;SEE IF READ REVERSE
2724  010160  001007                  BNE     RD4A0           ;IF SO: BR
2725  010162  032737  002000  000552          BIT     #2000,UDES      ;SEE IF IN PE
2726  010170  001025                  BNE     RD4A2           ;IF SO: BR
2727  010172  062703  000002                  ADD     #2,R3
2728  010176  000422                  BR      RD4A2
2729  010200  013704  000556          RD4A0:  MOV     FMCNT,R4
2730  010204  005104                  COM     R4
2731  010206  032737  000020  000552          BIT     #20,UDES        ;SEE IF CORE DUMP
2732  010214  001402                  BEQ     RD4A1           ;IF NOT: BR
2733  010216  000241                  CLC
2734  010220  006004                  ROR     R4              ;SET TO FC/2
2735  010222  060403          RD4A1:  ADD     R4,R3           ;SET EXPT BUS ADDRESS
2736  010224  042703  000001                  BIC     #1,R3           ;MAKE EXPT ADDRESS EVEN
2737  010230  032737  002000  000552          BIT     #2000,UDES      ;SEE IF IN PE
2738  010236  001002                  BNE     RD4A2           ;IF SO: BR
2739  010240  162703  000002                  SUB     #2,R3
2740  010244  004737  016730          RD4A2:  JSR     PC,ER2
2741  010250  000402                  BR      RD4C
2742  010252  004737  016636          RD4B:   JSR     PC,ERCHK        ;GO CHECK ERRORS
2743  010256  005737  000714          RD4C:   TST     SERFL
2744  010262  001416                  BEQ     RD5             ;IF NO ERROR: BR
```

```
2745  010264  013704  000702                  MOV     UNP,R4
2746  010270  005737  000570                  TST     RDCMD           ;SEE IF READ REVERSE
2747  010274  001003                          BNE     RD4D            ;IF SO: BR
2748  010276  005264  001114                  INC     RDER1(R4)       ;BUMP READ FORWARD ERROR
2749  010302  000402                          BR      RD4E
2750  010304  005264  001154          RD4D:   INC     RDERR1(R4)      ;BUMP READ REVERSE ERROR
2751  010310  004737  010506          RD4E:   JSR     PC,RDRTY        ;GO RETRY
2752  010314  005037  000720                  CLR     RTYFL           ;CLEAR RETRY FLAG
2753  010320  032777  020000  170270  RD5:    BIT     #20000,@SWR     ;SEE IF SHOULD DO DATA CHECK
2754  010326  001005                          BNE     RD6             ;IF NOT; BR
2755  010330  005737  000704                  TST     TMFLG
2756  010334  001002                          BNE     RD6
2757  010336  004737  015064                  JSR     PC,DCHK         ;GO CHECK DATA
2758  010342  005037  000714          RD6:    CLR     SERFL           ;CLEAR STATUS ERROR FLAG
2759  010346  004737  013730                  JSR     PC,DS3          ;CLEAR BUFFER
2760  010352  032777  000040  170236          BIT     #40,@SWR        ;SEE IF SHOULD YOZZLE
2761  010360  001402                          BEQ     RD7             ;IF NOT: BR
2762  010362  004737  011102                  JSR     PC,YOZ          ;ELSE GO YOZZLE
2763  010366  013737  000602  000674  RD7:    MOV     RSTAL,STAL      ;SET DELAY
2764  010374  004737  011766                  JSR     PC,STALL        ;STALL
2765  010400  005737  000570                  TST     RDCMD           ;SEE IF READ REVERSE
2766  010404  001403                          BEQ     RD7A            ;IF NOT: BR
2767  010406  005037  000704                  CLR     TMFLG           ;CLEAR TAPE MARK FLAG
2768  010412  000405                          BR      RD10
2769  010414  005737  000666          RD7A:   TST     EOTREC          ;SEE IF EOT FOUND
2770  010420  100002                          BPL     RD10            ;IF NOT; BR
2771  010422  012700  000001                  MOV     #1,R0           ;SET TO EOT
2772  010426  005300                  RD10:   DEC     R0
2773  010430  001402                          BEQ     RD11            ;IF DONE ALL: BR
2774  010432  000137  007670                  JMP     RD0
2775  010436  005737  000570          RD11:   TST     RDCMD           ;SEE IF READ REVERSE
2776  010442  001016                          BNE     RDEX            ;IF SO: BR
2777  010444  005737  000666                  TST     EOTREC          ;SEE IF FOUND EOT
2778  010450  100413                          BMI     RDEX            ;IF SO: BR
2779  010452  005737  000572                  TST     TMEX            ;SEE IF TM EXPECTED
2780  010456  001410                          BEQ     RDEX            ;IF NOT: BR
2781  010460  005737  000704                  TST     TMFLG           ;SEE IF TM FOUND
2782  010464  001005                          BNE     RDEX            ;IF SO: BR
2783  010466  005237  000704                  INC     TMFLG           ;ELSE SET FLAG
2784  010472  005200                          INC     R0              ;SET RECORD COUNT TO ONE
2785  010474  000137  007670                  JMP     RD0             ;GO READ TM
2786  010500  005037  000704          RDEX:   CLR     TMFLG
2787  010504  000207                  RDX:    RTS     PC              ;EXIT
```

```
2789                                    ;*****************************************************
2790                                    ;READ ERROR RETRY SUBROUTINE:
2791                                    ;
2792                                    ;THIS SUBROUTINE WILL RETRY ALL DATA RELATED
2793                                    ;READ ERRORS UP TO EIGHT (8) TIMES.  IF ALL
2794                                    ;FOUR RETRIES ARE BAD, IT IS CONSIDERED
2795                                    ;A HARD ERROR.  IF ANY ARE GOOD, IT IS A
2796                                    ;SOFT ERROR.  RETRIES MAY BE INHIBITED
2797                                    ;VIA SWITCH FOUR (SW4=0: INHIBIT RETRIES)
2798                                    ;*****************************************************
2799
2800  010506  032777  000020  170102  RDRTY:  BIT     #20,@SWR        ;SEE IF RETRY INHIBITED
2801  010514  001001                          BNE     RDRT0           ;IF NOT: BR
2802  010516  000207                          RTS     PC              ;ELSE RETURN
2803
2804  010520  013703  000730          RDRT0:  MOV     ERSAV,R3
2805  010524  022703  100000                  CMP     #100000,R3      ;++B BRANCH IF OTHER THAN CORRECTED READ ERROR
2806  010530  001011                          BNE     1$              ;++B
2807  010532  032777  000040  167762          BIT     #40,@DS         ;++B BRANCH IF NRZ
2808  010540  001405                          BEQ     1$              ;++B
2809  010542  005037  000714                  CLR     SERFL           ;++B CLEAR ERROR FLAG
2810  010546  000004  026504                  TYPE,MSG124             ;++B TYPE 'CORRECTED PE DATA ERROR'
2811  010552  000447                          BR      RDRT2           ;++B INC SOFT COUNTS
2812  010554  042703  102720          1$:     BIC     #102720,R3      ;MARK NON-RECOVERABLE ERROR BITS
2813  010560  001407                          BEQ     RDRT1           ;IF NOT: BR
2814  010562  004737  022126                  JSR     PC,PAPRT        ;PRINT HEADER
2815  010566  000004  025571                  TYPE,MSG79              ;TYPE MSG
2816  010572  004737  011066                  JSR     PC,NRTP         ;PRINT ER FOR NON-RETRYABLE ERROR
2817  010576  000207                          RTS     PC              ;RETURN
2818  010600  032777  002000  170010  RDRT1:  BIT     #2000,@SWR      ;SEE IF PRINT INHIBITED
2819  010606  001002                          BNE     RDRT1B          ;IF SO: BR
2820  010610  000004  025301                  TYPE,MSG64              ;TYPE MSG
2821  010614  005037  000710          RDRT1B: CLR     RTCNT           ;CLEAR RETRY COUNTER
2822  010620  005037  000714          RDRTG:  CLR     SERFL           ;CLEAR STATUS ERROR FLAG
2823  010624  012737  000002  000720          MOV     #2,RTYFL        ;SET READ RETRY FLAG
2824  010632  004737  011102                  JSR     PC,YOZ          ;GO TO YOZZLE TO RETRY READ
2825  010636  005737  000714                  TST     SERFL           ;SEE IF RETRY ERROR
2826  010642  001026                          BNE     RDRT5           ;IF SO: BR
2827  010644  032777  002000  167744          BIT     #2000,@SWR
2828  010652  001007                          BNE     RDRT2
2829  010654  000004  025714                  TYPE,MSG105             ;TYPE MSG
2830  010660  000004  025323                  TYPE,MSG65              ;TYPE MSG
2831  010664  013703  000710                  MOV     RTCNT,R3
2832  010670  104400                          TYPOCT                  ;PRINT RETRY NUMBER
2833  010672  013704  000700          RDRT2:  MOV     UNP,R4
2834  010676  005737  000570                  TST     RDCMD           ;SEE IF READ REVERSE
2835  010702  001003                          BNE     RDRT3           ;IF SO: BR
2836  010704  005264  002674                  INC     RFSOFT(R4)      ;ELSO BUMP FORWARD SOFT ERROR COUNTER
2837  010710  000402                          BR      RDRT4
2838  010712  005264  002714          RDRT3:  INC     RRSOFT(R4)      ;BUMP ERRORS SOFT CNTR
2839  010716  000207                  RDRT4:  RTS     PC              ;RETURN
2840  010720  013703  000730          RDRT5:  MOV     ERSAV,R3        ;GET ER
2841  010724  005037  000656                  CLR     TEMP3           ;CLEAR RECOVERABLE ERROR INDICATOR
2842  010730  042703  102720                  BIC     #102720,R3      ;MASK RECOVERABLE BITS
2843  010734  001417                          BEQ     RDRT5A          ;IF RECOVERABLE: BR
2844  010736  004737  022126                  JSR     PC,PAPRT        ;PRINT HEADER
```

```
2845  010742  000004  025571                    TYPE,MSG79              ;TYPE MSG
2846  010746  004737  011066                    JSR     PC,NRTP         ;PRINT ER
2847  010752  012737  000001  000656            MOV     #1,TEMP3        ;SET FLAG
2848  010760  000404                            BR      RDRT5B
2849  010762  032777  002000  167626  RDRT5A:   BIT     #2000,@SWR      ;SEE IF PRINT INHIBITED
2850  010770  001013                            BNE     RDRT6           ;IF SO: BR
2851  010772  000004  025323          RDRT5B:   TYPE,MSG65              ;TYPE MSG
2852  010776  013703  000710                    MOV     RTCNT,R3
2853  011002  104400                            TYPOCT                  ;PRINT RETRY NUMBER
2854  011004  005737  000656                    TST     TEMP3           ;SEE IF DID NON-RECOVERABLE
2855  011010  001403                            BEQ     RDRT6           ;IF NOT: BR
2856  011012  005037  000656                    CLR     TEMP3           ;CLEAR FLAG
2857  011016  000207                            RTS     PC              ;EXIT
2858  011020  005237  000710          RDRT6:    INC     RTCNT
2859  011024  023737  000710  000612            CMP     RTCNT,RETRY     ;SEE IF DONE 8 RETRIES
2860  011032  001272                            BNE     RDRTG           ;IF NOT: BR
2861  011034  000004  026223                    TYPE,MSG115             ;TYPE MSG
2862  011040  013704  000702                    MOV     UNP,R4
2863  011044  005737  000570                    TST     RDCMD           ;SEE IF READ REVERSE
2864  011050  001003                            BNE     RDRT7           ;IF SO: BR
2865  011052  005264  002734                    INC     RFHARD(R4)      ;BUMP FORWARD HARD ERROR CNTR
2866  011056  000402                            BR      RDRTX
2867  011060  005264  002754          RDRT7:    INC     RRHARD(R4)      ;BUMP REVERSE HARD ERROR CNTR
2868  011064  000207                  RDRTX:    RTS     PC              ;RETURN
2869
2870  011066  013703  000730          NRTP:     MOV     ERSAV,R3        ;GET ER REGISTER
2871  011072  104400                            TYPOCT                  ;PRINT ER
2872  011074  004737  020364                    JSR     PC,FRPRT        ;PRINT F OR R
2873  011100  000207                            RTS     PC              ;RETURN
2874                                            ;*******************************************************
2875                                            ;YOZZLE SUBROUTINE:
2876                                            ;
2877                                            ;THIS SUBROUTINE, ENTERED VIA SWITCH FIVE (5), IS USED TO PERFORM
2878                                            ;A CONTINUOUS READ AND SPACE OVER OF THE CURRENT RECORD ON TAPE.
2879                                            ;FULL STATUS AND DATA CHECKING MAY BE PERFORMED
2880                                            ;OR NOT VIA CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13).
2881                                            ;A SOFTWARE DELAY IS PERFORMED BETWEEN EACH READ
2882                                            ;AND SPACE OPERATION AND MAY BE VARIED BY TYPING
2883                                            ;CNTRL C ON THE TTY AND ENTERING A VALUE IN RESPONSE
2884                                            ;TO THE PRINTED REQUEST.
2885                                            ;*******************************************************
2886
2887  011102  013737  000610  000674  YOZ:     MOV     YSTAL,STAL
2888  011110  004737  011766                    JSR     PC,STALL        ;DO YOZZLE STALL
2889  011114  012737  177777  167374  YOZO:     MOV     #-1,@#C         ;SET TO 1 RECORD SPACING
2890  011122  005737  000570                    TST     RDCMD           ;SEE IF READ REVERSE
2891  011126  001404                            BEQ     YOZA            ;IF NOT: BR
2892  011130  112737  000030  000700            MOVB    #30,MTC1        ;SET TO SPACE FORWARD
2893  011136  000403                            BR      YOZB
2894  011140  112737  000032  000700  YOZA:     MOVB    #32,MTC1        ;SET TO SPACE REVERSE
2895  011146  012737  011166  000670  YOZB:     MOV     #YOZC,RTRN      ;SET RETURN ADDRESS
2896  011154  012737  177775  000674            MOV     #177775,STAL    ;SET TIME MULTIPLIER
2897  011162  000137  020506                    JMP     TAPG            ;GO YOZZLE
2898  011166  005737  000704          YOZC:     TST     TMFLG           ;SEE IF TM
2899  011172  001404                            BEQ     1$              ;IF NOT: BR
2900  011174  012737  040000  000674            MOV     #40000,STAL     ;SET TM STALL
```

```
2901  011202  000403                          BR      2$
2902  011204  013737  000610  000674  1$:     MOV     YSTAL,STAL
2903  011212  004737  011766          2$:     JSR     PC,STALL        ;DO YOZZLE STALL
2904  011216  013777  000564  167270          MOV     @#RDATA,@BA     ;SET BUS ADDRESS
2905  011224  005737  000570                  TST     RDCMD           ;SEE IF READ REVERSE
2906  011230  001416                          BEQ     YOZC1           ;IF NOT: BR
2907  011232  013703  000556                  MOV     FMCNT,R3
2908  011236  005103                          COM     R3
2909  011240  032737  000020  000552          BIT     #20,UDES        ;SEE IF CORE DUMP
2910  011246  001401                          BEQ     YOZC0           ;IF NOT: BR
2911  011250  006203                          ASR     R3              ;R3 = FC/2
2912  011252  060377  167236          YOZC0:  ADD     R3,@BA          ;SET REVERSE BUS ADDRESS
2913  011256  012737  000076  000700          MOV     #76,MTC1        ;SET READ REVERSE
2914  011264  000403                          BR      YOZC2
2915  011266  012737  000070  000700  YOZC1:  MOV     #70,MTC1        ;SET READ FORWARD
2916  011274  013777  000556  167214  YOZC2:  MOV     FMCNT,@FC       ;SET CHARACTER COUNT
2917  011302  012737  011314  000670          MOV     #YOZD,RTRN      ;SET RETURN ADDRESS
2918  011310  000137  020506                  JMP     TAPG            ;GO READ
2919  011314  032777  004000  167274  YOZD:   BIT     #4000,@SWR      ;SEE IF SHOULD CHECK ERRORS
2920  011322  001047                          BNE     YOZE            ;IF NOT: BR
2921  011324  005737  000704                  TST     TMFLG           ;SEE IF TAPE MARK TIME
2922  011330  001442                          BEQ     YOZD1           ;IF NOT: BR
2923  011332  005737  000570                  TST     RDCMD           ;SEE IF READ REVERSE
2924  011336  001425                          BEQ     YOZD0           ;IF NOT: BR
2925  011340  013703  000564                  MOV     @#RDATA,R3
2926  011344  013704  000556                  MOV     FMCNT,R4
2927  011350  005104                          COM     R4
2928  011352  032737  000020  000552          BIT     #20,UDES        ;SEE IF CORE DUMP
2929  011360  001401                          BEQ     YOZD4           ;IF NOT: BR
2930  011362  006204                          ASR     R4              ;SET TO FC/2
2931  011364  060403                  YOZD4:  ADD     R4,R3           ;SET EXPT BUS ADDRESS
2932  011366  042703  000001                  BIC     #1,R3           ;MAKE EXPT ADDRESS EVEN
2933  011372  032737  002000  000552          BIT     #2000,UDES      ;SEE IF PE
2934  011400  001001                          BNE     YOZD2           ;IF SO: BR
2935  011402  005743                          TST     -(R3)           ;SET EXPT BA
2936  011404  004737  016730          YOZD2:  JSR     PC,ER2          ;GO CHECK ERRORS
2937  011410  000430                          BR      YOZF
2938  011412  013703  000564          YOZD0:  MOV     @#RDATA,R3
2939  011416  032737  002000  000552          BIT     #2000,UDES      ;SEE IF PE
2940  011424  001001                          BNE     YOZD3           ;IF SO: BR
2941  011426  005723                          TST     (R3)+           ;SET EXPT BA
2942  011430  004737  016730          YOZD3:  JSR     PC,ER2          ;GO CHECK ERRORS
2943  011434  000416                          BR      YOZF
2944  011436  004737  016636          YOZD1:  JSR     PC,ERCHK        ;ELSE GO CHECK ERRORS
2945  011442  005737  000720          YOZE:   TST     RTYFL           ;SEE IF RETRY
2946  011446  001013                          BNE     YOZG            ;IF SO: BR
2947  011450  032777  020000  167140          BIT     #20000,@SWR     ;SEE IF SHOULD CHECK DATA
2948  011456  001005                          BNE     YOZF            ;IF NOT: BR
2949  011460  005737  000704                  TST     TMFLG           ;SEE IF TAPE MARK
2950  011464  001002                          BNE     YOZF            ;IF SO: BR
2951  011466  004737  015064                  JSR     PC,DCHK         ;ELSE GO CHECK DATA
2952  011472  004737  013730          YOZF:   JSR     PC,DS3          ;GO CLEAR DATA AREA
2953  011476  032777  000040  167112  YOZG:   BIT     #40,@SWR        ;SEE IF SHOULD CONTINUE YOZZLE
2954  011504  001402                          BEQ     YOZH            ;IF NOT: BR
2955  011506  000137  011114                  JMP     YOZO
2956  011512  000207                  YOZH:   RTS     PC              ;EXIT
```

```
                                        ;*********************************************************
                                        ;BACKSPACE SUBROUTINE:
                                        ;
                                        ;THIS SUBROUTINE IS USED TO PERFORM THE
                                        ;BACKSPACE OPERATION REQUIRED BY THE READ
                                        ;ROUTINE FOR READ FORWARD AFTER WRITING.
                                        ;IF A TAPE MARK IS EXPECTED (TM=1) THEN THE SPACE
                                        ;ROUTINE ASSUMES THAT THE TM WILL BE FIRST WHEN
                                        ;BACKSPACING.  THEREFORE TWO OPERATIONS ARE REQUIRED
                                        ;TO SPACE OVER A BLOCK.  FIRST SPACE OVER THE TM, THEN
                                        ;SPACE OVER THE DATA RECORDS.
                                        ;A CHECK FOR RECORD COUNT ZERO IS MADE AT THE
                                        ;END OF THE SPACE OPERATION TO ASSURE THAT PROPER
                                        ;TAPE POSITIONING WAS DONE.
                                        ;*********************************************************
011514  013737  000606  000674  BKSP:   MOV     TSTAL,STAL
011522  004737  011766                  JSR     PC,STALL        ;DO TURN AROUND STALL
011526  012737  024002  000660          MOV     #MSG10,EMADDR
011534  013703  000564                  MOV     @#RDATA,R3      ;SET EXPECTED BA
011540  010377  166750                  MOV     R3,@BA
011544  005737  000572                  TST     TMEX            ;SEE IF TM
011550  001436                          BEQ     B0              ;IF NOT: BR
011552  012777  177777  166736          MOV     #-1,@FC
011560  012737  000032  000700          MOV     #32,MTC1
011566  012737  011600  000670          MOV     #1$,RTRN
011574  000137  020506                  JMP     TAPG            ;SPACE TO TM
011600  032777  010000  167010  1$:     BIT     #10000,@SWR     ;SEE IF SHOULD CHECK ERROR
011606  001017                          BNE     B0              ;IF NOT: BR
011610  012737  025221  000660          MOV     #MSG55,EMADDR
011616  032777  000004  166676          BIT     #4,@DS          ;SEE IF TM
011624  001006                          BNE     2$              ;IF SO: BR
011626  013737  000564  020340          MOV     @#RDATA,CADER
011634  004737  017466                  JSR     PC,ERPT         ;PRINT ERROR
011640  000402                          BR      B0
011642  004737  016730          2$:     JSR     PC,ER2
011646  013700  000554          B0:     MOV     RCNT,R0
011652  005737  000666                  TST     EOTREC          ;BRANCH IF EOT NOT DETECTED
011656  100007                          BPL     1$
011660  042737  100000  000666          BIC     #100000,EOTREC  ;CLEAR EOT INDICATOR
011666  013703  000666                  MOV     EOTREC,R3       ;GET # OF RECORDS LEFT IN BLOCK
011672  160300                          SUB     R3,R0           ;FORM # OF RECORDS TO BACK SPACE
011674  005200                          INC     R0
011676  012737  024002  000660  1$:     MOV     #MSG10,EMADDR   ;SET ERROR MESG ADDRESS
011704  012737  011742  000670          MOV     #2$,RTRN        ;SET RETURN PC
011712  012777  177777  166576          MOV     #-1,@#FC        ;SET TO BACKSPACE 1 RECORD
011720  013703  000564                  MOV     @#RDATA,R3      ;SET EXPECTED BA
011724  010377  166564                  MOV     R3,@BA
011730  012737  000032  000700          MOV     #32,MTC1        ;SET SPACE REVERSE
011736  000137  020506                  JMP     TAPG            ;GO DO SPACE
011742  004737  016730          2$:     JSR     PC,ER2
011746  013737  000606  000674          MOV     TSTAL,STAL      ;DO STALL
011754  004737  011766                  JSR     PC,STALL        ;STALL
011760  005300                          DEC     R0              ;DECREMENT # OF RECORD TO BACKSPACE
011762  001345                          BNE     1$
011764  000207                          RTS     PC              ;EXIT
```

```
3015                                    ;*****************************************************
3016                                    ;STALL ROUTINE:
3017                                    ;
3018                                    ;THIS ROUTINE IS USED TO PROVIDE SOFTWARE DELAYS
3019                                    ;DURING READ, WRITE, TURN AROUND, AND YOZZLE.
3020                                    ;THE DELAY TIMES MAY BE SET BY THE OPERATOR AT
3021                                    ;INITIAL START FROM 200(8) OR MAY BE MODIFIED
3022                                    ;AT ANY TIME BY ENTERING CNTRL C ON THE TTY AND
3023                                    ;INSERTING NEW VALUES IN RESPONSE TO THE REQUEST.
3024                                    ;THE READ STALL AND THE WRITE STALL ARE DELAYS
3025                                    ;EXECUTED BETWEEN EACH RECORD OF THE DATA BLOCK.
3026                                    ;THE TURN AROUND STALL IS EXECUTED EACH TIME
3027                                    ;THE DIRECTION OF TAPE MOVEMENT IS CHANGED AND
3028                                    ;ALSO EACH TIME THE TAPE OPERATION CHANGES FROM
3029                                    ;WRITE TO READ OR READ TO WRITE. THE YOZZLE
3030                                    ;STALL IS EXECUTED ONLY DURING THE YOZZLE ROUTINE.
3031                                    ;*****************************************************
3032
3033  011766  005337  000674    STALL:  DEC     STAL
3034  011772  001375                    BNE     STALL           ;DELAY
3035  011774  000207                    RTS     PC              ;EXIT
```

```
3037
3038                                    ;*****************************************************
3039                                    ;RANDOM CHARACTER COUNT GENERATOR:
3040                                    ;
3041                                    ;THIS ROUTINE ENTERED VIA CONSOLE SWITCH
3042                                    ;SEVEN (7) IS USED TO GENERATE A RANDOM
3043                                    ;CHARACTER COUNT FOR EACH DATA BLOCK.
3044                                    ;ALL RECORDS WITHIN A GIVEN BLOCK WILL BE
3045                                    ;THE SAME, BUT EACH BLOCK WILL VARY.
3046                                    ;THE LIMITS ARE TWENTY (20) TO THE MAX BUFFER SIZE.
3047                                    ;*****************************************************
3048
3049  011776  012701  177760      CCNTR:  MOV     #-20,R1         ;SET HIGH LIMIT
3050  012002  013702  000560              MOV     BUFMAX,R2       ;SET LOW LIMIT
3051  012006  004737  022430              JSR     PC,RANG         ;GO GENERATE NUMBER
3052  012012  042737  000001  000636      BIC     #1,RANSAV
3053  012020  013737  000636  000556      MOV     RANSAV,FMCNT    ;SET CHAR COUNT
3054  012026  012737  177777  013766      MOV     #-1,PATS        ;PRESET DATA PATTERN
3055  012034  000207                      RTS     PC              ;EXIT
3056
3057                                    ;*****************************************************
3058                                    ;RANDOM RECORD COUNT GENERATOR:
3059                                    ;
3060                                    ;THIS ROUTINE ENTERED VIA CONSOLE SWITCH SIX (6)
3061                                    ;IS USED TO GENERATE A RANDOM NUMBER OF RECORDS
3062                                    ;FOR EACH BLOCK OF DATA.
3063                                    ;THE LIMITS ARE ONE (1) TO FIVE HUNDRED (500) OCTAL
3064                                    ;RECORDS PER BLOCK.
3065                                    ;*****************************************************
3066
3067  012036  012702  000001      RCNTR:  MOV     #1,R2           ;SET LOW LIMIT
3068  012042  012701  000500              MOV     #500,R1         ;SET HIGH LIMIT
3069  012046  004737  022430              JSR     PC,RANG         ;GO GENERATE NUMBER
3070  012052  013737  000636  000554      MOV     RANSAV,RCNT     ;SET RECORD COUNT
3071  012060  000207                      RTS     PC              ;EXIT
3072
3073
```

```
3075
3076
3077                    ;*****************************************************
3078                    ;TEST CONDITION ENTRY ROUTINE:
3079                    ;
3080                    ;THIS ROUTINE IS USED TO ALLOW THE OPERATOR
3081                    ;TO ENTER, AT THE TTY, THE NECESSARY PARAMETERS
3082                    ;TO RUN THE PROGRAM AS HE WISHES.  THE
3083                    ;ROUTINE IS ONLY ENTERED UPON INITIAL STARTING
3084                    ;FROM LOCATION 200(8).
3085                    ;THE MAIN PURPOSE OF THIS ROUTINE IS TO ESTABLISH
3086                    ;A TABLE OF DEVICES TO BE TESTED.  THIS TABLE
3087                    ;CONSISTS OF AN ENTRY FOR EACH OF ONE (1) TO
3088                    ;EIGHT (8) DEVICES.  EACH ENTRY CONTAINS THE
3089                    ;SLAVE NUMBER, DENSITY, PARITY, AND
3090                    ;FORMAT.  THE INFORMATION IS ENTERED
3091                    ;IN RESPONSE TO PRINTED REQUESTS AT THE TTY.
3092                    ;SLAVES MAY BE ENTERED IN ANY ORDER.  EACH
3093                    ;PARAMETER IS CHECKED FOR LEGALITY BEFORE BEING
3094                    ;SET INTO THE TABLE.
3095                    ;THE DRIVE NUMBER REQUEST WILL ALSO CHECK THE MASSBUS
3096                    ;FOR THE PRESENCE OF THE REQUESTED DRIVE.  IF IT IS NOT FOUND,
3097                    ;A NON-EXIST DRIVE MESSAGE WILL BE PRINTED AND ANOTHER DRIVE
3098                    ;REQUEST MADE.  WHEN THE DRIVE IS FOUND, THE RESPONSE IS STORED
3099                    ;AND CONTROL PASSED TO THE SLAVE SELECT ROUTINE.
3100                    ;THE SLAVE SELECT ROUTINE ALSO CHECKS FOR THE PRESENCE OF THE
3101                    ;SLAVE.  IF IT IS NOT PRESENT, A MESSAGE IS PRINTED AND ANOTHER
3102                    ;REQUEST IS ISSUED.  WHEN THE SELECTED SLAVE IS FOUND TO BE
3103                    ;PRESENT, A MESSAGE IS PRINTED IF IT IS A 7 CHANNEL DRIVE
3104                    ;TO ASSIST IN SELECTING DENSITY, PARITY, AND FORMAT.
3105                    ;UPON COMPLETION OF THE DEVICE TABLE, REQUESTS
3106                    ;ARE PRINTED FOR ENTRY OF THE NUMBER OF CHARACTERS
3107                    ;PER RECORD AND THE NUMBER OF RECORDS PER BLOCK.  THE
3108                    ;NEXT REQUEST IS FOR A PATTERN NUMBER TO BE USED
3109                    ;FOR WRITING AND CHECKING OF READ DATA.
3110                    ;FOLLOWING THE PATTERN REQUEST IS THE TAPE MARK OPTION.
3111                    ;RESPONDING TO THE REQUEST (TM=) WITH A ONE (1)
3112                    ;WILL CAUSE THE PROGRAM TO WRITE A TM AT THE
3113                    ;END OF EACH DATA BLOCK AND TO EXPECT THE
3114                    ;TM TO BE DETECTED IN EITHER READ FORWARD AND REVERSE
3115                    ;OR DURING SPACE OPERATION.  A RESPONSE OF ZERO (TM=0)
3116                    ;DISALLOWS WRITTING OF THE TM AND CAUSES THE READ
3117                    ;AND SPACE ROUTINES TO EXPECT NO TM TO BE PRESENT.
3118                    ;THE LAST REQUESTS ARE FOR ENTRY OF THE DESIRED
3119                    ;WRITE, READ, AND TURN AROUND STALLS.
3120                    ;*****************************************************
3121  012062  005737  000644          TINP:   TST     TINF            ;SEE IF SHOULD INPUT FROM TTY
3122  012066  001002                          BNE     1$              ;IF SO: BR
3123  012070  000137  013404                  JMP     TINP4           ;GET SWITCHES
3124  012074  005037  000702          1$:     CLR     UNP             ;CLEAR TABLE POINTER
3125  012100  005037  005042                  CLR     REOTC           ;CLEAR EOT UNIT COUNTER
3126  012104  012737  024431  012130          MOV     @MSG31,41$      ;GET TITLE MSG
3127  012112  005737  000742                  TST     ASEQF           ;SEE IF AUTO SEQ
3128  012116  001403                          BEQ     4$              ;IF NOT: BR
3129  012120  012737  024357  012130          MOV     @MSG30,41$      ;SET AUTO SEQ HDR
3130  012126  000004                  4$:     TYPE                    ;TYPE MSG
```

```
3131  012130  000000                  41$:    .WORD   0               ;ADDRESS OF APPROPRIATE TITLE MSG
3132  012132  105077  177772                  CLRB    @41$            ;DO NOT TYPE TITLE ON RESTART
3133  012136  000004  024513                  TYPE,MSG31A             ;TYPE INSTRUCTIONS
3134  012142  105037  024513                  CLRB    MSG31A          ;DO NOT TYPE STARTUP INSTRUCTIONS ON RESTART
3135  012146  005737  013560                  TST     SCVFL           ;SEE IF SHORT CONVERSATION
3136  012152  001065                          BNE     6$              ;IF SO: BR
3137  012154  000004  025445                  TYPE,MSG74              ;REQUEST REGISTER START
3138  012160  013703  000544                  MOV     REGS,R3
3139  012164  104400                          TYPOCT                  ;PRINT CURRENT REG START
3140  012166  012705  000544                  MOV     #REGS,R5        ;SAVE ADDRESS LOCATION
3141  012172  012701  000007                  MOV     #7,R1           ;SET SIZE OF ENTRY
3142  012176  012702  176400                  MOV     #176400,R2      ;SET UPPER LIMIT
3143  012202  012703  172300                  MOV     #172300,R3      ;SET LOWER LIMIT
3144  012206  004737  022612                  JSR     PC,TTR          ;GO GET RESPONSE
3145
3146  012212  000004  025470                  TYPE,MSG75              ;REQUEST INTERRUPT VECTOR ADDRESS
3147  012216  013703  000546                  MOV     VECT,R3
3148  012222  104400                          TYPOCT                  ;PRINT CURRENT VECTOR
3149  012224  012705  000546                  MOV     #VECT,R5        ;SET SAVE LOCATION
3150  012230  012701  000004                  MOV     #4,R1           ;SET SIZE OF ENTRY
3151  012234  012702  000224                  MOV     #224,R2         ;SET UPPER LIMIT
3152  012240  012703  000150                  MOV     #150,R3         ;SET LOWER LIMIT
3153  012244  004737  022612                  JSR     PC,TTR          ;GO GET RESPONSE
3154  012250  013700  000546                  MOV     VECT,R0         ;GET VECTOR ADDRESS
3155  012254  012720  021274                  MOV     #MTINT,(R0)+    ;LOAD VECTOR WITH HANDLER ADDRESS
3156  012260  012710  000340                  MOV     #340,(R0)       ;LOAD PRIORITY LEVEL
3157  012264  013700  000544                  MOV     REGS,R0         ;GET STARTING REGISTER ADDRESS
3158  012270  012701  000016                  MOV     #16,R1          ;SET NUMBER OF REGISTERS
3159  012274  012702  000510                  MOV     #C1,R2          ;GET FIRST ADDRESS LOCATION
3160  012300  010022                  5$:     MOV     R0,(R2)+        ;BUILD TABLE OF ADDRESSES
3161  012302  062700  000002                  ADD     #2,R0           ;BUMP ADDRESS
3162  012306  005301                          DEC     R1              ;SEE IF DONE
3163  012310  001373                          BNE     5$              ;IF NOT: BR
3164  012312  005737  000742                  TST     ASEQF           ;SEE IF AUTO SEQ
3165  012316  001403                          BEQ     6$              ;IF NOT: BR
3166  012320  005726                          TST     (SP)+           ;RESET STACK POINTER
3167  012322  000137  021312                  JMP     ASEQ            ;GO TO AUTO SEQUENCE
3168
3169  012326  012777  000040  166164  6$:     MOV     #40,@CS         ;INITIALIZE
3170  012334  000004  025156                  TYPE,MSG52A             ;REQUEST DRIVE (TM03) #
3171  012340  012705  000550                  MOV     #DVN,R5         ;GET ADDRESS
3172  012344  012701  000002                  MOV     #2,R1           ;SET SIZE OF RESPONSE
3173  012350  012702  000007                  MOV     #7,R2           ;SET UPPER LIMIT
3174  012354  012703  000000                  MOV     #0,R3           ;SET LOWER LIMIT
3175  012360  004737  022612                  JSR     PC,TTR          ;GO GET DRIVE NUMBER
3176  012364  013777  000550  166126          MOV     DVN,@CS
3177  012372  005777  166112                  TST     @C1             ;ACCESS DRIVE
3178  012376  032777  010000  166114          BIT     #10000,@CS      ;SEE IF NED
3179  012404  001403                          BEQ     TINP0           ;IF NOT: BR
3180  012406  000004  025402                  TYPE,MSG71              ;TYPE 'NON-EXISTANT DRIVE'
3181  012412  000745                          BR      6$              ;RETRY DVN
3182
3183  012414  012705  000654          TINP0:  MOV     #TEMP2,R5       ;SET ADDRESS FOR RESPONSE
3184  012420  000004  024600                  TYPE,MSG32              ;REQUEST SLAVE (TE16,TU77) #
3185  012424  005037  000654                  CLR     TEMP2           ;CLEAR BUFFER
3186  012430  012701  000002                  MOV     #2,R1           ;SET NUMBER OF CHARACTERS TO INPUT
```

```
3187  012434  012702  000007                  MOV     #7,R2           ;SET MAXIMUM LIMIT
3188  012440  012703  000000                  MOV     #0,R3           ;SET MINIMUM LIMIT
3189  012444  004737  022612                  JSR     PC,TTR          ;GO GET UNIT NUMBER
3190  012450  005737  000652                  TST     TEMP1           ;SEE IF HAVE NEW PARAMETER
3191  012454  001010                          BNE     TINP0B          ;IF SO: BR
3192  012456  013700  000702                  MOV     UNP,R0
3193  012462  001754                          BEQ     TINP0           ;BRANCH IF FIRST ENTRY
3194  012464  012760  177777  000752          MOV     #-1,UN1(R0)     ;SET END UNIT TABLE
3195  012472  000137  013012                  JMP     TINP2C          ;GO GET RECORD COUNT
3196  012476  013700  000702          TINP0B: MOV     UNP,R0
3197  012502  011560  000752                  MOV     (R5),UN1(R0)    ;SET NEW SLAVE #
3198  012506  012777  000040  166004          MOV     #40,@CS         ;DO A MASS BUS CLEAR
3199  012514  013777  000550  165776          MOV     DVN,@CS         ;LOAD DRIVE #
3200  012522  016077  000752  166012          MOV     UN1(R0),@TC     ;LOAD SLAVE NUMBER
3201  012530  032777  002000  166000          BIT     #2000,@DT       ;SEE  IF SLAVE PRESENT
3202  012536  001003                          BNE     TINP0D          ;IF SO: BR
3203  012540  000004  025234                  TYPE,MSG57              ;TYPE NON-EXISTANT SLAVE'
3204  012544  000723                          BR      TINP0           ;REDO
3205  012546  017703  165764          TINP0D: MOV     @DT,R3          ;GET CONTENTS OF DT REG
3206  012552  042703  000007                  BIC     #7,R3           ;CLEAR DRIVE TYPE #
3207  012556  022703  142050                  CMP     #142050,R3      ;SEE IF 9TRK TM03
3208  012562  001407                          BEQ     TINP0E          ;IF SO: BR
3209  012564  000004  025127                  TYPE,MSG50              ;TYPE 'ILLEGAL DRIVE TYPE'
3210  012570  017703  165742                  MOV     @DT,R3
3211  012574  042703  000007                  BIC     #7,R3           ;CLEAR SLAVE #
3212  012600  104400                          TYPOCT                  ;PRINT DRIVE TYPE REGISTER
3213  012602  004737  023562          TINP0E: JSR     PC,SNPT         ;PRINT SERIAL NUMBER
3214
3215  012606  000004  024613          TINP1:  TYPE,MSG33              ;REQUEST DENSITY
3216  012612  005037  000654                  CLR     TEMP2           ;CLEAR BUFFER
3217  012616  012701  000002                  MOV     #2,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3218  012622  012702  000004                  MOV     #4,R2           ;SET MAXIMUM LIMIT
3219  012626  012703  000003                  MOV     #3,R3           ;SET MINIMUM LIMIT
3220  012632  004737  022612                  JSR     PC,TTR          ;GO GET DENSITY
3221  012636  012703  000010                  MOV     #10,R3          ;SET POSITION FACTOR
3222  012642  004737  013562                  JSR     PC,TPOS         ;GO LOAD DENSITY INTO PROPER POSITION
3223
3224  012646  000315          TINP2:  SWAB    (R5)                    ;IF DENSITY
3225  012650  022715  000004                  CMP     #4,(R5)         ;IS 1600BPI
3226  012654  001415                          BEQ     1$              ;THEN SKIP PARITY REQUEST
3227  012656  000004  024626                  TYPE,MSG34              ;REQUEST PARITY
3228  012662  005037  000654                  CLR     TEMP2           ;CLR BFR
3229  012666  012701  000002                  MOV     #2,R1           ;SET NUMBER OF CHAR. TO INPUT
3230  012672  012702  000001                  MOV     #1,R2           ;SET HIGH LIMIT
3231  012676  012703  000000                  MOV     #0,R3           ;SET LOW LIMIT
3232  012702  004737  022612                  JSR     PC,TTR          ;GO INPUT PARITY
3233  012706  000402                          BR      2$              ;SKIP 1600 BPI PAROTY SETTING
3234  012710  012715  000000          1$:     MOV     #0,(R5)         ;SET ODD PARITY FOR 1600 BPI
3235  012714  012703  000003          2$:     MOV     #3,R3           ;SET POSITION FACTOR
3236  012720  004737  013562                  JSR     PC,TPOS         ;GO POSITION PARITY
3237
3238  012724  000004  025200          TINP2A: TYPE,MSG53              ;REQUEST FORMAT
3239  012730  005037  000654                  CLR     TEMP2
3240  012734  012701  000003                  MOV     #3,R1
3241  012740  012702  000017                  MOV     #17,R2
3242  012744  012703  000000                  MOV     #0,R3
```

```
3243  012750  004737  022612                 JSR     PC,TTR          ;GO GET FORMAT
3244  012754  012703  000004                 MOV     #4,R3
3245  012760  004737  013562                 JSR     PC,TPOS
3246  012764  005237  005042       TINP2B:   INC     REOTC           ;BUMP EOT UNIT COUNTER
3247  012770  022737  000016  000702         CMP     #16,UNP         ;SEE IF DONE UNITS
3248  012776  001405                         BEQ     TINP2C          ;IF SO: BR
3249  013000  062737  000002  000702         ADD     #2,UNP          ;POINT TO NEXT UNIT
3250  013006  000137  012414                 JMP     TINP0           ;ELSE LOOK FOR NEXT UNIT
3251
3252
3253  013012  005037  000702       TINP2C:   CLR     UNP             ;CLEAR UNIT POINTER
3254  013016  113737  005042  005043         MOVB    REOTC,REOTC+1   ;SET # OF UNITS TO TEST
3255
3256  013024  000004  024640       TINP3:    TYPE,MSG35              ;REQUEST RECORDS PER BLOCK
3257  013030  013703  000554                 MOV     RCNT,R3
3258  013034  104400                         TYPOCT                  ;PRINT RECORD COUNT
3259  013036  012705  000554                 MOV     #RCNT,R5        ;SET RECORD COUNT ADDRESS
3260  013042  012701  000007                 MOV     #7,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3261  013046  012702  177777                 MOV     #177777,R2      ;SET MAXIMUM LIMIT
3262  013052  012703  000001                 MOV     #1,R3           ;SET MINIMUM LIMIT
3263  013056  004737  022612                 JSR     PC,TTR          ;GO GET RECORD COUNT
3264  013062  013737  000554  000640         MOV     RCNT,RCSAV      ;SAVE RECORD COUNT
3265
3266  013070  000004  024660                 TYPE,MSG36              ;REQUEST CHARACTERS PER RECORD
3267  013074  005437  000556                 NEG     FMCNT
3268  013100  013703  000556                 MOV     FMCNT,R3
3269  013104  104400                         TYPOCT                  ;PRINT CHAR COUNT
3270  013106  012705  000556                 MOV     #FMCNT,R5       ;SET CHARACTER COUNT ADDRESS
3271  013112  012701  000007                 MOV     #7,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3272  013116  013702  000560                 MOV     BUFMAX,R2       ;SET MAXIMUM LIMIT
3273  013122  005402                         NEG     R2              ;MAKE IT POSITIVE
3274  013124  012703  000004                 MOV     #4,R3           ;SET MINIMUM LIMIT
3275  013130  004737  022612                 JSR     PC,TTR          ;GO GET CHARACTER COUNT
3276  013134  005437  000556                 NEG     FMCNT           ;SET TO TWO'S COMPLIMENT
3277  013140  013737  000556  000642         MOV     FMCNT,FCSAV     ;SAVE FRAME COUNT
3278
3279  013146  000004  024676                 TYPE,MSG37              ;REQUEST PATTERN #
3280  013152  013703  000566                 MOV     PATRN,R3
3281  013156  104400                         TYPOCT                  ;PRINT PATTERN
3282  013160  005037  014130                 CLR     DOFL            ;CLEAR EXTERNAL DATA FLAG
3283  013164  012705  000566                 MOV     #PATRN,R5       ;SET PATTERN NUMBER ADDRESS
3284  013170  012701  000003                 MOV     #3,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3285  013174  012702  000015                 MOV     #15,R2          ;SET MAXIMUM LIMIT
3286  013200  012703  000000                 MOV     #0,R3           ;SET MINIMUM LIMIT
3287  013204  004737  022612                 JSR     PC,TTR          ;GO GET PATTERN NUMBER
3288
3289  013210  000004  025366                 TYPE,MSG69              ;REQUEST TAPE MARK
3290  013214  013703  000572                 MOV     TMFX,R3
3291  013220  104400                         TYPOCT                  ;PRINT CURRENT TM FLAG SETTING
3292  013222  012705  000572                 MOV     #TMFX,R5        ;GET TM FLAG ADDRESS
3293  013226  012701  000002                 MOV     #2,R1           ;SET SIZE OF RESPONSE
3294  013232  012702  000001                 MOV     #1,R2           ;SET UPPER LIMIT
3295  013236  012703  000000                 MOV     #0,R3           ;SET LOWER LIMIT
3296  013242  004737  022612                 JSR     PC,TTR          ;TM 1-YES
3297
3298  013246  000004  024070                 TYPE,MSG21              ;REQUEST INTERCHANGE READ
```

```
3299  013252  013703  000576                  MOV     INTRF,R3
3300  013256  104400                          TYPOCT
3301  013260  012705  000576                  MOV     #INTRF,R5       ;GET FLAG ADDRESS
3302  013264  012701  000002                  MOV     #2,R1           ;SET SIZE OF RESPONSE
3303  013270  012702  000001                  MOV     #1,R2           ;SET UPPER LIMIT
3304  013274  012703  000000                  MOV     #0,R3           ;SET LOWER LIMIT
3305  013300  004737  022612                  JSR     PC,TTR          ;GO GET RESPONSE
3306
3307  013304  000004  024713                  TYPE,MSG38              ;REQUEST SINGLE PASS
3308  013310  013703  000600                  MOV     SPFLG,R3
3309  013314  104400                          TYPOCT                  ;PRINT CURRENT SETTING
3310  013316  012705  000600                  MOV     #SPFLG,R5       ;SET ADDRESS OF FLAG
3311  013322  012701  000002                  MOV     #2,R1           ;SET SIZE OF RESPONSE
3312  013326  012702  000001                  MOV     #1,R2           ;SET UPPER LIMIT
3313  013332  012703  000000                  MOV     #0,R3           ;SET LOWER LIMIT
3314  013336  004737  022612                  JSR     PC,TTR          ;GO GET RESPONSE
3315
3316  013342  000004  024731          TINP3A: TYPE,MSG39              ;REQUEST CRC CORRECTION
3317  013346  013703  000574                  MOV     CRCC,R3
3318  013352  104400                          TYPOCT
3319  013354  012705  000574                  MOV     #CRCC,R5
3320  013360  012701  000002                  MOV     #2,R1
3321  013364  012702  000001                  MOV     #1,R2
3322  013370  012703  000000                  MOV     #0,R3
3323  013374  004737  022612                  JSR     PC,TTR
3324  013400  004737  022462                  JSR     PC,GTSWR        ;GET SWITCHES
3325  013404  005737  013560          TINP4:  TST     SCVFL           ;BRANCH IF SHORT CONVERSATION
3326  013410  001060                          BNE     TINPX
3327  013412  005737  000644          1$:     TST     TINF            ;BRANCH IF NO TTY INPUT
3328  013416  001455                          BEQ     TINPX
3329  013420  000004  024767                  TYPE,MSG40              ;REQUEST READ STALL
3330  013424  013703  000602                  MOV     RSTAL,R3
3331  013430  104400                          TYPOCT                  ;PRINT READ STALL
3332  013432  012705  000602                  MOV     #RSTAL,R5       ;SET READ STALL ADDRESS
3333  013436  012701  000007                  MOV     #7,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3334  013442  012702  177777                  MOV     #-1,R2          ;SET MAXIMUM LIMIT
3335  013446  012703  000001                  MOV     #1,R3           ;SET MINIMUM LIMIT
3336  013452  004737  022612                  JSR     PC,TTR          ;GO GET READ STALL
3337
3338  013456  000004  025016                  TYPE,MSG41              ;REQUEST WRITE STALL
3339  013462  013703  000604                  MOV     WSTAL,R3
3340  013466  104400                          TYPOCT                  ;PRINT READ STALL
3341  013470  012705  000604                  MOV     #WSTAL,R5       ;SET WRITE STALL ADDRESS
3342  013474  012701  000007                  MOV     #7,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3343  013500  012702  177777                  MOV     #-1,R2          ;SET MAXIMUM LIMIT
3344  013504  012703  000001                  MOV     #1,R3           ;SET MINIMUM LIMIT
3345  013510  004737  022612                  JSR     PC,TTR          ;GO GET WRITE STALL
3346
3347  013514  000004  025027                  TYPE,MSG42              ;REQUEST TURN AROUND STALL
3348  013520  013703  000606                  MOV     TSTAL,R3
3349  013524  104400                          TYPOCT                  ;PRINT TA STALL
3350  013526  012705  000606                  MOV     #TSTAL,R5       ;SET TURN AROUND STALL ADDRESS
3351  013532  012701  000007                  MOV     #7,R1           ;SET NUMBER OF CHARACTERS TO INPUT
3352  013536  012702  177777                  MOV     #-1,R2          ;SET MAXIMUM LIMIT
3353  013542  012703  000001                  MOV     #1,R3           ;SET MINIMUM LIMIT
3354  013546  004737  022612                  JSR     PC,TTR          ;GO GET TURN AROUND STALL
```

```
3355  013552  005037  013560          TINPX:  CLR     SCVFL           ;CLEAR SHORT CONVERSATION FLAG
3356  013556  000207                          RTS     PC              ;EXIT
3357  013560  000000                  SCVFL:  0                       ;SHORT CONVERSATION FLAG
3358
3359                                          ;UNIT DESCRIPTION POSITIONING SUBROUTINE**********
3360
3361  013562  006337  000654          TPOS:   ASL     TEMP2           ;POSITION CHARACTER
3362  013566  005303                          DEC     R3              ;SEE IF DONE
3363  013570  001374                          BNE     TPOS            ;IF NOT: BR
3364  013572  013700  000702                  MOV     UNP,R0          ;LOAD UNIT POINTER
3365  013576  053760  000654  000752          BIS     TEMP2,UN1(R0)   ;LOAD CHARACTER INTO UN1(R0)
3366  013604  000207                          RTS     PC              ;EXIT
3367
```

```
3369
3370                                    ;*********************************************************
3371                                    ;DATA SETUP ROUTINE:
3372                                    ;
3373                                    ;THIS ROUTINE IS USED TO GENERATE INTO THE ENTIRE
3374                                    ;WRITE BUFFER (4000 OCTAL CHARACTERS) THE DATA PATTERN
3375                                    ;SELECTED BY THE OPERATOR.  THERE ARE 15 (8) FIXED
3376                                    ;DATA PATTERNS AVAILABLE AND ONE SELECTION (DATA PATTERN 0)
3377                                    ;WHICH WILL READ ANY PATTERN PRESENTED AT THE
3378                                    ;HIGH SPEED PAPER TAPE READER.  THIS TAPE MUST BE PREPARED
3379                                    ;BY USING THE PROGRAM CALLED DTC. (MAINDEC-11-DZTUF-A-D)
3380                                    ;RANDOM DATA MAY ALSO BE USED VIA CONSOLE
3381                                    ;SWITCH EIGHT (8).
3382                                    ;THIS ROUTINE IS ALSO USED TO CLEAR OUT THE
3383                                    ;READ BUFFER (4000 OCTAL CHARACTERS) BEFORE EACH
3384                                    ;RECORD IS READ.
3385                                    ;*********************************************************
3386
3387  013606  005737  014520        DSUP:   TST     RDFL            ;SEE IF DID RANDOM DATA
3388  013612  001044                        BNE     DS2A            ;IF NOT: BR
3389  013614  005737  000742        DS0:    TST     ASEQF           ;SEE IF AUTO SEQ
3390  013620  001406                        BEQ     DS0C            ;IF NOT: BR
3391  013622  005737  000566                TST     PATRN           ;SEE IF AUTO RANDOM
3392  013626  100003                        BPL     DS0C            ;IF NOT: BR
3393  013630  004737  014456                JSR     PC,DATR         ;ELSE GO GENERATE RANDOM DATA
3394                           ;            RTS     PC              ;**B DELETED
3395  013634  000433                        BR      DS2A            ;**B GENERATE EXPECTED LRC/CRC & CLEAR READ BFR
3396  013636  023737  000566  013766  DS0C: CMP     PATRN,PATS      ;SEE IF NEW PATTERN
3397  013644  001014                        BNE     DS0A            ;IF SO: BR
3398  013646  013703  000552                MOV     UDES,R3         ;GET UNIT DESCRIPTION
3399  013652  042703  177767                BIC     #177767,R3      ;MASK EVEN PARITY
3400  013656  023703  013770                CMP     PARS,R3         ;SEE IF SAME AS LAST TIME
3401  013662  001404                        BEQ     DS0B            ;IF SO: BR
3402  013664  010337  013770                MOV     R3,PARS         ;SAVE PARITY
3403  013670  004737  014522                JSR     PC,CRCLRC       ;GO GENERATE EXPT CRC/LRC
3404  013674  000207                DS0B:   RTS     PC
3405  013676  013703  000562        DS0A:   MOV     @#WDATA,R3      ;R3 = ADDRS OF WRITE BUFFER
3406  013702  013701  000566                MOV     PATRN,R1        ;R1 = PATTERN SELECTOR
3407  013706  010137  013766                MOV     R1,PATS
3408  013712  062701  000001                ADD     #1,R1           ;BUMP POINTER
3409  013716  006301                        ASL     R1              ;MAKE PATTERN SELECTOR EVEN
3410  013720  004771  002771                JSR     PC,@DATBL(R1)   ;GO GENERATE PATTERN
3411  013724  004737  014522        DS2A:   JSR     PC,CRCLRC       ;GO GENERATE EXPT CRC/LRC
3412  013730  013702  000556        DS3:    MOV     FMCNT,R2        ;R2=BUFFER SIZE
3413  013734  006202                        ASR     R2              ;R2=FRAME CNT/2
3414  013736  013701  000564                MOV     @#RDATA,R1      ;R1=READ DATA START
3415  013742  005021                DS4:    CLR     (R1)+           ;CLEAR BUFFER
3416  013744  005202                        INC     R2              ;SEE IF DONE ALL
3417  013746  001375                        BNE     DS4             ;IF NOT: BR
3418  013750  013737  000552  013770        MOV     UDES,PARS       ;GET UNIT DESCRIPTION
3419  013756  042737  177767  013770        BIC     #177767,PARS    ;MASK PARITY
3420  013764  000207                        RTS     PC              ;EXIT
3421  013766  177777                PATS:   -1                      ;PATTERN NUMBER SAVE
3422  013770  000000                PARS:   0
3423
3424
```

```
3426
3427                                        ;EXTERNAL DATA INPUT FROM H/S READER (256 CHARACTER MAXIMUM)
3428
3429  013772  005737  014130        DATO:   TST     DOFL            ;BRANCH IF SHOULD DO EXTERNAL INPUT
3430  013776  001401                        BEQ     1$
3431  014000  000207                        RTS     PC              ;++B RETURN
3432  014002  012737  000001  014130  1$:   MOV     #1,DOFL         ;SET EXTERNAL FLAG
3433  014010  005077  164614                CLR     @PRS            ;CLEAR READER STATUS
3434  014014  005037  000652                CLR     TEMP1           ;CLEAR FOR USE AS CHARACTER FLAG
3435  014020  052777  000001  164602  DATOA:  BIS     #1,@PRS         ;START READER
3436  014026  105777  164576        DATOB:  TSTB    @PRS            ;SEE IF DONE
3437  014032  100375                        BPL     DATOB           ;IF NOT : BR
3438  014034  005001                        CLR     R1              ;CLEAR SAVE LOCATION
3439  014036  117701  164570                MOVB    @PRB,R1         ;SAVE CHARACTER
3440  014042  005737  000652                TST     TEMP1           ;SEE IF HAVE FOUND START CHARACTER
3441  014046  001011                        BNE     DATOC           ;IF SO : BR
3442  014050  105701                        TSTB    R1              ;SEE IF CHARACTER IS 0
3443  014052  001762                        BEQ     DATOA           ;IF SO : BR
3444  014054  012737  000001  000652        MOV     #1,TEMP1        ;ELSE SET CHARACTER FOUND FLAG
3445  014062  010137  000654                MOV     R1,TEMP2        ;SAVE DATA SIZE
3446  014066  010102                        MOV     R1,R2           ;SAVE DATA SIZE
3447  014070  000753                        BR      DATOA           ;GO GET FIRST DATA CHAR
3448  014072  110123        DATOC:  MOVB    R1,(R3)+        ;LOAD BUFFER
3449  014074  005302                        DEC     R2              ;SEE IF READ ALL
3450  014076  001350                        BNE     DATOA           ;IF NOT : BR
3451  014100  013701  000562        DATOD:  MOV     @#WDATA,R1      ;R1 = START OF WRITE BUFFER
3452  014104  013702  000654                MOV     TEMP2,R2        ;R2 = SIZE OF DATA FIELD
3453  014110  112123        DATOE:  MOVB    (R1)+,(R3)+     ;REPEAT LOAD OF DATA FIELD
3454  014112  023703  000564                CMP     @#RDATA,R3      ;SEE IF DONE
3455  014116  003001                        BGT     DATOF           ;IF NOT: BR
3456  014120  000207                        RTS     PC              ;++B RETURN
3457  014122  005302        DATOF:  DEC     R2              ;SEE IF AT END OF DATA FIELD
3458  014124  001371                        BNE     DATOE           ;IF NOT : BR
3459  014126  000764                        BR      DATOD           ;ELSE RESTART FILL
3460  014130  000000        DOFL:   0                               ;EXTERNAL DATA FLAG=1 IF ALREADY DONE
3461
```

```
3463                                        ;ALL ONES**********
3464
3465  014132  012701  177777      DAT1:     MOV     #-1,R1          ;R1=DATA
3466  014136  012702  002002      DAT1A:    MOV     #2002,R2        ;R2=WORD COUNT *2
3467  014142  010123              1$:       MOV     R1,(R3)+        ;LOAD BUFFER
3468  014144  005302                        DEC     R2              ;SEE IF DONE
3469  014146  001375                        BNE     1$              ;IF NOT: BR
3470  014150  000207                        RTS     PC
3471
3472                                        ;ALL ZEROS**********
3473
3474  014152  005001              DAT2:     CLR     R1              ;R1=DATA
3475  014154  000770                        BR      DAT1A           ;LOAD BUFFER
3476
3477                                        ;WALKING ONE**********
3478
3479  014156  012701  000001      DAT3:     MOV     #1,R1           ;R1=DATA
3480  014162  000241                        CLC
3481  014164  012702  004004      DAT3A:    MOV     #4004,R2        ;R2=CHARACTER COUNT*4
3482  014170  110123              1$:       MOVB    R1,(R3)+        ;LOAD BUFFER
3483  014172  106101                        ROLB    R1              ;SET NEXT CHARACTER
3484  014174  005302                        DEC     R2              ;SEE IF DONE
3485  014176  001374                        BNE     1$              ;IF NOT: BR
3486  014200  000207                        RTS     PC
3487
3488                                        ;WALKING ZERO**********
3489
3490  014202  012701  000376      DAT4:     MOV     #376,R1         ;R1=START OF DATA
3491  014206  000261                        SEC
3492  014210  000765                        BR      DAT3A           ;LOAD BUFFER
3493
3494                                        ;ALTERNATING ONE/ZERO**********
3495
3496
3497  014212  012701  052525      DAT5:     MOV     #052525,R1      ;R1=DATA
3498  014216  000747                        BR      DAT1A           ;LOAD BUFFER
3499
3500                                        ;ALTERNATING ZERO/ONE**********
3501
3502  014220  012701  125252      DAT6:     MOV     #125252,R1      ;R1=DATA
3503  014224  000744                        BR      DAT1A           ;LOAD BUFFER
3504
3505                                        ;ONE/ZERO IN ALTERNATING WORDS**********
3506
3507  014226  012701  125252      DAT7:     MOV     #125252,R1      ;SET WORD 1
3508  014232  012702  052525                MOV     #052525,R2      ;SET WORD 2
3509  014236  012704  001002                MOV     #1002,R4        ;SET NUMBER OF ENTRIES
3510  014242  010123              1$:       MOV     R1,(R3)+        ;LOAD WORD 1
3511  014244  010223                        MOV     R2,(R3)+        ;LOAD WORD 2
3512  014246  005304                        DEC     R4              ;SEE IF DONE
3513  014250  001374                        BNE     1$              ;IF NOT: BR
3514  014252  000207                        RTS     PC
3515
```

```
                                        ;WALKING ONE/ALL ONE IN ALTERNATING CHARS****
                                
014254  012702  002002          DAT10:  MOV     #2002,R2        ;SET BUFFER SIZE
014260  012701  000001                  MOV     #1,R1           ;SET WALK BASE
014264  000241                          CLC
014266  012713  177400          1$:     MOV     #177400,(R3)    ;LOAD ALL ONE BYTE
014272  050123                          BIS     R1,(R3)+        ;LOAD WALK BYTE
014274  106101                          ROLB    R1              ;WALK ONE
014276  005302                          DEC     R2
014300  001372                          BNE     1$              ;DO FULL BUFFER
014302  000207                          RTS     PC

                                        ;ALL BITS 0-377**********

014304  005001          DAT11:  CLR     R1              ;R1=STARTING DATA
014306  012702  004004                  MOV     #4004,R2        ;R2=CHARACTER COUNT*4
014312  110123          1$:     MOVB    R1,(R3)+        ;LOAD BUFFER
014314  105201                  INCB    R1              ;BUMP DATA
014316  005302                  DEC     R2              ;SEE IF DONE
014320  001374                  BNE     1$              ;IF NOT: BR
014322  000207                  RTS     PC              ;RETURN

                                        ;ALL BITS 377-0**********

014324  012701  000377          DAT12:  MOV     #377,R1         ;R1=STARTING DATA
014330  012702  004004                  MOV     #4004,R2        ;R2=CHARACTER COUNT*4
014334  110123          1$:     MOVB    R1,(R3)+        ;LOAD BUFFER
014336  105301                  DECB    R1              ;BUMP DATA
014340  005302                  DEC     R2              ;SEE IF DONE
014342  001374                  BNE     1$              ;IF NOT: BR
014344  000207                  RTS     PC              ;RETURN

                                        ;ALTERNATING CHARACTERS 0 AND 377**********

014346  012701  000377          DAT13:  MOV     #377,R1         ;R1    DATA
014352  000137  014136                  JMP     DAT1A           ;LOAD BUFFER

                                        ;WALKING ZERO/ALL ZERO IN ALTERNATING CHARS******

014356  012702  002002          DAT14:  MOV     #2002,R2        ;SET BUFFER SIZE
014362  012701  000376                  MOV     #376,R1         ;SET WALK BASE
014366  000261                          SEC
014370  010113          1$:     MOV     R1,(R3)         ;LOAD WALK BYTE
014372  042723  177400                  BIC     #177400,(R3)+   ;CLEAR HIGH BYTE
014376  106101                          ROLB    R1              ;WALK ZERO BIT
014400  005302                          DEC     R2
014402  001372                          BNE     1$              ;FILL BUFFER
014404  000207                          RTS     PC              ;RETURN
```

```
3570                                        ;AUTO SEQUENCE PATTERN*********
3571
3572  014406  012702  000200        DAT15:  MOV     #200,R2         ;SET NUMBER OF ENTRIES
3573  014412  012701  014436        1$:     MOV     #APATS,R1       ;SET START OF PATTERN
3574  014416  012704  000010                MOV     #10,R4          ;SET SIZE OF PATTERN
3575  014422  012123                2$:     MOV     (R1)+,(R3)+     ;FILL BUFFER
3576  014424  005304                        DEC     R4              ;SEE IF DONE PATTERN
3577  014426  001375                        BNE     2$              ;IF NOT: BR
3578  014430  005302                        DEC     R2              ;SEE IF DONE BUFER
3579  014432  001367                        BNE     1$              ;IF NOT: BR
3580  014434  000207                        RTS     PC              ;RETURN
3581
3582  014436  000000                APATS:  0
3583  014440  177400                        177400
3584  014442  000377                        377
3585  014444  000000                        0
3586  014446  177777                        -1
3587  014450  000377                        377
3588  014452  177400                        177400
3589  014454  177777                        -1
3590
3591                                        ;RANDOM DATA GENERATOR SUBROUTINE**********
3592
3593  014456  013704  000556        DATR:   MOV     FMCNT,R4        ;SET NUMBER OF FRAMES
3594  014462  013703  000562                MOV     @#WDATA,R3      ;SET ADDRESS OF START OF BUFFER
3595  014466  012701  177777                MOV     #-1,R1          ;SET HIGH LIMIT
3596  014472  005002                        CLR     R2              ;SET LOW LIMIT
3597  014474  004737  022430        1$:     JSR     PC,RANG         ;GO GENERATE NUMBER
3598  014500  013723  000636                MOV     RANSAV,(R3)+    ;LOAD BUFFER
3599  014504  005204                        INC     R4              ;SEE IF DONE WHOLE BUFFER
3600  014506  001372                        BNE     1$              ;IF NOT; BR
3601  014510  012737  000001  014520        MOV     #1,RDFL         ;SET RANDOM DATA FLAG
3602  014516  000207                        RTS     PC              ;EXIT
3603  014520  000000                RDFL:   0                       ;RANDOM DATA SELECT FLAG
```

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

014522  013700  000556          CRCLRC: MOV     FMCNT,R0        ;SET RECORD SIZE
014526  005400                          NEG     R0
014530  013701  000562                  MOV     @#WDATA,R1      ;SET START OF BUFFER
014534  005037  015056                  CLR     XORS
014540  111104                  CL0:    MOVB    (R1),R4         ;GET CHARACTER
014542  004737  014730                  JSR     PC,CLP          ;GO GET PARITY OF CHARACTER
014546  004737  015032                  JSR     PC,XOR          ;XOR CHARACTER
014552  000241                          CLC
014554  006004                          ROR     R4              ;ROTATE 1 RIGHT
014556  103014                          BCC     CL2             ;IF NO CARRY: BR
014560  052704  000400                  BIS     #400,R4         ;SET BIT NINE
014564  000241                          CLC
014566  010405                  CL1:    MOV     R4,R5           ;SAVE CHARACTER
014570  042705  177703                  BIC     #177703,R5
014574  005105                          COM     R5
014576  042705  177703                  BIC     #177703,R5
014602  042704  000074                  BIC     #74,R4
014606  050504                          BIS     R5,R4           ;COMPLIMENT BITS 2,3,4,5
014610  010437  015056          CL2:    MOV     R4,XORS
014614  005300                          DEC     R0
014616  001350                          BNE     CL0             ;BRANCH IF NOT LAST CHAR
014620  013704  015056          CLLAST: MOV     XORS,R4
014624  005137  015056                  COM     XORS
014630  042737  177050  015056          BIC     #177050,XORS
014636  042704  177727                  BIC     #177727,R4      ;COMPLIMENT ALL BUT BITS 3&5
014642  050437  015056                  BIS     R4,XORS
014646  013737  015056  015060          MOV     XORS,EXCRC      ;SAVE EXPECTED CRC
014654  013700  000556                  MOV     FMCNT,R0
014660  005400                          NEG     R0
014662  013701  000562                  MOV     @#WDATA,R1      ;DO EXPT LRC
014666  005037  015056                  CLR     XORS
014672  111104                  CL3:    MOVB    (R1),R4
014674  004737  014730                  JSR     PC,CLP          ;GET PARITY
014700  004737  015032                  JSR     PC,XOR          ;XOR CHARACTER
014704  005300                          DEC     R0
014706  001371                          BNE     CL3             ;DO ALL FOR LRC
014710  013704  015060                  MOV     EXCRC,R4
014714  004737  015032                  JSR     PC,XOR          ;XOR CRC TO DATA
014720  013737  015056  015062          MOV     XORS,EXLRC      ;SAVE EXPT LRC
014726  000207                          RTS     PC              ;RETURN
014730  005704                  CLP:    TST     R4              ;SEE IF 0 CHAR
014732  001010                          BNE     CLPE            ;IF NOT: BR
014734  032737  000010  000552          BIT     #10,DDES        ;SEE IF EVEN PARITY
014742  001404                          BEQ     CLPE            ;IF NOT: BR
014744  012704  000420                  MOV     #420,R4         ;SET 0 CHAR EVEN PARITY
014750  005201                          INC     R1              ;BUMP POINTER
014752  000207                          RTS     PC              ;RETURN
```

```
014754  005046                    CLPE:   CLR     (SP)            ;CLEAR WORLING SPACE ON STACK
014756  106304                    1$:     ASLB    R4              ;SHIFT DATA
014760  005516                            ADC     (SP)            ;ADDUP # OF 1 BITS
014762  105704                            TSTB    R4              ;BRANCH IF ALL 0"S LEFT
014764  001374                            BNE     1$
014766  112104                            MOVB    (R1)+,R4
014770  042704  177400                    BIC     #177400,R4
014774  106026                            RORB    (SP)+           ;BRANCH IF ODD # OF 1 BITS
014776  103405                            BCS     CLP2
015000  032737  000010  000552            BIT     #10,UDES        ;SEE IF SHOULD BE EVEN PARITY
015006  001406                            BEQ     CLP3            ;IF NOT: BR
015010  000207                            RTS     PC              ;ELSE EXIT
015012  032737  000010  000552    CLP2:   BIT     #10,UDES        ;SEE IF SHOULD BE ODD PARITY
015020  001001                            BNE     CLP3            ;IF NOT: BR
015022  000207                            RTS     PC              ;ELSE EXIT
015024  052704  000400            CLP3:   BIS     #400,R4         ;SET PARITY BIT
015030  000207                            RTS     PC

015032  010446                    XOR:    MOV     R4,-(SP)
015034  043716  015056                    BIC     XORS,(SP)
015040  040437  015056                    BIC     R4,XORS         ;XOR SUBROUTINE: R4 WITH XORS
015044  052637  015056                    BIS     (SP)+,XORS
015050  013704  015056                    MOV     XORS,R4
015054  000207                            RTS     PC

015056  000000                    XORS:   0                       ;XOR SAVE
015060  000000                    EXCRC:  0                       ;EXPECTED CRC
015062  000000                    EXLRC:  0                       ;EXPECTED LRC
```

```
3691
3692                                    ;*********************************************************
3693                                    ;DATA CHECK SUBROUTINE:
3694                                    ;
3695                                    ;THIS SUBROUTINE IS USED TO COMPARE EACH CHARACTER
3696                                    ;OF DATA READ FROM TAPE WITH THE EXPECTED CHARACTER.
3697                                    ;ANY ERROR DETECTED WILL CAUSE CONTROL TO BE
3698                                    ;PASSED TO AN ERROR PRINT SUBROUTINE AND A
3699                                    ;SUBROUTINE TO ACCUMULATE THE NUMBER OF BITS
3700                                    ;DROPPED AND PICKED UP FROM EACH CHARACTER.
3701                                    ;THE NUMBER OF READ ERRORS IS ALSO ACCUMULATED.
3702                                    ;DATA CHECKING MAY BE TERMINATED BY USE OF
3703                                    ;CONSOLE SWITCH THIRTEEN (13).
3704                                    ;*********************************************************
3705
3706  015064  005037  000664        DCHK:   CLR     BBC             ;CLEAR BAD RECORD CNTR
3707  015070  005037  000712                CLR     DERFL           ;CLEAR DATA ERROR FLAG
3708  015074  013705  000556                MOV     FMCNT,R5        ;LOAD CHAR COUNT
3709  015100  032737  000020  000552        BIT     #20,UDES        ;SEE IF CORE DUMP
3710  015106  001401                        BEQ     DCHKO           ;IF NOT: BR
3711  015110  006205                        ASR     R5              ;R5 = FC/2
3712  015112  013701  000562        DCHKO:  MOV     @#WDATA,R1      ;SET WRITE DATA ADDR
3713  015116  013702  000564                MOV     @#RDATA,R2      ;SET READ DATA ADDR
3714  015122  032737  000010  000552        BIT     #10,UDES        ;SEE IF EVEN PARITY
3715  015130  001430                        BEQ     DFOCO           ;IF NOT: BR
3716  015132  032737  000020  000552        BIT     #20,UDES        ;SEE IF CORE DUMP PARITY
3717  015140  001024                        BNE     DFOCO           ;IF SO: BR
3718  015142  032737  002000  000552        BIT     #2000,UDES      ;SEE IF PE MODE
3719  015150  001020                        BNE     DFOCO           ;IF SO: BR
3720  015152  105711                DFOF:   TSTB    (R1)            ;SEE IF 0 CHAR
3721  015154  001404                        BEQ     DFOD            ;IF SO: BR
3722  015156  005201                        INC     R1              ;BUMP POINTER
3723  015160  005205                DFOE:   INC     R5              ;SEE IF DONE
3724  015162  001373                        BNE     DFOF            ;IF NOT: BR
3725  015164  000406                        BR      DFOC            ;ELSE CONTINUE
3726  015166  112721  000020        DFOD:   MOVB    #20,(R1)+       ;SET 20 IN PLACE OF 0
3727  015172  012737  177777  013766        MOV     #-1,PATS        ;SET PATTERN GENERATE FLAG
3728  015200  000767                        BR      DFOE
3729  015202  013705  000556        DFOC:   MOV     FMCNT,R5        ;RESET CHAR CNT
3730  015206  013701  000562                MOV     @#WDATA,R1      ;RESET DATA ADDRESS
3731  015212  005737  000570        DFOCO:  TST     RDCMD           ;SEE IF READ REVERSE
3732  015216  001462                        BEQ     DFO             ;IF NOT: BR
3733  015220  013704  000556        DFOB:   MOV     FMCNT,R4        ;GET FRAME COUNT
3734  015224  005404                        NEG     R4              ;SET TO WHOLE NUMBER
3735  015226  032737  000020  000552        BIT     #20,UDES        ;SEE IF CORE DUMP
3736  015234  001402                        BEQ     DFOBO           ;IF NOT: BR
3737  015236  000241                        CLC
3738  015240  006004                        ROR     R4              ;SET TO FC/2
3739  015242  060401                DFOBO:  ADD     R4,R1           ;POINT TO START OF WRITE DATA
3740  015244  060402                        ADD     R4,R2           ;POINT TO START OF READ DATA
3741  015246  032737  000001  000556        BIT     #1,FMCNT        ;SEE IF ODD FRAME COUNT
3742  015254  001401                        BEQ     DFOA            ;IF NOT: BR
3743  015256  105722                        TSTB    (R2)+           ;BUMP POINTER
3744  015260  032737  000020  000552  DFOA:  BIT     #20,UDES        ;SEE IF CORE DUMP
3745  015266  001431                        BEQ     DFOA4           ;IF NOT: BR
3746  015270  000241                        CLC
```

```
3747  015272  132742  000001              BITB    #1,-(R2)        ;SEE IF BIT 0 = 1
3748  015276  001401                      BEQ     DFOA0           ;IF NOT: BR
3749  015300  000261                      SEC
3750  015302  106012              DFOA0:  RORB    (R2)
3751  015304  000241                      CLC
3752  015306  132712  000001              BITB    #1,(R2)
3753  015312  001401                      BEQ     DFOA1
3754  015314  000261                      SEC
3755  015316  106012              DFOA1:  RORB    (R2)            ;POSITION BITS FOR REVERSE CORE DUMP
3756  015320  000241                      CLC
3757  015322  132712  000001              BITB    #1,(R2)
3758  015326  001401                      BEQ     DFOA2
3759  015330  000261                      SEC
3760  015332  106012              DFOA2:  RORB    (R2)
3761  015334  000241                      CLC
3762  015336  132712  000001              BITB    #1,(R2)
3763  015342  001401                      BEQ     DFOA3
3764  015344  000261                      SEC
3765  015346  106012              DFOA3:  RORB    (R2)
3766  015350  005202                      INC     R2              ;RESET POINTER
3767  015352  124142              DFOA4:  CMPB    -(R1),-(R2)     ;TEST DATA CHARACTER
3768  015354  001010                      BNE     DF1             ;IF NOT GOOD: BR
3769  015356  105037  000664              CLRB    BBC             ;CLEAR BAD RECORD COUNTER
3770  015362  000411                      BR      DF2
3771  015364  122122              DFO:    CMPB    (R1)+,(R2)+     ;CHECK DATA
3772  015366  001003                      BNE     DF1             ;IF BAD: BR
3773  015370  105037  000664              CLRB    BBC             ;CLEAR BAD RECORD CNTR
3774  015374  000404                      BR      DF2
3775  015376  004737  016134      DF1:    JSR     PC,DRPKF        ;GO GET DROPS AND PICKS
3776  015402  004737  015470              JSR     PC,DERR         ;GO DO PRINT
3777  015406  005205              DF2:    INC     R5              ;BUMP CHAR CNTR
3778  015410  001404                      BEQ     DF3             ;IF DONE ALL: BR
3779  015412  005737  000570              TST     RDCMD           ;SEE IF READ REVERSE
3780  015416  001762                      BEQ     DFO             ;IF NOT: BR
3781  015420  000717                      BR      DFOA            ;ELSE CONTINUE READ REV
3782  015422  005037  000672      DF3:    CLR     HDRFL           ;CLEAR HEADER FLAG
3783  015426  005737  000712              TST     DERFL           ;SEE IF HAD DATA ERROR
3784  015432  001415                      BEQ     DFX             ;IF NOT: BR
3785  015434  005737  000714              TST     SERFL
3786  015440  001012                      BNE     DFX             ;IF NOT DATA ERROR ONLY: BR
3787  015442  013704  000700              MOV     UNP,R4
3788  015446  005737  000570              TST     RDCMD           ;SEE IF READ REVERSE
3789  015452  001003                      BNE     DF4             ;IF SO: BR
3790  015454  005264  001134              INC     DATER1(R4)      ;BUMP DATA ERROR FORWARD COUNTER
3791  015460  000402                      BR      DFX
3792  015462  005264  001174      DF4:    INC     DEREV1(R4)      ;BUMP REVERSE DATA ERROR
3793  015466  000207              DFX:    RTS     PC              ;EXIT
3794
```

```
3796
3797                                    ;*********************************************************
3798                                    ;DATA ERROR SUBROUTINE:
3799                                    ;
3800                                    ;THIS SUBROUTINE IS USED TO PRINT OUT ANY
3801                                    ;ERRORS FOUND DURING THE DATA CHECK.
3802                                    ;EACH CHARACTER FOUND BAD WILL BE PRINTED
3803                                    ;IN BIT FORMAT ALONG WITH ITS EXPECTED CHARACTER.
3804                                    ;AN ERROR HEADER CONSISTING OF THE UNIT NUMBER,
3805                                    ;BLOCK NUMBER, RECORD NUMBER, SIZE OF RECORD, AND
3806                                    ;ERROR TYPE (READ FORWARD, READ REVERSE, WRITE, ETC)
3807                                    ;IS PRINTED ONLY ONCE FOR EACH RECORD FOUND BAD.
3808                                    ;A COUNT IS MADE OF THE NUMBER OF SUCCESSIVE BAD
3809                                    ;CHARACTERS, AND IF TEN (10) SUCCESSIVE BAD CHARACTERS
3810                                    ;ARE FOUND IN A SINGLE RECORD, A MESSAGE INDICATING
3811                                    ;A BAD RECORD CONDITION IS PRINTED AND THE NEXT
3812                                    ;TWENTY (20) CHARACTERS ARE SKIPPED BEFORE CHECKING
3813                                    ;IS RESUMED.  IF THE BAD RECORD CONDITION IS FOUND
3814                                    ;THREE TIMES IN A RECORD, ALL REMAINING DATA IS
3815                                    ;SKIPPED EXCEPT THE FINAL TEN (10) CHARACTERS.
3816                                    ;THIS SKIPPING IS OF COURSE ONLY POSSIBLE IN
3817                                    ;RECORDS WHICH CONTAIN A SUFFICIENT NUMBER OF CHARACTERS.
3818                                    ;PRINTING OF ERRORS MAY BE DISALLOWED AT ANY TIME
3819                                    ;BY SETTING CONSOLE SWITCH TEN (10) TO A ONE.
3820                                    ;THE OPERATOR MAY CAUSE THE PROGRAM TO HALT ON ANY ERROR
3821                                    ;BY SETTING CONSOLE SWITCH FIFTEEN (15) TO A ONE.
3822                                    ;*********************************************************
3823
3824  015470  032777  002000  163120  DERR:   BIT     #2000,@SWR      ;BRANCH IF NO ERROR
3825  015476  001057                   BNE     DERR4           ;PRINTOUT DESIRED
3826  015500  005237  000676          DERRO:  INC     PFLG            ;SET PRINT FLAG
3827  015504  005737  000672                  TST     HDRFL           ;SEE IF HAVE PRINTED HEADER
3828  015510  001006                   BNE     DERROA          ;IF SO; BR
3829  015512  004737  022126                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
3830  015516  000004  023722                  TYPE,MSG1               ;TYPE DATA ERROR TAG '*DE'
3831  015522  004737  020364                  JSR     PC,FRPRT        ;PRINT F OR R
3832  015526  000004  023741          DERROA: TYPE,MSG4               ;TYPE CHAR # TAG 'CN'
3833  015532  010203                   MOV     R2,R3
3834  015534  163703  000564                  SUB     @#RDATA,R3      ;POINT TO CHAR
3835  015540  005303                   DEC     R3
3836  015542  005737  000570                  TST     RDCMD           ;SEE IF READ REVERSE
3837  015546  001402                   BEQ     DERROB          ;IF NOT; BR
3838  015550  010503                   MOV     R5,R3           ;GET CHAR NUMBER
3839  015552  005103                   COM     R3
3840  015554  104400          DERROB: TYPOCT                  ;PRINT CHAR NUMBER
3841  015556  000004  023727                  TYPE,MSG2               ;TYPE GOOD CHAR TAG 'G'
3842  015562  005737  000570                  TST     RDCMD           ;SEE IF READ REVERSE
3843  015566  001402                   BEQ     DERROC          ;IF NOT; BR
3844  015570  111103                   MOVB    (R1),R3         ;GET CHAR
3845  015572  000401                   BR      DERROD
3846  015574  114103          DERROC: MOVB    -(R1),R3        ;LOAD EXPECTED DATA
3847  015576  004737  023504          DERROD: JSR     PC,DOUT         ;GO PRINT CHAR
3848  015602  000004  023734                  TYPE,MSG3               ;TYPE BAD CHARACTER TAG 'B'
3849  015606  005737  000570                  TST     RDCMD           ;SEE IF READ REVERSE
3850  015612  001402                   BEQ     DERR1           ;IF NOT; BR
3851  015614  111203                   MOVB    (R2),R3         ;GET CHAR
```

```
3852  015616  000401                          BR      DERR2
3853  015620  114203                  DERR1:  MOVB    -(R2),R3
3854  015622  004737  023504          DERR2:  JSR     PC,DOUT         ;PRINT BAD CHAR
3855  015626  005737  000570                  TST     RDCMD           ;BRANCH IF READ
3856  015632  001001                          BNE     DERR4           ;REVERSE
3857  015634  122122                  DERR3:  CMPB    (R1)+,(R2)+     ;RESET POINTERS
3858  015636  105237  000664          DERR4:  INCB    BBC             ;BUMP BAD RECORD CNTR
3859  015642  122737  000010  000664          CMPB    #10,BBC         ;SEE IF BLD BTH
3860  015650  001107                          BNE     DEREX           ;IF NOT: BR
3861  015652  032777  002000  162736          BIT     #2000,@SWR      ;SEE IF PRINT INHIBIT
3862  015660  001002                          BNE     1$              ;IF SO: BR
3863  015662  000004  024022                  TYPE,MSG15              ;TYPE 'BAD RECORD'
3864  015666  105037  000664          1$:     CLRB    BBC             ;RESET BAD RECORD CNTR
3865  015672  105237  000665                  INCB    BBC+1           ;BUMP AMOUNT
3866  015676  122737  000003  000665          CMPB    #3,BBC+1        ;SEE IF HAD 3 BLD BTHS
3867  015704  101047                          BHI     DERR4B          ;IF NOT: BR
3868  015706  022705  177767                  CMP     #177767,R5      ;SEE IF ON LAST EIGHT CHARS
3869  015712  101464                          BLOS    DERR6           ;IF SO: BR
3870  015714  012705  177767                  MOV     #177767,R5      ;SET CHAR CNTR TO 8
3871  015720  005737  000570                  TST     RDCMD           ;SEE IF READ REVERSE
3872  015724  001416                          BEQ     DERR4A          ;IF NOT: BR
3873  015726  013701  000562                  MOV     @#WDATA,R1      ;GET START OF BUFFER
3874  015732  013702  000564                  MOV     @#RDATA,R2      ;GET START OF BUFFER
3875  015736  062701  000010                  ADD     #10,R1
3876  015742  062702  000010                  ADD     #10,R2          ;POINT TO START +10
3877  015746  032737  000001  000556          BIT     #1,FMCNT        ;SEE IF ODD FRAME COUNT
3878  015754  001445                          BEQ     DEREX           ;IF NOT: BR
3879  015756  105722                          TSTB    (R2)+           ;BUMP POINTER
3880  015760  000443                          BR      DEREX
3881  015762  013737  000556  000652  DERR4A: MOV     FMCNT,TEMP1     ;LOAD CHAR COUNT
3882  015770  005437  000652                  NEG     TEMP1           ;*-8
3883  015774  162737  000010  000652          SUB     #10,TEMP1       ;POINT TO BUFFER -8
3884  016002  013701  000652                  MOV     TEMP1,R1        ;POINT TO NEXT CHAR
3885  016006  063701  000562                  ADD     @#WDATA,R1      ;POINT TO NEXT WRITE CHAR
3886  016012  013702  000652                  MOV     TEMP1,R2        ;POINT TO END OF READ DATA -8 FORWARD
3887  016016  063702  000564                  ADD     @#RDATA,R2      ;POINT TO NEXT CHAR
3888  016022  000422                          BR      DEREX           ;EXIT
3889  016024  062705  000024          DERR4B: ADD     #24,R5          ;SKIP 20 CHARS
3890  016030  103415                          BCS     DERR6           ;IF EXCEED RECORD SIZE: BR
3891  016032  005737  000570                  TST     RDCMD           ;SEE IF READ REVERSE
3892  016036  001405                          BEQ     DERR5           ;IF NOT: BR
3893  016040  162701  000024                  SUB     #24,R1
3894  016044  162702  000024                  SUB     #24,R2          ;RESET POINTERS
3895  016050  000407                          BR      DEREX
3896  016052  062701  000024          DERR5:  ADD     #24,R1          ;SKIP 20 CHARS
3897  016056  062702  000024                  ADD     #24,R2          ;SKIP FORWARD 20 CHARS
3898  016062  000402                          BR      DEREX
3899  016064  012705  177777          DERR6:  MOV     #-1,R5          ;SET TO EOR
3900  016070  005777  162522          DEREX:  TST     @SWR            ;BRANCH IF NOT HALT ON ERROR
3901  016074  100012                          BPL     DEREX1
3902  016076  000000                          HALT
3903  016100  005737  000676                  TST     PFLG            ;SEE IF PRINTED
3904  016104  001006                          BNE     DEREX1          ;IF SO: BR
3905  016106  032777  002000  162502          BIT     #2000,@SWR      ;SEE IF SHOULD PRINT
3906  016114  001002                          BNE     DEREX1          ;IF NOT: BR
3907  016116  000137  015500                  JMP     DERR0           ;ELSE PRINT
```

```
 3908  016122  005037  000676          DEREX1: CLR     PFLG            ;CLEAR FLAG
 3909  016126  005237  000712                  INC     DERFL           ;BUMP DATA ERROR FLAG
 3910  016132  000207                          RTS     PC              ;RETURN
 3911
```

```
3913
3914                                    ;*********************************************************
3915                                    ;DROPS AND PICKS SUBROUTINE:
3916                                    ;
3917                                    ;THIS SUBROUTINE IS USED TO ACCUMULATE FROM
3918                                    ;EACH BAD DATA CHARACTER FOUND THE NUMBER
3919                                    ;OF BITS WHICH WERE EITHER DROPPED OR PICKED UP.
3920                                    ;TWO COUNTERS PER SLAVE ARE USED TO ACCUMULATE THIS
3921                                    ;INFORMATION AND CAN STORE UP TO 32K DROPS
3922                                    ;OR PICKS BEFORE OVERFLOWING.  IF OVERFLOW IS
3923                                    ;ABOUT TO OCCUR, THESE ACCUMULATORS ARE
3924                                    ;PRINTED IN OCTAL AND RESET TO ZERO.
3925                                    ;THE CONTENTS OF THE ACCUMULATORS MAY BE
3926                                    ;DISPLAYED AT ANY TIME BY SETTING CONSOLE
3927                                    ;SWITCH FOURTEEN TO A ONE (1).  THE PRINTOUT WILL OCCUR
3928                                    ;AT THE END OF THE CURRENT BLOCK CYCLE.
3929                                    ;*********************************************************
3930
3931  016134  005037  000652       DRPKF:  CLR     TEMP1
3932  016140  005037  000654               CLR     TEMP2
3933  016144  005037  000656               CLR     TEMP3
3934  016150  111137  000652               MOVB    (R1),TEMP1      ;LOAD GOOD CHAR
3935  016154  111237  000654               MOVB    (R2),TEMP2      ;LOAD BAD CHAR
3936  016160  013704  000702               MOV     UNP,R4
3937  016164  016437  000774  000726       MOV     PIK1(R4),BPKP
3938  016172  016437  001014  000724       MOV     DRP1(R4),BDPP
3939  016200  005737  000570               TST     RDCMD           ;SEE IF READ REVERSE
3940  016204  001005                       BNE     DRPK            ;IF SO: BR
3941  016206  124142                       CMPB    -(R1),-(R2)     ;POINT TO CHAR
3942  016210  112137  000652               MOVB    (R1)+,TEMP1     ;LOAD GOOD CHAR
3943  016214  112237  000654               MOVB    (R2)+,TEMP2     ;LOAD BAD CHAR
3944  016220  004737  016232       DRPK:   JSR     PC,DROP         ;GET DROPS
3945  016224  004737  016440               JSR     PC,PICK         ;GET PICKS
3946  016230  000207                       RTS     PC              ;EXIT
3947
3948  016232  113703  000652       DROP:   MOVB    TEMP1,R3        ;R3 = GOOD CHAR
3949  016236  113704  000654               MOVB    TEMP2,R4        ;R4 = BAD CHAR
3950  016242  140403               DPC:    BICB    R4,R3           ;GET DROPS/PICKS
3951  016244  001001                       BNE     DPCG            ;IF SOME: BR
3952  016246  000207                       RTS     PC              ;RETURN
3953  016250  012737  000010  000716 DPCG:  MOV     #10,BCNT        ;SET NUMBER TO CHECK
3954  016256  132703  000001       DPC0:   BITB    #1,R3           ;SEE IF DROPPED OR PICKED THIS BIT
3955  016262  001451                       BEQ     DPC2            ;IF NOT: BR
3956  016264  105737  000656               TSTB    TEMP3           ;SEE IF ON PICKS
3957  016270  001014                       BNE     DPC1            ;IF SO: BR
3958  016272  005277  162426               INC     @BDPP           ;BUMP DROP CNTR
3959  016276  100043                       BPL     DPC2            ;IF NO OVERFLOW: BR
3960  016300  032777  002000  162310       BIT     #2000,@SWR      ;SEE IF HAVE PRINTED DATA
3961  016306  001402                       BEQ     DPC0A           ;IF SO: BR
3962  016310  004737  022126               JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
3963  016314  004737  016504       DPC0A:  JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
3964  016320  000413                       BR      DPC2A
3965  016322  005277  162400       DPC1:   INC     @BPKP           ;BUMP PICK CNTR
3966  016326  100027                       BPL     DPC2            ;& BR IF NO OVERFLOW
3967  016330  032777  002000  162260       BIT     #2000,@SWR      ;SEE IF HAVE PRINTED DATA
3968  016336  001402                       BEQ     DPC1A           ;IF SO: BR
```

```
3969  016340  004737  022126            JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
3970  016344  004737  016504    DPC1A:  JSR     PC,DPPRT        ;PRINT DROPS AND PICKS
3971  016350  013704  000702    DPC2A:  MOV     UNP,R4
3972  016354  016403  001014            MOV     DRP1(R4),R3     ;SET DROP POINTER
3973  016360  016404  000774            MOV     PIK1(R4),R4     ;SET PICK POINTER
3974  016364  012737  000010  000716    MOV     #10,BCNT        ;SET NUMBER OF BITS
3975  016372  005023            DPC2B:  CLR     (R3)+           ;CLEAR DROPS
3976  016374  005024                    CLR     (R4)+           ;CLEAR PICK
3977  016376  005337  000716            DEC     BCNT            ;SEE IF DONE
3978  016402  001375                    BNE     DPC2B           ;IF NOT: BR
3979  016404  000207                    RTS     PC              ;EXIT
3980  016406  000241            DPC2:   CLC
3981  016410  106003                    RORB    R3              ;GET NEXT BIT
3982  016412  005337  000716            DEC     BCNT            ;SEE IF DONE
3983  016416  001407                    BEQ     DPC3
3984  016420  062737  000002  000726    ADD     #2,BPKP
3985  016426  062737  000002  000724    ADD     #2,BDPP
3986  016434  000710                    BR      DPC0            ;CONTINUE
3987  016436  000207            DPC3:   RTS     PC              ;RETURN
3988  016440  013704  000702    PICK:   MOV     UNP,R4          ;GET UNIT POINTER
3989  016444  016437  000774  000726    MOV     PIK1(R4),BPKP   ;SET PICK POINTER
3990  016452  016437  001014  000724    MOV     DRP1(R4),BDPP   ;SET DROP POINTER
3991  016460  113704  000652            MOVB    TEMP1,R4        ;R4 = GOOD CHAR
3992  016464  113703  000654            MOVB    TEMP2,R3        ;R3 = BAD CHAR
3993  016470  112737  000001  000656    MOVB    #1,TEMP3        ;SET PICK FLAG
3994  016476  004737  016242            JSR     PC,DPC          ;GO CHECK PICKS
3995  016502  000207                    RTS     PC              ;EXIT
3996  016504  000004  024333    DPPRT:  TYPE,MSG26              ;TYPE 'DROPS'
3997  016510  013704  000702            MOV     UNP,R4
3998  016514  016437  001014  000724    MOV     DRP1(R4),BDPP   ;SET DROP POINTER
3999  016522  016437  000774  000726    MOV     PIK1(R4),BPKP   ;SET PICK POINTER
4000  016530  062737  000016  000724    ADD     #16,BDPP
4001  016536  062737  000016  000726    ADD     #16,BPKP
4002  016544  012737  000010  000716    MOV     #10,BCNT        ;SET NUMBER TO PRINT
4003  016552  017703  162146    DPPRT0: MOV     @BDPP,R3
4004  016556  104400                    TYPOCT                  ;PRINT DROPS
4005  016560  005337  000716            DEC     BCNT            ;SEE IF DONE
4006  016564  001404                    BEQ     DPPRT1          ;IF NOT: BR
4007  016566  162737  000002  000724    SUB     #2,BDPP         ;BUMP POINTER
4008  016574  000766                    BR      DPPRT0          ;CONTINUE FOR ALL 8 BITS
4009  016576  012737  000010  000716  DPPRT1: MOV     #10,BCNT        ;SET NUMBER TO PRINT
4010  016604  000004  024344            TYPE,MSG27              ;TYPE 'PICKS'
4011  016610  017703  162112    DPPRT2: MOV     @BPKP,R3
4012  016614  104400                    TYPOCT                  ;PRINT PICKS
4013  016616  005337  000716            DEC     BCNT            ;SEE IF DONE
4014  016622  001404                    BEQ     DPPRTX          ;IF SO: BR
4015  016624  162737  000002  000726    SUB     #2,BPKP         ;BUMP POINTER
4016  016632  000766                    BR      DPPRT2          ;CONTINUE FOR ALL 8 BITS
4017  016634  000207            DPPRTX: RTS     PC              ;RETURN
```

```
                                        ;*********************************************************
                                        ;STATUS CHECK SUBROUTINE:
                                        ;
                                        ;THIS SUBROUTINE IS USED TO PERFORM A CHECK OF
                                        ;BOTH THE MASSBUS CONTROLLER (RH11) AND THE TAPE
                                        ;CONTROLLER (TM02).  THE RH11 IS CHECKED FOR ERRORS
                                        ;AS REFLECTED IN REGISTERS CS1 AND CS2 AND ALSO THAT
                                        ;THE BUS ADDRESS (BA) AND WORD COUNT (WC) ARE
                                        ;CORRECT.  THE TM02 IS CHECKED FOR DRIVE STATIS (DS),
                                        ;DRIVE ERRORS (ER), AND PROPER FRAME COUNT.  THE SPECIAL
                                        ;CHECK CHARACTERS (CRC+LRC) ARE ALSO CHECKED WHEN
                                        ;APPROPIATE (IE: NRZ READ OR WRITE).  CERTAIN TYPES
                                        ;OF DRIVE ERRORS IN PE OPERATION WILL BE ACCOMPANIED
                                        ;BY THE DISPLAY OF THE DEAD TRACK REGISTER (CC).  THESE
                                        ;TYPES ARE ER BITS 15,10,7,6.  THE PRINTOUTS OF BAD
                                        ;CRC,LRC,FC, AND BA WILL SHOW BOTH THE EXPECTED AND
                                        ;RECEIVED VALUES (IE: EXPT-RCVD).  ONLY THOSE REGISTERS
                                        ;WHICH ARE IN ERROR WILL BE PRINTED AND ALL PRINTOUTS
                                        ;ARE IN OCTAL FORMAT WITH NO LEADING ZEROS.  AS IN
                                        ;DATA ERRORS, STATUS ERRORS ARE PRECEEDED BY HEADER
                                        ;DESCRIBING THE HARDWARE UNDER TEST, THE BLOCKING
                                        ;INFORMATION, AND THE ERROR TYPE.
                                        ;*********************************************************

016636  013703  000556          ERCHK:  MOV     FMCNT,R3        ;GET FRAME COUNT
016642  032703  000001                  BIT     #1,R3           ;SEE IF ODD
016646  001401                          BEQ     1$              ;IF NOT: BR
016650  005303                          DEC     R3              ;BUMP COUNT
016652  005403                  1$:     NEG     R3
016654  032737  000020  000552          BIT     #20,UDES        ;SEE IF CORE DUMP
016662  001401                          BEQ     2$              ;IF NOT: BR
016664  006203                          ASR     R3              ;SET TO FC/2
016666  032737  000010  000700  2$:     BIT     #10,MTC1        ;SEE IF WRITE OP
016674  001413                          BEQ     4$              ;IF SO: BR
016676  005737  000570                  TST     RDCMD
016702  001405                          BEQ     3$
016704  013703  000564                  MOV     @#RDATA,R3
016710  162703  000002                  SUB     #2,R3           ;SET POINTER
016714  000405                          BR      ER2
016716  063703  000564          3$:     ADD     @#RDATA,R3      ;BUILD EXPT READ ADDRESS
016722  000402                          BR      ER2
016724  063703  000562          4$:     ADD     @#WDATA,R3      ;BUILD EXPT WRITE ADDRESS

016730  032777  040000  161566  ER2:    BIT     #40000,@ER      ;BRANCH IF NOT UNSAFE
016736  001403                          BEQ     1$
016740  005726                          TST     (SP)+           ;ADJUST STACK
016742  000137  020426                  JMP     OFFLINE         ;GO MARK UNIT OFFLINE
016746  010337  020340          1$:     MOV     R3,CADER        ;SAVE ADDRESS
016752  012704  000007                  MOV     #7,R4
016756  012701  020342                  MOV     #BAER,R1
016762  005021                  2$:     CLR     (R1)+           ;CLEAR FLAGS
016764  005304                          DEC     R4
016766  001375                          BNE     2$
016770  020377  161520                  CMP     R3,@BA          ;SEE IF ADDRESS OK
016774  001402                          BEQ     3$              ;IF SO: BR
```

```
4075  016776  005237  020342                  INC     BAER            ;SET BUS ADDRESS ERROR
4076  017002  032737  000010  000700  3$:     BIT     #10,MTC1        ;SEE IF WRITE OPER
4077  017010  001006                          BNE     5$              ;IF NOT; BR
4078  017012  005777  161500          4$:     TST     @FC             ;SEE IF FC=0
4079  017016  001440                          BEQ     ER3             ;IF SO; BR
4080  017020  005237  020350                  INC     FCER            ;SET FC ERROR
4081  017024  000435                          BR      ER3
4082  017026  032737  000040  000700  5$:     BIT     #40,MTC1        ;SEE IF SPACE OPER
4083  017034  001766                          BEQ     4$              ;IF SO; BR
4084  017036  005737  000704                  TST     TMFLG           ;SEE IF TM TIME
4085  017042  001011                          BNE     7$              ;IF SO; BR
4086  017044  013703  000556                  MOV     FMCNT,R3
4087  017050  005403                          NEG     R3              ;R3 = EXPT RECORD SIZE
4088  017052  020377  161440          6$:     CMP     R3,@FC          ;SEE IF FC = EXPT
4089  017056  001420                          BEQ     ER3             ;IF SO; BR
4090  017060  005237  020350                  INC     FCER            ;SET FC ERROR FLAG
4091  017064  000415                          BR      ER3
4092  017066  032737  002000  000552  7$:     BIT     #2000,UDES      ;SEE IF PE
4093  017074  001346                          BNE     4$              ;IF SO; BR
4094  017076  005737  000570                  TST     RDCMD           ;SEE IF READ REVERSE
4095  017102  001003                          BNE     8$              ;IF SO; BR
4096  017104  012703  000002                  MOV     #2,R3
4097  017110  000760                          BR      6$              ;LOOK FOR EXPT = 2
4098  017112  012703  000001          8$:     MOV     #1,R3
4099  017116  000755                          BR      6$              ;GO CHECK FC FOR TM
4100
4101  017120  032777  160000  161362  ER3:    BIT     #160000,@C1     ;SEE IF COUNT ERROR
4102  017126  001437                          BEQ     ER4
4103  017130  017703  161364                  MOV     @CS,R3          ;GET CONT STATUS REG
4104  017134  042703  000307                  BIC     #307,R3         ;MASK OUT IR,OR,UNIT NO. & SEE IF OTHER ERRORS
4105  017140  001406                          BEQ     1$              ;IF NOT; BR
4106  017142  005737  000704                  TST     TMFLG           ;SEE IF TAPE MARK TIME
4107  017146  001425                          BEQ     3$              ;IF NOT; BR
4108  017150  042703  001000                  BIC     #1000,R3        ;MASK MISSED TRANS & BR IF OTHER ERRORS
4109  017154  001022                          BNE     3$
4110  017156  032777  060000  161324  1$:     BIT     #60000,@C1      ;SEE IF EITHER TRE OR MCPE
4111  017164  001420                          BEQ     ER4             ;IF NOT; BR
4112  017166  005737  000704                  TST     TMFLG           ;SEE IF TM TIME
4113  017172  001413                          BEQ     3$              ;IF NOT; BR
4114  017174  017703  161324                  MOV     @ER,R3          ;GET ERROR REGISTER
4115  017200  032737  000010  000552          BIT     #10,UDES        ;SEE IF EVEN PARITY
4116  017206  001402                          BEQ     2$              ;IF NOT; BR
4117  017210  042703  000100                  BIC     #100,R3         ;MASK PAR
4118  017214  042703  001000          2$:     BIC     #1000,R3        ;MASK FCE
4119  017220  001402                          BEQ     ER4             ;IF NO ERRORS EXCEPT FCE; BR
4120  017222  005237  020344          3$:     INC     CONER           ;SET CONT ERROR FLAG
4121
4122  017226  032777  040000  161266  ER4:    BIT     #40000,@DS      ;SEE IF DRIVE ERROR
4123  017234  001420                          BEQ     ER6             ;IF NOT; BR
4124  017236  005737  000704                  TST     TMFLG           ;SEE IF TAPE MARK TIME
4125  017242  001413                          BEQ     2$              ;IF NOT; BR
4126  017244  017703  161254                  MOV     @ER,R3          ;GET ER
4127  017250  032737  000010  000552          BIT     #10,UDES        ;SEE IF EVEN PARITY
4128  017256  001402                          BEQ     1$              ;IF NOT; BR
4129  017260  042703  000100                  BIC     #100,R3         ;MASK PAR
4130  017264  042703  001000          1$:     BIC     #1000,R3        ;MASK OUT FCE & BRANCH IF
```

M7


```
4131  017270  001402                          BEQ     ER6             ;NO OTHER ERRORS
4132  017272  005237  020346          2$:     INC     DRVER           ;SET DRIVER ERROR FLAG
4133
4134  017276  013737  015060  020362  ER6:    MOV     EXCRC,CRCSV     ;SAVE EXPECTED CRC
4135  017304  013737  015062  020360          MOV     EXLRC,LRCSV     ;AND EXPECTED LRC
4136  017312  032737  002000  000552          BIT     #2000,UDES
4137  017320  001062                          BNE     ERPT            ;IF IN PE MODE: BR
4138  017322  032777  020000  161266          BIT     #20000,@SWR     ;SEE IF NO DATA CHECK
4139  017330  001056                          BNE     ERPT            ;IF NOT: BR (ALLOW READ OF UNKNOWN TAPES)
4140  017332  032737  000040  000700          BIT     #40,MTC1        ;SEE IF WRITE OR READ OP
4141  017340  001452                          BEQ     ERPT            ;IF NOT: BR
4142  017342  005737  000704                  TST     TMFLG           ;SEE IF TAPE MARK TIME
4143  017346  001405                          BEQ     1$              ;IF NOT: BR
4144  017350  005037  015060                  CLR     EXCRC
4145  017354  012737  000023  015062          MOV     #23,EXLRC       ;SET CRC/LRC FOR TM
4146  017362  032737  000060  000552  1$:     BIT     #60,UDES        ;SEE IF FORMAT 14
4147  017370  001036                          BNE     ERPT            ;IF NOT: BR
4148  017372  017703  161132                  MOV     @CC,R3          ;GET CRC CHARACTER
4149  017376  042703  177000                  BIC     #177000,R3
4150  017402  023703  015060                  CMP     EXCRC,R3
4151  017406  001402                          BEQ     2$              ;IF CRC GOOD: BR
4152  017410  005237  020354                  INC     CRCER           ;SET ERROR FLAG
4153  017414  017703  161114          2$:     MOV     @MR,R3          ;GET LRC
4154  017420  000303                          SWAB    R3
4155  017422  005703                          TST     R3
4156  017424  100002                          BPL     3$
4157  017426  052703  000400                  BIS     #400,R3
4158  017432  042703  177000          3$:     BIC     #177000,R3
4159  017436  023703  015062                  CMP     EXLRC,R3
4160  017442  001411                          BEQ     ERPT            ;IF LRC GOOD: BR
4161  017444  010337  020356                  MOV     R3,ACTLRC       ;SAVE ACTUAL LRC
4162  017450  005237  020352                  INC     LRCER           ;SET LRC ERROR FLAG
4163  017454  005737  000570                  TST     RDCMD           ;SEE IF READ REVERSE
4164  017460  001402                          BEQ     ERPT            ;IF NOT: BR
4165  017462  005037  020352                  CLR     LRCER           ;ELSE CLEAR LRC ERROR
4166  017466  012703  000006          ERPT:   MOV     #6,R3
4167  017472  005037  000714                  CLR     SERFL           ;CLEAR ERROR FLAG
4168  017476  005037  000730                  CLR     ERSAV
4169  017502  012704  020342                  MOV     #BAER,R4
4170  017506  005724                  ERPTT:  TST     (R4)+           ;SEE IF ANY ERROR
4171  017510  001004                          BNE     ERPTG           ;IF SO: BR
4172  017512  005303                          DEC     R3
4173  017514  001374                          BNE     ERPTT
4174  017516  000137  020304                  JMP     ERPX1
4175  017522  005237  000714          ERPTG:  INC     SERFL           ;SET ERROR FLAG
4176  017526  017737  160772  000730          MOV     @ER,ERSAV       ;SAVE ERROR REGISTER
4177  017534  032777  002000  161054          BIT     #2000,@SWR      ;SEE IF PRINT
4178  017542  001420                          BEQ     ERPTO           ;IF SO: BR
4179  017544  022737  000002  000720          CMP     #2,RTYFL        ;SEE IF READ RETRY
4180  017552  001006                          BNE     ERPTG1          ;IF NOT: BR
4181  017554  013703  000710                  MOV     RTCNT,R3
4182  017560  005203                          INC     R3              ;BUMP RETRY COUNT
4183  017562  020337  000612                  CMP     R3,RETRY        ;SEE IF LAST RETRY
4184  017566  001406                          BEQ     ERPTO           ;IF SO: BR
4185  017570  022737  000002  020346  ERPTG1: CMP     #2,DRVER        ;SEE IF TM STATUS ERROR
4186  017576  001402                          BEQ     ERPTO           ;IF SO: BR
```

```
 4187  017600  000137  020164                    JMP     ERPX0
 4188  017604  005237  000676          ERPT0:    INC     PFLG
 4189  017610  004737  022126                    JSR     PC,PAPRT        ;PRINT HEADER
 4190  017614  013737  000660  017624            MOV     EMADDR,1$       ;GET ADDRESS OF ERROR MSG HEADER
 4191  017622  000004                            TYPE
 4192  017624  000000          1$:               .WORD   0               ;ADDRESS OF ERROR MESSAGE HEADER
 4193  017626  004737  020364                    JSR     PC,FRPRT        ;PRINT F OR R
 4194  017632  005737  000704                    TST     TMFLG
 4195  017636  001406                            BEQ     ERPT1
 4196  017640  022737  025212  000660            CMP     #MSG54,EMADDR
 4197  017646  001402                            BEQ     ERPT1
 4198  017650  000004  025230                    TYPE,MSG56              ;TYPE 'TM'
 4199  017654  005737  020344          ERPT1:    TST     CONER
 4200  017660  001412                            BEQ     ERPT2           ;IF NO CONT ERROR: BR
 4201  017662  000004  025137                    TYPE,MSG23              ;TYPE'CS1'
 4202  017666  017703  160616                    MOV     @C1,R3
 4203  017672  104400                            TYPOCT                  ;PRINT CONTROL 1
 4204  017674  000004  024164                    TYPE,MSG23D             ;TYPE CS TAG
 4205  017700  017703  160614                    MOV     @CS,R3
 4206  017704  104400                            TYPOCT                  ;PRINT CONT STATUS
 4207  017706  005737  020346          ERPT2:    TST     DRVER
 4208  017712  001412                            BEQ     ERPT3           ;IF SO DRIVE ERROR: BR
 4209  017714  000004  024172                    TYPE,MSG23E             ;TYPE DS TAG
 4210  017720  017703  160576                    MOV     @DS,R3
 4211  017724  104400                            TYPOCT                  ;PRINT DRIVE STATUS
 4212  017726  000004  024177                    TYPE,MSG23F             ;TYPE ER TAG
 4213  017732  017703  160566                    MOV     @ER,R3
 4214  017736  104400                            TYPOCT                  ;PRINT DRIVE ERROR
 4215  017740  005737  020342          ERPT3:    TST     BAER
 4216  017744  001412                            BEQ     ERPT4           ;IF NO BA ERROR: BR
 4217  017746  000004  024152                    TYPE,MSG23B             ;TYPE BA TAG
 4218  017752  017703  160536                    MOV     @BA,R3
 4219  017756  104400                            TYPOCT                  ;PRINT BUS ADDRESS
 4220  017760  000004  023720                    TYPE,DASH
 4221  017764  013703  020340                    MOV     CADER,R3
 4222  017770  104400                            TYPOCT                  ;PRINT EXPT BUS ADDRESS
 4223  017772  005737  020350          ERPT4:    TST     FCER
 4224  017776  001405                            BEQ     ERPT5           ;IF NO FC ERROR: BR
 4225  020000  000004  024157                    TYPE,MSG23C             ;TYPE FC TAG
 4226  020004  017703  160506                    MOV     @FC,R3
 4227  020010  104400                            TYPOCT                  ;PRINT FRAME COUNT
 4228  020012  000004  024145          ERPT5:    TYPE,MSG23A             ;TYPE WC TAG
 4229  020016  017703  160470                    MOV     @WC,R3
 4230  020022  104400                            TYPOCT                  ;PRINT WORD COUNT
 4231  020024  005737  020354                    TST     CRCER
 4232  020030  001414                            BEQ     ERPT5A          ;IF NO CRC ERROR: BR
 4233  020032  000004  025255                    TYPE,MSG58              ;TYPE CRC TAG
 4234  020036  017703  160466                    MOV     @CC,R3
 4235  020042  042703  177000                    BIC     #177000,R3
 4236  020046  104400                            TYPOCT                  ;PRINT ACTUAL CRC
 4237  020050  000004  023720                    TYPE,DASH
 4238  020054  013703  015060                    MOV     EXCRC,R3
 4239  020060  104400                            TYPOCT                  ;PRINT EXPECTED CRC
 4240  020062  005737  020352          ERPT5A:   TST     LRCER
 4241  020066  001412                            BEQ     ERPT6           ;IF NO LRC ERROR: BR
 4242  020070  000004  025263                    TYPE,MSG59              ;TYPE LRC ERR TAG
```

```
4243  020074  013703  020356          MOV     ACTLRC,R3
4244  020100  104400                  TYPOCT                  ;PRINT ACTUAL LRC
4245  020102  000004  023720          TYPE,DASH
4246  020106  013703  015062          MOV     EXLRC,R3
4247  020112  104400                  TYPOCT                  ;PRINT EXPECTED LRC
4248  020114  005737  020346  ERPT6:  TST     DRVER
4249  020120  001420                  BEQ     ERPT7           ;IF NO DRIVE ERROR; BR
4250  020122  032737  002000  000552  BIT     #2000,UDES
4251  020130  001414                  BEQ     ERPT7           ;IF NO PE; BR
4252  020132  017704  160366          MOV     @ER,R4
4253  020136  042704  075477          BIC     #75477,R4       ;MASK OUT ALL BUT BITS 15,10,7,6
4254  020142  001407                  BEQ     ERPT7           ;IF NO CONDITIONALS SET; BR
4255  020144  000004  024211          TYPE,MSG23H             ;TYPE CC TAG
4256  020150  017703  160354          MOV     @CC,R3
4257  020154  042703  177000          BIC     #177000,R3      ;MASK CC
4258  020160  104400                  TYPOCT                  ;PRINT CHECK CHARACTERS
4259  020162  000240          ERPT7:  NOP
4260  020164  005777  160426  ERPX0:  TST     @SWR            ;BRANCH IF NOT HALT ON ERROR
4261  020170  100012                  BPL     ERPX
4262  020172  000000                  HALT
4263  020174  005737  000676          TST     PFLG            ;SEE IF HAVE PRINTED
4264  020200  001006                  BNE     ERPX            ;IF SO: BR
4265  020202  032777  002000  160406  BIT     #2000,@SWR      ;SEE IF SHOULD PRINT
4266  020210  001002                  BNE     ERPX            ;IF NOT: BR
4267  020212  000137  017604          JMP     ERPTO           ;PRINT ERROR
4268  020216  005037  000676  ERPX:   CLR     PFLG
4269  020222  005737  000574          TST     CRCC            ;BRANCH IF CRC ERROR
4270  020226  001007                  BNE     1$              ;CORRECTION DESIRED
4271  020230  012777  000040  160262  MOV     #40,@CS         ;ELSE INIT
4272  020236  013777  000550  160254  MOV     DVN,@CS         ;RESET DRIVE NUMBER
4273  020244  000414                  BR      2$
4274  020246  012777  000011  160234  1$:     MOV     #11,@C1         ;DRIVE CLEAR
4275  020254  017704  160246          MOV     @AS,R4
4276  020260  010477  160242          MOV     R4,@AS          ;CLEAR AS
4277  020264  013704  000510          MOV     C1,R4
4278  020270  005204                  INC     R4
4279  020272  152714  000100          BISB    #100,(R4)       ;RESET TRE
4280  020276  013777  000552  160236  2$:     MOV     UDES,@TC        ;RESET TC
4281  020304  032737  000040  000700  ERPX1:  BIT     #40,MTC1
4282  020312  001411                  BEQ     ERPX2           ;IF NOT READ/WRITE OP; BR
4283  020314  005737  000704          TST     TMFLG
4284  020320  001406                  BEQ     ERPX2           ;IF NOT TM TIME; BR
4285  020322  013737  020362  015060  MOV     CRCSV,EXCRC     ;RESTORE CRC
4286  020330  013737  020360  015062  MOV     LRCSV,EXLRC     ;RESTORE LRC
4287  020336  000207          ERPX2:  RTS     PC              ;EXIT
4288  020340  000000          CADER:  0                       ;EXPT ADDRESS SAVE
4289  020342  000000          BAER:   0
4290  020344  000000          CONER:  0
4291  020346  000000          DRVER:  0
4292  020350  000000          FCER:   0
4293  020352  000000          LRCER:  0
4294  020354  000000          CRCER:  0
4295  020356  000000          ACTLRC: 0
4296  020360  000000          LRCSV:  0
4297  020362  000000          CRCSV:  0
4298
```

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

020364  052737  000010  000700  FRPRT:  BIT     #10,MTC1        ;SEE IF WRITE COMMAND
020370  001414                          BEQ     3$              ;IF SO: BR
020372  012737  024055  020422          MOV     #MSG17,2$       ;PRSET MESSAGE TO READ REVERSE
020400  052737  000002  000700          BIT     #2,MTC1         ;BRANCH IF REVERSE
020410  001003                          BNE     1$
020412  012737  024052  020422          MOV     #MSG16,2$       ;SET FORWARD MESSAGE
020420  000004                  1$:     TYPE                    ;TYPE MSG
020422  000000                  2$:     .WORD   0
020424  000207                  3$:     RTS     PC              ;EXIT


                                        ;ROUTINE TO MARK UNIT OFF LINE

020426  013701  000702          OFFLINE:MOV     UNP,R1  ;GET UNIT POINTER
020432  052761  040000  000752          BIS     #40000,UN1(R1)  ;MARK UNIT OFF LINE
020440  000004  024261                  TYPE,MSG2%              ;TYPE 'SLAVE UNSAFE NO FURTHER TESTING ON SLAVE
020444  005737  000742                  TST     ASEQF           ;BRANCH IF NOT IN AUTO SEQUENCE
020450  001406                          BEQ     1$
020452  000004  026446                  TYPE,MSG125             ;TYPE 'AUTO SEQ TEST WILL RESTART
020456  012706  000500                  MOV     #500,SP         ;RESET STACK PTR
020462  000137  021342                  JMP     ASEQ0           ;RESTART AUTO-SEQ
020466  105337  005043          1$:     DECB    REOTC+1         ;DECREMENT UNITS TO TEST CTR
020472  001003                          BNE     2$
020474  000004  026410                  TYPE,MSG122             ;TYPE 'NO UNITS LEFT TO TEST; HALT
020500  000000                          HALT
020502  000137  004266          2$:     JMP     REOT
```

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

020506  005037  000652          TAPG:   CLR     TEMP1
020512  013777  000550  160000          MOV     DVN,@CS         ;SET DRIVE NO.
020520  032777  040000  157776  1$:     BIT     #40000,@ER      ;SEE IF UNIT SAFE
020526  001402                          BEQ     TAPG3           ;IF SO: BR
020530  000137  020426                  JMP     OFFLINE         ;GO MARK UNIT OFF-LINE
020534  032777  020000  157760  TAPG3:  BIT     #20000,@DS      ;SEE IF PIP RESET
020542  001410                          BEQ     TAPG3F          ;IF SO: BR
020544  004737  022126                  JSR     PC,PAPRT        ;PRINT HEADER
020550  000004  026244                  TYPE,MSG116             ;TYPE MSG
020554  032777  020000  157740  1$:     BIT     #20000,@DS
020562  001374                          BNE     1$              ;AWAIT PIP RESET
020564  022737  000026  000700  TAPG3F: CMP     #26,MIC1        ;SEE IF WRITE TM
020572  001003                          BNE     TAPG3A          ;IF NOT: BR
020574  012704  177777                  MOV     #-1,R4          ;ELSE SET FC FOR -1
020600  000406                          BR      TAPG3B
020602  013704  000556          TAPG3A: MOV     FMCNT,R4
020606  032704  000001                  BIT     #1,R4
020612  001401                          BEQ     TAPG3B
020614  005304                          DEC     R4
020616  000261          TAPG3B: SEC
020620  006004                          ROR     R4              ;SET WC = FC/2 FOR NORMAL FORMAT
020622  032737  000020  000552          BIT     #20,UDES        ;SEE IF CORE DUMP FORMAT
020630  001402                          BEQ     TAPG3C          ;IF NOT: BR
020632  000261                          SEC
020634  006004                          ROR     R4              ;SET WC = FC/4 FOR CORE DUMP
020636  010477  157650          TAPG3C: MOV     R4,@WC          ;SET WORD COUNT
020642  012777  000011  157640          MOV     #11,@C1         ;DRIVE CLEAR
```

```
4393  020650  017777  157642  157640          MOV     @FC,@FC         ;RESET FC LOADED
4394  020656  005737  000576                  TST     INTRF           ;SEE IF INTERCHANGE READ
4395  020662  001407                          BEQ     TAPG3D          ;IF NOT: BR
4396  020664  032737  000040  000700          BIT     #40,MTC1        ;SEE IF READ OP
4397  020672  001403                          BEQ     TAPG3D          ;IF NOT: BR
4398  020674  012777  000003  157632          MOV     #3,@MR          ;SET INTERCHANGE READ MAINT. MODE
4399  020702  013704  000700          TAPG3D: MOV     MTC1,R4         ;GET COMMAND
4400  020706  042704  177707                  BIC     #177707,R4      ;MASK OP CODE
4401  020712  022704  000030                  CMP     #30,R4          ;SEE IF SPACE OP CODE
4402  020716  001403                          BEQ     TAPG3E          ;IF SO: BR
4403  020720  012737  177740  000674          MOV     #-40,STAL       ;SET INTERRUPT DELAY MULT TO 40
4404  020726  052737  000101  000700  TAPG3E: BIS     #101,MTC1       ;SET INTERRUPT ENABLE AND GO
4405  020734  000240                          NOP
4406  020736  013777  000700  157544          MOV     MTC1,@C1        ;EXECUTE COMMAND
4407  020744  005077  157644                  CLR     @PSW            ;CLEAR PRIORITY
4408  020750  005037  000652                  CLR     TEMP1
4409  020754  005237  000652          TAPG4:  INC     TEMP1           ;SEE IF HAVE TIMED OUT
4410  020760  001375                          BNE     TAPG4           ;IF NOT: BR
4411  020762  005237  000674                  INC     STAL
4412  020766  001372                          BNE     TAPG4           ;DO TIME DELAY MULTIPLIER
4413  020770  012777  000340  157616  TAPG5:  MOV     #340,@PSW       ;RESET PRIORITY
4414  020776  032777  002000  157612          BIT     #2000,@SWR      ;SEE IF SHOULD PRINT ERRORS
4415  021004  001013                          BNE     TAPG6           ;IF NOT: BR
4416  021006  004737  022126                  JSR     PC,PAPRT        ;PRINT CYCLE NUMBER
4417  021012  013737  000660  021022          MOV     EMADDR,1$
4418  021020  000004                          TYPE                    ;TYPE MSG
4419  021022  000000                  1$:     .WORD   0
4420  021024  004737  020364                  JSR     PC,FRPRT        ;PRINT F OR R
4421  021030  000004  024242                  TYPE,MSG24              ;TYPE 'NO INTERRUPT'
4422  021034  005777  157556          TAPG6:  TST     @SWR            ;BRANCH IF NOT HALT ON ERROR
4423  021040  100001                          BPL     TAPG7
4424  021042  000000                          HALT
4425  021044  000137  021276          TAPG7:  JMP     MTINTA          ;RETURN TO CALLING ROUTINE
4426
```

```
4428
4429                                        ;TTY INTERRUPT HANDLER
4430  021050  017746  157546        TTINT:  MOV     @TKB,-(SP)      ;GET CHARACTER
4431  021054  042716  000200                BIC     #200,(SP)       ;STRIP PARITY BIT
4432  021060  122716  000003                CMPB    #3,(SP)         ;BRANCH IF NOT †C
4433  021064  001005                        BNE     1$
4434  021066  000005                        RESET                   ;RESET ALL I/O
4435  021070  005077  157520                CLR     @PSW            ;CLEAR PSW
4436  021074  000137  000200                JMP     @#200           ;RESTART PROGRAM
4437  021100  122716  000001        1$:     CMPB    #1,(SP)         ;BRANCH IF NOT †A
4438  021104  001015                        BNE     2$
4439  021106  022737  000176  000616        CMP     #SWREG,SWR      ;BRANCH IF HARDWARE SWR IS INVOKED
4440  021114  001014                        BNE     3$
4441  021116  012737  177570  000616        MOV     #177570,SWR     ;INVOKE HARWARE SWR
4442  021124  004737  022546                JSR     PC,.SAVE        ;SAVE REGISTERS ON THE STACK
4443  021130  000004  026362                TYPE,MSG121             ;TYPE 'HARDWARE SWR IN USE'
4444  021134  004737  022570                JSR     PC,.RESTORE     ;RESTORE REGISTERS
4445  021140  022716  000007        2$:     CMP     #7,(SP)         ;BRANCH IF NOT †G
4446  021144  001005                        BNE     4$
4447  021146  012737  000176  000616  3$:   MOV     #SWREG,SWR      ;INVOKE SOFTWARE SWR
4448  021154  004737  022462                JSR     PC,GTSWR        ;GET SWITCHES
4449  021160  022716  000002        4$:     CMP     #2,(SP)         ;BRANCH IF NOT †B
4450  021164  001041                        BNE     6$
4451  021166  004737  022546                JSR     PC,.SAVE        ;SAVE REGISTERS ON THE STACK
4452  021172  005237  013560                INC     SCVFL           ;SET FLAG
4453  021176  004737  013342                JSR     PC,TINP3A       ;GO CHECK CRC CORRECTION
4454  021202  032777  000040  157406        BIT     #40,@SWR        ;BRANCH IF NOT YOZZLING
4455  021210  001425                        BEQ     5$
4456  021212  000004  025052                TYPE,MSG44              ;REQUEST NEW YOZZLE STALL
4457  021216  013703  000610                MOV     YSTAL,R3
4458  021222  104400                        TYPOCT                  ;PRINT PRESENT STALL
4459  021224  012705  000610                MOV     #YSTAL,R5       ;SET ADDRESS OF YSTL
4460  021230  012701  000007                MOV     #7,R1           ;SET NUMBER OF CHAR TO INPUT
4461  021234  012702  177777                MOV     #-1,R2          ;SET MAXIMUM LIMIT
4462  021240  012703  002000                MOV     #2000,R3        ;SET MINIMUM LIMIT
4463  021244  004737  022612                JSR     PC,TTR          ;GO GET VALUE
4464  021250  004737  022570                JSR     PC,.RESTORE     ;RESTORE REGISTERS
4465  021254  005726                        TST     (SP)+           ;POP CHARACTER OF THE STACK
4466  021256  012716  011102                MOV     #YOZ,(SP)       ;RETURN TO 'YOZ'
4467  021262  000002                        RTI                     ;RETURN TO YOZ
4468  021264  004737  022570        5$:     JSR     PC,.RESTORE
4469  021270  005726                6$:     TST     (SP)+           ;POP CHARACTER OFF THE STACK
4470  021272  000002                        RTI                     ;RETURN
4471
4472                                        ;MAG TAPE INTERRUPT HANDLER
4473  021274  000240                MTINT:  NOP
4474  021276  042777  000037  157230  MTINTA: BIC     #37,@MR         ;CLEAR MAINT MODE
4475  021304  013716  000670                MOV     RTRN,(SP)       ;SET RETURN TO (RTRN)
4476  021310  000002                        RTI                     ;RETURN
```

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

021312  000004  025677          ASEQ:   TYPE,MSG104             ;REQUEST 'AUTO CONT'
021316  012705  000746                  MOV     #ASEQCF,R5      ;SET ADDRESS OF ENTRY
021322  012701  000002                  MOV     #2,R1           ;SET SIZE OF ENTRY
021326  012702  000001                  MOV     #1,R2           ;SET UPPER LIMIT
021332  012703  000000                  MOV     #0,R3           ;SET LOWER LIMIT
021336  004737  022612                  JSR     PC,TTR          ;GO GET INPUT
021342  005037  000550          ASEQ0:  CLR     DVN             ;SET DRIVE # 0
021346  004737  021454          ASEQ1:  JSR     PC,HRDS         ;GO SELECT HARDWARE CONFIGURATION
021352  000004  025647                  TYPE,MSG101             ;TYPE '*******...***'
021356  000004  025156                  TYPE,MSG52A             ;TYPE 'DRIVE (TM03) = '
021362  013703  000550                  MOV     DVN,R3
021366  104400                          TYPOCT                  ;PRINT DRIVE #
021370  000004  026535                  TYPE,SPACE
021374  000004  024600                  TYPE,MSG32              ;TYPE ' SLAVE # = '
021400  012700  000752                  MOV     #UN1,R0         ;POINT TO START OF SLAVE TABLE
021404  012003                  1$:     MOV     (R0)+,R3
021406  100402                          BMI     2$
021410  104400                          TYPOCT                  ;PRINT SLAVE TABLE
021412  000774                          BR      1$              ;DO ALL
021414  004737  021640          2$:     JSR     PC,AMOD1        ;GO DO MODE 1(NRZ)
021420  004737  021772                  JSR     PC,AMOD2        ;GO DO MODE 2(PE)
021424  022737  000007  000550  ASEQ4:  CMP     #7,DVN          ;SEE IF DONE ALL DRIVES
021432  001403                          BEQ     ASEQX           ;IF SO: BR
021434  005237  000550                  INC     DVN             ;BUMP DRIVE NUMBER
021440  000742                          BR      ASEQ1           ;CONTINUE
021442  005737  000746          ASEQX:  TST     ASEQCF          ;SEE IF CONTINUOUS AUTO SEQ
021446  001335                          BNE     ASEQ0           ;++B CONTINUE TESTING
021450  000137  004774                  JMP     TEND
```

```
4516
4517                                        ;SUBROUTINE TO SELECT AUTO SEQUENCE HARDWARE********
4518
4519  021454  005037  005042        HRDS:   CLR     REOTC           ;CLEAR EOT UNIT CNTR
4520  021460  012777  000040  157032        MOV     #40,@CS         ;INIT
4521  021466  013777  000550  157024        MOV     DVN,@CS         ;SET DRIVE
4522  021474  005777  157010                TST     @C1             ;ACCESS DRIVE
4523  021500  032777  010000  157012        BIT     #10000,@CS      ;TEST FOR NON-EXISTANT DRIVE
4524  021506  001403                        BEQ     2$              ;IF DRIVE AVAIL: BR
4525  021510  005726                1$:     TST     (SP)+           ;RESET STACK POINTER
4526  021512  000137  021424                JMP     ASEQ4           ;GO SEE IF TRIED ALL DRIVES
4527  021516  017700  157014        2$:     MOV     @DT,R0          ;++B GET CONTENTS OF DRIVE TYPE REG
4528  021522  042700  002007                BIC     #2007,R0        ;++B CLEAR SPR AND SPEED BITS
4529  021526  022700  140050                CMP     #140050,R0      ;++B BRANCH IF NOT TM03 MAGTAPE DRIVE
4530  021532  001366                        BNE     1$
4531  021534  005000                        CLR     R0
4532  021536  012701  000752                MOV     #UN1,R1         ;SET START OF SLAVE TABLE
4533  021542  005737  003146                TST     CHNFLG          ;BRANCH IF NOT IN CHAIN MODE
4534  021546  001410                        BEQ     3$
4535  021550  122737  000006  000041        CMPB    #6,@#41         ;BRANCH IF NOT LOADED VIA TMDP
4536  021556  001004                        BNE     3$
4537  021560  005737  000550                TST     DVN             ;BRANCH IF NOT DRIVE 0
4538  021564  001001                        BNE     3$
4539  021566  005200                        INC     R0              ;DO NOT TEST SLAVE 0
4540  021570  010077  156746        3$:     MOV     R0,@TC          ;SELECT SLAVE
4541  021574  032777  010000  156720        BIT     #10000,@DS      ;SEE IF SLAVE AVAIL FOR TEST(MOL)
4542  021602  001404                        BEQ     4$              ;IF NOT: BR
4543  021604  062737  000401  005042        ADD     #401,REOTC      ;INCREMENT UNITS TO TEST COUNT
4544  021612  010021                        MOV     R0,(R1)+        ;LOAD SLAVE # INTO SLAVE TABLE
4545  021614  005200                4$:     INC     R0              ;STEP TO NEXT SLAVE
4546  021616  022700  000010                CMP     #10,R0          ;BRANCH IF ALL SLAVE NOT DONE
4547  021622  001362                        BNE     3$
4548  021624  005737  005042        5$:     TST     REOTC           ;SEE IF FOUND ANY SLAVES
4549  021630  001727                        BEQ     1$              ;IF NOT: BR
4550  021632  012711  177777                MOV     #-1,(R1)        ;TERMINATE SLAVE TABLE
4551  021636  000207                        RTS     PC              ;RETURN TO SEQ
```

```
4553
4554                                    ;SUBROUTINE TO SELECT NRZ AUTO TEST MODE**********
4555
4556  021640  005037  000662      AMOD1:  CLR     BLCNTR          ;ASSURE BLOCK COUNTER IS 0
4557  021644  012701  000752              MOV     #UN1,R1         ;GET START OF SLAVE TABLE
4558  021650  052721  001700      1$:     BIS     #1700,(R1)+     ;SET ALL SLAVE TO NRZ,NORM,ODD
4559  021654  022711  177777              CMP     #-1,(R1)        ;LOOP UNTIL REACED END OF TABLE
4560  021660  001373                      BNE     1$
4561  021662  004737  005056              JSR     PC,RWNDA        ;GO REWIND ALL AVAIL SLAVES
4562  021666  012737  000006  000744      MOV     #6,ABLCNT       ;SET NUMBER OF BLOCKS FOR MODE 1
4563  021674  013737  000560  000556      MOV     BUFMAX,FMCNT    ;SET FC = MAX BUFFER SIZE
4564  021702  012737  000100  000554      MOV     #100,RCNT       ;SET REC CNTR = 100
4565  021710  012737  000001  000566      MOV     #1,PATRN        ;SELECT PATTERN 1
4566  021716  005037  000572              CLR     TMEX            ;ASSURE NO TMK
4567  021722  005037  000576              CLR     INTRF           ;ASSURE NORMAL READ
4568  021726  004737  003464              JSR     PC,STAUTO       ;GO DO AUTO MODE 1
4569  021732  012737  000010  000566      MOV     #10,PATRN       ;SELECT PATTERN 10
4570  021740  004737  003464              JSR     PC,STAUTO       ;GO DO PATTERN 10
4571  021744  012737  000014  000566      MOV     #14,PATRN       ;SELECT PATTERN 14
4572  021752  004737  003464              JSR     PC,STAUTO
4573  021756  012737  177777  000566  3$: MOV     #-1,PATRN       ;SELECT AUTO RANDOM DATA
4574  021764  004737  003464              JSR     PC,STAUTO
4575  021770  000207                      RTS     PC              ;RETURN TO SEQ
```

```
                                                    ;SUBROUTINE TO SELECT PE AUTO TEST MODE**********

021772  005037  000662          AMOD2:  CLR     BLCNTR                  ;CLEAR BLOCK CNTR
021776  012701  000752                  MOV     #UN1,R1                 ;SET START OF SLAVE TABLE
022002  042711  001700          1$:     BIC     #1700,(R1)              ;CLEAR NRZ
022006  052721  002300                  BIS     #2300,(R1)+             ;SET TO PE NORM, ODD
022012  022711  177777                  CMP     #-1,(R1)                ;LOOP UNTIL END OF TABLE
022016  001371                          BNE     1$
022020  004737  005056                  JSR     PC,RWNDA                ;REWIND ALL SLAVES
022024  012737  000006  000744          MOV     #6,ABLCNT               ;SET AUTO BLOCK COUNT
022032  013737  000560  000556          MOV     BUFMAX,FMCNT            ;SET FC = MAX BUFFER SIZE
022040  012737  000100  000554          MOV     #100,RCNT               ;SET REC CNTR TO 100
022046  012737  000010  000566          MOV     #10,PATRN               ;SELECT PATTERN 10
022054  004737  003464                  JSR     PC,STAUTO               ;GO DO AUTO SEQ
022060  012737  000014  000566          MOV     #14,PATRN               ;SELECT PATTERN 14
022066  004737  003464                  JSR     PC,STAUTO
022072  012737  000015  000566          MOV     #15,PATRN               ;SELECT PATTERN 15
022100  004737  003464                  JSR     PC,STAUTO
022104  012737  177777  000744          MOV     #-1,ABLCNT              ;FORCE TO END OF TAPE
022112  012737  177777  000566          MOV     #-1,PATRN               ;SELECT AUTO RANDOM DATA
022120  004737  003464                  JSR     PC,STAUTO
022124  000207                  3$:     RTS     PC                      ;RETURN TO SEQ
```

```

                                                 ;****************************************************
                                                 ;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.
                                                 ;****************************************************

022126  000004  025154            PAPRT:  TYPE,MSG52              ;TYPE 'DRIVE # = '
022132  013703  000550                    MOV     DVN,R3
022136  104400                            TYPOCT                  ;PRINT DRIVE NUMBER
022140  000004  024600                    TYPE,MSG32              ;TYPE 'SLAVE # = '
022144  013703  000552                    MOV     UDES,R3
022150  042703  177770                    BIC     #177770,R3
022154  104400                            TYPOCT                  ;PRINT SLAVE NUMBER
022156  000004  023722                    TYPE,MSG1               ;TYPE DENSITY TAG  '*DE'
022162  013703  000552                    MOV     UDES,R3
022166  000303                            SWAB    R3
022170  042703  177770                    BIC     #177770,R3
022174  104400                            TYPOCT                  ;PRINT DENSITY
022176  000004  025271                    TYPE,MSG61              ;TYPE PARITY TAG '*P'
022202  005003                            CLR     R3
022204  032737  000010  000552            BIT     #10,UDES
022212  001401                            BEQ     PAPRT0
022214  005203                            INC     R3              ;SET PARITY INDICATOR = EVEN
022216  104400                    PAPRT0: TYPOCT                  ;PRINT PARITY BIT STATE
022220  000004  025275                    TYPE,MSG62              ;TYPE FORMAT TAG '*F'
022224  013703  000552                    MOV     UDES,R3
022230  006003                            ROR     R3
022232  006003                            ROR     R3
022234  006003                            ROR     R3              ;POSTION FORMAT BITS
022236  006003                            ROR     R3
022240  042703  177760                    BIC     #177760,R3
022244  104400                            TYPOCT                  ;PRINT FORMAT
022246  000004  023765                    TYPE,MSG8               ;TYPE PATTERN # TAG '*PATRN'
022252  005737  000566                    TST     PATRN           ;BRANCH IF NOT RANDOM PATTERN
022256  100003                            BPL     PAPRTC
022260  000004  024055            PAPRTA: TYPE,MSG17              ;TYPE 'R' FOR RANDOM
022264  000403                            BR      PAPRTD
022266  013703  000566            PAPRTC: MOV     PATRN,R3
022272  104400                            TYPOCT                  ;PRINT PATRN NUMBER
022274  000004  024007            PAPRTD: TYPE,MSG13              ;TYPE BLOCK # TAG '*BN'
022300  013703  000662                    MOV     BLCNTR,R3
022304  104400                            TYPOCT                  ;PRINT NUMBER
022306  000004  024015                    TYPE,MSG14              ;TYPE RECORD # TAG '*RN'
022312  010003                            MOV     R0,R3           ;GET # OF RECORDS LEFT TO PROCESS
022314  032737  000010  000700            BIT     #10,MTC1        ;SEE IF WRITE OPERATION
022322  001416                            BEQ     PAPRT1          ;IF SO: BR
```

```
 4659  022324  022737  000030  000700          CMP     #30,MTC1        ;BRANCH IF SPACE FORWARD
 4660  022332  001412                          BEQ     PAPRT1
 4661  022334  005737  000570                  TST     RDCMD           ;BRANCH IF READ FORWARD
 4662  022340  001407                          BEQ     PAPRT1
 4663  022342  022737  000032  000700          CMP     #32,MTC1        ;BRANCH IF NOT SPACE REVERSE
 4664  022350  001007                          BNE     PAPRT3
 4665  022352  005737  000720                  TST     RTYFL           ;BRANCH IF NOT RETRYING
 4666  022356  001404                          BEQ     PAPRT3
 4667  022360  013703  000554          PAPRT1: MOV     RCNT,R3         ;GET # OF RECORDS TO PROCESS
 4668  022364  160003                  PAPRTY: SUB     R0,R3           ;FORM RECORD NUMBER
 4669  022366  005203                          INC     R3              ;++MP ADJUST RECORD NUMBER IN FORWARD DIRECTION
 4670  022370  104400                  PAPRT3: TYPOCT                  ;PRINT RECORD NUMBER
 4671  022372  000004  023720                  TYPE,DASH               ;TYPE A DASH '-'
 4672  022376  013703  000554                  MOV     RCNT,R3
 4673  022402  104400                          TYPOCT                  ;PRINT RECORD COUNT
 4674  022404  000004  023760                  TYPE,MSG7               ;TYPE RECORD SIZE TAG '*RS'
 4675  022410  013703  000556                  MOV     FMCNT,R3        ;GET CHARACTER COUNT
 4676  022414  005403                          NEG     R3              ;FORM TWO'S COMPLEMENT
 4677  022416  104400                          TYPOCT                  ;PRINT RECORD SIZE
 4678  022420  012737  000001  000672          MOV     #1,HDRFL        ;SET HEADER FLAG
 4679  022426  000207                          RTS     PC              ;RETURN
 4680
```

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

022430  063737  000636  000634  RANG:   ADD     RANSAV,RANBAS
022436  063737  000634  000636          ADD     RANBAS,RANSAV   ;GET NEW NUMBER
022444  023701  000636                  CMP     RANSAV,R1       ;SEE IF NUMBER TOO BIG
022450  101367                          BHI     RANG            ;IF SO: BR
022452  020237  000636                  CMP     R2,RANSAV       ;SEE IF NUMBER TOO SMALL
022456  101364                          BHI     RANG            ;IF SO: BR
022460  000207                          RTS     PC              ;EXIT

                                ;SUBROUTINE TO GET NEW SOFTWARE SWR

022462  022737  000176  000616  GTSWR:  CMP     #SWREG,SWR      ;BRANCH IF SOFTWARE SWR
022470  001025                          BNE     1$              ;NOT INVOKED
022472  004737  022546                  JSR     PC,.SAVE        ;SAVE REGISTERS ON THE STACK
022476  000004  023700                  TYPE,$MSWR
022502  017703  156110                  MOV     @SWR,R3         ;GET CURRENT SWR
022506  104400                          TYPOCT
022510  000004  023710                  TYPE,$MNEW              ;REQUEST NEW SWR SETTING
022514  013705  000616                  MOV     SWR,R5          ;TTR ROUTINE RETURNS VALUE TO (R5)
022520  012701  000007                  MOV     #7,R1           ;LIMIT RESPONSE TO 7 CHARS
022524  012702  177777                  MOV     #177777,R2      ;BETWEEN 0 AND 177777
022530  012703  000000                  MOV     #0,R3
022534  004737  022612                  JSR     PC,TTR          ;GET RESPONSE
022540  004737  022570                  JSR     PC,.RESTORE     ;RESTORE REGISTERS
022544  000207                  1$:     RTS     PC              ;RETURN


                                ;;ROUTINE TO SAVE REGISTERS ON THE STACK
 (1)  022546  010546            .SAVE:  MOV     R5,-(SP)        ;;R5 IS SAVED AT 12(SP)
 (1)  022550  010446                    MOV     R4,-(SP)        ;;R4 IS SAVED AT 10(SP)
 (1)  022552  010346                    MOV     R3,-(SP)        ;;R3 IS SAVED AT 6(SP)
 (1)  022554  010246                    MOV     R2,-(SP)        ;;R2 IS SAVED AT 4(SP)
 (1)  022556  010146                    MOV     R1,-(SP)        ;;R1 IS SAVED AT 2(SP)
 (1)  022560  010046                    MOV     R0,-(SP)        ;;R0 IS SAVED AT (SP)
 (1)  022562  016646  000014            MOV     14(SP),-(SP)    ;;PUSH RETURN PC ON THE STACK
 (1)  022566  000207                    RTS     PC              ;;RETURN TO CALLER
 (1)

                                ;;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
 (1)  022570  012666  000014    .RESTORE:MOV    (SP)+,14(SP)    ;;S E RETURN PC ON STACK
 (1)  022574  012600                    MOV     (SP)+,R0
 (1)  022576  012601                    MOV     (SP)+,R1
 (1)  022600  012602                    MOV     (SP)+,R2
 (1)  022602  012603                    MOV     (SP)+,R3
 (1)  022604  012604                    MOV     (SP)+,R4
 (1)  022606  012605                    MOV     (SP)+,R5
 (1)  022610  000207                    RTS     PC              ;;RETURN
 (1)
```

```
                                        ;*********************************************************
                                        ;TTY ENTRY SUBROUTINE:
                                        ;
                                        ;THIS SUBROUTINE IS USED BY THE TEST CONDITION
                                        ;ENTRY ROUTINE TO READ THE RESPONSE ENTERED
                                        ;AT THE TTY AND CHECK THEM FOR LEGALITY AND
                                        ;LIMITS.  ALL RESPONSE MUST BE TYPED IN OCTAL
                                        ;(0-7) AND MUST FALL WITHIN THE LIMITS SET BY
                                        ;THE CALLING ROUTINE.
                                        ;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
                                        ;A QUESTION MARK IS TYPED (?) AND THE RESPONSE
                                        ;MAY BE REENTERED.
                                        ;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
                                        ;MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
                                        ;CARRIAGE RETURN
                                        ;*********************************************************

022612  010146              TTR:    MOV     R1,-(SP)        ;SAVE CHAR COUNT
022614  011601              10$:    MOV     (SP),R1         ;RESTORE CHAR COUNT (FOR ↑U)
022616  005037  000652              CLR     TEMP1           ;CLEAR FIRST CHARACTER FLAG
022622  005000                      CLR     R0
022624  004737  023036      1$:     JSR     PC,TTIN         ;GO READ CHARACTER
022630  122737  000003  000650      CMPB    #3,TIB          ;BRANCH IF NOT ↑C
022636  001003                      BNE     11$
022640  000005                      RESET
022642  000137  000200              JMP     @#200           ;RESTART AT 200
022646  122737  000015  000650  11$:CMPB    #15,TIB         ;SEE IF CR
022654  001004                      BNE     2$              ;IF NOT: BR
022656  005737  000652              TST     TEMP1           ;SEE IF FIRST CHARACTER
022662  001455                      BEQ     9$              ;IF SO: BR
022664  000447                      BR      6$              ;ELSE GO LOAD VALUE
022666  122737  000025  000650  2$: CMPB    #25,TIB         ;BRANCH IF NOT CONTROL U
022674  001003                      BNE     21$
022676  006004  024355              TYPE,MSG28              ;TYPE <CR><LF>
022702  000744                      BR      10$
022704  122737  000177  000650  21$:CMPB    #177,TIB        ;BRANCH IF NOT 'RUBOUT'
022712  001010                      BNE     3$
022714  000241                      CLC                     ;REMOVE LAST CHARACTER
022716  006000                      ROR     R0
022720  006200                      ASR     R0
022722  006200                      ASR     R0
022724  000004  026312              TYPE,MSG118             ;TYPE '\'
022730  005201                      INC     R1              ;DEC CHAR RECEIVED COUNT
022732  000734                      BR      1$              ;GET NEXT CHARACTER
022734  122737  000060  000650  3$: CMPB    #60,TIB         ;SEE IF CHAR IS LESS THAN 0
022742  101027                      BHI     TINER
022744  122737  000070  000650  4$: CMPB    #70,TIB         ;SEE IF CHAR IS GREATER THAN 7
022752  101423                      BLOS    TINER
022754  005237  000652      5$:     INC     TEMP1           ;SET FIRST CHARACTER FLAG
022760  006300                      ASL     R0
022762  006300                      ASL     R0              ;SHIFT 3 LEFT
022764  006300                      ASL     R0
022766  042737  177770  000650      BIC     #177770,TIB     ;STRIP ASCII
022774  053700  000650              BIS     TIB,R0          ;LOAD CHARACTER
023000  005301                      DEC     R1              ;SEE IF DONE
023002  001310                      BNE     1$              ;IF NOT: BR
```

```
 4735  023004  020002          6$:     CMP     R0,R2           ;SEE IF EXCEEDED MAXIMUM LIMIT
 4736  023006  101005                  BHI     TINER
 4737  023010  020300          7$:     CMP     R3,R0           ;SEE IF BELOW MINIMUM LIMIT
 4738  023012  101003                  BHI     TINER
 4739  023014  010015          8$:     MOV     R0,(R5)         ;LOAD VALUE
 4740  023016  005726          9$:     TST     (SP)+           ;POP CHAR COUNT OFF STACK
 4741  023020  000207                  RTS     PC              ;EXIT
 4742
 4743  023022  000004  025046  TINER:  TYPE,@MSG43             ;TYPE '?'
 4744  023026  005726                  TST     (SP)+           ;POP CHAR COUNT OFF STACK
 4745  023030  162716  000020          SUB     #20,(SP)        ;RESET SP TO START OF VALUE ROUTINE
 4746  023034  000207                  RTS     PC              ;REDO VALUE ENTRY
```

```
                                        ;TTY READ SUBROUTINE**********
                                
023036  005277  155556          TTIN:   INC     @TKS
023042  105777  155552          1$:     TSTB    @TKS
023046  100375                          BPL     1$
023050  017737  155546  000650          MOV     @TKB,TIB
023056  042737  177600  000650          BIC     #177600,TIB     ;STRIP PARITY BIT
023064  022737  000015  000650          CMP     #15,TIB         ;BRANCH IF NOT <CR>
023072  001003                          BNE     2$
023074  000004  024355                  TYPE,MSG28              ;TYPE '<CR><LF>
023100  000402                          BR      3$
023102  000004  000650          2$:     TYPE,TIB                ;ECHO CHARACTER
023106  000207                  3$:     RTS     PC
                                
                                        ;TTY OUTPUT SUBROUTINE***********
                                
023110  010446                  TTOUT:  MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
023112  010346                          MOV     R3,-(SP)
023114  017604  000004                  MOV     @4(SP),R4       ;GET ADDRESS OF MESSAGE TO TYPE
023120  062766  000002  000004          ADD     #2,4(SP)        ;ADJUST RETURN PC
023126  111437  000646          10$:    MOVB    (R4),TOB        ;GET A CHARACTER
023132  001431                          BEQ     3$              ;AND BRANCH IF END OF MSG
023134  122724  000045                  CMPB    #45,(R4)+       ;BRANCH IF CRLF CHARACTER (*)
023140  001403                          BEQ     1$
023142  004737  023224                  JSR     PC,TOG          ;ECHO CHARACTER
023146  000767                          BR      10$
                                
023150  112737  000015  000646  1$:     MOVB    #15,TOB
023156  004737  023224                  JSR     PC,TOG
023162  012703  000006                  MOV     #6,R3
023166  005037  000646          2$:     CLR     TOB
023172  004737  023224                  JSR     PC,TOG
023176  005303                          DEC     R3
023200  001372                          BNE     2$              ;DO FILLERS
023202  112737  000012  000646          MOVB    #12,TOB
023210  004737  023224                  JSR     PC,TOG
023214  000744                          BR      10$
023216  012603                  3$:     MOV     (SP)+,R3        ;RESTORE REGISTERS
023220  012604                          MOV     (SP)+,R4
023222  000002                          RTI                     ;RETURN
```

```
023224  105777  155370          TOG:    TSTB    @TKS                ;SEE IF INPUT AT KEYBOARD
023230  100024                          BPL     3$                  ;IF SO, THEN
023232  117737  155364  000751          MOVB    @TKB,$CNTRLS+1      ;MOVE CHARACTER AND
023240  142737  000200  000751          BICB    #200,$CNTRLS+1      ;MASK OFF PARITY BIT.
023246  122737  000023  000751          CMPB    #23,$CNTRLS+1       ;SEE IF CHARACTER IS XOFF
023254  001004                          BNE     2$                  ;IF XOFF, THEN
023256  112737  000377  000750          MOVB    #377,$CNTRLS        ;SET XOFF FLAG
023264  000757                          BR      TOG
023266  122737  000021  000751  2$:     CMPB    #21,$CNTRLS+1       ;SEE IF CHARACTER IS XON
023274  001002                          BNE     3$                  ;IF SO THEN
023276  105037  000750                  CLRB    $CNTRLS             ;CLEAR XOFF FLAG
023302  105737  000750          3$:     TSTB    $CNTRLS             ;SEE IF IN XOFF MODE
023306  100746                          BMI     TOG                 ;IF NOT THEN
023310  105777  155310                  TSTB    @TPS                ;CHECK IF PRINTER READY
023314  100343                          BPL     TOG                 ;
023316  113777  000646  155302          MOVB    TOB,@TPB
023324  000207                          RTS     PC                  ;RETURN
```

```
                                        ;OCTAL OUTPUT SUBROUTINE**********
                                
023326  005037  023502          OCTP:   CLR     OFL             ;CLEAR FLAG FOR LEADING ZERO
023332  010304                          MOV     R3,R4           ;SEE IF NUMBER IS ZERO
023334  001003                          BNE     1$              ;IF NOT ZERO: BR
023336  000004  026537                  TYPE,DIGITO
023342  000434                          BR      4$              ;SPACE AND EXIT
023344  100004                  1$:     BPL     3$              ;BRANCH IF MSD IS A '0'
023346  012704  000001                  MOV     #1,R4
023352  004737  023442                  JSR     PC,OCTPG        ;PRINT 1
023356  006004                  3$:     ROR     R4
023360  006004                          ROR     R4
023362  006004                          ROR     R4              ;POSITION DIGIT
023364  006004                          ROR     R4
023366  000304                          SWAB    R4
023370  004737  023442                  JSR     PC,OCTPG        ;PRINT DIGIT 2
023374  006004                          ROR     R4
023376  000304                          SWAB    R4
023400  004737  023442                  JSR     PC,OCTPG        ;PRINT DIGIT 3
023404  006104                          ROL     R4
023406  006104                          ROL     R4
023410  000304                          SWAB    R4
023412  004737  023442                  JSR     PC,OCTPG        ;PRINT DIGIT 4
023416  006004                          ROR     R4
023420  006004                          ROR     R4
023422  006004                          ROR     R4
023424  004737  023442                  JSR     PC,OCTPG        ;PRINT DIGIT 5
023430  004737  023442                  JSR     PC,OCTPG        ;PRINT DIGIT 6
023434  000004  026535          4$:     TYPE,SPACE              ;TYPE A SPACE
023440  000002                          RTI                     ;EXIT

023442  042704  177770          OCTPG:  BIC     #177770,R4
023446  001003                          BNE     1$
023450  005737  023502                  TST     OFL
023454  001410                          BEQ     2$
023456  005237  023502          1$:     INC     OFL
023462  052704  000260                  BIS     #260,R4
023466  010437  000646                  MOV     R4,TOB
023472  004737  023224                  JSR     PC,TOG
023476  010304                  2$:     MOV     R3,R4
023500  000207                          RTS     PC
023502  000000                  OFL:    0                       ;FIRST CHAR FLAG


                                        ;DATA CHARACTER OUTPUT SUBROUTINE**********

023504  012704  000010          DOUT:   MOV     #10,R4          ;SET NUMBER TO PRINT
023510  110346                          MOVB    R3,-(SP)        ;GET CHAR TO OUTPUT
023512  106316                  1$:     ASLB    (SP)            ;BRANCH IF BIT IS A ZERO
023514  103003                          BCC     2$
023516  000004  026541                  TYPE,DIGIT1
023522  000402                          BR      3$
023524  000004  026537          2$:     TYPE,DIGITO
023530  005304                  3$:     DEC     R4
023532  001367                          BNE     1$
023534  005726                          TST     (SP)+           ;POP STACK
```

```
4905  023536  000207                          RTS     PC
4906
4907  023540  113703  000657          DOUTD:  MOVB    TEMP3+1,R3
4908  023544  004737  023504                  JSR     PC,DOUT
4909  023550  013703  000656                  MOV     TEMP3,R3
4910  023554  004737  023504                  JSR     PC,DOUT
4911  023560  000207                          RTS     PC
4912
4913                                          ;TU16 SERIAL NUMBER PRINT SUBROUTINE*******
4914
4915  023562  017703  154752          SNPT:   MOV     @SN,R3          ;GET CONTENTS OF SERIAL # REG
4916  023566  000004  023775                  TYPE,MSG9               ;TYPE SN TAG
4917  023572  010304                          MOV     R3,R4
4918  023574  000304                          SWAB    R4
4919  023576  006004                          ROR     R4
4920  023600  006004                          ROR     R4
4921  023602  006004                          ROR     R4
4922  023604  006004                          ROR     R4
4923  023606  004737  023654                  JSR     PC,SNPG         ;PRINT FIRST DIGIT
4924  023612  010304                          MOV     R3,R4
4925  023614  000304                          SWAB    R4
4926  023616  004737  023654                  JSR     PC,SNPG         ;PRINT SECOND DIGIT
4927  023622  010304                          MOV     R3,R4
4928  023624  006004                          ROR     R4
4929  023626  006004                          ROR     R4
4930  023630  006004                          ROR     R4
4931  023632  006004                          ROR     R4
4932  023634  004737  023654                  JSR     PC,SNPG         ;PRINT THIRD DIGIT
4933  023640  010304                          MOV     R3,R4
4934  023642  004737  023654                  JSR     PC,SNPG         ;PRINT FOURTH DIGIT
4935  023646  000004  024355                  TYPE,MSG28              ;TYPE <CR><LF>
4936  023652  000207                          RTS     PC              ;EXIT
4937  023654  012737  000260  000646  SNPG:   MOV     #260,TOB        ;SET NUMBER BASE
4938  023662  042704  177760                  BIC     #177760,R4      ;MASK NUMBER
4939  023666  050437  000646                  BIS     R4,TOB          ;BUILD DIGIT
4940  023672  004737  023224                  JSR     PC,TOG          ;GO TYPE
4941  023676  000207                          RTS     PC              ;RETURN
4942
```

```
                                        ;ERROR MESSAGES**********
023700  051445  051127  036440  $MSWR:  .ASCIZ  /*SWR - /
023706  000040
023710  047040  053505  036440  $MNEW:  .ASCIZ  / NEW = /
023716  000040
023720  000055                  DASH:   .ASCIZ  /-/
023722  042052  020105     000  MSG1:   .ASCIZ  /*DE /
023727     045  035507  000040  MSG2:   .ASCIZ  /*G; /
023734  041045  020073     000  MSG3:   .ASCIZ  /*B; /
023741     045  047103  000040  MSG4:   .ASCIZ  /*CN /
023746  053452  020105     000  MSG5:   .ASCIZ  /*WE /
023753     052  042522  000040  MSG6:   .ASCIZ  /*RE /
023760  051052  020123     000  MSG7:   .ASCIZ  /*RS /
023765     052  040520  051124  MSG8:   .ASCIZ  /*PATRN /
023772  020116     000
023775     123  035116  000040  MSG9:   .ASCIZ  /SN: /
024002  051452  020105     000  MSG10:  .ASCIZ  /*SE /
024007     045  041052  020116  MSG13:  .ASCIZ  /**BN /
024014     000
024015     052  047122  000040  MSG14:  .ASCIZ  /*R; /
024022  020045  020040  020040  MSG15:  .ASCIZ  /*          BAD RECORD**/
024030  020040  020040  041040
024036  042101  051040  041505
024044  051117  022504  000045
024052  043040     000          MSG16:  .ASCIZ  / F/
024055     040  000122          MSG17:  .ASCIZ  / R/
024060  042440  052117  021440  MSG20:  .ASCIZ  / EOT # /
024066  000040
024070  047111  042524  041522  MSG21:  .ASCIZ  /INTERCHANGE READ? /
024076  040510  043516  020105
024104  042522  042101  020077
024112     000
024113     045  046111  042514  MSG22:  .ASCIZ  /*ILLEGAL BOT: HALT*/
024120  040507  020114  047502
024126  035124  044040  046101
024134  022524     000
024137     045  051503  020061  MSG23:  .ASCIZ  /*CS1 /
024144     000
024145     045  041527  000040  MSG23A: .ASCIZ  /*WC /
024152  041045  020101     000  MSG23B: .ASCIZ  /*BA /
024157     045  041506  000040  MSG23C: .ASCIZ  /*FC /
024164  041445  031123  000040  MSG23D: .ASCIZ  /*CS2 /
024172  042045  020123     000  MSG23E: .ASCIZ  /*DS /
024177     045  051105  000040  MSG23F: .ASCIZ  /*ER /
024204  040445  020123     000  MSG23G: .ASCIZ  /*AS /
024211     045  045503  000040  MSG23H: .ASCIZ  /*CK /
024216  042045  020102     000  MSG23I: .ASCIZ  /*DB /
024223     045  051115  000040  MSG23J: .ASCIZ  /*MR /
024230  042045  020124     000  MSG23K: .ASCIZ  /*DT /
024235     045  041524  000040  MSG23L: .ASCIZ  /*TC /
024242  047045  020117  047111  MSG24:  .ASCIZ  /*NO INTERRUPT*/
024250  042524  051122  050125
024256  022524     000
024261     045  046123  053101  MSG25:  .ASCIZ  /*SLAVE UNSAFE-TEST DISCONTINUED ON SLAVE*/
```

```
                024266  020105  047125  040523
                024274  042506  052055  051505
                024302  020124  044504  041523
                024310  047117  044524  052516
                024316  042105  047440  020116
                024324  046123  053101  022505
                024332     000
4983            024333     045  051104  050117  MSG26:  .ASCIZ  /*DROPS: /
                024340  035123  000040
4984            024344  050045  041511  051513  MSG27:  .ASCIZ  /*PICKS: /
                024352  020072     000
4985            024355     045     000          MSG28:  .ASCIZ  /*/
4986            024357     045  052045  047515  MSG30:  .ASCIZ  '**TM03-TE16/TU77 AUTO SEQUENCE (CZTEDE0)*';+-B
                024364  026463  042524  033061
                024372  052057  033525  020067
                024400  052501  047524  051440
                024406  050505  042525  041516
                024414  020105  041450  052132
                024422  042105  030105  022451
                024430     000
4987            024431     045  052045  030115  MSG31:  .ASCIZ  '**TM03-TE16/TU77 DATA RELIABILITY TEST (CZTEDE0)*';+-B
                024436  026463  042524  033061
                024444  052057  033525  020067
                024452  040504  040524  051040
                024460  046105  040511  044502
                024466  044514  054524  052040
                024474  051505  020124  041450
                024502  052132  042105  030105
                024510  022451     000
4988            024513     124  050131  020105  MSG31A: .ASCIZ  /TYPE <CR> TO TERMINATE ALL REQUESTS & ^C TO RESTART*/
                024520  041474  037122  052040
                024526  020117  042524  046522
                024534  047111  052101  020105
                024542  046101  020114  042522
                024550  052521  051505  051524
                024556  023040  057040  020103
                024564  047524  051040  051505
                024572  040524  052122  000045
4989            024600  046123  053101  020105  MSG32:  .ASCIZ  /SLAVE # = /
                024606  020043  020075     000
4990            024613     104  047105  044523  MSG33:  .ASCIZ  /DENSITY = /
                024620  054524  036440  000040
4991            024626  040520  044522  054524  MSG34:  .ASCIZ  /PARITY = /
                024634  036440  000040
4992            024640  042522  047503  042122  MSG35:  .ASCIZ  /RECORD COUNT = /
                024646  041440  052517  052116
                024654  036440  000040
4993            024660  044103  051101  041440  MSG36:  .ASCIZ  /CHAR COUNT = /
                024666  052517  052116  036440
                024674  000040
4994            024676  040520  052124  051105  MSG37:  .ASCIZ  /PATTERN # = /
                024704  020116  020043  020075
                024712     000
4995            024713     123  047111  046107  MSG38:  .ASCIZ  /SINGLE PASS? /
                024720  020105  040520  051523
                024726  020077     000
```

```
 4996  024731     103  041522  041440  MSG39:  .ASCIZ  /CRC CORRECTION (YES=1,NO=0)? /
       024736  051117  042522  052103
       024744  047511  020116  054450
       024752  051505  030475  047054
       024760  036517  024460  020077
       024766     000
 4997  024767     045  042445  052116  MSG40:  .ASCIZ  /**ENTER STALLS*READ = /
       024774  051105  051440  040524
       025002  046114  022523  042522
       025010  042101  036440  000040
 4998  025016  051127  052111  020105  MSG41:  .ASCIZ  /WRITE = /
       025024  020075     000
 4999  025027     124  051125  020116  MSG42:  .ASCIZ  /TURN AROUND = /
       025034  051101  052517  042116
       025042  036440  000040
 5000  025046  037445  000045          MSG43:  .ASCIZ  /*?*/
 5001  025052  042445  052116  051105  MSG44:  .ASCIZ  /*ENTER YOZZLE STALL = /
       025060  054440  055117  046132
       025066  020105  052123  046101
       025074  020114  020075     000
 5002  025101     045  051105  020122  MSG45:  .ASCIZ  /*ERR AMT /
       025106  046501  020124     000
 5003  025113     045  047516  020124  MSG49:  .ASCIZ  /*NOT AVAIL /
       025120  053101  044501  020114
       025126     000
 5004  025127     045  046111  042514  MSG50:  .ASCIZ  /*ILLEGAL DRIVE TYPE /
       025134  040507  020114  051104
       025142  053111  020105  054524
       025150  042520  000040
 5005  025154  022445          MSG52:  .ASCII  /**/
 5006  025156  051104  053111  020105  MSG52A: .ASCIZ  /DRIVE (TM03) # = /
       025164  052050  030115  024463
       025172  021440  036440  000040
 5007  025200  047506  046522  052101  MSG53:  .ASCIZ  /FORMAT = /
       025206  036440  000040
 5008  025212  053452  020105  046524  MSG54:  .ASCIZ  /*WE TM/
       025220     000
 5009  025221     052  042523  052040  MSG55:  .ASCIZ  /*SE TM/
       025226  000115
 5010  025230  052040  000115          MSG56:  .ASCIZ  / TM/
 5011  025234  047045  047117  042455  MSG57:  .ASCIZ  /*NON-EXIST SLAVE/
       025242  044530  052123  051440
       025250  040514  042526     000
 5012  025255     045  051103  020103  MSG58:  .ASCIZ  /*CRC /
       025262     000
 5013  025263     045  051114  020103  MSG59:  .ASCIZ  /*CRC /
       025270     000
 5014  025271     052  020120     000  MSG61:  .ASCIZ  /*P /
 5015  025275     052  020106     000  MSG62:  .ASCIZ  /*F /
 5016  025301     045  047452  044522  MSG64:  .ASCIZ  /**ORIGINAL ERROR*/
       025306  044507  040516  020114
       025314  051105  047522  025122
       025322     000
 5017  025323     045  042522  051124  MSG65:  .ASCIZ  /*RETRY: /
       025330  035131  000040
 5018  025334  051452  020105  052122  MSG66:  .ASCIZ  /*SE RTRY /
```

```
                025342  054522  000040
5019            025346  042452  040522  042523  MSG67:  .ASCIZ  /*ERASE/
                025354     000
5020            025355     045  042522  042522  MSG68:  .ASCIZ  /*REREV: /
                025362  035126  000040
5021            025366  040524  042520  046440  MSG69:  .ASCIZ  /TAPE MARK? /
                025374  051101  037513  000040
5022            025402  047045  047117  042455  MSG71:  .ASCIZ  /*NON-EXIST DRIVE/
                025410  044530  052123  042040
                025416  044522  042526     000
5023            025423     045  042522  053506  MSG72:  .ASCIZ  /*REFWD: /
                025430  035104  000040
5024            025434  053445  042524  051122  MSG73:  .ASCIZ  /*WTERR: /
                025442  020072     000
5025            025445     045  042522  044507  MSG74:  .ASCIZ  /*REGISTER START = /
                025452  052123  051105  051440
                025460  040524  052122  036440
                025466  000040
5026            025470  042526  052103  051117  MSG75:  .ASCIZ  /VECTOR ADRS = /
                025476  040440  051104  020123
                025504  020075     000
5027            025507     045  042504  042522  MSG76:  .ASCIZ  /*DEREV: /
                025514  035126  000040
5028            025520  042045  043105  042127  MSG77:  .ASCIZ  /*DEFWD: /
                025526  020072     000
5029            025531     045  047516  026516  MSG78:  .ASCIZ  /*NON-RETRYABLE WRITE ERROR: ER /
                025536  042522  051124  040531
                025544  046102  020105  051127
                025552  052111  020105  051105
                025560  047522  035122  042440
                025566  020122     000
5030            025571     045  047516  026516  MSG79:  .ASCIZ  /*NON-RETRYABLE READ ERROR: ER /
                025576  042522  051124  040531
                025604  046102  020105  042522
                025612  042101  042440  051122
                025620  051117  020072  051105
                025626  000040
5031            025630  042445  042116  047440  MSG100: .ASCIZ  /*END OF PASS */
                025636  020106  040520  051523
                025644  022440     000
5032            025647     045  025045  025052  MSG101: .ASCIZ  /**********************/
                025654  025052  025052  025052
                025662  025052  025052  025052
                025670  025052  025052  022452
                025676     000
5033            025677     101  052125  020117  MSG104: .ASCIZ  /AUTO CONT.? /
                025704  047503  052116  037456
                025712  000040
5034            025714  051045  041505  053117  MSG105: .ASCIZ  /*RECOVERED/
                025722  051105  042105     000
5035            025727     052  040502  020104  MSG106: .ASCIZ  /*BAD TAPE OVERFLOW/
                025734  040524  042520  047440
                025742  042526  043122  047514
                025750  000127
5036            025753  051045  053505  047111  MSG16A: .ASCIZ  /*REWIND TAPE; RESTART AT BLOCK 1/
                025760  020104  040524  042520
```

```
                025766  020073  042522  052123
                025774  051101  020124  052101
                026002  041040  047514  045503
                026010  030440     000
5037            026013     045  047125  042522  MSG107: .ASCII  /%UNRECOVERABLE BAD SPOT/
                026020  047503  042526  040522
                026026  046102  020105  040502
                026034  020104  050123  052117
5038            026042  041045  042101  051040          .ASCIZ  /%BAD RECORD LEFT ON TAPE%/
                026050  041505  051117  020104
                026056  042514  052106  047440
                026064  020116  040524  042520
                026072  000045
5039            026074  050052  051517  052111  MSG109: .ASCIZ  /APOSITION LOST IN RETRY/
                026102  047511  020116  047514
                026110  052123  044440  020116
                026116  042522  051124  000131
5040            026124  051445  051525  042520  MSG110: .ASCIZ  /%SUSPECT BAD TAPE/
                026132  052103  041040  042101
                026140  052040  050101  000105
5041            026146  051045  050105  040505  MSG111: .ASCIZ  /%REPEAT: /
                026154  035124  000040
5042            026160  041040  042101  052040  MSG112: .ASCIZ  / BAD TAPE SPOTS%/
                026166  050101  020105  050123
                026174  052117  022523     000
5043
5044            026201     045  051440  043117  MSG113: .ASCIZ  /% SOFT: /
                026206  035124  000040
5045
5046            026212  020045  040510  042122  MSG114: .ASCIZ  /% HARD: /
                026220  020072     000
5047
5048            026223     045  040510  042122  MSG115: .ASCIZ  /%HARD READ ERROR/
                026230  051040  040505  020104
                026236  051105  047522  000122
5049            026244  051445  040514  042526  MSG116: .ASCIZ  /%SLAVE REWINDING: WILL RESTART AT BOT/
                026252  051040  053505  047111
                026260  044504  043516  020072
                026266  044527  046114  051040
                026274  051505  040524  052122
                026302  040440  020124  047502
                026310  000124
5050            026312  000134                  MSG118: .ASCIZ  /\/
5051            026314  051045  046505  053117  MSG120: .ASCIZ  /%REMOVE  TMDP FROM SLAVE TO BE TESTED%/
                026322  020105  046524  050104
                026330  043040  047522  020115
                026336  046123  053101  020105
                026344  047524  041040  020105
                026352  042524  052123  042105
                026360  000045
5052            026362  044045  051101  053504  MSG121: .ASCIZ  /%HARDWARE SWR IN USE%/
                026370  051101  020105  053523
                026376  020122  047111  052440
                026404  042523  000045
5053            026410  047516  051440  040514  MSG122: .ASCIZ  /NO SLAVES LEFT TO TEST: HALT%/
                026416  042526  020123  042514
```

```
        026424  052106  052040  020117
        026432  042524  052123  020072
        026440  040510  052114  000045
 5054   026446  040445  052125  026517  MSG123: .ASCIZ  /*AUTO SEQ: TEST WILL RESTART*/
        026454  042523  035121  052040
        026462  051505  020124  044527
        026470  046114  051040  051505
        026476  040524  052122  000045
 5055   026504  041445  051117  042522  MSG124: .ASCIZ  /*CORRECTED PE DATA ERROR/
        026512  052103  042105  050040
        026520  020105  040504  040524
        026526  042440  051122  051117
        026534     000
 5056   026535     040     000          SPACE:  .ASCIZ  ' '
 5057   026537     060     000          DIGIT0: .ASCIZ  '0'
 5058   026541     061     000          DIGIT1: .ASCIZ  '1'
 5059
 5060           026544                       .EVEN
 5061   026544  036544                 BUFBEG: .=.+10000               ;READ AND WRITE BUFFER AREA
 5062           000001                       .END
```

```
AA1$     003076            1890    1898#
ABLCNT   000744            1614#   1983    4562*   4587*   4596*
ACTLRC   020356            4161*   4243    4295#
AMOD1    021640            4506    4556#
AMOD2    021772            4507    4580#
APATS    014436            3573    3582#
AS       000526            1535#   4275    4276*
ASEQ     021312            3167    4487#
ASEQCF   000746            1615#   4488    4512
ASEQF    000742            1516*   1613#   1883*   1917*   1981    2162    3127    3164    3389    4324
ASEQX    021442            4509    4512#
ASEQ0    021342            1919    2175    4328    4493#   4513
ASEQ1    021346            4494#   4511
ASEQ4    021424            4508#   4526
BA       000514            1530#   2303*   2354*   2482*   2503*   2677*   2686*   2904*   2912*   2978*   3005*   4073    4218
BAER     020342            4069    4075*   4169    4215    4289#
BBC      000664            1589#   3706*   3769*   3773*   3858*   3859    3864*   3865*   3866
BB1$     003112            1894    1896    1902#
BCNT     000716            1602#   3953*   3974*   3977*   3982*   4002*   4005*   4009*   4013*
BDPP     000724            1605#   3938*   3958*   3985*   3990*   3998*   4000*   4003    4007*
BD00     001414            1641    1747#
BD10     001434            1642    1749#
BD20     001454            1643    1751#
BD30     001474            1644    1753#
BD40     001514            1645    1755#
BD50     001534            1646    1757#
BD60     001554            1647    1759#
BD70     001574            1648    1761#
BKSP     011514            2605    2615    2974#
BLCNTR   000662            1588#   1950*   1980*   1983    1985*   2457    4556*   4580*   4653
BPKP     000726            1606#   3937*   3965*   3984*   3989*   3999*   4001*   4011    4015*
BP00     001214            1633    1731#
BP10     001234            1634    1733#
BP20     001254            1635    1735#
BP30     001274            1636    1737#
BP40     001314            1637    1739#
BP50     001334            1638    1741#
BP60     001354            1639    1743#
BP70     001374            1640    1745#
BTADDR   001034            1652#   2451    2552
BTFLG    000732            1608#   2095    2101    2105    2117*   2490*   2514*
BTOV     007104            2464    2513#
BTOVX    007236            2538    2544#
BTOV0    007124            2517#   2561
BTOV1    007134            2520#   2536
BTOV2    007212            2534    2537#
BTOV3    007230            2541#   2543
BTPRT    007240            2025    2119    2550#
BTPRT1   007304            2559    2561#
BTPT     000736            1610#   2451*   2452    2454*   2455    2462    2517    2533    2540    2552*   2553    2558
BTSTF    000734            1609#   2024*   2026*   2537
BTUR     007310            2472    2565#
BT00     001614            1652    1769#
BT01     001720            1653    1771#
BT02     002024            1654    1773#
BT03     002130            1655    1775#
```

```
B104    002234          1656    1777#
BT05    002340          1657    1779#
BT06    002444          1658    1781#
BT07    002550          1659    1783#
BUFDEC  026544          1552    1553    1904    1905    5061#
BUFMAX  000560          1551#   1895*   1898*   1902    3050    3272    4563    4588
BO      011646          2980    2986    2992    2994#
CADER   020340          2361*   2712*   2990*   4067*   4221    4288#
CC      000530          1536#   4148    4234    4256
CCNTR   011776          1993    3049#
CHNFLG  003146          1910#*  1915    2173    4533
CLLAST  014620          3635#
CLP     014730          3619    3646    3654#
CLPE    014754          3655    3657    3661#
CLP2    015012          3669    3673#
CLP3    015024          3671    3674    3676#
CL0     014540          3618#   3634
CL1     014566          3626#
CL2     014610          3623    3632#
CL3     014672          3645#   3649
CONER   020344          4120*   4199    4290#
CRCC    000574          1557#   3317    3319    4269
CRCER   020354          4152*   4231    4294#
CRCLRC  014522          3403    3411    3614#
CRCSV   020362          4134*   4285    4297#
CS      000520          1532#   1966*   2002*   2003*   2245*   2265*   3169*   3176*   3178    3198*   3199*   4103    4205
                        4271*   4272*   4367*   4520*   4521*   4523
C1      000510          1528#   2091*   2094*   2146*   2149*   2203*   2204*   2227*   2256*   2261*   3159    3177    4101
                        4110    4202    4274*   4277    4392*   4406*   4522
DASH    023720          4220    4237    4245    4671    4949#
DATA0   002776          1848#
DATA1   003000          1849#
DATA10  003016          1856#
DATA11  003020          1857#
DATA12  003022          1858#
DATA13  003024          1859#
DATA14  003026          1860#
DATA15  003030          1861#
DATA2   003002          1850#
DATA3   003004          1851#
DATA4   003006          1852#
DATA5   003010          1853#
DATA6   003012          1854#
DATA7   003014          1855#
DATBL   002774          1847#   3410
DATER1  001134          1698#   2050    3790*
DATR    014456          1996    3393    3593#
DAT0    013772          1848    3429#
DAT0A   014020          3435#   3443    3447    3450
DAT0B   014026          3436#   3437
DAT0C   014072          3441    3448#
DAT0D   014100          3451#   3459
DAT0E   014110          3453#   3458
DAT0F   014122          3455    3457#
DAT1    014132          1849    3465#
DAT1A   014136          3466#   3475    3498    3503    3552
```

DAT10	014256	1856	35190					
DAT11	014304	1857	35310					
DAT12	014324	1858	35410					
DAT13	014346	1859	35510					
DAT14	014356	1860	35560					
DAT15	014406	1861	35720					
DAT2	014152	1850	34740					
DAT3	014156	1851	34790					
DAT3A	014164	34810	3492					
DAT4	014200	1852	34900					
DAT5	014212	1853	34970					
DAT6	014220	1854	35020					
DAT7	014226	1855	35070					
DB	000532	15370						
DCHK	015064	2757	2951	37060				
DCHK0	015112	3710	37120					
DEREV1	001174	17200	2062	3792*				
DEREX	016070	3860	3878	3880	3888	3895	3898	39000
DEREX1	016122	3901	3904	3906	39080			
DERFL	000712	16000	3707*	3783	3909*			
DERR	015470	3776	38240					
DERR0	015500	38260	3907					
DERR0A	015526	3828	38320					
DERR0B	015554	3837	38400					
DERR0C	015574	3843	38460					
DERR0D	015576	3845	38470					
DERR1	015620	3850	38530					
DERR2	015622	3852	38540					
DERR3	015634	38570						
DERR4	015636	3825	3856	38580				
DERR4A	015762	3872	38810					
DERR4B	016024	3867	38830					
DERR5	016052	3892	38960					
DERR6	016064	3869	3890	38990				
DF*	015466	3784	3786	3791	37930			
DF0	015364	3732	37710	3780				
DF0A	015260	3742	37440	3781				
DF0A0	015302	3748	37500					
DF0A1	015316	3753	37550					
DF0A2	015332	3758	37600					
DF0A3	015346	3763	37650					
DF0A4	015352	3745	37670					
DF0B	015220	37330						
DF0B0	015242	3736	37390					
DF0C	015202	3725	37290					
DF0C0	015212	3715	3717	3719	37310			
DF0D	015166	3721	37260					
DF0E	015160	37230	3728					
DF0F	015152	37200	3724					
DF1	015376	3768	3772	37750				
DF2	015406	3770	3774	37770				
DF3	015420	3778	37820					
DF4	015462	3789	37920					
DIGIT0	026532	4854	4901	50570				
DIGIT1	026541	4899	50580					
DOUT	023564	3847	3854	48950	4908	4910		

```
CZTEDE0 TM03-TE16/1077 DRT      MACY11 30(1046)  07-MAR-84  14:21  PAGE 72
CZTEDE.P11    07-MAR-84 14:04          CROSS REFERENCE TABLE -- MACRO NAMES                    SEQ 0129

$CHAIN    1365#    1910
$RESTO    1365#    4717
$SAVE     1365#    4716
.$ACT1    1365#    1485
.$EOP     1365#    2172


. ABS.  036544     000


 ERRORS DETECTED:  0

 CZTEDE,BIN,CZTEDE,LST/CRF/NL:TOC=CZTEAE,SML/ML,CZTEDE.P11
 RUN-TIME:  5 10 1 SECONDS
 RUN-TIME RATIO: 46/17=2.6
 CORE USED:  14K  (28 PAGES)
```

CZTEDE.P11	07-MAR....B1	CZTEDE.P11	07-MAR....B5	CZTEDE.P11	07-MAR....B9
CZTEDE.P11	07-MAR....C1	CZTEDE.P11	07-MAR....C5	CZTEDE.P11	07-MAR....C9
CZTEDE.P11	07-MAR....D1	CZTEDE.P11	07-MAR....D5	CZTEDE.P11	07-MAR....D9
CZTEDE.P11	07-MAR....E1	CZTEDE.P11	07-MAR....E5	CZTEDE.P11	07-MAR....E9
CZTEDE.P11	07-MAR....F1	CZTEDE.P11	07-MAR....F5	CZTEDE.P11	07-MAR....F9
CZTEDE.P11	07-MAR....G1	CZTEDE.P11	07-MAR....G5	CZTEDE.P11	07-MAR....G9
CZTEDE.P11	07-MAR....H1	CZTEDE.P11	07-MAR....H5	CZTEDE.P11	07-MAR....H9
CZTEDE.P11	07-MAR....I1	CZTEDE.P11	07-MAR....I5	CZTEDE.P11	07-MAR....I9
CZTEDE.P11	07-MAR....J1	CZTEDE.P11	07-MAR....J5	CZTEDE.P11	07-MAR....J9
CZTEDE.P11	07-MAR....K1	CZTEDE.P11	07-MAR....K5	CZTEDE.P11	07-MAR....K9
CZTEDE.P11	07-MAR....L1	CZTEDE.P11	07-MAR....L5	CZTEDE.P11	07-MAR....L9
CZTEDE.P11	07-MAR....M1	CZTEDE.P11	07-MAR....M5	CZTEDE.P11	07-MAR....M9
CZTEDE.P11	07-MAR....N1	CZTEDE.P11	07-MAR....N5	CZTEDE.P11	07-MAR....N9

CZTEDE.P11	07-MAR....B2	CZTEDE.P11	07-MAR....B6	CZTEDE.P11	07-MAR....B10
CZTEDE.P11	07-MAR....C2	CZTEDE.P11	07-MAR....C6	CZTEDE.P11	07-MAR....C10
CZTEDE.P11	07-MAR....D2	CZTEDE.P11	07-MAR....D6	CZTEDE.P11	07-MAR....D10
CZTEDE.P11	07-MAR....E2	CZTEDE.P11	07-MAR....E6	CZTEDE.P11	07-MAR....E10
CZTEDE.P11	07-MAR....F2	CZTEDE.P11	07-MAR....F6	CZTEDE.P11	07-MAR....F10
CZTEDE.P11	07-MAR....G2	CZTEDE.P11	07-MAR....G6	CZTEDE.P11	07-MAR....G10
CZTEDE.P11	07-MAR....H2	CZTEDE.P11	07-MAR....H6	CZTEDE.P11	07-MAR....H10
CZTEDE.P11	07-MAR....I2	CZTEDE.P11	07-MAR....I6	CZTEDE.P11	07-MAR....I10
CZTEDE.P11	07-MAR....J2	CZTEDE.P11	07-MAR....J6	CZTEDE.P11	07-MAR....J10
CZTEDE.P11	07-MAR....K2	CZTEDE.P11	07-MAR....K6	CZTEDE.P11	07-MAR....K10
CZTEDE.P11	07-MAR....L2	CZTEDE.P11	07-MAR....L6	CZTEDE.P11	07-MAR....L10
CZTEDE.P11	07-MAR....M2	CZTEDE.P11	07-MAR....M6	CZTEDE.P11	07-MAR....M10
CZTEDE.P11	07-MAR....N2	CZTEDE.P11	07-MAR....N6		

CZTEDE.P11	07-MAR....B3	CZTEDE.P11	07-MAR....B7
CZTEDE.P11	07-MAR....C3	CZTEDE.P11	07-MAR....C7
CZTEDE.P11	07-MAR....D3	CZTEDE.P11	07-MAR....D7
CZTEDE.P11	07-MAR....E3	CZTEDE.P11	07-MAR....E7
CZTEDE.P11	07-MAR....F3	CZTEDE.P11	07-MAR....F7
CZTEDE.P11	07-MAR....G3	CZTEDE.P11	07-MAR....G7
CZTEDE.P11	07-MAR....H3	CZTEDE.P11	07-MAR....H7
CZTEDE.P11	07-MAR....I3	CZTEDE.P11	07-MAR....I7
CZTEDE.P11	07-MAR....J3	CZTEDE.P11	07-MAR....J7
CZTEDE.P11	07-MAR....K3	CZTEDE.P11	07-MAR....K7
CZTEDE.P11	07-MAR....L3	CZTEDE.P11	07-MAR....L7
CZTEDE.P11	07-MAR....M3	CZTEDE.P11	07-MAR....M7
CZTEDE.P11	07-MAR....N3	CZTEDE.P11	07-MAR....N7

CZTEDE.P11	07-MAR....B4	CZTEDE.P11	07-MAR....B8
CZTEDE.P11	07-MAR....C4	CZTEDE.P11	07-MAR....C8
CZTEDE.P11	07-MAR....D4	CZTEDE.P11	07-MAR....D8
CZTEDE.P11	07-MAR....E4	CZTEDE.P11	07-MAR....E8
CZTEDE.P11	07-MAR....F4	CZTEDE.P11	07-MAR....F8
CZTEDE.P11	07-MAR....G4	CZTEDE.P11	07-MAR....G8
CZTEDE.P11	07-MAR....H4	CZTEDE.P11	07-MAR....H8
CZTEDE.P11	07-MAR....I4	CZTEDE.P11	07-MAR....I8
CZTEDE.P11	07-MAR....J4	CZTEDE.P11	07-MAR....J8
CZTEDE.P11	07-MAR....K4	CZTEDE.P11	07-MAR....K8
CZTEDE.P11	07-MAR....L4	CZTEDE.P11	07-MAR....L8
CZTEDE.P11	07-MAR....M4	CZTEDE.P11	07-MAR....M8
CZTEDE.P11	07-MAR....N4	CZTEDE.P11	07-MAR....N8