```
.REM  %
```

IDENTIFICATION

PRODUCT CODE:	AC-E490B-MC
PRODUCT NAME:	CZTUPB0 TM03/TU45 LGC PRT 2
DATE CREATED:	25 MAY 1978
MAINTAINER:	CSS - NASHUA
AUTHOR:	J. G. ADAMS/R. J. COLLINS
UPDATE INFORMATION:	DATE 29-FEB-1980 AUTHOR VIJAY ANANDWALA

TABLE OF CONTENTS

1. ABSTRACT

THIS PROGRAM IS DESIGNED TO SEQUENTIALLY TEST
THE DATA FORMATTING FUNCTIONALITY OF THE TM03
FORMATTER. EACH TEST WILL ATTEMPT TO ISOLATE FAILURES
TO THE MODULE LEVEL AND PROVIDE PRINTOUT INFORMATION
WHICH WILL IDENTIFY THE FAILING MODULE.
THE LEVEL OF FAULT ISOLATION IS POSSIBLE BECAUSE OF
TM03 THE STRUCTURE AND ITS MAINTAINENCE MODES.

2. REQUIREMENTS (HARDWARE)

A. ANY PDP-11 PROCESSOR
B. 8K OF CORE
C. CONSOLE TTY
D. TM03 MAGTAPE CONTROLLER
E. MASSBUS CONTROLLER (RH)
F. TU45 MAGTAPE TRANSPORT

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING BINARY PAPER TAPE.

4. STARTING PROCEDURE

THERE ARE TWO (2) STARTING ADDRESSES THAT MAY BE USED:
200(8) AND 210(8).

A. 200(8): STARTING AT THIS ADDRESS WILL CAUSE A PROGRAM
IDENTIFICATION HEADER TO BE PRINTED BEFORE TESTING
IS BEGUN.

B. 210(8): STARTING AT THIS ADDRESS WILL NOT PRINT THE
IDENTIFICATION HEADER AND IS THEREFORE GENERALLY
TO BE USED FOR RESTARTS RATHER THAN INITIAL START

** NOTE: SEE ALSO SEC 5. CONSOLE SWITCH SETTINGS
** TYPE ^C TO RESTART PROGRAM (@200)

4.1 AUTOMATIC MODE OPERATION

IF THIS PROGRAM IS LOADED AND RUN IN AUTOMATIC (CHAIN) MODES DEFAULT RESPONSES TO OPERATOR REQUESTS ARE USED. THE SOFTWARE SWR IS INVOKED WITH A SWITCH SETTING OF 100000 (HALT ON ERROR) IF IN ACT11 CHAIN MODE. NO OPERATOR INTERVENTION IS REQUIRED.

**EXCEPTION: IF THIS PROGRAM IS LOADED VIA TMDP CHAIN MODE THE PROGRAM WILL NOT TEST TM03 DRIVE #0, TU45 SLAVE #0.

4.2 SAMPLE START AT 200

NOTE: DEFAULT RESPONSES ARE SHOWN IN ANGLE BRACKETS <>, OPERATOR RESPONSES ARE SHOWN IN PARENTESES (), AND MEMORY LOCATIONS CONTAINING THE DEFAULT ARE SHOWN IN SQUARE BRACKETS [].

```
PARAMETER REQUEST: <DEFAULT> (RESPONSE)          [LOCATION:]

TM03-TU45 CONTROL LOGIC TEST PART II (CZTUPB0)
***ASSURE TAPE IS AT BOT***
TYPE ^C TO RESTART

REGISTER START: <172440> (CR)            [REGS:]
VECTOR ADDRESS: <224> (CR)               [VECT:]
TM03 DRIVE: <0> (CR)                     [DRVN:]
TU45 SLAVE: <0> (CR)                     [SLVN:]
        IF THE SOFTWARE SWR IS INVOKED:
SWR = <000000> NEW = (CR)                [SWREG:]
```

5. CONSOLE SWITCH SETTINGS

CONTROL:

1) CONTROL G <^G>;
   SELECTS THE SOFTWARE SWR AND ALLOWS NEW SWITCH SETTINGS.

   THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW=
   WHERE: XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWR.
   AFTER 'NEW=' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE
   OF THE FOLLOWING AT THE TTY:
     A) TYPE AN OCTAL NUMBER TO BE LOADED INTO THE SOFTWARE SWR.
     B) IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH
        REGISTER CONTENTS WILL NOT BE CHANGED.

2) CONTROL A <^A>;
   ALTERNATES THE SWITCH REGISTER FROM HARDWARE TO SOFTWARE & VICE VERSA.

3) CONTROL C <^C>;
   RESTART PROGRAM AT 200

4) CONTROL U <^U>;
   DELETES ALL CHARACTERS TYPED IN RESPONSE TO A REQUEST.

ALL SWITCHES ARE USED (0-15) AND THE NORMAL, OR DEFAULT, RUN
IS DONE WITH ALL SWITCHES SET TO ZERO (0).
ALL SWITCHES ARE DYNAMIC AND MAY BE CHANGED AT ANY TIME.

SW15: 1=HALT ON ERROR
       0=CONTINUE
SW14: 1=LOOP ON ERROR (SCOPE)
       0=CONTINUE
SW13: 1=DO NOT PRINT ERRORS
       0=PRINT ALL ERRORS
SW12: 1=DO CONTINUOUS CYCLE
       0=HALT AT END OF PASS
SW11: 1=INHIBIT ITERATIONS
       0=ITERATE EACH TEST ITS ASSIGNED AMOUNT
SW10: 1=HALT AT END OF CURRENT TEST
       0=CONTINUE TO NEXT TEST
SW8:   1=INHIBIT WRAP AROUND DATA CHECK
       0=DO DATA CHECKS
SW7:   1=INHIBIT WRAP AROUND STATUS CHECK
       0=DO STATUS CHECK
SW6:   1=SELECTABLE WRAP DATA PATTERN (IN SINGLE TEST)
       0=AUTO PATTERN
SW5-0: SELECT INDIVIDUAL TEST ** 00=DO ALL TESTS

6. ERROR PRINTOUTS

ERROR PRINTOUTS WILL APPEAR IN TWO FORMS, ONE FOR THE CONTROL LOGIC TESTS AND ANOTHER FOR THE DATA TESTS.

CONTROL LOGIC PRINTOUTS WILL CONTAIN A HEADER WHICH CALLS OUT THE TEST NUMBER, FUNCTION BEING TESTED, AND THE SUSPECT MODULE, OR MODULES ON THE FIRST LINE. THE SECOND LINE WILL CONTAIN INFORMATION AS TO THE ACTUAL ERROR. BOTH THE EXPECTED RESULT AND THE ACTUAL RESULT OF THE TEST WILL BE GIVEN. LINE THREE WILL SHOW THE CONTENTS OF THE MAJOR REGISTERS AT THE TIME OF THE ERROR AND LINE FOUR WILL PRINT THE ITERATION NUMBER WHEN APPLICABLE.

DATA TESTS WILL PRINT A HEADER CONTAINING THE TEST NUMBER, AND A DESCRIPTION OF THE WRAP AROUND FUNCTION UNDER TEST. FOLLOWING THE HEADER WILL BE A LIST OF THE MAJOR REGISTERS WITH THE EXPECTED AND ACTUAL VALUES. ANY BAD DATA WILL BE PRINTED (PER CHARACTER) FOLLOWING THE REGISTER INFORMATION OR FOLLOWING THE HEADER IF NO STATUS ERRORS WERE ENCOUNTERED.

5. THE FOLOWING ARE TWO EXAMPLES OF ERRORS DETECTED BY THE WRAP AROUND DATA TESTS. NOTE THAT EACH WRAP AROUND TEST MAY BE ACCOMPANIED BY EITHER A STATUS ERROR OF A DATA ERROR OR BOTH.

```
LOGIC TEST 1: WRAP 3, NRZ, NORMAL, ODD
BAD STATUS
CS1 EXPT 004270 RCVD 144270
CS2 EXPT 000100 RCVD 000100
DS  EXPT 010600 RCVD 150600
ER  EXPT 000000 RCVD 000100
```

THIS MESSAGE INDICATES BAD STATUS OF VPE (BIT 6 OF ER)

```
LOGIC TEST 3: WRAP 2, NRZ, NORMAL, ODD
BAD DATA
CN:0
G: 11111111
B: 11111011
CN:10
G: 00000000
B: 00001000
```

THIS MESSAGE SHOWS THAT DATA RECEIVED WAS NOT AS EXPECTED. CHARACTER ZERO (CN: 0) SHOWS THAT BIT TWO (2) WAS DROPPED, WHILE CHARACTER TEN (CN: 10) SHOWS BIT THREE (3) HAS BEEN PICKED UP
G: = EXPECTED DATA (GOOD)
B: = ACTUAL DATA (BAD)

7. OPERATION

THE PROCEDURES FOR OPERATING THIS PROGRAM ARE QUITE SIMPLE AND REQUIRE ONLY A FEW STEPS:

1. LOAD ADDRESS 200 OR 210
2. SET SWITCHES FOR DESIRED TEST CYCLE
3. PRESS START

ALL CONSOLE SWITCHES ARE DYNAMIC AND MAY BE CHANGED AT ANY TIME. THE NORMAL OPERATING SEQUENCE IS ALL SWITCHES DOWN (0). THE TEST WILL TAKE APPROXIMATELY 3 MINUTES TO RUN; HOWEVER, IF ITERATIONS ARE INHIBITED (SW11=1) THE TEST WILL RUN IN ABOUT 30 SECONDS. THE END OF PASS IS NOTED BY A PRINTOUT STATING END OF PASS, AND THE NUMBER OF THAT PASS.

SINGLE TEST SELECTION: (SW0-SW5)

WHEN SW0-SW5 ARE SET TO ZERO (00), THE SCHEDULAR WILL EXECUTE ALL TESTS IN SEQUENCE. IF SW0-SW5 ARE SET TO SOME SPECIFIC TEST NUMBER THEN THAT PARTICULAR TEST ONLY WILL BE EXECUTED UNTIL THE TEST SELECT NUMBER IS CHANGED. WHEN YOU WISH TO SELECT A PARTICULAR TEST, SET SW10 TO A ONE (1) IN ORDER TO STOP AT THE END OF THE CURRENT TEST BEFORE SELECTING A DIFFERENT TEST NUMBER. YOU MAY SELECT THAT NUMBER IN ANY DIRECTION (HIGHER OR LOWER) BECAUSE EACH TEST IS SELF CONTAINED.

WRAP AROUND DATA PATTERNS MAY BE SELECTED VIA SW6 WHEN IN SINGLE TEST MODE. A TELETYPE REQUEST IS MADE FOR THE DESIRED DATA PATTERN WHENEVER SWITCH TEN (SW10) AND SWITCH SIX (SW6) ARE SET TO A ONE (1) WHILE ONE OF THE WRAP TESTS IS SELECTED IN SW0-SW5.

8. SUBTEST SUMMARIES

NOTE: FOR TESTS 1-15

FOR THE MOST PART, THIS DIAGNOSTIC TESTS PARTICULAR AREAS OF THE TM03 LOGIC INDEPENDENT OF THE TU45. HOWEVER THERE ARE A FEW SIGNALS WHICH ARE REQUIRED FROM THE TU45 TO COMPLETE THE TESTS, AND AT LEAST ONE CASE WHERE TU45 FAILURES INTERFERE WITH THE TESTS. THE KNOWN CASES ARE LISTED HERE AND SHOULD BE CHECKED AS PART OF THE DEBUGGING.

1. MOL(SB)L: REQUIRED TO ENABLE CLOCK
2. CLOCK(SB)L: REQUIRED TO GENERATE ACCELERATION AND SHUTDOWN.
3. WRITE CLOCK(SB)L: USED IN WAM0 TO GENERATE DATA AND REC(SB)L
4. RSD0(SB)L: SHOULD NOT OCCUR DURING WRAP AROUND TESTS, BUT WILL INTERFERE WITH THEIR OPERATION IF CAUSED BY A FAILURE SUCH AS A GROUNDED OUTPUT FROM THE G056.

LOGIC TEST 1: WRP3, NRZ, NORMAL, ODD (BIT FIDDLER READ)

PROGRAMMED SEQUENCE:

TAPE CONTROL REGISTER IS LOADED WITH DENSITY 3, FORMAT 14, ODD PARITY WRP3 IS LOADED IN MAINT. REGISTER. READ FUNCTION IS LOADED, EXECUTING WRAP3 CONSISTS LOADING DATA CHARACTERS INTO MAINT. REGISTER DATA FIELD, WHERE THERE ARE MULTI-PLEXERS TO BIT FIDDLER, MM CLK IS TOGGLED TO CREATE RDS. THE BIT FIDDLER TRANSMIT DATA ACCESS MASSBUS DATA LINES. WHEN ALL THE DATA HAS BEEN TRANSMITTED AN EOR CLK IS TRANSMITTED TO N REGISTER WHICH BRINGS OPERATION TO A CLOSE.

LIKELY FAULT LOCATIONS: M8906, M8905-YB, MASSBUS P-LINES

CIRCUITS	PRINT REFERENCES
MASSBUS CHAR. ASSEMBLE	BF5
CLK. GENERATOR	BF2
MAINT. REGISTER DATA FIELD	MR2, MR3
RDS GENERATION	MR5

LOGIC TEST 2: WARP3, PE, NORMAL, ODD

JUST LIKE TEST 1 EXCEPT FOR DENSITY BITS.

LOGIC TEST 3: WRAP2, NRZ, NORMAL, ODD

PROGRAMMED SEQUENCE:

WRAP2 IS BIT FIDDLER WRITE. MM CLOCK IS MULTIPLEXED INTO WRT CLK SO THAT IT FORMS WRT STROBE. THE OUTPUT OF THE BIT FIDDLER IS CLOCKED INTO THE DATA FIELD OF THE MAINTENANCE REGISTER. SET UP CONSISTS OF MOVING NRZ, NORMAL FORMAT, ODD PARITY TO UNIT DESCRIPTION MAINT. REGISTER IS LOADED WITH WAM2 WRITE COMMAND IS ISSUED. AFTER THE ACCELERATION DELAY, MM CLOCK ARE GENERATED UNTIL ALL THE DATA HAS BEEN CLOCKED. SEQUENCE IS COMPLETED BY LOADING MANTENANCE REGISTER WITH EOR CLR. THE SEQUENCE IS REPEATED WITH VARYING DATA PATTERNS.

LIKELY FAULT LOCATIONS: M8906, M8905-YB, M8933

CIRCUITS	PRINT REFERENCES
BIT FIDDLER CHAR UNPACK	BF4
BIT FIDDLER DATA REQUEST	BF2
WRT STRB.	TCCM4
MAINT. REG. DATA FIELD	MR2, MR3

LOGIC TEST 4: WRP2, PE, NORMAL, ODD

THE TEST IS EXACTLY LIKE TEST 43 EXCEPT THAT PE WRT CLK ENBL L MUST BE ASSERTED BY M8932 TO ENABLE WR TO STROBE. THIS DOES NOT APPEN UNTIL THE PE WRITE CONTROL CIRCUIT HAS CLOCKED THROUGH THE PREAMBLE.

CIRCUITS	PRINT REFERENCES
(IN ADDITION TO TEST 44)	
PE WRITE CONTROL	TCPE3

LOGIC TEST 5: WRP1, NRZ, NORMAL, ODD

THIS TEST IS EXACTLY LIKE TEST 43 EXCEPT THE WRITE BUFFER (TCCM2) IS MULTIPLEXED TO THE MAINTENANCE REGISTER.

LIKELY FAULT LOCATIONS: M8933, M8934 (CRC GENERATOR)

CIRCUITS	PRINT REFERENCES
WRITE BUFFER	TCCM2
CRC GENERATOR	CNRZ2

LOGIC TEST 6: WRAP1, PE, NORMAL, ODD

IN PE MODE BOTH THE PREAMBLE AND POSAMBLE ARE CLOCKED THROUGH THE WRITE BUFFER IN ADDITION TO PHASE ENCODED DATA.

LIKELY FAULT LOCATIONS: M8932 (WRITE CONTROL STATES), M8933

CIRCUITS	PRINT REFERENCE
WRITE BUFFER	TCCM2
WRITE CONTROL	TCPE3

LOGIC TEST 7: WRAP0, NORMAL, ODD

WRAP 0 IS THE MOST COMPLETE OF THE TM03 WRAPAROUND DATA PATH. IT CONSISTS OF A WRITE OPERATION IN WHICH THE OUTPUT OF THE WRITE DATA BUFFER IS MULTIPLEXED TO THE READ DATA INPUTS, CHECKED AND LOADED INTO THE MAINTENANCE REGISTER FOR RETRIEVAL BY THE PROCESSOR. THE WHOLE OPERATION USES THE TYPE SYSTEM CLOCKS AND HAPPENS AT THE PROPER DATA RATES. MM CLK SERVES AS A FLAG ANNOUNCING WHEN A NEW CHARACTER HAS BEEN LOADED INTO THE MAINTENANCE REGISTER. IN PE MODE EVERY OTHER CHARACTER IS READ TO ALLOW SUFFICIENT PROCESSOR LOOP TIME. IN NRZ WRAP 0 IS EXPECTED TO PRODUCE LRC ERRORS BECAUSE THE TM03 DOES NOT WRITE THE LRC CHARACTER.

LIKELY FAULT LOCATIONS: M8934, M8933

CIRCUITS	PRINT REFERENCES
CRC GENERATOR	CNRZ2
CRC CHECKOUT	CNRZ3
CRC, CRC STROBE	TCCM4
READ LINE MULTIPLEXERS	TCCM6
MM CLK	MR5
CRC READ TIMING	CNRZ4
SHUTDOWN CIRCUITRY	TCCM5

LOGIC TEST 10: WRP0, PE, NORMAL, ODD

REPEAT OF TEST 7 IN PE MODE.

LIKELY FAULT LOCATIONS: M8901, M8932, M8933

CIRCUITS	PRINT REFERENCES
DATA DISCRIMINATOR	DS2, DS4, DS6
PHASE LOCKED CLOCK	DS3, DS5, DS7
SKEW BUFFER	DS3, DS5, DS7
PE WRITE MAJOR STATES	TCPE3
PE READ MAJOR STATES	TCPE5
WRAP 0 CIRCUIT TO BLOCK RLT RDS	TCPE3
DESKEW BUFFER READ COUNTER	TCPE4

LOGIC TEST 11: CORE DUMP WRITE, WAM2

REPEAT OF TEST 3 EXCEPT BIT FIDDLER OPERATES IN CORE DUMP MODE!

LIKELY FAULT LOCATION: M8906

LOGIC TEST 12: CORE DUMP READ, WAM3

REPEAT OF TEST 1 EXCEPT BIT FIDDLER OPERATES IN CORE DUMP MODE!

LIKELY FAULT LOCATION: M8906

LOGIC TEST 13: EVEN PARITY WRITE - WAM1

REPEAT OF TEST 5 EXCEPT EVEN PARITY IS SPECIFIED.

LIKELY FAULT LOCATION: M8933

LOGIC TEST 14: EVEN PARITY READ: WAM0,

REPEAT OF TEST 7 EXCEPT EVEN PARITY IS USED.

LIKELY FAULT LOCATIONS: M8933, M8934

LOGIC TEST 15: READ REVERSE, WAM3 (M8906)

REPEAT OF TEST 2 EXCEPT READ REVERSE COMMAND IS ISSUED.

LIKELY FAULT LOCATIONS: M8908, M8939

LOGIC TEST 16: CRC ERROR CORRECTION

THIS TEST SIMULATES A BAD TRACK ON TAPE RESULTING IN A CRC ERROR &
SUBSEQUENT CORRECTION OF DATA IN THE FAILING TRACK.
THE TEST PROCEEDS THROUGH THE FOLLOWING STEPS:

A:WRITE DATA USING WRAP 0
B:REWRITE DATA WITH DATA BITS IN ONE TRACK ALTERED USING WRAP 4
C:READ REVERSE USING WRAP3
D:REWRITE DATA AS IN STEPB USING WRAP4
AT THIS POINT THE DATA READ BACK HAS BEEN CORRECTED TO MATCH
THE DATA WRITTEN IN STEP A
E:REPEAT STEPS A-D ABOVE FOR EACH TRACK
F:REPEAT STEPS A-E ABOVE FOR ALL 1'S,ALL 0'S &
125125 DATA PATTERNS.

LOGIC TEST 17: CRC ERROR CORRECTION

THIS TEST SIMULATES MULTIPLE FAILING TRACKS & TEST THAT
NO ERROR CORRECTION IS PERFORMED. THE TEST SEQUENCE IS THE
SAME AS TEST 16 STEP A--STEP E. THE DATA PATTERN USED IS
125125.

LOGIC TEST 20: READ REVERSE WAM3 NRZ

REPEAT OF TEST 15 ABOVE (SEE ALSO TEST 2) EXCEPT THE TEST IS
PERFORMED IN NRZ MODE.

```
%
1844                            .LIST BIN,LOC,SEQ
```

```
                        .TITLE TM03-TU45 CONTROL LOGIC TEST-PART II
                        ;CZTUPB0
                        ;25 MAY 78
                        ;J.G. ADAMS/R. J. COLLINS

                        ;REVISED JUN 1977 BY J.G.ADAMS  ;++B CHANGED MODULE TYPEOUTS TO
                                                        ;++B REFLECT TM03 REGISTER SET
                        .MCALL  .$ACT11,.$EOP,$CATCH,$SAVE,$RESTORE,$CHAIN,$CHNMODE
                        .NLIST  MC
                        .LIST   ME
                        .ENABLE ABS,AMA


                        ;CONSOLE SWITCHES********
                        ;
                        ;SW15:  1=HALT ON ERROR
                        ;       0=CONTINUE
                        ;SW14:  1=LOOP ON ERROR
                        ;       0=CONTINUE
                        ;SW13:  1=DO NOT PRINT ERRORS
                        ;       0=PRINT ERRORS
                        ;SW12:  1=HALT AT END OF PASS
                        ;       0=CONTINOUS CYCLE
                        ;SW11:  1=INHIBIT ITERATIONS
                        ;       0=DO ITERATIONS
                        ;SW10:  1=HALT AT END OF EACH TEST
                        ;       0=CONTINUE
                        ;SW8:   1=NO WRAP DATA CHECK
                        ;       0=DO WRAP DATA CHECK
                        ;SW7:   1=NO WRAP STATUS CHECK
                        ;       0=DO WRAP STATUS CHECK
                        ;SW6:   1=SELECTABLE WRAP DATA PATTERN (IN SINGLE TEST)
                        ;       0=AUTO PATTERNS
                        ;SW0-5: SELECT TEST NUMBER :: 00=ALL TESTS
                    ;IF HARDWARE SWR <15::00> = 177777 OR NOT AVAILABLE USE SOFTWARE SWR
```

```
                                ;TU45 REGISTER BITS**********

                                ;15 ;14 ;13 ;12 ;11 ;10 ;09 ;08 ;07 ;06 ;05 ;04 ;03 ;02 ;01 ;00 ;
                                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;CS1:                           ;SC ;TRE;MCP;SPR;DNA;PSL;A17;A16;RDY;IE ;FUN;FUN;FUN;FUN;FUN;GO ;
                                ;   ;   ;   ;   ;=1 ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;
                                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;WC:                            ; WORD                                                          ;
                                ; COUNT                                                         ;
                                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;BA:                            ; BUS                                                           ;
                                ; ADDRESS                                                       ;
                                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;FC:                            ; FRAME                                                         ;
                                ; COUNT                                                         ;
                                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;CS2:                           ;DLT;WCE;UPE;NED;NEM;PGE;MXF;MDP;OR ;IR ;CLR;PAT;BAI;U2 ;U1 ;U0 ;
                                ;   ;   ;   ;   ;   ;   ;   ;ERR;   ;   ;   ;   ;   ;   ;   ;
                                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;DS:                            ;ATA;ERR;PIP;MOL;WRL;EOT;SPR;DPR;DRY;SSC;PES;SDN;IDB;TMK;BOT;SLA;
                                ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;
                                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;ER:                            ;CDE;UNS;OPI;DTE;NEF;CS ;FCE;NSG;PEF;INC;DAT;FMT;CNT;RMR;ILR;ILF;
                                ;CRC;   ;   ;   ;ITM;   ;   ;   ;LRC;VPE;BPE;   ;BPE;   ;   ;   ;
                                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;ATS:                           ; ATTENTION                                                     ;
                                ; SUMMARY                                                       ;
                                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;CC:                            ; CHECK                                                         ;
                                ; CHARACTER                                                     ;
                                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;DB:                            ; DATA                                                          ;
                                ; BUFFER                                                        ;
                                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;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 ;
                                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;DT:                            ; DRIVE      ; 7 ;   ;SLV;                                     ;
                                ; TYPE       ;CH ;   ;PR ;                                     ;
                                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;SN:                            ; SERIAL                                                        ;
                                ; NUMBER                                                        ;
                                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;TC:                            ;ACL;FCS;SAC;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 ;
                                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
                                        ;REGISTER EQUIVS**********

            000000                      R0=%0
            000001                      R1=%1
            000002                      R2=%2
            000003                      R3=%3
            000004                      R4=%4
            000005                      R5=%5
            000006                      SP=%6
            000007                      PC=%7


                                    ;ACT11 HOOK **********
            000764                      $SVPC=.                 ;SAVE CURRENT LOCATION CTR
            000046                      .=46
  000046    002374                      .WORD   $ENDAD          ;SET LOCATION 46
            000052                      .=52
  000052    000000                      .WORD   0               ;SET LOCATION 52 = 0
            000764                      .=$SVPC                 ;RESTORE LOCATION CTR


                                        ;TTY INTERRUPT VECTOR**********

            000060                      .=60
  000060    012372                      .WORD   TTINT   ;TTY INTERRUPT HANDLER ADDRESS
  000062    000340                      .WORD   340     ;PRIORITY LEVEL 7

                                        ;SOFTWARE SWITCH REGISTER**********
                                        ;USED IF HARDWARE SWR = 177777, OR NOT AVAIL.
            000176                      .=176
  000176    000000 .            SWREG:  .WORD   0


                                        ;START ADDRESS**********
            000200                      .=200
  000200    000137  001200              JMP     START   ;PROGRAM START

                                    ;RESTART ADDRESS**********
            000210                      .=210
  000210    000137  001670              JMP     ST2

                                        ;TM03 INTERRUPT VECTOR**********

            000224                      .=224
  000224    012362                      MTINT           ;TAPE INTERRUPT HANDLER ADDRESS
  000226    000340                      340
```

.

```
                                                .=510
                                                ;MASS BUS 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
00054C  172470                        SN:     172470
000542  172472                        TC:     172472

                                                ;ILLEGAL FUNCTION CODES

000544  005405                        ILFT:   5405
000546  007415                                7415
000550  016423                                16423
000552  020437                                20437
000554  022443                                22443
000556  025447                                25447
000560  031455                                31455
000562  033465                                33465
000564  036473                                36473

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

000566  177776                        PSW:    177776          ;PROCESSOR STATUS
000570  177570                        SWR:    177570          ;SWITCH REGISTER
000572  177560                        TKS:    177560          ;TTY READER STATUS
000574  177562                        TKB:    177562          ;TTY READ BUFFER
000576  177564                        TPS:    177564          ;TTY PUNCH STATUS
000600  177566                        TPB:    177566          ;TTY PUNCH BUFFER
000602  177777                        SERNUM: 177777          ;SERIAL NUMBER
000604  000011                        DRVTP:  011             ;DRIVE TYPE
000606  000020                        ITAMT:  20              ;ITERATION AMOUNT
000610  000224                        VECT:   224             ;INTERRUPT VECTOR(RH)
000612  172440                        REGS:   172440          ;STARTING REGISTER ADDRESS
```

```
2017                                    ;FLAGS AND COUNTERS*********
2018                                    ;NOTE ALL FLAGS AND COUNTERS ARE CLEARED ON STARTUP. PUT ANY
2019                                    ;ADDITIONAL FLAGS BETWEEN STFLGS (START OF FLAGS) AND ENDFLG
2020                                    ;(END OF FLAGS)
2021
2022  000614                            STFLGS:
2023  000614  000000                   TOB:    0
2024  000616  000000                   TIB:    0
2025  000620  000000                   HDRFL:  0
2026  000622  000000                   EMADDR: 0
2027  000624  000000                   DRVN:   0
2028  000626  000000                   TR00:   0
2029  000630  000000                   TR01:   0
2030  000632  000000                   TR02:   0
2031  000634  000000                   TR03:   0
2032  000636  000000                   TR04:   0
2033  000640  000000                   TR05:   0
2034  000642  000000                   TR06:   0
2035  000644  000000                   TR07:   0
2036  000646  000000                   TR10:   0
2037  000650  000000                   TR11:   0
2038  000652  000000                   TR12:   0
2039  000654  000000                   TR13:   0
2040  000656  000000                   TR14:   0
2041  000660  000000                   TR15:   0
2042  000662  000000                   NRZOF:  0
2043  000664  000000                   SLVN:   0
2044  000666  000000                   PFLG:   0
2045  000670  000000                   RTRN:   0
2046  000672  000000                   ERADD:  0
2047  000674  000000                   TEMP1:  0
2048  000676  000000                   TEMP2:  0
2049  000700  000000                   TEMP3:  0
2050  000702  000000                   ITCNT:  0
2051  000704  000000                   SAV1:   0
2052  000706  000000                   SAV2:   0
2053  000710  000000                   SAV3:   0
2054  000712  000000                   SCOLP:  0
2055  000714  000000                   ITRLP:  0
2056  000716  000000                   EXFL:   0
2057  000720  000000                   ATAF:   0
2058  000722  000000                   SLAF:   0
2059  000724  000000                   SSCF:   0
2060  000726  000000                   ERRF:   0
2061  000730  000000                   ASF:    0
2062  000732  000000                   SCF:    0
2063  000734  000000                   TREF:   0
2064  000736  000000                   PEXFL:  0
2065  000740  000000                   STFLG:  0
2066  000742  000000                   LTADD:  0
2067  000744  000000                   T24FL:  0
2068  000746  000000                   ADDFL:  0
2069  000750  000000                   WAM:    0
2070  000752  000000                   FUN:    0
2071  000754  000000                   DATC:   0
2072  000756  000000                   WTAD:   0
```

```
2073  000760  000000                DATAD:  0
2074  000762  000000                RDAD:   0
2075  000764  000000                W2FLG:  0
2076  000766  000000                DERFL:  0
2077  000770  000000                PREFL:  0
2078  000772  000000                SERFL:  0
2079  000774  000000                CRCNT:  0
2080  000776  000000                UDES:   0
2081  001000  000000                WPGFL:  0
2082  001002  000000                PATRN:  0
2083  001004  000000                STATF:  0
2084  001006  000000                RDRVF:  0
2085  001010  000000                RCDP:   0
2086  001012  000000                STATC:  0
2087  001014  000000                SKAT:   0
2088  001016  000000                PCNTR:  0               ;PASS COUNTER
2089  001020  000000                DCHKFL: .WORD   0               ;DATA CHECK FLAG 0/1 = CHECK/DO NOT CHECK
2090  001022  000000                CRCFLG: .WORD   0               ;CRC CORRECTION TEST IN PROGRESS
2091  001024                        ENDFLG:
2092
2093                                        ;EXPT WRAP STATUS*******
2094
2095  001024  000000                WCS1:   0
2096  001026  000000                WCS2:   0
2097  001030  000000                WDS:    0
2098  001032  000000                WER:    0
2099
2100                                        ;DATA PATTERN GENERATORS*****
2101
2102  001034                        DATBL:
2103  001034  005162                DATA0:  DAT1    ;ALL ONE BITS
2104  001036  005202                DATA1:  DAT2    ;ALL ZERO BITS
2105  001040  005206                DATA2:  DAT3    ;ALTERNATING ONE/ZERO BITS
2106  001042  005214                DATA3:  DAT4    ;ALTERNATING PARITY CHARACTERS
2107
2108                                        ;CORE DUMP PATTERNS*******
2109
2110  001044  000005                WCDP2:  5
2111  001046  000005                        5
2112  001050  000012                        12
2113  001052  000012                        12
2114  001054  000000                        0
2115  001056  000017                WCDP0:  17
2116  001060  000017                        17
2117  001062  000017                        17
2118  001064  000017                        17
2119  001066  000000                        0
```

```
2121
2122                                    ;LOGIC TEST ENTRY TABLE**********
2123
2124  001070  000000                TSTTBL: 0
2125  001072  000000                        0
2126  001074  002436                        LT1
2127  001076  002436                        LT1
2128  001100  002550                        LT2
2129  001102  002550                        LT2
2130  001104  002622                        LT3
2131  001106  002622                        LT3
2132  001110  002730                        LT4
2133  001112  002730                        LT4
2134  001114  003002                        LT5
2135  001116  003002                        LT5
2136  001120  003110                        LT6
2137  001122  003110                        LT6
2138  001124  003166                        LT7
2139  001126  003166                        LT7
2140  001130  003274                        LT10
2141  001132  003274                        LT10
2142  001134  003346                        LT11
2143  001136  003346                        LT11
2144  001140  003460                        LT12
2145  001142  003460                        LT12
2146  001144  003606                        LT13
2147  001146  003606                        LT13
2148  001150  003656                        LT14
2149  001152  003656                        LT14
2150  001154  003726                        LT15
2151  001156  003726                        LT15
2152  001160  004010                        LT16
2153  001162  004010                        LT16
2154  001164  004376                        LT17
2155  001166  004376                        LT17
2156  001170  004710                        LT20
2157  001172  004710                        LT20
2158  001174  002330                TADX:   .WORD   TEND
2159  001176  000020                TLAST:  .WORD   20              ;CONTAINS # OF TESTS
```

```
                                        .EVEN
                                        ;PROGRAM START AND HOUSEKEEPING**********

001200  012706  000500          START: MOV     #500,SP          ;SET STACK POINTER
001204  013746  000004                 MOV     @#4,-(SP)        ;SAVE ERROR TRAP
001210  013746  000006                 MOV     @#6,-(SP)
001214  012737  001240  000004         MOV     #1$,@#4          ;SET TIME OUT TRAP TO GO TO 1$
001222  005037  000006                 CLR     @#6
001226  022777  177777  177334         CMP     #177777,@SWR     ;USE SOFTARE SWITCH.IF SWR - 177777
001234  001402                        BEQ     2$               ;OR TIMES OUT
001236  000404                        BR      3$               ;OTHERWISE USE HARDWARE SWR
001240  022626                 1$:    CMP     (SP)+,(SP)+      ;RESET STACK
001242  012737  000176  000570  2$:    MOV     #SWREG,SWR       ;SET SWR = TO ADDRESS OF SOFTARE SWR
001250  012637  000006         3$:    MOV     (SP)+,@#6        ;RESTORE ERROR TRAP
001254  012637  000004                 MOV     (SP)+,@#4
001260  005027                        CLR     (PC)+            ;;CLEAR CHAIN INDICATOR
 (1)    001262  000000          CHNFLG: .WORD   0                ;;CHAIN MODE INDICATOR
 (1)                                                           ;;1/0 = CHAIN/NOT CHAIN MODE
 (1)    001264  022737  002374  000042         CMP     #$ENDAD,@#42     ;;BRANCH IF LOADED VIA ACT11 CHAIN MODE
 (1)    001272  001404                        BEQ     50$
 (1)    001274  005737  000042                TST     @#42             ;;BRANCH IF IN DUMP MODE
 (1)    001300  001413                        BEQ     52$
 (1)    001302  000406                        BR      51$
 (1)    001304  012737  000176  000570  50$:   MOV     #SWREG,SWR       ;;INVOKE SOFTWARE SWR
 (1)    001312  012777  100000  177250         MOV     #100000,@SWR     ;;WITH HALT ON ERROR SET
 (1)    001320  005237  001262         51$:   INC     CHNFLG           ;;SET CHNFLG = CHAIN MODE
 (1)    001324  000137  001714                JMP     TSCD             ;;GO TO CHAIN ADDRESS
 (1)    001330                         52$:
2177    001330  122737  000006  000041  4$:    CMPB    #6,@#41          ;BRANCH IF NOT LOADED VIA TMDP
2178    001336  001004                        BNE     5$
2179    001340  012704  014573                MOV     #MSG62,R4        ;ADVISE USER TO REMOVE MEDIA FROM UUT
2180    001344  004737  013054                JSR     PC,TTOUT
2181    001350  012704  013776         5$:    MOV     #MSG1,R4
2182    001354  004737  013054                JSR     PC,TTOUT         ;PRINT TITLE
2183    001360  112737  000043  013776         MOVB    #'#,MSG1         ;DO NOT PRINT TITLE ON RESTART
2184    001366  012704  014467                MOV     #MSG44,R4
2185    001372  004737  013054                JSR     PC,TTOUT         ;REQUEST REGISTER ADDRESS
2186    001376  013703  000612                MOV     REGS,R3
2187    001402  004737  013202                JSR     PC,OCTP          ;PRINT CURRENT ADDRESS
2188    001406  012705  000612                MOV     #REGS,R5         ;SET ADDRESS SAVE LOC
2189    001412  012701  000007                MOV     #7,R1            ;SET SIZE OF RESPONSE
2190    001416  012702  176400                MOV     #176400,R2       ;SET UPPER LIMIT
2191    001422  012703  172300                MOV     #172300,R3       ;SET LOWER LIMIT
2192    001426  004737  012514                JSR     PC,TTR           ;GO GET RESPONSE
2193    001432  012704  014511                MOV     #MSG45,R4
2194    001436  004737  013054                JSR     PC,TTOUT         ;REQUEST VECTOR
2195    001442  013703  000610                MOV     VECT,R3
2196    001446  004737  013202                JSR     PC,OCTP          ;PRINT CURRENT VECTOR
2197    001452  012705  000610                MOV     #VECT,R5         ;SET ADDRESS SAVE LOC
2198    001456  012701  000004                MOV     #4,R1            ;SET SIZE OF RESPONSE
2199    001462  012702  000224                MOV     #224,R2          ;SET UPPER LIMIT
2200    001466  012703  000150                MOV     #150,R3          ;SET LOWER LIMIT
2201    001472  004737  012514                JSR     PC,TTR           ;GO GET RESPONSE
2202    001476  013700  000610                MOV     VECT,R0          ;GET VECTOR
2203    001502  012720  012362                MOV     #MTINT,(R0)+     ;LOAD INTERRUPT ADDRESS IN VECTOR
2204    001506  012710  000340                MOV     #340,(R0)        ;LOAD PRIORITY
```

```
2205  001512  013700  000612                MOV     REGS,R0         ;GET START OF REGS
2206  001516  012701  000016                MOV     #16,R1          ;SET NUMBER OF REGS
2207  001522  012702  000510                MOV     #C1,R2          ;GET START OF TABLE
2208  001526  010022                6$:     MOV     R0,(R2)+        ;BUILD TABLE
2209  001530  062700  000002                ADD     #2,R0           ;BUMP ADDRESS
2210  001534  005301                        DEC     R1              ;SEE IF DONE
2211  001536  001373                        BNE     6$              ;IF NOT: BR
2212  001540  012702  000614                MOV     #STFLGS,R2
2213  001544  012700  000210                MOV     #ENDFLG-STFLGS,R0       ;GET # OF FLAGS TO CLEAR
2214  001550  006200                        ASR     R0              ;FORM COUNT
2215  001552  005022                7$:     CLR     (R2)+           ;CLEAR FLAGS + COUNTERS
2216  001554  005300                        DEC     R0
2217  001556  001375                        BNE     7$
2218  001560  012704  014533                MOV     #MSG57,R4       ;REQUEST TM03 DRIVE #
2219  001564  004737  013054                JSR     PC,TTOUT
2220  001570  013703  000624                MOV     DRVN,R3         ;GET CURRENT DRIVE
2221  001574  004737  013202                JSR     PC,OCTP         ;AND TYPE IT
2222  001600  012705  000624                MOV     #DRVN,R5        ;TTR ROUTINE RETURNS DRIVE TO (R5)
2223  001604  012701  000002                MOV     #2,R1           ;LIMIT RESPONSE TO 1 CHARACTER
2224  001610  012702  000007                MOV     #7,R2           ;BETWEEN 0 AND 7
2225  001614  012703  000000                MOV     #0,R3
2226  001620  004737  012514                JSR     PC,TTR          ;GET RESPONSE & PUT IN DRVN
2227  001624  012704  014551                MOV     #MSG58,R4       ;REQUEST SLAVE #
2228  001630  004737  013054                JSR     PC,TTOUT
2229  001634  013703  000664                MOV     SLVN,R3         ;GET CURRENT SLAVE #
2230  001640  004737  013202                JSR     PC,OCTP         ;AND TYPE IT
2231  001644  012705  000664                MOV     #SLVN,R5        ;TTR ROUTINE RETURNS REPONSE TO (R5)
2232  001650  012701  000002                MOV     #2,R1           ;LIMIT RESPONSE TO 1 CHARACTER
2233  001654  012702  000007                MOV     #7,R2           ;BETWEEN 0-7
2234  001660  012703  000000                MOV     #0,R3
2235  001664  004737  012514                JSR     PC,TTR          ;GET RESPUNSE & PUT IT IN SLVN
2236
2237                           ;START 210
2238  001670  012706  000500        ST2:    MOV     #500,SP         ;SET STACK PTR
2239  001674  005037  001006                CLR     RDRVF           ;CLEAR READ REVERSE FLAG
2240  001700  005037  001016                CLR     PCNTR
2241  001704  005037  001022                CLR     CRCFLG          ;SET CRC FLAG = CRC NOT IN PROGRESS
2242  001710  004737  013634                JSR     PC,GTSWR        ;GET SOFTWARE SWITCHES
```

```
2244
2245                                                    ;TEST SCHEDULAR*********
2246
2247  001714  052777  000100  176650  TSCD:   BIS     #100,@TKS       ;SET KEYBOARD IE BIT
2248  001722  005037  001000          CLR     WPGFL           ;CLEAR WRAP PATRN FLAG
2249  001726  005037  000740          CLR     STFLG           ;CLEAR SINGLE TEST FLAG
2250  001732  017700  176632          MOV     @SWR,R0
2251  001736  042700  177700          BIC     #177700,R0      ;BRANCH IF SINGLE TEST SELECTED
2252  001742  001154                  BNE     STSCD           ;GO SELECT SINGLE TEST
2253  001744  005737  001262          TST     CHNFLG          ;;BRANCH IF NOT IN CHAIN MODE
 (1)  001750  001511                  BEQ     TSCDA
 (1)  001752  012737  177777  000624  MOV     #-1,DRVN        ;;INITIALIZE DRIVE #
 (1)  001760  012737  177777  000664  NXTDRV: MOV     #-1,SLVN        ;;INITIALIZE SLAVE #
 (1)  001766  012777  000040  176524  1$:     MOV     #40,@CS         ;;INIT CONTROLLER
 (1)  001774  005237  000624          INC     DRVN            ;;STEP DRIVE #
 (1)  002000  022737  000010  000624  CMP     #10,DRVN        ;;EXIT IF ALL DRIVES TESTED
 (1)  002006  001553                  BEQ     $DONE           ;;FOR AVAILABILITY
 (1)  00201C  013777  000624  176502  MOV     DRVN,@CS        ;;LOAD DRIVE #
 (1)  002016  005777  176466          TST     @C1             ;;ACCESS DRIVE
 (1)  002022  032777  010000  176470  BIT     #10000,@CS      ;BRANCH IF DRIVE NON EXISTANT
 (1)  002030  001356                  BNE     1$              ;(NED = 1)
 (1)  002032  005237  000664  NXTSLV: INC     SLVN            ;;STEP SLAVE # AND BRANCH
 (1)  002036  001011                  BNE     1$              ;;IF NOT SLAVE 0
 (1)  002040  005737  000624          TST     DRVN            ;;BRANCH IF NOT DRIVE # 0
 (1)  002044  001006                  BNE     1$
 (1)  002046  122737  000006  000041  CMPB    #6,@#41         ;;BRANCH IF NOT TMDP
 (1)  002054  001002                  BNE     1$
 (1)  002056  005237  000664          INC     SLVN            ;;STEP TO SLAVE # 1
 (1)  002062  022737  000010  000664  1$:     CMP     #10,SLVN        ;;BRANCH IF ALL SLAVES TESTED
 (1)  002070  001733                  BEQ     NXTDRV          ;;FOR AVAILABILITY
 (1)  002072  013777  000664  176442  MOV     SLVN,@TC        ;;LOAD SLAVE UNIT #
 (1)  002100  032777  002000  176430  BIT     #2000,@DT       ;;BRANCH IF SLAVE NOT
 (1)  002106  001751                  BEQ     NXTSLV          ;;PRESENT (SPR = 0)
2254  002110  032777  010000  176404  BIT     #10000,@DS      ; TEST FOR MOL
2255  002116  001026                  BNE     TSCDA           ;BRANCH IF MOL
2256  002120  012704  014667          MOV     #MSG64,R4       ;ELSE PRINT NOT ON LINE
2257  002124  004737  013054          JSR     PC,TTOUT        ;PRINT MESSAGE
2258  002130  013703  000624          MOV     DRVN,R3         ;DRIVE NUMBER
2259  002134  004737  013202          JSR     PC,OCTP         ;PRINT NUMBER
2260  002140  012704  014701          MOV     #MSG65,R4       ;ADDRESS OF MESSAGE
2261  002144  004737  013054          JSR     PC,TTOUT        ;PRINT MESSAGE
2262  002150  013703  000664          MOV     SLVN,R3         ;SLAVE NUMBER
2263  002154  004737  013202          JSR     PC,OCTP         ;PRINT NUMBER
2264  002160  012704  014711          MOV     #MSG66,R4       ;ADDR OF MESSAGE
2265  002164  004737  013054          JSR     PC,TTOUT        ;PRINT IT
2266  002170  000137  002032          JMP     NXTSLV
2267  002174  012737  001070  000742  TSCDA:  MOV     #TSTTBL,LTADD
2268  002202  062737  000004  000742  TSCD0:  ADD     #4,LTADD
2269  002210  013737  000742  000714  TSCD1:  MOV     LTADD,ITRLP
2270  002216  062737  000002  000714          ADD     #2,ITRLP        ;SET ITERATION ADDRESS
2271  002224  005037  000620          CLR     HDRFL           ;CLEAR PRINT HEADER FLAG
2272  002230  017700  176506          MOV     @LTADD,R0       ;SET POINTER TO TEST
2273  002234  000110                  JMP     (R0)            ;GO TO TEST
2274  002236  032777  002000  176324  TSCD2:  BIT     #2000,@SWR      ;SEE IF HALT ON TEST
2275  002244  001403                  BEQ     TSCD3           ;IF NOT: BR
2276  002246  000000                  HALT
```

```
 2277  002250  005037  001000                  CLR     WPGFL           ;CLEAR WRAP DATA GENERATOR FLAG
 2278  002254  005737  000740          TSCD3:  TST     STFLG           ;SE IF SINGLE TEST
 2279  002260  001750                          BEQ     TSCD0           ;IF NOT: BR
 2280  002262  017700  176302                  MOV     @SWR,R0
 2281  002266  042700  177700                  BIC     #177700,R0      ;MASK TEST NUMBER
 2282  002272  001610                          BEQ     TSCD            ;IF SO: BR
 2283  002274  012737  000001  000740  STSCD:  MOV     #1,STFLG        ;SET SINGLE TEST FLAG
 2284  002302  023700  001176                  CMP     TLAST,R0        ;SEE IF EXCEEDED TESTS
 2285  002306  002410                          BLT     TEND            ;IF SO: BR
 2286  002310  006300                          ASL     R0
 2287  002312  006100                          ROL     R0              ;SET TABLE MODIFIER
 2288  002314  012737  001070  000742          MOV     #TSTTBL,LTADD
 2289  002322  060037  000742                  ADD     R0,LTADD        ;SET TEST POINTER
 2290  002326  000730                          BR      TSCD1
 2291  002330  005737  001262          TEND:   TST     CHNFLG          ;BRANCH IF IN CHAIN MODE
 2292  002334  001236                          BNE     NXTSLV
 2293  002336  012704  014450          $DONE:  MOV     #MSG41,R4
 2294  002342  004737  013054                  JSR     PC,TTOUT        ;PRINT END OF PASS
 2295  002346  013703  001016                  MOV     PCNTR,R3
 2296  002352  004737  013202                  JSR     PC,OCTP         ;PRINT PASS NUMBER
 2297  002356  005000                          CLR     R0
 2298  002360  005300                  1$:     DEC     R0              ;DELAY WAITING FOR
 2299  002362  001376                          BNE     1$              ;TTY TO FINISH
 2300  002364  013700  000042                  MOV     @#42,R0         ;GET ACT11 RETURN ADDRESS
  (1)  002370  001405                          BEQ     HERE            ;BRANCH IF NOT ACT11
  (1)  002372  000005                          RESET
  (1)  002374  004710                  $ENDAD: JSR     PC,(R0)
  (1)  002376  000240                          NOP
  (1)  002400  000240                          NOP
  (1)  002402  000240                          NOP
  (1)  002404  000240                  HERE:   NOP
 2301  002406  005737  001262                  TST     CHNFLG          ;BRANCH IF IN CHAIN MODE
 2302  002412  001005                          BNE     TENDX
 2303  002414  032777  010000  176146          BIT     #10000,@SWR     ;SEE IF HALT ON PASS
 2304  002422  001001                          BNE     TENDX           ;IF NOT: BR
 2305  002424  000000                          HALT
 2306  002426  005237  001016          TENDX:  INC     PCNTR           ;BUMP PASS COUNTER
 2307  002432  000137  001714                  JMP     TSCD            ;RESTART
 2308
```

```
                                                ;THESE TESTS CHECK DATA FORMATTING
                                                ;AND TRANSFER THROUGH THE TM03 WRAP AROUND MODFS

                                                ;LOGIC TEST 1: WRAP 3, NRZ, NORMAL ODD *****

002436  012737  004270  001024  LT1:    MOV     #4270,WCS1      ;SET EXPT CS1
002444  012737  000100  001026          MOV     #100,WCS2       ;SET EXPT CS2
002452  012737  010600  001030          MOV     #10600,WDS      ;SET EXPT DS
002460  012737  000000  001032          MOV     #0,WER          ;SET EXPT ER
002466  012737  014726  000622          MOV     #MSLT1,EMADDR   ;SET HEADER
002474  012737  001700  000776          MOV     #1700,UDES      ;SET NRZ,NORMAL, ODD
002502  005037  001002          LT1A:   CLR     PATRN           ;POINT TO PATTERN 0
002506  012737  002514  000712          MOV     #LT1B,SCOLP     ;SET SCOPE ADDRESS
002514  004737  005342          LT1B:   JSR     PC,WAM3         ;GO DO WRAP 3
002520  005237  001002                  INC     PATRN           ;BUMP PATTERN POINTER
002524  032737  000004  001002          BIT     #4,PATRN        ;SEE IF DONE
002532  001770                          BEQ     LT1B            ;IF NOT: BR
002534  004737  012136                  JSR     PC,ITER         ;GO SEE IF ITERATIONS
002540  005037  001006                  CLR     RDRVF           ;CLEAR READ REVERSE FLAG
002544  000137  002236                  JMP     TSCD2           ;RETURN TO SCHEDULAR

                                                ;LOGIC TEST 2: WRAP 3, PE, NORMAL, ODD*****

002550  000240          LT2:    NOP
002552  012737  004270  001024  LT2A:   MOV     #4270,WCS1      ;SET EXPT CS1
002560  012737  000100  001026          MOV     #100,WCS2       ;SET EXPT CS2
002566  012737  010640  001030          MOV     #10640,WDS      ;SET EXPT DS
002574  012737  000000  001032          MOV     #0,WER          ;SET EXPT WER
002602  012737  014774  000622          MOV     #MSLT2,EMADDR   ;SET HEADER
002610  012737  002300  000776          MOV     #2300,UDES      ;SET PE, NORMAL, ODD
002616  000137  002502                  JMP     LT1A            ;EXECUTE TEST SEQUENCE

                                                ;LOGIC TEST 3: WRAP 2, NRZ, NORMAL, ODD*****

002622  012737  004260  001024  LT3:    MOV     #4260,WCS1      ;SET EXPT CS1
002630  012737  000100  001026          MOV     #100,WCS2       ;SET EXPT CS2
002636  012737  010600  001030          MOV     #10600,WDS      ;SET EXPT DS
002644  012737  000000  001032          MOV     #0,WER          ;SET EXPT WER
002652  012737  015041  000622          MOV     #MSLT3,EMADDR   ;SET HEADER
002660  012737  001700  000776          MOV     #1700,UDES      ;SET TO NRZ,NORMAL, ODD
002666  005037  001002          LT3A:   CLR     PATRN           ;POINT TO PATTERN 0
002672  012737  002700  000712          MOV     #LT3B,SCOLP     ;SET SCOPE ADDRESS
002700  004737  005276          LT3B:   JSR     PC,WAM2         ;GO DO WRAP 2
002704  005237  001002                  INC     PATRN           ;BUMP POINTER
002710  032737  000004  001002          BIT     #4,PATRN        ;SEE IF DONE
002716  001770                          BEQ     LT3B            ;IF NOT: BR
002720  004737  012136                  JSR     PC,ITER         ;GO SEE IF ITERATIONS
002724  000137  002236                  JMP     TSCD2           ;RETURN TO SCHEDULAR
```

```
2359
2360                                                    ;LOGIC TEST 4: WRAP 2, PE, NORMAL, ODD*****
2361
2362  002730  000240                          LT4:      NOP
2363  002732  012737  004260  001024          LT4A:     MOV     #4260,WCS1      ;SET EXPT CS1
2364  002740  012737  000100  001026                    MOV     #100,WCS2       ;SET EXPT CS2
2365  002746  012737  010640  001030                    MOV     #10640,WDS      ;SET EXPT DS
2366  002754  012737  000000  001032                    MOV     #0,WER          ;SET EXPT WER
2367  002762  012737  015107  000622                    MOV     #MSLT4,EMADDR   ;SET HEADER
2368  002770  012737  002300  000776                    MOV     #2300,UDES      ;SET PE, NORMAL, ODD
2369  002776  000137  002666                          JMP     LT3A            ;GO EXECUTE TEST SEQUENCES
2370
2371                                                    ;LOGIC TEST 5: WRAP 1, NRZ, NORMAL, ODD*****
2372
2373  003002  012737  004260  001024  LT5:    MOV     #4260,WCS1      ;SET EXPT CS1
2374  003010  012737  000100  001026          MOV     #100,WCS2       ;SET EXPT CS2
2375  003016  012737  010600  001030          MOV     #10600,WDS      ;SET EXPT DS
2376  003024  012737  000000  001032          MOV     #0,WER          ;SET EXPT WER
2377  003032  012737  015154  000622          MOV     #MSLT5,EMADDR   ;SET HEADER
2378  003040  012737  001700  000776          MOV     #1700,UDES      ;SET NRZ, NORMAL, ODD
2379  003046  005037  001002          LT5A:   CLR     PATRN           ;POINT TO PATTERN ZERO
2380  003052  012737  003060  000712          MOV     #LT5B,SCOLP     ;SET SCOPE ADDRESS
2381  003060  004737  005266          LT5B:   JSR     PC,WAM1         ;GO DO WRAP 1
2382  003064  005237  001002                  INC     PATRN           ;BUMP POINTER
2383  003070  032737  000004  001002          BIT     #4,PATRN        ;SEE IF DONE
2384  003076  001770                          BEQ     LT5B            ;IF NOT: BR
2385  003100  004737  012136                  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2386  003104  000137  002236                  JMP     TSCD2           ;RETURN TO SCHEDULAR
2387
2388                                                    ;LOGIC TEST 6: WRAP 1, PE, NORMAL, ODD*****
2389
2390  003110  000240                          LT6:      NOP
2391  003112  004737  011634          LT6A:   JSR     PC,PPGEN        ;GO GENERATE PRE/POSTAMBLE
2392  003116  012737  004260  001024          MOV     #4260,WCS1      ;SET EXPT CS1
2393  003124  012737  000100  001026          MOV     #100,WCS2       ;SET EXPT CS2
2394  003132  012737  010640  001030          MOV     #10640,WDS      ;SET EXPT DS
2395  003140  012737  000000  001032          MOV     #0,WER          ;SET EXPT WER
2396  003146  012737  015222  000622          MOV     #MSLT6,EMADDR   ;SET HEADER
2397  003154  012737  002300  000776          MOV     #2300,UDES      ;SET PE, NORMAL, ODD
2398  003162  000137  003046                  JMP     LT5A            ;GO EXECUTE TEST SEQUENCE
```

```
2400
2401                                                    ;LOGIC TEST 7: WRAP 0, NRZ,NORMAL, ODD*****
2402
2403  003166  012737  144260  001024  LT7:    MOV     #144260,WCS1    ;SET EXPT CS1
2404  003174  012737  000100  001026          MOV     #100,WCS2       ;SET EXPT CS2
2405  003202  012737  150600  001030          MOV     #150600,WDS     ;SET EXPT DS
2406  003210  012737  000200  001032          MOV     #200,WER        ;SET EXPT ER
2407  003216  012737  015267  000622          MOV     #MSLT7,EMADDR   ;SET HEADER
2408  003224  012737  001700  000776          MOV     #1700,UDES      ;SET NRZ, NORMAL, ODD
2409  003232  005037  001002          LT7A:   CLR     PATRN           ;POINT TO PATTERN 0
2410  003236  012737  003244  000712          MOV     #LT7B,SCOLP     ;SET SCOPE ADDRESS
2411  003244  004737  005222          LT7B:   JSR     PC,WAMO         ;GO DO WRAP 0
2412  003250  005237  001002                  INC     PATRN           ;BUMP POINTER
2413  003254  032737  000004  001002          BIT     #4,PATRN        ;SEE IF DONE
2414  003262  001770                          BEQ     LT7B            ;IF NOT: BR
2415  003264  004737  012136                  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2416  003270  000137  002236                  JMP     TSCD2           ;RETURN TO SCHEDULAR
2417
2418                                                    ;LOGIC TEST 10: WRAP 0, PE, NORMAL, ODD*****
2419
2420  003274  000240          LT10:   NOP
2421  003276  012737  004260  001024  LT10A:  MOV     #4260,WCS1      ;SET EXPT CS1
2422  003304  012737  000100  001026          MOV     #100,WCS2       ;SET EXPT CS2
2423  003312  012737  010640  001030          MOV     #10640,WDS      ;SET EXPT DS
2424  003320  012737  000000  001032          MOV     #0,WER          ;SET EXPT ER
2425  003326  012737  015335  000622          MOV     #MSLT10,EMADDR  ;SET HEADER
2426  003334  012737  002300  000776          MOV     #2300,UDES      ;SET PE, NORMAL, ODD
2427  003342  000137  003232                  JMP     LT7A            ;GO EXECUTE TEST SEQUENCE
```

```
2429                                                ;LOGIC TEST 11: CORE DUMP WRITE, WAM2*******
2430
2431  003346  012737  004260  001024  LT11:   MOV     #4260,WCS1      ;SET EXPT CS1
2432  003354  012737  000100  001026          MOV     #100,WCS2       ;SET EXPT CS2
2433  003362  012737  010600  001030          MOV     #10600,WDS      ;SET EXPT DS
2434  003370  012737  000000  001032          MOV     #0,WER          ;SET EXPT ER
2435  003376  012737  015403  000622          MOV     #MSLT11,EMADDR  ;SET HEADER
2436  003404  012737  001720  000776          MOV     #1720,UDES      ;SET NRZ, CORE DUMP, ODD
2437  003412  005037  001002                  CLR     PATRN           ;POINT TO PATTERN 0
2438  003416  012737  003424  000712          MOV     #LT11A,SCOLP    ;SET SCOPE ADDRESS
2439  003424  004737  005276          LT11A:  JSR     PC,WAM2         ;GO DO WAM 2
2440  003430  022737  000002  001002          CMP     #2,PATRN        ;SEE IF DONE
2441  003436  001404                          BEQ     LT11X           ;IF SO: BR
2442  003440  012737  000002  001002          MOV     #2,PATRN        ;SELECT PATTERN 2
2443  003446  000766                          BR      LT11A           ;CONTINUE
2444  003450  004737  012136          LT11X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2445  003454  000137  002236                  JMP     TSCD2           ;RETURN TO SCHEDULES
2446
2447                                                ;LOGIC TEST 12: CORE DUMP READ, WAM 3*****
2448
2449  003460  012737  004270  001024  LT12:   MOV     #4270,WCS1      ;SET EXPT CS1
2450  003466  012737  000100  001026          MOV     #100,WCS2       ;SET EXPT CS2
2451  003474  012737  010600  001030          MOV     #10600,WDS      ;SET EXPT DS
2452  003502  012737  000000  001032          MOV     #0,WER          ;SET EXPT ER
2453  003510  012737  015454  000622          MOV     #MSLT12,EMADDR  ;SET HEADER
2454  003516  012737  001720  000776          MOV     #1720,UDES      ;SELECT NRZ, CORE DUMP, ODD
2455  003524  005037  001002                  CLR     PATRN           ;SELECT PATTERN 0
2456  003530  012737  003544  000712          MOV     #LT12A,SCOLP    ;SET SCOPE ADDRESS
2457  003536  012737  001056  001010          MOV     #WCDP0,RCDP     ;POINT TO PATTERN 0
2458  003544  004737  005342          LT12A:  JSR     PC,WAM3         ;GO DO WAM3
2459  003550  022737  000002  001002          CMP     #2,PATRN        ;SEE IF DONE
2460  003556  001407                          BEQ     LT12X           ;IF SO: BR
2461  003560  012737  000002  001002          MOV     #2,PATRN        ;SELECT PATTERN 2
2462  003566  012737  001044  001010          MOV     #WCDP2,RCDP     ;POINT TO PATTERN 2
2463  003574  000763                          BR      LT12A           ;CONTINUE
2464  003576  004737  012136          LT12X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2465  003602  000137  002236                  JMP     TSCD2           ;RETURN TO SCHEDULE
```

```
2467
2468                                            ;LOGIC TEST 13: EVEN PARITY WRITE: WAM 1(M8933)*****
2469
2470  003606  012737  004260  001024  LT13:     MOV     #4260,WCS1      ;SET EXPT CS1
2471  003614  012737  000100  001026           MOV     #100,WCS2       ;SET EXPT CS2
2472  003622  012737  010600  001030           MOV     #10600,WDS      ;SET EXPT DS
2473  003630  012737  000000  001032           MOV     #0,WER          ;SET EXPT ER
2474  003636  012737  015524  000622           MOV     #MSLT13,EMADDR  ;SET HEADER
2475  003644  012737  001710  000776           MOV     #1710,UDES      ;SET NRZ, NORMAL, EVEN
2476  003652  000137  003046                   JMP     LT5A            ;GO EXECUTE WAM 1
2477
2478                                            ;LOGIC TEST 14: EVEN PARITY READ: WAM 0(M8933 M8934)*****
2479
2480  003656  012737  144260  001024  LT14:     MOV     #144260,WCS1    ;SET EXPT CS1
2481  003664  012737  000100  001026           MOV     #100,WCS2       ;SET EXPT CS2
2482  003672  012737  150600  001030           MOV     #150600,WDS     ;SET EXPT DS
2483  003700  012737  000200  001032           MOV     #200,WER        ;SET EXPT ER
2484  003706  012737  015605  000622           MOV     #MSLT14,EMADDR  ;SET HEADER
2485  003714  012737  001710  000776           MOV     #1710,UDES      ;SET NRZ, NORMAL, EVEN
2486  003722  000137  003232                   JMP     LT7A            ;GO DO WAM 0
2487
2488                                            ;LOGIC TEST 15: READ REVERSE: WAM 3(M8906)*****
2489
2490  003726  012737  004276  001024  LT15:     MOV     #4276,WCS1      ;SET EXPT CS1
2491  003734  012737  000100  001026           MOV     #100,WCS2       ;SET EXPT CS2
2492  003742  012737  010640  001030           MOV     #10640,WDS      ;SET EXPT DS
2493  003750  012737  000000  001032           MOV     #0,WER          ;SET EXPT ER
2494  003756  012737  015664  000622           MOV     #MSLT15,EMADDR  ;SET HEADER
2495  003764  012737  002300  000776           MOV     #2300,UDES      ;SELECT PE,NORMAL,ODD
2496  003772  000240                           NOP
2497  003774  000240                           NOP
2498  003776  012737  000001  001006           MOV     #1,RDRVF        ;SET READ REVERSE FLAG
2499  004004  000137  002502                   JMP     LT1A            ;GO DO WAM 3, REVERSE
2500
```

```
2502
2503                                        ;LOGIC TEST 16: CRC CORRECTION-SINGLE TRACK,EVERY FRAME
2504                                        ;THIS IS A TEST OF THE CRC CORRECTION LOGIC. THE TEST WRITES
2505                                        ;A KNOWN PATTERN (ALL 1'S , ALL 0'S & 125252) WITH A DATA BIT
2506                                        ;ALTERED IN EACH OF THE DATA TRACKS. THIS TEST INSURES THAT A
2507                                        ;CRC CORRECTABLE ERROR IS CORRECTED.
2508                                        ;THE TEST PROCEEDS AS FOLLOWS:
2509
2510                                ;       STEP A  WRITE A KNOWN PATTERN USING WRAP AROUND MODE 0
2511
2512                                ;       STEP B  REWRITE THE PATTERN ABOVE WITH DATA BIT(S) MODIFIED
2513                                ;               IN TRACKS SPECIFIED BY CRCPAT USING WRAP AROUND MODE 4
2514                                ;               THIS WILL GENERATE A CRC ERROR
2515
2516                                ;       STEP C  EXECUTE A READ REVERSE USING WRAP AROUND MODE 3
2517
2518                                ;       STEP D  REPEAT STEP B ABOVE. UPON COMPLETION THE DATA READ
2519                                ;               BACK WILL MATCH THE DATA WRITTEN IN STEP A.
2520
2521
2522  004010  005037  001002        LT16:   CLR     PATRN           ;SELECT PATTERN # 0 (ALL 1'S)
2523  004014  012737  000401  004366        MOV     #401,CRCPAT     ;SELECT BITS TO BE ALTERED (TRACK 1)
2524  004022  012737  015731  000622        MOV     #MSLT16,EMADDR  ;SET ERROR MESSAGE HEADER
2525  004030  012737  004036  000712        MOV     #LT16A,SCOLP    ;SET SCOPE LOOP
2526
2527  004036  012737  144260  001024  LT16A: MOV     #144260,WCS1    ;SET EXPECTED VALUES UPON COMPLETION
2528  004044  012737  000100  001026        MOV     #100,WCS2
2529  004052  012737  150600  001030        MOV     #150600,WDS
2530  004060  012737  000200  001032        MOV     #200,WER
2531  004066  012737  001700  000776        MOV     #1700,UDES      ;SET UNIT DESCRIPTION-NRZ,800BPI,ODD
2532                                                                ;PARITY & PDP11 NORMAL MODE
2533  004074  005037  001022                CLR     CRCFLG          ;CLEAR CRC CORRECTION FLAG
2534  004100  004737  005222                JSR     PC,WAM0         ;DO A WRAP 0 --- STEP A
2535  004104  012737  000001  001022        MOV     #1,CRCFLG       ;SET CRC ERROR CORRECTION IN PROGRESS
2536  004112  013737  017474  004370        MOV     WBUFF,CRCDAT    ;GET DATA WRITTEN BY WRAP 0
2537                                ;XOR CRCPAT WITH CRCDAT
2538  004120  013737  004370  004372        MOV     CRCDAT,XORDAT   ;GET DATA TO BE MODIFIED
2539  004126  013737  004366  004374        MOV     CRCPAT,XORPAT   ;GET MODIFIER
2540  004134  043737  004366  004372        BIC     CRCPAT,XORDAT   ;CLEAR SET BITS IN DATA TO BE MODIFIED
2541  004142  043737  004370  004374        BIC     CRCDAT,XORPAT   ;CLEAR SETTING BITS
2542  004150  053737  004374  004372        BIS     XORPAT,XORDAT   ;SET CLEAR BITS IN DATA TO BE MODIFIED
2543  004156  013737  004372  004370        MOV     XORDAT,CRCDAT   ;RESTORE MODIFIED DATA
2544
2545  004164  012737  144260  001024        MOV     #144260,WCS1    ;SET EXPECTED VALUES UPON COMPLETION
2546  004172  012737  000110  001026        MOV     #110,WCS2       ;OF WRAP 4
2547  004200  012737  150600  001030        MOV     #150600,WDS
2548  004206  012737  100300  001032        MOV     #100300,WER
2549  004214  004737  005450                JSR     PC,WAM4         ;DO A WRAP 4 --- STEP B
2550  004220  000240                        NOP
2551  004222  012737  000001  001006        MOV     #1,RDRVF        ;SET TO READ REVERSE
2552  004230  012737  144276  001024        MOV     #144276,WCS1    ;SET EXPECTED VALUES UPON COMPLETION
2553  004236  012737  000110  001026        MOV     #110,WCS2
2554  004244  012737  150600  001030        MOV     #150600,WDS
2555  004252  012737  001000  001032        MOV     #1000,WER
2556  004260  004737  005342                JSR     PC,WAM3         ;DO A WRAP 3 --- STEP C
2557  004264  000240                        NOP
```

```
2558  004266  012737  004260  001024          MOV     #4260,WCS1
2559  004274  012737  000110  001026          MOV     #110,WCS2
2560  004302  012737  010600  001030          MOV     #10600,WDS
2561  004310  012737  000000  001032          MOV     #0,WER
2562  004316  005037  001022                  CLR     CRCFLG          ;CLEAR CRC CORRECTION IN PROGRESS FLAG
2563  004322  004737  005450                  JSR     PC,WAM4         ;GO TO WRAP 4 --- STEP D
2564
2565  004326  006337  004366                  ASL     CRCPAT          ;SELECT NEXT TRACK TO BE ALTERED
2566  004332  103241                          BCC     LT16A           ;CONTINUE FOR ALL TRACKS
2567  004334  012737  000401  004366          MOV     #401,CRCPAT     ;RESET BITS TO TRACK 1
2568  004342  005237  001002                  INC     PATRN           ;SELECT NEXT PATTERN
2569  004346  022737  000003  001002          CMP     #3,PATRN        ;BRANCH IF NOT DONE
2570  004354  001230                          BNE     LT16A
2571  004356  004737  012136                  JSR     PC,ITER         ;ITERATION LOOP
2572  004362  000137  002236                  JMP     TSCD2           ;RETURN TO SCHEDULER
2573
2574
2575  004366  000000          CRCPAT: .WORD   0               ;CONTAINS BITS TO BE ALTERED
2576  004370  000000          CRCDAT: .WORD   0               ;CONTAINS DATA TO BE WRITTEN BY WRAP4
2577  004372  000000          XORDAT: .WORD   0               ;TEMPRARY STORAGE FOR XOR
2578  004374  000000          XORPAT: .WORD   0               ;TEMPOARY STORAGE FOR XOR
2579
```

```
2581
2582
2583                                    ;LOGIC TEST 17:CRC CORRECTION - MULTIPLE BAD TRACKS
2584                                    ;THIS TEST CHECKS THAT CRC ERROR CORRECTION IS NOT PERFORMED WHEN
2585                                    ;MULTIPLE BAD TRACKS ARE DETECTED.
2586
2587  004376  012737  000002  001002  LT17:   MOV     #2,PATRN        ;SELECT PATTERN #2 (125125)
2588  004404  012737  001001  004366          MOV     #1001,CRCPAT    ;SELECT 2 BAD TRACKS
2589  004412  012737  016021  000622          MOV     #MSLT17,EMADDR  ;SET ERROR MESSAGE HEADER
2590  004420  012737  004426  000712          MOV     #LT17A,SCOLP    ;SET SCOPE LOOP ADDRESS
2591  004426  012737  144260  001024  LT17A:  MOV     #144260,WCS1    ;SET EXPECTED VALUES ON COMPLETION
2592  004434  012737  000100  001026          MOV     #100,WCS2       ;OF WRAP 0 BELOW
2593  004442  012737  150600  001030          MOV     #150600,WDS
2594  004450  012737  000200  001032          MOV     #200,WER
2595  004456  012737  001700  000776          MOV     #1700,UDES      ;SET UNIT DESCRIPTION
2596  004464  005037  001022                  CLR     CRCFLG          ;SET CRC CORRECTION NOT IN PROGRESS
2597  004470  004737  005222                  JSR     PC,WAM0         ;DO A WRAP 0 --- STEP A
2598  004474  000240                          NOP
2599  004476  012737  000002  001022          MOV     #2,CRCFLG       ;SET CRC CORRECTION IN PROGRESS
2600  004504  013737  017474  004370          MOV     WBUFF,CRCDAT    ;GET DATA TO BE WRITTEN
2601  004512  043737  004366  004370          BIC     CRCPAT,CRCDAT   ;MODIFY DATA TO BE WRITTEN
2602  004520  012737  144260  001024          MOV     #144260,WCS1    ;SET EXPECTED VALUES ON COMPLETION
2603  004526  012737  000110  001026          MOV     #110,WCS2       ;OF WRAP 4 BELOW
2604  004534  012737  150600  001030          MOV     #150600,WDS
2605  004542  012737  100300  001032          MOV     #100300,WER
2606  004550  004737  005450                  JSR     PC,WAM4         ;DO A WRAP 4 --- STEP B
2607  004554  000240                          NOP
2608  004556  012737  000001  001006          MOV     #1,RDRVF        ;SET READ REVERSE FLAG
2609  004564  012737  144276  001024          MOV     #144276,WCS1    ;SET EXPECTED VALUES ON COMPLETION
2610  004572  012737  000110  001026          MOV     #110,WCS2       ;OF WRAP 3 BELOW
2611  004600  012737  150600  001030          MOV     #150600,WDS
2612  004606  012737  001000  001032          MOV     #1000,WER
2613  004614  004737  005342                  JSR     PC,WAM3         ;DO A WRAP 3 --- STEP C
2614  004620  000240                          NOP
2615  004622  012737  144260  001024          MOV     #144260,WCS1    ;SET EXPECTED VALUES ON COMPLETION
2616  004630  012737  000110  001026          MOV     #110,WCS2       ;OF WRAP 4 BELOW
2617  004636  012737  150600  001030          MOV     #150600,WDS
2618  004644  012737  100100  001032          MOV     #100100,WER
2619  004652  005037  001022                  CLR     CRCFLG          ;CLEAR CRC IN PROGRESS FLAG
2620  004656  013701  004370                  MOV     CRCDAT,R1       ;GET DATA THAT WAS WRITTEN IN STEP B
2621  004662  012703  017474                  MOV     #WBUFF,R3       ;SET START OF WRITE BUFFER
2622  004666  004737  005166                  JSR     PC,DATIA        ;GO SET WRITE BUFFER
2623  004672  004737  005450                  JSR     PC,WAM4         ;GO DO A WRAP 4 --- STEP D
2624  004676  000240                          NOP
2625  004700  004737  012136                  JSR     PC,ITER         ;ITERATE TEST
2626  004704  000137  002236                  JMP     TSCD2           ;RETURN TO SCHEDULER
```

```
2628
2629                                    ;LOGIC TEST 20: READ REVERSE,NRZ,WRAP 3
2630                                    ;THIS TEST TESTS THAT A CRC ERROR OCCURS AFTER A READ REVERSE USING
2631                                    ;WRAP AROUND MODE 3 IN NRZ MODE
2632
2633  004710  005037  001002            LT20:   CLR     PATRN           ;SET PATTERN # 0 (ALL 1'S)
2634  004714  012737  144276  001024            MOV     #144276,WCS1    ;SET EXPECTED VALUES ON COMPLETION
2635  004722  012737  000100  001026            MOV     #100,WCS2
2636  004730  012737  150600  001030            MOV     #150600,WDS
2637  004736  012737  100000  001032            MOV     #100000,WER
2638  004744  012737  016105  000622            MOV     #MSLT20,EMADDR  ;SET ERROR MESSAGE ADDRESS
2639  004752  012737  001700  000776            MOV     #1700,UDES      ;SET UNIT DESCRIPTION
2640  004760  012737  004766  000712            MOV     #LT20A,SCOLP    ;SET SCOPE LOOP ADDRESS
2641  004766  012737  000001  001006  LT20A:    MOV     #1,RDRVF        ;SET READ REVERSE FLAG
2642  004774  012737  000001  001020            MOV     #1,DCHKFL       ;SET DATA CHECK FLAG TO NOT CHACK DATA
2643  005002  004737  005342                    JSR     PC,WAM3
2644  005006  005037  001006                    CLR     RDRVF           ;CLEAR READ REVERSE FLAG
2645  005012  005037  001020                    CLR     DCHKFL          ;CLEAR DATA CHECK FLAG
2646  005016  004737  012136                    JSR     PC,ITER
2647  005022  000137  002236                    JMP     TSCD2           ;RETURN TO SCHEDULER
2648
```

```
2650
2651
2652                                    ;*************************************************
2653                                    ;DATA SETUP ROUTINE:
2654                                    ;
2655                                    ;THIS ROUTINE IS USED TO GENERATE THE DATA PATTERNS
2656                                    ;THE PATTERN TO BE WRITTEN IS SPECIFIED BY:
2657                                    ;PATRN  =0      ALL 1'S
2658                                    ;       =1      ALL 0'S
2659                                    ;       =2      ALTERNATE 1'S & 0'S (125125)
2660                                    ;       =3      ALTERNATING PARITY (177377)
2661                                    ;*************************************************
2662
2663  005026  005737  000740           DSUP:   TST     STFLG           ;SEE IF SINGLE TEST
2664  005032  001434                           BEQ     DS0             ;IF NOT: BR
2665  005034  032777  000100  173526           BIT     #100,@SWR       ;SEE IF SELECT PATTERN
2666  005042  001430                           BEQ     DS0             ;IF NOT: BR
2667  005044  012704  016614                   MOV     #WMSG3,R4
2668  005050  004737  013054                   JSR     PC,TTOUT        ;REQUEST PATTERN NUMBER
2669  005054  013703  001002                   MOV     PATRN,R3
2670  005060  004737  013202                   JSR     PC,OCTP         ;PRINT PATTERN NUMBER
2671  005064  012705  001002                   MOV     #PATRN,R5       ;GET ADDRESS OF PATRN ENTRY
2672  005070  012701  000002                   MOV     #2,R1           ;SET SIZE OF ENTRY
2673  005074  012702  000003                   MOV     #3,R2           ;SET UPPER LIMIT
2674  005100  012703  000000                   MOV     #0,R3           ;SET LOWER LIMIT
2675  005104  004737  012514                   JSR     PC,TTR          ;GO GET PATTERN NUMBER
2676  005110  112737  000001  001001           MOVB    #1,WPGFL+1      ;SET FLAG
2677  005116  113737  001002  001000           MOVB    PATRN,WPGFL     ;SET PATTERN NUMBER
2678  005124  012703  017474           DS0:    MOV     #WBUFF,R3       ;R3 = ADDRS OF WRITE BUFFER
2679  005130  013701  001002                   MOV     PATRN,R1        ;R1 = PATTERN SELECTOR
2680  005134  006301                           ASL     R1              ;MAKE PATTERN SELECTOR EVEN
2681  005136  004771  001034                   JSR     PC,@DATBL(R1)   ;GO GENERATE PATTERN
2682  005142  012702  000202           DS3:    MOV     #202,R2         ;R2=BUFFER SIZE +2
2683  005146  012701  020106                   MOV     #RBUFF,R1       ;R1=READ DATA START
2684  005152  005021                   DS4:    CLR     (R1)+           ;CLEAR BUFFER
2685  005154  005302                           DEC     R2              ;SEE IF DONE ALL
2686  005156  001375                           BNE     DS4             ;IF NOT: BR
2687  005160  000207                           RTS     PC              ;EXIT
2688
2689                                    ;ALL ONES**********
2690
2691  005162  012701  177777           DAT1:   MOV     #-1,R1          ;R1=DATA
2692  005166  012702  000202           DAT1A:  MOV     #202,R2         ;R2=WORD COUNT +2
2693  005172  010123                   1$:     MOV     R1,(R3)+        ;LOAD BUFFER
2694  005174  005302                           DEC     R2              ;SEE IF DONE
2695  005176  001375                           BNE     1$              ;IF NOT: BR
2696  005200  000207                           RTS     PC              ;RETURN TO CALLER
2697
2698                                    ;ALL ZEROS**********
2699
2700  005202  005001                   DAT2:   CLR     R1              ;R1=DATA
2701  005204  000770                           BR      DAT1A           ;LOAD BUFFER
2702
2703                                    ;ONE/ZERO IN ALTERNATING CHARACTERS**********
2704
2705  005206  012701  125125           DAT3:   MOV     #125125,R1      ;R1=DATA
```

```
2706  005212  000765                          BR      DAT1A           ;LOAD BUFFER
2707
2708                                          ;ALTERNATING PARITY CHARACTERS********
2709
2710  005214  012701  177377        DAT4:     MOV     #177377,R1      ;R1=ALTERNATING PARITY DATA
2711  005220  000762                          BR      DAT1A           ;GO LOAD BUFFER
2712
```

```
2714
2715                                                ;WRAP AROUND MODE 0 GLOBAL*****
2716
2717  005222  012737  000006  000750  WAM0:   MOV     #6,WAM          ;SET WAM NUMBER
2718  005230  012737  000060  000752  WAM01:  MOV     #60,FUN
2719  005236  005037  000754                  CLR     DATC
2720  005242  012737  017474  000760          MOV     #WBUFF,DATAD    ;SET BUFFER ADDRESS
2721  005250  012737  020106  000762          MOV     #RBUFF,RDAD     ;SET POINTER TO READ BUFFER
2722  005256  004737  005514                  JSR     PC,SETUP        ;GO SET UP
2723  005262  000137  006074                  JMP     EXEC
2724
2725                                                ;WRAP AROUND MODE 1 WRITE BUFFER**********
2726
2727  005266  012737  000010  000750  WAM1:   MOV     #10,WAM
2728  005274  000755                  BR      WAM01
2729
2730                                                ;WRAP AROUND MODE 2 BIT FIDDLER WRITE**********
2731
2732  005276  012737  000012  000750  WAM2:   MOV     #12,WAM
2733  005304  012737  000060  000752          MOV     #60,FUN
2734  005312  005037  000754                  CLR     DATC
2735  005316  012737  017474  000760          MOV     #WBUFF,DATAD
2736  005324  012737  020106  000762          MOV     #RBUFF,RDAD
2737  005332  004737  005514          WAM2A:  JSR     PC,SETUP
2738  005336  000137  006074                  JMP     EXEC
2739
2740                                                ;WRAP AROUND MODE 3 BIT FIDDLER READ**********
2741
2742  005342  012737  000014  000750  WAM3:   MOV     #14,WAM         ;SET WAM NUMBER
2743  005350  012737  000070  000752          MOV     #70,FUN         ;SET FUNCTION
2744  005356  012737  020106  000760          MOV     #RBUFF,DATAD    ;SET BUFFER ADDRESS
2745  005364  012737  017474  000756          MOV     #WBUFF,WTAD     ;SET POINTER TO WRITE BUFFER
2746  005372  005737  001006                  TST     RDRVF
2747  005376  001411                  BEQ     WAM3A
2748  005400  062737  000376  000760          ADD     #376,DATAD
2749  005406  062737  000377  000756          ADD     #377,WTAD
2750  005414  012737  000076  000752          MOV     #76,FUN         ;SET READ REVERSE CODE
2751  005422  032737  000020  000776  WAM3A:  BIT     #20,UDES
2752  005430  001403                  BEQ     WAM3B
2753  005432  013737  001010  000756          MOV     RCDP,WTAD
2754  005440  004737  005514          WAM3B:  JSR     PC,SETUP        ;GO SET UP
2755  005444  000137  006074                  JMP     EXEC            ;GO EXECUTE
2756
2757                                                ;WRAP AROUND MODE 4 CRC CORRECTION
2758                                                ;CALL:  JSR     PC,WAM4
2759
2760  005450  012737  000030  000750  WAM4:   MOV     #30,WAM         ;SET MAINTENANCE MODE FUNCTION   WRAP 4
2761  005456  012737  000060  000752          MOV     #60,FUN         ;SET TAPE FUNCTION = WRITE FWD
2762  005464  005037  000754                  CLR     DATC
2763  005470  012737  004370  000760          MOV     #CRCDAT,DATAD   ;SET ADRS OF WRITE BUFFER
2764  005476  012737  020106  000762          MOV     #RBUFF,RDAD     ;SET READ BUFFER ADDRESS
2765  005504  004737  005514                  JSR     PC,SETUP        ;GO SETUP REGISTERS
2766  005510  000137  006074                  JMP     EXEC            ;GO EXECUTE
```

```
2768                                            ;REGISTER SETUP ROUTINE*****
2769
2770  005514  005737  001022          SETUP:    TST     CRCFLG          ;DO NOT DO DATA SETUP NOR INIT
2771  005520  001004                           BNE     1$              ;IF CRC CORRECTION IS IN PROGRESS
2772  005522  022737  000030  000750           CMP     #30,WAM         ;DO NOT INIT IF DOING WAM 4 --- STEP D
2773  005530  001004                           BNE     2$
2774  005532  012777  000011  172750  1$:      MOV     #11,@C1         ;DO DRIVE CLEAR IF WAM4---STEP D
2775  005540  000412                           BR      SET1A
2776  005542  005737  000740          2$:      TST     STFLG           ;SEE IF SINGLE TEST
2777  005546  001403                           BEQ     SET0            ;IF NOT: BR
2778  005550  005737  001000                    TST     WPGFL           ;SEE IF HAVE SELECTED PATTERN
2779  005554  001002                           BNE     SET1            ;IF SO: BR
2780  005556  004737  005026          SET0:     JSR     PC,DSUP         ;GO DO DATA SETUP
2781  005562  004737  012206          SET1:     JSR     PC,INIT         ;INIT CONTROLLER,SELECT UNIT & DRIVE
2782                                            ;LOAD SLAVE DESC AND MOVE OFF BOT
2783  005566  012777  177400  172722  SET1A:   MOV     #-400,@FC       ;SET FC=WCX2
2784  005574  032737  000020  000776           BIT     #20,UDES        ;SEE IF CORE DUMP
2785  005602  001403                           BEQ     SET2            ;IF NOT: BR
2786  005604  012777  177000  172704           MOV     #-1000,@FC      ;SET FC=WCX4
2787  005612  012777  177600  172672  SET2:    MOV     #-200,@WC       ;SET WC
2788  005620  013777  000760  172666           MOV     DATAD,@BA       ;SET BUS ADDRESS
2789  005626  032777  010000  172664           BIT     #10000,@CS      ;ASSURE DRIVE THERE
2790  005634  001417                           BEQ     SP1             ;IF SO: BR
2791  005636  032777  020000  172724           BIT     #20000,@SWR     ;SEE IF PRINT ERRORS
2792  005644  001004                           BNE     SP01            ;IF NOT: BR
2793  005646  012704  016635                    MOV     #WMSG4,R4
2794  005652  004737  013054                    JSR     PC,TTOUT        ;PRINT NON-EXISTANT DRIVE
2795  005656  032777  100000  172704  SP01:    BIT     #100000,@SWR    ;SEE IF HALT ON ERROR
2796  005664  001401                           BEQ     SP0             ;IF NOT: BR
2797  005666  000000                           HALT
2798  005670  000137  005562          SP0:      JMP     SET1            ;RESETUP
2799  005674  022737  000014  000750  SP1:      CMP     #14,WAM         ;SEE IF WAM 3
2800  005702  001031                           BNE     SP1B            ;IF NOT: BR
2801  005704  117737  173046  000754           MOVB    @WTAD,DATC      ;GET FIRST CHAR
2802  005712  042737  177400  000754           BIC     #177400,DATC
2803  005720  000337  000754                    SWAB    DATC
2804  005724  052737  000200  000754           BIS     #200,DATC       ;SET PARITY
2805  005732  005737  001006                    TST     RDRVF           ;SEE IF READ REVERSE
2806  005736  001403                           BEQ     SP1A            ;IF NOT: BR
2807  005740  005337  000756                    DEC     WTAD            ;DECREMENT POINTER
2808  005744  000410                           BR      SP1B
2809  005746  005237  000756          SP1A:     INC     WTAD            ;BUMP POINTER
2810  005752  032737  000020  000776           BIT     #20,UDES        ;SEE IF CORE DUMP
2811  005760  001402                           BEQ     SP1B            ;IF NOT: BR
2812  005762  005237  000756                    INC     WTAD            ;BUMP POINTER AGAIN
2813  005766  053777  000776  172546  SP1B:     BIS     UDES,@TC        ;SET UNIT DESCRIPTION (DEN,PAR,FMT)
2814  005774  052777  000001  172532           BIS     #1,@MR          ;SET MAINT MODE
2815  006002  053777  000750  172524           BIS     WAM,@MR         ;SET WAM
2816  006010  053777  000754  172516           BIS     DATC,@MR        ;SET DATA
2817  006016  013777  000752  172464           MOV     FUN,@C1         ;SET FUNCTION
2818  006024  032777  040000  172470           BIT     #40000,@DS      ;ASSURE NO ERROR
2819  006032  001002                           BNE     SP3             ;IF ERROR: BR
2820  006034  000240                           NOP
2821  006036  000207                           RTS     PC              ;RETURN
2822  006040  032777  020000  172522  SP3:      BIT     #20000,@SWR     ;SEE IF PRINT ERRORS
2823  006046  001004                           BNE     SP4             ;IF NOT: BR
```

```
2824  006050  012704  016576                    MOV     #WMSG2,R4
2825  006054  004737  013054                    JSR     PC,TTOUT        ;PRINT SETUP ERROR
2826  006060  032777  100000  172502  SP4:      BIT     #100000,@SWR    ;SEE IF HAL' ON ERROR
2827  006066  001401                           BEQ     SP5             ;IF NOT: BR
2828  006070  000000                           HALT
2829  006072  000207                  SP5:     RTS     PC              ;RETURN
```

```
2831                                            ;EXECUTE WAM ROUTINE*****
2832
2833  006074  000240                   EXEC:    NOP
2834  006076  000240                            NOP
2835  006100  032777  000040  172426            BIT     #40,@MR
2836  006106  001403                            BEQ     EX0             ;ASSURE MAINT CLOCK IS ZERO
2837  006110  042777  000040  172416            BIC     #40,@MR         ;IF NOT: CLEAR IT
2838  006116  022737  000010  000750  EX0:      CMP     #10,WAM         ;SEE IF WAM 1 OR 2 OR 3
2839  006124  003402                            BLE     2$
2840  006126  000137  006540           1$:      JMP     EXW2            ;GO DO WAM 0
2841  006132  022737  000030  000750  2$:       CMP     #30,WAM         ;BRANCH IF WRAP AROUND MODE 4
2842  006140  001772                            BEQ     1$
2843  006142  052777  000001  172340  EX1:      BIS     #1,@C1          ;SET GO BIT
2844  006150  005000                            CLR     R0
2845  006152  012701  000002                   MOV     #2,R1           ;SET DELAY
2846  006156  032777  100000  172356  EX1A:     BIT     #100000,@TC     ;SEE IF ALPHA
2847  006164  001404                            BEQ     EX2             ;IF SO: BR
2848  006166  005300                            DEC     R0
2849  006170  001372                            BNE     EX1A            ;AWAIT ALPHA
2850  006172  005301                            DEC     R1
2851  006174  001370                            BNE     EX1A
2852  006176  005077  172364           EX2:     CLR     @PSW
2853  006202  012701  000400                   MOV     #400,R1         ;SET NUMBER OF CLKS
2854  006206  032737  000020  000776            BIT     #20,UDES        ;SEE IF CORE DUMP
2855  006214  001402                            BEQ     EX3             ;IF NOT: BR
2856  006216  012701  001000                   MOV     #1000,R1        ;SET CLOCKS LWCX4
2857  006222  022737  000014  000750  EX3:      CMP     #14,WAM         ;SEE IF WAM 3
2858  006230  001010                            BNE     1$              ;IF NOT: BR
2859  006232  032737  002000  000776            BIT     #2000,UDES      ;IS IT PE?
2860  006240  001016                            BNE     EX5A            ;IF YES: BR
2861  006242  052777  000040  172264            BIS     #40,@MR         ;CLOCK UP
2862  006250  000412                            BR      EX5A
2863  006252  032737  002000  000776  1$:       BIT     #2000,UDES      ;SEE IF PE
2864  006260  001404                            BEQ     EX5             ;IF NOT PE: BR
2865  006262  006301                            ASL     R1
2866  006264  062701  000246                   ADD     #246,R1         ;SET TO ALLOW FOR PRE/POSTAMBLE
2867  006270  000402                            BR      EX5A
2868  006272  062701  000010           EX5:     ADD     #10,R1          ;ADD CLOCKS FOR CRC AND LRC
2869  006276  022737  000014  000750  EX5A:     CMP     #14,WAM         ;SEE IF WAM 3
2870  006304  001053                            BNE     EX5C            ;IF NOT: BR
2871  006306  117700  172444                   MOVB    @WTAD,R0
2872  006312  042700  177400                   BIC     #177400,R0
2873  006316  005737  001006                   TST     RDRVF           ;SEE IF REVERSE
2874  006322  001403                            BEQ     EX5A1           ;IF NOT: BR
2875  006324  005337  000756                   DEC     WTAD            ;DEC POINTER
2876  006330  000416                            BR      EX5B
2877  006332  005237  000756           EX5A1:   INC     WTAD
2878  006336  032737  000020  000776            BIT     #20,UDES        ;SEE IF CORE DUMP
2879  006344  001410                            BEQ     EX5B            ;IF NOT: BR
2880  006346  005237  000756                   INC     WTAD            ;BUMP POINTER
2881  006352  005777  172400                   TST     @WTAD           ;SEE IF END
2882  006356  001003                            BNE     EX5B            ;IF NOT: BR
2883  006360  162737  000010  000756            SUB     #10,WTAD        ;RESTORE POINTER
2884  006366  052777  000040  172140  EX5B:     BIS     #40,@MR         ;CLOCK UP
2885  006374  017702  172134                   MOV     @MR,R2          ;READ MR
2886  006400  042702  177600                   BIC     #177600,R2      ;MASK OUT DATA
```

```
 2887  006404  000300                          SWAB    R0              ;POSITION DATA
 2888  006406  022700  177000                  CMP     #177000,R0      ;SEE IF PATTERN 3
 2889  006412  001404                          BEQ     2$              ;IF SO: BR
 2890  006414  000240                          NOP
 2891  006416  000240                          NOP
 2892  006420  052700  000200          1$:     BIS     #200,R0         ;SET ODD PARITY
 2893  006424  050002                  2$:     BIS     R0,R2           ;LOAD NEW DATA
 2894  006426  010277  172102                  MOV     R2,@MR          ;CLOCK DOWN AND LOAD NEW DATA
 2895  006432  000426                          BR      EX5D
 2896  006434  052777  000040  172072  EX5C:   BIS     #40,@MR         ;CLOCK UP
 2897  006442  042777  000040  172064          BIC     #40,@MR         ;CLOCK DOWN
 2898  006450  017700  172060                  MOV     @MR,R0          ;GET MR
 2899  006454  000300                          SWAB    R0
 2900  006456  032737  000010  000776          BIT     #10,UDES        ;SEE IF EVEN PAR
 2901  006464  001405                          BEQ     EX5C0           ;IF NOT: BR
 2902  006466  010077  172270                  MOV     R0,@RDAD
 2903  006472  005237  000762                  INC     RDAD
 2904  006476  000402                          BR      EX5C1
 2905  006500  110077  172256          EX5C0:  MOVB    R0,@RDAD        ;PUT CHAR IN CORE
 2906  006504  005237  000762          EX5C1:  INC     RDAD
 2907  006510  000240                  EX5D:   NOP
 2908  006512  005301                          DEC     R1              ;SEE IF DONE CLKS
 2909  006514  001270                          BNE     EX5A            ;IF NOT: BR
 2910  006516  032777  000040  172010          BIT     #40,@MR         ;CLOCK UP?
 2911  006524  001403                          BEQ     1$              ;IF NOT: BR
 2912  006526  042777  000040  172000          BIC     #40,@MR         ;CLOCK DOWN
 2913  006534  000137  011712          1$:     JMP     EORP            ;GO DO EOR
```

```
                                        ;EXECUTE WAM 0******

000540  000240                  EXW2:   NOP
000542  012737  006726  000670          MOV     #EXW2H,RTRN     ;SET INTERRUPT RETURN ADDRESS
000550  012701  000200                  MOV     #200,R1         ;SET NUMBER OF CLOCKS = FC/2
000554  032737  002000  000776          BIT     #2000,UDES      ;SEE IF PE
000562  001402                          BEQ     EXW2A           ;IF NOT: BR
000564  012701  000100                  MOV     #100,R1         ;ELSE SET CLKS = FC/4
000570  012702  020106          EXW2A:  MOV     #RBUFF,R2               ;SET BUFFER ADDRESS
000574  022737  000030  000750          CMP     #30,WAM         ;BRANCH IF NOT WRAP 4
000602  001003                          BNE     1$
000604  052777  000010  171706          BIS     #10,@CS         ;SET INHIBIT BUS ADDRESS INCREMENT
000612  012777  000161  171670  1$:     MOV     #161,@C1        ;SET WRITE COMMAND  AND GO
000620  005077  171742                  CLR     @PSW            ;ALLOW INTERRUPT
000624  032777  000040  171702  EXW2B:  BIT     #40,@MR
000632  001774                          BEQ     EXW2B           ;AWAIT CLOCK UP
000634  017722  171674                  MOV     @MR,(R2)+       ;GET DATA
000640  032777  000040  171666  EXW2C:  BIT     #40,@MR
000646  001374                          BNE     EXW2C           ;AWAIT CLOCK DOWN
000650  017722  171660                  MOV     @MR,(R2)+       ;GET DATA
000654  005301                          DEC     R1              ;SEE IF DONE ALL
000656  001362                          BNE     EXW2B           ;IF NOT: BR
000660  012701  000003          EXW2E:  MOV     #3,R1
000664  005000                          CLR     R0              ;SET DELAY
000666  005300                  EXW2F:  DEC     R0
000670  001376                          BNE     EXW2F
000672  005301                          DEC     R1
000674  001374                          BNE     EXW2F           ;DELAY
000676  032777  020000  171664          BIT     #20000,@SWR     ;SEE IF ERROR PRINT
000704  001004                          BNE     EXW2G           ;IF NOT: BR
000706  012704  017040                  MOV     #WMSG24,R4
000712  004737  013054                  JSR     PC,TTOUT        ;PRINT NO INTERUPT
000716  005777  171646          EXW2G:  TST     @SWR            ;SEE IF HALT ON ERROR
000722  100001                          BPL     EXW2H           ;IF NOT: BR
000724  000000                          HALT
000726  000240                  EXW2H:  NOP
000730  012701  020106                  MOV     #RBUFF,R1       ;GET START OF READ BUFFER
000734  012700  000400                  MOV     #400,R0         ;SET SIZE
000740  010102                          MOV     R1,R2
000742  012203                  EXW2J:  MOV     (R2)+,R3
000744  000303                          SWAB    R3
000746  032737  000010  000776          BIT     #10,UDES        ;SEE IF EVEN PAR
000754  001402                          BEQ     EXW2J0          ;IF NOT: BR
000756  010321                          MOV     R3,(R1)+        ;SAVE PAR + DATA
000760  000401                          BR      EXW2J1
000762  110321                  EXW2J0: MOVB    R3,(R1)+        ;ASSEMBLE DATA IN BYTES
000764  005300                  EXW2J1: DEC     R0
000766  001365                          BNE     EXW2J           ;CONTINUE FOR ALL
000770  032777  000200  171572          BIT     #200,@SWR       ;SEE IF STATUS CHECK
000776  001002                          BNE     EXW2K           ;IF NOT: BR
001000  004737  007026                  JSR     PC,WSTCK        ;ELSE GO CHECK STATUS
001004  000240                  EXW2K:  NOP
001006  032777  000400  171554          BIT     #400,@SWR       ;SEE IF DATA CHECK
001014  001002                          BNE     EXW2X           ;IF NOT: BR
001016  004737  007416                  JSR     PC,DCHK         ;ELSE GO CHECK DATA
001022  000240                  EXW2X:  NOP
```

```
 2971  007024  000207                        RTS     PC              ;EXIT
```

```
2973
2974                                        ;WRAP AROUND STATUS CHECK ROUTINE*********
2975
2976  007026  000240                WSTCK:  NOP
2977  007030  005037  000772                CLR     SERFL           ;CLEAR ERROR FLAG
2978  007034  005037  000620                CLR     HDRFL           ;CLEAR HEADER FLAG
2979  007040  022737  014774  000622        CMP     #MSLT2,EMADDR   ;SEE IF TEST 2
2980  007046  001404                        BEQ     2$              ;IF SO: BR
2981  007050  022737  015664  000622        CMP     #MSLT15,EMADDR  ;SEE IF TEST 15
2982  007056  001025                        BNE     5$              ;IF NOT: BR
2983  007060  005737  001000          2$:   TST     WPGFL           ;SEE IF SINGLE PATTERN
2984  007064  001405                        BEQ     3$              ;IF NOT: BR
2985  007066  122737  000003  001000        CMPB    #3,WPGFL        ;SEE IF PATTERN 3
2986  007074  001016                        BNE     5$              ;IF NOT: BR
2987  007076  000404                        BR      4$              ;ELSE GO DO EXPT CHANGE
2988  007100  022737  000003  001002  3$:   CMP     #3,PATRN        ;SEE IF PATTERN 3
2989  007106  001011                        BNE     5$              ;IF NOT: BR
2990  007110  052737  140000  001024  4$:   BIS     #140000,WCS1
2991  007116  052737  140000  001030        BIS     #140000,WDS
2992  007124  052737  000100  001032        BIS     #100,WER        ;SET EXPT PARITY ERROR
2993  007132  012737  016662  000672  5$:   MOV     #WMSG6,ERADD    ;SET CODE=CS1
2994  007140  017702  171344                MOV     @C1,R2          ;GET RCVD CS1
2995  007144  013705  001024                MOV     WCS1,R5         ;GET EXPT CS1
2996  007150  004737  007272                JSR     PC,WSTG         ;GO CHK
2997  007154  012737  016707  000672        MOV     #WMSG6D,ERADD   ;SET CODE=CS2
2998  007162  017702  171332                MOV     @CS,R2          ;SET RCVD CS2
2999  007166  013705  001026                MOV     WCS2,R5         ;GET EXPT CS2
3000  007172  053705  000624                BIS     DRVN,R5         ;SET DRIVE NUMBER IN EXPT CS2
3001  007176  004737  007272                JSR     PC,WSTG         ;GO CHK
3002  007202  012737  016715  000672        MOV     #WMSG6E,ERADD   ;SET CODE=DS
3003  007210  017702  171306                MOV     @DS,R2          ;SET RCVD DS
3004  007214  013705  001030                MOV     WDS,R5          ;GET EXPT DS
3005  007220  004737  007272          1$:   JSR     PC,WSTG         ;GO CHK
3006  007224  012737  016722  000672        MOV     #WMSG6F,ERADD   ;SET CODE=ER
3007  007232  017702  171266                MOV     @ER,R2          ;GET RCVD ER
3008  007236  000240                        NOP
3009  007240  000240                        NOP
3010  007242  013705  001032                MOV     WER,R5          ;GET EXPT ER
3011  007246  004737  007272                JSR     PC,WSTG         ;GO CHK
3012  007252  005737  000772                TST     SERFL           ;SEE IF ANY ERRORS
3013  007256  001456                        BEQ     WSTX            ;IF NOT: BR
3014  007260  005777  171304                TST     @SWR            ;SEE IF HALT ON ERROR
3015  007264  100053                        BPL     WSTX            ;IF NOT: BR
3016  007266  000000                        HALT
3017  007270  000451                        BR      WSTX            ;CONTINUE
3018  007272  000240                WSTG:   NOP
3019  007274  020205                        CMP     R2,R5           ;SEE IF EXPT=RCVD
3020  007276  001446                        BEQ     WSTX            ;IF SO: BR
3021  007300  005237  000772                INC     SERFL           ;SET ERROR FLAG
3022  007304  032777  020000  171256        BIT     #20000,@SWR     ;SEE IF PRINT ERRORS
3023  007312  001040                        BNE     WSTX            ;IF NOT: BR
3024  007314  005737  000620                TST     HDRFL           ;SEE IF DONE HEADER
3025  007320  001010                        BNE     WSTG0           ;IF SO: BR
3026  007322  013704  000622                MOV     EMADDR,R4
3027  007326  004737  013054                JSR     PC,TTOUT        ;PRINT TEST HEADER
3028  007332  012704  017024                MOV     #WMSG23,R4
```

```
3029  007336  004737  013054                  JSR     PC,TTOUT        ;PRINT STATUS TAG
3030  007342  012737  000001  000620  WSTGO:  MOV     #1,HDRFL        ;SET HEADER FLAG
3031  007350  013704  000672                  MOV     ERADD,R4
3032  007354  004737  013054                  JSR     PC,TTOUT        ;PRINT CODE
3033  007360  012704  014325                  MOV     #MSG12,R4
3034  007364  004737  013054                  JSR     PC,TTOUT        ;PRINT EXPT TAG
3035  007370  010503                          MOV     R5,R3
3036  007372  004737  013202                  JSR     PC,OCTP         ;PRINT EXPT STATUS
3037  007376  012704  014334                  MOV     #MSG13,R4
3038  007402  004737  013054                  JSR     PC,TTOUT        ;PRINT RCVD TAG
3039  007406  010203                          MOV     R2,R3
3040  007410  004737  013202                  JSR     PC,OCTP         ;PRINT RCVD STATUS
3041  007414  000207                  WSTX:   RTS     PC              ;RETURN
3042
```

```
3044
3045                                        ;DATA CHECK ROUTINE*****
3046
3047  007416  000240                DCHK:   NOP
3048  007420  005737  001022                TST     CRCFLG          ;DO NOT DO A DATA CHACK
3049  007424  001402                        BEQ     2$              ;IF CRC CORRECTION IN PROGRESS
3050  007426  000137  010256        1$:     JMP     DCHKX1
3051  007432  005737  001020        2$:     TST     DCHKFL          ;BRANCH IF DATA IS NOT TO BE CHECKED
3052  007436  001373                        BNE     1$
3053  007440  005037  000620                CLR     HDRFL           ;CLEAR HEADER FLAG
3054  007444  005037  000766                CLR     DERFL           ;CLEAR DATA ERROR FLAG
3055  007450  005037  000774                CLR     CRCNT           ;CLEAR CHAR CNTR
3056  007454  032737  000010  000776        BIT     #10,UDES        ;SEE IF EVEN PARITY
3057  007462  001402                        BEQ     DCHKA0          ;IF NOT: BR
3058  007464  000137  010300                JMP     DCHKE           ;ELSE GO CHECK EVEN
3059  007470  022737  000010  000750  DCHKA0: CMP    #10,WAM         ;SEE IF WAM 1
3060  007476  001006                        BNE     DCHKA           ;IF NOT: BR
3061  007500  032737  002000  000776        BIT     #2000,UDES      ;SEE IF PE
3062  007506  001402                        BEQ     DCHKA           ;IF NOT: BR
3063  007510  000137  011002                JMP     PRCHK           ;GO CHK DATA
3064  007514  012700  177400        DCHKA·  MOV     #-400,R0        ;SET NUMBER OF CHARACTERS
3065  007520  022737  000012  000750        CMP     #12,WAM
3066  007526  001006                        BNE     DCHKA1          ;IF NOT WRAP 2: BR
3067  007530  032737  000020  000776        BIT     #20,UDES
3068  007536  001402                        BEQ     DCHKA1          ;IF NOT CORE DUMP: BR
3069  007540  012700  177000                MOV     #-1000,R0
3070  007544  022737  000030  000750  DCHKA1: CMP    #30,WAM         ;BRANCH IF WRAP 4
3071  007552  001404                        BEQ     1$
3072  007554  022737  000006  000750        CMP     #6,WAM          ;SEE IF WRAP 0
3073  007562  001007                        BNE     DCHKA2          ;IF NOT: BR
3074  007564  012700  177744        1$:     MOV     #-34,R0         ;SET NUMBER OF CHARACTERS READ
3075  007570  012701  017504                MOV     #WBUFF+10,R1    ;SET POINTER
3076  007574  005037  000766                CLR     DERFL           ;CLEAR DATA ERROR FLAG
3077  007600  000431                        BR      DCHKB0
3078  007602  022737  000012  000750  DCHKA2: CMP    #12,WAM         ;SEE IF WRAP 2
3079  007610  001021                        BNE     DCHKB           ;IF NOT. BR
3080  007612  032737  002000  000776        BIT     #2000,UDES      ;SEE IF PE
3081  007620  001415                        BEQ     DCHKB           ;IF NOT: BR
3082  007622  012700  177653                MOV     #-125,R0        ;POINT TO START OF DATA
3083  007626  012737  000001  000764        MOV     #1,W2FLG        ;SET WRAP 2 FLAG
3084  007634  004737  007654                JSR     PC,DCHKB        ;GO CHECK DATA
3085  007640  004737  011302                JSR     PC,W1DCHK       ;GO CHECK WRAP 1 DATA
3086  007644  005037  000764                CLR     W2FLG
3087  007650  000137  010240                JMP     DCHKX
3088  007654  005037  000766        DCHKB:  CLR     DERFL
3089  007660  012701  017474                MOV     #WBUFF,R1       ;SET GOOD POINTER
3090  007664  012702  020106        DCHKB0: MOV     #RBUFF,R2       ;SET READ POINTER
3091  007670  032737  000020  000776        BIT     #20,UDES        ;SEE IF CORE DUMP
3092  007676  001416                        BEQ     DCHKD           ;IF NOT: BR
3093  007700  022737  000012  000750        CMP     #12,WAM         ;SEE IF WAM 2
3094  007706  001011                        BNE     DCHKD           ;IF NOT: BR
3095  007710  005737  001002                TST     PATRN           ;SEE IF PATTERN 0
3096  007714  001003                        BNE     DCHKC           ;IF NOT: BR
3097  007716  012701  001056                MOV     #WCDP0,R1       ;SET CORE DUMP PATTERN 0
3098  007722  000404                        BR      DCHKD           ;GO CHECK DATA
3099  007724  012701  001044        DCHKC:  MOV     #WCDP2,R1       ;SET CORE DUMP WRITE PATTERN 2
```

```
3100  007730  000401                          BR      DCHK0           ;GO CHECK DATA
3101  007732  000240                  DCHKD:  NOP
3102  007734  121112                  DCHK0:  CMPB    (R1),(R2)       ;SEE IF DATA OK
3103  007736  001466                          BEQ     DCHK2           ;IF SO: BR
3104  007740  032777  020000  170622          BIT     #20000,@SWR     ;SEE IF PRINT ERRORS
3105  007746  001062                          BNE     DCHK2           ;IF NOT: BR
3106  007750  005737  000620                  TST     HDRFL           ;SEE IF DONE HEADER
3107  007754  001004                          BNE     DCHK1           ;IF SO: BR
3108  007756  013704  000622                  MOV     EMADDR,R4
3109  007762  004737  013054                  JSR     PC,TTOUT        ;PRINT HEADER
3110  007766  005737  000766          DCHK1:  TST     DERFL           ;SEE IF FIRST ERROR
3111  007772  001014                          BNE     DCHK1A          ;IF NOT: BR
3112  007774  012704  016772                  MOV     #WMSG16,R4
3113  010000  004737  013054                  JSR     PC,TTOUT        ;PRINT DATA ERROR TAG
3114  010004  012704  017217                  MOV     #WMSG32,R4
3115  010010  004737  013054                  JSR     PC,TTOUT        ,PRINT PATRN TAG
3116  010014  013703  001002                  MOV     PATRN,R3
3117  010020  004737  013202                  JSR     PC,OCTP         ;PRINT PATTERN NUMBER
3118  010024  012737  000001  000620  DCHK1A: MOV     #1,HDRFL        ;SET HEADER FLAG
3119  010032  012737  000001  000766          MOV     #1,DERFL        ;SET DATA ERROR FLAG
3120  010040  012704  017016                  MOV     #WMSG21,R4
3121  010044  004737  013054                  JSR     PC,TTOUT        ;PRINT CHARACTER NUMBER TAG
3122  010050  013703  000774                  MOV     CRCNT,R3
3123  010054  004737  013202                  JSR     PC,OCTP         ;PRINT CHARACTER NUMBER
3124  010060  012704  017004                  MOV     #WMSG17,R4
3125  010064  004737  013054                  JSR     PC,TTOUT        ;PRINT GOOD TAG
3126  010070  111103                          MOVB    (R1),R3
3127  010072  004737  013430                  JSR     PC,DOUT         ;PRINT GOOD DATA
3128  010076  012704  017011                  MOV     #WMSG20,R4
3129  010102  004737  013054                  JSR     PC,TTOUT        ;PRINT BAD TAG
3130  010106  111203                          MOVB    (R2),R3
3131  010110  004737  013430                  JSR     PC,DOUT         ;PRINT BAD DATA
3132  010114  005737  000764          DCHK2:  TST     W2FLG           ;SEE IF WRAP 2 NRZ
3133  010120  001020                          BNE     DCHK2B          ;IF SO: BR
3134  010122  005201                          INC     R1              ;BUMP POINTER
3135  010124  032737  000020  000776          BIT     #20,UDES        ;SEE IF CORE DUMP
3136  010132  001413                          BEQ     DCHK2B          ;IF NOT: BR
3137  010134  022737  000012  000750          CMP     #12,WAM         ;SEE IF WAM 2
3138  010142  001006                          BNE     DCHK2A          ;IF NOT: BR
3139  010144  005201                          INC     R1              ;BUMP POINTER
3140  010146  005711                          TST     (R1)            ;SEE IF END OF PATTERN
3141  010150  001004                          BNE     DCHK2B          ;IF NOT: BR
3142  010152  162701  000010                  SUB     #10,R1          ;RESET POINTER TO START OF PATTERN
3143  010156  000401                          BR      DCHK2B          ;CONTINUE CHECK
3144  010160  000240                  DCHK2A: NOP
3145  010162  005202                  DCHK2B: INC     R2
3146  010164  022737  000030  000750          CMP     #30,WAM         ;BRANCH IF WAM 4
3147  010172  001404                          BEQ     1$
3148  010174  022737  000006  000750          CMP     #6,WAM          ;SEE IF WAM 0
3149  010202  001002                          BNE     DCHK3           ;IF NOT: BR
3150  010204  062701  000010          1$:     ADD     #10,R1          ;BUMP POINTER
3151  010210  005237  000774          DCHK3:  INC     CRCNT           ;BUMP CHAR CNTR
3152  010214  032777  000400  170346          BIT     #400,@SWR       ;SEE IF CONT DATA CHK
3153  010222  001006                          BNE     DCHKX           ;IF NOT: BR
3154  010224  005200                          INC     R0              ;SEE IF DONE
3155  010226  001242                          BNE     DCHK0           ;IF NOT: BR
```

```
3156  010230  005737  000764            TST     W2FLG
3157  010234  001401                    BEQ     DCHKX
3158  010236  000207                    RTS     PC
3159  010240  005777  170324  DCHKX:    TST     @SWR            ;SEE IF HALT ON ERROR
3160  010244  100004                    BPL     DCHKX1          ;IF NOT: BR
3161  010246  005737  000766            TST     DERFL           ;SEE IF DATA EPROR OCCURED
3162  010252  001401                    BEQ     DCHKX1          ;IF NOT: BR
3163  010254  000000                    HALT
3164  010256  005037  000774  DCHKX1:   CLR     CRCNT           ;CLEAR CHAR CNTR
3165  010262  005037  000620            CLR     HDRFL           ;CLEAR HEADER FLAG
3166  010266  005037  000766            CLR     DERFL           ;CLEAR DATA ERROR FLAG
3167  010272  005037  000770            CLR     PREFL           ;CLEAR PREAMBLE FLAG
3168  010276  000207                    RTS     PC              ;RETURN
```

```
3170
3171                                        ;EVEN PARITY DATA CHECK*****
3172
3173  010300  000240                DCHKE:  NOP
3174  010302  022737  000006 000750        CMP     #6,WAM          ;SEE IF WRAP 0
3175  010310  001005                        BNE     1$              ;IF NOT: BR
3176  010312  012700  177744              MOV     #-34.,R0        ;SET NUMBER OF CHARACTERS READ
3177  010316  012701  017504              MOV     #WBUFF+10,R1    ;SET POINTER
3178  010322  000404                      BR      2$
3179  010324  012700  177400        1$:   MOV     #-400,R0        ;SET NUMBER OF CHARACTERS
3180  010330  012701  017474              MOV     #WBUFF,R1       ;R1=START OF WRITE BUFFER
3181  010334  012702  020106        2$:   MOV     #RBUFF,R2       ;R2=START OF READ BUFFER
3182  010340  111105                DCKE0:  MOVB    (R1),R5         ;GET EXPT DATA
3183  010342  005003                        CLR     R3
3184  010344  012704  000010              MOV     #10,R4          ;SET NUMBER OF BITS
3185  010350  032705  000001        DCKE1:  BIT     #1,R5           ;SEE IF ONE BIT
3186  010354  001401                        BEQ     DCKE2           ;IF NOT: BR
3187  010356  005203                        INC     R3              ;COUNT ONE BITS FOR PARITY CHECK
3188  010360  005304                DCKE2:  DEC     R4              ;SEE IF DONE
3189  010362  001402                        BEQ     DCKE3           ;IF SO: BR
3190  010364  006005                        ROR     R5              ;POINT TO NEXT BIT
3191  010366  000770                        BR      DCKE1
3192  010370  000240                DCKE3:  NOP
3193  010372  111105                        MOVB    (R1),R5         ;GET EXPT DATA
3194  010374  042705  177400               BIC     #177400,R5      ;MASK DATA FIELD
3195  010400  005703                        TST     R3
3196  010402  001003                        BNE     DCKE4           ;IF NO ONE BITS SET: BR
3197  010404  012705  100020               MOV     #100020,R5
3198  010410  000405                        BR      DCKE5
3199  010412  032703  000001        DCKE4:  BIT     #1,R3           ;SEE IF ODD NUMBER OF ONE BITS
3200  010416  001402                        BEQ     DCKE5           ;IF NOT: BR
3201  010420  052705  100000               BIS     #100000,R5      ;SET EVEN PARITY BIT=1
3202  010424  042712  077400        DCKE5:  BIC     #77400,(R2)     ;MASK DATA FIELD
3203  010430  020512                        CMP     R5,(R2)         ;SEE IF DATA + PARITY GOOD
3204  010432  001477                        BEQ     DCKE10          ;IF SO: BR
3205  010434  032777  020000 170126        BIT     #20000,@SWR     ;SEE IF ERROR PRINT
3206  010442  001073                        BNE     DCKE10          ;IF NOT: BR
3207  010444  005737  000620               TST     HDRFL           ;SEE IF DONE HEADER
3208  010450  001004                        BNE     DCKE6           ;IF SO: BR
3209  010452  013704  000622               MOV     EMADDR,R4
3210  010456  004737  013054               JSR     PC,TTOUT        ;PRINT HEADER
3211  010462  005737  000766        DCKE6:  TST     DERFL           ;SEE IF FIRST BAD CHAR
3212  010466  001014                        BNE     DCKE7           ;IF NOT: BR
3213  010470  012704  016772               MOV     #WMSG16,R4
3214  010474  004737  013054               JSR     PC,TTOUT        ;PRINT BAD DATA TAG
3215  010500  012704  017217               MOV     #WMSG32,R4
3216  010504  004737  013054               JSR     PC,TTOUT        ;PRINT PATTERN TAG
3217  010510  013703  001002               MOV     PATRN,R3
3218  010514  004737  013202               JSR     PC,OCTP         ;PRINT PATTERN NUMBER
3219  010520  000240                DCKE7:  NOP
3220  010522  012737  000001 000766        MOV     #1,DERFL        ;SET DATA ERROR FLAG
3221  010530  012737  000001 000620        MOV     #1,HDRFL        ;SET HEADER FLAG
3222  010536  012704  017016               MOV     #WMSG21,R4
3223  010542  004737  013054               JSR     PC,TTOUT        ;PRINT CHAR NUMBER TAG
3224  010546  013703  000774               MOV     CRCNT,R3
3225  010552  004737  013202               JSR     PC,OCTP         ;ORINT CHAR NUMBER
```

```
3226  010556  012704  017004            MOV     #WMSG17,R4
3227  010562  004737  013054            JSR     PC,TTOUT        ;PRINT GOOD DATA TAG
3228  010566  110503                    MOVB    R5,R3
3229  010570  004737  013430            JSR     PC,DOUT         ;PRINT EXPT DATA
3230  010574  010503                    MOV     R5,R3
3231  010576  004737  010706            JSR     PC,DCKEP        ;GO PRINT PARITY BIT
3232  010602  000240                    NOP
3233  010604  012704  017011            MOV     #WMSG20,R4
3234  010610  004737  013054            JSR     PC,TTOUT        ;PRINT BAD TAG
3235  010614  111203                    MOVB    (R2),R3
3236  010616  004737  013430            JSR     PC,DOUT         ;PRINT BAD DATA
3237  010622  011203                    MOV     (R2),R3
3238  010624  004737  010706            JSR     PC,DCKEP        ;GO PRINT PARITY BIT
3239  010630  000240                    NOP
3240  010632  005201            DCKE10: INC     R1
3241  010634  022737  000006  000750    CMP     #6,WAM          ;SEE IF WRAP 0
3242  010642  001002                    BNE     1$              ;IF NOT: BR
3243  010644  062701  000010            ADD     #10,R1          ;BUMP POINTER
3244  010650  005722            1$:     TST     (R2)+           ;BUMP POINTERS
3245  010652  005237  000774            INC     CRCNT           ;BUMP CHAR CNTR
3246  010656  032777  000400  167704    BIT     #400,@SWR       ;SEE IF CONTINUE DATA CHECK
3247  010664  001402                    BEQ     DCKE11          ;IF SO: BR
3248  010666  000137  010240            JMP     DCHKX           ;GO TO END OF DATA CHECK
3249  010672  005200            DCKE11: INC     R0              ;SEE IF DONE
3250  010674  001402                    BEQ     DCKE12          ;IF SO: BR
3251  010676  000137  010340            JMP     DCKE0           ;ELSE CONTINUE
3252  010702  000137  010240    DCKE12: JMP     DCHKX           ;GO TO END OF DATA CHECK
3253  010706  000240            DCKEP:  NOP
3254  010710  012737  000240  000614    MOV     #240,TOB
3255  010716  004737  013154            JSR     PC,TOG          ;SPACE
3256  010722  012737  000260  000614    MOV     #260,TOB        ;SET PAR=0
3257  010730  005703                    TST     R3              ;SEE IF PARITY REALLY=0
3258  010732  100002                    BPL     DCKEP0          ;IF SO: BR
3259  010734  005237  000614            INC     TOB             ;ELSE SET TO 1
3260  010740  004737  013154    DCKEP0: JSR     PC,TOG          ;PRINT PARITY BIT
3261  010744  000207                    RTS     PC              ;RETURN
3262
```

```
3264
3265                                        ;PREAMBLE/POSTAMBLE CHECK*******
3266
3267  010746  012700  000051        PSCHK:  MOV     #51,R0          ;SET SIZE OF POSTAMBLE
3268  010752  012701  017352                MOV     #POST,R1        ;SET POINTER TO POSTAMBLE
3269  010756  005037  000620                CLR     HDRFL           ;CLEAR HEADER FLAG
3270  010762  005037  000774                CLR     CRCNT           ;CLEAR CHAR CNTR
3271  010766  005037  000766                CLR     DERFL           ;CLEAR DATA ERROR FLAG
3272  010772  000240                        NOP
3273  010774  000240                        NOP
3274  010776  000137  011020                JMP     PD0             ;GO CHECK POSTAMPLE
3275
3276  011002  012700  000051        PRCHK:  MOV     #51,R0          ;SET SIZE OF PREAMBLE
3277  011006  012701  017230                MOV     #PRE,R1         ;SET POINTER TO PREAMBLE
3278  011012  012702  020106                MOV     #RBUFF,R2       ;SET POINTER TO START OF READ BUFFER
3279  011016  022122                        CMP     (R1)+,(R2)+     ;BUMP ADDRESS POINTERS
3280  011020  121112          PD0:    CMPB    (R1),(R2)       ;CHECK DATA
3281  011022  001004                        BNE     PD1             ;IF NOT GOOD: BR
3282  011024  126162  000001  000001        CMPB    1(R1),1(R2)     ;COMPARE COMPLIMENT BYTE
3283  011032  001477                        BEQ     PD5             ;IF GOOD: BR
3284  011034  032777  020000  167526  PD1:  BIT     #20000,@SWR     ;SEE IF PRINT INHIBIT
3285  011042  001073                        BNE     PD5             ;IF SO: BR
3286  011044  005737  000620                TST     HDRFL           ;SEE IF DONE HEADER
3287  011050  001020                        BNE     PD4             ;IF SO: BR
3288  011052  013704  000622                MOV     EMADDR,R4
3289  011056  004737  013054                JSR     PC,TTOUT        ;PRINT TEST HEADER
3290  011062  005737  000770                TST     PREFL           ;SEE IF PREAMBLE CHECK
3291  011066  001403                        BEQ     PD2             ;IF NOT: BR
3292  011070  012704  017143                MOV     #WMSG29,R4      ;SET POSTAMBLE HEADER
3293  011074  000402                        BR      PD3
3294  011076  012704  017125          PD2:  MOV     #WMSG28,R4      ;SET PREAMBLE HEADER
3295  011102  004737  013054          PD3:  JSR     PC,TTOUT        ;PRINT HEADER
3296  011106  005237  000620                INC     HDRFL
3297  011112  012704  017016          PD4:  MOV     #WMSG21,R4
3298  011116  004737  013054                JSR     PC,TTOUT        ;PRINT CHAR NUMBER TAG
3299  011122  013703  000774                MOV     CRCNT,R3
3300  011126  004737  013202                JSR     PC,OCTP         ;PRINT CHAR NUMBER
3301  011132  012704  017004                MOV     #WMSG17,R4
3302  011136  004737  013054                JSR     PC,TTOUT        ;PRINT GOOD TAG
3303  011142  116103  000001                MOVB    1(R1),R3
3304  011146  004737  013430                JSR     PC,DOUT         ;PRINT GOOD CHAR
3305  011152  012737  000240  000614        MOV     #240,TOB
3306  011160  004737  013154                JSR     PC,TOG
3307  011164  111103                        MOVB    (R1),R3
3308  011166  004737  013430                JSR     PC,DOUT         ;PRINT COMPLEMENT
3309  011172  012704  017011                MOV     #WMSG20,R4
3310  011176  004737  013054                JSR     PC,TTOUT        ;PRINT BAD TAG
3311  011202  116203  000001                MOVB    1(R2),R3
3312  011206  004737  013430                JSR     PC,DOUT         ;PRINT BAD CHAR
3313  011212  012737  000240  000614        MOV     #240,TOB
3314  011220  004737  013154                JSR     PC,TOG
3315  011224  111203                        MOVB    (R2),R3
3316  011226  004737  013430                JSR     PC,DOUT         ;PRINT COMPLEMENT
3317  011232  022122          PD5:  CMP     (R1)+,(R2)+     ;BUMP ADDRESS POINTERS
3318  011234  005237  000774                INC     CRCNT           ;BUMP CHAR NUMBER
3319  011240  005300                        DEC     R0              ;SEE IF DONE
```

```
3320  011242  001266                    BNE     PD0             ;IF NOT: BR
3321  011244  005737  000770            TST     PREFL           ;SEE IF PREAMBLE
3322  011250  001402                    BEQ     PD6             ;IF SO: BR
3323  011252  000137  010240            JMP     DCHKX           ;GO TO EXIT ROUTINE
3324  011256  005237  000770    PD6:    INC     PREFL           ;SET PREAMBLE FLAG
3325  011262  005037  000620            CLR     HDRFL           ;CLEAR HEADER FLAG
3326  011266  005037  000774            CLR     CRCNT           ;CLEAR CHAR CNTR
3327  011272  005037  000766            CLR     DERFL           ;CLEAR DATA ERROR FLAG
3328  011276  000137  011302            JMP     W1DCHK          ;GO CHECK WRAP 1 DATA
3329
```

```
3331
3332                                            ;WAM 1 PE DATA CHECK*****
3333
3334  011302  012700  177400          W1DCHK: MOV     #-400,R0        ;SET NUMBER OF CHAR TO CHECK
3335  011306  012701  017474                  MOV     #WBUFF,R1       ;SET WRITE DATA POINTER
3336  011312  012702  020106                  MOV     #RBUFF,R2       ;SET READ DATA POINTER
3337  011316  062702  000124                  ADD     #124,R2         ;POINT TO START OF DATA
3338  011322  005737  000764                  TST     W2FLG           ;SEE IF WRAP 2
3339  011326  001401                          BEQ     W1D0            ;IF NOT WAM 2: BR
3340  011330  005302                          DEC     R2              ;RESET POINTER
3341  011332  111105          W1D0:   MOVB    (R1),R5
3342  011334  120512                  CMPB    R5,(R2)         ;CHECK DATA
3343  011336  001007                  BNE     W1D1            ;IF NOT GOOD:BR
3344  011340  005737  000764                  TST     W2FLG           ;SEE IF WRAP 2
3345  011344  001001                          BNE     W1D0A           ;IF SO: BR
3346  011346  105105                          COMB    R5              ;COMPLIMENT EXPT DATA
3347  011350  120562  000001          W1D0A:  CMPB    R5,1(R2)        ;CHECK COMPLIMENT DATA
3348  011354  001510                          BEQ     W1D3            ;IF GOOD: BR
3349  011356  032777  020000  167204  W1D1:   BIT     #20000,@SWR     ;SEE IF PRINT INHIBIT
3350  011364  001104                          BNE     W1D3            ;IF SO  BR
3351  011366  005737  000620                  TST     HDRFL           ;SEE IF DONE HEADER
3352  011372  001020                          BNE     W1D2            ;IF SO: BR
3353  011374  013704  000622                  MOV     EMADDR,R4
3354  011400  004737  013054                  JSR     PC,TTOUT        ;PRINT TEST HEADER
3355  011404  012704  016772                  MOV     #WMSG16,R4
3356  011410  004737  013054                  JSR     PC,TTOUT        ;PRINT BAD DATA TAG
3357  011414  012704  017217                  MOV     #WMSG32,R4
3358  011420  004737  013054                  JSR     PC,TTOUT        ;PRINT PATRN TAG
3359  011424  013703  001002                  MOV     PATRN,R3
3360  011430  004737  013202                  JSR     PC,OCTP         ;PRINT PATTERN NUMBER
3361  011434  012737  000001  000620  W1D2:   MOV     #1,HDRFL        ;SET HEADER FLAG
3362  011442  012704  017016                  MOV     #WMSG21,R4
3363  011446  004737  013054                  JSR     PC,TTOUT        ;PRINT CHAR NUMBER TAG
3364  011452  013703  000774                  MOV     CRCNT,R3
3365  011456  004737  013202                  JSR     PC,OCTP         ;PRINT CHAR NUMBER
3366  011462  012704  017004                  MOV     #WMSG17,R4
3367  011466  004737  013054                  JSR     PC,TTOUT        ;PRNT GOOD TAG
3368  011472  111105                          MOVB    (R1),R5
3369  011474  110503                          MOVB    R5,R3           ;GET GOOD CHAR
3370  011476  005737  000764                  TST     W2FLG           ;SEE IF WRAP 2
3371  011502  001001                          BNE     W1D2A           ;IF SO: BR
3372  011504  105103                          COMB    R3              ;ELSE COMPLIMENT CHAR
3373  011506  004737  013430          W1D2A:  JSR     PC,DOUT         ;PRINT CHARACTER
3374  011512  012737  000240  000614          MOV     #240,TOB
3375  011520  004737  013154                  JSR     PC,TOG          ;SPACE
3376  011524  110503                          MOVB    R5,R3
3377  011526  004737  013430                  JSR     PC,DOUT         ;PRINT CHAR
3378  011532  012704  017011                  MOV     #WMSG20,R4
3379  011536  004737  013054                  JSR     PC,TTOUT        ;PRINT BAD TAG
3380  011542  116203  000001                  MOVB    1(R2),R3
3381  011546  004737  013430                  JSR     PC,DOUT         ;PRINT BAD CHAR
3382  011552  012737  000240  000614          MOV     #240,TOB
3383  011560  004737  013154                  JSR     PC,TOG          ;SPACE
3384  011564  111203                          MOVB    (R2),R3
3385  011566  004737  013430                  JSR     PC,DOUT         ;PRINT CHAR
3386  011572  005237  000766                  INC     DERFL           ;SET DATA ERROR FLAG
```

```
 3387  011576  122122                 W1D3:   CMPB    (R1)+,(R2)+     ;BUMP ADDRESS
 3388  011600  105722                         TSTB    (R2)+           ;BUMP ADDRESS
 3389  011602  005237  000774                 INC     CRCNT           ;BUMP CHAR CNTR
 3390  011606  000406                         BR      W1D5
 3391  011610  005737  000764         W1D4:   TST     W2FLG           ;SEE IF WRAP 2
 3392  011614  001401                         BEQ     W1D4A           ;IF NOT: BR
 3393  011616  000207                         RTS     PC              ;ELSE RETURN
 3394  011620  000137  010746         W1D4A:  JMP     PSCHK           ;GO CHECK POSTAMBLE
 3395  011624  005200                 W1D5:   INC     R0
 3396  011626  001770                         BEQ     W1D4
 3397  011630  000137  011332                 JMP     W1D0
 3398
 3399                                         ;PREAMBLE/POSTAMBLE GENERATE SUBROUTINE*********
 3400
 3401  011634  000240                 PPGEN:  NOP
 3402  011636  012700  000050                 MOV     #50,R0          ;SET SIZE OF PREAMBLE
 3403  011642  012701  017230                 MOV     #PRE,R1
 3404  011646  005721                         TST     (R1)+           ;SET ADDRESS OF PRE
 3405  011650  012721  177400         1$:     MOV     #177400,(R1)+   ;FILL TABLE
 3406  011654  005300                         DEC     R0              ;SEE IF DONE
 3407  011656  001374                         BNE     1$              ;IF NOT: BR
 3408  011660  012701  017352                 MOV     #POST,R1        ;SET ADDRESS OF POST
 3409  011664  012700  000050                 MOV     #50,R0          ;SET SIZE OF POST
 3410  011670  012721  000377                 MOV     #377,(R1)+      ;SET SYNC CHAR
 3411  011674  012721  177400         2$:     MOV     #177400,(R1)+   ;FILL TABLE
 3412  011700  005300                         DEC     R0              ;SEE IF DONE
 3413  011702  001374                         BNE     2$              ;IF NOT: BR
 3414  011704  000207                         RTS     PC              ;RETURN
```

```
3416
3417                                            ;END OF RECORD FORCE SUBROUTINE*****
3418
3419  011706  005237  000676          EORPA:  INC     TEMP2           ;SET WRAP FLAG
3420  011712  017700  166616          EORP:   MOV     @MR,R0          ;GET MAINT REG
3421  011716  042700  000036                  BIC     #36,R0          ;CLEAR CURRENT OP CODE
3422  011722  052700  000024                  BIS     #24,R0          ;SET EOR CLEAR OP CODE
3423  011726  010077  166602                  MOV     R0,@MR          ;DO EOR
3424  011732  042777  000037  166574          BIC     #37,@MR         ;CLEAR EOR AND MM
3425  011740  005000                          CLR     R0
3426  011742  012701  000002                  MOV     #2,R1
3427  011746  032777  000001  166534  EORP1:  BIT     #1,@C1          ;SEE IF GO GONE
3428  011754  001427                          BEQ     EORP2           ;IF SO: BR
3429  011756  005300                          DEC     R0
3430  011760  001372                          BNE     EORP1           ;AWAIT GO RESET
3431  011762  005301                          DEC     R1
3432  011764  001370                          BNE     EORP1
3433  011766  032777  020000  166574          BIT     #20000,@SWR     ;SEE IF ERROR PRINT INHIBIT
3434  011774  001017                          BNE     EORP2           ;IF SO: BR
3435  011776  005737  000620                  TST     HDRFL           ;SEE IF DONE HEADER
3436  012002  001004                          BNE     EORP1A          ;IF SO: BR
3437  012004  013704  000622                  MOV     EMADDR,R4
3438  012010  004737  013054                  JSR     PC,TTOUT        ;PRINT HEADER
3439  012014  012704  017162          EORP1A: MOV     #WMSG31,R4
3440  012020  004737  013054                  JSR     PC,TTOUT        ;PRINT EOR GO BIT ERROR
3441  012024  005777  166540                  TST     @SWR            ;SEE IF HALT ON ERROR
3442  012030  100001                          BPL     EORP2           ;IF NOT: BR
3443  012032  000000                          HALT
3444  012034  000240          EORP2:  NOP
3445  012036  005737  000676                  TST     TEMP2           ;SEE IF WAM
3446  012042  001015                          BNE     EORPX           ;IF NOT: BR
3447  012044  032777  000200  166516          BIT     #200,@SWR       ;SEE IF STATUS CHECK
3448  012052  001002                          BNE     EORP3           ;IF NOT: BR
3449  012054  004737  007026                  JSR     PC,WSTCK        ;ELSE GO CHECK STATUS
3450  012060  000240          EORP3:  NOP
3451  012062  032777  000400  166500          BIT     #400,@SWR       ;SEE IF DATA CHECK
3452  012070  001002                          BNE     EORPX           ;IF NOT: BR
3453  012072  004737  007416                  JSR     PC,DCHK         ;ELSE GO CHECK DATA
3454  012076  000240          EORPX:  NOP
3455  012100  005037  000676                  CLR     TEMP2           ;CLEAR FLAG
3456  012104  000207                          RTS     PC              ;RETURN
3457
```

```
3459
3460                                        ;SCOPE LOOP ON ERROR SUBROUTINE**********
3461
3462  012106  000240                SCOPE:  NOP
3463  012110  032777  040000  166452        BIT     #40000,@SWR     ;SEE IF LOOP ON ERROR
3464  012116  001001                        BNE     1$              ;IF SO: BR
3465  012120  000207                        RTS     PC              ;ELSE EXIT
3466  012122  000240                1$:     NOP
3467  012124  005726                        TST     (SP)+           ;RESET STACK
3468  012126  000240                        NOP
3469  012130  000240                        NOP
3470  012132  000177  166554                JMP     @SCOLP          ;LOOP ON ERROR
3471
3472                                        ;TEST ITERATION SUBROUTINE**********
3473
3474  012136  032777  004000  166424  ITER:  BIT     #4000,@SWR      ;SEE IF ITERATIONS
3475  012144  001403                        BEQ     2$              ;IF SO: BR
3476  012146  005037  000702        1$:     CLR     ITCNT           ;CLEAR ITERATION COUNTER
3477  012152  000207                        RTS     PC              ;ELSE EXIT
3478  012154  005737  001016        2$:     TST     @#PCNTR         ;DO SINGLE SUBTEST ITERATION
3479  012160  001772                        BEQ     1$              ;ON FIRST PASS
3480  012162  005237  000702                INC     ITCNT           ;BUMP COUNTER
3481  012166  023737  000702  000606        CMP     ITCNT,ITAMT     ;SEE IF DONE ALL
3482  012174  001764                        BEQ     1$              ;IF SO: BR
3483  012176  005726                        TST     (SP)+           ;RESET STACK
3484  012200  017700  166510                MOV     @ITRLP,R0       ;SET ITERATION POINTER
3485  012204  000110                        JMP     (R0)            ;GO ITERATE
3486
```

```
3488
3489                                        ;INITIALIZE SUBROUTINE**********
3490
3491  012206  012777  000040  166304  INIT:  MOV   #40,@CS          ;INIT
3492  012214  013777  000624  166276         MOV   DRVN,@CS         ;SELECT DRIVE
3493  012222  013777  000664  166312         MOV   SLVN,@TC         ;SELECT SLAVE
3494  012230  013746  000776                 MOV   UDES,-(SP)       ;GET TEST'S UNIT DESCRIPTION
3495  012234  042716  174377                 BIC   #174377,(SP)     ;CLEAR ALL BUT DENSITY SELECT BITS
3496  012240  022726  001400                 CMP   #1400,(SP)+      ;BRANCH IF NOT NRZ (800 BPI)
3497  012244  001005                         BNE   1$
3498  012246  032777  000040  166246         BIT   #40,@DS          ;BRANCH IF SLAVE IS IN NRZ MODE
3499  012254  001420                         BEQ   4$               ;(PES = 0)
3500  012256  000404                         BR    2$               ;GO CHANGE DENSITY
3501  012260  032777  000040  166234  1$:    BIT   #40,@DS          ;BRANCH IF SLAVE IS IN PE MODE
3502  012266  001013                         BNE   4$               ;(PES = 1)
3503  012270  012777  000007  166212  2$:    MOV   #7,@C1           ;REWIND SLAVE
3504  012276  032777  000200  166216  20$:   BIT   #200,@DS         ;WAIT FOR READY
3505  012304  001774                         BEQ   20$
3506  012306  032777  020000  166206  3$:    BIT   #20000,@DS       ;LOOP UNTIL REWIND IS COMPLETE
3507  012314  001374                         BNE   3$               ;(PIP = 0)
3508  012316  053777  000776  166216  4$:    BIS   UDES,@TC         ;LOAD UNIT DESCRIPTION
3509  012324  032777  000002  166170         BIT   #2,@DS           ;BRANCH IF NOT AT BOT
3510  012332  001407                         BEQ   6$
3511  012334  012777  000025  166146         MOV   #25,@C1          ;ERASE TO GET OFF BOT
3512  012342  032777  000200  166152  5$:    BIT   #200,@DS         ;LOOP UNTIL DONE
3513  012350  001774                         BEQ   5$
3514  012352  012777  000011  166130  6$:    MOV   #11,@C1          ;DO A DRIVE CLEAR
3515  012360  000207                         RTS   PC               ;RETURN TO CALLER
3516
3517                                        ;MAG TAPE INTERRUPT HANDLER**********
3518
3519  012362  000240                 MTINT:  NOP
3520  012364  013716  000670                 MOV   RTRN,(SP)        ;SET RETURN TO (RTRN)
3521  012370  000002                         RTI                    ;RETURN
3522
3523                                        ;TTY INTERRUPT HANDLER**********
3524
3525  012372  017746  166176         TTINT:  MOV   @TKB,-(SP)       ;GET CHARACTER
3526  012376  042716  000200                 BIC   #200,(SP)        ;CLEAR PARITY BIT
3527  012402  122716  000003                 CMPB  #3,(SP) ;BRANCH IF NOT CONTROL C
3528  012406  001006                         BNE   1$
3529  012410  005737  001262                 TST   CHNFLG           ;INHIBIT ^C IF CHAIN MODE
3530  012414  001003                         BNE   1$
3531  012416  000005                         RESET
3532  012420  000137  000200                 JMP   @#200            ;RESTART PROGRAM
3533  012424  122716  000001         1$:     CMPB  #1,(SP)          ;BRANCH IF NOT ^A
3534  012430  001017                         BNE   2$
3535  012432  022737  000176  000570         CMP   #SWREG,SWR       ;BRANCH IF HARDWARE SWR INVOKED
3536  012440  001016                         BNE   3$
3537  012442  012737  177570  000570         MOV   #177570,SWR      ;INVOKE HARDWARE SWR
3538  012450  004737  013732                 JSR   PC,.SAVE         ;SAVE REGISTERS ON THE STACK
3539  012454  012704  014641                 MOV   #MSG63,R4        ;TYPE 'HARDWARE SWR IN USE'
3540  012460  004737  013054                 JSR   PC,TTOUT
3541  012464  004737  013754                 JSR   PC,.RESTORE      ;RESTORE REGISTERS
3542  012470  022716  000007         2$:     CMP   #7,(SP)          ;BRANCH IF NOT ^G
3543  012474  001005                         BNE   4$
```

```
 3544  012476  012737  000176  000570  3$:     MOV     #SWREG,SWR      ;INVOKE SOFTWARE SWR
 3545  012504  004737  013634          4$:     JSR     PC,GTSWR        ;GET SOFTWARE SWITCHES
 3546  012510  005726                  4$:     TST     (SP)+           ;POP CHARACTER OFF THE STACK
 3547  012512  000002                          RTI                     ;RETURN
 3548
```

```
3550                                    ;*****************************************************
3551                                    ;TTY ENTRY SUBROUTINE:
3552                                    ;
3553                                    ;THIS SUBROUTINE IS USED BY THE TEST CONDITION
3554                                    ;ENTRY ROUTINE TO READ THE RESPONSE ENTERED
3555                                    ;AT THE TTY AND CHECK THEM FOR LEGALITY AND
3556                                    ;LIMITS.  ALL RESPONSE MUST BE TYPED IN OCTAL
3557                                    ;(0-7) AND MUST FALL WITHIN THE LIMITS SET BY
3558                                    ;THE CALLING ROUTINE.
3559                                    ;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
3560                                    ;A QUESTION MARK IS TYPED (?) AND THE RESPONSE
3561                                    ;MAY BE REENTERED.
3562                                    ;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
3563                                    ;MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
3564                                    ;CARRIAGE RETURN
3565                                    ;*****************************************************
3566
3567  012514  010146            TTR:   MOV     R1,-(SP)          ;SAVE CHAR COUNT
3568  012516  011601            10$:   MOV     (SP),R1           ;RESET CHAR COUNT (FOR ^U)
3569  012520  005037  000674           CLR     TEMP1             ;CLEAR FIRST CHARACTER FLAG
3570  012524  005000                   CLR     R0
3571  012526  004737  012774    1$:    JSR     PC,TTIN           ;GO READ CHARACTER
3572  012532  122737  000003  000616   CMPB    #3,TIB            ;BRANCH IF NOT ^C
3573  012540  001003                   BNE     11$
3574  012542  000005                   RESET
3575  012544  000137  000200           JMP     @#200             ;RESTART AT 200
3576  012550  122737  000015  000616  11$:   CMPB    #15,TIB           ;SEE IF CR
3577  012556  001004                   BNE     2$                ;IF NOT: BR
3578  012560  005737  000674           TST     TEMP1             ;SEE IF FIRST CHARACTER
3579  012564  001471                   BEQ     9$                ;IF SO: BR
3580  012566  000457                   BR      6$                ;ELSE GO LOAD VALUE
3581  012570  122737  000025  000616   2$:    CMPB    #25,TIB           ;BRANCH IF NOT CONTROL U
3582  012576  001005                   BNE     21$
3583  012600  012704  014567           MOV     #MSG59,R4         ;TYPE<CR><LF>
3584  012604  004737  013054           JSR     PC,TTOUT
3585  012610  000742                   BR      10$
3586  012612  122737  000177  000616  21$:   CMPB    #177,TIB                  ;BRANCH IF NOT 'RUBOUT'
3587  012620  001012                   BNE     3$
3588  012622  000241                   CLC                       ;REMOVE LAST TYPED CHAR
3589  012624  006000                   ROR     R0
3590  012626  006200                   ASR     R0
3591  012630  006200                   ASR     R0
3592  012632  012704  014571           MOV     #MSG60,R4         ;TYPE '\'
3593  012636  004737  013054           JSR     PC,TTOUT
3594  012642  005201                   INC     R1                ;DECREMENT CHAR RECEIVED COUNT
3595  012644  000730                   BR      1$                ;GET NEXT CHAR
3596  012646  122737  000060  000616   3$:    CMPB    #60,TIB           ;SEE IF CHAR IS LESS THAN 0
3597  012654  101402                   BLOS    4$                ;IF NOT: BR
3598  012656  000137  012754           JMP     TINER             ;ELSE GO TO ERROR
3599  012662  122737  000070  000616   4$:    CMPB    #70,TIB           ;SEE IF CHAR IS GREATER THAN 7
3600  012670  101002                   BHI     5$                ;IF NOT: BR
3601  012672  000137  012754           JMP     TINER             ;ELSE GO TO ERROR
3602  012676  005237  000674           5$:    INC     TEMP1             ;SET FIRST CHARACTER FLAG
3603  012702  006300                   ASL     R0
3604  012704  006300                   ASL     R0                ;SHIFT 3 LEFT
3605  012706  006300                   ASL     R0
```

```
3606  012710  042737  177770  000616          BIC     #177770,TIB     ;STRIP ASCII
3607  012716  053700  000616                  BIS     TIB,R0          ;LOAD CHARACTER
3608  012722  005301                          DEC     R1              ;SEE IF DONE
3609  012724  001300                          BNE     1$              ;IF NOT: BR
3610  012726  020002                  6$:     CMP     R0,R2           ;SEE IF EXCEEDED MAXIMUM LIMIT
3611  012730  101402                          BLOS    7$              ;IF NOT: BR
3612  012732  000137  012754                  JMP     TINER           ;ELSE GO TO ERROR
3613  012736  020300                  7$:     CMP     R3,R0           ;SEE IF BELOW MINIMUM LIMIT
3614  012740  101402                          BLOS    8$              ;IF NOT: BR
3615  012742  000137  012754                  JMP     TINER           ;ELSE GO TO ERROR
3616  012746  010015                  8$:     MOV     R0,(R5)         ;LOAD VALUE
3617  012750  005726                  9$:     TST     (SP)+           ;POP CHAR COUNT OFF STACK
3618  012752  000207                          RTS     PC              ;EXIT
```

```
3620
3621                                        ;TTY ENTRY ERROR SUBROUTINE**********
3622
3623  012754  012704  014444        TINER:  MOV     #MSG40,R4
3624  012760  004737  013054                JSR     PC,TTOUT        ;PRINT?
3625  012764  005726                        TST     (SP)+           ;POP CHAR COUNT OFF STACK
3626  012766  162716  000020                SUB     #20,(SP)        ;RESET SP TO START OF VALUE ROUTINE
3627  012772  000207                        RTS     PC              ;REDO VALUE ENTRY
3628
3629                                        ;TTY READ SUBROUTINE*********
3630
3631  012774  017746  165566        TTIN:   MOV     @PSW,   -(SP)   ;SAVE CURRENT PSW
3632  013000  052777  000340  165560        BIS     #340,   @PSW    ;SET TO BR7 TO PREVENT INTRPT
3633  013006  005277  165560                INC     @TKS
3634  013012  105777  165554        1$:     TSTB    @TKS
3635  013016  100375                        BPL     1$
3636  013020  012677  165542                MOV     (SP)+,  @PSW    ;RESTORE PSW ,OK TO INTRPT NOW
3637  013024  017737  165544  000616        MOV     @TKB,TIB
3638  013032  042737  000200  000616        BIC     #200,TIB
3639  013040  013737  000616  000614        MOV     TIB,TOB         ;MOVE CHAR TO TTY OUPUT BFR
3640  013046  004737  013154                JSR     PC,TOG          ;ECHO CHARACTER
3641  013052  000207                        RTS     PC
3642
3643                                        ;TTY OUTPUT SUBROUTINE***********
3644
3645  013054  112437  000614        TTOUT:  MOVB    (R4)+,TOB
3646  013060  122737  000043  000614        CMPB    #43,TOB
3647  013066  001440                        BEQ     TEX
3648  013070  122737  000045  000614        CMPB    #45,TOB
3649  013076  001403                        BEQ     1$
3650  013100  004737  013154                JSR     PC,TOG
3651  013104  000763                        BR      TTOUT
3652  013106  112737  000015  000614  1$:   MOVB    #15,TOB
3653  013114  004737  013154                JSR     PC,TOG
3654  013120  012703  000004                MOV     #4,R3
3655  013124  005037  000614        2$:     CLR     TOB
3656  013130  004737  013154                JSR     PC,TOG
3657  013134  005303                        DEC     R3
3658  013136  001372                        BNE     2$              ;DO FILLERS
3659  013140  112737  000012  000614        MOVB    #12,TOB
3660  013146  004737  013154                JSR     PC,TOG
3661  013152  000740                        BR      TTOUT
3662  013154  105777  165416        TOG:    TSTB    @TPS
3663  013160  100375                        BPL     TOG
3664  013162  113777  000614  165410        MOVB    TOB,@TPB
3665  013170  000207                TEX:    RTS     PC
3666
3667
3668                                        ;OCTAL OUTPUT SUBROUTINE***********
3669
3670  013172  012737  000001  013426  OCTPE:  MOV     #1,OFL
3671  013200  000402                        BR      OCTPE1
3672  013202  005037  013426          OCTP:  CLR     OFL             ;CLEAR FLAG FOR LEADING ZERO
3673  013206  010304                OCTPE1: MOV     R3,R4           ;SEE IF NUMBER IS ZERO
3674  013210  001007                        BNE     OCTP0           ;IF NOT ZERO: BR
3675  013212  005737  013426                TST     OFL             ;SEE IF PRINT ALL 0
```

```
3676  013216  001004                          BNE     OCTPO           ;IF SO: BR
3677  013220  004737  013406                  JSR     PC,OCTPG1       ;ELSE PRINT ZERO
3678  013224  000137  013350                  JMP     OCTP3           ;SPACE AND EXIT
3679  013230  032704  100000          OCTPO:  BIT     #100000,R4      ;SEE IF MSD = 1
3680  013234  001406                          BEQ     OCTP1           ;IF NOT: BR
3681  013236  012704  000001                  MOV     #1,R4
3682  013242  004737  013364                  JSR     PC,OCTPG        ;PRINT 1
3683  013246  000137  013260                  JMP     OCTP2
3684  013252  005004                  OCTP1:  CLR     R4
3685  013254  004737  013364                  JSR     PC,OCTPG        ;PRINT 0
3686  013260  010304                  OCTP2:  MOV     R3,R4
3687  013262  006004                          ROR     R4
3688  013264  006004                          ROR     R4
3689  013266  006004                          ROR     R4              ;POSITION DIGIT
3690  013270  006004                          ROR     R4
3691  013272  000304                          SWAB    R4
3692  013274  004737  013364                  JSR     PC,OCTPG        ;PRINT DIGIT 2
3693  013300  010304                          MOV     R3,R4
3694  013302  006004                          ROR     R4
3695  013304  000304                          SWAB    R4
3696  013306  004737  013364                  JSR     PC,OCTPG        ;PRINT DIGIT 3
3697  013312  010304                          MOV     R3,R4
3698  013314  006104                          ROL     R4
3699  013316  006104                          ROL     R4
3700  013320  000304                          SWAB    R4
3701  013322  004737  013364                  JSR     PC,OCTPG        ;PRINT DIGIT 4
3702  013326  010304                          MOV     R3,R4
3703  013330  006004                          ROR     R4
3704  013332  006004                          ROR     R4
3705  013334  006004                          ROR     R4
3706  013336  004737  013364                  JSR     PC,OCTPG
3707  013342  010304                          MOV     R3,R4
3708  013344  004737  013364                  JSR     PC,OCTPG        ;PRINT DIGIT 5
3709  013350  012737  000240  000614  OCTP3:  MOV     #240,TOB
3710  013356  004737  013154                  JSR     PC,TOG          ;PRINT SPACE
3711  013362  000207                          RTS     PC              ;EXIT
3712  013364  042704  177770          OCTPG:  BIC     #177770,R4
3713  013370  001004                          BNE     OCTPG0
3714  013372  005737  013426                  TST     OFL
3715  013376  001001                          BNE     OCTPG0
3716  013400  000207                          RTS     PC
3717
3718  013402  005237  013426          OCTPG0: INC     OFL
3719  013406  052704  000260          OCTPG1: BIS     #260,R4
3720  013412  010437  000614                  MOV     R4,TOB
3721  013416  004737  013154                  JSR     PC,TOG
3722  013422  010304                          MOV     R3,R4
3723  013424  000207                          RTS     PC
3724  013426  000000                  OFL:    0                       ;FIRST CHAR FLAG
3725
```

```
3727
3728                                        ;DATA CHARACTER OUTPUT SUBROUTINE**********
3729
3730  013430  012704  000010        DOUT:   MOV     #10,R4          ;SET NUMBER TO PRINT
3731  013434  110337  000614                MOVB    R3,TOB
3732  013440  105777  165132        1$:     TSTB    @TPS
3733  013444  100375                        BPL     1$
3734  013446  132737  000200  000614      BITB    #200,TOB
3735  013454  001404                        BEQ     2$
3736  013456  012777  000061  165114      MOV     #061,@TPB
3737  013464  000403                        BR      3$
3738  013466  012777  000060  165104  2$:   MOV     #060,@TPB
3739  013474  006337  000614        3$:     ASL     TOB
3740  013500  005304                        DEC     R4
3741  013502  001356                        BNE     1$
3742  013504  000207                        RTS     PC
3743
3744  013506  013703  000700        DOUTD:  MOV     TEMP3,R3
3745  013512  000303                        SWAB    R3
3746  013514  004737  013430                JSR     PC,DOUT
3747  013520  013703  000700                MOV     TEMP3,R3
3748  013524  004737  013430                JSR     PC,DOUT
3749  013530  000207                        RTS     PC
3750
3751                                        ;TU45 SERIAL NUMBER PRINT SUBROUTINE**********
3752
3753  013532  010304                SNPT:   MOV     R3,R4
3754  013534  000304                        SWAB    R4
3755  013536  006004                        ROR     R4
3756  013540  006004                        ROR     R4
3757  013542  006004                        ROR     R4
3758  013544  006004                        ROR     R4              ;GET FIRST DIGIT
3759  013546  004737  013610                JSR     PC,SNPG         ;PRINT
3760  013552  010304                        MOV     R3,R4
3761  013554  000304                        SWAB    R4              ;GET SECOND DIGIT
3762  013556  004737  013610                JSR     PC,SNPG         ;PRINT
3763  013562  010304                        MOV     R3,R4
3764  013564  006004                        ROR     R4
3765  013566  006004                        ROR     R4
3766  013570  006004                        ROR     R4
3767  013572  006004                        ROR     R4
3768  013574  004737  013610                JSR     PC,SNPG         ;PRINT THIRD DIGIT
3769  013600  010304                        MOV     R3,R4
3770  013602  004737  013610                JSR     PC,SNPG         ;PRINT FOURTH DIGIT
3771  013606  000207                        RTS     PC              ;EXIT
3772  013610  012737  000260  000614  SNPG:   MOV     #260,TOB        ;SET BASE = 0
3773  013616  042704  177760                BIC     #177760,R4      ;MASK DIGIT
3774  013622  050437  000614                BIS     R4,TOB          ;SET ASCII
3775  013626  004737  013154                JSR     PC,TOG          ;TYPE DIGIT
3776  013632  000207                        RTS     PC              ;RETURN
```

```
3778
3779                                    ;THIS ROUTINE GETS THE NEW VALUE FOR THE SOFTWARE SWITCH REG
3780
3781  013634  022737  000176  000570  GTSWR:  CMP     #SWREG,SWR      ;BRANCH IF SOFTWARE SWR NOT
3782  013642  001032                          BNE     1$              ;INVOKED
3783  013644  004737  013732                  JSR     PC,.SAVE        ;SAVE REGISTERS ON THE STACK
3784  013650  012704  016156                  MOV     #$MSWR,R4       ;TYPE 'SWR = '
3785  013654  004737  013054                  JSR     PC,TTOUT
3786  013660  017703  164704                  MOV     @SWR,R3         ;GET CURRENT SETTING
3787  013664  004737  013172                  JSR     PC,OCTPE        ;AND TYPE THEM
3788  013670  012704  016166                  MOV     #$MNEW,R4       ;TYPE ' NEW = '
3789  013674  004737  013054                  JSR     PC,TTOUT
3790  013700  013705  000570                  MOV     SWR,R5          ;TTR ROUTINE RETURN NEW VALUE TO (R5)
3791  013704  012701  000007                  MOV     #7,R1           ;LIMIT RESPONSE TO 7 CHARS
3792  013710  012702  177777                  MOV     #177777,R2      ;BETWEEN 0 AND 177777
3793  013714  012703  000000                  MOV     #0,R3
3794  013720  004737  012514                  JSR     PC,TTR          ;GET RESPONSE
3795  013724  004737  013754                  JSR     PC,.RESTORE     ;RESTORE REGISTERS
3796  013730  000207                  1$:     RTS     PC              ;RETURN TO CALLER
3797
3798                                    ;;ROUTINE TO SAVE REGISTERS ON THE STACK
 (1)  013732  010546                  .SAVE:  MOV     %5,-(SP)        ;;R5 IS SAVED AT 12(SP)
 (1)  013734  010446                          MOV     %4,-(SP)        ;;R4 IS SAVED AT 10(SP)
 (1)  013736  010346                          MOV     %3,-(SP)        ;;R3 IS SAVED AT 6(SP)
 (1)  013740  010246                          MOV     %2,-(SP)        ;;R2 IS SAVED AT 4(SP)
 (1)  013742  010146                          MOV     %1,-(SP)        ;;R1 IS SAVED AT 2(SP)
 (1)  013744  010046                          MOV     %0,-(SP)        ;;R0 IS SAVED AT (SP)
 (1)  013746  016646  000014                  MOV     14(SP),-(SP)    ;;PUSH RETURN PC ON THE STACK
 (1)  013752  000207                          RTS     PC              ;;RETURN TO CALLER
 (1)
3799                                    ;;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
 (1)  013754  012666  000014          .RESTORE:MOV    (SP)+,14(SP)    ;;STORE RETURN PC ON STACK
 (1)  013760  012600                          MOV     (SP)+,%0
 (1)  013762  012601                          MOV     (SP)+,%1
 (1)  013764  012602                          MOV     (SP)+,%2
 (1)  013766  012603                          MOV     (SP)+,%3
 (1)  013770  012604                          MOV     (SP)+,%4
 (1)  013772  012605                          MOV     (SP)+,%5
 (1)  013774  000207                          RTS     PC              ;;RETURN
 (1)
```

```
3801
3802                                                    ;MESSAGE TABLE**********
3803
3804  013776  022445  046524  031460  MSG1:   .ASCII'%%TM03-TU45 CONTROL LOGIC TEST PART II  (CZTUPB0)';++B
      014004  052055  032125  020065
      014012  047503  052116  047522
      014020  020114  047514  044507
      014026  020103  042524  052123
      014034  050040  051101  020124
      014042  044511  020040  041450
      014050  052132  050125  030102
      014056     051
3805  014057     045  025052  040452          .ASCII  /%***ASSURE TAPE IS AT BOT***/
      014064  051523  051125  020105
      014072  040524  042520  044440
      014100  020123  052101  041040
      014106  052117  025052     052
3806  014113     045  054524  042520          .ASCII  /%TYPE <CR> TO TERMINATE RESPONSE & ^C TO RESTART%#/
      014120  036040  051103  020076
      014126  047524  052040  051105
      014134  044515  040516  042524
      014142  051040  051505  047520
      014150  051516  020105  020046
      014156  041536  052040  020117
      014164  042522  052123  051101
      014172  022524     043
3807  014175     105  050130  026524  MSG6:   .ASCII  /EXPT-NOT RECVD#/
      014202  047516  020124  042522
      014210  053103  021504
3808  014214  041522  042126  047055  MSG7:   .ASCII  /RCVD-NOT EXPT#/
      014222  052117  042440  050130
      014230  021524
3809  014232  047045  047117  042455  MSG9:   .ASCII  /%NON-EXIST SLAVE #/
      014240  044530  052123  051440
      014246  040514  042526  021440
3810  014254  051045  040505  020104  MSG10:  .ASCII  /%READ CONT BUS PAR #/
      014262  047503  052116  041040
      014270  051525  050040  051101
      014276  021440
3811  014300  053445  044522  042524  MSG11:  .ASCII  /%WRITE CONT BUS PAR #/
      014306  041440  047117  020124
      014314  052502  020123  040520
      014322  020122     043
3812  014325     040  054105  052120  MSG12:  .ASCII  / EXPT #/
      014332  021440
3813  014334  050040  053103  020104  MSG13:  .ASCII  / RCVD #/
      014342     043
3814  014343     045  051115  041040  MSG14:  .ASCII  /%MR BITS 4-0#/
      014350  052111  020123  026464
      014356  021460
3815  014360  046445  020122  044502  MSG15:  .ASCII  /%MR BITS 15-7#/
      014366  051524  030440  026465
      014374  021467
3816  014376  044445  042524  035122  MSG16:  .ASCII  /%ITER: #/
      014404  021440
3817  014406  052045  020103  044502  MSG18:  .ASCII  /%TC BITS 12-0 #/
```

```
                014414  051524  030440  026462
                014422  020060     043
3818            014425     045  041506  041040  MSG19:  .ASCII  /%FC BITS 15-0 #/
                014432  052111  020123  032461
                014440  030055  021440
3819            014444  037440  021440          MSG40:  .ASCII  / ? #/
3820            014450  022445  047105  020104  MSG41:  .ASCII  /%%END OF PASS #/
                014456  043117  050040  051501
                014464  020123     043
3821            014467     045  042522  044507  MSG44:  .ASCII  /%REGISTER START: #/
                014474  052123  051105  051440
                014502  040524  052122  020072
                014510     043
3822            014511     045  042526  052103  MSG45:  .ASCII  /%VECTOR ADDRESS: #/
                014516  051117  040440  042104
                014524  042522  051523  020072
                014532     043
3823            014533     045  046524  031460  MSG57:  .ASCII  /%TM03 DRIVE: #/
                014540  042040  044522  042526
                014546  020072     043
3824            014551     045  052524  032464  MSG58:  .ASCII  /%TU45 SLAVE: #/
                014556  051440  040514  042526
                014564  020072     043
3825            014567     045     043          MSG59:  .ASCII  /%#/
3826            014571     134     043          MSG60:  .ASCII  /\#/
3827            014573     045  042522  047515  MSG62:  .ASCII  /%REMOVE TMDP FROM SLAVE TO BE TESTED%#/
                014600  042526  052040  042115
                014606  020120  051106  046517
                014614  051440  040514  042526
                014622  052040  020117  042502
                014630  052040  051505  042524
                014636  022504     043
3828            014641     045  040510  042122  MSG63:  .ASCII  /%HARDWARE SWR IN USE%#/
                014646  040527  042522  051440
                014654  051127  044440  020116
                014662  051525  022505     043
3829            014667     045  020045  051104  MSG64:  .ASCII  /%% DRIVE #/
                014674  053111  020105     043
3830            014701     054  046123  053101  MSG65:  .ASCII  /,SLAVE #/
                014706  020105     043
3831            014711     116  052117  047440  MSG66:  .ASCII  /NOT ON LINE #/
                014716  020116  044514  042516
                014724  021440
```

```
3833                                                    ;TEST HEADER**********
3834
3835  014726  022445  047514  044507  MSLT1:  .ASCII  /%%LOGIC TEST 1: WRAP 3,NRZ,NORMAL,ODD#/
      014734  020103  042524  052123
      014742  030440  020072  051127
      014750  050101  031440  047054
      014756  055122  047054  051117
      014764  040515  026114  042117
      014772  021504
3836  014774  022445  047514  044507  MSLT2:  .ASCII  /%%LOGIC TEST 2: WRAP 3,PE,NORMAL,ODD#/
      015002  020103  042524  052123
      015010  031040  020072  051127
      015016  050101  031440  050054
      015024  026105  047516  046522
      015032  046101  047454  042104
      015040     043
3837  015041     045  046045  043517  MSLT3:  .ASCII  /%%LOGIC TEST 3: WRAP 2,NRZ,NORMAL,ODD#/
      015046  041511  052040  051505
      015054  020124  035063  053440
      015062  040522  020120  026062
      015070  051116  026132  047516
      015076  046522  046101  047454
      015104  042104     043
3838  015107     045  046045  043517  MSLT4:  .ASCII  /%%LOGIC TEST 4: WRAP 2,PE,NORMAL,ODD#/
      015114  041511  052040  051505
      015122  020124  035064  053440
      015130  040522  020120  026062
      015136  042520  047054  051117
      015144  040515  026114  042117
      015152  021504
3839  015154  022445  047514  044507  MSLT5:  .ASCII  /%%LOGIC TEST 5: WRAP 1,NRZ,NORMAL,ODD#/
      015162  020103  042524  052123
      015170  032440  020072  051127
      015176  050101  030440  047054
      015204  055122  047054  051117
      015212  040515  026114  042117
      015220  021504
3840  015222  022445  047514  044507  MSLT6:  .ASCII  /%%LOGIC TEST 6: WRAP 1,PE,NORMAL,ODD#/
      015230  020103  042524  052123
      015236  033040  020072  051127
      015244  050101  030440  050054
      015252  026105  047516  046522
      015260  046101  047454  042104
      015266     043
3841  015267     045  046045  043517  MSLT7:  .ASCII  /%%LOGIC TEST 7: WRAP 0,NRZ,NORMAL,ODD#/
      015274  041511  052040  051505
      015302  020124  035067  053440
      015310  040522  020120  026060
      015316  051116  026132  047516
      015324  046522  046101  047454
      015332  042104     043
3842  015335     045  046045  043517  MSLT10: .ASCII  /%%LOGIC TEST 10: WRAP 0,PE,NORMAL,ODD#/
      015342  041511  052040  051505
      015350  020124  030061  020072
      015356  051127  050101  030040
      015364  050054  026105  047516
```

```
                015372  046522  046101  047454
                015400  042104    043
 3843           015403     045  046045  043517  MSLT11: .ASCII  /%%LOGIC TEST 11: CORE DUMP WRITE (M8906)#/
                015410  041511  052040  051505
                015416  020124  030461  020072
                015424  047503  042522  042040
                015432  046525  020120  051127
                015440  052111  020105  046450
                015446  034470  033060  021451
 3844           015454  022445  047514  044507  MSLT12: .ASCII  /%%LOGIC TEST 12: CORE DUMP READ (M8906)#/
                015462  020103  042524  052123
                015470  030440  035062  041440
                015476  051117  020105  052504
                015504  050115  051040  040505
                015512  020104  046450  034470
                015520  033060  021451
 3845           015524  022445  047514  044507  MSLT13: .ASCII  /%%LOGIC TEST 13: EVEN PARITY WRITE (M8933 M8934)#/
                015532  020103  042524  052123
                015540  030440  035063  042440
                015546  042526  020116  040520
                015554  044522  054524  053440
                015562  044522  042524  024040
                015570  034115  031471  020063
                015576  034115  031471  024464
                015604     043
 3846           015605     045  046045  043517  MSLT14: .ASCII  /%%LOGIC TEST 14: EVEN PARITY READ(M8933 M8934)#/
                015612  041511  052040  051505
                015620  020124  032061  020072
                015626  053105  047105  050040
                015634  051101  052111  020131
                015642  042522  042101  046450
                015650  034470  031463  046440
                015656  034470  032063  021451
 3847           015664  022445  047514  044507  MSLT15: .ASCII  /%%LOGIC TEST 15: READ REVERSE(M8906)#/
                015672  020103  042524  052123
                015700  030440  035065  051040
                015706  040505  020104  042522
                015714  042526  051522  024105
                015722  034115  030071  024466
                015730     043
 3848           015731     045  046045  043517  MSLT16: .ASCII  /%%LOGIC TEST 16: CRC CORRECTION SINGLE TRACK,ALL FRAMES#/
                015736  041511  052040  051505
                015744  020124  033061  020072
                015752  051103  020103  047503
                015760  051122  041505  044524
                015766  047117  051440  047111
                015774  046107  020105  051124
                016002  041501  026113  046101
                016010  020114  051106  046501
                016016  051505     043
 3849           016021     045  046045  043517  MSLT17: .ASCII  /%%LOGIC TEST 17: CRC CORRECTION MULTIPLE BAD TRACKS#/
                016026  041511  052040  051505
                016034  020124  033461  020072
                016042  051103  020103  047503
                016050  051122  041505  044524
                016056  047117  046440  046125
```

```
        016064  044524  046120  020105
        016072  040502  020104  051124
        016100  041501  051513     043
  3850  016105     045  046045  043517  MSLT20: .ASCII  /%%LOGIC TEST 20: READ REVERSE,NRZ,WRAP 3#/
        016112  041511  052040  051505
        016120  020124  030062  020072
        016126  042522  042101  051040
        016134  053105  051105  042523
        016142  047054  055122  053454
        016150  040522  020120  021463
```

```
                                                ;TAG MESSAGE
                                                
016156  051445  051127  036440  $MSWR:  .ASCII  /%SWR = #/
016164  021440
016166  047040  053505  036440  $MNEW:  .ASCII  / NEW = #/
016174  021440
                                                
                                                .EVEN
                                                ;WRITE BUFFER
                                                
016176  000100                  WDA1A:
016176  177777                          -1
(1) 016200  177777                      -1
(1) 016202  177777                      -1
(1) 016204  177777                      -1
(1) 016206  177777                      -1
(1) 016210  177777                      -1
(1) 016212  177777                      -1
(1) 016214  177777                      -1
(1) 016216  177777                      -1
(1) 016220  177777                      -1
(1) 016222  177777                      -1
(1) 016224  177777                      -1
(1) 016226  177777                      -1
(1) 016230  177777                      -1
(1) 016232  177777                      -1
(1) 016234  177777                      -1
(1) 016236  177777                      -1
(1) 016240  177777                      -1
(1) 016242  177777                      -1
(1) 016244  177777                      -1
(1) 016246  177777                      -1
(1) 016250  177777                      -1
(1) 016252  177777                      -1
(1) 016254  177777                      -1
(1) 016256  177777                      -1
(1) 016260  177777                      -1
(1) 016262  177777                      -1
(1) 016264  177777                      -1
(1) 016266  177777                      -1
(1) 016270  177777                      -1
(1) 016272  177777                      -1
(1) 016274  177777                      -1
(1) 016276  177777                      -1
(1) 016300  177777                      -1
(1) 016302  177777                      -1
(1) 016304  177777                      -1
(1) 016306  177777                      -1
(1) 016310  177777                      -1
(1) 016312  177777                      -1
(1) 016314  177777                      -1
(1) 016316  177777                      -1
(1) 016320  177777                      -1
(1) 016322  177777                      -1
(1) 016324  177777                      -1
```

```
   (1)  016326  177777                    -1
   (1)  016330  177777                    -1
   (1)  016332  177777                    -1
   (1)  016334  177777                    -1
   (1)  016336  177777                    -1
   (1)  016340  177777                    -1
   (1)  016342  177777                    -1
   (1)  016344  177777                    -1
   (1)  016346  177777                    -1
   (1)  016350  177777                    -1
   (1)  016352  177777                    -1
   (1)  016354  177777                    -1
   (1)  016356  177777                    -1
   (1)  016360  177777                    -1
   (1)  016362  177777                    -1
   (1)  016364  177777                    -1
   (1)  016366  177777                    -1
   (1)  016370  177777                    -1
   (1)  016372  177777                    -1
   (1)  016374  177777                    -1
3864
3865
3866                                       ;READ BUFFER
3867
3868  016376  000100               RDATA:
3870  016376  000000                       0
  (1)  016400  000000                       0
  (1)  016402  000000                       0
  (1)  016404  000000                       0
  (1)  016406  000000                       0
  (1)  016410  000000                       0
  (1)  016412  000000                       0
  (1)  016414  000000                       0
  (1)  016416  000000                       0
  (1)  016420  000000                       0
  (1)  016422  000000                       0
  (1)  016424  000000                       0
  (1)  016426  000000                       0
  (1)  016430  000000                       0
  (1)  016432  000000                       0
  (1)  016434  000000                       0
  (1)  016436  000000                       0
  (1)  016440  000000                       0
  (1)  016442  000000                       0
  (1)  016444  000000                       0
  (1)  016446  000000                       0
  (1)  016450  000000                       0
  (1)  016452  000000                       0
  (1)  016454  000000                       0
  (1)  016456  000000                       0
  (1)  016460  000000                       0
  (1)  016462  000000                       0
  (1)  016464  000000                       0
  (1)  016466  000000                       0
  (1)  016470  000000                       0
  (1)  016472  000000                       0
```

```
 (1)  016474  000000                              0
 (1)  016476  000000                              0
 (1)  016500  000000                              0
 (1)  016502  000000                              0
 (1)  016504  000000                              0
 (1)  016506  000000                              0
 (1)  016510  000000                              0
 (1)  016512  000000                              0
 (1)  016514  000000                              0
 (1)  016516  000000                              0
 (1)  016520  000000                              0
 (1)  016522  000000                              0
 (1)  016524  000000                              0
 (1)  016526  000000                              0
 (1)  016530  000000                              0
 (1)  016532  000000                              0
 (1)  016534  000000                              0
 (1)  016536  000000                              0
 (1)  016540  000000                              0
 (1)  016542  000000                              0
 (1)  016544  000000                              0
 (1)  016546  000000                              0
 (1)  016550  000000                              0
 (1)  016552  000000                              0
 (1)  016554  000000                              0
 (1)  016556  000000                              0
 (1)  016560  000000                              0
 (1)  016562  000000                              0
 (1)  016564  000000                              0
 (1)  016566  000000                              0
 (1)  016570  000000                              0
 (1)  016572  000000                              0
 (1)  016574  000000                              0
3871
3872                                              ;WRAP AROUND MESSAGES******
3873
3874  016576  051445  052105  050125  WMSG2:  .ASCII  /%SETUP ERROR%#/
       016604  042440  051122  051117
       016612  021445
3875  016614  050045  052101  047122  WMSG3:  .ASCII  /%PATRN NUMBER = #/
       016622  047040  046525  042502
       016630  020122  020075     043
3876  016635     045  047516  026516  WMSG4:  .ASCII  /%NON-EXISTANT DRIVE%#/
       016642  054105  051511  040524
       016650  052116  042040  044522
       016656  042526  021445
3877  016662  041445  030523  021440  WMSG6:  .ASCII  /%CS1 #/
3878  016670  053445  020103     043  WMSG6A: .ASCII  /%WC #/
3879  016675     045  040502  021440  WMSG6B: .ASCII  /%BA #/
3880  016702  043045  020103     043  WMSG6C: .ASCII  /%FC #/
3881  016707     045  051503  020062  WMSG6D: .ASCII  /%CS2 #/
       016714     043
3882  016715     045  051504  021440  WMSG6E: .ASCII  /%DS #/
3883  016722  042445  020122     043  WMSG6F: .ASCII  /%ER #/
3884  016727     045  051501  021440  WMSG6G: .ASCII  /%AS #/
3885  016734  041445  020103     043  WMSG6H: .ASCII  /%CC #/
```

```
                                                                    H 6
 3886  016741     045  041104  021440  WMSG6I: .ASCII  /%DB #/
 3887  016746  046445  020122     043  WMSG6J: .ASCII  /%MR #/
 3888  016753     045  052104  021440  WMSG6K: .ASCII  /%DT #/
 3889  016760  052045  020103     043  WMSG6L: .ASCII  /%TC #/
 3890  016765     045  047123  021440  WMSG6M: .ASCII  /%SN #/
 3891  016772  041045  042101  042040  WMSG16: .ASCII  /%BAD DATA#/
       017000  052101  021501
 3892  017004  043445  020072     043  WMSG17: .ASCII  /%G: #/
 3893  017011     045  035102  021440  WMSG20: .ASCII  /%B: #/
 3894  017016  041445  035116  021440  WMSG21: .ASCII  /%CN: #/
 3895  017024  041045  042101  051440  WMSG23: .ASCII  /%BAD STATUS#/
       017032  040524  052524  021523
 3896  017040  047045  020117  047111  WMSG24: .ASCII  /%NO INTERRUPT#/
       017046  042524  051122  050125
       017054  021524
 3897  017056  047045  020117  046103  WMSG25: .ASCII  /%NO CLOCK UP#/
       017064  041517  020113  050125
       017072     043
 3898  017073     045  047516  041440  WMSG26: .ASCII  /%NO CLOCK DOWN#/
       017100  047514  045503  042040
       017106  053517  021516
 3899  017112  042045  052101  020101  WMSG27: .ASCII  /%DATA PAT:#/
       017120  040520  035124     043
 3900  017125     045  040502  020104  WMSG28: .ASCII  /%BAD PREAMBLE#/
       017132  051120  040505  041115
       017140  042514     043
 3901  017143     045  040502  020104  WMSG29: .ASCII  /%BAD POSTAMBLE#/
       017150  047520  052123  046501
       017156  046102  021505
 3902  017162  042445  051117  041440  WMSG31: .ASCII  /%EOR CLEAR DID NOT CLEAR GO%#/
       017170  042514  051101  042040
       017176  042111  047040  052117
       017204  041440  042514  051101
       017212  043440  022517     043
 3903  017217     040  040520  051124  WMSG32: .ASCII  / PATRN #/
       017224  020116     043
 3904
 3905          017230                          .EVEN
 3906  017230  000000                  PRE:    0
 3909  017232  000000                          0
  (1)  017234  000000                          0
  (1)  017236  000000                          0
  (1)  017240  000000                          0
  (1)  017242  000000                          0
  (1)  017244  000000                          0
  (1)  017246  000000                          0
  (1)  017250  000000                          0
  (1)  017252  000000                          0
  (1)  017254  000000                          0
  (1)  017256  000000                          0
  (1)  017260  000000                          0
  (1)  017262  000000                          0
  (1)  017264  000000                          0
  (1)  017266  000000                          0
  (1)  017270  000000                          0
  (1)  017272  000000                          0
```

```
   (1)  017274  000000                          0
   (1)  017276  000000                          0
   (1)  017300  000000                          0
   (1)  017302  000000                          0
   (1)  017304  000000                          0
   (1)  017306  000000                          0
   (1)  017310  000000                          0
   (1)  017312  000000                          0
   (1)  017314  000000                          0
   (1)  017316  000000                          0
   (1)  017320  000000                          0
   (1)  017322  000000                          0
   (1)  017324  000000                          0
   (1)  017326  000000                          0
   (1)  017330  000000                          0
   (1)  017332  000000                          0
   (1)  017334  000000                          0
   (1)  017336  000000                          0
   (1)  017340  000000                          0
   (1)  017342  000000                          0
   (1)  017344  000000                          0
   (1)  017346  000000                          0
   (1)  017350  000000                          0
3910    017352  000000                POST:     0
3913    017354  000000                          0
   (1)  017356  000000                          0
   (1)  017360  000000                          0
   (1)  017362  000000                          0
   (1)  017364  000000                          0
   (1)  017366  000000                          0
   (1)  017370  000000                          0
   (1)  017372  000000                          0
   (1)  017374  000000                          0
   (1)  017376  000000                          0
   (1)  017400  000000                          0
   (1)  017402  000000                          0
   (1)  017404  000000                          0
   (1)  017406  000000                          0
   (1)  017410  000000                          0
   (1)  017412  000000                          0
   (1)  017414  000000                          0
   (1)  017416  000000                          0
   (1)  017420  000000                          0
   (1)  017422  000000                          0
   (1)  017424  000000                          0
   (1)  017426  000000                          0
   (1)  017430  000000                          0
   (1)  017432  000000                          0
   (1)  017434  000000                          0
   (1)  017436  000000                          0
   (1)  017440  000000                          0
   (1)  017442  000000                          0
   (1)  017444  000000                          0
   (1)  017446  000000                          0
   (1)  017450  000000                          0
   (1)  017452  000000                          0
```

```
   (1)  017454  000000                        0
   (1)  017456  000000                        0
   (1)  017460  000000                        0
   (1)  017462  000000                        0
   (1)  017464  000000                        0
   (1)  017466  000000                        0
   (1)  017470  000000                        0
   (1)  017472  000000                        0
 3914   017474  000000              WBUFF:    0
 3915           020106                        .=.+410
 3916   020106  000000              RBUFF:    0
 3917
 3918           000001                        .END
```

```
ADDFL   000746          2068#
AS      000526          1983#
ASF     000730          2061#
ATAF    000720          2057#
BA      000514          1978#   2788*
CC      000530          1984#
CHNFLG  001262          2176#*  2253    2291    2301    3529
CRCDAT  004370          2536*   2538    2541    2543*   2576#   2600*   2601*   2620    2763
CRCFLG  001022          2090#   2241*   2533*   2535*   2562*   2596*   2599*   2619*   2770    3048
CRCNT   000774          2079#   3055*   3122    3151*   3164*   3224    3245*   3270*   3299    3318*   3326*   3364    3389*
CRCPAT  004366          2523*   2539    2540    2565*   2567*   2575#   2588*   2601
CS      000520          1980#   2253*   2789    2926*   2998    3491*   3492*
C1      000510          1976#   2207    2253    2774*   2817*   2843*   2927*   2994    3427    3503*   3511*   3514*
DATAD   000760          2073#   2720*   2735*   2744*   2748*   2763*   2788
DATA0   001034          2103#
DATA1   001036          2104#
DATA2   001040          2105#
DATA3   001042          2106#
DATBL   001034          2102#   2681
DATC    000754          2071#   2719*   2734*   2762*   2801*   2802*   2803*   2804*   2816
DAT1    005162          2103    2691#
DAT1A   005166          2622    2692#   2701    2706    2711
DAT2    005202          2104    2700#
DAT3    005206          2105    2705#
DAT4    005214          2106    2710#
DB      000532          1985#
DCHK    007416          2969    3047#   3453
DCHKA   007514          3060    3062    3064#
DCHKA0  007470          3057    3059#
DCHKA1  007544          3066    3068    3070#
DCHKA2  007602          3073    3078#
DCHKB   007654          3079    3081    3084    3088#
DCHKB0  007664          3077    3090#
DCHKC   007724          3096    3099#
DCHKD   007732          3094    3101#
DCHKE   010300          3058    3173#
DCHKFL  001020          2089#   2642*   2645*   3051
DCHKX   010240          3087    3153    3157    3159#   3248    3252    3323
DCHKX1  010256          3050    3160    3162    3164#
DCHK0   007734          3092    3098    3100    3102#   3155
DCHK1   007766          3107    3110#
DCHK1A  010024          3111    3118#
DCHK2   010114          3103    3105    3132#
DCHK2A  010160          3138    3144#
DCHK2B  010162          3133    3136    3141    3143    3145#
DCHK3   010210          3149    3151#
DCKEP   010706          3231    3238    3253#
DCKEP0  010740          3258    3260#
DCKE0   010340          3182#   3251
DCKE1   010350          3185#   3191
DCKE10  010632          3204    3206    3240#
DCKE11  010672          3247    3249#
DCKE12  010702          3250    3252#
DCKE2   010360          3186    3188#
DCKE3   010370          3189    3192#
DCKE4   010412          3196    3199#
```

```
DCKE5   010424          3198    3200    3202#
DCKE6   010462          3208    3211#
DCKE7   010520          3212    3219#
DERFL   000766          2076#   3054*   3076*   3088*   3110    3119*   3161    3166*   3211    3220*   3271*   3327*   3386*
DOUT    013430          3127    3131    3229    3236    3304    3308    3312    3316    3373    3377    3381    3385    3730#
                        3746    3748
DOUTD   013506          3744#
DRVN    000624          2027#   2220    2222    2253*   2258    3000    3492
DRVTP   000604          2012#
DS      000522          1981#   2254    2818    3003    3498    3501    3504    3506    3509    3512
DSUP    005026          2663#   2780
DS0     005124          2664    2666    2678#
DS3     005142          2682#
DS4     005152          2684#   2686
DT      000536          1987#   2253
EMADDR  000622          2026#   2319*   2338*   2348*   2367*   2377*   2396*   2407*   2425*   2435*   2453*   2474*   2484*
                        2494*   2524*   2589*   2638*   2979    2981    3026    3108    3209    3288    3353    3437
ENDFLG  001024          2091#   2213
EORP    011712          2913    3420#
EORPA   011706          3419#
EORPX   012076          3446    3452    3454#
EORP1   011746          3427#   3430    3432
EORP1A  012014          3436    3439#
EORP2   012034          3428    3434    3442    3444#
EORP3   012060          3448    3450#
ER      000524          1982#   3007
ERADD   000672          2046#   2993*   2997*   3002*   3006*   3031
ERRF    000726          2060#
EXEC    006074          2723    2738    2755    2766    2833#
EXFL    000716          2056#
EXW2    006540          2840    2917
EXW2A   006570          2921    2923#
EXW2B   006624          2929#   2930    2936
EXW2C   006640          2932#   2933
EXW2E   006660          2937#
EXW2F   006666          2939#   2940    2942
EXW2G   006716          2944    2947#
EXW2H   006726          2918    2948    2950#
EXW2J   006742          2954#   2962
EXW2J0  006762          2957    2960#
EXW2J1  006764          2959    2961#
EXW2K   007004          2964    2966#
EXW2X   007022          2968    2970#
EX0     006116          2836    2838#
EX1     006142          2843#
EX1A    006156          2846#   2849    2851
EX2     006176          2847    2852#
EX3     006222          2855    2857#
EX5     006272          2864    2868#
EX5A    006276          2860    2862    2867    2869#   2909
EX5A1   006332          2874    2877#
EX5B    006366          2876    2879    2882    2884#
EX5C    006434          2870    2896#
EX5C0   006500          2901    2905#
EX5C1   006504          2904    2906#
EX5D    006510          2895    2907#
```

```
FC      000516  1979#  2783*  2786*
FUN     000752  2070#  2718*  2733*  2743*  2750*  2761*  2817
GTSWR   013634  2242   3545   3781#
HDRFL   000620  2025#  2271*  2978*  3024   3030*  3053*  3106   3118*  3165*  3207   3221*  3269*  3286
                3296*  3325*  3351   3361*  3435
HERE    002404  2300#
ILFT    000544  1993#
INIT    012206  2781   3491#
ITAMT   000606  2013#  3481
ITCNT   000702  2050#  3476*  3480*  3481
ITER    012136  2327   2356   2385   2415   2444   2464   2571   2625   2646   3474#
ITRLP   000714  2055#  2269*  2270*  3484
LTADD   000742  2066#  2267*  2268*  2269   2272   2288*  2289*
LT1     002436  2126   2127   2315#
LT1A    002502  2321#  2340   2499
LT1B    002514  2322   2323#  2326
LT10    003274  2140   2141   2420#
LT10A   003276  2421#
LT11    003346  2142   2143   2431#
LT11A   003424  2438   2439#  2443
LT11X   003450  2441   2444#
LT12    003460  2144   2145   2449#
LT12A   003544  2456   2458#  2463
LT12X   003576  2460   2464#
LT13    003606  2146   2147   2470#
LT14    003656  2148   2149   2480#
LT15    003726  2150   2151   2490#
LT16    004010  2152   2153   2522#
LT16A   004036  2525   2527#  2566   2570
LT17    004376  2154   2155   2587#
LT17A   004426  2590   2591#
LT2     002550  2128   2129   2333#
LT2A    002552  2334#
LT20    004710  2156   2157   2633#
LT20A   004766  2640   2641#
LT3     002622  2130   2131   2344#
LT3A    002666  2350#  2369
LT3B    002700  2351   2352#  2355
LT4     002730  2132   2133   2362#
LT4A    002732  2363#
LT5     003002  2134   2135   2373#
LT5A    003046  2379#  2398   2476
LT5B    003060  2380   2381#  2384
LT6     003110  2136   2137   2390#
LT6A    003112  2391#
LT7     003166  2138   2139   2403#
LT7A    003232  2409#  2427   2486
LT7B    003244  2410   2411#  2414
MR      000534  1986#  2814*  2815*  2816*  2835   2837*  2861*  2884*  2885   2894*  2896*  2897*  2898
                2910   2912*  2929   2931   2932   2934   3420   3423*  3424*
MSG1    013776  2181   2183*  3804#
MSG10   014254  3810#
MSG11   014300  3811#
MSG12   014325  3033   3812#
MSG13   014334  3037   3813#
MSG14   014343  3814#
```

```
MSG15   014360          3815#
MSG16   014376          3816#
MSG18   014406          3817#
MSG19   014425          3818#
MSG40   014444          3623    3819#
MSG41   014450          2293    3820#
MSG44   014467          2184    3821#
MSG45   014511          2193    3822#
MSG57   014533          2218    3823#
MSG58   014551          2227    3824#
MSG59   014567          3583    3825#
MSG6    014175          3807#
MSG60   014571          3592    3826#
MSG62   014573          2179    3827#
MSG63   014641          3539    3828#
MSG64   014667          2256    3829#
MSG65   014701          2260    3830#
MSG66   014711          2264    3831#
MSG7    014214          3808#
MSG9    014232          3809#
MSLT1   014726          2319    3835#
MSLT10  015335          2425    3842#
MSLT11  015403          2435    3843#
MSLT12  015454          2453    3844#
MSLT13  015524          2474    3845#
MSLT14  015605          2484    3846#
MSLT15  015664          2494    2981    3847#
MSLT16  015731          2524    3848#
MSLT17  016021          2589    3849#
MSLT2   014774          2338    2979    3836#
MSLT20  016105          2638    3850#
MSLT3   015041          2348    3837#
MSLT4   015107          2367    3838#
MSLT5   015154          2377    3839#
MSLT6   015222          2396    3840#
MSLT7   015267          2407    3841#
MTINT   012362          1968    2203    3519#
NRZOF   000662          2042#
NXTDRV  001760          2253#
NXTSLV  002032          2253#   2266    2292
OCTP    013202          2187    2196    2221    2230    2259    2263    2296    2670    3036    3040    3117    3123    3218
                        3225    3300    3360    3365    3672#
OCTPE   013172          3670#   3787
OCTPE1  013206          3671    3673#
OCTPG   013364          3682    3685    3692    3696    3701    3706    3708    3712#
OCTPG0  013402          3713    3715    3718#
OCTPG1  013406          3677    3719#
OCTP0   013230          3674    3676    3679#
OCTP1   013252          3680    3684#
OCTP2   013260          3683    3686#
OCTP3   013350          3678    3709#
OFL     013426          3670*   3672*   3675    3714    3718*   3724#
PATRN   001002          2082#   2321*   2324*   2325    2350*   2353*   2354    2379*   2382*   2383    2409*   2412*   2413
                        2437*   2440    2442*   2455*   2459    2461*   2522*   2568*   2569    2587*   2633*   2669    2671
                        2677    2679    2988    3095    3116    3217    3359
PCNTR   001016          2088#   2240*   2295    2306*   3478
```

```
PD0     011020          3274    3280#   3320
PD1     011034          3281    3284#
PD2     011076          3291    3294#
PD3     011102          3293    3295#
PD4     011112          3287    3297#
PD5     011232          3283    3285    3317#
PD6     011256          3322    3324#
PEXFL   000736          2064#
PFLG    000666          2044#
POST    017352          3268    3408    3910#
PPGEN   011634          2391    3401#
PRCHK   011002          3063    3276#
PRE     017230          3277    3403    3906#
PREFL   000770          2077#   3167*   3290    3321    3324*
PSCHK   010746          3267#   3394
PSW     000566          2005#   2852*   2928*   3631    3632*   3636*
RBUFF   020106          2683    2721    2736    2744    2764    2923    2951    3090    3181    3278    3336    3916#
RCDP    001010          2085#   2457*   2462*   2753
RDAD    000762          2074#   2721*   2736*   2764*   2902*   2903*   2905*   2906*
RDATA   016376          3868#
RDRVF   001006          2084#   2239*   2328*   2498*   2551*   2608*   2641*   2644*   2746    2805    2873
REGS    000612          2015#   2186    2188    2205
RTRN    000670          2045#   2918*   3520
SAV1    000704          2051#
SAV2    000706          2052#
SAV3    000710          2053#
SCF     000732          2062#
SCOLP   000712          2054#   2322*   2351*   2380*   2410*   2438*   2456*   2525*   2590*   2640*   3470
SCOPE   012106          3462#
SERFL   000772          2078#   2977*   3012    3021*
SERNUM  000602          2011#
SETUP   005514          2722    2737    2754    2765    2770#
SET0    005556          2777    2780#
SET1    005562          2779    2781#   2798
SET1A   005566          2775    2783#
SET2    005612          2785    2787#
SRAT    001014          2087#
SLAF    000722          2058#
SLVN    000664          2043#   2229    2231    2253*   2262    3493
SN      000540          1988#
SNPG    013610          3759    3762    3768    3770    3772#
SNPT    013532          3753#
SP0     005670          2796    2798#
SP01    005656          2792    2795#
SP1     005674          2790    2799#
SP1A    005746          2806    2809#
SP1B    005766          2800    2808    2811    2813#
SP3     006040          2819    2822#
SP4     006060          2823    2826#
SP5     006072          2827    2829#
SSCF    000724          2059#
START   001200          1959    2164#
STATC   001012          2086#
STATF   001004          2083#
STFLG   000740          2065#   2249*   2278    2283*   2663    2776
STFLGS  000614          2022#   2212    2213
```

```
STSCD   002274          2252    2283#
ST2     001670          1963    2238#
SWR     000570          2006#   2169    2173*   2176*   2250    2274    2280    2303    2665    2791    2795    2822    2826
                        2943    2947    2963    2967    3014    3022    3104    3152    3159    3205    3246    3284    3349
                        3433    3441    3447    3451    3463    3474    3535    3537*   3544*   3781    3786    3790
SWREG   000176          1954#   2173    2176    3535    3544    3781
TADX    001174          2158#
TC      000542          1989#   2253*   2813*   2846    3493*   3508*
TEMP1   000674          2047#   3569*   3578    3602*
TEMP2   000676          2048#   3419*   3445    3455*
TEMP3   000700          2049#   3744    3747
TEND    002330          2158    2285    2291#
TENDX   002426          2302    2304    2306#
TEX     013170          3647    3665#
TIB     000616          2024#   3572    3576    3581    3586    3596    3599    3606*   3607    3637*   3638*   3639
TINER   012754          3598    3601    3612    3615    3623#
TKB     000574          2008#   3525    3637
TKS     000572          2007#   2247*   3633*   3634
TLAST   001176          2159#   2284
TOB     000614          2023#   3254*   3256*   3259*   3305*   3313*   3374*   3382*   3639*   3645*   3646    3648    3652*
                        3655*   3659*   3664    3709*   3720*   3731*   3734    3739*   3772*   3774*
TOG     013154          3255    3260    3306    3314    3375    3383    3640    3650    3653    3656    3660    3662#   3663
                        3710    3721    3775
TPB     000600          2010#   3664*   3736*   3738*
TPS     000576          2009#   3662    3732
TREF    000734          2063#
TR00    000626          2028#
TR01    000630          2029#
TR02    000632          2030#
TR03    000634          2031#
TR04    000636          2032#
TR05    000640          2033#
TR06    000642          2034#
TR07    000644          2035#
TR10    000646          2036#
TR11    000650          2037#
TR12    000652          2038#
TR13    000654          2039#
TR14    000656          2040#
TR15    000660          2041#
TSCD    001714          2176    2247#   2282    2307
TSCDA   002174          2253    2255    2267#
TSCD0   002202          2268#   2279
TSCD1   002210          2269#   2290
TSCD2   002236          2274#   2329    2357    2386    2416    2445    2465    2572    2626    2647
TSCD3   002254          2275    2278#
TSTTBL  001070          2124#   2267    2288
TTIN    012774          3571    3631#
TTINT   012372          1948    3525#
TTOUT   013054          2180    2182    2185    2194    2219    2228    2257    2261    2265    2294    2668    2794    2825
                        2946    3027    3029    3032    3034    3038    3109    3113    3115    3121    3125    3129    3210
                        3214    3216    3223    3227    3234    3289    3295    3298    3302    3310    3354    3356    3358
                        3363    3367    3379    3438    3440    3540    3584    3593    3624    3645#   3651    3661    3785
                        3789
TTR     012514          2192    2201    2226    2235    2675    3567#   3794
T24FL   000744          2067#
```

```
UDES    000776  2080# 2320* 2339* 2349* 2368* 2378* 2397* 2408* 2426* 2436* 2454* 2475* 2485*
                2495* 2531* 2595* 2639* 2751  2784  2810  2813  2854  2859  2863  2878  2900
                2920  2956  3056  3061  3067  3080  3091  3135  3494  3508
VECT    000610  2014# 2195  2197  2202
WAM     000750  2069# 2717* 2727* 2732* 2742* 2760* 2772  2799  2815  2838  2841  2857  2869
                2924  3059  3065  3070  3072  3078  3093  3137  3146  3148  3174  3241
WAM0    005222  2411  2534  2597  2717#
WAM01   005230  2718# 2728
WAM1    005266  2381  2727#
WAM2    005276  2352  2439  2732#
WAM2A   005332  2737#
WAM3    005342  2323  2458  2556  2613  2643  2742#
WAM3A   005422  2747  2751#
WAM3B   005440  2752  2754#
WAM4    005450  2549  2563  2606  2623  2760#
WBUFF   017474  2536  2600  2621  2678  2720  2755  2745  3075  3089  3177  3180  3335  3914#
WC      000512  1977# 2787*
WCDP0   001056  2115# 2457  3097
WCDP2   001044  2110# 2462  3099
WCS1    001024  2095# 2315* 2334* 2344* 2363* 2373* 2392* 2403* 2421* 2431* 2449* 2470* 2480*
                2490* 2527* 2545* 2552* 2558* 2591* 2602* 2609* 2615* 2634* 2990* 2995
WCS2    001026  2096# 2316* 2335* 2345* 2364* 2374* 2393* 2404* 2422* 2432* 2450* 2471* 2481*
                2491* 2528* 2546* 2553* 2559* 2592* 2603* 2610* 2616* 2635* 2999
WDATA   016176  3861#
WDS     001030  2097# 2317* 2336* 2346* 2365* 2375* 2394* 2405* 2423* 2433* 2451* 2472* 2482*
                2492* 2529* 2547* 2554* 2560* 2593* 2604* 2611* 2617* 2636* 2991* 3004
WER     001032  2098# 2318* 2337* 2347* 2366* 2376* 2395* 2406* 2424* 2434* 2452* 2473* 2483*
                2493* 2530* 2548* 2555* 2561* 2594* 2605* 2612* 2618* 2637* 2992* 3010
WMSG16  016772  3112  3213  3355  3891#
WMSG17  017004  3124  3226  3301  3366  3892#
WMSG2   016576  2824  3874#
WMSG20  017011  3128  3233  3309  3378  3893#
WMSG21  017016  3120  3222  3297  3362  3894#
WMSG23  017024  3028  3895#
WMSG24  017040  2945  3896#
WMSG25  017056  3897#
WMSG26  017073  3898#
WMSG27  017112  3899#
WMSG28  017125  3294  3900#
WMSG29  017143  3292  3901#
WMSG3   016614  2667  3875#
WMSG31  017162  3439  3902#
WMSG32  017217  3114  3215  3357  3903#
WMSG4   016635  2793  3876#
WMSG6   016662  2993  3877#
WMSG6A  016670  3878#
WMSG6B  016675  3879#
WMSG6C  016702  3880#
WMSG6D  016707  2997  3881#
WMSG6E  016715  3002  3882#
WMSG6F  016722  3006  3883#
WMSG6G  016727  3884#
WMSG6H  016734  3885#
WMSG6I  016741  3886#
WMSG6J  016746  3887#
WMSG6K  016753  3888#
```

```
WMSG6L  016760          3889#
WMSG6M  016765          3890#
WPGFL   001000          2081#   2248*   2277*   2676*   2677*   2778    2983    2985
WSTCK   007026          2965    2976#   3449
WSTG    007272          2996    3001    3005    3011    3018#
WSTGO   007342          3025    3030#
WSTX    007414          3013    3015    3017    3020    3023    3041#
WTAD    000756          2072#   2745*   2749*   2753*   2801    2807*   2809*   2812*   2871    2875*   2877*   2880*   2881
                        2883*
W1DCHK  011302          3085    3328    3334#
W1DO    011332          3339    3341#   3397
W1DOA   011350          3345    3347#
W1D1    011356          3343    3349#
W1D2    011434          3352    3361#
W1D2A   011506          3371    3373#
W1D3    011576          3348    3350    3387#
W1D4    011610          3391#   3396
W1D4A   011620          3392    3394#
W1D5    011624          3390    3395#
W2FLG   000764          2075#   3083*   3086*   3132    3156    3338    3344    3370    3391
XORDAT  004372          2538*   2540*   2542*   2543    2577#
XORPAT  004374          2539*   2541*   2542    2578#
$DONE   002336          2253    2293#
$ENDAD  002374          1944    2176    2300#
$MNEW   016166          3788    3856#
$MSWR   016156          3784    3855#
$SVPC =  000764         1944#
.       = 020110        1942#   1944#   1947#   1953#   1958#   1962#   1967#   1973#   3905#   3915#
.RESTO  013754          3541    3795    3799#
.SAVE   013732          3538    3783    3798#
```

```
DIBOOT   1243#
GETANS    767#
LDPDR     515#
LPDP11   1268#
PSPTAG    746#
REGBOX    132#
RESLDR    873#
SAVLDR    855#
SVTK$    1141#
$CATCH   1124#    1855#    1942
$CHAIN     89#    1855#    2176
$CHNMO    105#    1855#    2253
$CNV16    606#
$CNV18    635#
$CNV48    704#
$CPCHK    897#
$CPREG     17#
$CPVEC    167#
$FPREG     46#
$GETAN    771#
$KMMRE    347#
$KWPDR    998#
$KW11     929#
$LCTRL      2#
$LPREG    186#
$MAMFO   1176#
$MMBIT    207#
$MMREG    264#
$PDRBI    385#
$POWER    439#
$PSWBI    147#
$RECO     795#
$RESLD    876#
$RESTO    476#    1855#    3799
$SAVE     464#    1855#    3798
$SAVLD    858#
$SETTB    508#
$SHIFT    489#
$SMMRE    309#
$STINS      8#
$STKPT    202#
$ST200   1136#
$SVTK    1146#
$SWOPT     56#
$TCDRV   1029#
$TCREG    192#
$TRAPS    402#
$TYPE     518#
$TYPEF    592#
$UMMRE    272#
$VECTA   1163#
.$ACT1     67#    1855#    1944
.$EOP      78#    1855#    2300

. ABS.  020110     000
```

ERRORS DETECTED: 0

CZTUPB,CZTUPB/CRF=CZTUPB.P11
RUN-TIME: 14 23 2 SECONDS
RUN-TIME RATIO: 101/41-2.4
CORE USED: 14K (27 PAGES)