TR79F

TR79F TAPE DIAGNOSTIC

CZTRABO

AH-9428B-MC

FICHE 1 OF 1

MAR 1978

digital

MADE IN USA

```
.REM ;
```

IDENTIFICATION

PRODUCT CODE: AC-9426B-MC

PRODUCT NAME: CZTRABO TR79F TAPE DIAG

DATE RELEASED: FEB 1978

MAINTAINER: DIAGNOSTIC GROUP

MODIFIED BY: BILL SCHLITZKUS

A2. REQUIREMENTS

A. PDP-11 COMPUTER

B. MAGNETIC TAPE TRANSPORT CONTROL INTERFACE

C. 12K WORDS OF MEMORY
D. CONSOLE TELEPRINTER
E. PAPER TAPE READER

A3. LOADING PROCEDURE

THE DIAGNOSTIC IS IN ABSOLUTE BINARY FORMAT AND IS LOADED USING THE ABSOLUTE LOADER.

A4. STARTING PROCEDURE

A. LOAD ADDRESS 200
B. DEPRESS START

TEST SELECTION:

AFTER THE TYPEOUT "SELECT TEST NUMBER..." THE OPERATOR SHOULD SELECT THE DESIRED TEST BY TYPING THE NUMBER FOLLOWED BY A CARRIAGE RETURN.

NOTE: THE NORMAL SWITCH SETTINGS ARE ALL SWITCHES DOWN

A6. TEST GROUP ABSTRACT

PRETST: PRETEST

THIS TEST GROUP CHECKS THE RESET FUNCTION. IT IS PERFORMED EACH TIME A TEST GROUP IS SELECTED.

NOTE: IN THE EVENT OF ERRORS IN THE PRETEST GROUP IT IS RECOMMENDED THAT THE RESET SWITCH ON THE TAPE TRANSPORT BE DEPRESSED AND THE TEST GROUP BE RESTARTED AT THE RESTART ADDRESS.

NOTE: WHEN TEST 000 IS SEEN IN AN ERROR TYPEOUT, AN ERROR DURING AUTOSIZING HAS OCCURRED. TEST 0 IS NOT A TEST GROUP.

TEST 1

TEST ONE IS A MANUAL INTERVENTION TEST REQUIRING THE OPERATOR TO PERFORM THE TASKS AS DIRECTED BY THE TTY. IT IS SUGGESTED THAT THE EOT STICKER BE PLACED ABOUT 50 FEET FROM THE LOAD POINT.

TEST 2

TEST GROUP TWO IS A LOGIC TEST GROUP WHICH CHECKS THE INTERFACE LOGIC.

TEST 3

TEST GROUP THREE IS A LOGIC TEST GROUP WHICH TESTS THE TRANSPORTS RESPONSE TO ALL TYPES OF COMMANDS.

TEST 4

TEST GROUP FOUR IS A RELIBALITY TEST GROUP WHICH CHECKS THE ABILITY OF THE TRANSPORT AND THE INTERFACE OVER AN EXTENDED PERIOD OF TIME.

TEST 5

TEST GROUP FIVE IS THE READ PART OF A COMPATIBILITY TEST. THIS TEST GROUP READS TAPES WRITTEN BY TEST GROUP 4.

TEST 6

TEST GROUP SIX IS A MAINTENANCE AID THIS ROUTINE ALLOWS THE OPERATOR TO LOOP ON ANY SUBTEST BY ENTERING THE STARTING ADDRESS AT THE TTY. THE LOOP CONTROL WILL KEEP THE SELECTED TEST GROUP RUNNING IF LOOP IS NOT SELECTED THE TEST GROUP WILL EXECUTE ONCE AND ANOTHER STARTING ADDRESS MUST BE ENTERED.

NOTE: IF ANY ERRORS DETECTED IN BOTH TEST GROUP 3 AND 4 HAVE THE SAME PASS NUMBER A DEFECTIVE TAPE SHOULD BE THE FIRST CONSIDERATION.

```
                IF SWITCHES #3 AND #5 ARE DOWN THEN TESTS 2 THROUGH 4 ARE LOOPED ON WHERE
                NO OPERATOR INTERVENTION IS REQUIRED.
                IF SWITCHES #3 AND #5 ARE UP THEN ALL TESTS ARE PERFORMED.
```

A7. ERRORS

THERE ARE TWO BASIC ERROR TYPEOUTS ONE OF WHICH CONTAINS THE FOLLOWING INFORMATION. THE OTHER ONLY GIVES THE PASS, PC, AND SR. WHEN THIS SHORT ERROR PRINTOUT IS TYPED THE MESSAGE WHICH FOLLOWS WILL GIVE AMPLE INFORMATION ABOUT THE ERROR.

```
PASS  XXX
SUBTEST    PC          STATUS      CORRECT     ACTUAL      LOCATION
XXXXXX     XXXXXX      XXXXXX      XXXXXX      XXXXXX      XXXXXX



        PASS=    XXXXXX  (THIS IS THE CURRENT PASS NUMBER)
        SUBTEST=XXXXXX   (THIS IS THE STARTING ADDRESS OF THE
                         CURRENT SUBTEST BEING EXECUTED)
        PC=      XXXXXX  (THIS IS THE PC FROM WHICH ERROR WAS CALLED)
        STATUS=  XXXXXX  (THIS IS THE CONTENTS OF THE PROCESSOR STATUS
                         REGISTER AT THE TIME OF THE ERROR)
        CORRECT=XXXXXX   (THIS IS WHAT SHOULD HAVE BEEN IN THE
                         REGISTER OR LOCATION BEING TESTED)
        ACTUAL=  XXXXXX  (THIS IS WHAT WAS IN THE REGISTER OR
                         LOCATION AT THE TIME IT WAS EXAMINED)
        LOCATION=XXXXXX (THIS IS THE LOCATION THAT WAS IN ERROR)
        COMMENT          (OPTIONAL)
;
```

```
                        ;*AC-9426B-MC /(377)/TR79F CHECKOUT TESTS
                        ;*COPYRIGHT 1975, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
                        ;*-------------------------------------------------------------

                                ;STARTING PROCEDURE
                                ;LOAD PROGRAM
                                ;LOAD ADDRESS 000200
                                ;PRESS START
                                ;PROGRAM WILL TYPE "AC-9426B-MC /(377)/TR79F CHECKOUT TESTS"
                                ;PROGRAM WILL TYPE "AUTOSIZING" TO INDICATE THAT AUTOSIZING HAS STARTED
                                ;(SWITCH #6 MUST BE DOWN FOR AUTOSIZING)
                                ;PROGRAM WILL TYPE "MAP OF TR79 STATUS" TO INDICATE
                                ;CSR ADDRESS, VECTOR, AND BR LEVEL
                                ;PROGRAM WILL TYPE "RUNNING" TO INDICATE THAT TESTING HAS STARTED
                                ;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
                                ;AND THEN RESUME TESTING


                        ;SWITCH REGISTER OPTIONS
                        ;----------------------

  100000                SW15=100000             ;=1,HALT ON ERROR
  040000                SW14=40000              ;=1,LOOP ON CURRENT TEST GROUP
  020000                SW13=20000              ;=1,INHIBIT ERROR TYPEOUT
  010000                SW12=10000              ;=1,BELL ON ERROR.
  004000                SW11=4000
  002000                SW10=2000               ;=1,ESCAPE TO NEXT TEST ON ERROR
  001000                SW09=1000               ;=1,LOOP WITH CURRENT DATA
  000400                SW08=400                ;=1,LOOP ON ERROR
                                                ;NOTE...FOR TEST 4&5 LOOPS BACK TO BEGINING.
  000200                SW07=200
  000100                SW06=100                ;=1, INHIBIT AUTOSIZING
  000040                SW05=40                 ;=0, INHIBIT TEST 5
                                                ;=1, ENABLE TEST 5
  000020                SW04=20
  000010                SW03=10                 ;=0, INHIBITS TESTS 1 AND 6.
                                                ;(IE, NO MANUAL INTERVENTION REQUIRED)
                                                ;=1, ENABLES TESTS 1 AND 6.

  000004                SW02=4                  ;=1, LOCK ON TEST SELECT
  000002                SW01=2                  ;=1, RESTART PROGRAM AT SELECTED TEST
  000001                SW00=1                  ;=1, SELECT DEVICE ADDRESS, VECTOR, ETC.
```

```
                        ;SWITCH #15 - WHEN SET, THE DIAGNOSTIC WILL HALT WHEN AN
                        ;       ERROR IS DETECTED.  THE LOCATION AT WHICH THE ERROR
                        ;       WAS DETECTED WILL BE DISPLAYED IN THE CONSOLE DATA
                        ;       LIGHTS.
                        ;
                        ;       IF NOT SET, THE DIAGNOSTIC WILL CONTINUE TESTING
                        ;       AS PER SWITCHES #8 AND #10.
                        ;
                        ;SWITCH #14 - WHEN SET THE DIAGNOSTIC WILL LOOP ON THE
                        ;       CURRENT TEST.  IF NOT SET, THE DIAGNOSTIC WILL FALL
                        ;       THROUGH TO THE NEXT TEST UPON COMPLETION OF THE
                        ;       CURRENT TEST.
                        ;
                        ;SWITCH #13 - WHEN SET, ERROR REPORTS WILL NOT BE
                        ;       PRINTED AT THE TTY WHEN ERRORS ARE DETECTED.
                        ;       IF NOT SET, ERROR REPORTS WILL BE PRINTED.
                        ;
                        ;       MESSAGES THAT MAY FOLLOW ERROR REPORTS
                        ;       ARE NOT INHIBITED BY THIS SWITCH.
                        ;
                        ;SWITCH #12 - WHEN SET, IF AN ERROR IS DETECTED, THE
                        ;       TTY BELL WILL RING.
                        ;
                        ;       IF NOT SET, THE TTY BELL WILL NOT RING WHEN
                        ;       AN ERROR IS DETECTED.
                        ;
                        ;SWITCH #10 - IF THIS SWITCH IS SET WHEN AN ERROR IS
                        ;       DETECTED, AFTER THE ERROR IS REPORTED, TESTING
                        ;       WILL CONTINUE, STARTING WITH THE NEXT TEST.
                        ;
                        ;       IF SWITCH #8 IS SET, THIS SWITCH HAS NO EFFECT.
                        ;
                        ;       IF SWITCH #8 IS NOT SET AND THIS SWITCH IS NOT
                        ;       SET, TESTING RESUMES AT THE POINT IN THE DIAGNOSTIC
                        ;       FOLLOWING THE LOCATION AT WHICH THE ERROR WAS DETECTED.
                        ;
                        ;SWITCH #9 - IF SET, THE DIAGNOSTIC USES THE CURRENT PARAMETERS
                        ;       TO CONTINUE TESTING.
                        ;
                        ;SWITCH #8 - IF THIS SWITCH IS SET, WHEN AN ERROR IS
                        ;       DETECTED, TESTING WILL RESUME AT THE BEGINNING OF
                        ;       THE TEST IN WHICH THE ERROR WAS DETECTED. (I.E. LOOP
                        ;       ON ERROR.) SEE SWITCH #10.
                        ;
                        ;SWITCH #6 - WHEN SET, THIS SWITCH INHIBITS AUTOSIZING.
                        ;       WHEN CLEAR, THE DEVICE WILL AUTOSIZE.
```

```
                        ;SWITCH #5 - WHEN SET, TEST GROUP #5 WILL BE EXECUTED.
                        ;       WHEN CLEAR, TEST GROUP #5 WILL NOT BE EXECUTED.


                        ;SWITCH #3 - WHEN NOT SET, THE TEST REQUIRING MANUAL
                        ;       INTERVENTION, TEST #1, WILL NOT BE EXECUTED.  IF
                        ;       SET, TEST #1 WILL BE EXECUTED.


                        ;SWITCH #2 - IF SET, THE DIAGNOSTIC WILL LOOP ON THE TEST
                        ;       SELECTED.  USED WITH SWITCH #1.


                        ;SWITCH #1 - WHEN SET, THIS SWITCH ALLOWS THE
                        ;       OPERATOR TO SELECT THE STARTING TEST, VIA A
                        ;       DIALOG AT THE TTY.


                        ;SWITCH #0 - WHEN SET, THIS SWITCH ALLOWS THE OPERATOR
                        ;       TO SPECIFY THE DEVICE ADDRESSES AND VECTORS
                        ;       IF THEY ARE DIFFERENT FROM THE STANDARD ONES.
```

```
                        ;REGISTER DEFINITIONS
                        ;--------------------

000000                  R0=%0           ;GENERAL REGISTER
000001                  R1=%1           ;GENERAL REGISTER
000002                  R2=%2           ;GENERAL REGISTER
000003                  R3=%3           ;GENERAL REGISTER
000004                  R4=%4           ;GENERAL REGISTER
000005                  R5=%5           ;GENERAL REGISTER
000006                  SP=%6           ;PROCESSOR STACK POINTER
000007                  PC=%7           ;PROGRAM COUNTER

                        ;LOCATION EQUIVALENCES
                        ;---------------------

177776                  PS=177776       ;PROCESSOR STATUS WORD
001200                  STACK=1200      ;START OF PROCESSOR STACK

                        ;INSTRUCTION DEFINITIONS
                        ;----------------------

005746                  PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD
005726                  POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD
010046                  PUSHR0=10046    ;SAVE R0 ON STACK
012600                  POPR0=12600     ;RESTORE R0 FROM STACK
024646                  PUSH2SP=24646   ;DECREMENT STACK TWICE
022626                  POP2SP=22626    ;INCREMENT STACK TWICE
                        .EQUIV  EMT,ERROR       ;BASIC DEFINITION OF ERROR CALL

                        ;BIT DEFINITIONS
                        ;---------------

100000                  BIT15=!100000
040000                  BIT14=40000
020000                  BIT13=20000
010000                  BIT12=10000
004000                  BIT11=4000
002000                  BIT10=2000
001000                  BIT9=1000
000400                  BIT8=400
000200                  BIT7=200
000100                  BIT6=100
000040                  BIT5=40
000020                  BIT4=20
000010                  BIT3=10
000004                  BIT2=4
000002                  BIT1=2
000001                  BIT0=1

                        ;PROCESSOR PRIORITY LEVELS
                        ;------------------------

000340                  LEVEL7=340
000300                  LEVEL6=300
```

```
                        LEVEL5=240
                        LEVEL4=200
                        LEVEL3=140
                        LEVEL2=100
                        LEVEL1=040
                        LEVEL0=000

                                ;TR79 CONTROL AND STATUS REGISTER DEFINITIONS
                                ;(TRCR) BIT DEFINITIONS
                                ;-----------------------------

                                        ;TAPE TEST EQUALS
                                                        ;UNIT   COMMAND
                        ILC00   =0                      ;0      ILLEGAL COMMAND
                        WRITE   =402                    ;1      WRITE FORWARD
                        READ    =4                      ;0      READ FORWARD
                        ILC03   =406                    ;1      ILLEGAL COMMAND
                        SPACER  =1410                   ;3      SPACE REVERSE 1 RECORD
                        ILC05   =1012                   ;2      ILLEGAL COMMAND
                        ILC06   =1414                   ;3      ILLEGAL COMMAND
                        ERASE   =16                     ;0      ERASE WORD COUNT FORWARD
                        REWIND  =1020                   ;1      REWIND TAPE
                        ILC11   =22                     ;0      ILLEGAL COMMAND
                        ILC12   =424                    ;1      ILLEGAL COMMAND
                        PWRCLR=BIT11                    ;DEFINITION OF POWER CLEAR BIT
                        GOEOT   =1026                   ;2      FAST FORWARD TO EOT
                        ILC14   =1030                   ;2      ILLEGAL COMMAND
                        WIDB    =432                    ;1      WRITE I.D. BLOCK
                        WEOF    =1034                   ;2      WRITE END OF FILE
                        OFLINE  =1436                   ;3      TAKE UNIT OFF UNIT

                        TCR=    164000                  ;DEFAULT CONTROL REGISTER ADDRESS
                        TSR=    164002                  ;DEFAULT STATUS REGISTER ADDRESS
                        TWC=    164004                  ;DEFAULT WORD COUNT ADDRESS
                        TBA=    164006                  ;DEFAULT BUS ADDRESS ADDRESS
                        TVA=    170                     ;DEFAULT TAPE VECTOR ADDRESS
                        TSA=    172                     ;DEFAULT TAPE STATUS ADDRESS
```

```
                                ;:*****************************************************************
                                ;-----------------------------------------------------------------
                                        ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
                                        ;THE STANDARD "TRAP CATCHER" IS PLACED
                                        ;BETWEEN ADDRESS 0 TO ADDRESS 776.
                                        ;IT LOOKS LIKE "PC+2 HALT".
                                ;-----------------------------------------------------------------
                                ;:*****************************************************************

        000000                  .=0
                                        ;STANDARD INTERRUPT VECTORS
                                        ;-------------------------

        000024                  .=24
000024  005230                          .PFAIL                  ;POWER FAIL HANDLER
000026  000340                          340                     ;SERVICE AT LEVEL 7
000030  004642                          .EROR                   ;ERROR HANDLER
000032  000340                          340                     ;SERVICE AT LEVEL 7
000034  004576                          .TRPSRV                 ;GENERAL HANDLER DISPATCH SERVICE
000036  000340                          340                     ;SERVICE AT LEVEL 7
        000040                  .=40
000040  000000                          0                       ;SAVE FOR ACT-11 OR XXDP
000042  000000                          0                       ;RETURN ADDRESS IF UNDER ACT-11 OR XXDP
000044  000000                          0                       ;SAVE FOR ACT-11 OR XXDP
000046  003470                          $ENDAD                  ;FOR USE WITH ACT-11 OR XXDP
        000052                  .=52
000052  000000                          0                       ;ACT-11 PROGRAM CHARACTERISTICS

        000174                  .=174
000174  000000                  DISPREG:0                       ;SOFTWARE DISPLAY REGISTER FOR SMALL 11S
000176  000000                  SWREG:  0                       ;SOWTARE SWITCH REGISTER FOR SMALL 11S
        000200                  .=200
000200  000137  001510                  JMP     .START          ;GO TO START OF PROGRAM


        001000                  .=1000
001000  005377  041501  034455  MTITLE: .ASCIZ  <377><12>/AC-9426B-MC /<377>/TR79F CHECKOUT TESTS/<377>
(2)
        001200                  .=1200
001200  177570                  DISPLAY:177570
001202  177570                  SWR:    177570
                                        ;INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
                                        ;--------------------------------------------------


001204  177560                  TKCSR:  177560                  ;TELETYPE KEYBOARD CONTROL REGISTER
001206  177562                  TKDBR:  177562                  ;TELETYPE KEYBOARD DATA BUFFER
001210  177564                  TPCSR:  177564                  ;TELEPRINTER CONTROL REGISTER
001212  177566                  TPDBR:  177566                  ;TELEPRINTER DATA BUFFER

                                        ;PROGRAM CONTROL PARAMETERS
                                        ;--------------------------


001214  000000                  RETURN: 0                       ;SCOPE ADDRESS FOR LOOP ON TEST
001216  000000                  NEXT:   0                       ;ADDRESS OF NEXT TEST TO BE EXECUTED
001220  000000                  LOCK:   0                       ;ADDRESS FOR LOCK ON CURRENT DATA
001222  000001                  ICOUNT: 1                       ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
```

```
   456  001224  000000                LPCNT:  0                       ;NUMBER OF ITEREATIONS COMPLETED
   457  001226  000000                TSTNO:  0                       ;NUMBER OF TEST IN PROGRESS
   458  001230  000000                PASCNT: 0                       ;NUMBER OF PASSES COMPLETED
   459  001232  000000                ERRCNT: 0                       ;TOTAL NUMBER OF ERRORS
   460  001234  000000                LSTERR: 0                       ;PC OF LAST ERROR CALL
   461  001236  000000                SAVLIN: .WORD           ;PROGRAM PARAMETER
   462
   463                                ;PROGRAM VARIABLES
   464                                ;-----------------
   465
   466  001240  000010                HOLD:   10                      ;DEFAULT DELAY TIME FOR 11/70 PROCESSOR
   467  001242  000000                TRBASE: 0                       ;BASE ADDRESS OF A GIVEN TR79 UNDER TEST
   468  001244  000000                REGIST: 0                       ;DEVICE ADDRESS STORAGE LOCATION
   469  001246  000000                STAT:   0                       ;TR STATUS WORD STORAGE
   470  001250  000000                TSTPTR: 0
   471  001252  000000                ERR:    0
   472  001254  000000                XPC:    0
   473  001256  000000                XSR:    0
   474  001260  000000                COR:    0
   475  001262  000000                ACT:    0
   476  001264  000000                WRTFLG: 0
   477  001266  000000                LOC:    0
   478  001270  000215                        CR:     215     ;TTY CARRIAGE RETURN
   479  001272  000212                        LF:     212     ;TTY LINE FEED
   480  001274  000377                        RO:     377     ;TTY RUBOUT
   481  001276  000000                        XTEST:  0
   482  001300  000000                        ZERO:   0
   483  001302  000000                TC:     0                       ;CONTROL STORAGE
   484  001304  000000                TEMP1:  0                       ;TEMPORARY STORAGE
   485  001306  000000                TEMP2:  0                       ;TEMPORARY STORAGE
   486  001310  000000                TEMP3:  0                       ;TEMPORARY STORAGE
   487  001312  000000                TEMP4:  0                       ;TEMPORARY STORAGE
   488  001314  000000                TEMP5:  0                       ;TEMPORARY STORAGE
   489  001316  000000                SAVR0:  0                       ;R0 STORAGE
   490  001320  000000                SAVR1:  0                       ;R1 STORAGE
   491  001322  000000                SAVR2:  0                       ;R2 STORAGE
   492  001324  000000                SAVR3:  0                       ;R3 STORAGE
   493  001326  000000                SAVR4:  0                       ;R4 STORAGE
   494  001330  000000                SAVR5:  0                       ;R5 STORAGE
   495  001332  000000                SAVSP:  0                       ;STACK POINTER STORAGE
   496  001334  000000                SAVPC:  0                       ;PROGRAM COUNTER STORAGE
   497  001336  000000                SAVE0:  0
   498  001340  000000                SAVE1:  0
   499                                .EVEN
   500  001342  001500                ACTIVE: TR.MAP                  ;TABLE POINTER.
```

```
                                        ;PROGRAM CONTROL FLAGS
                                        ;---------------------

001344      000                 INIFLG: .BYTE   0               ;PROGRAM INITIALIZATION FLAG
001345      000                 ERRFLG: .BYTE   0               ;ERROR OCCURRED FLAG
001346      000                 LOKFLG: .BYTE   0               ;LOCK ON CURRENT TEST FLAG
001347      000                 QV.FLG: .BYTE   0               ;QUICK VERIFY FLAG.
001350      000                 MNTFLG: .BYTE   0               ;MAINTENANCE BIT SET FLAG
                                                                ;ON FIRST PASS OF EACH TR79 ITERATIONS WILL BE SUPPRESSE
        001352                  .EVEN

                                        ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
                                        ;POINTERS TO SUBROUTINES CAN BE FOUND
                                        ;IN THE TABLE IMMEDIATELY FOLLOWING THE DEFINITIONS

                                ;:*****************************************************************
                                ;-----------------------------------------------------------------
001352                          .TRPTAB:
        104400                  SCOPE=TRAP+0            ;CALL TO SCOPE LOOP AND ITERATION HANDLER
001352  003544                          .SCOPE
        104401                  SCOP1=TRAP+1            ;CALL TO LOOP ON CURRENT DATA HANDLER
001354  003662                          .SCOP1
        104402                  TYPE=TRAP+2             ;CALL TO TELETYPE OUTPUT ROUTINE
001356  003706                          .TYPE
        104403                  TYPEF=TRAP+3            ;CALL TO FAILURE MESSAGE OUTPUT ROUTINE
001360  003752                          .TYPEF
        104404                  TYPEL=TRAP+4            ;CALL TO REPEAT MESSAGE OUTPUT ROUTINE
001362  003764                          .TYPEL
        104405                  INSTR=TRAP+5            ;CALL TO ASCII STRING INPUT ROUTINE
001364  003776                          .INSTR
        104406                  INSTER=TRAP+6           ;CALL TO INPUT ERROR HANDLER
001366  004102                          .INSTER
        104407                  PARAM=TRAP+7            ;CALL TO NUMERICAL DATA INPUT ROUTINE
001370  004122                          .PARAM
        104410                  SAVOS=TRAP+10           ;CALL TO REGISTER SAVE ROUTINE
001372  004322                          .SAVOS
        104411                  RESOS=TRAP+11           ;CALL TO REGISTER RESTORE ROUTINE
001374  004362                          .RESOS
        104412                  CONVRT=TRAP+12          ;CALL TO DATA OUTPUT ROUTINE
001376  004414                          .CONVRT
        104413                  CNVRT=TRAP+13           ;CALL TO DATA OUTPUT ROUNTINE WITHOUT CR/LF.
001400  004420                          .CNVRT

                                ;-----------------------------------------------------------------
                                ;:*****************************************************************
```

```
                                        ;TR79 VECTOR AND REGISTER INDIRECT POINTERS
                                        ;WORKING AREA

001402  164000                  TRCR:   164000  ;R/W
001404  164001                  HTRCR:  164001  ;R/W
001406  164002                  TRSR:   164002  ;READ ONLY
001410  164003                  HTRSR:  164003  ;READ ONLY
001412  164004                  TRWC:   164004  ;R/W
001414  164005                  HTRWC:  164005  ;R/W
001416  164006                  TRBA:   164006  ;READ ONLY
                                        ;DEFAULT TR VECTORS
001420  000170                  TRVCT:  170     ;REC INTR VECTOR
001422  000172                  TRRIS:  172     ;REC INTR STATUS
```

```
                                        ;TR79 STATUS TABLE AND ADDRESS ASSIGNMENTS
                                        ;----------------- ----------------------

            001500                      .=1500
001500      001500                      TR.MAP:

001500      000001                      TRCR0:  .BLKW   1       ;CONTROL  REGISTER FOR TR79 NUMBER 0
001502      000001                      TRVC0:  .BLKW   1       ;BASE VECTOR  FOR TR79 NUMBER 0
001504      000001                      TRLV0:  .BLKW   1       ;PRIORITY LEVEL

001506      000000                      TR.END: 000000
```

```
                                        ;PROGRAM INITIALIZATION
                                        ;LOCK OUT INTERRUPTS
                                        ;SET UP PROCESSOR STACK
                                        ;SET UP POWER FAIL VECTOR
                                        ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
                                        ;TYPE TITLE MESSAGE

001510                          .START:
001510  012706  001200                  MOV     #STACK,SP               ;SET UP STACK
001514  012737  000300  177776          MOV     #LEVEL6,PS              ;LOCK OUT INTERRUPTS
001522  012737  005230  000024          MOV     #.PFAIL,@#24    ;SET UP POWER FAIL VECTOR
001530  005037  001230                  CLR     PASCNT                  ;CLEAR PASS COUNT
001534  105037  001345                  CLRB    ERRFLG                  ;CLEAR ERROR FLAG
001540  105037  001347                  CLRB    QV.FLG                  ;ZERO QUICK VERIFY FLAG
001544  012737  001500  001342          MOV     #TR.MAP,ACTIVE  ;GET MAP POINTER.
001552  005037  001232                  CLR     ERRCNT                  ;CLEAR ERROR COUNT
001556  005037  001234                  CLR     LSTERR                  ;CLEAR LAST ERROR POINTER
001562  012737  000001  001226          MOV     #1,TSTNO                ;SET UP FOR TEST 1
001570  012737  001510  001214          MOV     #.START,RETURN  ;SET UP FOR POWER FAIL BEFORE
                                                                        ;TESTING STARTS
                                        ;SET UP FOR SMALL 11 SWITCH REGISTER COMPATIBILITY
001576  013746  000006                  MOV     6,-(SP)                 ;SAVE BUS ERROR PS
001602  013746  000004                  MOV     4,-(SP)                 ;SAVE BUS ERROR PC
001606  012737  001626  000004          MOV     #22$,4                  ;SET UP TO TRAP TO THIS ROUTINE
001614  005037  177776                  CLR     PS                              ;ALLOW INTERRUPTS
001620  005777  177356                  TST     @SWR                    ;CAN 177570 BE REFERENCED?
001624  000407                          BR      21$                     ;IF YES,SKIP AROUND THE SETUP
001626  012737  000176  001202  22$:    MOV     #SWREG,SWR              ;IF NO,TRAP COMES HERE.POINT TO SOFTWARE SWR
001634  012737  000174  001200          MOV     #DISPREG,DISPLAY ;POINT TO SOFTWARE DISPLAY REGISTER
001642  022626                          POP2SP                  ;REMOVE THE TRAP FROM THE STACK
001644  012637  000004          21$:    MOV     (SP)+,4                 ;RESTORE THE BUS ERROR VECTOR
001650  012637  000006                  MOV     (SP)+,6
001654  022737  003470  000042          CMP     #$ENDAD,        @#42    ;UNDER ACT11?
001662  001402                          BEQ     31$                     ;YES - SKIP TITLE PRINTOUT
001664  104402  001000                  TYPE    ,MTITLE                 ;PRINT THE DIAGNOSTIC'S NAME
001670  005737  000042          31$:    TST     42                      ;UNDER ACT11 OR XXDP?
001674  001004                          BNE     11$                     ;YES - USE DEFAULT PARAMETERS
001676  032777  000001  177276  29$:    BIT     #SW00,@SWR              ;RESELECT ?
001704  001013                          BNE     32$                     ;IF YES, GO SET UP THE INFORMATION
001706  012737  164000  001500  11$:    MOV     #164000,TRCR0   ;SET DEFAULT PARAMETERS
001714  012737  000170  001502          MOV     #170,TRVC0
001722  012737  000004  001504          MOV     #4,TRLV0
001730  000137  002146                  JMP     66$                     ;IF NO, SKIP THE INTERROGATION
001734  012700  001500          32$:    MOV     #TR.MAP,R0              ;POINT TO THE BEGINNING OF THE MAP TABLE
001740  005020                  68$:    CLR     (R0)+                   ;CLEAR A TABLE LOCATION
001742  020027  001506                  CMP     R0,#TR.END              ;HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
001746  001374                          BNE     68$                     ;IF NOT ,CLEAR THE NEXT LOCATION IN THE TABLE

                                        ;THE FOLLOWING ARE PARAMETERS USED TO FILL IN THE MAP
                                        ;TABLE AND SET UP THE DIAGNOSTIC.

                                                ;GET THE BASE ADDRESS OF THE TR79'S

001750                          33$:
001750  104405                          INSTR                           ;CALL THE STRING INPUT ROUTINE
```

```
                                                69$                     ;POINTER TO MESSAGE TO BE PRINTED
                                                PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
                                                160000                  ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
                                                175500                  ;HIGHEST LEGITIMATE VALUE OF EXPECTED REPONSE
                                                TRCRD                   ;POINTER TO MAP LOCATION TO BE FILLED
                                                .BYTE   7               ;MASK OF INVALID BITS FOR THIS PARAMETER
                                                .BYTE   1               ;NUMBER OF PARAMETERS TO STORE

                                                ;GET THE BASE VECTOR ADDRESS

                                        34$:
                                                INSTR                   ;CALL THE STRING INPUT ROUTINE
                                                70$                     ;POINTER TO MESSAGE TO BE PRINTED
                                                PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
                                                170                     ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
                                                776                     ;HIGHEST LEGITIMATE VALUE OF EXPECTED REPONSE
                                                TRVCD                   ;POINTER TO MAP LOCATION TO BE FILLED
                                                .BYTE   3               ;MASK OF INVALID BITS FOR THIS PARAMETER
                                                .BYTE   1               ;NUMBER OF PARAMETERS TO STORE

                                                ;GET THE BUS REQUEST LEVEL

                                                INSTR                   ;CALL THE STRING INPUT ROUTINE
                                                72$                     ;POINTER TO MESSAGE TO BE PRINTED
                                                PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
                                                4                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
                                                7                       ;HIGHEST LEGITIMATE VALUE OF EXPECTED REPONSE
                                                TRLVD                   ;POINTER TO MAP LOCATION TO BE FILLED
                                                .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
                                                .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
                                                JMP     66$             ;GO CONTINUE PROCESSING

                                        69$:    .ASCIZ  <377>/CONTROLLER ADDRESS  /
                                        70$:    .ASCIZ  <377>/1ST VECTOR ADDRESS  /
                                        72$:    .ASCII  <377>/NOTE: ONE TR79 ONLY  /<377>
                                        71$:    .ASCIZ  <377>/BR LEVEL  /
                                                .EVEN
                                        66$:    BIT     #BIT6,  @SWR    ;INHIBIT AUTOSIZING?
                                                BEQ     1$              ;NO
                                                JMP     HEADER          ;YES-GO PRINTOUT MAP TABLE
                                        1$:     TYPE    ,MCRLF          ;
                                                TYPE    ,SIZE           ;TYPE "AUTOSIZING"
                                                TYPE    ,MCRLF          ;
                                                MOV     @#TR.MAP,       SAVE0   ;SAVE PRE-AUTOSIZE CSR ADDRESS
                                        CSRSIZ: MOV     #STACK, SP      ;RESET STACK
                                                CLR     TSTNO           ;CALL THIS "TEST 0"
                                                MOV     #CSRSIZ,        RETURN  ;SETUP FOR LOOP ON ERROR
                                                MOV     #22$,   NEXT    ;SETUP FOR SKIP ON ERROR
                                                JSR     PC,     TESTE   ;CHECK INT ENB BIT SETS
                                                JSR     PC,     TESTF   ;CHECK INT ENB BIT CLEARS
                                                JSR     PC,     TESTG   ;CHECK PWR CLR WORKS
                                                MOV     #99$,   @#4     ;SETUP FOR TIMEOUT
                                                MOV     #340,   @#6
                                                MOV     #160000,        R0      ;FIRST VALUE IN RANGE
                                        1$:     TST     (R0)            ;IS THIS THE ADDRESS?
                                                MOV     #6,     @#4     ;YES-RESTORE TRAP CATCHER
```

```
   680  002274  012737  000000  000006          MOV     #0,     @#6
   681  002302  010037  001340                  MOV     R0,     SAVE1   ;STORE AUTOSIZED VALUE
   682  002306  000421                          BR      20$
   683  002310  022626                  99$:    POP2SP                  ;TIMEOUT-RESTORE STACK
   684  002312  062700  000002                  ADD     #2,     R0      ;SETUP FOR NEXT ADDRESS
   685  002316  022700  175502                  CMP     #175502,        R0      ;END OF RANGE^
   686  002322  001401                          BEQ     2$              ;YES
   687  002324  000757                          BR      1$              ;NO
   688  002326  012737  000006  000004  2$:     MOV     #6,     @#4     ;RESTORE TRAP CATCHER
   689  002334  012737  000000  000006          MOV     #0,     @#6
   690  002342  104014                          ERROR   14              ;CSR NEVER SIZED-ERROR MSG
   691  002344  104402  010323                  TYPE    ,PRHLT
   692  002350  000000                          HALT
   693  002352  023737  001336  001340  20$:     CMP     SAVE0,  SAVE1   ;EXPECTED CSR ADDRESS?
   694  002360  001406                          BEQ     22$             ;YES
   695  002362  013737  001340  001500          MOV     SAVE1,  @#TR.MAP        ;NO-PUT SIZED VALUE INTO TABLE
   696  002370  104012                          ERROR   12              ;ERROR MSG
   697  002372  104403  010415                  TYPEF   ,CSRER2
   698  002376  013737  001502  001336  22$:     MOV     @#TR.MAP+2,     SAVE0   ;SAVE PRE-SIZE VECTOR
   699  002404  012737  002404  001214  VECSIZ: MOV     #VECSIZ,        RETURN  ;SETUP FOR LOOP ON ERROR
   700  002412  012737  002710  001216          MOV     #23$,   NEXT    ;SETUP FOR SKIP ON ERROR
   701  002420  012706  001200                  MOV     #STACK, SP      ;RESET STACK
   702  002424  012737  000340  000022          MOV     #340,   @#22
   703  002432  012737  002632  000020          MOV     #4$,    @#20    ;SETUP FOR IOT TRAP
   704  002440  012737  000340  000006          MOV     #340,   @#6
   705  002446  012737  002604  000004          MOV     #8$,    @#4     ;SETUP FOR TIMEOUT
   706  002454  012737  000172  000170          MOV     #172,   @#170   ;SETUP FOR INT TO THIS ADDRESS
   707  002462  012737  000004  000172          MOV     #4,     @#172
   708  002470  012700  000210                  MOV     #210,   R0      ;SETUP FOR INT TO ALL POSSIBLE ADDRESSES
   709  002474  012701  000212                  MOV     #212,   R1
   710  002500  010120                  10$:    MOV     R1,     (R0)+
   711  002502  012721  000004                  MOV     #4,     (R1)+
   712  002506  022021                          CMP     (R0)+,  (R1)+
   713  002510  020127  001000                  CMP     R1,     #1000
   714  002514  101771                          BLOS    10$
   715  002516  005037  177776                  CLR     PS                      ;MAKE CPU PRIORITY 0
   716  002522  005037  001340                  CLR     SAVE1
   717  002526  052777  004100  176744          BIS     #4100,  @TR.MAP ;ENABLE INTERRUPT
   718  002534  005000                          CLR     R0                      ;WAIT 900 MSEC
   719  002536  012701  177766                  MOV     #-12,   R1
   720  002542  005200                  1$:     INC     R0
   721  002544  001376                          BNE     1$
   722  002546  005201                          INC     R1
   723  002550  001374                          BNE     1$
   724  002552  042777  004100  176720          BIC     #4100,  @TR.MAP ;NO INTERRUPT-DISABLE INTERRUPT
   725  002560  012737  000006  000004          MOV     #6,     @#4     ;RESTORE TRAP CATCHER
   726  002566  012737  000000  000006          MOV     #0,     @#6
   727  002574  104013                          ERROR   13              ;ERROR MSG
   728  002576  104402  010323                  TYPE    ,PRHLT
   729  002602  000000                          HALT
   730  002604  012737  000006  000004  8$:     MOV     #6,     @#4     ;TIMEOUT-RESTORE TRAP CATCHER
   731  002612  012737  000000  000006          MOV     #0,     @#6
   732  002620  022626                          POP2SP                  ;RESTORE STACK
   733  002622  104015                          ERROR   15              ;ERROR MSG
   734  002624  104402  010323                  TYPE    ,PRHLT
   735  002630  000000                          HALT
```

```
002632  012737  000006  000004  4$:      MOV     #6,       @#4       ;RESTORE TRAP CATCHER
002640  012737  000000  000006           MOV     #0,       @#6
002646  011637  001340                   MOV     (SP),     SAVE1     ;STACK POINTS TO VECTOR ADDRESS PLUS 4
002652  162737  000004  001340           SUB     #4,       SAVE1
002660  022626                           POP2SP                      ;RESTORE STACK
002662  022626                           POP2SP
002664  023737  001336  001340           CMP     SAVE0,    SAVE1     ;EXPECTED VECTOR ADDRESS?
002672  001406                           BEQ     23$                 ;YES
002674  013737  001340  001502           MOV     SAVE1,    @#TR.MAP+2          ;NO-PUT SIZED VALUE INTO TABLE
002702  104012                           ERROR   12                  ;ERROR MSG
002704  104403  010433                   TYPEF   ,VECER3
002710  013737  001504  001336  23$:     MOV     @#TR.MAP+4,         SAVE0   ;SAVE PRE-SIZE BR LEVEL
002716  012737  000300  001304           MOV     #300,     TEMP1     ;STORE CPU PRIORITY TO START BR SIZING
002724  012737  000006  001340           MOV     #6,       SAVE1     ;#300 IS PRIORITY 6
002732  012737  002732  001214  BRSIZ:   MOV     #BRSIZ,   RETURN    ;SETUP FOR LOOP ON ERROR
002740  012737  003140  001216           MOV     #HEADER,            NEXT    ;SETUP FOR SKIP ON ERROR
002746  012706  001200                   MOV     #STACK,   SP        ;RESET STACK
002752  013737  001304  177776           MOV     TEMP1,    PS        ;SET CPU PRIORITY
002760  013702  001502                   MOV     @#TR.MAP+2,         R2      ;GET VECTOR ADDRESS
002764  012722  003100                   MOV     #34$,     (R2)+     ;SETUP FOR INT
002770  012712  000340                   MOV     #340,     (R2)      ;
002774  052777  004100  176476           BIS     #4100,    @TR.MAP   ;ENABLE INT
003002  005000                           CLR     R0                  ;WAIT 900 MSEC
003004  012701  177766                   MOV     #-12,     R1
003010  005200                  1$:      INC     R0
003012  001376                           BNE     1$
003014  005201                           INC     R1
003016  001374                           BNE     1$
003020  042777  004100  176452           BIC     #4100,    @TR.MAP   ;NO INT OCCURRED
003026  023737  001340  001336           CMP     SAVE1,    SAVE0     ;EXPECTED INT?
003034  002003                           BGE     2$                  ;NO
003036  104013                           ERROR   13                  ;YES-ERROR MSG
003040  104403  010454                   TYPEF   ,BRER1
003044  162737  000040  001304  2$:      SUB     #40,      TEMP1     ;DECREMENT CPU PRIORITY
003052  005337  001340                   DEC     SAVE1
003056  022737  000002  001340           CMP     #2,       SAVE1     ;IS CPU AT LEVEL 2?
003064  001402                           BEQ     11$                 ;YES
003066  000137  002732                   JMP     @#BRSIZ             ;NO-CONTINUE
003072  104402  010323          11$:     TYPE    ,PRHLT              ;IF NO INT AT CPU LEVEL 3-
003076  000000                           HALT                        ;HALT
003100  042777  004100  176372  34$:     BIC     #4100,    @TR.MAP   ;INT OCCURRED
003106  022626                           POP2SP                      ;RESTORE STACK
003110  013737  001340  001504           MOV     SAVE1,    @#TR.MAP+4          ;DEVICE PRIORITY-
003116  005237  001504                   INC     @#TR.MAP+4                    ;IS CPU'S AT INT PLUS 1
003122  023737  001340  001336           CMP     SAVE1,    SAVE0     ;EXPECTED INT?
003130  002403                           BLT     HEADER              ;YES
003132  104016                           ERROR   16                  ;NO-ERROR MSG
003134  104403  010454                   TYPEF   ,BRER1
003140  012737  000001  001226  HEADER:  MOV     #1,       TSTNO
003146  104402  005320                   TYPE    ,MCRLF
003152  104402  005631          16$:     TYPE    ,XHEAD              ;TYPE MAP HEADER
003156  012737  001500  001304           MOV     #1500,    TEMP1
003164  012700  001500                   MOV     #TR.MAP,RO          ;SET POINTER
003170  012037  001306          5$:      MOV     (R0)+,TEMP2         ;SET DATA
003174  001406                           BEQ     MON                 ;ALL DONE WITH DATA
003176  104412                           CONVRT                      ;CALL THE OCTAL TO ASCII CONVERSION ROUTINE
```

```
792  003200  010502                        XSTATQ                  ;CONVERT THE DATA AT THIS ADDRESS
793  003202  062737  000002  001304        ADD     #2,     TEMP1
794  003210  000767                        BR      5$
795  003212  005037  001304        MON:    CLR     TEMP1
796  003216  005737  000042                TST     @#42            ;IS PROGRAM RUNNING UNDER MONITOR
797  003222  001000                        BNE     3$              ;YES
798  003224  012700  000210        3$:     MOV     #210,R0         ;PREPARE TO CLEAR THE REMAINING
799  003230  012701  000212                MOV     #212,R1         ;VECTOR AREA. 210-776
800  003234  010120                4$:     MOV     R1,(R0)+        ;START PUTTING "PC+2 - HALT"
801  003236  005021                        CLR     (R1)+           ;IN VECTOR AREA.
802  003240  022021                        CMP     (R0)+,(R1)+     ;POP POINTERS
803  003242  022700  001000                CMP     #1000,R0        ;ALL DONE??
804  003246  001372                        BNE     4$              ;BR IF NO.
805  003250  005037  000172                CLR     @#172
806
807                                  ;TEST START AND RESTART
808                                  ;----------------------
809
810  003254  012737  000340  177776  .BEGIN: MOV     #340,PS         ;LOCK OUT INTERRUPTS
811  003262  012706  001200                MOV     #STACK,SP       ;SET UP STACK
812  003266  005737  000042                TST     @#42            ;IS PROGRAM UNDER MONITOR CONTROL
813  003272  001023                        BNE     2$              ;BR IF YES
814  003274  032777  000004  175700        BIT     #BIT2,@SWR      ;CHECK FOR LOCK ON TEST
815  003302  001411                        BEQ     1$              ;BR IF NO LOCK DESIRED.
816  003304  104402  005523                TYPE    ,MLOCK          ;TYPE LOCK SELECTED.
817  003310  012737  000240  003556        MOV     #NOP,TTST       ;ADJUST SCOPE ROUTINE.
818  003316  012737  000240  003560        MOV     #NOP,TTST+2     ;SET UP TO LOCK
819  003324  000406                        BR      2$              ;CONTINUE ALONG.
820  003326  013737  003656  003556  1$:    MOV     BRW,TTST        ;PREPARE NORMAL SCOPE ROUTINE
821  003334  013737  003660  003560        MOV     BRX,TTST+2      ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
822  003342  012737  010664  001214  2$:    MOV     #CYCLE,RETURN   ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
823  003350  104402  005320                TYPE    ,MCRLF
824  003354  104402  005404                TYPE    ,MR             ;TYPE "RUNNING"
825  003360  000177  175630                JMP     @RETURN         ;START TESTING
```

```
                                              ;END OF PASS
                                              ;TYPE NAME OF TEST
                                              ;UPDATE PASS COUNT
                                              ;CHECK FOR EXIT TO ACT-11
                                              ;RESTART TEST

003364                               .EOP:
003364  005037  001234                        CLR     LSTERR           ;CLEAR LAST ERROR PC
003370  105037  001345                        CLRB    ERRFLG           ;CLEAR ERROR FLAG
003374  005237  001230                        INC     PASCNT           ;UPDATE PASS COUNT
003400  013777  001230  175572                MOV     PASCNT,@DISPLAY  ;DISPLAY PASS COUNT
003406  104402  005361                        TYPE    ,MEPASS          ;TYPE END PASS
003412  104402  005552                        TYPE    ,MCSRX           ;TYPE CSR
003416  104413  003514                        CNVRT   ,XCSR            ;SHOW IT
003422  104402  005560                        TYPE    ,MVECX           ;TYPE VECTOR
003426  104413  003522                        CNVRT   ,XVEC            ;SHOW IT
003432  104402  005566                        TYPE    ,MPASSX          ;TYPE PASSES
003436  104413  003530                        CNVRT   ,XPASS           ;SHOW IT
003442  104402  005577                        TYPE    ,MERRX           ;TYPE ERRORS
003446  104413  003536                        CNVRT   ,XERR            ;SHOW IT
003452  112737  000377  001347                MOVB    #377,QV.FLG      ;SET THE QUICK VERIFY FLAG.
003460  013701  000042                        MOV     @#42,R1          ;CHECK FOR ACT-11 OR DDP
003464  001406                                BEQ     RESTRT           ;IF NOT, CONTINUE TESTING
003466  000005                                RESET                    ;STOP THE SHOW--CLEAR THE WORLD
003470                               $ENDAD:
003470  004711                                JSR     PC,(R1)
003472  000240                                NOP
003474  000240                                NOP
003476  000240                                NOP
003500  000240                                NOP
003502  012737  010664  001214       RESTRT:  MOV     #CYCLE,RETURN
003510  000137  010664                        JMP     CYCLE
003514  000001                       XCSR:    1
003516     006     002                        .BYTE   6,2
003520  001402                                TRCR
003522  000001                       XVEC:    1
003524     003     002                        .BYTE   3,2
003526  001420                                TRVCT
003530  000001                       XPASS:   1
003532     006     002                        .BYTE   6,2
003534  001230                                PASCNT
003536  000001                       XERR:    1
003540     006     002                        .BYTE   6,2
003542  001232                                ERRCNT

                                              ;SCOPE LOOP AND ITERATION HANDLER
                                              ;-------------------------------

003544  005037  001234               .SCOPE:  CLR     LSTERR           ;CLEAR LAST ERROR PC.
003550  032777  040000  175424                BIT     #BIT14,@SWR      ;"LOOP ON THIS TEST"?
003556  001407                       TTST:    BEQ     1$               ;BR IF NO.   (IF LOCK SW01=1;  THIS LOC =240)
003560  000433                                BR      3$               ;GOTO 3$     (IF LOCK SW01=1;  THIS LOC =240)
003562  105777  175416                        TSTB    @TKCSR           ;KEYBOARD DONE?
003566  100030                                BPL     3$               ;BR IF NO. (LOCK: HIT KEY TO GOTO NEXT TEST)
003570  017716  175412                        MOV     @TKDBR,(SP)      ;CLEAR DONE BIT
003574  000411                                BR      2$               ;CONTINUE
```

```
                                1$:     TSTB    @V.FLG          ;HAVE PASSES BEEN COMPLETED?
                                        BEQ     2$              ;BR IF QUICK PASS.
                                        INC     LPCNT           ;UPDATE ITERATION COUNTER
                                        CMP     LPCNT,ICOUNT    ;ARE ALL ITERATIONS DONE??
                                        BLOS    3$              ;BR IF NOT YET
                                2$:     CLRB    EPRFLG          ;PREPARE FOR NEW TEST
                                        CLR     LPCNT           ;START ICOUNTER AT 0
                                        CLR     LOCK
                                        MOV     #0,ICOUNT       ;RESET ITERATIONS
                                        MOV     NEXT,RETURN     ;GET NEXT TEST
                                3$:     POP2SP                  ;FAKE AN "RTI"
                                4$:     JMP     @RETURN         ;GO DO THE TEST
                                BRW:    1407
                                BRX:    437

                                        ;CHECK FOR FREEZE ON CURRENT DATA
                                        ;--------------------------------

                        .SCOP1: BIT     #SW09,@SWR      ;IS SW09=1(SET)?
                                BEQ     1$              ;BR IF NOT SET.
                                TST     LOCK            ;IS THER A TIGHT LOOP SPECIFIED?
                                BEQ     1$              ;I NO, RETURN
                                MOV     LOCK,(SP)       ;IF YES, GOTO THE ADDRESS IN LOCK.
                        1$:     RTI                     ;GO BACK.

                                        ;TELETYPE OUTPUT ROUTINE
                                        ;-----------------------

                        .TYPE:  MOV     R5,-(SP)        ;SAVE R5 ON THE STACK.
                                MOV     @2(SP),R5       ;GET ADDRESS OF MESSAGE.
                                ADD     #2,2(SP)        ;POP OVER ADDRESS.
                        1$:     TSTB    (R5)
                                BPL     2$
                                TYPE    ,MCRLF
                        2$:     TSTB    @TPCSR          ;TTY READY?
                                BPL     2$              ;BR IF NO.
                                MOVB    (R5)+,@TPDBR    ;PRINT CURRENT CHAR.
                                BNE     1$              ;IF NOT ZERO KEEP PRINTING!
                        3$:     MOV     (SP)+,R5        ;END OF OUTPUT. RESTORE R5
                                RTI                     ;GO HOME

                                        ;AUXILLIARY TELETYPE OUTPUT ROUTINES
                                        ;-----------------------------------

                        .TYPEF: MOV     @(SP),LSTMSG    ;GET THE ADDRESS OF THE MESSAGE TO PRINT
                                ADD     #2,(SP) ;POINT TO THE NEXT LOCATION IN THE MAIN PROCEDURE
                        .TYPEL: TYPE                    ;BOTH ROUTINES TYPE HERE
                        LSTMSG: .WORD   0               ;POINTER TO MESSAGE TO PRINT
                                TYPE                    ;THIS PART PRINTS THE REMAINDER OF THE MESSAGE
                                MFAIL                   ;THIS IS THE 'FAILURE' PART OF IT
                        1$:     RTI                     ;RETURN TO THE MAIN PROCEDURE

                                        ;STRING INPUT ROUTINE
                                        ;--------------------

                        .INSTR: MOV     R3,-(SP)        ;SAVE R3 ON STACK
```

```
                                        MOV     R4,-(SP)        ;SAVE R4 ON STACK
                                        MOV     @4(SP),.MSG     ;GET THE ADDRESS OF THE MESSAGE TO BE PRINTED
                                        ADD     #2,4(SP)        ;POINT TO INSTRUCTION AFTER THE ADDRESS POINTER
                                .INST1: TYPE                    ;PRINT THE MESSAGE
                                .MSG:   0                       ;MESSAGE IS POINTED TO FROM HERE
                                        MOV     #INBUF,R4       ;POINT R4 TO THE INPUT BUFFER
                                        MOV     #7,R3           ;SET THE MAXIMUM NUMBER OF CHARACTERS ALLOWED
                                1$:     TSTB    @TKCSR          ;HAS A CHARACTER BEEN RECEIVED?
                                        BPL     1$              ;IF NO, KEEP WAITING FOR IT
                                        MOVB    @TKDBR,(R4)     ;IF YES, SAVE IT IN THE INPUT BUFFER
                                        BICB    #200,(R4)       ;KEEP ONLY THE 7-BIT ASCII INFORMATION
                                        CMPB    (R4)+,#15       ;IS THIS CHARACTER A LINE FEED?
                                        BEQ     INSTR2          ;IF SO, TERMINATE THE INPUT SEQUENCE
                                2$:     TSTB    @TPCSR          ;IF NOT, CHECK TO SEE IF THE CHARACTER CAN PRINT
                                        BPL     2$              ;IF WE CAN'T, WAIT UNTIL WE CAN
                                        MOV     @TKDBR,@TPDBR   ;ECHO THE CHARACTER BACK
                                        DEC     R3              ;REDUCE THE NUMBER OF CHARACTERS RECEIVED
                                        BNE     1$              ;IF WE DON'T HAVE 7, GO GET SOME MORE
                                        MOV     (SP)+,R4        ;IF WE HAVE 7, RESTORE R4
                                        MOV     (SP)+,R3        ;RESTORE R3
                                .INSTE: MOV     R3,-(SP)        ;SAVE R3 ON THE STACK
                                        MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
                                        TYPE    ,MQM            ;PRINT A QUESTION MARK... WHAT'S GOING ON?
                                        BR      .INST1          ;GO PRINT THE MESSAGE AGAIN
                                INSTR2: MOV     (SP)+,R4        ;RESTORE R4
                                        MOV     (SP)+,R3        ;RESTORE R3
                                        RTI                     ;RETURN TO THE MAIN PROCEDURE

                                        ;CONVERT ASCII STRING TO OCTAL
                                        ;-----------------------------

                                .PARAM: MOV     R5,-(SP)        ;SAVE R5 ON THE STACK
                                        MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
                                        MOV     4(SP),R5        ;GET THE SETUP INFORMATION POINTER
                                        MOV     (R5)+,LOLIM     ;SET THE LOW LIMIT FOR THE INPUT
                                        MOV     (R5)+,HILIM     ;SET THE HIGH LIMIT FOR THE INPUT
                                        MOV     (R5)+,DEVADR    ;SAVE THE ADDRESS WHERE THE RESULT WILL BE STORED
                                        MOVB    (R5)+,LOBITS    ;GET THE MASK OF THE INCORRECT BITS
                                        MOVB    (R5)+,ADRCNT    ;GET THE COUNT OF ITEMS TO BE STORED
                                        MOV     R5,4(SP)        ;POINT TO WHERE THE MAIN PROGRAM WILL RESUME
                                PARAM1: CLR     R5              ;INITIALIZE THE ASCII TO OCTAL RESULT WORD
                                        MOV     #INBUF,R4       ;POINT TO THE INPUT BUFFER
                                        CMPB    #15,(R4)        ;IS THIS CHARACTER A CARRIAGE RETURN?
                                        BEQ     PARERR          ;IF SO, PRINT THE MESSAGE AGAIN
                                1$:     CMPB    (R4),#60        ;IS THIS CHARACTER BELOW THE NUMERIC RANGE?
                                        BLT     PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN
                                        CMPB    (R4),#67        ;IS THIS CHARACTER ABOVE THE NUMERIC RANGE?
                                        BGT     PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN
                                        BICB    #60,(R4)        ;ISOLATE THE NUMBER THE CHARACTER REPRESENTS
                                        BISB    (R4)+,R5        ;CONCATENATE THESE BITS TO THE EXISTING STRING
                                        CMPB    #15,(R4)        ;IS THE NEXT CHARACTER A CARRIAGE RETURN?
                                        BEQ     LIMITS          ;IF SO, GO SEE IF THE NUMBER IS WITHIN LIMITS
                                        ASL     R5              ;CLEAR BIT POSITION 0, MOVE THE EXISTING STRING TO THE L
                                        ASL     R5              ;CLEAR POSITION 1, MOVE THE STRING TO THE LEFT AGAIN
                                        ASL     R5              ;MOVE THE STRING ONE MORE TIME TO MAKE ROOM FOR
                                                                ;NEXT THREE BITS
```

```
                                        BR      1$              ;GO GET THE NEXT CHARACTER
004234  000760
004236  104406                  PARERR: INSTER                  ;THERE WAS AN ERROR- GO PRINT THE MESSAGE AGAIN
004240  000750                          BR      PARAM1          ;TRY GETTING THE PARAMETERS AGAIN

                                        ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
                                        ;--------------------------------------

004242  020537  004314          LIMITS: CMP     R5,HILIM        ;DOES RESULT EXCEED ITS MAXIMUM CORRECT VALUE?
004246  101373                          BHI     PARERR          ;IF YES, GO PRINT THE MESSAGE AGAIN
004250  020537  004312                  CMP     R5,LOLIM        ;IS THE RESULT LOWER THAN ALLOWED?
004254  1037*0                          BLO     PARERR          ;IF YES, GOR PRINT THE MESSAGE AGAIN
004256  133705  004320                  BITB    LOBITS,R5       ;ARE ANY INCORRECT BITS SET IN THE RESULT?
004262  001365                          BNE     PARERR          ;IF SO, GO PRINT THE MESSAGE AGAIN

                                        ;STORE NUMBER AT SPECIFIED ADDRESS

004264  013704  004316          1$:     MOV     DEVADR,R4       ;POINT TO THE LOCATION WHERE THE RESULT WILL BE STORED
004270  010524                          MOV     R5,(R4)+        ;STORE THE RESULT
004272  062705  000002                  ADD     #2,R5           ;CALCULATE THE NEXT DATUM
004276  105337  004321                  DECB    ADRCNT          ;REDUCE COUNT OF STORED RESULTS. IS IT EXCEEDED?
004302  001372                          BNE     1$              ;IF NOT, GO STORE THE NEXT DATUM
004304  012604                          MOV     (SP)+,R4        ;RESTORE R4
004306  012605                          MOV     (SP)+,R5        ;RESTORE R5
004310  000002                          RTI                     ;RETURN TO THE MAIN PROGRAM

004312  000000          LOLIM:  0                               ;LOWEST ACCEPTABLE VALUE
004314  000000          HILIM:  0                               ;HIGHEST ACCEPTABLE
004316  000000          DEVADR: 0                               ;LOCATION WHERE RESULT WILL BE STORED
004320     000          LOBITS: .BYTE   0                       ;INCORRECT BITS MASK
004321     000          ADRCNT: .BYTE   0                       ;COUNT OF ITEMS TO BE STORED

                                        ;SAVE PC OF TEST THAT FAILED AND R0-R5
                                        ;-------------------------------------

004322  016637  000004  001334  .SAV05: MOV     4(SP),SAVPC     ;SAVE R7 (PC)

                                        ;SAVE R0-R5

004330  010537  001330          SV05:   MOV     R5,SAVR5        ;SAVE R5
004334  010437  001326                  MOV     R4,SAVR4        ;SAVE R4
004340  010337  001324                  MOV     R3,SAVR3        ;SAVE R3
004344  010237  001322                  MOV     R2,SAVR2        ;SAVE R2
004350  010137  001320                  MOV     R1,SAVR1        ;SAVE R1
004354  010037  001316                  MOV     R0,SAVR0        ;SAVE R0
004360  000002                          RTI                     ;LEAVE.

                                        ;RESTORE R0-R5

004362  013700  001316          .RES05: MOV     SAVR0,R0        ;RESTORE R0
004366  013701  001320                  MOV     SAVR1,R1        ;RESTORE R1
004372  013702  001322                  MOV     SAVR2,R2        ;RESTORE R2
004376  013703  001324                  MOV     SAVR3,R3        ;RESTORE R3
004402  013704  001326                  MOV     SAVR4,R4        ;RESTORE R4
004406  013705  001330                  MOV     SAVR5,R5        ;RESTORE R5
004412  000002                          RTI                     ;LEAVE
```

```
                                        ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
                                        ;-------------------------------------------------------
004414  104402  005320        .CONVR: TYPE    ,MCRLF          ;PRINT A CARRIAGE RETURN
004420  010046                .CNVRT: MOV     R0,-(SP)        ;SAVE R0
004422  010146                        MOV     R1,-(SP)        ;SAVE R1
004424  010346                        MOV     R3,-(SP)        ;SAVE R3
004426  010446                        MOV     R4,-(SP)        ;SAVE R4
004430  010546                        MOV     R5,-(SP)        ;SAVE R5
004432  017601  000012                MOV     @12(SP),R1      ;PLACE THE ADDRESS OF THE ARGUMENTS IN R1
004436  062766  000002  000012        ADD     #2,12(SP)       ;POINT TO WHERE THE MAIN PROGRAM WILL RESUME
004444  012137  004570                MOV     (R1)+,WRDCNT    ;GET THE NUMBER OF WORDS TO BE PRINTED
004450  112105                1$:     MOVB    (R1)+,R5        ;GET THE NUMBER OF CHARACTERS TO BE PRINTED
004452  112100                        MOVB    (R1)+,R0        ;GET THE NUMBER OF SPACES TO PRINT
004454  013104                        MOV     @(R1)+,R4       ;COPY THE WORD TO BE CONVERTED
004456  110537  004572                MOVB    R5,CHRCNT       ;COPY THE CHARACTER COUNT
004462  010403                3$:     MOV     R4,R3           ;COPY THE ARGUMENT WORD AGAIN
004464  042703  177770                BIC     #177770,R3      ;ISOLATE THREE BITS TO BE TREATED AS A CHARACTER
004470  062703  000060                ADD     #060,R3         ;MAKE AN ASCII CHARACTER OUT OF THEM
004474  110346                        MOVB    R3,-(SP)        ;SAVE THAT CHARACTER
004476  006004                        ROR     R4              ;MOVE THE NEXT THREE BITS INTO PLACE
004500  006204                        ASR     R4              ;MOVE THEM AGAIN
004502  006204                        ASR     R4              ;AND FINALLY A THIRD TIME
004504  005305                        DEC     R5              ;REDUCE THE CHARACTER COUNT.ARE ALL CHARACTERS
                                                                ;BUILT?
004506  001365                        BNE     3$              ;IF NO, GO BUILD THE NEXT ONE.
004510  012703  010622                MOV     #MDATA,R3       ;NOW POINT TO WHERE NUMBER WILL BE PRINTED FROM
004514  112623                4$:     MOVB    (SP)+,(R3)+     ;STORE THE CHARACTER, STARTING WITH THE MOST
004516  105337  004572                DECB    CHRCNT          ;REDUCE COUNT. ARE ALL CHARACTERS TRANSFERRED?
004522  001374                        BNE     4$              ;IF NO, GO TRANSFER ANOTHER
004524  105700                        TSTB    R0              ;ARE ANY SPACES TO BE PRINTED?
004526  001404                        BEQ     6$              ;IF NO, DON'T SET UP ANY
004530  112723  000040        5$:     MOVB    #040,(R3)+      ;ADD A SPACE TO THE OUTPUT BUFFER
004534  105300                        DECB    R0              ;REDUCE THE COUNT. SHOULD WE PRINT MORE?
004536  001374                        BNE     5$              ;IF YES, GO ADD ANOTHER SPACE
004540  105013                6$:     CLRB    (R3)            ;TERMINATE THE OUTPUT BUFFER WITH A ZERO
004542  104402  010622                TYPE    ,MDATA          ;PRINT THE STRING WE JUST BUILT
004546  005337  004570                DEC     WRDCNT          ;REDUCE THE WORD COUNT. ARE ANY MORE WORDS LEFT?
004552  001336                        BNE     1$              ;IF YES, GO CONVERT THEM
004554  012605                        MOV     (SP)+,R5        ;RESTORE R5
004556  012604                        MOV     (SP)+,R4        ;RESTORE R4
004560  012603                        MOV     (SP)+,R3        ;RESTORE R3
004562  012601                        MOV     (SP)+,R1        ;RESTORE R1
004564  012600                        MOV     (SP)+,R0        ;RESTORE R0
004566  000002                        RTI                     ;REUTRN TO THE MAIN PROGRAM
004570  000000        WRDCNT: 0
004572     000        CHRCNT: .BYTE                           ;NUMBER OF CHARACTERS TO PRINT
004573     000        SPACNT: .BYTE   0                       ;NUMBER OF SPACES TO PRINT

004574  000000        BINWRD: 0


                                        ;TRAP DISPATCH SERVICE
                                        ;ARGUMENT OF TRAP IS EXTRACTED
                                        ;AND USED AS OFFSET TO OBTAIN POINTER
                                        ;TO SELECTED SUBROUTINE
```

```
1106
1107  004576  011646                  .TRPSR: MOV     (SP),-(SP)      ;GET PC OF RETURN
1108  004600  162716  000002                  SUB     #2,(SP)         ;=PC OF TRAP
1109  004604  017616  000000                  MOV     @(SP),(SP)      ;GET TRP
1110  004610  006316                  TRPOK:  ASL     (SP)            ;MULTIPLY TRAP ARG BY 2
1111  004612  042716  117001                  BIC     #117001,(SP)    ;CLEAR UNWANTED BITS
1112  004616  062716  001352                  ADD     #.TRPTAB,(SP)   ;POINTER TO SUBROUTINE ADDRESS
1113  004622  017616  000000                  MOV     @(SP),(SP)      ;SUBROUTINE ADDRESS
1114  004626  000136                          JMP     @(SP)+          ;GO TO SUBROUTINE
1115
1116
1117                                          ;SOFTWARE TIMER FOR RELATIVE TIMING TESTS
1118                                          ;----------------------------------------
1119
1120  004630  005203                  CLOCK:  INC     R3              ;COUNT A TIME TICK
1121  004632  001376                          BNE     CLOCK           ;KEEP GOING UNTIL WE REACH 0
1122  004634  005237  001304                  INC     TEMP1           ;NOW ADD TO THE OVERALL TIMER
1123  004640  000773                          BR CLOCK                ;KEEP GOING
1124                                          ;ERROR HAND'LER
1125                                          ;-------------
1126
1127  004642  032777  010000  174332  .EROR:  BIT     #SW12,@SWR      ;BELL ON ERROR?
1128  004650  001406                          BEQ     XBX             ;BR IF NO BELL
1129  004652  105777  174332                  TSTB    @TPCSR          ;TTY READY.
1130  004656  100003                          BPL     XBX             ;DON'T WAIT IF TTY NOT READY.
1131  004660  112777  000207  174324          MOVB    #207,@TPDBR     ;PUSH A BELL AT THE TTY.
1132  004666  032777  020000  174306  XBX:    BIT     #SW13,@SWR      ;DELETE ERROR PRINT OUT?
1133  004674  001111                          BNE     HALTS           ;BR IF NO PRINT OUT WANTED.
1134  004676  021637  001234                  CMP     (SP),LSTERR     ;WAS THIS ERROR FOUND LAST TIME?
1135  004702  001404                          BEQ     1$              ;BR IF YES
1136  004704  011637  001234                  MOV     (SP),LSTERR     ;RECORD BEING HERE
1137  004710  105037  001345                  CLRB    ERRFLG          ;PREPARE HEADER
1138  004714  104410                  1$:     SAV05                   ;SAVE ALL PROC REGISTERS
1139  004716  011605                          MOV     (SP),R5         ;GET THE PC OF ERROR
1140  004720  162705  000002                  SUB     #2,R5           ;GET ADDRESS OF TRAP CALL
1141  004724  011504                          MOV     (R5),R4         ;GET ERROR INSTRUCTION
1142  004726  006304                          ASL     R4              ;MULT BY TWO
1143  004730  061504                          ADD     (R5),R4         ;DOUBLE IT
1144  004732  006304                          ASL     R4              ;MULT AGAIN
1145  004734  042704  137001                  BIC     #137001,R4      ;CLEAR JUNK
1146  004740  062704  042126                  ADD     #.ERRTAB,R4     ;GET POINTER
1147  004744  012437  005070                  MOV     (R4)+,ERRMSG    ;GET ERROR MESSAGE
1148  004750  012437  005102                  MOV     (R4)+,DATAHD    ;GET DATA HEADRER
1149  004754  011437  005114                  MOV     (R4),DATABP     ;GET DATA TABLE
1150  004760  105737  001345                  TSTB    ERRFLG          ;TYPE HEADER
1151  004764  001403                          BEQ     TYPMSG          ;BR IF YES
1152  004766  005737  005114                  TST     DATABP          ;DOES DATA TABLE EXIST?
1153  004772  001044                          BNE     TYPDAT          ;BR IF YES.
1154  004774  104402  005320  TYPMSG: TYPE    ,MCRLF                  ;TYPE A CARRIAGE RETURN
1155  005000  104402  005320          TYPE    ,MCRLF                  ;AND TYPE ANOTHER
1156  005004  005737  001220          TST     LOCK
1157  005010  001402                  BEQ     1$
1158  005012  104402  005622          TYPE    ,MASTEK
1159  005016  104402  005610  1$:     TYPE    ,MTSTN                  ;TYPE TEST NO
1160  005022  104413  005222          CNVRT   ,XTSTN                  ;SHOW IT
1161  005026  104402  005624          TYPE    ,MERRPC                 ;TYPE PC.
```

```
1162  005032  104413  005214                CNVRT   ,ERTABO         ;SHOW IT
1163  005036  104402  005566                TYPE    ,MPASSX         ;TYPE PASS NO
1164  005042  104413  003530                CNVRT   ,XPASS          ;SHOW IT
1165  005046  104402  005320                TYPE    ,MCRLF          ;GIVE A CR/LF
1166  005052  112737  177777  001345        MOVB    #-1,ERRFLG      ;NO MORE HEADER UNLESS NO DATA TABLE.
1167  005060  005737  005070                TST     EPRMSG          ;IS THERE AN ERROR MESSAGE?
1168  005064  001402                        BEQ     WTBS.FM         ;BR IF NO.
1169  005066  104402                        TYPE                    ;TYPE
1170  005070  000000              ERRMSG:   0                       ;    ERROR MESSAGE
1171  005072                      WTBS.FM:                          ;
1172  005072  005737  005102                TST     DATAHD          ;DATA HEADER?
1173  005076  001402                        BEQ     TYPDAT          ;BR IF NO
1174  005100  104402                        TYPE                    ;TYPE
1175  005102  000000              DATAHD:   0                       ;    DATA HEADER
1176  005104  005737  005114      TYPDAT:   TST     DATABP          ;DATA TABLE?
1177  005110  001402                        BEQ     RESREG          ;BR IF NO.
1178  005112  104412                        CONVRT                  ;SHOW
1179  005114  000000              ATABP:    0                       ;    DATA TABLE
1180  005116  104411              ESREG:    RESOS                   ;RESTORE PROC REGISTERS
1181  005120  022737  003470  000042  ALTS:   CMP     #$ENDAD,@#42    ;CHECK TO SEE IF IN ACT-11 MODE
1182  005126  001403                        BEQ     1$              ;IF SO, HANDLE ACCORDINGLY
1183  005130  005777  174046                TST     @SWR            ;HALT ON ERROR?
1184  005134  100005                        BPL     EXITER          ;BR IF NO HALT ON ERROR
1185  005136  010046              1$:       PUSHRO                  ;SAVE R0
1186  005140  016600  000002                MOV     2(SP),R0        ;SHOW ERROR PC IN DATA DISPLAY
1187  005144  000000                        HALT                    ;HALT
1188  005146  012600                        POPRO                   ;GET R0
1189  005150  005237  001232      EXITER:   INC     ERRCNT          ;UPDATE ERROR COUNT
1190  005154  032777  000400  174020        BIT     #SW08,@SWR      ;GOTO TOP OF TEST?
1191  005162  001007                        BNE     1$              ;BR IF YES
1192  005164  032777  002000  174010        BIT     #SW10,@SWR      ;GOTO NEXT TEST?
1193  005172  001407                        BEQ     2$              ;BR IF NO
1194  005174  013737  001216  001214        MOV     NEXT,RETURN     ;SET FOR NEXT TEST
1195  005202  012706  001200              1$:       MOV     #STACK,SP       ;RESET SP
1196  005206  000177  174002                JMP     @RETURN         ;GOTO SPECIFIED TEST
1197  005212  000002              2$:       RTI                     ;RETURN
1198  005214  000001              ERTABO:   1
1199  005216     006     002                .BYTE   6,2
1200  005220  001334                        SAVPC
1201  005222  000001              XTSTN:    1
1202  005224     003     002                .BYTE   3,2
1203  005226  001226                        TSTNO
1204                                        ;ENTER HERE ON POWER FAILURE
1205                                        ;--------------------------
1206
1207
1208  005230                      .PFAIL:
1209  005230  012737  005242  000024        MOV     #RESTART,24     ;SET UP FOR POWER UP TRAP
1210  005236  000000                        HALT                    ;HALT ON POWER DOWN NORMAL
1211  005240  000777                        BR      .               ;PREVENT ANY FURTHER PROGRESS
1212
1213                                        ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
1214
1215  005242                      RESTART:
1216  005242  012737  005230  000024        MOV     #.PFAIL,24      ;SET UP FOR POWER FAILURE
1217  005250  012706  001200                MOV     #STACK,SP       ;RESET THE STACK POINTER
```

```
 1218  005254  005000                  CLR     R0              ;READY FOR TIMMER
 1219  005256  005200          1$:     INC     R0              ;PLUS ONE TO THE TIMER!
 1220  005260  001376                  BNE     1$              ;BR IF MORE TO GO
 1221  005262  104402  005323          TYPE    ,MPFAIL         ;TYPE THE MESSAGE
 1222  005266  104413  005306          CNVRT   ,PFTAB          ;TELL WHAT TEST TO RETURN TO.
 1223  005272  105037  001345          CLRB    EPRFLG          ;START CLEAN
 1224  005276  005037  001234          CLR     LSTERR          ;  "     "
 1225  005302  000177  173706          JMP     @RETURN         ;START DOING THAT TEST AGAIN.
 1226  005306  000001                  PFTAB:  1
 1227  005310     003     002                  .BYTE   3,2
 1228  005312  001226                  TSTNO
 1229  005314  020040  000077          MQM:    .ASCIZ  /  ?/
   (2) 005320  005015     000          MCRLF:  .ASCIZ  <15><12>
   (2) 005323     377  053520  020122 MPFAIL: .ASCIZ  <377>/PWR FAILED. RESTART AT TEST /
   (2) 005361     377  047105  020104 MEPASS: .ASCIZ  <377>/END PASS CZTRAB  /
   (2) 005404  051377  047125  044516 MR:     .ASCIZ  <377>/RUNNING /
   (2) 005416  050377  047522  051107 MERR2:  .ASCIZ  <377>/PROGRAM INDICATES NO DEVICES PRESENT./
   (2) 005465     377  047111  052523 MERR3:  .ASCIZ  <377>/INSUFFICIENT DATA!/
   (2) 005511     377  042524  052123 MTSTPC: .ASCIZ  <377>/TEST PC-/
   (2) 005523     377  047514  045503 MLOCK:  .ASCIZ  <377>/LOCK ON SELECTED TEST/
   (2) 005552  051503  035122  000040 MCSRX:  .ASCIZ  /CSR: /
   (2) 005560  042526  035103  000040 MVECX:  .ASCIZ  /VEC: /
   (2) 005566  040520  051523  051505 MPASSX: .ASCIZ  /PASSES: /
   (2) 005577     105  051122  051117 MERRX:  .ASCIZ  /ERRORS: /
   (2) 005610  042524  052123  047040 MTSTN:  .ASCIZ  /TEST NO: /
   (2) 005622  000052                  MASTEK: .ASCIZ  /*/
   (2) 005624  041520  020072     000 MERRPC: .ASCIZ  /PC: /
   (2) 005631     377  040515  020120 XHEAD:  .ASCIZ  <377>/MAP OF TR79 STATUS/<377>
   (2) 005656  051377  051505  052105 MRSTFL: .ASCIZ  <377>/RESET  /
   (2) 005667     377  051127  052111 MWRTFL: .ASCIZ  <377>/WRITE  /
   (2) 005700  051377  040505  020104 MRDFL:  .ASCIZ  <377>/READ  /
   (2) 005710  051777  040520  042503 MSRVFL: .ASCIZ  <377>/SPACE REVERSE  /
   (2) 005731     377  051105  051501 MERFL:  .ASCIZ  <377>/ERASE  /
   (2) 005742  020377  042522  047515 MRMVRG: .ASCIZ  <377>/ REMOVE WRITE ENABLE RING LOAD & PLACE ONLINE /
   (2) 006022  044777  051516  051105 MINSRG: .ASCIZ  <377>/INSERT WRITE ENABLE RING LOAD & PLACE ONLINE/
   (2) 006100  044777  052116  051105 MIENFL: .ASCIZ  <377>/INTERRUPT ENABLE  /
   (2) 006124  044777  052116  051105 MIDSFL: .ASCIZ  <377>/INTERRUPT DISABLE  /
   (2) 006151     377  047520  042527 MPCFL:  .ASCIZ  <377>/POWER CLEAR  /
   (2) 006170  053777  044522  042524 MWIPLFL:.ASCIZ  <377>/WRITE ID BLOCK PAST LOAD POINT  /
   (2) 006232  051377  053505  047111 MRWLPFL:.ASCIZ  <377>/REWIND FROM LOAD POINT  /
   (2) 006264  044777  046114  043505 MILFFL: .ASCIZ  <377>/ILLEGAL FUNCTION  /
   (2) 006310  053777  044522  042524 MWLPFL: .ASCIZ  <377>   /WRITE FROM LOAD  POINT  /
   (2) 006342  053777  044522  042524 MWEFFL: .ASCIZ  <377>/WRITE END OF FILE  /
   (2) 006367     377  051127  052111 MWELPFL:.ASCIZ  <377>/WRITE END OF FILE FROM  LOAD POINT  /
   (2) 006435     377  046120  041501 MUOLN:  .ASCIZ  <377>/PLACE UNIT ON LINE AFTER TAPE MOTION HAS STOPPED /
   (2) 006520  047777  020116  044514 MOLEFL: .ASCIZ  <377>/ON LINE ERROR  /
   (2) 006541     377  051127  052111 MWILPFL:.ASCIZ  <377>/WRITE ID BLOCK FROM LOAD POINT  /
   (2) 006603     377  047125  052111 MUNOFLN:.ASCIZ  <377>/UNIT DID NOT GO OFF LINE  /
   (2) 006637     377  043117  020106 MOFLEFL:.ASCIZ  <377>/OFF LINE ERROR  /
   (2) 006661     377  042522  044527 MRWOFL: .ASCIZ  <377>/REWIND OFF LINE  /
   (2) 006704  051377  040505  020104 MRDERR: .ASCIZ  <377>/READ  DATA ERROR /
   (2) 006727     377  050123  041501 MSREFFL:.ASCIZ  <377>/SPACE REVERSE OVER END OF FILE  /
   (2) 006771     377  042522  042101 MREFFL: .ASCIZ  <377>/READ END OF FILE  /
   (2) 007015     377  051127  052111 MWTOFL: .ASCIZ  <377>/WRITE TIME OUT  /
   (2) 007037     377  042522  042101 MRTOFL: .ASCIZ  <377>/READ TIME OUT  /
   (2) 007060  053777  044522  042524 MWNXFL: .ASCIZ  <377>/WRITE NON-EXISTENT MEMORY  /
```

```
   (2)  007115     377   040505  041040  MBNXFL: .ASCIZ  <377>/EA BIT 12- NON EXISTENT MEMORY  /
   (2)  007157     377   054105  042524  MNXBFL: .ASCIZ  <377>/EXTENSION BIT 13 NON EXISTENT MEMORY  /
   (2)  007227     377   040505  041040  MNXMFL: .ASCIZ  <377>/EA BITS 12 AND 13- NON EXISTENT MEMORY  /
   (2)  007301     377   042522  042101  MRCFL:  .ASCIZ  <377>/READ COUNT  /
   (2)  007317     377   047105  020104  METFL:  .ASCIZ  <377>/END OF TAPE  /
   (2)  007336  051377   051505  052105  MRSTU:  .ASCIZ  <377>/RESET TAPE UNIT  /
   (2)  007361     377   040515  052516  MMRSFL: .ASCIZ  <377>/MANUAL RESET  /
   (2)  007401     377   051127  052111  MWPETFL:.ASCIZ  <377>/WRITE  PAST END OF TAPE  /
   (2)  007434  051777   040520  042503  MSRETFL:.ASCIZ  <377>/SPACE REVERSE PAST END OF TAPE  /
   (2)  007476  051377   040505  020104  MRPETFL:.ASCIZ  <377>/READ PAST END OF TAPE  /
   (2)  007527     377   051127  052111  MWEFETF:.ASCIZ  <377>/WRITE END  OF FILE  PAST END OF TAPE  /
   (2)  007600  051377   040505  020104  MREFETF:.ASCIZ  <377>/READ END OF FILE  PAST END  OF TAPE  /
   (2)  007650  053777   044522  042524  MWENFL: .ASCIZ  <377>/WRITE ENABLE  /
   (2)  007670  051777   040520  042503  MSRLPFL:.ASCIZ  <377>/SPACE REVERSE AT LOAD POINT  /
   (2)  007727     377   050123  041501  MSRIFL: .ASCIZ  <377>/SPACE REVERSE AT ID BLOCK  /
   (2)  007764  051377   040505  020104  MRIBFL: .ASCIZ  <377>/READ ID BLOCK  /
   (2)  010005     377   053105  047105  MPARFL: .ASCIZ  <377>/EVEN PARITY  /
   (2)  010024  042777   042526  020116  MEPSRFL:.ASCIZ  <377>/EVEN PARITY SPACE REVERSE  /
   (2)  010061     377   042522  042101  MREPFL: .ASCIZ  <377>/READ EVEN PARITY  /
   (2)  010105     377   047527  042122  MWCFL:  .ASCIZ  <377>/WORD COUNT  /
   (2)  010123     377   047125  052111  MUNERR: .ASCIZ  <377>/UNIT HAS EPROR FLAG SET /
   (2)  010155     377   040524  042520  MUNRDY: .ASCIZ  <377>/TAPE UNIT NOT READY /
   (2)  010203     377   177777  042607  MEOT:   .ASCIZ  <377><377><377><207>/END OF TAPE... TAPE REWOUND /
   (2)  010244  052377   054524  041040  MTBEX:  .ASCIZ  <377>/TTY BUFFER EXCEEDED. RESTART INPUT. /
   (2)  010312  040506   046111  051125  MFAIL:  .ASCIZ  /FAILURE /
   (2)  010323     015   050012  047522  PRHLT:  .ASCIZ <15><12>/PROGRAM HALT - DEVICE FAILED TO AUTOSIZE /<15><12>
   (2)  010401     377   052501  047524  SIZE:   .ASCIZ<377>/AUTOSIZING/
   (2)  010415     377   051503  020122  CSRER2: .ASCIZ <377>/CSR ADDRESS /
   (2)  010433     377   042526  052103  VECER3: .ASCIZ <377>/VECTOR ADDRESS /
   (2)  010454  041377   020122  042514  BRER1:  .ASCIZ <377>/BR LEVEL INTERRUPT /
   (2)          010502
   (2)  010502  000002                   XSTATQ: 2
1230    010504     006      003                   .BYTE   6,3
1231    010506  001304                           TEMP1
1232    010510     006      002                   .BYTE   6,2
1233    010512  001306                           TEMP2
1234                                  .EVEN
1235
1236                                           ;BUFFERS FOR INPUT-OUTPUT
1237
1238    010514  000000                  INBUF:  0
1239            010556                  .=.+40
1240    010556  000000                  TTYEND: 0
1241    010560  000000                  TEMP:   0
1242            010622                  .=.+40
1243    010622  000000                  MDATA:  0
1244            010664                  .=.+40
```

```
                                                ;
                                                ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
                                                ;THIS ROUTINE *MUST*
                                                ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
                                                ;SETUP NECESSARY.
                                                ;
010664  013700  001342          CYCLE:  MOV     ACTIVE,R0       ;GET ADDRESS POINTER.
010670  012037  001242                  MOV     (R0)+,TRBASE    ;LOAD SYSTEM CTRL. REG
010674  012037  001420                  MOV     (R0)+,TRVCT     ;LOAD VECTOR
010700  012037  041766                  MOV     (R0)+,TRPRT     ;LOAD PRIORITY
010704  004737  041640                  JSR     PC,TRLEV        ;SET UP
010710  005737  000042                  TST     @#42            ;ARE WE UNDER MONITOR CONTROL?
010714  001046                          BNE     4$              ;IF YES, SKIP THIS SETUP
010716  032777  000002  170256          BIT     #SW01,@SWR      ;IF SW01=1, GET STARTING TEST #
010724  001442                          BEQ     4$              ;BR IF NO TEST IS TO BE INPUTTED
010726  104402  005320          7$:     TYPE    ,MCRLF
                                        ;GET THE STARTING TEST NUMBER
010732  104405                          INSTR                   ;CALL THE STRING INPUT ROUTINE
010734  005610                          MTSTN                   ;POINTER TO MESSAGE TO BE PRINTED
010736  104407                          PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
010740  000001                          1                       ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
010742  001000                          1000                    ;HIGHEST LEGITIMATE VALUE OF EXPECTED REPONSE
010744  001226                          TSTNO                   ;POINTER TO MAP LOCATION TO BE FILLED
010746     000                          .BYTE   0               ;MASK OF INVALID BITS FOR THIS PARAMETER
010747     001                          .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
010750  012700  011336                  MOV     #TST1,R0
010754  022710  012737          5$:     CMP     #12737,(R0)
010760  001015                          BNE     6$
010762  023760  001226  000002          CMP     TSTNO,2(R0)
010770  001011                          BNE     6$
010772  022760  001226  000004          CMP     #TSTNO,4(R0)
011000  001005                          BNE     6$
011002  010037  001214                  MOV     R0,RETURN       ;SAVE PC
011006  104402  005320                  TYPE    ,MCRLF
011012  000412                          BR      8$
011014  005720                  6$:     TST     (R0)+
011016  020027  012326                  CMP     R0,#TLAST+10
011022  001354                          BNE     5$
011024  104402  005314                  TYPE    ,MQM
011030  000736                          BR      7$
011032  012737  011336  001214  4$:     MOV     #TST1,RETURN    ;PREPARE RETURN ADDRESS
011040  000177  170150          8$:     JMP     @RETURN         ;GO START TESTING.
```

```
1292
1293            000062                          MAXERR=50.      ;MAXIMUM NUMBER OF DEROR PRINTOUTS
1294            000006                          MAXTST=6.       ;MAXIMUM NUMBER OF TESTS
1295            000012                          PASRAT=10.      ;PASS RATE PRINT OUT
1296            040000                          CORSIZ=40000    ;CORE REQUIRED
1297
1298                                    ;-------------------------------------------------------------------*
1299                                    .SBTTL  PRETEST ROUTINE
1300
1301
1302
1303  011044    012737  011044  001250  PRETST: MOV     #PRETST,TSTPTR  ;SET TEST ADDRESS
1304  011052    004737  014664                  JSR     PC,TESTN
1305  011056    013727  001240                  MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
1306  011062    000000                  64$:    .WORD   0               ;USE THIS WORD AS A TIME COUNTER
1307  011064                            66$:
1308  011064    005227  000000                  INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
1309  011070    001375                          BNE     66$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
1310  011072    005337  011062                  DEC     64$             ;HAS THE TOTAL TIME ELAPSED?
1311  011076    001372                          BNE     66$             ;IF NO,GO WAIT A LITTLE LONGER
1312  011100                            65$:
1313  011100    052777  004000  170274          BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
1314  011106    032777  004000  170266  67$:    BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
1315  011114    001374                          BNE     67$             ;IF NO, WAIT FOR IT TO CLEAR
1316  011116    000005                          RESET
1317  011120    022777  006026  170254          CMP     #6026,@TRCR     ;CHECK COMMAND REGISTER
1318  011126    001412                          BEQ     1$              ;SKIP IF OK
1319  011130    012705  006026                  MOV     #6026,R5
1320  011134    017704  170242                  MOV     @TRCR,R4
1321  011140    013737  001402  001244          MOV     TRCR,REGIST
1322  011146    104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
1323  011150    104403  005656                  TYPEF   ,MRSTFL
1324
1325  011154                            1$:
1326  011154    012737  000050  001240          MOV     #50,HOLD        ;SET UP FOR A MASSIVE DELAY
1327  011162    013727  001240                  MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
1328  011166    000000                  68$:    .WORD   0               ;USE THIS WORD AS A TIME COUNTER
1329  011170                            70$:
1330  011170    005227  000000                  INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
1331  011174    001375                          BNE     70$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
1332  011176    005337  011166                  DEC     68$             ;HAS THE TOTAL TIME ELAPSED?
1333  011202    001372                          BNE     70$             ;IF NO,GO WAIT A LITTLE LONGER
1334  011204                            69$:
1335  011204    012737  000010  001240          MOV     #10,HOLD        ;RESTORE THE NORMAL DELAY FACTOR
1336  011212    022777  002226  170162          CMP     #2226,@TRCR     ;TEST COMMAND REGISTER
1337  011220    001412                          BEQ     2$              ;SKIP IF OK
1338  011222    012705  002226                  MOV     #2226,R5
1339  011226    017704  170150                  MOV     @TRCR,R4
1340  011232    013737  001402  001244          MOV     TRCR,REGIST
1341  011240    104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
1342  011242    104403  005656                  TYPEF   ,MRSTFL
1343
1344  011246                            2$:
1345  011246    017701  170134                  MOV     @TRSR,R1                ;GET STATUS
1346  011252    042701  002247                  BIC     #2247,R1        ;CLEAR UNWANTED BITS
1347  011256    005701                          TST     R1              ;TEST THE STATUS REGISTER
```

```
1348  011260  001410                          BEQ       3$          ;SKIP IF OK
1349  011262  012705  000000                  MOV       #0,R5
1350  011266  010104                          MOV       R1,R4
1351  011270  013737  001406  001244          MOV       TRSR,REGIST
1352  011276  104001                          ERROR     1           ;INCORRECT REGISTER MATCHUP
1353  011300  104404                          TYPEL
1354
1355  011302                            3$:
1356  011302  005737  001252                  TST       ERR         ;TEST THE DEROR FLAG
1357  011306  001406                          BEQ       71$         ;SKIP IF NO DEROR
1358  011310  032777  040000  167664          BIT       #BIT14,@SWR ;TEST LOOP BIT
1359  011316  001402                          BEQ       71$         ;BRANCH IF CLEAR
1360  011320  000137  011044                  JMP       PRETST      ;JUMP IF SET
1361  011324  005037  001252            71$:  CLR       ERR         ;CLEAR THE DEROR FLAG
1362  011330  004737  030232                  JSR       PC,FIXBUF
1363
1364  011334  000207                          RTS       PC
1365
1366                                          ;************************ TEST 1 *********************************
1367                                          ;*       TEST GROUP 1
1368                                          ;;*************************************************************
1369                                    ;;*************
1370                                    ;;*           *
1371                                    ;;*  TEST 1   *
1372                                    ;;*           *
1373                                    ;;*************
1374  011336  012737  000001  001226    TST1: MOV       #1,@#TSTNO  ;LOAD THE NUMBER OF THIS TEST IN THE INDICATOR
1375  011344  012737  011516  001216          MOV       #TST2,NEXT  ;POINT TO THE START OF THE NEXT TEST
1376  011352  032777  000010  167622          BIT       #10,@SWR    ;SKIP??
1377  011360  001002                          BNE       2$          ;NO
1378  011362  000137  011456                  JMP       END1
1379  011366  012737  011460  001302  2$:     MOV       #TABLE1,TC  ;SET TEST CONTROL INDEX
1380  011374  012706  001200                  MOV       #STACK,SP   ;RESET THE STACK
1381  011400                            1$:
1382  011400  013737  001302  001250          MOV       TC,TSTPTR
1383  011406  013705  001302                  MOV       TC,R5       ;SET DEROR SUBTEST ADDRESS
1384  011412  062737  000002  001302          ADD       #2,TC       ;POINT TO NEXT TEST
1385  011420  004775  000000                  JSR       PC,@(R5)    ;PERFORM THE TEST
1386  011424  000240                          NOP                   ;THIS LOCATION MAY BE CHANGED TO A HALT TO INSPECT
1387                                                                ; EQUIPMENT STATUS OR PROGRAM OPERATION
1388  011426  005737  001252                  TST       ERR         ;TEST THE DEROR FLAG
1389  011432  001406                          BEQ       64$         ;SKIP IF NO DEROR
1390  011434  032777  040000  167540          BIT       #BIT14,@SWR ;TEST LOOP BIT
1391  011442  001402                          BEQ       64$         ;BRANCH IF CLEAR
1392  011444  000137  011400                  JMP       1$          ;JUMP IF SET
1393  011450  005037  001252            64$:  CLR       ERR         ;CLEAR THE DEROR FLAG
1394  011454  000751                          BR        1$          ;DO NEXT TEST
1395  011456  104400                    END1: SCOPE                 ;PASS COMPLETED
1396
1397
1398
1399                                          ;THIS TABLE CONTAINS THE SUBTESTS TO BE PERFORMED BY TEST 1
1400
1401  011460  012564                    TABLE1: TESTB   ;REMOVE WRITE RING
1402  011462  024014                            RUNAS   ;TRY TO WRITE WITHOUT RING
1403  011464  012646                            TESTC   ;INSERT WRITE RING
```

```
011466 021752                          TESTAJ   ;DELAY 10 SECONDS
011470 015072                          TESTP    ;TAKE UNIT OFF LINE
011472 022024                          TESTAK   ;TELL OPERATOR TO RESET
011474 013626                          TESTI    ;REWIND THE TAPE
011476 022260                          TESTAL   ;WRITE PAST EOT
011500 022616                          TESTAM   ;SPACE REVERSE PAST EOT
011502 023024                          TESTAN   ;READ PAST EOT
011504 023172                          TESTAO   ;WRITE EOF PAST EOT
011506 023400                          TESTAP   ;SPACE REVERSE OVER EOF PAST EOT
011510 023606                          TESTAQ   ;READ EOF PAST EOT
011512 015214                          TESTQ    ;REWIND AND GO OFF LINE
011514 011456                          END1

                                       ;*********************** TEST 2 ******************************
                                       ;*            TEST GROUP 2
                                       ;:*********************************************************
                              ;:************
                              ;:*          *
                              ;:*   TEST 2
                              ;:*          *
                              ;:************
011516 012737 000002 001226   TST2:    MOV      #2,@#TSTNO         ;LOAD THE NUMBER OF THIS TEST IN THE INDICATOR
011524 012737 011656 001216            MOV      #TST3,NEXT         ;POINT TO THE START OF THE NEXT TEST
011532 012737 011624 001302            MOV      #TABLE2,TC         ;SET TEST CONTROL INDEX
011540 012706 001200                   MOV      #STACK,SP          ;RESET THE STACK
011544                        1$:
011544 013737 001302 001250            MOV      TC,TSTPTR
011552 013705 001302                   MOV      TC,R5              ;SET DEROR SUBTEST ADDRESS
011556 062737 000002 001302            ADD      #2,TC              ;POINT TO NEXT TEST
011564 004775 000000                   JSR      PC,@(R5)           ;PERFORM THE TEST
011570 000240                          NOP                         ;THIS LOCATION MAY BE CHANGED TO A HALT TO INSPECT
                                                          ;         EQUIPMENT STATUS OR PROGRAM OPERATION
011572 005737 001252                   TST      ERR                ;TEST THE DEFOR FLAG
011576 001406                          BEQ      64$                ;SKIP IF NO DEROR
011600 032777 040000 167374            BIT      #BIT14,@SWR        ;TEST LOOP BIT
011606 001402                          BEQ      64$                ;BRANCH IF CLEAR
011610 000137 011544                   JMP      1$                 ;JUMP IF SET
011614 005037 001252          64$:     CLR      ERR                ;CLEAR THE DEROR FLAG
011620 000751                          BR       1$                 ;DO NEXT TEST
011622 104400                 END2:    SCOPE                       ;PASS COMPLETED



                                       ;THIS TABLE CONTAINS THE SUBTESTS TO BE PERFORMED BY TEST 2

011624 013626                          TABLE2: TESTI    ;REWIND THE TAPE
011626 012646                                  TESTC    ;INSURE WRITE RING
011630 012760                                  TESTD    ;UNIBUS ENTENTION BIT TEST
011632 013146                                  TESTE    ;INTERRUPT BIT SET TEST
011634 013202                                  TESTF    ;INTERRUPT BIT CLEAR TEST
011636 014034                                  TESTJ    ;ILLEGAL COMMAND TEST
011640 014270                                  TESTK    ;WRITE FROM LOAD POINT
011642 014550                                  TESTM    ;WRITE EOF FROM LOAD POINT
011644 024314                                  TESTAT   ;SPACE REVERSE AT LOAD POINT
011646 020070                                  TESTAB   ;NXM TEST
011650 024522                                  TESTAU   ;BUSS ADDRESS BIT TEST
```

```
                                        TESTBA  ;WORD COUNT BIT TEST
                                        END2


                        ;************************ TEST 3 *********************************
                        ;*      TEST GROUP 3
                        ;:*************************************************************
                ;::************
                ;::*           *
                ;::*  TEST 3
                ;::*           *
                ;::************
011656  012737  000003  001226  TST3:   MOV     #3,@#TSTNO      ;LOAD THE NUMBER OF THIS TEST IN THE INDICATOR
011664  012737  012030  001216          MOV     #TST4,NEXT      ;POINT TO THE START OF THE NEXT TEST
011672  012737  011764  001302          MOV     #TABLE3,TC      ;SET TEST CONTROL INDEX
011700  012706  001200                  MOV     #STACK,SP       ;RESET THE STACK
011704                          1$:
011704  013737  001302  001250          MOV     TC,TSTPTR
011712  013705  001302                  MOV     TC,R5           ;SET DEROR SUBTEST ADDRESS
011716  062737  000002  001302          ADD     #2,TC           ;POINT TO NEXT TEST
011724  004775  000000                  JSR     PC,@(R5)        ;PERFORM THE TEST
011730  000240                          NOP                     ;THIS LOCATION MAY BE CHANGED TO A HALT TO INSPECT
                                                                 EQUIPMENT STATUS OR PROGRAM OPERATION
011732  005737  001252                  TST     ERR             ;TEST THE DEROR FLAG
011736  001406                          BEQ     64$             ;SKIP IF NO DEROR
011740  032777  040000  167234          BIT     #BIT14,@SWR     ;TEST LOOP BIT
011746  001402                          BEQ     64$             ;BRANCH IF CLEAR
011750  000137  011704                  JMP     1$              ;JUMP IF SET
011754  005037  001252          64$:    CLR     ERR             ;CLEAR THE DEROR FLAG
011760  000751                          BR      1$              ;DO NEXT TEST
011762  104400                  END3:   SCOPE                   ;PASS COMPLETED


                        ;THIS TABLE CONTAINS THE SUBTESTS TO BE PERFORMED BY TEST 3

011764  012646          TABLE3: TESTC   ;INSURE WRITE ENABLE RING
011766  013626                  TESTI   ;REWIND THE TAPE
011770  014764                  TESTO   ;WRITE ID BLOCK
011772  013500                  TESTH   ;ILLEGAL ID BLOCK TEST
011774  024562                  TESTAV  ;SPACE REVERSE OVER ID BLOCK
011776  024772                  TESTAW  ;READ ID BLOCK
012000  014414                  RUNL    ;WRITE EOF MARKS
012002  015706                  RUNU    ;SPACE REVERSE OVER EOF MARKS
012004  016116                  RUNV    ;READ EOF MARKS
012006  016322                  TESTW   ;TIME OUT TEST
012010  020350                  TESTAC  ;READ COUNT
012012  020610                  TESTAD  ;READ COUNT
012014  021050                  TESTAE  ;READ COUNT
012016  021242                  TESTAF  ;SPACE COUNT
012020  025236                  TESTAX  ;WRITE EVEN PARITY
012022  025500                  TESTAY  ;SPACE REVERSE OVER EVEN PARITY
012024  025732                  TESTAZ  ;READ EVEN PARITY
012026  011762                  END3


                        ;************************ TEST 4 *********************************
```

```
                                                    ;* TEST GROUP 4
                                                    ;;*********************************************************************
                                        ;;************
                                        ;;*          *
                                        ;;*  TEST 4
                                        ;;*          *
                                        ;;************
012030  012737  000004  001226          TST4:   MOV     #4,@#TSTNO      ;LOAD THE NUMBER OF THIS TEST IN THE INDICATOR
012036  012737  012146  001216                  MOV     #TST5,NEXT      ;POINT TO THE START OF THE NEXT TEST
012044  012737  012136  001302                  MOV     #TABLE4,TC      ;SET TEST CONTROL INDEX
012052  012706  001200                          MOV     #STACK,SP       ;RESET THE STACK
012056                                  AAA:
012056  013737  001302  001250                  MOV     TC,TSTPTR
012064  013705  001302                          MOV     TC,R5           ;SET DEROR SUBTEST ADDRESS
012070  062737  000002  001302                  ADD     #2,TC           ;POINT TO NEXT TEST
012076  004775  000000                          JSR     PC,@(R5)        ;PERFORM THE TEST
012102                                  RTA:
012102  000240                                  NOP                     ;THIS LOCATION MAY BE CHANGED TO A HALT TO INSPECT
                                                                ;       EQUIPMENT STATUS OR PROGRAM OPERATION
012104  005737  001252                          TST     ERR             ;TEST THE DEROR FLAG
012110  001406                                  BEQ     64$             ;SKIP IF NO DEROR
012112  032777  040000  167062                  BIT     #BIT14,@SWR     ;TEST LOOP BIT
012120  001402                                  BEQ     64$             ;BRANCH IF CLEAR
012122  000137  012056                          JMP     AAA             ;JUMP IF SET
012126  005037  001252                  64$:    CLR     ERR             ;CLEAR THE DEROR FLAG
012132  000751                                  BR      AAA             ;DO NEXT TEST
012134  104400                          END4:   SCOPE                   ;PASS COMPLETED



                                                ;THIS TABLE CONTAINS THE SUBTESTS TO BE PERFORMED BY TEST 4

012136  012646                          TABLE4: TESTC   ;INSURE WRITE RING
012140  014764                                  TESTO   ;REWIND AND WRITE ID BLOCK
012142  027210                                  TESTXX  ;WRITE, ERASE,SPACE REV, READ AND TEST DATA
012144  012134                                  END4

                                                ;*********************** TEST 5 ******************************
                                                ;*      TEST GROUP 5
                                                ;;*********************************************************************
                                        ;;************
                                        ;;*          *
                                        ;;*  TEST 5
                                        ;;*          *
                                        ;;************
012146  012737  000005  001226          TST5:   MOV     #5,@#TSTNO      ;LOAD THE NUMBER OF THIS TEST IN THE INDICATOR
012154  012737  012316  001216                  MOV     #TST6,NEXT      ;POINT TO THE START OF THE NEXT TEST
012162  032777  000040  167012                  BIT     #BIT5,  @SWR    ;SKIP THIS TEST?
012170  001002                                  BNE     1$              ;NO
012172  000137  012266                          JMP     END5            ;YES
012176  012737  012306  001302          1$:     MOV     #TABLE5,TC      ;SET TEST CONTROL INDEX
012204  012706  001200                          MOV     #STACK,SP       ;RESET THE STACK
012210                                  BBB:
012210  013737  001302  001250                  MOV     TC,TSTPTR
012216  013705  001302                          MOV     TC,R5           ;SET DEROR SUBTEST ADDRESS
012222  062737  000002  001302                  ADD     #2,TC           ;POINT TO NEXT TEST
```

```
1572  012230  004775  000000            JSR     PC,@(R5)        ;PERFORM THE TEST
1573  012234                    RTB:
1574  012234  000240                    NOP                     ;THIS LOCATION MAY BE CHANGED TO A HALT TO INSPECT
1575                                                               EQUIPMENT STATUS OR PROGRAM OPERATION
1576  012236  005737  001252            TST     ERR             ;TEST THE DEROR FLAG
1577  012242  001406                    BEQ     64$             ;SKIP IF NO DEROR
1578  012244  032777  040000  166730    BIT     #BIT14,@SWR     ;TEST LOOP BIT
1579  012252  001402                    BEQ     64$             ;BRANCH IF CLEAR
1580  012254  000137  012210            JMP     BBB             ;JUMP IF SET
1581  012260  005037  001252    64$:    CLR     ERR             ;CLEAR THE DEROR FLAG
1582  012264  000751                    BR      BBB             ;DO NEXT TEST
1583  012266                    END5:
1584  012266  032777  000010  166706    BIT     #10,@SWR        ;SKIP NEXT TEST?
1585  012274  001003                    BNE     1$              ;NO
1586  012276  012737  003364  001216    MOV     #.EOP,NEXT      ;SET EOP
1587  012304  104400            1$:     SCOPE                   ;PASS COMPLETED
1588
1589                                    ;THIS TABLE CONTAINS THE SUBTESTS TO BE PERFORMED BY TEST 5
1590
1591  012306  013626            TABLE5: TEST1   ;REWIND THE TAPE
1592  012310  024772                    TESTAW  ;READ THE ID BLOCK
1593  012312  027436                    TESTXY  ;READ THE TAPE
1594  012314  012266                    END5
1595
1596                                    ;************************ TEST 6 *******************************
1597                                    ;*      TEST GROUP 6
1598                                    ;*************************************************************
1599                            ;;*************
1600                            ;;*           *
1601                            ;;*  TEST 6   *
1602                            ;;*           *
1603                            ;;*************
1604  012316  012737  000006  001226    TST6:   MOV     #6,@#TSTNO      ;LOAD THE NUMBER OF THIS TEST IN THE INDICATOR
1605  012324  012737  003364  001216            MOV     #.EOP,NEXT      ;POINT TO THE END-OF-PASS HANDLER
1606  012332  012737  012520  001302            MOV     #TABLE6,TC      ;SET TEST CONTROL INDEX
1607  012340  012706  001200                    MOV     #STACK,SP       ;RESET THE STACK
1608  012344                    1$:
1609  012344  104405                    INSTR                   ;CALL THE STRING INPUT ROUTINE
1610  012346  012456                    MESTADD                 ;POINTER TO MESSAGE TO BE PRINTED
1611  012350  104407                    PARAM                   ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1612  012352  011336                    TST1                    ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1613  012354  027210                    TESTXX                  ;HIGHEST LEGITIMATE VALUE OF EXPECTED REPONSE
1614  012356  012520                    TABLE6                  ;POINTER TO MAP LOCATION TO BE FILLED
1615  012360     001                    .BYTE   1               ;MASK OF INVALID BITS FOR THIS PARAMETER
1616  012361     001                    .BYTE   1               ;NUMBER OF PARAMETERS TO STORE
1617  012362  012737  012520  001302    2$:     MOV     #TABLE6,TC      ;SET TEST CONTROL INDEX
1618  012370  017737  166706  001250            MOV     @TC,TSTPTR              ;SET DEROR SUBTEST ADDRESS
1619  012376  013705  001302                    MOV     TC,R5
1620  012402  062737  000002  001302            ADD     #2,TC
1621  012410  004775  000000                    JSR     PC,@(R5)        ;PERFORM THE TEST
1622  012414  000240                    NOP                     ;THIS LOCATION MAY BE CHANGED TO A HALT TO INSPECT
1623                                                               EQUIPMENT STATUS OR PROGRAM OPERATION
1624  012416  012737  177777  001252            MOV     #-1,ERR         ;SET DEROR FLAG
1625  012424  005737  001252                    TST     ERR             ;TEST THE DEROR FLAG
1626  012430  001406                    BEQ     64$             ;SKIP IF NO DEROR
1627  012432  032777  040000  166542            BIT     #BIT14,@SWR     ;TEST LOOP BIT
```

```
                                        BEQ     64$             ;BRANCH IF CLEAR
                                        JMP     2$              ;JUMP IF SET
                                64$:    CLR     ERR             ;CLEAR THE DEROR FLAG
                                        BR      1$              ;DO NEXT TEST
                                END6:   SCOPE                   ;PASS COMPLETED

                                MESTADD:        .ASCIZ  <377>/ENTER STARTING ADDRESS OF TEST /
                                .EVEN

                                        ;THIS TABLE CONTAINS THE SUBTESTS TO BE PERFORMED BY TEST 6

                                TABLE6: 0
                                        END6

                                ;-------------------------------------------------------------------*

                                .SBTTL  SUB-TEST ROUTINES

                                        ;THIS ROUTINE WAITS FOR TAPE UNIT TO GET READY OR DEROR
                                        ;OR TIMES OUT AND EXITS WHEN IT DOES

                                TESTA:  CLR     R1              ;CLEAR TIMER
                                1$:
                                        INC     R1              ;COUNT UP
                                        BEQ     2$              ;SKIP IF ZERO
                                        BIT     #BIT15,@TRCR
                                        BNE     2$              ;SKIP IF SET
                                        BIT     #BIT10,@TRCR
                                        BEQ     1$              ;NO-BRANCH
                                        BIT     #BIT7,@TRCR
                                        BEQ     1$              ;NO-BRANCH
                                2$:
                                        RTS     PC

                                ;-------------------------------------------------------------------*
                                ;THIS ROUTINE INSURES THE WRITE ENABLE RING
                                ;REMOVED AND EXITS WHEN IT IS

                                TESTB:
                                        BIT     #BIT2,@TRSR
                                        BNE     2$              ;SKIP IF ONE
                                        TYPE    ,MRMVRG
                                1$:
                                        BIT     #BIT2,@TRSR
                                        BNE     2$              ;SKIP IF ONE
                                        INC     TEMP
                                        BNE     64$
                                        MOV     #207,@TPDBR     ;RING THE BELL
                                64$:
                                        BR      1$
```

```
1684  012626                          2$:
1685  012626  052777  004000  166546          BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
1686  012634  032777  004000  166540  65$:    BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
1687  012642  001374                          BNE     65$             ;IF NO, WAIT FOR IT TO CLEAR
1688  012644  000207                          RTS     PC
1689
1690                                  ;-----------------------------------------------------------------------*
1691                                  ;THIS ROUTINE INSURES THE WRITE ENABLE RING
1692                                  ;IS INSERTED AND EXITS WHEN IT IS
1693
1694  012646                          TESTC:
1695  012646  032777  000004  166532          BIT     #BIT2,@TRSR
1696  012654  001420                          BEQ     2$              ;SKIP IF ZERO
1697  012656  104402  006022                  TYPE    ,MINSRG
1698  012662                          1$:
1699  012662  005237  010560                  INC     TEMP
1700  012666  001003                          BNE     64$
1701  012670  012777  000207  166314          MOV     #207,@TPDBR     ;RING THE BELL
1702  012676                          64$:
1703  012676  032777  000004  166502          BIT     #BIT2,@TRSR
1704  012704  001366                          BNE     1$              ;LOOP TILL ZERO
1705  012706  032777  002000  166466          BIT     #BIT10,@TRCR
1706  012714  001762                          BEQ     1$              ;BRANCH IF CLEAR
1707  012716                          2$:
1708  012716  052777  004000  166456          BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
1709  012724  032777  004000  166450  65$:    BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
1710  012732  001374                          BNE     65$             ;IF NO, WAIT FOR IT TO CLEAR
1711  012734  004737  014664                  JSR     PC,TESTN
1712  012740  052777  004000  166434          BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
1713  012746  032777  004000  166426  66$:    BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
1714  012754  001374                          BNE     66$             ;IF NO, WAIT FOR IT TO CLEAR
1715  012756  000207                          RTS     PC
1716
1717                                  ;+-------------------------------------------------------------------
1718
1719                                          ;THIS ROUTINE TESTS THE READ WRITE ABILITY OF THE
1720                                          ;UNIBUS EXTENSION ADDRESS BITS
1721
1722  012760                          TESTD:
1723  012760  052777  004000  166414          BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
1724  012766  032777  004000  166406  64$:    BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
1725  012774  001374                          BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
1726  012776  013737  001402  001244          MOV     TRCR,REGIST             ;INDICATE THE ERROR REGIS ?
1727  013004  052777  010000  166370          BIS     #010000,@TRCR   ;SET UNIBUS EXTENSION BIT 16
1728  013012  032777  010000  166362          BIT     #010000,@TRCR   ;TEST UNIBUS EXTENSION BIT 16
1729  013020  001001                          BNE     1$              ;SKIP IF OK
1730  013022  104000                          ERROR                   ;ERROR-BIT WAS NOT CORRECT
1731
1732  013024                          1$:
1733
1734  013024  042777  010000  166350          BIC     #010000,@TRCR   ;CLEAR THE BIT
1735  013032  032777  010000  166342          BIT     #010000,@TRCR   ;TEST THE BIT
1736  013040  001401                          BEQ     2$              ;SKIP IF CLEAR
1737  013042  104000                          ERROR                   ;ERROR-BIT WAS NOT CORRECT
1738
1739  013044                          2$:
```

```


013044  052777  020000  166330          BIS     #020000,@TRCR   ;SET UNIBUS EXTENSION BIT 17
013052  032777  020000  166322          BIT     #020000,@TRCR   ;TEST UNIBUS EXTENSION BIT 17
013060  001001                          BNE     3$              ;SKIP IF OK
013062  104000                          ERROR                   ;ERROR-BIT WAS NOT CORRECT

013064                          3$:


013064  042777  020000  166310          BIC     #020000,@TRCR   ;CLEAR THE BIT
013072  032777  020000  166302          BIT     #20000,@TRCR    ;TEST THE BIT
013100  001401                          BEQ     4$              ;SKIP IF OK
013102  104000                          ERROR                   ;ERROR-BIT WAS NOT CORRECT

013104                          4$:


013104  052777  030000  166270          BIS     #30000,@TRCR    ;SET BOTH EA BITS
013112  032777  030000  166262          BIT     #30000,@TRCR    ;TEST BITS ON
013120  001001                          BNE     5$
013122  104000                          ERROR

013124                          5$:

013124  042777  030000  166250          BIC     #30000,@TRCR    ;CLEAR EA BITS
013132  032777  030000  166242          BIT     #30000,@TRCR    ;MAKE SURE
013140  001401                          BEQ     6$
013142  104000                          ERROR

013144                          6$:
013144  000207                          RTS     PC

                                ;+-------------------------------------------------------------------------

                                        ;THIS ROUTINE SETS THE INTERRUPT BIT AND TESTS IT
                                        ;TO INSURE IT IS SET

013146  012737  000340  177776  TESTE:  MOV     #LEVEL7,PS      ;SET PRIORITY TO 7
013154  052777  000100  166220          BIS     #BIT6,@TRCR
013162  032777  000100  166212          BIT     #BIT6,@TRCR
013170  001003                          BNE     1$              ;SKIP IF SET
013172  104000                          ERROR                   ;ERROR-BIT WAS NOT CORRECT
013174  104403  006100                  TYPEF   ,MIENFL

013200                          1$:
013200  000207                          RTS     PC

                                ;+-------------------------------------------------------------------------

                                        ;THIS ROUTINE CLEARS THE INTERRUPT ENABLE AND TESTS
                                        ;TO INSURE IT IS CLEAR

013202                          TESTF:
013202  042777  000100  166172          BIC     #BIT6,@TRCR
013210  032777  000100  166164          BIT     #BIT6,@TRCR
```

```
1796  013216  001403                        BEQ     1$              ;SKIP IF ZERO
1797  013220  104000                        ERROR                   ;ERROR-BIT WAS NOT CORRECT
1798  013222  104403  006124                TYPEF   ,MIDSFL
1799
1800  013226                        1$:
1801  013226  005037  177776                CLR     PS              ;SET PROCESSOR PRIORITY TO 00
1802  013232  000207                        RTS     PC
1803
1804                                ;+--------------------------------------------------------------------
1805
1806                                ;THIS ROUTINE TESTS THE FUNCTION OF THE POWER CLEAR
1807
1808  013234  000005                TESTG:  RESET
1809  013236  012777  000026  166136        MOV     #26,@TRCR       ;SET COMMAND REGISTER
1810  013244  052777  004000  166130        BIS     #BIT11,@TRCR
1811  013252  012701  177300                MOV     #-500,R1        ;SET DELAY
1812  013256                        1$:
1813  013256  005201                        INC     R1              ;DELAY
1814  013260  001376                        BNE     1$              ;CONTINUE
1815
1816  013262  032777  004000  166112        BIT     #BIT11,@TRCR
1817  013270  001001                        BNE     2$              ;SKIP IF SET
1818  013272  104000                        ERROR                   ;ERROR-BIT WAS NOT CORRECT
1819
1820  013274                        2$:
1821
1822  013274  013727  001240                MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
1823  013300  000000                64$:    .WORD   0               ;USE THIS WORD AS A TIME COUNTER
1824  013302                        66$:
1825  013302  005227  000000                INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
1826  013306  001375                        BNE     66$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
1827  013310  005337  013300                DEC     64$             ;HAS THE TOTAL TIME ELAPSED?
1828  013314  001372                        BNE     66$             ;IF NO,GO WAIT A LITTLE LONGER
1829  013316                        65$:
1830  013316  013727  001240                MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
1831  013322  000000                67$:    .WORD   0               ;USE THIS WORD AS A TIME COUNTER
1832  013324                        69$:
1833  013324  005227  000000                INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
1834  013330  001375                        BNE     69$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
1835  013332  005337  013322                DEC     67$             ;HAS THE TOTAL TIME ELAPSED?
1836  013336  001372                        BNE     69$             ;IF NO,GO WAIT A LITTLE LONGER
1837  013340                        68$:
1838  013340                        3$:
1839  013340  032777  004000  166034        BIT     #BIT11,@TRCR
1840  013346  001403                        BEQ     4$              ;SKIP IF CLEAR
1841  013350  005237  010560                INC     TEMP
1842  013354  001371                        BNE     3$              ;DELAY
1843
1844  013356                        4$:
1845
1846  013356  032777  004000  166016        BIT     #BIT11,@TRCR
1847  013364  001401                        BEQ     5$              ;SKIP IF CLEAR
1848  013366  104000                        ERROR                   ;ERROR-BIT WAS NOT CORRECT
1849
1850  013370                        5$:
1851
```

```
1852  013370  032777  002000  166010          BIT     #BIT10,@TRSR
1853  013376  001001                          BNE     6$              ;SKIP IF ON LINE
1854  013400  000207                          RTS     PC
1855  013402                          6$:
1856
1857  013402  017701  165774                  MOV     @TRCR,R1                ;GET COMMAND REGISTER
1858  013406  042701  000036                  BIC     #00036,R1       ;CLEAR UNWANTED BITS
1859  013412  022701  002200                  CMP     #2200,R1        ;TEST COMMAND REGISTER
1860  013416  001411                          BEQ     7$              ;SKIP IF OK
1861  013420  012705  002200                  MOV     #2200,R5
1862  013424  010104                          MOV     R1,R4
1863  013426  013737  001402  001244          MOV     TRCR,REGIST
1864  013434  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
1865  013436  104403  006151                  TYPEF   ,MPCFL
1866
1867  013442                          7$:
1868
1869  013442  017701  165740                  MOV     @TRSR,R1                ;GET STATUS REGISTER
1870  013446  042701  002247                  BIC     #2247,R1        ;CLEAR UNWANTED BITS
1871  013452  005701                          TST     R1              ;TEST STATUS REGISTER
1872  013454  001410                          BEQ     8$              ;SKIP IF OK
1873  013456  012705  002247                  MOV     #2247,R5
1874  013462  010104                          MOV     R1,R4
1875  013464  013737  001406  001244          MOV     TRSR,REGIST
1876  013472  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
1877  013474  104404                          TYPEL
1878
1879  013476                          8$:
1880
1881  013476  000207                          RTS     PC
1882
1883                                  ;+-------------------------------------------------------------------
1884                                  ;THIS ROUTINE IS USED TO TEST THE FUNCTION OF WRITING A ID
1885                                  ;BLOCK FROM LOAD POINT AND AFTER LOAD POINT.
1886
1887  013500                          TESTH:
1888  013500  004737  014764                  JSR     PC,TESTO
1889  013504  004737  026610                  JSR     PC,TESTCK
1890
1891  013510  004537  027546                  JSR     R5,TRYIT        ;;;
1892  013514  000432                          .WORD   WIDB            ;FUNCTION
1893  013516  000000                          .WORD   -0              ;WORD COUNT
1894  013520  000000                          .WORD   0               ;BUS DDDRESS
1895
1896  013522  022777  142632  165652          CMP     #142632,@TRCR   ;TEST COMMAND REGISTER
1897  013530  001412                          BEQ     1$              ;SKIP IF OK
1898  013532  012705  142632                  MOV     #142632,R5
1899  013536  017704  165640                  MOV     @TRCR,R4
1900  013542  013737  001402  001244          MOV     TRCR,REGIST
1901  013550  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
1902  013552  104403  006170                  TYPEF   ,MWIPLFL
1903
1904  013556                          1$:
1905
1906  013556  022777  002000  165622          CMP     #2000,@TRSR     ;TEST THE STATUS REGISTER
1907  013564  001410                          BEQ     2$              ;SKIP IF OK
```

```
 1908  013566  012705  002000                    MOV     #2000,R5
 1909  013572  017704  165610                    MOV     @TRSR,R4
 1910  013576  013737  001406  001244            MOV     TRSR,REGIST
 1911  013604  104001                            ERROR   1               ;INCORRECT REGISTER MATCHUP
 1912
 1913  013606                          2$:
 1914
 1915  013606  052777  004000  165566            BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
 1916  013614  032777  004000  165560  64$:      BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
 1917  013622  001374                            BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
 1918  013624  000207                            RTS     PC
 1919
 1920                                  ;+--------------------------------------------------------------------
 1921                                  ;THIS ROUTINE TESTS THE FUNCTION OF REWINDING
 1922                                  ;WHILE AT LOAD POINT THIS IS THE MAIN REWIND ROUTINE
 1923
 1924  013626                          TESTI:
 1925  013626  052777  004000  165546            BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
 1926  013634  032777  004000  165540  64$:      BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
 1927  013642  001374                            BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
 1928  013644  004537  027546                    JSR     R5,TRYIT        ;;;
 1929  013650  001020                            .WORD   REWIND          ;FUNCTION
 1930  013652  000000                            .WORD   -0              ;WORD COUNT
 1931  013654  000000                            .WORD   0               ;BUS DDDRESS
 1932          013656                  A=.
 1933  013656  032777  000002  165522            BIT     #BIT1,@TRSR
 1934  013664  001374                            BNE     A               ;BRANCH IF SET
 1935  013666  032777  000040  165512            BIT     #BIT5,@TRSR
 1936  013674  001770                            BEQ     A               ;BRANCH IF CLEAR
 1937  013676  052777  004000  165476            BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
 1938  013704  032777  004000  165470  65$:      BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
 1939  013712  001374                            BNE     65$             ;IF NO, WAIT FOR IT TO CLEAR
 1940  013714  004537  027546                    JSR     R5,TRYIT        ;;;
 1941  013720  001020                            .WORD   REWIND          ;FUNCTION
 1942  013722  000000                            .WORD   -0              ;WORD COUNT
 1943  013724  000000                            .WORD   0               ;BUS DDDRESS
 1944
 1945  013726  022777  003220  165446            CMP     #3220,@TRCR     ;TEST COMMAND REGISTER
 1946  013734  001412                            BEQ     A$0
 1947  013736  012705  003220                    MOV     #3220,R5
 1948  013742  017704  165434                    MOV     @TRCR,R4
 1949  013746  013737  001402  001244            MOV     TRCR,REGIST
 1950  013754  104001                            ERROR   1               ;INCORRECT REGISTER MATCHUP
 1951  013756  104403  006232                    TYPEF   ,MRWLPFL
 1952
 1953  013762  000001                  A$0:      $TAGA$=$TAGA$+1
 1954
 1955  013762  022777  002040  165416            CMP     #2040,@TRSR     ;TEST STATUS REGISTER
 1956  013770  001411                            BEQ     A$1
 1957  013772  012705  002040                    MOV     #2040,R5
 1958  013776  017704  165404                    MOV     @TRSR,R4
 19 3  014002  013737  001406  001244            MOV     TRSR,REGIST
 1960  014010  104001                            ERROR   1               ;INCORRECT REGISTER MATCHUP
 1961  014012  104404                            TYPEL
 1962
 1963  014014  000002                  A$1:      $TAGA$=$TAGA$+1
```

```
                                                ;+
                                                ;THIS ROUTINE TESTS THE FUNCTIONS OF ILLEGAL COMMANDS
                                                ;
                                        TESTJ:
014034  004737  013626                          JSR     PC,TESTI
                                                ;
014040  004737  031212                          JSR     PC,ENAINT
014044  012703  014254                          MOV     #ILCTAB,R3      ;POINT TO THE ILLEGAL INSTRUCTION CODE TABLE
014050  012704  000006                          MOV     #6,R4           ;COUNT THE NUMBER OF ILLEGAL INSTRUCTIONS
014054  012337  014064                  1$:     MOV     (R3)+,2$        ;INSERT AN ILLEGAL INSTRUCTION CODE
014060  004537  027546                          JSR     R5,TRYIT        ;GO DO THE FUNCTION
014064  000000                          2$:     .WORD                   ;LOCATION OF FUNCTION TO DO
014066  000000  000000                          .WORD   0,0             ;PARAMETERS TO TRYIT
014072  004737  014104                          JSR     PC,TESTJ1       ;CHECK IT OUT
014076  005304                                  DEC     R4              ;REDUCE THE COUNT. ARE WE DONE?
014100  001365                                  BNE     1$              ; IF NOT, CONTINUE
014102  000207                                  RTS     PC
                                                ;
014104  012737  000030  001240    TESTJ1: MOV     #30,HOLD        ;DELAY PAST TIME OUT
014112  013727  001240                          MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
014116  000000                          64$:    .WORD   0               ;USE THIS WORD AS A TIME COUNTER
014120                                  66$:
014120  005227  000000                          INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
014124  001375                                  BNE     66$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
014126  005337  014116                          DEC     64$             ;HAS THE TOTAL TIME ELAPSED?
014132  001372                                  BNE     66$             ;IF NO,GO WAIT A LITTLE LONGER
014134                                  65$:
014134  012737  000010  001240                  MOV     #10,HOLD        ;RESTORE NORMAL DELAY FACTOR
014142  017701  165234                          MOV     @TRCR,R1                ;GET COMMAND REGISTER
014146  042701  001436                          BIC     #1436,R1        ;CLEAR UNWANTED BITS
014152  022701  142200                          CMP     #142200,R1      ;TEST COMMAND REGISTER
014156  001411                                  BEQ     A$2
014160  012705  142200                          MOV     #142200,R5
014164  010104                                  MOV     R1,R4
014166  013737  001402  001244                  MOV     TRCR,REGIST
014174  104001                                  ERROR   1               ;INCORRECT REGISTER MATCHUP
014176  104403  006264                          TYPEF   ,MILFFL
                                                ;
                                                ;
014202  000003                          A$2:    $TAGA$=$TAGA$+1
014202  017701  165200                          MOV     @TRSR,R1                ;GET STATUS REGISTER
014206  022701  002040                          CMP     #2040,R1                ;TEST STATUS REGISTER
014212  001410                                  BEQ     A$3
014214  012705  002040                          MOV     #2040,R5
014220  010104                                  MOV     R1,R4
014222  013737  001406  001244                  MOV     TRSR,REGIST
014230  104001                                  ERROR   1               ;INCORRECT REGISTER MATCHUP
014232  104404                                  TYPEL
                                                ;
```

```
2020
2021  014234  000004                    A$3:    $TAGA$=$TAGA$+1
2022  014234  052777  004000  165140            BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
2023  014242  032777  004000  165132    64$:    BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
2024  014250  001374                            BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
2025  014252  000207                            RTS     PC
2026  014254  000406  001012  001414    ILCTAB: .WORD   ILC03,ILC05,ILC06,ILC11,ILC12,ILC14
2027  014262  000022  000424  001030
2028
2029                                    ;+-----------------------------------------------------------------
2030
2031                                    ;THIS ROUTINE TESTS THE DEROR CONDITIONS OF WRITING
2032                                    ;FROM LOAD POINT
2033
2034  014270                            TESTK:
2035  014270  004737  013626                    JSR     PC,TESTI
2036  014274  004537  027546                    JSR     R5,TRYIT        ;;;
2037  014300  000402                            .WORD   WRITE           ;FUNCTION
2038  014302  177700                            .WORD   -100            ;WORD COUNT
2039  014304  031600                            .WORD   OUTPUT          ;BUS DDDRESS
2040
2041  014306  022777  142602  165066            CMP     #142602,@TRCR   ;TEST CONTROL REGISTER
2042  014314  001412                            BEQ     A$4
2043  014316  012705  142602                    MOV     #142602,R5
2044  014322  017704  165054                    MOV     @TRCR,R4
2045  014326  013737  001402  001244            MOV     TRCR,REGIST
2046  014334  104001                            ERROR   1               ;INCORRECT REGISTER MATCHUP
2047  014336  104403  006310                    TYPEF   ,MWLPFL
2048
2049  014342  000005                    A$4:    $TAGA$=$TAGA$+1
2050
2051  014342  022777  002040  165036            CMP     #2040,@TRSR     ;TEST STATUS REGISTER
2052  014350  001411                            BEQ     A$5
2053  014352  012705  002040                    MOV     #2040,R5
2054  014356  017704  165024                    MOV     @TRSR,R4
2055  014362  013737  001406  001244            MOV     TRSR,REGIST
2056  014370  104001                            ERROR   1               ;INCORRECT REGISTER MATCHUP
2057  014372  104404                            TYPEL
2058
2059  014374  000006                    A$5:    $TAGA$=$TAGA$+1
2060
2061  014374  052777  004000  165000            BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
2062  014402  032777  004000  164772    64$:    BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
2063  014410  001374                            BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
2064  014412  000207                            RTS     PC
2065
2066                                    ;+-----------------------------------------------------------------
2067                                    ;THIS ROUTINE WRITES A END OF FILE MARK
2068                                    ;TAPE MUST BE POSITIONED PRIOR TO ENTERING THIS ROUTINE
2069
2070  014414                            RUNL:
2071  014414  004737  021712                    JSR     PC,TESTAH
2072  014420  012737  000062  014444            MOV     #50.,3$
2073  014426  005737  014444            2$:     TST     3$
2074  014432  001001                            BNE     1$
2075  014434  000207                            RTS     PC
```

```
2076  014436  005337  014444          1$:     DEC     3$
2077  014442  000401                          BR      4$
2078  014444  000000                  3$:     0
2079  014446                          4$:
2080
2081  014446                          TESTL:
2082  014446  004537  027546                  JSR     R5,TRYIT        ;;;
2083  014452  001034                          .WORD   WEOF            ;FUNCTION
2084  014454  000000                          .WORD   -0              ;WORD COUNT
2085  014456  000000                          .WORD   0               ;BUS DDDRESS
2086  014460  022777  003234  164714          CMP     #3234,@TRCR     ;TEST COMMAND REGISTER
2087  014466  001412                          BEQ     A$6
2088  014470  012705  003234                  MOV     #3234,R5
2089  014474  017704  164702                  MOV     @TRCR,R4
2090  014500  013737  001402  001244          MOV     TRCR,REGIST
2091  014506  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2092  014510  104403  006342                  TYPEF   ,MWEFFL
2093
2094  014514  000007                  A$6:    $TAGA$=$TAGA$+1
2095
2096  014514  022777  002011  164664          CMP     #2011,@TRSR             ;TEST THE STATUS REGISTER
2097  014522  001411                          BEQ     A$7
2098  014524  012705  002011                  MOV     #2011,R5
2099  014530  017704  164652                  MOV     @TRSR,R4
2100  014534  013737  001406  001244          MOV     TRSR,REGIST
2101  014542  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2102  014544  104404                          TYPEL
2103
2104  014546  000010                  A$7:    $TAGA$=$TAGA$+1
2105
2106
2107  014546  000207                          RTS     PC
2108
2109                                  ;+-------------------------------------------------------------------
2110
2111                                  ;THIS ROUTINE ROUTINE TESTS THE DEROR CONDITION OF WRITING A END
2112                                  ;OF FILE MARK FROM LOAD POINT.
2113
2114  014550                          TESTM:
2115  014550  004737  013626                  JSR     PC,TESTI
2116  014554  004537  027546                  JSR     R5,TRYIT        ;;;
2117  014560  001034                          .WORD   WEOF            ;FUNCTION
2118  014562  000000                          .WORD   -0              ;WORD COUNT
2119  014564  000000                          .WORD   0               ;BUS DDDRESS
2120  014566  022777  143234  164606          CMP     #143234,@TRCR   ;TEST COMMAND REGISTER
2121  014574  001412                          BEQ     A$10
2122  014576  012705  143234                  MOV     #143234,R5
2123  014602  017704  164574                  MOV     @TRCR,R4
2124  014606  013737  001402  001244          MOV     TRCR,REGIST
2125  014614  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2126  014616  104403  006367                  TYPEF   ,MWELPFL
2127
2128  014622  000011                  A$10:   $TAGA$=$TAGA$+1
2129
2130  014622  022777  002040  164556          CMP     #2040,@TRSR     ;TEST THE STATUS REGISTER
2131  014630  001411                          BEQ     A$11
```

```
 2132  014632  012705  002040                  MOV     #2040,R5
 2133  014636  017704  164544                  MOV     @TRSR,R4
 2134  014642  013737  001406  001244          MOV     TRSR,REGIST
 2135  014650  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
 2136  014652  104404                          TYPEL
 2137
 2138  014654  000012              A$11:       $TAGA$=$TAGA$+1
 2139
 2140  014654  004737  013234                  JSR     PC,TESTG
 2141  014660  000207                          RTS     PC
 2142
 2143
 2144
 2145                                  ;-----------------------------------------------------------------*
 2146
 2147                                  ;THIS ROUTINE TESTS FOR TAPE TRANSPORT ON LINE AND EXITS WHEN
 2148                                  ;IT IS.
 2149
 2150          014662                  A=.
 2151  014662  000207                          RTS     PC
 2152  014664                          TESTN:
 2153  014664  032777  002000  164514          BIT     #BIT10,@TRSR
 2154  014672  001373                          BNE     A               ;EXIT IF ON LINE
 2155  014674  104402  006435                  TYPE    ,MUOLN
 2156          014700                  A=.
 2157  014700  032777  002000  164500          BIT     #BIT10,@TRSR
 2158  014706  001007                          BNE     A$12
 2159  014710  005237  010560                  INC     TEMP
 2160  014714  001003                          BNE     64$
 2161  014716  012777  000207  164266          MOV     #207,@TPDBR     ;RING THE BELL
 2162  014724                          64$:
 2163  014724  000765                          BR      A
 2164  014726  000013              A$12:       $TAGA$=$TAGA$+1
 2165  014726  032777  100000  164446          BIT     #BIT15,@TRCR
 2166  014734  001403                          BEQ     A$13
 2167  014736  104000                          ERROR                   ;ERROR-BIT WAS NOT CORRECT
 2168  014740  104403  006520                  TYPEF   ,MOLEFL
 2169
 2170  014744  000014              A$13:       $TAGA$=$TAGA$+1
 2171  014744  052777  004000  164430          BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
 2172  014752  032777  004000  164422  64$:     BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
 2173  014760  001374                          BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
 2174  014762  000207                          RTS     PC
 2175
 2176                                  ;+-----------------------------------------------------------------
 2177                                  ;THIS ROUTINE IS USED TO TEST THE FUNCTION OF WRITING A ID
 2178                                  ;BLOCK FROM LOAD POINT THIS IS THE MAIN ID BLOCK ROUTINE
 2179
 2180  014764                          TESTO:
 2181  014764  004737  013626                  JSR     PC,TESTI
 2182  014770  004537  027546                  JSR     R5,TRYIT        ;;;
 2183  014774  000432                          .WORD   WIDB            ;FUNCTION
 2184  014776  000000                          .WORD   -0              ;WORD COUNT
 2185  015000  000000                          .WORD   0               ;BUS DDDRESS
 2186  015002  022777  002632  164372          CMP     #2632,@TRCR     ;TEST COMMAND REGISTER
 2187  015010  001412                          BEQ     A$14
```

```
2188  015012  012705  002632                  MOV     #2632,R5
2189  015016  017704  164360                  MOV     @TRCR,R4
2190  015022  013737  001402  001244          MOV     TRCR,RECIST
2191  015030  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2192  015032  104403  006541                  TYPEF   ,MWILPFL
2193
2194  015036  000015                  A$14:   $TAGA$=$TAGA$+1
2195
2196  015036  022777  002021  164342          CMP     #2021,@TRSR             ;TEST THE STATUS REGISTER
2197  015044  001411                          BEQ     A$15
2198  015046  012705  002021                  MOV     #2021,R5
2199  015052  017704  164330                  MOV     @TRSR,R4
2200  015056  013737  001406  001244          MOV     TRSR,RECIST
2201  015064  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2202  015066  104404                          TYPEL
2203
2204  015070  000016                  A$15:   $TAGA$=$TAGA$+1
2205
2206
2207  015070  000207                          RTS     PC
2208
2209
2210                                  ;-----------------------------------------------------------------------*
2211                                  ;THIS ROUTINE IS USED TO TAKE THE TAPE UNIT OFF LINE
2212                                  ;USING THE OFF LINE COMMAND
2213
2214
2215  015072                          TESTP:
2216  015072  004537  027546                  JSR     R5,TRYIT        ;;;
2217  015076  001436                          .WORD   OFLINE          ;FUNCTION
2218  015100  000000                          .WORD   -0              ;WORD COUNT
2219  015102  000000                          .WORD   0               ;BUS DDDRESS
2220
2221  015104  032777  002000  164274          BIT     #BIT10,@TRSR
2222  015112  001403                          BEQ     A$16
2223  015114  104000                          ERROR                   ;ERROR-BIT WAS NOT CORRECT
2224  015116  104403  006603                  TYPEF   ,MUNOFLN
2225
2226  015122  000017                  A$16:   $TAGA$=$TAGA$+1
2227  015122  032777  100000  164252          BIT     #BIT15,@TRCR
2228  015130  001003                          BNE     A$17
2229  015132  104000                          ERROR                   ;ERROR-BIT WAS NOT CORRECT
2230  015134  104403  006637                  TYPEF   ,MOFLEFL
2231
2232  015140  000020                  A$17:   $TAGA$=$TAGA$+1
2233  015140  052777  004000  164234          BIS     #BIT11,@TRCR
2234  015146  013727  001240                  MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
2235  015152  000000                  64$:    .WORD   0               ;USE THIS WORD AS A TIME COUNTER
2236  015154                          66$:
2237  015154  005227  000000                  INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
2238  015160  001375                          BNE     66$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2239  015162  005337  015152                  DEC     64$             ;HAS THE TOTAL TIME ELAPSED?
2240  015166  001372                          BNE     66$             ;IF NO,GO WAIT A LITTLE LONGER
2241  015170                          65$:
2242  015170  004737  014664                  JSR     PC,TESTN
2243  015174  052777  004000  164200          BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
```

```
2244  015202  032777  004000  164172  67$:    BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
2245  015210  001374                          BNE     67$             ;IF NO, WAIT FOR IT TO CLEAR
2246  015212  000207                          RTS     PC
2247
2248
2249                                  ;-----------------------------------------------------------------------*
2250                                  ;THIS ROUTINE TESTS THE OFF LINE FUNCTION OF POWER CLEAR WHILE
2251                                  ;REWINDING THE TAPE
2252
2253  015214                          TESTQ:
2254  015214  012737  001026  027536          MOV     #GOEOT,USEA             ;SET FUNCTION
2255  015222  012737  000000  027540          MOV     #-0,USEB        ;SET WORD COUNT
2256  015230  012737  000000  027542          MOV     #0,USEC         ;SET BUS ADDRESS
2257  015236  004737  030012                  JSR     PC,EOTTST
2258  015242  013727  001240                  MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
2259  015246  000000                  64$:    .WORD   0               ;USE THIS WORD AS A TIME COUNTER
2260  015250                          66$:
2261  015250  005227  000000                  INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
2262  015254  001375                          BNE     66$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2263  015256  005337  015246                  DEC     64$             ;HAS THE TOTAL TIME ELAPSED?
2264  015262  001372                          BNE     66$             ;IF NO,GO WAIT A LITTLE LONGER
2265  015264                          65$:
2266  015264  013727  001240                  MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
2267  015270  000000                  67$:    .WORD   0               ;USE THIS WORD AS A TIME COUNTER
2268  015272                          69$:
2269  015272  005227  000000                  INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
2270  015276  001375                          BNE     69$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2271  015300  005337  015270                  DEC     67$             ;HAS THE TOTAL TIME ELAPSED?
2272  015304  001372                          BNE     69$             ;IF NO,GO WAIT A LITTLE LONGER
2273  015306                          68$:
2274  015306  013727  001240                  MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
2275  015312  000000                  70$:    .WORD   0               ;USE THIS WORD AS A TIME COUNTER
2276  015314                          72$:
2277  015314  005227  000000                  INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
2278  015320  001375                          BNE     72$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2279  015322  005337  015312                  DEC     70$             ;HAS THE TOTAL TIME ELAPSED?
2280  015326  001372                          BNE     72$             ;IF NO,GO WAIT A LITTLE LONGER
2281  015330                          71$:
2282  015330  052777  004000  164044          BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
2283  015336  032777  004000  164036  73$:    BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
2284  015344  001374                          BNE     73$             ;IF NO, WAIT FOR IT TO CLEAR
2285  015346  004537  027546                  JSR     R5,TRYIT        ;;
2286  015352  001020                          .WORD   REWIND          ;FUNCTION
2287  015354  000000                          .WORD   -0              ;WORD COUNT
2288  015356  000000                          .WORD   0               ;BUS DDDRESS
2289  015360  052777  004000  164014          BIS     #BIT11,@TRCR
2290
2291  015366  013727  001240                  MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
2292  015372  000000                  74$:    .WORD   0               ;USE THIS WORD AS A TIME COUNTER
2293  015374                          76$:
2294  015374  005227  000000                  INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
2295  015400  001375                          BNE     76$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2296  015402  005337  015372                  DEC     74$             ;HAS THE TOTAL TIME ELAPSED?
2297  015406  001372                          BNE     76$             ;IF NO,GO WAIT A LITTLE LONGER
2298  015410                          75$:
2299  015410  022777  004020  163764          CMP     #4020,@TRCR     ;TEST COMMAND REGISTER
```

```
2300  015416  001412                          BEQ     A$20
2301  015420  012705  000020                  MOV     #20,R5
2302  015424  017704  163752                  MOV     @TRCR,R4
2303  015430  013737  001402  001244          MOV     TRCR,REGIST
2304  015436  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2305  015440  104403  006661                  TYPEF   ,MRWOFL
2306
2307  015444  000021                  A$20:   $TAGA$=$TAGA$+1
2308
2309  015444  022777  000001  163734          CMP     #1,@TRSR                ;TEST THE STATUS REGISTER
2310  015452  001411                          BEQ     A$21
2311  015454  012705  000001                  MOV     #1,R5
2312  015460  017704  163722                  MOV     @TRSR,R4
2313  015464  013737  001406  001244          MOV     TRSR,REGIST
2314  015472  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2315  015474  104404                          TYPEL
2316
2317  015476  000022                  A$21:   $TAGA$=$TAGA$+1
2318
2319  015476  005777  163714                  TST     @TRBA           ;TEST BUFFER ADDRESS
2320  015502  001411                          BEQ     A$22
2321  015504  012705  000000                  MOV     #0,R5
2322  015510  017704  163702                  MOV     @TRBA,R4
2323  015514  013737  001416  001244          MOV     TRBA,REGIST
2324  015522  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2325  015524  104404                          TYPEL
2326
2327  015526  000023                  A$22:   $TAGA$=$TAGA$+1
2328
2329  015526  005777  163660                  TST     @TRWC           ;TEST THE WORD COUNT
2330  015532  001411                          BEQ     A$23
2331  015534  012705  000000                  MOV     #0,R5
2332  015540  017704  163646                  MOV     @TRWC,R4
2333  015544  013737  001412  001244          MOV     TRWC,REGIST
2334  015552  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2335  015554  104404                          TYPEL
2336
2337  015556  000024                  A$23:   $TAGA$=$TAGA$+1
2338
2339          015556                  A=.
2340  015556  032777  000002  163622          BIT     #BIT1,@TRSR
2341  015564  001374                          BNE     A               ;WAIT FOR REWIND TO FINISH
2342  015566  032777  000040  163612          BIT     #BIT5,@TRSR
2343  015574  001770                          BEQ     A               ;WAIT FOR LOADPOINT TO SET
2344  015576  004737  014664                  JSR     PC,TESTN
2345  015602  052777  004000  163572          BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
2346  015610  032777  004000  163564  64$:    BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
2347  015616  001374                          BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
2348  015620  000207                          RTS     PC
2349
2350
2351
2352
2353                                  ;-----------------------------------------------------------------------*
2354                                  ;THIS ROUTINE CHECKS THE DATA JUST READ BY THE
2355                                  ;MAG TAPE UNIT AND REPORTS ANY DERORS THAT EXIST
```

```
                                        ;THIS ROUTINE ALSO CLEARS THE INPUT AREA AFTER TESTING
                                        ;THE DATA
                                        
015622  012701  035640          TESTS:  MOV     #INPUT,R1       ;SET INDEX
015626  012702  031600                  MOV     #OUTPUT,R2      ;SET INDEX
015632  013703  027540                  MOV     USEB,R3         ;GET THE WORD COUNT
        015636                  A=.
015636  021112                          CMP     @R1,@R2         ;TEST INPUT
015640  001410                          BEQ     A$24
015642  011205                          MOV     @R2,R5
015644  011104                          MOV     @R1,R4
015646  010137  001244                  MOV     R1,REGIST
015652  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
015654  104402  006704                  TYPE    ,MRDERR
015660  010200                          MOV     R2,R0
                                        ;------------- CONVRT CALL SHOULD GO HERE
                                        
015662  000025                  A$24:   $TAGA$=$TAGA$+1
015662  022122                          CMP     (R1)+,(R2)+     ;ADVANCE INDEXES
015664  005203                          INC     R3              ;INCREMENT WORD COUNT
015666  001363                          BNE     A               ;CONTINUE TESTING INPUT
                                        
015670  012700  035640                  MOV     #INPUT,R0       ;SET INDEX
015674  012701  177000                  MOV     #-1000,R1       ;SET COUNTER
015700  000026                  A$25:   $TAGA$=$TAGA$+1
        015700                  A=.
015700  005020                          CLR     (R0)+           ;CLEAR INPUT AREA
015702  001376                          BNE     A               ;CONTINUE CLEARING
015704  000207                          RTS     PC
                                        
                                        
                                        
                                        
                                        
                                        
                                        
                                        ;+-----------------------------------------------------------------------
                                        ;THIS ROUTINE CHECKS THE FUNCTION OF SPACING REVERSE OVER A
                                        ;END OF FILE
                                        
015706                          RUNU:
015706  012737  000062  015732                  MOV     #50.,3$
015714  005737  015732          2$:     TST     3$
015720  001001                          BNE     1$
015722  000207                          RTS     PC
015724  005337  015732          1$:     DEC     3$
015730  000401                          BR      4$
015732  000000                  3$:     0
015734                          4$:
                                        
015734                          TESTU:
015734  004537  027546                  JSR     R5,TRYIT        ;;;
015740  001410                          .WORD   SPACER          ;FUNCTION
015742  000000                          .WORD   -0              ;WORD COUNT
015744  000000                          .WORD   0               ;BUS DDDRESS
```

```
2412  015746  022777  003610  163426          CMP     #3610,@TRCR     ;TEST COMMAND REGISTER
2413  015754  001412                          BEQ     A$26
2414  015756  012705  003610                  MOV     #3610,R5
2415  015762  017704  163414                  MOV     @TRCR,R4
2416  015766  013737  001402  001244          MOV     TRCR,REGIST
2417  015774  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2418  015776  104403  006727                  TYPEF   ,MSREFFL
2419
2420  016002  000027                  A$26:   $TAGA$=$TAGA$+1
2421
2422  016002  022777  002011  163376          CMP     #2011,@TRSR     ;TEST THE STATUS REGISTER
2423  016010  001411                          BEQ     A$27
2424  016012  012705  002011                  MOV     #2011,R5
2425  016016  017704  163364                  MOV     @TRSR,R4
2426  016022  013737  001406  001244          MOV     TRSR,REGIST
2427  016030  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2428  016032  104404                          TYPEL
2429
2430  016034  000030                  A$27:   $TAGA$=$TAGA$+1
2431
2432  016034  005777  163356                  TST     @TRBA           ;TEST BUFFER ADDRESS
2433  016040  001411                          BEQ     A$30
2434  016042  012705  000000                  MOV     #0,R5
2435  016046  017704  163344                  MOV     @TRBA,R4
2436  016052  013737  001416  001244          MOV     TRBA,REGIST
2437  016060  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2438  016062  104404                          TYPEL
2439
2440  016064  000031                  A$30:   $TAGA$=$TAGA$+1
2441
2442  016064  005777  163322                  TST     @TRWC           ;TEST THE WORD COUNT
2443  016070  001411                          BEQ     A$31
2444  016072  012705  000000                  MOV     #0,R5
2445  016076  017704  163310                  MOV     @TRWC,R4
2446  016102  013737  001412  001244          MOV     TRWC,REGIST
2447  016110  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2448  016112  104404                          TYPEL
2449
2450  016114  000032                  A$31:   $TAGA$=$TAGA$+1
2451
2452  016114  000207                          RTS     PC
2453
2454                                  ;+-------------------------------------------------------------------
2455                                  ;THIS ROUTINE CHECKS THE FUNCTION OF READING A END OF FILE
2456
2457  016116                          RUNV:
2458  016116  012737  000062  016142          MOV     #50.,3$
2459  016124  005737  016142          2$:     TST     3$
2460  016130  001001                          BNE     1$
2461  016132  000207                          RTS     PC
2462  016134  005337  016142          1$:     DEC     3$
2463  016140  000401                          BR      4$
2464  016142  000000                  3$:     0
2465  016144                          4$:
2466
2467  016144                          TESTV:
```

```
2468  016144  004537  027546            JSR     R5,TRYIT        ;;;
2469  016150  000004                    .WORD   READ            ;FUNCTION
2470  016152  000000                    .WORD   -0              ;WORD COUNT
2471  016154  035640                    .WORD   INPUT           ;BUS DDDRESS
2472  016156  022777  002204  163216    CMP     #2204,@TRCR     ;TEST COMMAND REGISTER
2473  016164  001412                    BEQ     A$32
2474  016166  012705  002204            MOV     #2204,R5
2475  016172  017704  163204            MOV     @TRCR,R4
2476  016176  013737  001402  001244    MOV     TRCR,REGIST
2477  016204  104001                    ERROR   1               ;INCORRECT REGISTER MATCHUP
2478  016206  104403  006771            TYPEF   ,MREFFL
2479
2480  016212  000033            A$32:   $TAGA$=$TAGA$+1
2481
2482  016212  022777  002011  163166    CMP     #2011,@TRSR     ;TEST THE STATUS REGISTER
2483  016220  001410                    BEQ     A$33
2484  016222  012705  002011            MOV     #2011,R5
2485  016226  017704  163154            MOV     @TRSR,R4
2486  016232  013737  001406  001244    MOV     TRSR,REGIST
2487  016240  104001                    ERROR   1               ;INCORRECT REGISTER MATCHUP
2488
2489  016242  000034            A$33:   $TAGA$=$TAGA$+1
2490
2491  016242  022777  035640  163146    CMP     #INPUT,@TRBA    ;TEST BUFFER ADDRESS
2492  016250  001410                    BEQ     A$34
2493  016252  012705  035640            MOV     #INPUT,R5
2494  016256  017704  163134            MOV     @TRBA,R4
2495  016262  013737  001416  001244    MOV     TRBA,REGIST
2496  016270  104001                    ERROR   1               ;INCORRECT REGISTER MATCHUP
2497
2498  016272  000035            A$34:   $TAGA$=$TAGA$+1
2499
2500  016272  005777  163114            TST     @TRWC           ;TEST THE WORD COUNT
2501  016276  001410                    BEQ     A$35
2502  016300  012705  000000            MOV     #0,R5
2503  016304  017704  163102            MOV     @TRWC,R4
2504  016310  013737  001412  001244    MOV     TRWC,REGIST
2505  016316  104001                    ERROR   1               ;INCORRECT REGISTER MATCHUP
2506  016320  000036            A$35:   $TAGA$=$TAGA$+1
2507
2508  016320  000207                    RTS     PC
2509
2510                            ;-----------------------------------------------------------------*
2511                            ;THIS ROUTINE TESTS THE FUNCTION OF THE TIME OUT WHILE WRITING
2512
2513  016322                    TESTW:
2514  016322  004737  021712            JSR     PC,TESTAH
2515  016326  042777  000001  163052    BIC     #BIT0,@TRSR
2516  016334  005077  163052            CLR     @TRWC           ;SET WORD COUNT TO MAX
2517  016340  012777  031600  163050    MOV     #OUTPUT,@TRBA   ;SET BUS ADDRESS AT START
2518  016346  012777  000402  163026    MOV     #WRITE,@TRCR    ;SET COMMAND REGISTER
2519  016354  052777  000001  163020    BIS     #BIT0,@TRCR     ;SET THE GO BIT
2520  016362  005037  010560            CLR     TEMP            ;CLEAR THE TIMER
2521          016366            A=.
2522  016366  005077  163020            CLR     @TRWC           ;RESET THE WORD COUNT
2523  016372  012777  031600  163016    MOV     #OUTPUT,@TRBA   ;RESET THE BUS ADDRESS
```

```
2524  016400  005237  010560              INC     TEMP            ;INCREMENT TIME
2525  016404  001370                      BNE     A               ;DELAY MORE
2526          016406              A=.
2527  016406  005077  163000              CLR     @TRWC           ;RESET THE WORD COUNT
2528  016412  012777  031600  162776      MOV     #OUTPUT,@TRBA   ;RESET THE BUS ADDRESS
2529  016420  005237  010560              INC     TEMP            ;INCREMENT TIME
2530  016424  001370                      BNE     A               ;DELAY MORE
2531          016426              A=.
2532  016426  005077  162760              CLR     @TRWC           ;RESET THE WORD COUNT
2533  016432  012777  031600  162756      MOV     #OUTPUT,@TRBA   ;RESET THE BUS ADDRESS
2534  016440  005237  010560              INC     TEMP            ;INCREMENT TIME
2535  016444  001370                      BNE     A               ;DELAY MORE
2536  016446  022777  102602  162726      CMP     #102602,@TRCR   ;TEST COMMAND REGISTER
2537  016454  001412                      BEQ     A$36
2538  016456  012705  102602              MOV     #102602,R5
2539  016462  017704  162714              MOV     @TRCR,R4
2540  016466  013737  001402  001244      MOV     TRCR,REGIST
2541  016474  104001                      ERROR   1               ;INCORRECT REGISTER MATCHUP
2542  016476  104403  007015              TYPEF   ,MWTOFL
2543
2544  016502  000037              A$36:   $TAGA$=$TAGA$+1
2545
2546  016502  017701  162700              MOV     @TRSR,R1                ;GET THE STATUS
2547  016506  042701  170000              BIC     #170000,R1      ;CLEAR BAD BITS
2548  016512  022701  002401              CMP     #2401,R1        ;TEST THE STATUS REGISTER
2549  016516  001411                      BEQ     A$37
2550  016520  012705  002401              MOV     #2401,R5
2551  016524  017704  162656              MOV     @TRSR,R4
2552  016530  013737  001406  001244      MOV     TRSR,REGIST
2553  016536  104001                      ERROR   1               ;INCORRECT REGISTER MATCHUP
2554  016540  104404                      TYPEL
2555
2556  016542  000040              A$37:   $TAGA$=$TAGA$+1
2557
2558  016542  022777  031600  162646      CMP     #OUTPUT,@TRBA   ;TEST BUFFER ADDRESS
2559  016550  001411                      BEQ     A$40
2560  016552  012705  031600              MOV     #OUTPUT,R5
2561  016556  017704  162634              MOV     @TRBA,R4
2562  016562  013737  001416  001244      MOV     TRBA,REGIST
2563  016570  104001                      ERROR   1               ;INCORRECT REGISTER MATCHUP
2564  016572  104404                      TYPEL
2565
2566  016574  000041              A$40:   $TAGA$=$TAGA$+1
2567
2568  016574  005777  162612              TST     @TRWC           ;TEST THE WORD COUNT
2569  016600  001411                      BEQ     A$41
2570  016602  012705  000000              MOV     #0,R5
2571  016606  017704  162600              MOV     @TRWC,R4
2572  016612  013737  001412  001244      MOV     TRWC,REGIST
2573  016620  104001                      ERROR   1               ;INCORRECT REGISTER MATCHUP
2574  016622  104404                      TYPEL
2575
2576  016624  000042              A$41:   $TAGA$=$TAGA$+1
2577
2578  016624  052777  004000  162550      BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
2579  016632  032777  004000  162542  64$:  BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
```

```
2580  016640  001374                        BNE     64$                 ;IF NO, WAIT FOR IT TO CLEAR
2581  016642  004737  014764                JSR     PC,TESTO
2582  016646  000207                        RTS     PC
2583
2584                                  ;-------------------------------------------------------------------------*
2585                                  ;THIS ROUTINE CHECKS THE FUNCTION OF THE TIME OUT WHILE READING
2586
2587  016650                          TESTX:
2588  016650  004737  021712                JSR     PC,TESTAH
2589  016654  004737  026560                JSR     PC,TESTCA
2590  016660  004737  026650                JSR     PC,TESTFO
2591  016664  004737  026650                JSR     PC,TESTFO
2592  016670  042777  000001  162510        BIC     #BIT0,@TRSR
2593  016676  005077  162510                CLR     @TRWC               ;SET WORD COUNT TO MAX
2594  016702  012777  035640  162506        MOV     #INPUT,@TRBA        ;SET BUS ADDRESS AT START
2595  016710  012777  001410  162464        MOV     #SPACER,@TRCR       ;SET COMMAND REGISTER
2596  016716  052777  000001  162456        BIS     #BIT0,@TRCR         ;SET THE GO BIT
2597  016724  004737  012524                JSR     PC,TESTA
2598
2599  016730  022777  102050  162444        CMP     #102050,@TRCR       ;TEST COMMAND REGISTER
2600  016736  001412                        BEQ     A$42
2601  016740  012705  102050                MOV     #102050,R5
2602  016744  017704  162432                MOV     @TRCR,R4
2603  016750  013737  001402  001244        MOV     TRCR,REGIST
2604  016756  104001                        ERROR   1                   ;INCORRECT REGISTER MATCHUP
2605  016760  104403  007037                TYPEF   ,MRTOFL
2606
2607  016764  000043                  A$42: $TAGA$=$TAGA$+1
2608
2609  016764  022777  002400  162414        CMP     #2400,@TRSR         ;TEST THE STATUS REGISTER
2610  016772  001411                        BEQ     A$43
2611  016774  012705  002400                MOV     #2400,R5
2612  017000  017704  162402                MOV     @TRSR,R4
2613  017004  013737  001406  001244        MOV     TRSR,REGIST
2614  017012  104001                        ERROR   1                   ;INCORRECT REGISTER MATCHUP
2615  017014  104404                        TYPEL
2616
2617  017016  000044                  A$43: $TAGA$=$TAGA$+1
2618
2619  017016  022777  035640  162372        CMP     #INPUT,@TRBA        ;TEST BUFFER ADDRESS
2620  017024  001411                        BEQ     A$44
2621  017026  012705  035640                MOV     #INPUT,R5
2622  017032  017704  162360                MOV     @TRBA,R4
2623  017036  013737  001416  001244        MOV     TRBA,REGIST
2624  017044  104001                        ERROR   1                   ;INCORRECT REGISTER MATCHUP
2625  017046  104404                        TYPEL
2626
2627  017050  000045                  A$44: $TAGA$=$TAGA$+1
2628
2629  017050  005777  162336                TST     @TRWC               ;TEST THE WORD COUNT
2630  017054  001411                        BEQ     A$45
2631  017056  012705  000000                MOV     #0,R5
2632  017062  017704  162324                MOV     @TRWC,R4
2633  017066  013737  001412  001244        MOV     TRWC,REGIST
2634  017074  104001                        ERROR   1                   ;INCORRECT REGISTER MATCHUP
2635  017076  104404                        TYPEL
```

```
2636
2637  017100  000046                    A$45:   $TAGA$=$TAGA$+1
2638
2639  017100  052777  004000  162274            BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
2640  017106  032777  004000  162266  64$:      BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
2641  017114  001374                            BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
2642  017116  000207                            RTS     PC
2643
2644
2645                                    ;-----------------------------------------------------------------*
2646                                    ;THIS ROUTINE FORCES A NXM AND BGL DEROR WHILE WRITING
2647
2648  017120                            TESTY:
2649  017120  004737  021712                    JSR     PC,TESTAH
2650  017124  004537  027546                    JSR     R5,TRYIT        ;;
2651  017130  000402                            .WORD   WRITE           ;FUNCTION
2652  017132  000000                            .WORD   -0              ;WORD COUNT
2653  017134  177777                            .WORD   -1              ;BUS DDDRESS
2654  017136  013727  001240                    MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
2655  017142  000000            64$:            .WORD   0               ;USE THIS WORD AS A TIME COUNTER
2656  017144                    66$:
2657  017144  005227  000000                    INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
2658  017150  001375                            BNE     66$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2659  017152  005337  017142                    DEC     64$             ;HAS THE TOTAL TIME ELAPSED?
2660  017156  001372                            BNE     66$             ;IF NO,GO WAIT A LITTLE LONGER
2661  017160                    65$:
2662  017160  022777  112602  162214            CMP     #112602,@TRCR   ;TEST COMMAND REGISTER
2663  017166  001412                            BEQ     A$46
2664  017170  012705  112602                    MOV     #112602,R5
2665  017174  017704  162202                    MOV     @TRCR,R4
2666  017200  013737  001402  001244            MOV     TRCR,REGIST
2667  017206  104001                            ERROR   1               ;INCORRECT REGISTER MATCHUP
2668  017210  104403  007060                    TYPEF   ,MWNXFL
2669
2670  017214  000047                    A$46:   $TAGA$=$TAGA$+1
2671
2672  017214  022777  006101  162164            CMP     #6101,@TRSR     ;TEST THE STATUS REGISTER
2673  017222  001411                            BEQ     A$47
2674  017224  012705  006101                    MOV     #6101,R5
2675  017230  017704  162152                    MOV     @TRSR,R4
2676  017234  013737  001406  001244            MOV     TRSR,REGIST
2677  017242  104001                            ERROR   1               ;INCORRECT REGISTER MATCHUP
2678  017244  104404                            TYPEL
2679
2680  017246  000050                    A$47:   $TAGA$=$TAGA$+1
2681
2682  017246  022777  000001  162142            CMP     #1,@TRBA                ;TEST BUFFER ADDRESS
2683  017254  001411                            BEQ     A$50
2684  017256  012705  000001                    MOV     #1,R5
2685  017262  017704  162130                    MOV     @TRBA,R4
2686  017266  013737  001416  001244            MOV     TRBA,REGIST
2687  017274  104001                            ERROR   1               ;INCORRECT REGISTER MATCHUP
2688  017276  104404                            TYPEL
2689
2690  017300  000051                    A$50:   $TAGA$=$TAGA$+1
2691
```

```
2692  017300  022777  000001  162104          CMP     #1,@TRWC                ;TEST THE WORD COUNT
2693  017306  001411                          BEQ     A$51
2694  017310  012705  000001                  MOV     #1,R5
2695  017314  017704  162072                  MOV     @TRWC,R4
2696  017320  013737  001412  001244          MOV     TRWC,REGIST
2697  017326  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2698  017330  104404                          TYPEL
2699
2700  017332  000052                  A$51:   $TAGA$=$TAGA$+1
2701
2702  017332  000207                          RTS     PC
2703
2704                                  ;-----------------------------------------------------------------*
2705                                  ;THIS ROUTINE FORCES A NXM DEROR USING BIT 12
2706
2707  017334                          TESTZ:
2708  017334  004737  021712                  JSR     PC,TESTAH
2709  017340  012777  000402  162034          MOV     #WRITE,@TRCR    ;SET COMMAND
2710  017346  052777  010000  162026          BIS     #BIT12,@TRCR
2711  017354  042777  000001  162024          BIC     #BIT0,@TRSR
2712  017362  005077  162024                  CLR     @TRWC           ;CLEAR THE WORD COUNT
2713  017366  005077  162024                  CLR     @TRBA           ;CLEAR BUS ADDRESS
2714  017372  005277  162004                  INC     @TRCR           ;SET GO BIT
2715
2716  017376  013727  001240                  MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
2717  017402  000000                  64$:    .WORD   0               ;USE THIS WORD AS A TIME COUNTER
2718  017404                          66$:
2719  017404  005227  000000                  INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
2720  017410  001375                          BNE     66$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2721  017412  005337  017402                  DEC     64$             ;HAS THE TOTAL TIME ELAPSED?
2722  017416  001372                          BNE     66$             ;IF NO,GO WAIT A LITTLE LONGER
2723  017420                          65$:
2724  017420  022777  112602  161754          CMP     #112602,@TRCR   ;TEST COMMAND REGISTER
2725  017426  001412                          BEQ     A$52
2726  017430  012705  112602                  MOV     #112602,R5
2727  017434  017704  161742                  MOV     @TRCR,R4
2728  017440  013737  001402  001244          MOV     TRCR,REGIST
2729  017446  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2730  017450  104403  007115                  TYPEF   ,MBNXFL
2731
2732  017454  000053                  A$52:   $TAGA$=$TAGA$+1
2733
2734  017454  022777  006101  161724          CMP     #6101,@TRSR     ;TEST THE STATUS REGISTER
2735  017462  001411                          BEQ     A$53
2736  017464  012705  006101                  MOV     #6101,R5
2737  017470  017704  161712                  MOV     @TRSR,R4
2738  017474  013737  001406  001244          MOV     TRSR,REGIST
2739  017502  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2740  017504  104404                          TYPEL
2741
2742  017506  000054                  A$53:   $TAGA$=$TAGA$+1
2743
2744  017506  022777  000002  161702          CMP     #2,@TRBA                ;TEST BUFFER ADDRESS
2745  017514  001411                          BEQ     A$54
2746  017516  012705  000002                  MOV     #2,R5
2747  017522  017704  161670                  MOV     @TRBA,R4
```

```
2748  017526  013737  001416  001244          MOV     TRBA,REGIST
2749  017534  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2750  017536  104404                          TYPEL
2751
2752  017540  000055                  A$54:   $TAGA$=$TAGA$+1
2753
2754  017540  022777  000001  161644          CMP     #1,@TRWC                ;TEST THE WORD COUNT
2755  017546  001411                          BEQ     A$55
2756  017550  012705  000001                  MOV     #1,R5
2757  017554  017704  161632                  MOV     @TRWC,R4
2758  017560  013737  001412  001244          MOV     TRWC,REGIST
2759  017566  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2760  017570  104404                          TYPEL
2761
2762  017572  000056                  A$55:   $TAGA$=$TAGA$+1
2763
2764  017572  052777  004000  161602          BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
2765  017600  032777  004000  161574  64$:    BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
2766  017606  001374                          BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
2767  017610  000207                          RTS     PC
2768
2769                                  ;-------------------------------------------------------------------------*
2770                                  ;THIS ROUTINE FORCES A NXM DEROR USING BIT 13
2771
2772  017612                          TESTAA:
2773  017612  004737  021712                  JSR     PC,TESTAH
2774  017616  012777  000402  161556          MOV     #WRITE,@TRCR    ;SET COMMAND
2775  017624  052777  020000  161550          BIS     #BIT13,@TRCR
2776  017632  042777  000001  161546          BIC     #BIT0,@TRSR
2777  017640  005077  161546                  CLR     @TRWC           ;CLEAR THE WORD COUNT
2778  017644  005077  161546                  CLR     @TRBA           ;CLEAR BUS ADDRESS
2779  017650  005277  161526                  INC     @TRCR           ;SET GO BIT
2780
2781  017654  013727  001240                  MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
2782  017660  000000                  64$:    .WORD   0               ;USE THIS WORD AS A TIME COUNTER
2783  017662                          66$:
2784  017662  005227  000000                  INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
2785  017666  001375                          BNE     66$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2786  017670  005337  017660                  DEC     64$             ;HAS THE TOTAL TIME ELAPSED?
2787  017674  001372                          BNE     66$             ;IF NO,GO WAIT A LITTLE LONGER
2788  017676                          65$:
2789  017676  022777  122602  161476          CMP     #122602,@TRCR   ;TEST COMMAND REGISTER
2790  017704  001412                          BEQ     A$56
2791  017706  012705  122602                  MOV     #122602,R5
2792  017712  017704  161464                  MOV     @TRCR,R4
2793  017716  013737  001402  001244          MOV     TRCR,REGIST
2794  017724  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2795  017726  104403  007157                  TYPEF   ,MNXBFL
2796
2797  017732  000057                  A$56:   $TAGA$=$TAGA$+1
2798
2799  017732  022777  006101  161446          CMP     #6101,@TRSR     ;TEST THE STATUS REGISTER
2800  017740  001411                          BEQ     A$57
2801  017742  012705  006101                  MOV     #6101,R5
2802  017746  017704  161434                  MOV     @TRSR,R4
2803  017752  013737  001406  001244          MOV     TRSR,REGIST
```

```
2804  017760  104001                        ERROR   1               ;INCORRECT REGISTER MATCHUP
2805  017762  104404                        TYPEL
2806
2807  017764  000060              A$57:     $TAGA$=$TAGA$+1
2808
2809  017764  022777  000002  161424        CMP     #2,@TRBA                ;TEST BUFFER ADDRESS
2810  017772  001411                        BEQ     A$60
2811  017774  012705  000002                MOV     #2,R5
2812  020000  017704  161412                MOV     @TRBA,R4
2813  020004  013737  001416  001244        MOV     TRBA,REGIST
2814  020012  104001                        ERROR   1               ;INCORRECT REGISTER MATCHUP
2815  020014  104404                        TYPEL
2816
2817  020016  000061              A$60:     $TAGA$=$TAGA$+1
2818
2819  020016  022777  000001  161366        CMP     #1,@TRWC                ;TEST THE WORD COUNT
2820  020024  001411                        BEQ     A$61
2821  020026  012705  000001                MOV     #1,R5
2822  020032  017704  161354                MOV     @TRWC,R4
2823  020036  013737  001412  001244        MOV     TRWC,REGIST
2824  020044  104001                        ERROR   1               ;INCORRECT REGISTER MATCHUP
2825  020046  104404                        TYPEL
2826
2827  020050  000062              A$61:     $TAGA$=$TAGA$+1
2828
2829  020050  052777  004000  161324        BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
2830  020056  032777  004000  161316  64$:  BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
2831  020064  001374                        BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
2832  020066  000207                        PTS     PC
2833
2834                                        ;-------------------------------------------------------------------------*
2835                                        ;THIS ROUTINE FORCFS A NXM DEROR USING BIT 12-13
2836
2837  020070                              TESTAB:
2838  020070  004737  021712                JSR     PC,TESTAH
2839  020074  012777  000402  161300        MOV     #WRITE,@TRCR    ;SET COMMAND
2840  020102  052777  030000  161272        BIS     #BIT12!BIT13,@TRCR ;SET THE EXTENDED ADDRESS BITS
2841  020110  042777  000001  161270        BIC     #BIT0,@TRSR
2842  020116  005077  161270                CLR     @TRWC           ;CLEAR THE WORD COUNT
2843  020122  012777  160000  161266        MOV     #160000,@TRBA   ;SET UP AN I/O PAGE REFERENCE
2844  020130  005277  161246                INC     @TRCR           ;SET GO BIT
2845  020134  013727  001240                MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
2846  020140  000000              64$:      .WORD   0               ;USE THIS WORD AS A TIME COUNTER
2847  020142                      66$:
2848  020142  005227  000000                INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
2849  020146  001375                        BNE     66$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2850  020150  005337  020140                DEC     64$             ;HAS THE TOTAL TIME ELAPSED?
2851  020154  001372                        BNE     66$             ;IF NO,GO WAIT A LITTLE LONGER
2852  020156                      65$:
2853  020156  022777  132602  161216        CMP     #132602,@TRCR   ;TEST COMMAND REGISTER
2854  020164  001412                        BEQ     A$62
2855  020166  012705  132602                MOV     #132602,R5
2856  020172  017704  161204                MOV     @TRCR,R4
2857  020176  013737  001402  001244        MOV     TRCR,REGIST
2858  020204  104001                        ERROR   1               ;INCORRECT REGISTER MATCHUP
2859  020206  104403  007227                TYPEF   ,MNXMFL
```

```
2860  020212  000063                    A$62:   $TAGA$=$TAGA$+1
2861  020212  022777  006101  161166            CMP     #6101,@TRSR     ;TEST THE STATUS REGISTER
2862  020220  001411                            BEQ     A$63
2863  020222  012705  006101                    MOV     #6101,R5
2864  020226  017704  161154                    MOV     @TRSR,R4
2865  020232  013737  001406  001244            MOV     TPSR,REGIST
2866  020240  104001                            ERROR   1               ;INCORRECT REGISTER MATCHUP
2867  020242  104404                            TYPEL
2868
2869  020244  000064                    A$63:   $TAGA$=$TAGA$+1
2870
2871  020244  032777  000002  161144            BIT     #2,@TRBA                ;TEST BUFFER ADDRESS
2872  020252  001011                            BNE     A$64
2873  020254  012705  000002                    MOV     #2,R5
2874  020260  017704  161132                    MOV     @TRBA,R4
2875  020264  013737  001416  001244            MOV     TRBA,REGIST
2876  020272  104001                            ERROR   1               ;INCORRECT REGISTER MATCHUP
2877  020274  104404                            TYPEL
2878
2879  020276  000065                    A$64:   $TAGA$=$TAGA$+1
2880
2881  020276  022777  000001  161106            CMP     #1,@TRWC                ;TEST THE WORD COUNT
2882  020304  001411                            BEQ     A$65
2883  020306  012705  000001                    MOV     #1,R5
2884  020312  017704  161074                    MOV     @TRWC,R4
2885  020316  013737  001412  001244            MOV     TRWC,REGIST
2886  020324  104001                            ERROR   1               ;INCORRECT REGISTER MATCHUP
2887  020326  104404                            TYPEL
2888
2889  020330  000066                    A$65:   $TAGA$=$TAGA$+1
2890
2891  020330  052777  004000  161044            BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
2892  020336  032777  004000  161036  64$:      BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
2893  020344  001374                            BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
2894  020346  000207                            RTS     PC
2895
2896                                    ;-----------------------------------------------------------------*
2897                                    ;THIS ROUTINE FORCS A READ COUNT DEROR WHILE READING A LONG RECORD
2898
2899  020350                            TESTAC:
2900  020350  004737  021712                    JSR     PC,TESTAH
2901  020354  004737  026610                    JSR     PC,TESTCK
2902  020360  004737  026630                    JSR     PC,TESTEK
2903  020364  004537  027546                    JSR     R5,TRYIT        ;;;
2904  020370  000004                            .WORD   READ            ;FUNCTION
2905  020372  177701                            .WORD   -77             ;WORD COUNT
2906  020374  035640                            .WORD   INPUT           ;BUS DDDRESS
2907  020376  013727  001240                    MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
2908  020402  000000            64$:    66$:    .WORD   0               ;USE THIS WORD AS A TIME COUNTER
2909  020404
2910  020404  005227  000000                    INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
2911  020410  001375                            BNE     66$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2912  020412  005337  020402                    DEC     64$             ;HAS THE TOTAL TIME ELAPSED?
2913  020416  001372                            BNE     66$             ;IF NO,GO WAIT A LITTLE LONGER
2914  020420                            65$:
2915
```

```
2916  020420  022777  102204  160754          CMP     #102204,@TRCR   ;TEST COMMAND REGISTER
2917  020426  001412                          BEQ     A$66
2918  020430  012705  102204                  MOV     #102204,R5
2919  020434  017704  160742                  MOV     @TRCR,R4
2920  020440  013737  001402  001244          MOV     TRCR,REGIST
2921  020446  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2922  020450  104403  007301                  TYPEF   ,MRCFL
2923
2924  020454  000067                  A$66:   $TAGA$=$TAGA$+1
2925
2926  020454  022777  003001  160724          CMP     #3001,@TRSR     ;TEST THE STATUS REGISTER
2927  020462  001411                          BEQ     A$67
2928  020464  012705  003001                  MOV     #3001,R5
2929  020470  017704  160712                  MOV     @TRSR,R4
2930  020474  013737  001406  001244          MOV     TRSR,REGIST
2931  020502  104001                          ERROR   !               ;INCORRECT REGISTER MATCHUP
2932  020504  104404                          TYPEL
2933
2934  020506  000070                  A$67:   $TAGA$=$TAGA$+1
2935
2936  020506  022777  036036  160702          CMP     #INPUT+176,@TRBA        ;TEST BUFFER ADDRESS
2937  020514  001411                          BEQ     A$70
2938  020516  012705  036036                  MOV     #INPUT+176,R5
2939  020522  017704  160670                  MOV     @TRBA,R4
2940  020526  013737  001416  001244          MOV     TRBA,REGIST
2941  020534  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2942  020536  104404                          TYPEL
2943
2944  020540  000071                  A$70:   $TAGA$=$TAGA$+1
2945
2946  020540  005777  160646                  TST     @TRWC           ;TEST THE WORD COUNT
2947  020544  001411                          BEQ     A$71
2948  020546  012705  000000                  MOV     #0,R5
2949  020552  017704  160634                  MOV     @TRWC,R4
2950  020556  013737  001412  001244          MOV     TRWC,REGIST
2951  020564  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
2952  020566  104404                          TYPEL
2953
2954  020570  000072                  A$71:   $TAGA$=$TAGA$+1
2955
2956  020570  052777  004000  160604          BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
2957  020576  032777  004000  160576  64$:    BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
2958  020604  001374                          BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
2959  020606  000207                          RTS     PC
2960
2961                                  ;-----------------------------------------------------------------------*
2962                                  ;THIS ROUTINE FORCES A READ COUNT DEROR WHILE READING A SHORT RECORD
2963
2964  020610                          TESTAD:
2965  020610  004737  021712                  JSR     PC,TESTAH
2966  020614  004737  026610                  JSR     PC,TESTCK
2967  020620  004737  026630                  JSR     PC,TESTEK
2968  020624  004537  027546                  JSR     R5,TRYIT        ;;;
2969  020630  000004                          .WORD   READ            ;FUNCTION
2970  020632  177677                          .WORD   -101            ;WORD COUNT
2971  020634  035640                          .WORD   INPUT           ;BUS DDDRESS
```

```
2972  020636  013727  001240              MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
2973  020642  000000              64$:    .WORD   0               ;USE THIS WORD AS A TIME COUNTER
2974  020644                      66$:
2975  020644  005227  000000              INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
2976  020650  001375                      BNE     66$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2977  020652  005337  020642              DEC     64$             ;HAS THE TOTAL TIME ELAPSED?
2978  020656  001372                      BNE     66$             ;IF NO,GO WAIT A LITTLE LONGER
2979  020660                      65$:
2980
2981  020660  022777  102204  160514      CMP     #102204,@TRCR   ;TEST COMMAND REGISTER
2982  020666  001411                      BEQ     A$72
2983  020670  012705  102204              MOV     #102204,R5
2984  020674  017704  160502              MOV     @TRCR,R4
2985  020700  013737  001402  001244      MOV     TRCR,REGIST
2986  020706  104001                      ERROR   1               ;INCORRECT REGISTER MATCHUP
2987  020710  104404                      TYPEL
2988
2989  020712  000073              A$72:   $TAGA$=$TAGA$+1
2990
2991  020712  022777  003001  160466      CMP     #3001,@TRSR     ;TEST THE STATUS REGISTER
2992  020720  001411                      BEQ     A$73
2993  020722  012705  003001              MOV     #3001,R5
2994  020726  017704  160454              MOV     @TRSR,R4
2995  020732  013737  001406  001244      MOV     TRSR,REGIST
2996  020740  104001                      ERROR   1               ;INCORRECT REGISTER MATCHUP
2997  020742  104404                      TYPEL
2998
2999  020744  000074              A$73:   $TAGA$=$TAGA$+1
3000
3001  020744  022777  036040  160444      CMP     #INPUT+200,@TRBA        ;TEST BUFFER ADDRESS
3002  020752  001411                      BEQ     A$74
3003  020754  012705  036040              MOV     #INPUT+200,R5
3004  020760  017704  160432              MOV     @TRBA,R4
3005  020764  013737  001416  001244      MOV     TRBA,REGIST
3006  020772  104001                      ERROR   1               ;INCORRECT REGISTER MATCHUP
3007  020774  104404                      TYPEL
3008
3009  020776  000075              A$74:   $TAGA$=$TAGA$+1
3010
3011  020776  022777  177777  160406      CMP     #-1,@TRWC               ;TEST THE WORD COUNT
3012  021004  001411                      BEQ     A$75
3013  021006  012705  177777              MOV     #-1,R5
3014  021012  017704  160374              MOV     @TRWC,R4
3015  021016  013737  001412  001244      MOV     TRWC,REGIST
3016  021024  104001                      ERROR   1               ;INCORRECT REGISTER MATCHUP
3017  021026  104404                      TYPEL
3018
3019  021030  000076              A$75:   $TAGA$=$TAGA$+1
3020
3021  021030  052777  004000  160344      BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
3022  021036  032777  004000  160336  64$:BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
3023  021044  001374                      BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
3024  021046  000207                      RTS     PC
3025
3026                                      ;-------------------------------------------------------------------*
3027                                      ;THIS ROUTINE CHECKS THE FUNCTION OF SPACING REVERSE WITH A
```

```
                                        ;SHORT WORD COUNT
                                        
021050                          TESTAE:
021050  004737  021712                  JSR     PC,TESTAH
021054  004737  026610                  JSR     PC,TESTCK
021060  004537  027546                  JSR     R5,TRYIT        ;;;
021064  001410                          .WORD   SPACER          ;FUNCTION
021066  177701                          .WORD   -77             ;WORD COUNT
021070  000000                          .WORD   0               ;BUS DDDRESS
                                        
021072  022777  003610  160302          CMP     #3610,@TRCR     ;TEST COMMAND REGISTER
021100  001411                          BEQ     A$76
021102  012705  003610                  MOV     #3610,R5
021106  017704  160270                  MOV     @TRCR,R4
021112  013737  001402  001244          MOV     TRCR,REGIST
021120  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
021122  104404                          TYPEL
                                        
021124  000077                  A$76:   $TAGA$=$TAGA$+1
                                        
021124  022777  002001  160254          CMP     #2001,@TRSR     ;TEST THE STATUS REGISTER
021132  001411                          BEQ     A$77
021134  012705  002001                  MOV     #2001,R5
021140  017704  160242                  MOV     @TRSR,R4
021144  013737  001406  001244          MOV     TRSR,REGIST
021152  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
021154  104404                          TYPEL
                                        
021156  000100                  A$77:   $TAGA$=$TAGA$+1
                                        
021156  005777  160234                  TST     @TRBA           ;TEST BUFFER ADDRESS
021162  001411                          BEQ     A$100
021164  012705  000000                  MOV     #0,R5
021170  017704  160222                  MOV     @TRBA,R4
021174  013737  001416  001244          MOV     TRBA,REGIST
021202  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
021204  104404                          TYPEL
                                        
021206  000101                  A$100:  $TAGA$=$TAGA$+1
                                        
021206  022777  177701  160176          CMP     #177701,@TRWC   ;TEST THE WORD COUNT
021214  001411                          BEQ     A$101
021216  012705  177701                  MOV     #177701,R5
021222  017704  160164                  MOV     @TRWC,R4
021226  013737  001412  001244          MOV     TRWC,REGIST
021234  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
021236  104404                          TYPEL
                                        
021240  000102                  A$101:  $TAGA$=$TAGA$+1
                                        
021240  000207                          RTS     PC
                                        
                                        ;-------------------------------------------------------------------------*
                                        ;THIS ROUTINE CHECKS THE FUNCTION OF SPACING WITH A LONG WORD COUNT
                                        
021242                          TESTAF:
```

```
3084  021242  004737  021712                  JSR     PC,TESTAH
3085  021246  004737  026610                  JSR     PC,TESTCK
3086  021252  004537  027546                  JSR     R5,TRYIT        ;;;
3087  021256  001410                          .WORD   SPACER          ;FUNCTION
3088  021260  177677                          .WORD   -101            ;WORD COUNT
3089  021262  000000                          .WORD   0               ;BUS DDDRESS
3090
3091  021264  022777  003610  160110          CMP     #3610,@TRCR     ;TEST COMMAND REGISTER
3092  021272  001411                          BEQ     A$102
3093  021274  012705  003610                  MOV     #3610,R5
3094  021300  017704  160076                  MOV     @TRCR,R4
3095  021304  013737  001402  001244          MOV     TRCR,REGIST
3096  021312  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3097  021314  104404                          TYPEL
3098
3099  021316  000103                  A$102:  $TAGA$=$TAGA$+1
3100
3101  021316  022777  002001  160062          CMP     #2001,@TRSR     ;TEST THE STATUS REGISTER
3102  021324  001410                          BEQ     A$103
3103  021326  012705  002001                  MOV     #2001,R5
3104  021332  017704  160050                  MOV     @TRSR,R4
3105  021336  013737  001406  001244          MOV     TRSR,REGIST
3106  021344  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3107
3108  021346  000104                  A$103:  $TAGA$=$TAGA$+1
3109
3110  021346  005777  160044                  TST     @TRBA           ;TEST BUFFER ADDRESS
3111  021352  001410                          BEQ     A$104
3112  021354  012705  000000                  MOV     #0,R5
3113  021360  017704  160032                  MOV     @TRBA,R4
3114  021364  013737  001416  001244          MOV     TRBA,REGIST
3115  021372  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3116
3117  021374  000105                  A$104:  $TAGA$=$TAGA$+1
3118
3119  021374  022777  177677  160010          CMP     #177677,@TRWC   ;TEST THE WORD COUNT
3120  021402  001410                          BEQ     A$105
3121  021404  012705  177677                  MOV     #177677,R5
3122  021410  017704  157776                  MOV     @TRWC,R4
3123  021414  013737  001412  001244          MOV     TRWC,REGIST
3124  021422  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3125
3126  021424  000106                  A$105:  $TAGA$=$TAGA$+1
3127
3128  021424  000207                          RTS     PC
3129
3130                                  ;-----------------------------------------------------------------*
3131                                  ;THIS ROUTINE TESTS THE END OF TAPE SENCING FUNCTION
3132                                  ;BY GOING FAST FORWARD TO END OF TAPE
3133
3134  021426                          TESTAG:
3135  021426  052777  004000  157746          BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
3136  021434  032777  004000  157740  64$:    BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
3137  021442  001374                          BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
3138  021444  012777  001026  157730          MOV     #GOEOT,@TRCR    ;SET COMMAND REGISTER
3139  021452  042777  000001  157726          BIC     #BIT0,@TRSR     ;CLEAR INHIBIT
```

```
3140  021460  012777  177777  157730          MOV     #-1,@TRBA       ;SET BUFFER ADDRESS
3141  021466  012777  177777  157716          MOV     #-1,@TRWC       ;SET WORD COUNT
3142  021474  005277  157702                  INC     @TRCR           ;SET THE GO BIT
3143
3144          021500                  A=.
3145  021500  032777  000200  157700          BIT     #BIT7,@TRSR
3146  021506  001774                          BEQ     A               ;BRANCH IF CLEAR
3147
3148  021510  013727  001240                  MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
3149  021514  000000                  65$:    .WORD   0               ;USE THIS WORD AS A TIME COUNTER
3150  021516                          67$:
3151  021516  005227  000000                  INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
3152  021522  001375                          BNE     67$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
3153  021524  005337  021514                  DEC     65$             ;HAS THE TOTAL TIME ELAPSED?
3154  021530  001372                          BNE     67$             ;IF NO,GO WAIT A LITTLE LONGER
3155  021532                          66$:
3156  021532  004737  012524                  JSR     PC,TESTA
3157  021536  022777  003226  157636          CMP     #3226,@TRCR     ;TEST COMMAND REGISTER
3158  021544  001412                          BEQ     A$106
3159  021546  012705  003226                  MOV     #3226,R5
3160  021552  017704  157624                  MOV     @TRCR,R4
3161  021556  013737  001402  001244          MOV     TRCR,REGIST
3162  021564  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3163  021566  104403  007317                  TYPEF   ,METFL
3164
3165  021572  000107                  A$106:  $TAGA$=$TAGA$+1
3166
3167  021572  022777  002201  157606          CMP     #2201,@TRSR     ;TEST THE STATUS REGISTER
3168  021600  001411                          BEQ     A$107
3169  021602  012705  002201                  MOV     #2201,R5
3170  021606  017704  157574                  MOV     @TRSR,R4
3171  021612  013737  001406  001244          MOV     TRSR,REGIST
3172  021620  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3173  021622  104404                          TYPEL
3174
3175  021624  000110                  A$107:  $TAGA$=$TAGA$+1
3176
3177  021624  022777  177777  157564          CMP     #-1,@TRBA       ;TEST BUFFER ADDRESS
3178  021632  001411                          BEQ     A$110
3179  021634  012705  177777                  MOV     #-1,R5
3180  021640  017704  157552                  MOV     @TRBA,R4
3181  021644  013737  001416  001244          MOV     TRBA,REGIST
3182  021652  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3183  021654  104404                          TYPEL
3184
3185  021656  000111                  A$110:  $TAGA$=$TAGA$+1
3186
3187  021656  022777  177777  157526          CMP     #-1,@TRWC       ;TEST THE WORD COUNT
3188  021664  001411                          BEQ     A$111
3189  021666  012705  177777                  MOV     #-1,R5
3190  021672  017704  157514                  MOV     @TRWC,R4
3191  021676  013737  001412  001244          MOV     TRWC,REGIST
3192  021704  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3193  021706  104404                          TYPEL
3194
3195  021710  000112                  A$111:  $TAGA$=$TAGA$+1
```

```
3196
3197  021710  000207                          RTS     PC
3198
3199                                  ;-----------------------------------------------------------------------*
3200                                  ;THIS ROUTINE INSURES THE TAPE IS NOT AT LOAD POINT AND THERE
3201                                  ;IS GOOD DATA ON THE TAPE
3202
3203  021712                          TESTAH:
3204  021712  052777  004000  157462          BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
3205  021720  032777  004000  157454  64$:    BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
3206  021726  001374                          BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
3207  021730  032777  000040  157450          BIT     #BIT5,@TRSR
3208  021736  001402                          BEQ     A$112
3209  021740  004737  014764                  JSR     PC,TESTO
3210  021744  000113          A$112:  $TAGA$=$TAGA$+1
3211  021744  004737  026650                  JSR     PC,TESTFO
3212  021750  000207                          RTS     PC
3213
3214                                  ;-----------------------------------------------------------------------*
3215                                  ;THIS ROUTINE DELAYS ABOUT 10 SECONDS
3216
3217  021752                          TESTAJ:
3218  021752  012737  000024  021776          MOV     #20.,3$
3219  021760  005737  021776  2$:     TST     3$
3220  021764  001001                          BNE     1$
3221  021766  000207                          RTS     PC
3222  021770  005337  021776  1$:     DEC     3$
3223  021774  000401                          BR      4$
3224  021776  000000          3$:     0
3225  022000                  4$:
3226  022000  013727  001240                  MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
3227  022004  000000          64$:    .WORD   0               ;USE THIS WORD AS A TIME COUNTER
3228  022006                  66$:
3229  022006  005227  000000                  INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
3230  022012  001375                          BNE     66$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
3231  022014  005337  022004                  DEC     64$             ;HAS THE TOTAL TIME ELAPSED?
3232  022020  001372                          BNE     66$             ;IF NO,GO WAIT A LITTLE LONGER
3233  022022                  65$:
3234  022022  000207                          RTS     PC
3235
3236                                  ;-----------------------------------------------------------------------*
3237
3238                                  ;THIS ROUTINE TELLS THE OPERATOR TO RESET THE
3239                                  ;TAPE TRANSPORT AND TESTS THE REGISTERS AFTER HE DOES
3240
3241  022024                          TESTAK:
3242  022024  004737  013626                  JSR     PC,TESTI
3243  022030  104402  007336                  TYPE    ,MRSTU
3244
3245          022034                  A=.
3246  022034  005237  010560                  INC     TEMP
3247  022040  001003                          BNE     64$
3248  022042  012777  000207  157142          MOV     #207,@TPDBR     ;RING THE BELL
3249  022050                  64$:
3250  022050  032777  002000  157330          BIT     #BIT10,@TRSR
3251  022056  001366                          BNE     A               ;BRANCH IF CLEAR
```

```
3252
3253
3254  022060  013727  001240                  MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
3255  022064  000000              65$:        .WORD   0               ;USE THIS WORD AS A TIME COUNTER
3256  022066                      67$:
3257  022066  005227  000000                  INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
3258  022072  001375                          BNE     67$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
3259  022074  005337  022064                  DEC     65$             ;HAS THE TOTAL TIME ELAPSED?
3260  022100  001372                          BNE     67$             ;IF NO,GO WAIT A LITTLE LONGER
3261  022102                      66$:
3262  022102  032777  100000  157272          BIT     #100000,@TRCR   ;TEST COMMAND REGISTER
3263  022110  001012                          BNE     A$113
3264  022112  012705  100026                  MOV     #100026,R5
3265  022116  017704  157260                  MOV     @TRCR,R4
3266  022122  013737  001402  001244          MOV     TRCR,REGIST
3267  022130  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3268  022132  104403  007361                  TYPEF   ,MMRSFL
3269
3270  022136  000114              A$113:      $TAGA$=$TAGA$+1
3271
3272  022136  017701  157244                  MOV     @TRSR,R1                ;GET STATUS
3273  022142  042701  000040                  BIC     #BIT5,R1        ;CLEAR BIT 5
3274
3275  022146  022701  000001                  CMP     #1,R1           ;TEST THE STATUS
3276  022152  001411                          BEQ     A$114
3277  022154  012705  000001                  MOV     #1,R5
3278  022160  017704  157222                  MOV     @TRSR,R4
3279  022164  013737  001406  001244          MOV     TRSR,REGIST
3280  022172  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3281  022174  104404                          TYPEL
3282
3283  022176  000115              A$114:      $TAGA$=$TAGA$+1
3284
3285  022176  005777  157214                  TST     @TRBA           ;TEST BUFFER ADDRESS
3286  022202  001411                          BEQ     A$115
3287  022204  012705  000000                  MOV     #0,R5
3288  022210  017704  157202                  MOV     @TRBA,R4
3289  022214  013737  001416  001244          MOV     TRBA,REGIST
3290  022222  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3291  022224  104404                          TYPEL
3292
3293  022226  000116              A$115:      $TAGA$=$TAGA$+1
3294
3295  022226  005777  157160                  TST     @TRWC           ;TEST THE WORD COUNT
3296  022232  001411                          BEQ     A$116
3297  022234  012705  000000                  MOV     #0,R5
3298  022240  017704  157146                  MOV     @TRWC,R4
3299  022244  013737  001412  001244          MOV     TRWC,REGIST
3300  022252  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3301  022254  104404                          TYPEL
3302
3303  022256  000117              A$116:      $TAGA$=$TAGA$+1
3304
3305  022256  000207                          RTS     PC
3306
3307                              ;-------------------------------------------------------------------------*
```

```
3308                                        ;THIS ROUTINE WRITES RECORDS PAST EOT
3309
3310  022260                                TESTAL:
3311  022260  032777  000200  157120              BIT     #BIT7,@TRSR
3312  022266  001033                            BNE     A$117
3313  022270  012737  001026  027536            MOV     #GOEOT,USEA             ;SET FUNCTION
3314  022276  012737  000001  027540            MOV     #--1,USEB       ;SET WORD COUNT
3315  022304  012737  177777  027542            MOV     #-1,USEC                ;SET BUS ADDRESS
3316  022312  004737  030012                    JSR     PC,EOTTST
3317
3318          022316                    A=.
3319  022316  032777  000200  157062            BIT     #BIT7,@TRSR
3320  022324  001774                            BEQ     A               ;BRANCH IS CLEAR
3321  022326  032777  000040  157046            BIT     #BIT5,@TRCR
3322  022334  001370                            BNE     A               ;BRANCH IF SET
3323  022336  032777  002000  157036            BIT     #BIT10,@TRCR
3324  022344  001764                            BEQ     A               ;BRANCH IF CLEAR
3325  022346  032777  000200  157026            BIT     #BIT7,@TRCR
3326  022354  001760                            BEQ     A               ;BRANCH IF CLEAR
3327
3328  022356  000120                    A$117:  $TAGA$=$TAGA$+1
3329  022356  012737  000016  027536            MOV     #ERASE,USEA             ;SET FUNCTION
3330  022364  012737  177000  027540            MOV     #-1000,USEB     ;SET WORD COUNT
3331  022372  012737  031600  027542            MOV     #OUTPUT,USEC            ;SET BUS ADDRESS
3332  022400  004737  030012                    JSR     PC,EOTTST
3333  022404  052777  004000  156770            BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
3334  022412  032777  004000  156762  64$:      BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
3335  022420  001374                            BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
3336
3337  022422  012737  000012  022446            MOV     #10.,3$
3338  022430  005737  022446            2$:     TST     3$
3339  022434  001001                            BNE     1$
3340  022436  000207                            RTS     PC
3341  022440  005337  022446            1$:     DEC     3$
3342  022444  000401                            BR      4$
3343  022446  000000                    3$:     0
3344  022450                            4$:
3345  022450  012737  000402  027536            MOV     #WRITE,USEA             ;SET FUNCTION
3346  022456  012737  177700  027540            MOV     #-100,USEB      ;SET WORD COUNT
3347  022464  012737  031600  027542            MOV     #OUTPUT,USEC            ;SET BUS ADDRESS
3348  022472  004737  030012                    JSR     PC,EOTTST
3349
3350  022476  022777  002602  156676            CMP     #2602,@TRCR     ;TEST COMMAND REGISTER
3351  022504  001412                            BEQ     A$120
3352  022506  012705  002602                    MOV     #2602,R5
3353  022512  017704  156664                    MOV     @TRCR,R4
3354  022516  013737  001402  001244            MOV     TRCR,REGIST
3355  022524  104001                            ERROR   1               ;INCORRECT REGISTER MATCHUP
3356  022526  104403  007401                    TYPEF   ,MWPETFL
3357
3358  022532  000121                    A$120:  $TAGA$=$TAGA$+1
3359
3360  022532  022777  002201  156646            CMP     #2201,@TRSR     ;TEST THE STATUS REGISTER
3361  022540  001411                            BEQ     A$121
3362  022542  012705  002201                    MOV     #2201,R5
3363  022546  017704  156634                    MOV     @TRSR,R4
```

```
3364  022552  013737  001406  001244          MOV     TRSR,REGIST
3365  022560  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3366  022562  104404                          TYPEL
3367
3368  022564  000122                  A$121:  $TAGA$=$TAGA$+1
3369
3370
3371  022564  005777  156622                  TST     @TRWC           ;TEST THE WORD COUNT
3372  022570  001411                          BEQ     A$122
3373  022572  012705  000000                  MOV     #0,R5
3374  022576  017704  156610                  MOV     @TRWC,R4
3375  022602  013737  001412  001244          MOV     TRWC,REGIST
3376  022610  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3377  022612  104404                          TYPEL
3378
3379  022614  000123                  A$122:  $TAGA$=$TAGA$+1
3380
3381  022614  000207                          RTS     PC
3382
3393                                  ;-----------------------------------------------------------------------*
3384                                  ;;THIS ROUTINE TESTS THE FUNCTION OF SPACE REVERSE PAST EOT
3385
3386  022616                          TESTAM:
3387  022616  004737  022260                  JSR     PC,TESTAL
3388
3389  022622  012737  001410  027536          MOV     #SPACER,USEA            ;SET FUNCTION
3390  022630  012737  000001  027540          MOV     #--1,USEB       ;SET WORD COUNT
3391  022636  012737  177777  027542          MOV     #-1,USEC                ;SET BUS ADDRESS
3392  022644  004737  030012                  JSR     PC,EOTTST
3393
3394  022650  022777  003610  156524          CMP     #3610,@TRCR     ;TEST COMMAND REGISTER
3395  022656  001412                          BEQ     A$123
3396  022660  012705  003610                  MOV     #3610,R5
3397  022664  017704  156512                  MOV     @TRCR,R4
3398  022670  013737  001402  001244          MOV     TRCR,REGIST
3399  022676  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3400  022700  104403  007434                  TYPEF   ,MSRETFL
3401
3402  022704  000124                  A$123:  $TAGA$=$TAGA$+1
3403
3404  022704  022777  002201  156474          CMP     #2201,@TRSR     ;TEST THE STATUS REGISTER
3405  022712  001411                          BEQ     A$124
3406  022714  012705  002201                  MOV     #2201,R5
3407  022720  017704  156462                  MOV     @TRSR,R4
3408  022724  013737  001406  001244          MOV     TRSR,REGIST
3409  022732  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3410  022734  104404                          TYPEL
3411
3412  022736  000125                  A$124:  $TAGA$=$TAGA$+1
3413
3414  022736  022777  177777  156452          CMP     #-1,@TRBA       ;TEST BUFFER ADDRESS
3415  022744  001411                          BEQ     A$125
3416  022746  012705  177777                  MOV     #-1,R5
3417  022752  017704  156440                  MOV     @TRBA,R4
3418  022756  013737  001416  001244          MOV     TRBA,REGIST
3419  022764  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
```

```
3420  022766  104404                          TYPEL
3421
3422  022770  000126                  A$125:  $TAGA$=$TAGA$+1
3423
3424  022770  022777  000001  156414          CMP     #1,@TRWC                ;TEST THE WORD COUNT
3425  022776  001411                          BEQ     A$126
3426  023000  012705  000001                  MOV     #1,R5
3427  023004  017704  156402                  MOV     @TRWC,R4
3428  023010  013737  001412  001244          MOV     TRWC,REGIST
3429  023016  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3430  023020  104404                          TYPEL
3431
3432  023022  000127                  A$126:  $TAGA$=$TAGA$+1
3433
3434  023022  000207                          RTS     PC
3435
3436                                  ;-----------------------------------------------------------------------*
3437                                  ;THIS ROUTINE TESTS THE FUNCTION OF READING PAST EOT
3438                                  ;TAPE MUST BE POSITIONED FIRST
3439
3440  023024                          TESTAN:
3441  023024  012737  000004  027536          MOV     #READ,USEA              ;SET FUNCTION
3442  023032  012737  177700  027540          MOV     #-100,USEB      ;SET WORD COUNT
3443  023040  012737  035640  027542          MOV     #INPUT,USEC             ;SET BUS ADDRESS
3444  023046  004737  030012                  JSR     PC,EOTTST
3445
3446  023052  022777  002204  156322          CMP     #2204,@TRCR     ;TEST COMMAND REGISTER
3447  023060  001412                          BEQ     A$127
3448  023062  012705  002204                  MOV     #2204,R5
3449  023066  017704  156310                  MOV     @TRCR,R4
3450  023072  013737  001402  001244          MOV     TRCR,REGIST
3451  023100  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3452  023102  104403  007476                  TYPEF   ,MRPETFL
3453
3454  023106  000130                  A$127:  $TAGA$=$TAGA$+1
3455
3456  023106  022777  002201  156272          CMP     #2201,@TRSR     ;TEST THE STATUS REGISTER
3457  023114  001411                          BEQ     A$130
3458  023116  012705  002201                  MOV     #2201,R5
3459  023122  017704  156260                  MOV     @TRSR,R4
3460  023126  013737  001406  001244          MOV     TRSR,REGIST
3461  023134  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3462  023136  104404                          TYPEL
3463
3464  023140  000131                  A$130:  $TAGA$=$TAGA$+1
3465
3466
3467  023140  005777  156246                  TST     @TRWC           ;TEST THE WORD COUNT
3468  023144  001411                          BEQ     A$131
3469  023146  012705  000000                  MOV     #0,R5
3470  023152  017704  156234                  MOV     @TRWC,R4
3471  023156  013737  001412  001244          MOV     TRWC,REGIST
3472  023164  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3473  023166  104404                          TYPEL
3474
3475  023170  000132                  A$131:  $TAGA$=$TAGA$+1
```

```
3476
3477  023170  000207                          RTS     PC
3478
3479                                  ;-------------------------------------------------------------------------*
3480                                  ;THIS ROUTINE WRITES A EOF PAST END OF TAPE
3481
3482  023172                          TESTA0:
3483  023172  004737  022260                  JSR     PC,TESTAL
3484
3485  023176  012737  001034  027536          MOV     #WEOF,USEA              ;SET FUNCTION
3486  023204  012737  000001  027540          MOV     #--1,USEB       ;SET WORD COUNT
3487  023212  012737  177777  027542          MOV     #-1,USEC                ;SET BUS ADDRESS
3488  023220  004737  030012                  JSR     PC,EOTTST
3489
3490  023224  022777  003234  156150          CMP     #3234,@TRCR     ,TEST COMMAND REGISTER
3491  023232  001412                          BEQ     A$132
3492  023234  012705  003234                  MOV     #3234,R5
3493  023240  017704  156136                  MOV     @TRCR,R4
3494  023244  013737  001402  001244          MOV     TRCR,REGIST
3495  023252  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3496  023254  104403  007527                  TYPEF   ,MWEFETF
3497
3498  023260  000133                  A$132:  $TAGA$=$TAGA$+1
3499
3500  023260  022777  002211  156120          CMP     #2211,@TRSR     ;TEST THE STATUS REGISTER
3501  023266  001411                          BEQ     A$133
3502  023270  012705  002211                  MOV     #2211,R5
3503  023274  017704  156106                  MOV     @TRSR,R4
3504  023300  013737  001406  001244          MOV     TRSR,REGIST
3505  023306  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3506  023310  104404                          TYPEL
3507
3508  023312  000134                  A$133:  $TAGA$=$TAGA$+1
3509
3510  023312  022777  177777  156076          CMP     #-1,@TRBA       ;TEST BUFFER ADDRESS
3511  023320  001411                          BEQ     A$134
3512  023322  012705  177777                  MOV     #-1,R5
3513  023326  017704  156064                  MOV     @TRBA,R4
3514  023332  013737  001416  001244          MOV     TRBA,REGIST
3515  023340  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3516  023342  104404                          TYPEL
3517
3518  023344  000135                  A$134:  $TAGA$=$TAGA$+1
3519
3520  023344  022777  000001  156040          CMP     #1,@TRWC                ;TEST THE WORD COUNT
3521  023352  001411                          BEQ     A$135
3522  023354  012705  000001                  MOV     #1,R5
3523  023360  017704  156026                  MOV     @TRWC,R4
3524  023364  013737  001412  001244          MOV     TRWC,REGIST
3525  023372  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3526  023374  104404                          TYPEL
3527
3528  023376  000136                  A$135:  $TAGA$=$TAGA$+1
3529
3530  023376  000207                          RTS     PC
3531
```

```
3532                                    ;-------------------------------------------------------------------------*
3533                                    ;THIS ROUTINE TESTS THE FUNCTION OF SPACING REVERSE OVER
3534                                    ;END OF FILE PAST EOT
3535
3536  023400                            TESTAP:
3537  023400  004737  023172                   JSR     PC,TESTAO
3538
3539  023404  012737  001410  027536           MOV     #SPACER,USEA            ;SET FUNCTION
3540  023412  012737  000001  027540           MOV     #--1,USEB       ;SET WORD COUNT
3541  023420  012737  177777  027542           MOV     #-1,USEC                ;SET BUS ADDRESS
3542  023426  004737  030012                   JSR     PC,EOTTST
3543
3544  023432  022777  003610  155742           CMP     #3610,@TRCR     ;TEST COMMAND REGISTER
3545  023440  001412                           BEQ     A$136
3546  023442  012705  003610                   MOV     #3610,R5
3547  023446  017704  155730                   MOV     @TRCR,R4
3548  023452  013737  001402  001244           MOV     TRCR,REGIST
3549  023460  104001                           ERROR   1               ;INCORRECT REGISTER MATCHUP
3550  023462  104403  007600                   TYPEF   ,MREFETF
3551
3552  023466  000137                    A$136: $TAGA$=$TAGA$+1
3553
3554  023466  022777  002211  155712           CMP     #2211,@TRSR     ;TEST THE STATUS REGISTER
3555  023474  001411                           BEQ     A$137
3556  023476  012705  002211                   MOV     #2211,R5
3557  023502  017704  155700                   MOV     @TRSR,R4
3558  023506  013737  001406  001244           MOV     TRSR,REGIST
3559  023514  104001                           ERROR   1               ;INCORRECT REGISTER MATCHUP
3560  023516  104404                           TYPEL
3561
3562  023520  000140                    A$137: $TAGA$=$TAGA$+1
3563
3564  023520  022777  177777  155670           CMP     #-1,@TRBA       ;TEST BUFFER ADDRESS
3565  023526  001411                           BEQ     A$140
3566  023530  012705  177777                   MOV     #-1,R5
3567  023534  017704  155656                   MOV     @TRBA,R4
3568  023540  013737  001416  001244           MOV     TRBA,REGIST
3569  023546  104001                           ERROR   1               ;INCORRECT REGISTER MATCHUP
3570  023550  104404                           TYPEL
3571
3572  023552  000141                    A$140: $TAGA$=$TAGA$+1
3573
3574  023552  022777  000001  155632           CMP     #1,@TRWC                ;TEST THE WORD COUNT
3575  023560  001411                           BEQ     A$141
3576  023562  012705  000001                   MOV     #1,R5
3577  023566  017704  155620                   MOV     @TRWC,R4
3578  023572  013737  001412  001244           MOV     TRWC,REGIST
3579  023600  104001                           ERROR   1               ;INCORRECT REGISTER MATCHUP
3580  023602  104404                           TYPEL
3581
3582  023604  000142                    A$141: $TAGA$=$TAGA$+1
3583
3584  023604  000207                           RTS     PC
3585
3586                                    ;-------------------------------------------------------------------------*
3587                                    ;THIS ROUTINE TESTS THE FUNCTION OF READING A EOF PAST EOT
```

```
3588
3589  023606                                TESTAQ:
3590  023606  004737  023400                      JSR     PC,TESTAP
3591
3592  023612  012737  000004  027536              MOV     #READ,USEA              ;SET FUNCTION
3593  023620  012737  000001  027540              MOV     #--1,USEB       ;SET WORD COUNT
3594  023626  012737  177777  027542              MOV     #-1,USEC                ;SET BUS ADDRESS
3595  023634  004737  030012                      JSR     PC,EOTTST
3596
3597  023640  022777  002204  155534              CMP     #2204,@TRCR     ;TEST COMMAND REGISTER
3598  023646  001412                              BEQ     A$142
3599  023650  012705  002204                      MOV     #2204,R5
3600  023654  017704  155522                      MOV     @TRCR,R4
3601  023660  013737  001402  001244              MOV     TRCR,REGIST
3602  023666  104001                              ERROR   1               ;INCORRECT REGISTER MATCHUP
3603  023670  104403  007600                      TYPEF   ,MREFETF
3604
3605  023674  000143                    A$142:    $TAGA$=$TAGA$+1
3606
3607  023674  022777  002211  155504              CMP     #2211,@TRSR     ;TEST THE STATUS REGISTER
3608  023702  001411                              BEQ     A$143
3609  023704  012705  002211                      MOV     #2211,R5
3610  023710  017704  155472                      MOV     @TRSR,R4
3611  023714  013737  001406  001244              MOV     TRSR,REGIST
3612  023722  104001                              ERROR   1               ;INCORRECT REGISTER MATCHUP
3613  023724  104404                              TYPEL
3614
3615  023726  000144                    A$143:    $TAGA$=$TAGA$+1
3616
3617  023726  022777  177777  155462              CMP     #-1,@TRBA       ;TEST BUFFER ADDRESS
3618  023734  001411                              BEQ     A$144
3619  023736  012705  177777                      MOV     #-1,R5
3620  023742  017704  155450                      MOV     @TRBA,R4
3621  023746  013737  001416  001244              MOV     TRBA,REGIST
3622  023754  104001                              ERROR   1               ;INCORRECT REGISTER MATCHUP
3623  023756  104404                              TYPEL
3624
3625  023760  000145                    A$144:    $TAGA$=$TAGA$+1
3626
3627  023760  022777  000001  155424              CMP     #1,@TRWC                ;TEST THE WORD COUNT
3628  023766  001411                              BEQ     A$145
3629  023770  012705  000001                      MOV     #1,R5
3630  023774  017704  155412                      MOV     @TRWC,R4
3631  024000  013737  001412  001244              MOV     TRWC,REGIST
3632  024006  104001                              ERROR   1               ;INCORRECT REGISTER MATCHUP
3633  024010  104404                              TYPEL
3634
3635  024012  000146                    A$145:    $TAGA$=$TAGA$+1
3636
3637  024012  000207                              RTS     PC
3638
3639                                    ;-------------------------------------------------------------------------*
3640                                    ;THIS ROUTINE TRIES TO WRITE ON TAPE WITHOUT A WRITE
3641                                    ;RING
3642
3643  024014                            RUNAS:
```

```
3644  024014  012737  000012  024040          MOV     #10.,3$
3645  024022  005737  024040          2$:     TST     3$
3646  024026  001001                          BNE     1$
3647  024030  000207                          RTS     PC
3648  024032  005337  024040          1$:     DEC     3$
3649  024036  000401                          BR      4$
3650  024040  000000          3$:     0
3651  024042                  4$:
3652
3653  024042                  TESTAS:
3654  024042  052777  004000  155332          BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
3655  024050  032777  004000  155324  64$:    BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
3656  024056  001374                          BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
3657  024060  004537  027546                  JSR     R5,TRYIT        ;;;
3658  024064  000402                          .WORD   WRITE           ;FUNCTION
3659  024066  177700                          .WORD   -100            ;WORD COUNT
3660  024070  031600                          .WORD   OUTPUT          ;BUS DDDRESS
3661
3662  024072  013727  001240                  MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
3663  024076  000000          65$:    .WORD   0               ;USE THIS WORD AS A TIME COUNTER
3664  024100                  67$:
3665  024100  005227  000000                  INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
3666  024104  001375                          BNE     67$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
3667  024106  005337  024076                  DEC     65$             ;HAS THE TOTAL TIME ELAPSED?
3668  024112  001372                          BNE     67$             ;IF NO,GO WAIT A LITTLE LONGER
3669  024114                  66$:
3670
3671  024114  022777  142602  155260          CMP     #142602,@TRCR   ;TEST COMMAND REGISTER
3672  024122  001412                          BEQ     A$146
3673  024124  012705  142602                  MOV     #142602,R5
3674  024130  017704  155246                  MOV     @TRCR,R4
3675  024134  013737  001402  001244          MOV     TRCR,REGIST
3676  024142  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3677  024144  104403  007650                  TYPEF   ,MWENFL
3678
3679  024150  000147          A$146:  $TAGA$=$TAGA$+1
3680
3681  024150  017701  155232                  MOV     @TRSR,R1                ;GET STATUS REGISTER
3682  024154  042701  000040                  BIC     #BIT5,R1        ;CLEAR UNWANTED BITS
3683
3684  024160  022701  002004                  CMP     #2004,R1        ;TEST THE STATUS REGISTER
3685  024164  001411                          BEQ     A$147
3686  024166  012705  002004                  MOV     #2004,R5
3687  024172  017704  155210                  MOV     @TRSR,R4
3688  024176  013737  001406  001244          MOV     TRSR,REGIST
3689  024204  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3690  024206  104404                          TYPEL
3691
3692  024210  000150          A$147:  $TAGA$=$TAGA$+1
3693
3694
3695  024210  022777  031600  155200          CMP     #OUTPUT,@TRBA   ;TEST BUFFER ADDRESS
3696  024216  001411                          BEQ     A$150
3697  024220  012705  031600                  MOV     #OUTPUT,R5
3698  024224  017704  155166                  MOV     @TRBA,R4
3699  024230  013737  001416  001244          MOV     TRBA,REGIST
```

```
3700  024236  104001                    ERROR   1               ;INCORRECT REGISTER MATCHUP
3701  024240  104404                    TYPEL
3702
3703  024242  000151            A$150:  $TAGA$=$TAGA$+1
3704
3705  024242  022777  177700  155142    CMP     #177700,@TRWC   ;TEST THE WORD COUNT
3706  024250  001411                    BEQ     A$151
3707  024252  012705  177700            MOV     #177700,R5
3708  024256  017704  155130            MOV     @TRWC,R4
3709  024262  013737  001412  001244    MOV     TRWC,REGIST
3710  024270  104001                    ERROR   1               ;INCORRECT REGISTER MATCHUP
3711  024272  104404                    TYPEL
3712
3713  024274  000152            A$151:  $TAGA$=$TAGA$+1
3714
3715  024274  052777  004000  155100    BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
3716  024302  032777  004000  155072  64$:  BIT   #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
3717  024310  001374                    BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
3718  024312  000207                    RTS     PC
3719
3720                                    ;-----------------------------------------------------------------------*
3721                                    ;THIS ROUTINE TRIES TO SPACE REVERSE AT LOAD POINT
3722
3723  024314                    TESTAT:
3724  024314  004737  013626            JSR     PC,TESTI
3725  024320  004537  027546            JSR     R5,TRYIT        ;;;
3726  024324  001410                    .WORD   SPACER          ;FUNCTION
3727  024326  177777                    .WORD   -1              ;WORD COUNT
3728  024330  000000                    .WORD   0               ;BUS DDDRESS
3729
3730
3731  024332  022777  143610  155042    CMP     #143610,@TRCR   ;TEST COMMAND REGISTER
3732  024340  001412                    BEQ     A$152
3733  024342  012705  143610            MOV     #143610,R5
3734  024346  017704  155030            MOV     @TRCR,R4
3735  024352  013737  001402  001244    MOV     TRCR,REGIST
3736  024360  104001                    ERROR   1               ;INCORRECT REGISTER MATCHUP
3737  024362  104403  007670            TYPEF   ,MSRLPFL
3738
3739  024366  000153            A$152:  $TAGA$=$TAGA$+1
3740
3741  024366  022777  002040  155012    CMP     #2040,@TRSR     ;TEST THE STATUS REGISTER
3742  024374  001411                    BEQ     A$153
3743  024376  012705  002040            MOV     #2040,R5
3744  024402  017704  155000            MOV     @TRSR,R4
3745  024406  013737  001406  001244    MOV     TRSR,REGIST
3746  024414  104001                    ERROR   1               ;INCORRECT REGISTER MATCHUP
3747  024416  104404                    TYPEL
3748
3749  024420  000154            A$153:  $TAGA$=$TAGA$+1
3750
3751  024420  005777  154772            TST     @TRBA           ;TEST BUFFER ADDRESS
3752  024424  001411                    BEQ     A$154
3753  024426  012705  000000            MOV     #0,R5
3754  024432  017704  154760            MOV     @TRBA,R4
3755  024436  013737  001416  001244    MOV     TRBA,REGIST
```

```
3756  024444  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3757  024446  104404                          TYPEL
3758
3759  024450  000155                  A$154:  $TAGA$=$TAGA$+1
3760
3761  024450  022777  177777  154734          CMP     #-1,@TRWC               ;TEST THE WORD COUNT
3762  024456  001411                          BEQ     A$155
3763  024460  012705  177777                  MOV     #-1,R5
3764  024464  017704  154722                  MOV     @TRWC,R4
3765  024470  013737  001412  001244          MOV     TRWC,REGIST
3766  024476  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3767  024500  104404                          TYPEL
3768
3769  024502  000156                  A$155:  $TAGA$=$TAGA$+1
3770
3771  024502  052777  004000  154672          BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
3772  024510  032777  004000  154664  64$:    BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
3773  024516  001374                          BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
3774  024520  000207                          RTS     PC
3775
3776
3777                                  ;-----------------------------------------------------------------------*
3778                                  ;THIS ROUTINE TESTS THE BUS ADDRESS BITS BY USING A COUNT PATTERN
3779
3780
3781  024522  005001                  TESTAU: CLR     R1              ;CLEAR COUNT
3782          024524                  A=.
3783  024524  010177  154666                  MOV     R1,@TRBA                ;SET BUS ADDRESS
3784  024530  020177  154662                  CMP     R1,@TRBA                ;TEST BUS ADDRESS
3785  024534  001407                          BEQ     A$156
3786  024536  010105                          MOV     R1,R5
3787  024540  017704  154652                  MOV     @TRBA,R4
3788  024544  013737  001416  001244          MOV     TRBA,REGIST
3789  024552  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3790
3791  024554  000157                  A$156:  $TAGA$=$TAGA$+1
3792  024554  005201                          INC     R1              ;INCREMENT COUNT
3793  024556  001362                          BNE     A               ;CONTINUE
3794  024560  000207                          RTS     PC
3795
3796
3797                                  ;-----------------------------------------------------------------------*
3798                                  ;THIS ROUTINE SPACES REVERSE OVER ID BLOCK
3799
3800  024562                          TESTAV:
3801  024562  004737  014764                  JSR     PC,TESTO
3802
3803  024566  004537  027546                  JSR     R5,TRYIT        ;;;
3804  024572  001410                          .WORD   SPACER          ;FUNCTION
3805  024574  177777                          .WORD   -1              ;WORD COUNT
3806  024576  177777                          .WORD   -1              ;BUS DDDRESS
3807
3808
3809  024600  022777  003610  154574          CMP     #3610,@TRCR     ;TEST COMMAND REGISTER
3810  024606  001412                          BEQ     A$157
3811  024610  012705  003610                  MOV     #3610,R5
```

```
3812  024614  017704  154562                  MOV     @TRCR,R4
3813  024620  013737  001402  001244          MOV     TRCR,REGIST
3814  024626  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3815  024630  104403  007727                  TYPEF   ,MSRIFL
3816
3817  024634  000160              A$157:      $TAGA$=$TAGA$+1
3818
3819  024634  022777  002061  154544          CMP     #2061,@TRSR     ;TEST THE STATUS REGISTER
3820  024642  001411                          BEQ     A$160
3821  024644  012705  002061                  MOV     #2061,R5
3822  024650  017704  154532                  MOV     @TRSR,R4
3823  024654  013737  001406  001244          MOV     TRSR,REGIST
3824  024662  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3825  024664  104404                          TYPEL
3826
3827  024666  000161              A$160:      $TAGA$=$TAGA$+1
3828
3829  024666  022777  177777  154522          CMP     #-1,@TRBA       ;TEST BUFFER ADDRESS
3830  024674  001411                          BEQ     A$161
3831  024676  012705  177777                  MOV     #-1,R5
3832  024702  017704  154510                  MOV     @TRBA,R4
3833  024706  013737  001416  001244          MOV     TRBA,REGIST
3834  024714  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3835  024716  104404                          TYPEL
3836
3837  024720  000162              A$161:      $TAGA$=$TAGA$+1
3838
3839  024720  022777  177777  154464          CMP     #-1,@TRWC       ;TEST THE WORD COUNT
3840  024726  001411                          BEQ     A$162
3841  024730  012705  177777                  MOV     #-1,R5
3842  024734  017704  154452                  MOV     @TRWC,R4
3843  024740  013737  001412  001244          MOV     TRWC,REGIST
3844  024746  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3845  024750  104404                          TYPEL
3846
3847  024752  000163              A$162:      $TAGA$=$TAGA$+1
3848
3849  024752  052777  004000  154422          BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
3850  024760  032777  004000  154414  64$:     BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
3851  024766  001374                          BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
3852  024770  000207                          RTS     PC
3853
3854                                  ;*****************************************************************
3855                                  ;THIS ROUTINE READS A ID BLOCK
3856
3857  024772                          TESTAW:
3858  024772  004737  013626                  JSR     PC,TESTI
3859  024776  005037  035640                  CLR     INPUT           ;CLEAR INPUT AREA
3860
3861  025002  004537  027546                  JSR     R5,TRYIT        ;;;
3862  025006  000004                          .WORD   READ            ;FUNCTION
3863  025010  177777                          .WORD   -1              ;WORD COUNT
3864  025012  035640                          .WORD   INPUT           ;BUS DDDRESS
3865
3866
3867  025014  022777  002204  154360          CMP     #2204,@TRCR     ;TEST COMMAND REGISTER
```

```
3868  025022  001412                          BEQ     A$163
3869  025024  012705  002204                  MOV     #2204,R5
3870  025030  017704  154346                  MOV     @TRCR,R4
3871  025034  013737  001402  001244          MOV     TRCR,REGIST
3872  025042  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3873  025044  104403  007764                  TYPEF   ,MRIBFL
3874
3875  025050  000164                  A$163:  $TAGA$=$TAGA$+1
3876
3877  025050  022777  002021  154330          CMP     #2021,@TRSR     ;TEST THE STATUS REGISTER
3878  025056  001411                          BEQ     A$164
3879  025060  012705  002021                  MOV     #2021,R5
3880  025064  017704  154316                  MOV     @TRSR,R4
3881  025070  013737  001406  001244          MOV     TRSR,REGIST
3882  025076  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3883  025100  104404                          TYPEL
3884
3885  025102  000165                  A$164:  $TAGA$=$TAGA$+1
3886
3887  025102  022777  035640  154306          CMP     #INPUT,@TRBA    ;TEST BUFFER ADDRESS
3888  025110  001411                          BEQ     A$165
3889  025112  012705  035640                  MOV     #INPUT,R5
3890  025116  017704  154274                  MOV     @TRBA,R4
3891  025122  013737  001416  001244          MOV     TRBA,REGIST
3892  025130  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3893  025132  104404                          TYPEL
3894
3895  025134  000166                  A$165:  $TAGA$=$TAGA$+1
3896
3897  025134  022777  177777  154250          CMP     #-1,@TRWC       ;TEST THE WORD COUNT
3898  025142  001411                          BEQ     A$166
3899  025144  012705  177777                  MOV     #-1,R5
3900  025150  017704  154236                  MOV     @TRWC,R4
3901  025154  013737  001412  001244          MOV     TRWC,REGIST
3902  025162  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3903  025164  104404                          TYPEL
3904
3905  025166  000167                  A$166:  $TAGA$=$TAGA$+1
3906
3907  025166  005737  035640                  TST     INPUT           ;TEST INPUT AREA
3908  025172  001411                          BEQ     A$167
3909  025174  012705  000000                  MOV     #0,R5
3910  025200  013704  035640                  MOV     INPUT,R4
3911  025204  012737  035640  001244          MOV     #INPUT,REGIST
3912  025212  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3913  025214  104404                          TYPEL
3914  025216  000170                  A$167:  $TAGA$=$TAGA$+1
3915  025216  052777  004000  154156          BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
3916  025224  032777  004000  154150  64$:    BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
3917  025232  001374                          BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
3918
3919  025234  000207                          RTS     PC
3920
3921                                  ;-----------------------------------------------------------------------*
3922                                  ;THIS ROUTINE WRITES EVEN PARITY
3923
```

```
3924  025236                          TESTAX:
3925  025236  004737  021712                  JSR     PC,TESTAH
3926
3927  025242  004537  027546                  JSR     R5,TRYIT        ;;;
3928  025246  000402                          .WORD   WRITE           ;FUNCTION
3929  025250  177700                          .WORD   -100            ;WORD COUNT
3930  025252  031574                          .WORD   BADOUT          ;BUS DDDRESS
3931  025254  013727  001240                  MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
3932  025260  000000                  64$:    .WORD   0               ;USE THIS WORD AS A TIME COUNTER
3933  025262                          66$:
3934  025262  005227  000000                  INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
3935  025266  001375                          BNE     66$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
3936  025270  005337  025260                  DEC     64$             ;HAS THE TOTAL TIME ELAPSED?
3937  025274  001372                          BNE     66$             ;IF NO,GO WAIT A LITTLE LONGER
3938  025276                          65$:
3939
3940  025276  022777  102602  154076          CMP     #102602,@TRCR   ;TEST COMMAND REGISTER
3941  025304  001412                          BEQ     A$170
3942  025306  012705  102602                  MOV     #102602,R5
3943  025312  017704  154064                  MOV     @TRCR,R4
3944  025316  013737  001402  001244          MOV     TRCR,REGIST
3945  025324  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3946  025326  104403  010005                  TYPEF   ,MPARFL
3947
3948  025332  000171                  A$170:  $TAGA$=$TAGA$+1
3949
3950  025332  022777  162001  154046          CMP     #162001,@TRSR   ;TEST THE STATUS REGISTER
3951  025340  001411                          BEQ     A$171
3952  025342  012705  162001                  MOV     #162001,R5
3953  025346  017704  154034                  MOV     @TRSR,R4
3954  025352  013737  001406  001244          MOV     TRSR,REGIST
3955  025360  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3956  025362  104404                          TYPEL
3957
3958  025364  000172                  A$171:  $TAGA$=$TAGA$+1
3959
3960  025364  013701  027540                  MOV     USEB,R1         ;GET WORD COUNT
3961  025370  005101                          COM     R1              ;FIX IT
3962  025372  005201                          INC     R1              ;INCREMENT IT
3963  025374  006301                          ASL     R1              ;DOUBLE IT
3964  025376  062701  031574                  ADD     #BADOUT,R1      ;FINISH IT
3965
3966  025402  020177  154010                  CMP     R1,@TRBA                ;TEST BUFFER ADDRESS
3967  025406  001410                          BEQ     A$172
3968  025410  010105                          MOV     R1,R5
3969  025412  017704  154000                  MOV     @TRBA,R4
3970  025416  013737  001416  001244          MOV     TRBA,REGIST
3971  025424  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3972  025426  104404                          TYPEL
3973
3974  025430  000173                  A$172:  $TAGA$=$TAGA$+1
3975
3976  025430  005777  153756                  TST     @TRWC           ;TEST THE WORD COUNT
3977  025434  001411                          BEQ     A$173
3978  025436  012705  000000                  MOV     #0,R5
3979  025442  017704  153744                  MOV     @TRWC,R4
```

```
3980  025446  013737  001412  001244          MOV     TRWC,REGIST
3981  025454  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
3982  025456  104404                          TYPEL
3983
3984  025460  000174                  A$173:  $TAGA$=$TAGA$+1
3985
3986  025460  052777  004000  153714          BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
3987  025466  032777  004000  153706  64$:    BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
3988  025474  001374                          BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
3989  025476  000207                          RTS     PC
3990
3991                                  ;-----------------------------------------------------------------*
3992                                  ;THIS ROUTINE SPACES REVERSE OVER EVEN PARITY
3993
3994  025500                          TESTAY:
3995  025500  004737  025236                  JSR     PC,TESTAX
3996
3997  025504  004537  027546                  JSR     R5,TRYIT        ;;;
3998  025510  001410                          .WORD   SPACER          ;FUNCTION
3999  025512  177700                          .WORD   -100            ;WORD COUNT
4000  025514  035640                          .WORD   INPUT           ;BUS DDDRESS
4001  025516  013727  001240                  MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
4002  025522  000000                  64$:    .WORD   0               ;USE THIS WORD AS A TIME COUNTER
4003  025524                          66$:
4004  025524  005227  000000                  INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
4005  025530  001375                          BNE     66$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
4006  025532  005337  025522                  DEC     64$             ;HAS THE TOTAL TIME ELAPSED?
4007  025536  001372                          BNE     66$             ;IF NO,GO WAIT A LITTLE LONGER
4008  025540                          65$:
4009
4010
4011  025540  022777  103610  153634          CMP     #103610,@TRCR   ;TEST COMMAND REGISTER
4012  025546  001412                          BEQ     A$174
4013  025550  012705  103610                  MOV     #103610,R5
4014  025554  017704  153622                  MOV     @TRCR,R4
4015  025560  013737  001402  001244          MOV     TRCR,REGIST
4016  025566  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
4017  025570  104403  010024                  TYPEF   ,MEPSRFL
4018
4019  025574  000175                  A$174:  $TAGA$=$TAGA$+1
4020
4021  025574  022777  122001  153604          CMP     #122001,@TRSR   ;TEST THE STATUS REGISTER
4022  025602  001411                          BEQ     A$175
4023  025604  012705  122001                  MOV     #122001,R5
4024  025610  017704  153572                  MOV     @TRSR,R4
4025  025614  013737  001406  001244          MOV     TRSR,REGIST
4026  025622  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
4027  025624  104404                          TYPEL
4028
4029  025626  000176                  A$175:  $TAGA$=$TAGA$+1
4030
4031  025626  022777  035640  153562          CMP     #INPUT,@TRBA    ;TEST BUFFER ADDRESS
4032  025634  001411                          BEQ     A$176
4033  025636  012705  035640                  MOV     #INPUT,R5
4034  025642  017704  153550                  MOV     @TRBA,R4
4035  025646  013737  001416  001244          MOV     TRBA,REGIST
```

```
4036  025654  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
4037  025656  104404                          TYPEL
4038
4039  025660  000177              A$176:      $TAGA$=$TAGA$+1
4040
4041  025660  022777  177700  153524          CMP     #177700,@TRWC   ;TEST THE WORD COUNT
4042  025666  001411                          BEQ     A$177
4043  025670  012705  177700                  MOV     #177700,R5
4044  025674  017704  153512                  MOV     @TRWC,R4
4045  025700  013737  001412  001244          MOV     TRWC,REGIST
4046  025706  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
4047  025710  104404                          TYPEL
4048
4049  025712  000200              A$177:      $TAGA$=$TAGA$+1
4050
4051  025712  052777  004000  153462          BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
4052  025720  032777  004000  153454  64$:     BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
4053  025726  001374                          BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
4054  025730  000207                          RTS     PC
4055
4056                                  ;-----------------------------------------------------------------*
4057                                  ;THIS ROUTINE READS EVEN PARITY
4058
4059  025732                          TESTAZ:
4060  025732  004737  025500                  JSR     PC,TESTAY
4061
4062  025736  004537  027546                  JSR     R5,TRYIT        ;;
4063  025742  000004                          .WORD   READ            ;FUNCTION
4064  025744  177700                          .WORD   -100            ;WORD COUNT
4065  025746  035640                          .WORD   INPUT           ;BUS ODDRESS
4066  025750  013727  001240                  MOV     HOLD,(PC)+      ;PICK UP TIME PARAMETER
4067  025754  000000              64$:        .WORD   0               ;USE THIS WORD AS A TIME COUNTER
4068  025756                      66$:
4069  025756  005227  000000                  INC     #0              ;IF NO,COUNT 1 OF 65535 TICKS
4070  025762  001375                          BNE     66$             ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
4071  025764  005337  025754                  DEC     64$             ;HAS THE TOTAL TIME ELAPSED?
4072  025770  001372                          BNE     66$             ;IF NO,GO WAIT A LITTLE LONGER
4073  025772                      65$:
4074
4075
4076  025772  022777  102204  153402          CMP     #102204,@TRCR   ;TEST COMMAND REGISTER
4077  026000  001412                          BEQ     A$200
4078  026002  012705  102204                  MOV     #102204,R5
4079  026006  017704  153370                  MOV     @TRCR,R4
4080  026012  013737  001402  001244          MOV     TRCR,REGIST
4081  026020  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
4082  026022  104403  010061                  TYPEF   ,MREPFL
4083
4084  026026  000201              A$200:      $TAGA$=$TAGA$+1
4085
4086  026026  022777  163001  153352          CMP     #163001,@TRSR   ;TEST THE STATUS REGISTER
4087  026034  001411                          BEQ     A$201
4088  026036  012705  163001                  MOV     #163001,R5
4089  026042  017704  153340                  MOV     @TRSR,R4
4090  026046  013737  001406  001244          MOV     TRSR,REGIST
4091  026054  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
```

```
4092  026056  104404                          TYPEL
4093
4094  026060  000202                  A$201:  $TAGA$=$TAGA$+1
4095
4096  026060  022777  035642  153330          CMP     #INPUT+2,@TRBA  ;TEST BUFFER ADDRESS
4097  026066  001410                          BEQ     A$202
4098  026070  010105                          MOV     R1,R5
4099  026072  017704  153320                  MOV     @TRBA,R4
4100  026076  013737  001416  001244          MOV     TRBA,REGIST
4101  026104  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
4102  026106  104404                          TYPEL
4103
4104  026110  000203                  A$202:  $TAGA$=$TAGA$+1
4105
4106  026110  022777  177701  153274          CMP     #177701,@TRWC   ;TEST THE WORD COUNT
4107  026116  001411                          BEQ     A$203
4108  026120  012705  177701                  MOV     #177701,R5
4109  026124  017704  153262                  MOV     @TRWC,R4
4110  026130  013737  001412  001244          MOV     TRWC,REGIST
4111  026136  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
4112  026140  104404                          TYPEL
4113
4114  026142  000204                  A$203:  $TAGA$=$TAGA$+1
4115
4116  026142  000207                          RTS     PC
4117
4118                                  ;-----------------------------------------------------------------*
4119                                  ;THIS ROUTINE TESTS THE BITS IN THE WORD COUNT USING A COUNT PATTERN
4120
4121  026144  005001                  TESTBA: CLR     R1              ;CLEAR COUNT
4122          026146                  A=.
4123  026146  010177  153240                  MOV     R1,@TRWC                ;SET WORD COUNT
4124  026152  020177  153234                  CMP     R1,@TRWC                ;TEST WORD COUNT
4125  026156  001411                          BEQ     A$204
4126  026160  010105                          MOV     R1,R5
4127  026162  017704  153224                  MOV     @TRWC,R4
4128  026166  013737  001412  001244          MOV     TRWC,REGIST
4129  026174  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
4130  026176  104403  010105                  TYPEF   ,MWCFL
4131
4132  026202  000205                  A$204:  $TAGA$=$TAGA$+1
4133  026202  005201                          INC     R1              ;INCREMENT COUNT
4134  026204  001360                          BNE     A               ;CONTINUE
4135  026206  000207                          RTS     PC
4136
4137
4138                                  ;-----------------------------------------------------------------*
4139
4140
4141                                  ;THIS ROUTINE TESTS THE STATUS AND COMMAND REGISTERS
4142                                  ;AFTER VALID WRITE FUNCTIONS
4143
4144
4145  026210  022777  002602  153164  TESTCX: CMP     #2602,@TRCR     ;TEST COMMAND REGISTER
4146  026216  001422                          BEQ     A$205
4147  026220  032777  000200  153160          BIT     #200,@TRSR      ;EOT??
```

```
4148  026226  001404                          BEQ     1$              ;NO
4149  026230  022626                          CMP     (SP)+,(SP)+
4150  026232  022626                          CMP     (SP)+,(SP)+
4151  026234  022626                          CMP     (SP)+,(SP)+
4152  026236  000207                          RTS     PC
4153  026240                          1$:
4154  026240  012705  002602                  MOV     #2602,R5
4155  026244  017704  153132                  MOV     @TRCR,R4
4156  026250  013737  001402  001244          MOV     TRCR,REGIST
4157  026256  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
4158  026260  104403  005667                  TYPEF   ,MWRTFL
4159
4160  026264  000206                  A$205:  $TAGA$=$TAGA$+1
4161
4162  026264  022777  002001  153114          CMP     #2001,@TRSR     ;TEST THE STATUS REGISTER
4163  026272  001422                          BEQ     A$206
4164  026274  032777  000200  153104          BIT     #200,@TRSR      ;AT EOT??
4165  026302  001404                          BEQ     1$              ;NO
4166  026304  022626                          CMP     (SP)+,(SP)+
4167  026306  022626                          CMP     (SP)+,(SP)+
4168  026310  022626                          CMP     (SP)+,(SP)+
4169  026312  000207                          RTS     PC
4170  026314                          1$:
4171  026314  012705  002001                  MOV     #2001,R5
4172  026320  017704  153062                  MOV     @TRSR,R4
4173  026324  013737  001406  001244          MOV     TRSR,REGIST
4174  026332  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
4175  026334  104403  005667                  TYPEF   ,MWRTFL
4176
4177  026340  000207                  A$206:  $TAGA$=$TAGA$+1
4178
4179  026340  005777  153046                  TST     @TRWC           ;TEST WORD COUNT
4180  026344  001412                          BEQ     A$207
4181  026346  012705  000000                  MOV     #0,R5
4182  026352  017704  153034                  MOV     @TRWC,R4
4183  026356  013737  001412  001244          MOV     TRWC,REGIST
4184  026364  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
4185  026366  104403  005667                  TYPEF   ,MWRTFL
4186
4187  026372  000210                  A$207:  $TAGA$=$TAGA$+1
4188  026372  000207                          RTS     PC
4189                          ;-----------------------------------------------------------------*
4190
4191                          ;THIS ROUTINE TESTS THE COMMAND AND STATUS AFTER VALID
4192                          ;READ FUNCTIONS
4193
4194  026374  022777  002204  153000  TESTDX: CMP     #2204,@TRCR     ;TEST COMMAND REGISTER
4195  026402  001421                          BEQ     A$210
4196  026404  032777  000200  152774          BIT     #200,@TRSR      ;AT EOT??
4197  026412  001403                          BEQ     1$              ;NO
4198  026414  022626                          CMP     (SP)+,(SP)+
4199  026416  005726                          TST     (SP)+
4200  026420  000207                          RTS     PC
4201  026422                          1$:
4202  026422  012705  002204                  MOV     #2204,R5
4203  026426  017704  152750                  MOV     @TRCR,R4
```

```
 4204  026432  013737  001402  001244          MOV     TRCR,REGIST
 4205  026440  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
 4206  026442  104403  005700                  TYPEF   ,MRDFL
 4207
 4208  026446  000211                  A$210:  $TAGA$=$TAGA$+1
 4209
 4210  026446  022777  002001  152732          CMP     #2001,@TRSR     ;TEST THE STATUS REGISTER
 4211  026454  001421                          BEQ     A$21
 4212  026456  032777  000200  152722          BIT     #200,@TRSR      ;AT EOT??
 4213  026464  001403                          BEQ     1$              ;NO
 4214  026466  022626                          CMP     (SP)+,(SP)+
 4215  026470  005726                          TST     (SP)+
 4216  026472  000207                          RTS     PC
 4217  026474                          1$:
 4218  026474  012705  002001                  MOV     #2001,R5
 4219  026500  017704  152702                  MOV     @TRSR,R4
 4220  026504  013737  001406  001244          MOV     TRSR,REGIST
 4221  026512  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
 4222  026514  104403  005700                  TYPEF   ,MRDFL
 4223
 4224  026520  000212                  A$211:  $TAGA$=$TAGA$+1
 4225
 4226  026520  005777  152666                  TST     @TRWC           ;TEST WORD COUNT
 4227  026524  001412                          BEQ     A$212
 4228  026526  012705  000000                  MOV     #0,R5
 4229  026532  017704  152654                  MOV     @TRWC,R4
 4230  026536  013737  001412  001244          MOV     TRWC,REGIST
 4231  026544  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
 4232  026546  104403  005700                  TYPEF   ,MRDFL
 4233
 4234  026552  000213                  A$212:  $TAGA$=$TAGA$+1
 4235
 4236  026552  004737  015622                  JSR     PC,TESTS
 4237  026556  000207                          RTS     PC
 4238
 4239
 4240  026560                          TESTCA:
 4241  026560  004737  031212                  JSR     PC,ENAINT
 4242  026564  004537  027546                  JSR     R5,TRYIT        ;;;
 4243  026570  000402                          .WORD   WRITE           ;FUNCTION
 4244  026572  177777                          .WORD   -1              ;WORD COUNT
 4245  026574  031600                          .WORD   OUTPUT          ;BUS DDDRESS
 4246  026576  004737  031252                  JSR     PC,CHKINT
 4247  026602  004737  026210                  JSR     PC,TESTCX
 4248  026606  000207                          RTS     PC
 4249
 4250  026610                          TESTCK:
 4251  026610  004537  027546                  JSR     R5,TRYIT        ;;;
 4252  026614  000402                          .WORD   WRITE           ;FUNCTION
 4253  026616  177700                          .WORD   -100            ;WORD COUNT
 4254  026620  031600                          .WORD   OUTPUT          ;BUS DDDRESS
 4255  026622  004737  026210                  JSR     PC,TESTCX
 4256  026626  000207                          RTS     PC
 4257  026630                          TESTEK:
 4258  026630  004537  027546                  JSR     R5,TRYIT        ;;;
 4259  026634  001410                          .WORD   SPACER          ;FUNCTION
```

```
 4260  026636  177700                        .WORD   -100            ;WORD COUNT
 4261  026640  000000                        .WORD   0               ;BUS DDDRESS
 4262  026642  004737  026670                JSR     PC,TESTEX
 4263  026646  000207                        RTS     PC
 4264
 4265
 4266
 4267  026650                        TESTF0:
 4268  026650  004537  027546                JSR     R5,TRYIT        ;;;
 4269  026654  000016                        .WORD   ERASE           ;FUNCTION
 4270  026656  177400                        .WORD   -400            ;WORD COUNT
 4271  026660  031600                        .WORD   OUTPUT          ;BUS DDDRESS
 4272  026662  004737  027016                JSR     PC,TESTFX
 4273  026666  000207                        RTS     PC
 4274
 4275                                ;-----------------------------------------------------------------*
 4276
 4277                                ;THIS ROUTINE TESTS THE COMMAND AND STATUS AFTER VALID
 4278                                ;SPACE REVERSE FUNCTIONS
 4279
 4280  026670  022777  003610  152504 TESTEX: CMP     #3610,@TRCR     ;TEST COMMAND REGISTER
 4281  026676  001412                        BEQ     A$213
 4282  026700  012705  003610                MOV     #3610,R5
 4283  026704  017704  152472                MOV     @TRCR,R4
 4284  026710  013737  001402  001244        MOV     TRCR,REGIST
 4285  026716  104001                        ERROR   1               ;INCORRECT REGISTER MATCHUP
 4286  026720  104403  005710                TYPEF   ,MSRVFL
 4287
 4288  026724  000214                 A$213: $TAGA$=$TAGA$+1
 4289
 4290  026724  022777  002001  152454        CMP     #2001,@TRSR     ;TEST THE STATUS REGISTER
 4291  026732  001412                        BEQ     A$214
 4292  026734  012705  002001                MOV     #2001,R5
 4293  026740  017704  152442                MOV     @TRSR,R4
 4294  026744  013737  001406  001244        MOV     TRSR,REGIST
 4295  026752  104001                        ERROR   1               ;INCORRECT REGISTER MATCHUP
 4296  026754  104403  005710                TYPEF   ,MSRVFL
 4297
 4298  026760  000215                 A$214: $TAGA$=$TAGA$+1
 4299
 4300  026760  023777  027540  152424        CMP     USEB,@TRWC      ;TEST WORD COUNT
 4301  026766  001412                        BEQ     A$215
 4302  026770  013705  027540                MOV     USEB,R5
 4303  026774  017704  152412                MOV     @TRWC,R4
 4304  027000  013737  001412  001244        MOV     TRWC,REGIST
 4305  027006  104001                        ERROR   1               ;INCORRECT REGISTER MATCHUP
 4306  027010  104403  005710                TYPEF   ,MSRVFL
 4307
 4308  027014  000216                 A$215: $TAGA$=$TAGA$+1
 4309
 4310  027014  000207                        RTS     PC
 4311
 4312
 4313
 4314                                ;-----------------------------------------------------------------*
 4315                                ;THIS ROUTINE TESTS THE COMMAND AND STATUS REGISTERS AFTER
```

```
                                        ;AN ERASE FUNCTION
                                        
027016  022777  002216  152356  TESTFX: CMP     #2216,@TRCR     ;TEST COMMAND REGISTER
027024  001412                          BEQ     A$216
027026  012705  002216                  MOV     #2216,R5
027032  017704  152344                  MOV     @TRCR,R4
027036  013737  001402  001244          MOV     TRCR,REGIST
027044  104001                          ERROR   1               ;INCOPRECT REGISTER MATCHUP
027046  104403  005731                  TYPEF   ,MERFL
                                        
027052  000217                  A$216:  $TAGA$=$TAGA$+1
                                        
027052  022777  002001  152326          CMP     #2001,@TRSR     ;TEST THE STATUS REGISTER
027060  001412                          BEQ     A$217
027062  012705  002001                  MOV     #2001,R5
027066  017704  152314                  MOV     @TRSR,R4
027072  013737  001406  001244          MOV     TRSR,REGIST
027100  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
027102  104403  005731                  TYPEF   ,MERFL
                                        
027106  000220                  A$217:  $TAGA$=$TAGA$+1
                                        
027106  013701  027540                  MOV     USEB,R1         ;GET WORD COUNT
027112  005101                          COM     R1              ;FIX IT
027114  005201                          INC     R1
027116  006301                          ASL     R1              ;DOUBLE IT
027120  062701  031600                  ADD     #OUTPUT,R1      ;COMPUTE BUS ADDRESS
                                        
027124  020177  152266                  CMP     R1,@TRBA                ;TEST BUFFER ADDRESS
027130  001411                          BEQ     A$220
027132  010105                          MOV     R1,R5
027134  017704  152256                  MOV     @TRBA,R4
027140  013737  001416  001244          MOV     TRBA,REGIST
027146  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
027150  104403  005731                  TYPEF   ,MERFL
                                        
027154  000221                  A$220:  $TAGA$=$TAGA$+1
                                        
027154  005777  152232                  TST     @TRWC           ;TEST THE WORD COUNT
027160  001412                          BEQ     A$221
027162  012705  000000                  MOV     #0,R5
027166  017704  152220                  MOV     @TRWC,R4
027172  013737  001412  001244          MOV     TRWC,REGIST
027200  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
027202  104403  005731                  TYPEF   ,MERFL
                                        
027206  000222                  A$221:  $TAGA$=$TAGA$+1
                                        
027206  000207                          RTS     PC
                                        
                                        ;-------------------------------------------------------------------------*
                                        ;THIS ROUTINE WRITES,SPACES REVERSE, AND READS
                                        
027210  012704  000017          TESTXX: MOV     #15.,R4         ;LOAD THE TOTAL SUBTEST PASS COUNT
027214  012703  027356                  MOV     #SIZTAB,R3      ;POINT TO TABLE OF BLOCK SIZES
027220  012737  000001  001264          MOV     #1,WRTFLG       ;SET FLAG FOR RETURN
```

```
 4372  027226  012337  027306          1$:     MOV     (R3)+,4$
 4373  027232  012702  000003                  MOV     #3,R2           ;KEEP TRACK OF THE NUMBER OF FUNCTIONS TO DO
 4374  027236  012705  027422                  MOV     #BUFTAB,R5      ;POINT TO TABLE OF BUFFERS
 4375  027242  012701  027414                  MOV     #FUNCTB,R1      ;POINT TO TABLE OF FUNCTIONS
 4376  027246  012700  027430                  MOV     #SUBTST,R0      ;POINT TO TABLE OF FOLLOWUP TESTS
 4377  027252  004737  031212                  JSR     PC,ENAINT
 4378  027256  012137  027304          2$:     MOV     (R1)+,3$        ;LOAD A FUNCTION IN SUBROUTINE INTERFACE
 4379  027262  012537  027310                  MOV     (R5)+,5$        ;LOAD THE NAME OF THE BUFFER
 4380  027266  010146                          MOV     R1,-(SP)        ;SAVE REGISTERS.
 4381  027270  010246                          MOV     R2,-(SP)
 4382  027272  010346                          MOV     R3,-(SP)
 4383  027274  010446                          MOV     R4,-(SP)
 4384  027276  010546                          MOV     R5,-(SP)
 4385  027300  004537  027546                  JSR     R5,TRYIT        ;GO EXECUTE THE FUNCTION
 4386  027304  000000          3$:     .WORD   0
 4387  027306  000000          4$:     .WORD   0
 4388  027310  000000          5$:     .WORD   0
 4389  027312  004730                          JSR     PC,@(R0)+       ;DO THE APPROPRIATE FOLLOWUP TEST
 4390  027314  012605                          MOV     (SP)+,R5        ;RESTORE THE REGISTERS
 4391  027316  012604                          MOV     (SP)+,R4
 4392  027320  012603                          MOV     (SP)+,R3
 4393  027322  012602                          MOV     (SP)+,R2
 4394  027324  012601                          MOV     (SP)+,R1
 4395  027326  005302                          DEC     R2              ;REDUCE THE COUNT. ALL FUNCTIONS DONE?
 4396  027330  001352                          BNE     2$              ;IF NOT, GO DO THE NEXT ONE
 4397  027332  005304                          DEC     R4              ;REDUCE THE OVERALL COUNT. ALL BLOCK SIZES DONE?
 4398  027334  001334                          BNE     1$              ;IF NOT, GO DO THE NEXT ONE
 4399  027336  004737  014446                  JSR     PC,TESTL
 4400  027342  004737  015734                  JSR     PC,TESTU
 4401  027346  004737  016144                  JSR     PC,TESTV
 4402  027352  000137  027210                  JMP     TESTXX
 4403  027356  177777  177776  177775  SIZTAB: .WORD   -1,-2,-3,-4,-5,-6,-7,-10,-20,-40,-100,-200,-400,-1000,-2000
  (1)  027414  000402  001410  000004  FUNCTB: .WORD   WRITE,SPACER,READ
  (1)  027422  031600  000000  035640  BUFTAB: .WORD   OUTPUT,0,INPUT
  (1)  027430  026210  026670  026374  SUBTST: .WORD   TESTCX,TESTEX,TESTDX
  (1)
 4404                                     ;-----------------------------------------------------------------*
 4405                                     ;THIS ROUTINE READS DATA WRITTEN BY TESTXX.
 4406
 4407  027436  012704  000017          TESTXY: MOV     #15.,R4         ;COUNT THE NUMBER OF ITERATIONS
 4408  027442  012703  027356                  MOV     #SIZTAB,R3      ;POINT TO THE TABLE OF BLOCK SIZES
 4409  027446  012737  000002  001264          MOV     #2,WRTFLG       ;SET RETURN FLAG
 4410  027454  004737  031212                  JSR     PC,ENAINT
 4411  027460  012337  027476          1$:     MOV     (R3)+,2$        ;LOAD SUBROUTINE INTERFACE
 4412  027464  010346                          MOV     R3,-(SP)        ;SAVE REGISTERS
 4413  027466  010446                          MOV     R4,-(SP)
 4414  027470  004537  027546                  JSR     R5,TRYIT        ;GO DO A READ FUNCTION
 4415  027474  000004                          .WORD   READ            ;EXECUTE A READ
 4416  027476  000000          2$:     .WORD   0               ;BLOCK SIZE
 4417  027500  035640                          INPUT                   ;BUFFER TO USE
 4418  027502  004737  026374                  JSR     PC,TESTDX               ;EXECUTE THE FOLLOWUP ROUTINE
 4419  027506  012604                          MOV     (SP)+,R4        ;RESTORE REGISTERS.
 4420  027510  012603                          MOV     (SP)+,R3
 4421  027512  005304                          DEC     R4              ;REDUE COUNT. ALL BLOCK SIZES DONE?
 4422  027514  001361                          BNE     1$              ;IF NO, GO DO THE NEXT ONE
 4423  027516  004737  016144                  JSR     PC,TESTV
```

```
 4424  027522  000137  027436                    JMP     TESTXY
 4425
 4426
 4427
 4428                              ;-----------------------------------------------------------------*
 4429
 4430
 4431                              ;THIS ROUTINE PERFORMS TAPE FUNCTION IT IS CALLED
 4432                              ;BY THE TAPFUN MACRO
 4433
 4434  027526  000000          USEA0:  0
 4435  027530  000000          USEB0:  0
 4436  027532  000000          USEC0:  0
 4437  027534  000000          USED0:  0
 4438
 4439  027536  000000          USEA:   0                       ;FUNCTION
 4440  027540  000000          USEB:   0                       ;WORD COUNT
 4441  027542  000000          USEC:   0                       ;INITIAL BUFFER ADDRESS
 4442  027544  000000          USED:   0                       ;TEMP STORAGE LOCATION
 4443
 4444  027546  012537  027536  TRYIT:  MOV     (R5)+,USEA      ;GET THE FUNCTION
 4445  027552  012537  027540          MOV     (R5)+,USEB      ;GET THE WORD COUNT
 4446  027556  012537  027542          MOV     (R5)+,USEC      ;GET THE BUS ADDRESS
 4447  027562  000223          A$222:  $TAGA$=$TAGA$+1
 4448  027562  013737  027536  027526  MOV     USEA,USEA0      ;SAVE LAST FUNCTION
 4449  027570  013737  027540  027530  MOV     USEB,USEB0      ;SAVE LAST WORD COUNT
 4450  027576  013737  027542  027532  MOV     USEC,USEC0      ;SAVE LAST BUS ADDRESS
 4451  027604  004737  014664          JSR     PC,TESTN
 4452          027610          A=.
 4453  027610  000224          A$223:  $TAGA$=$TAGA$+1
 4454  027610  017737  151566  027544  MOV     @TRCR,USED      ;STORE COMMAND REGISTER
 4455  027616  042737  175577  027544  BIC     #175577,USED    ;CLEAR UNWANTED BITS
 4456  027624  022737  002200  027544  CMP     #002200,USED    ;TEST THE COMMAND REGISTER
 4457  027632  001413                  BEQ     A$224
 4458  027634  012705  002200          MOV     #02200,R5
 4459  027640  013704  027544          MOV     USED,R4
 4460  027644  013737  001402  001244  MOV     TRCR,REGIST
 4461  027652  104001                  ERROR   1               ;INCORRECT REGISTER MATCHUP
 4462  027654  104402  010155          TYPE    ,MUNRDY
 4463
 4464  027660  000753                  BR      A               ;GET RID OF DERORS
 4465  027662  000225          A$224:  $TAGA$=$TAGA$+1
 4466
 4467  027662  042777  000001  151516  BIC     #BIT0,@TRSR
 4468  027670  052737  000001  027536  BIS     #BIT0,USEA      ;SET GO BIT IN COMMAND
 4469
 4470  027676  013777  027540  151506  MOV     USEB,@TRWC      ;SET WORD COUNT
 4471  027704  013777  027542  151504  MOV     USEC,@TRBA      ;SET INITIAL BUS ADDRESS
 4472  027712  000240                  NOP                     ;THIS LOCATION MAY BE CHANGED TO A HALT TO INSPECT
 4473                                                          ; EQUIPMENT STATUS OR PROGRAM OPERATION
 4474  027714  013777  027536  151460  MOV     USEA,@TRCR      ;SET CONTROL REGISTER
 4475
 4476  027722  004737  012524          JSR     PC,TESTA
 4477  027726  000240                  NOP                     ;THIS LOCATION MAY BE CHANGED TO A HALT TO INSPECT
 4478                                                          ; EQUIPMENT STATUS OR PROGRAM OPERATION
 4479
```

```
4480  027730  032777  000200  151450          BIT     #BIT7,@TRSR
4481  027736  001424                          BEQ     A$225
4482  027740  005737  001264                  TST     WRTFLG          ;TEST FOR RETURN FLAG
4483  027744  001417                          BEQ     1$              ;NO FLAG
4484  027746  032737  000001  001264          BIT     #1,WRTFLG       ;WRITE RETURN?
4485  027754  001003                          BNE     2$              ;YES
4486  027756  022626                          CMP     (SP)+,(SP)+     ;OK READ
4487  027760  005726                          TST     (SP)+
4488  027762  000403                          BR      3$
4489  027764  022626                  2$:     CMP     (SP)+,(SP)+
4490  027766  022626                          CMP     (SP)+,(SP)+
4491  027770  022626                          CMP     (SP)+,(SP)+
4492  027772  005037  001264          3$:     CLR     WRTFLG
4493  027776  004737  013626                  JSR     PC,TESTI
4494  030002  000207                          RTS     PC
4495  030004                          1$:
4496  030004  004737  013626                  JSR     PC,TESTI
4497  030010  000226                  A$225:  $TAGA$=$TAGA$+1
4498  030010  000205                          RTS     R5              ;RETURN TO CALLING PROCEDURE
4499
4500
4501
4502                                  ;-----------------------------------------------------------------------*
4503                                  ;THIS ROUTINE PERFORMS TAPE FUNCTION IT IS CALLED
4504                                  ;BY THE EOTFUN MACRO THIS IS USED FOR EOT FUNCTIONS ONLY
4505
4506  030012  005737  031326          EOTTST: TST     INTFLG          ;CHECK INTERRUPT FLAG
4507  030016  001403                          BEQ     A$226
4508  030020  052737  000100  027536          BIS     #BIT6,USEA      ;ENABLE INTERRUPT
4509  030026  000227                  A$226:  $TAGA$=$TAGA$+1
4510  030026  013737  027536  027526          MOV     USEA,USEA0      ;SAVE LAST FUNCTION
4511  030034  013737  027540  027530          MOV     USEB,USEB0      ;SAVE LAST WORD COUNT
4512  030042  013737  027542  027532          MOV     USEC,USEC0      ;SAVE LAST BUS ADDRESS
4513  030050  004737  014664                  JSR     PC,TESTN
4514  030054  032777  100000  151320          BIT     #BIT15,@TRCR
4515  030062  001412                          BEQ     A$227
4516          030064                  A=.
4517  030064  104000                          ERROR                   ;ERROR-BIT WAS NOT CORRECT
4518  030066  104403  010123                  TYPEF   ,MUNERR
4519
4520  030072  052777  004000  151302          BIS     #PWRCLR,@TRCR   ;DEVICE MASTER RESET
4521  030100  032777  004000  151274  64$:    BIT     #PWRCLR,@TRCR   ;INSTRUCTION CLEAR ?
4522  030106  001374                          BNE     64$             ;IF NO, WAIT FOR IT TO CLEAR
4523  030110  000230                  A$227:  $TAGA$=$TAGA$+1
4524  030110  017737  151266  027544          MOV     @TRCR,USED      ;STORE COMMAND REGISTER
4525  030116  042737  175577  027544          BIC     #175577,USED    ;CLEAR UNWANTED BITS
4526  030124  022737  002200  027544          CMP     #002200,USED    ;TEST THE COMMAND REGISTER
4527  030132  001413                          BEQ     A$230
4528  030134  012705  002200                  MOV     #02200,R5
4529  030140  013704  027544                  MOV     USED,R4
4530  030144  013737  001402  001244          MOV     TRCR,REGIST
4531  030152  104001                          ERROR   1               ;INCORRECT REGISTER MATCHUP
4532  030154  104403  010155                  TYPEF   ,MUNRDY
4533
4534  030160  000741                          BR      A               ;GET RID OF DERORS
4535  030162  000231                  A$230:  $TAGA$=$TAGA$+1
```

```
4536
4537  030162  042777  000001  151216          BIC     #BIT0,@TIRSR
4538  030170  052737  000001  027536          BIS     #BIT0,USEA      ;SET GO BIT IN COMMAND
4539
4540  030176  013777  027540  151206          MOV     USEB,@TRWC      ;SET WORD COUNT
4541  030204  013777  027542  151204          MOV     USEC,@TRBA      ;SET INITIAL BUS ADDRESS
4542  030212  000240                          NOP                     ;THIS LOCATION MAY BE CHANGED TO A HALT TO INSPECT
4543                                                                   EQUIPMENT STATUS OR PROGRAM OPERATION
4544  030214  013777  027536  151160          MOV     USEA,@TRCR      ;SET CONTROL REGISTER
4545
4546  030222  004737  012524                  JSR     PC,TESTA
4547  030226  000240                          NOP                     ;THIS LOCATION MAY BE CHANGED TO A HALT TO INSPECT
4548                                                                   EQUIPMENT STATUS OR PROGRAM OPERATION
4549
4550  030230  000207                          RTS     PC
4551
4552                          ;-------------------------------------------------------------------*
4553
4554                          ;THIS ROUTINE PUTS THE CORRECT PARITY ON ALL WORDS IN
4555                          ;THE OUTPUT BUFFER
4556
4557  030232  012700  031600  FIXBUF: MOV     #OUTPUT,R0      ;SET INDEX
4558          030236          A=.
4559  030236  011001                  MOV     (R0),R1         ;GET THE WORD
4560  030240  005002                  CLR     R2              ;CLEAR THE COUNTER
4561          030242          B=.
4562  030242  105701                  TSTB    R1              ;TEST R1 LOWER
4563  030244  100001                  BPL     A$231
4564  030246  005202                  INC     R2              ;INCREMENT COUNTER
4565  030250  000232          A$231:  $TAGA$=$TAGA$+1
4566  030252  105701                  TSTB    R1              ;TEST R1 LOWER
4567  030254  001402                  BEQ     A$232
4568  030256  006301                  ASL     R1              ;SHIFT R1
4569  030260  000771                  BR      B               ;GO BACK TO B
4570  030262  000233          A$232:  $TAGA$=$TAGA$+1
4571  030264  032702  000001          BIT     #BIT0,R2        ;ODD OR EVEN
4572  030270  001002                  BNE     A$233
4573  030272  052710  000400          BIS     #BIT8,(R0)      ;SET PARITY
4574  030276  000234          A$233:  $TAGA$=$TAGA$+1
4575  030300  005720                  TST     (R0)+
4576  030302  022700  035640          CMP     #INPUT,R0       ;ALL DONE
4577  030306  001356                  BNE     A               ;NO
4578  030310  000207                  RTS     PC
4579
4580                          ;-------------------------------------------------------------------*
4581                          .SBTTL  DEROR HANDLING ROUTINE
4582
4583
4584                          ;-------------------------------------------------------------------*
4585  030312  012737  177777  001266  WRONG:  MOV     #-1,LOC         ;SET UP FOR ERROR                ;ERROR-BIT WAS N
4586  030320  012737  177777  001252          MOV     #-1,ERR         ;SET DEROR FLAG
4587  030326  012600                  MOV     (SP)+,R0        ;GET PC
4588  030330  005740                  TST     -(R0)           ;FIX IT
4589  030332  010037  001254          MOV     R0,XPC          ;STORE IT
4590  030336  011637  001256          MOV     (SP),XSR        ;STORE SR
4591  030342  005746                  TST     -(SP)           ;FIX THE STACK
```

```
4592  030336  005737  031210            TST     VECSTR          ;TEST VECTOR FLAG
4593  030342  001402                    BEQ     A$234
4594  030344  000137  031152            JMP     VECTRX          ;RETURN TO VECTOR
4595  030350  000235            A$234:  $TAGA$=$TAGA$+1
4596  030350  000002                    RTI
4597
4598
4599  030352                    WRONGX:
4600  030352  004537  031414            JSR     R5,SAVE
4601  030356  032777  000020  150616    BIT     #BIT4,@SWR      ;TEST SWITCH BIT 04
4602  030364  001406                    BEQ     A$235
4603  030366  023727  001232  000062    CMP     ERRCNT,#MAXERR  ;TEST DEROR COUNT
4604  030374  101024                    BHI     WRONGY          ;BRANCH IF HIGHER
4605  030376  005237  001232            INC     ERRCNT          ;INCREMENT COUNT
4606  030402  000236            A$235:  $TAGA$=$TAGA$+1
4607  030402  104402  030500            TYPE    ,WHED1
4608  030406  022737  177777  001266    CMP     @#-1,LOC        ;TEST LOCATION FOR ERROR        ;ERROR-B
4609  030414  001402                    BEQ     A$236
4610  030416  104402  030537            TYPE    ,WHED2
4611  030422  000237            A$236:  $TAGA$=$TAGA$+1
4612  030422  104403  005320            TYPEF   ,MCRLF          ;TYPE A CARRIAGE RETURN
4613  030426  104413                    CNVRT                   ;OCTAL TO ASCII CONVERT ROUTINE
4614  030430  030454                    WRDATA                  ;POINTER TO DATA TO PRINT
4615  030432  022737  177777  001266    CMP     @#-1,LOC
4616  030440  001402                    BEQ     WRONGY          ;YES-EXIT
4617  030442  104413                    CNVRT                   ;OCTAL TO ASCII CONVERT ROUTINE
4618  030444  030466                    WRDAT2                  ;DATA TO CONVERT AND PRINT
4619  030446  005237  001232    WRONGY: INC     ERRCNT          ;INCREMENT 1 OVER LIMIT
4620  030452  000207                    RTS     PC
4621  030454  000003            WRDATA: .WORD   3               ;NUMBER OF DATA TO PRINT
4622  030456     006     002            .BYTE   6,2             ;NUMBER OF CHARACTERS AND SPACES TO  PRINT
4623  030460  001250                    TSTPTR                  ;LOCATIONS OF DATA
4624  030462  001254                    XPC
4625  030464  001256                    XSR
4626  030466  000003            WRDAT2: .WORD   3               ;NUMBER OF WORDS TO PRINT
4627  030470     006     002            .BYTE   6,2             ;NUMBER OF CHARACTERS, SPACES
4628  030472  001260                    COR                     ;LOCATION OF  DATA
4629  030474  001262                    ACT
4630  030476  001266                    LOC
4631  030500  051777  041125  042524    WHED1:  .ASCIZ  <377>/SUBTEST    PC          STATUS /
4632  030506  052123  020040  020040
4633  030514  041520  020040  020040
4634  030522  020040  020040  051440
4635  030530  040524  052524  020123
4636  030536     000
4637  030537     040  020040  020040    WHED2:  .ASCIZ  /     CORRECT    ACTUAL      LOCATION /
4638  030544  047503  051122  041505
4639  030552  020124  020040  040440
4640  030560  052103  040525  020114
4641  030566  020040  020040  047514
4642  030574  040503  044524  047117
4643  030602  000040
4644                              .EVEN
4645
4646                              ;-----------------------------------------------------------------*
4647                              .SBTTL  CORE SIZE ROUTINE
```

```
4648
4649                                    ;THIS ROUTINE TESTS THE CORE SIZE AND INSURES IT IS OK
4650
4651  030604  000414                            BR      CORE
4652          030606                            X=.
4653  030606  162700  000002                    SUB     #2,R0
4654  030612  010037  030716                    MOV     R0,LASTAD       ;SET LAST ADDRESS
4655  030616  022626                            POP2SP
4656  030620  023727  030716  040000            CMP     LASTAD,#CORSIZ  ;TEST CORE SIZE
4657  030626  103001                            BHIS    A$237
4658  030630  104000                            ERROR                   ;ERROR-BIT WAS NOT CORRECT
4659  030632  000240                    A$237:  $TAGA$=$TAGA$+1
4660  030632  000137  003502                    JMP     RESTRT
4661  030636  012737  030606  000004  CORE:   MOV     #X,4            ;SET VECTOR
4662  030644  012737  000000  000006          MOV     #0,6            ;SET STATUS
4663  030652  005000                            CLR     R0              ;SET R0 TO ZERO
4664          030654                    A=.
4665  030654  105720                            TSTB    (R0)+           ;TEST ADDRESS
4666  030656  000776                            BR      A               ;CONTINUE
4667
4668          030660                            X=.
4669  030660  012737  177777  030720            MOV     #-1,ADDERR      ;SET DEROR FLAG
4670  030666  000002                            RTI
4671  030670  005037  030720          ADDTST:   CLR     ADDERR          ;CLEAR DEROR FLAG
4672  030674  012737  030660  000004          MOV     #X,4            ;SET VECTOR
4673  030702  012737  000340  000006          MOV     #340,6          ;SET STATUS
4674  030710  005710                            TST     (R0)            ;TEST ADDRESS
4675  030712  000240                            NOP                     ;DELAY
4676  030714  000207                            RTS     PC
4677
4678
4679
4680  030716  000000                    LASTAD: 0                       ;LAST ADDRESS IN CORE
4681  030720  000000                    ADDERR: 0                       ;ADDRESS DEROR FLAG
4682  030722                            VECADD:
4683  030722                            VECSTA:
4684  030722  000000                    PRIOR:  0                       ;PRIORITY LEVEL OF INTERRUPT
4685
4686  030724  104402  030732          TTYTAB: TYPE    ,MTAB
4687  030730  000207                            RTS     PC
4688  030732  000011                    MTAB:   .ASCIZ<11>
4689                                    .EVEN
4690
4691                                    ;-----------------------------------------------------------------------*
4692                                    .SBTTL  TTY DECIMAL OUTPUT ROUTINE
4693
4694                                    ;THIS ROUTINE CONVERTS A OCTAL VALUE IN THE EXCHANGE REGISTER TO
4695                                    ;A DECIMAL VALUE AND PRINTS IT AT THE TTY.
4696
4697
4698          030734                            X=.
4699  030734  000207                            RTS     PC
4700          030736                            CTR=.
4701  030736  000000                            0
4702          030740                    A=.
4703  030740  162700  000012                    SUB     #12,R0          ;SUBTRACT 12
```

```
4704  030744  005237  030736            INC     CTR             ;INCREMENT COUNTER
4705  030750  000407                    BR      A$240
4706  030752  032777  000001  150222    TYPTEN: BIT     #BIT0,@$WR      ;TEST INHIBIT PRINT BIT
4707  030760  001365                    BNE     X               ;EXIT IF SET
4708  030762  005046                    CLR     -(SP)           ;STORE STOP CODE
4709          030764                    B=.
4710  030764  005037  030736            CLR     CTR             ;SET THE COUNTER TO 000
4711  030770  000241            A$240:  $TAGA$=$TAGA$+1
4712  030770  020027  000012            CMP     R0,#12          ;TEST EXCHANGE
4713  030774  103361                    BHIS    A               ;BRANCH IF MORE THAN 12
4714  030776  062700  000260            ADD     #260,R0         ;CONVERT OT TTY CODE
4715  031002  010046                    MOV     R0,-(SP)        ;STACK THE CODE
4716  031004  013700  030736            MOV     CTR,R0          ;SET R0
4717  031010  001365                    BNE     B               ;BRANCH IF R0 NOT 000
4718                                    ;THIS ROUTINE PRINTS TTY CODES PREVIOUSLY PLACED ON THE STACK.
4719                                    ;TYPING A 00 CODE WILL TERMINATE THIS ROUTINE.
4720
4721
4722          031012                    A=.
4723  031012                    TYPSTK:
4724          031012                    $A$=.
4725  031012  105777  150172            TSTB    @TPCSR
4726  031016  100375                    BPL     $A$
4727  031020  012677  150166            MOV     (SP)+,@TPDBR    ;PRINT
4728  031024  001372                    BNE     A               ;NO-CONTINUE
4729          031026                    $A$=.
4730  031026  105777  150156            TSTB    @TPCSR
4731  031032  100375                    BPL     $A$
4732  031034  000207                    RTS     PC
4733
4734                                    ;-------------------------------------------------------------------------*
4735                                    .SBTTL  INTERRUPT VECTOR ROUTINE
4736
4737                                    ;THIS ROUTINE SETS UP CORE ADDRESSES FROM 0 TO 1000 AND WATCHES
4738                                    ;FOR ILLEGAL INTERRUPTS
4739
4740          031036                    A=.
4741  031036  005001                    CLR     R1              ;PRESET R1
4742  031040  012702  000002            MOV     #2,R2           ;PRESET R2
4743          031044                    B=.
4744  031044  010221                    MOV     R2,(R1)+        ;SET ADDRESS
4745  031046  022222                    CMP     (R2)+,(R2)+     ;INCREMENT R2
4746  031050  012721  020367            MOV     #20367,(R1)+    ;SET STATUS
4747  031054  022701  001000            CMP     #1000,R1        ;ALL ADDRESSES SET
4748  031060  003371                    BGT     B               ;NO
4749  031062  012737  000340  000016    MOV     #340,16         ;SET TRAP STATUS
4750  031070  012737  031114  000014    MOV     #VECTOR,14      ;SET TRAP VECTOR
4751  031076  012737  000340  000022    MOV     #340,22         ;SET UP FOR
4752  031104  012737  030304  000020    MOV     #WRONG,20       ;DEROR VECTOR
4753  031112  000207                    RTS     PC
4754  031114  005737  000000    VECTOR: TST     0               ;TEST ADDRESS 000
4755  031120  001746                    BEQ     A               ;BRANCH IF ZERO
4756  031122  021627  001000            CMP     (SP),#1000      ;TEST THE ADDRESS
4757  031126  003005                    BGT     A$241
4758  031130  011637  031210            MOV     (SP),VECSTR     ;STORE VECTOR ADDRESS
4759  031134  022626                    POP2SP
```

```
               031136                         A=.
031136         000137     030304                     JMP     WRONG              ;GET PC AND SR
031142         000242                         A$241: $TAGA$=$TAGA$+1
031142         012737     177777     031210          MOV     #-1,VECSTR         ;SET VECSTR TO -1
031150         000772                                BR      A
031152                                        VECTRX:
031152         004737     030352                     JSR     PC,WRONGX
031156         022737     177777     031210          CMP     #-1,VECSTR         ;TEST FOR BREAK POINT
031164         001410                                BEQ     A$242
031166         013700     031210                     MOV     VECSTR,R0          ;GET ADDRESS
031172         162700     000006                     SUB     #6,R0              ;FIX IT
               031176                         A=.
031176         000000                                HALT
031200         005037     031210                     CLR     VECSTR             ;CLEAR STORAGE
031204         000002                                RTI
031206         000243                         A$242: $TAGA$=$TAGA$+1
031206         000773                                BR      A
031210         000000                         VECSTR: 0

                                              ;-------------------------------------------------------------------*
                                              .SBTTL  INTERRUPT HANDLING ROUTINE

                                              ;THIS ROUTINE HANDLES INTERRUPTS FROM THE EQUIPMENT BEING
                                              ;TESTED.

031212         012737     177777     031326   ENAINT: MOV    #-1,INTFLG         ;SET INTERRUPT TEST FLAG
031220         012737     000340     177776          MOV     #340,PS ;SET PROCESSOR TO 7
031226         012777     031330     150164          MOV     #INTIN,@TRVCT      ;SET UP VECTOR
031234         012777     000340     150160          MOV     #340,@TRRIS        ;SET UP STATUS
031242         052777     000100     150132          BIS     #100,@TRCR         ;ENABLE INTERRUPT
031250         000207                                RTS     PC


031252         005037     177776              CHKINT: CLR    PS                 ;CLEAR PROCESSOR PRIORITY
031256         000240                                NOP                        ;DELAY
031260         000240                                NOP
031262         005737     031326                     TST     INTFLG             ;TEST INTERRUPT FLAG
031266         001401                                BEQ     A$243
031270         104000                                ERROR

031272         000244                         A$243: $TAGA$=$TAGA$+1


031272         042737     000100     164000   DISINT: BIC    #100,164000        ;DISABLE INTERRUPT
031300         005037     031326                     CLR     INTFLG             ;CLEAR INTERRUPT FLAG
031304         005037     000000                     CLR     0                  ;CLEAR VECTOR FLAG
031310         004537     031414                     JSR     R5,SAVE
031314         004737     031114                     JSR     PC,VECTOR
031320         004537     031446                     JSR     R5,SETALL
031324         000207                                RTS     PC

031326         000000                         INTFLG: 0               ;INTERRUPT FLAG = 0 WHEN INTERRUPT OCCURS

031330         042737     000100     164000   INTIN:  BIC    #100,164000        ;DISABLE INTERRUPT
031336         005037     031326                     CLR     INTFLG             ;CLEAR THE FLAG
031342         000240                                NOP
```

```
      031344  000002                       RTI



                                  ;------------------------------------------------------------------------*
                                  .SBTTL  PASS NUMBER ROUTINE

                                  ;THIS ROUTINE KEEPS TRACK OF THE CURRENT PASS COUNT AND
                                  ;DISPLAYS IT WHEN REQUESTED BY THE OPERATOR OR OTHER ROUTINES.


              031346                       PASSUB=.
      031346  000000                       0
              031350                       PASTOL=.
      031350  000000                       0
              031352              A=.
      031352  013700  031350      PAST:    MOV     PASTOL,R0          ;SET EXCHANGE
      031356  063700  031346               ADD     PASSUB,R0          ;ADD SUB TOTAL
      031362  004537  031414               JSR     R5,SAVE
      031366  004737  030752               JSR     PC,TYPTEN
      031372  004537  031446               JSR     R5,SETALL
              031376                       B=.
      031376  000207                       RTS     PC

                                  ;------------------------------------------------------------------------*
                                  .SBTTL  AUTO RESTART ROUTINE

      031400  000240              AUTRES:  NOP
                                  ;PRINT TOTAL PASSES COMPLETED
      031402  004737  011044               JSR     PC,PRETST
      031406  013746  001276               MOV     XTEST,-(SP)        ;STACK THE TEST ADDRESS
      031412  000207                       RTS     PC

                                  ;------------------------------------------------------------------------*
                                  .SBTTL  SAVE AND SET REGISTERS ROUTINE



      031414  010046              SAVE:    MOV     R0,-(SP)
      031416  010146                       MOV     R1,-(SP)
      031420  010246                       MOV     R2,-(SP)
      031422  010346                       MOV     R3,-(SP)
      031424  010446                       MOV     R4,-(SP)
      031426  000205                       RTS     R5



      031430  005726              SET:     TST     (SP)+
      031432  012604                       MOV     (SP)+,R4
      031434  012603                       MOV     (SP)+,R3
      031436  012602                       MOV     (SP)+,R2
      031440  012601                       MOV     (SP)+,R1
      031442  005726                       TST     (SP)+              ;DONT RESTORE R0
      031444  000205                       RTS     R5
```

```
 4872
 4873
 4874
 4875
 4876
 4877  031446  005726                  SETALL: TST     (SP)+
 4878  031450  012604                          MOV     (SP)+,R4
 4879  031452  012603                          MOV     (SP)+,R3
 4880  031454  012602                          MOV     (SP)+,R2
 4881  031456  012601                          MOV     (SP)+,R1
 4882  031460  000205                          RTS     R5
 4883
 4884                                  ;-----------------------------------------------------------------------*
 4885                                  .SBTTL  PRESET ROUTINE
 4886
 4887
 4888                                  ;THIS ROUTINE IS USED TO PRESET ALL VARIABLES, REGISTERS ECT. PRIOR TO
 4889                                  ;PERFORMING A TEST.
 4890
 4891
 4892  031462  012600                  PRESET: MOV     (SP)+,R0        ;GET THE RETURN
 4893  031464  012706  001200                  MOV     #STACK,SP       ;SET STACK
 4894  031470  010046                          MOV     R0,-(SP)        ;RESTORE RETURN
 4895  031472  005000                          CLR     R0              ;CLEAR THE REGISTERS
 4896  031474  005001                          CLR     R1
 4897  031476  005002                          CLR     R2
 4898  031500  005003                          CLR     R3
 4899  031502  005004                          CLR     R4
 4900  031504  005005                          CLR     R5
 4901  031506  005077  147472                  CLR     @TKCSR          ;CLEAR STATUS WORDS
 4902  031512  005077  147472                  CLR     @TPCSR
 4903  031516  005077  147464                  CLR     @TKDBR          ;CLEAR BUFFERS
 4904  031522  005077  147464                  CLR     @TPDBR
 4905  031526  005037  177776                  CLR     PS              ;CLEAR PROCESSOR STATUS
 4906  031532  005037  031346                  CLR     PASSUB          ;CLEAR PASS SUBTO TOTAL
 4907  031536  005037  031350                  CLR     PASTOL          ;CLEAR PASS TOTAL
 4908  031542  005077  147434                  CLR     @SWR            ;CLEAR SWITCH REGISTER
 4909  031546  005037  031210                  CLR     VECSTR
 4910  031552  005037  031326                  CLR     INTFLG          ;CLEAR INTERRUPT ENABLE
 4911  031556  005037  000000                  CLR     0
 4912  031562  005037  001252                  CLR     ERR             ;CLEAR THE DEROR FLAG
 4913  031566  005037  001232                  CLR     ERRCNT          ;CLEAR THE DEROR COUNT
 4914  031572  000207                          RTS     PC
 4915
 4916                                  ;-----------------------------------------------------------------------*
 4917                                  .SBTTL  INPUT - OUTPUT BUFFERS
 4918
 4919  031574  000000  000000          BADOUT: 0,0
 4920  031600                          OUTPUT:                         ;OUTPUT BUFFER STARTS HERE
 4921
  (1)  031600  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
  (1)  031640  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
  (1)  031700  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
  (1)  031740  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
  (1)                                  ;100
  (1
```

```
(1)  032000  000001  000002  000403  001,002,403,004,405,406,007,010,411,412,013,414,015,016,417,020
(1)  032040  000421  000422  000023  421,422,023,424,025,026,427,430,031,032,433,034,435,436,037,040
(1)  032100  000441  000442  000043  441,442,043,444,045,046,447,450,051,052,453,054,455,456,057,460
(1)  032140  000061  000062  000463  061,062,463,064,465,466,067,070,471,472,073,474,075,076,477,100
(1)                                 ;200
(1)
(1)  032200  000501  000502  000103  501,502,103,504,105,106,507,510,111,112,513,114,515,516,117,520
(1)  032240  000121  000122  000523  121,122,523,124,525,526,127,130,531,532,133,534,135,136,537,540
(1)  032300  000141  000142  000543  141,142,543,144,545,546,147,150,551,552,153,554,155,156,557,160
(1)  032340  000561  000562  000163  561,562,163,564,165,166,567,570,171,172,573,174,575,576,177,200
(1)                                 ;300
(1)
(1)  032400  000601  000602  000203  601,602,203,604,205,206,607,610,211,212,613,214,615,616,217,620
(1)  032440  000221  000222  000623  221,222,623,224,625,626,227,230,631,632,233,634,235,236,637,640
(1)  032500  000241  000242  000643  241,242,643,244,645,646,247,250,651,652,253,654,255,256,657,260
(1)  032540  000661  000662  000263  661,662,263,664,265,266,667,670,271,272,673,274,675,676,277,700
(1)                                 ;400
(1)
(1)  032600  000301  000302  000703  301,302,703,304,705,706,307,310,711,712,313,714,315,316,717,320
(1)  032640  000721  000722  000323  721,722,323,724,325,326,727,730,331,332,733,334,735,736,337,340
(1)  032700  000741  000742  000343  741,742,343,744,345,346,747,750,351,352,753,354,755,756,357,760
(1)  032740  000361  000362  000763  361,362,763,364,765,766,367,370,771,772,373,774,375,376,777,400
(1)                                 ;500
(1)
(1)  033000  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)  033040  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)  033100  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)  033140  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)                                 ;600
(1)
(1)  033200  000777  000400  000777  777,400,777,400,777,400,777,400,777,400,777,400,777,400,777,400
(1)  033240  000001  000001  000001  001,001,001,001,001,001,001,001,001,001,001,001,001,001,001,001
(1)  033300  000010  000010  000010  010,010,010,010,010,010,010,010,010,010,010,010,010,010,010,010
(1)  033340  000100  000100  000100  100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100
(1)                                 ;700
(1)
(1)  033400  000777  000400  000777  777,400,777,400,777,400,777,400,777,400,777,400,777,400,777,400
(1)  033440  000002  000002  000002  002,002,002,002,002,002,002,002,002,002,002,002,002,002,002,002
(1)  033500  000020  000020  000020  020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020
(1)  033540  000200  000200  000200  200,200,200,200,200,200,200,200,200,200,200,200,200,200,200,200
(1)                                 ;1000
(1)  033600  000777  000400  000777  777,400,777,400,777,400,777,400,777,400,777,400,777,400,777,400
(1)  033640  000004  000004  000004  004,004,004,004,004,004,004,004,004,004,004,004,004,004,004,004
(1)  033700  000040  000040  000040  040,040,040,040,040,040,040,040,040,040,040,040,040,040,040,040
(1)  033740  000400  000400  000400  400,400,400,400,400,400,400,400,400,400,400,400,400,400,400,400
(1)                                 ;1100
(1)
(1)  034000  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)  034040  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)  034100  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)  034140  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)                                 ;1200
(1)
(1)  034200  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)  034240  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)  034300  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
```

```
 (1)  034340  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)                                  ;1300
 (1)
 (1)  034400  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)  034440  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)  034500  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)  034540  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)                                  ;1400
 (1)
 (1)  034600  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)  034640  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)  034700  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)  034740  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)                                  ;1500
 (1)
 (1)  035000  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)  035040  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)  035100  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)  035140  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)                                  ;1600
 (1)
 (1)  035200  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)  035240  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)  035300  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)  035340  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)                                  ;1700
 (1)
 (1)  035400  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)  035440  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)  035500  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)  035540  000652  000525  000652  652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
 (1)                                  ;2000
 (1)
 (1)  035600  000777  000777  000777  777,777,777,777,777,777,777,777,777,777,777,777,777,777,777,777
 (1)
 (1)  035640  002000                  INPUT:   .BLKW   2000
 (1)
 (1)
4922
4923
4924
4925                                  ;THIS UTILITY CALCULATES PRIORITY LEVEL,SETS UP CSR'S,SETS UP VECTORS.
4926  041640  006337  041766          TRLEV:   ASL     TRPRT           ;BUILD PRIORITY IN THIS LOCATION
4927  041644  006337  041766                   ASL     TRPRT           ;USING ARITHMETIC SHIFTS, ROTATE
4928  041650  006337  041766                   ASL     TRPRT           ;       THE PRIORITY LEVEL PAST
4929  041654  006337  041766                   ASL     TRPRT           ;       THE BIT POSITIONS CORRE-
4930  041660  006337  041766                   ASL     TRPRT           ;       SPONDING TO THE CONDITION CODES
4931  041664  013737  041766  041770           MOV     TRPRT,LESS1     ;MOVE THIS TO LESS1
4932  041672  005337  041770                   DEC     LESS1   ;CREATE THE NEXT LOWEST PRIORITY
4933  041676  042737  000037  041770           BIC     #37,LESS1       ;INSURE THAT THE  TNZVC BITS ARE CLEAR
4934  041704  013700  001420                   MOV     TRVCT,R0        ;PLACE THE BASE VECTOR ADDRESS IN R0
4935  041710  005720                           TST     (R0)+           ;CALCULATE THE RECEIVER INTERRUPT STATUS ADDR.
4936  041712  010037  001422                   MOV     R0,TRRIS        ;STORE IT HERE
4937
4938                                  ;THIS SEGMENT SETS UP POINTERS FOR THE GIVEN TR79. TRBASE IS THE BASE ADDRESS
4939                                  ;OF THE DEVICE
```

```
 4940  041716  013700  001242                MOV     TRBASE,R0               ;COPY THE BASE ADDRESS INTO A SCRATCH REGISTER
 4941  041722  010037  001402                MOV     R0,TRCR ;XXX0
 4942  041726  005200                        INC     R0
 4943  041730  010037  001404                MOV     R0,HTRCR        ;XXX1
 4944  041734  005200                        INC     R0
 4945  041736  010037  001406                MOV     R0,TRSR ;XXX2
 4946  041742  005200                        INC     R0
 4947  041744  010037  001410                MOV     R0,HTRSR        ;XXX3
 4948  041750  005200                        INC     R0
 4949  041752  010037  001412                MOV     R0,TRWC ;XXX4
 4950  041756  005720                        TST     (R0)+
 4951  041760  010037  001416                MOV     R0,TRBA ;XXX6
 4952  041764  000207                        RTS     PC
 4953  041766  000200              TRPRT:    LEVEL4
 4954  041770  000140              LESS1:    LEVEL3  ;LEVEL TO ALLOW INTERRUPTS
 4955
 4956                                        ;THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
 4957  041772  013700  001304        ODD8:   MOV     TEMP1,R0                ;SAVE TEMP1
 4958  041776  005001                        CLR     R1                      ;USE R1 TO CREATE PARITY BIT
 4959  042000  012727  000010                MOV     #8.,(PC)+               ;COUNT THE NUMBER OF BITS TO CALCULATE
 4960  042004  000000              4$:       0                               ;USE THIS WORD AS A LOOP COUNTER
 4961  042006  006000              1$:       ROR     R0                      ;GET A BIT INTO THE CARRY BIT
 4962  042010  005501                        ADC     R1                      ;ADD THE CARRY BIT TO THE BIT COUNT
 4963  042012  005337  042004                DEC     4$                      ;REDUCE THE COUNT. ARE 8 BITS CHECKED?
 4964  042016  001373                        BNE     1$                      ;IF NOT, GO CHECK THE NEXT BIT
 4965  042020  006001                        ROR     R1                      ;IF SO,GET THE SCALE OF THE SUM INTO THE C BIT
 4966  042022  103404                        BCS     2$                      ;IF IT IS ODD, GO SET UP ODD PARITY
 4967  042024  052737  000400  001304        BIS     #BIT8,TEMP1             ;SET ODD PARITY
 4968  042032  000403                        BR      3$
 4969  042034  042737  000400  001304  2$:   BIC     #BIT8,TEMP1             ;CLR EVEN PARITY
 4970                                        ;TEMP1 NOW HAS ODD PARITY CHARACTER
 4971  042042  000207              3$:       RTS     PC
 4972
 4973                                        ;THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
 4974  042044  013700  001304        EVEN8:  MOV     TEMP1,R0                ;SAVE TEMP1
 4975  042050  005001                        CLR     R1
 4976  042052  012727  000010                MOV     #8.,(PC)+
 4977  042056  000000              4$:       0
 4978  042060  006000              1$:       ROR     R0
 4979  042062  005501                        ADC     R1
 4980  042064  005337  042056                DEC     4$
 4981  042070  001373                        BNE     1$
 4982  042072  006001                        ROR     R1
 4983  042074  103004                        BCC     2$
 4984  042076  052737  000400  001304        BIS     #BIT8,TEMP1             ;SET EVEN PARITY
 4985  042104  000403                        BR      3$
 4986  042106  042737  000400  001304  2$:   BIC     #BIT8,TEMP1             ;CLR ODD PARITY
 4987                                        ;TEMP1 NOW HAS EVEN PARITY CHARACTER
 4988  042114  000207              3$:       RTS     PC
 4989  042116  011637  001304        TRPREG: MOV     (SP),TEMP1              ;SAVE PC
 4990  042122  104007                        ERROR   7                       ;DEVICE NON-EXISTENT
 4991  042124  000002                        RTI
```

```
                                        ; ERROR TABLE
042126  000000          .ERRTAB:        0       ;ERROR 0        BIT ERROR (GENERAL)
042130  000000                          0
042132  000000                          0

042134  042260                          EM1     ;ERROR 1        REGISTER ERROR
042136  042713                          DH1
042140  043122                          DT1

042142  042302                          EM2     ;ERROR 2        CONTROLLER ERROR
042144  042713                          DH1
042146  043122                          DT1

042150  042326                          EM3     ;ERROR 3        UNIT ERROR
042152  042713                          DH1
042154  043122                          DT1

042156  042344                          MRECF   ;ERROR 4 REC FAILED TO INTERRUPT
042160  000000                          0
042162  000000                          0

042164  042405                          MTRANF  ;ERROR 5 TRANS FAILED TO INTERRUPT
042166  000000                          0
042170  000000                          0

042172  042440                          MDATAE  ;ERROR 6 DATA ERROR
042174  042750                          DH2
042176  043140                          DT2

042200  042455                          MBUS    ;ERROR  7       DEVICE NON-EXISTENT
042202  043005                          DH3
042204  043156                          DT3

042206  042504                          MUNITR  ;ERROR  10      MISC. UNIT ERROR
042210  043033                          DH4
042212  043170                          DT4

042214  042527                          MWUNIT  ;ERROR  11      INVALID UNIT
042216  000000                          0
042220  000000                          0

042222  042546                          SIZER0  ;ERROR  12      AUTOSIZE ERROR
042224  043044                          DH5
042226  043176                          DT5

042230  042344                          MRECF   ;ERROR  13      AUTOSIZE ERROR
042232  043067                          DH6
042234  043210                          DT6

042236  042567                          SIZER1  ;ERROR 14       AUTOSIZE ERROR
042240  000000                          0
042242  000000                          0

042244  042626                          SIZER2  ;ERROR 15       AUTOSIZE ERROR
042246  000000                          0
042250  000000                          0
```

```
     042252  042663                                   SIZER3   ;ERROR 16        AUTOSIZE ERROR
     042254  043067                                   DH6
     042256  043210                                   DT6



                                                     ;ERROR MESSAGES
     042260  020377  042522  044507  EM1:     .ASCIZ  <377>/ REGISTER ERROR /
(1)  042302  020377  047503  052116  EM2:     .ASCIZ  <377>/ CONTROLLER ERROR /
(1)  042326  020377  047125  052111  EM3:     .ASCIZ  <377>/ UNIT ERROR /
(1)  042344  041777  047117  051124  MRECF:   .ASCIZ  <377>/CONTROLLER FAILED TO INTERRUPT /
(1)  042405     377  047125  052111  MTRANF:  .ASCIZ  <377>/UNIT FAILED TO INTERRUPT /
(1)  042440  042377  052101  020101  MDATAE:  .ASCIZ  <377>/DATA ERROR /
(1)  042455     377  042504  044526  MBUS:    .ASCIZ  <377>/DEVICE NON-EXISTENT  /
(1)  042504  046777  051511  027103  MUNITR:  .ASCIZ  <377>/MISC. UNIT ERROR /
(1)  042527     377  047111  040526  MWUNIT:  .ASCIZ  <377>/INVALID UNIT /
(1)  042546  040777  052125  051517  SIZER0:  .ASCIZ <377>/AUTOSIZE ERROR /
(1)  042567     377  051503  020122  SIZER1:  .ASCIZ<377>/CSR DID NOT AUTOSIZE IN RANGE/
(1)  042626  052377  040522  020520  SIZER2:  .ASCIZ<377>/TRAP! - CSR ADDRESS FAILURE/
(1)  042663     377  047503  052116  SIZER3:  .ASCIZ<377>/CONTROLLER INTERRUPTED/
(1)                                           ;DATA HEADERS FOR ERROR MESSAGES
(1)  042713     377  042522  044507  DH1:     .ASCIZ  <377>/REGISTER    EXPECTED  ACTUAL/
(1)  042750  052777  044516  020124  DH2:     .ASCIZ  <377>/UNIT #      EXPECTED  ACTUAL/
(1)  043005     377  051105  047522  DH3:     .ASCIZ  <377>/ERROR PC    ADDRESS   /
(1)  043033     377  047125  052111  DH4:     .ASCIZ  <377>/UNIT # /
(1)  043044  042777  050130  041505  DH5:     .ASCIZ <377>/EXPECTED    ACTUAL/
(1)  043067     377  050103  020125  DH6:     .ASCIZ <377>/CPU PRIORITY LEVEL WAS: /
(1)
(1)          043122                  .EVEN
(1)                                           ;DATA TABLES FOR ERROR MESSAGES
(1)  043122  000003                  DT1:     3
(1)  043124     006     004                   .BYTE    6,4
(1)  043126  001244                           REGIST   ;DEVICE REGISTER
(1)  043130     006     004                   .BYTE    6,4
(1)  043132  001330                           SAVR5    ;EXPECTED DATA
(1)  043134     006     002                   .BYTE    6,2
(1)  043136  001326                           SAVR4    ;ACTUAL DATA
(1)
(1)  043140  000003                  DT2:     3
(1)  043142     003     007                   .BYTE    3,7
(1)  043144  001236                           SAVLIN   ;UNIT #
(1)  043146     003     007                   .BYTE    3,7
(1)  043150  001330                           SAVR5    ;EXPECTED DATA
(1)  043152     003     002                   .BYTE    3,2
(1)  043154  001326                           SAVR4    ;ACTUAL DATA
(1)
(1)  043156  000002                  DT3:     2
(1)  043160     006     004                   .BYTE    6,4
(1)  043162  001304                           TEMP1    ;PC
(1)  043164     006     002                   .BYTE    6,2
(1)  043166  001244                           REGIST   ;DEVICE REGISTER
(1)
(1)  043170  000001                  DT4:     1
(1)  043172     003     002                   .BYTE    3,2
(1)  043174  001236                           SAVLIN   ;UNIT #
```

INPUT - OUTPUT BUFFERS


```
(1)
(1)  043176  000002                    DT5:    2
(1)  043200     006      005                    .BYTE   6,5
(1)  043202  001336                             SAVED0  ;EXPECTED DATA
(1)  043204     006      002                    .BYTE   6,2
(1)  043206  001340                             SAVE1   ;ACTUAL DATA
(1)
(1)  043210  000001                    DT6:    1
(1)  043212     001      002                    .BYTE   1,2
(1)  043214  001340                             SAVE1   ;ACTUAL DATA
(1)  043216                             CORMAX:
     .NLIST  SEQ,LOC,BIN
     ;.ERROR ;YOU HAVE EXCEEDED 8K CORE LIMITS.
             .END
```

```
A       = 031352    1932#  1934   1936   2150#  2154   2156#  2163   2339#  2341   2343   2362#  2376   2381#
                    2383   2521#  2525   2526#  2530   2531#  2535   3144#  3146   3245#  3251   3318#  3320
                    3322   3324   3326   3782#  3793   4122#  4134   4452#  4464   4516#  4534   4558#  4577
                    4664#  4666   4702#  4713   4722#  4728   4740#  4755   4760#  4764   4771#  4776   4831#
AAA       012056    1527#  1539   1541
ACT       001262     475#  4629
ACTIVE    001342     500#   588*  1253
ADDERR    030720    4669#  4671*  4681#
ADDTST    030670    4671#
ADRCNT    004321     976*  1013*  1023#
AUTRES    031400    4843#
A$0       013762    1946   1953#
A$1       014014    1956   1963#
A$10      014622    2121   2128#
A$100     021206    3059   3066#
A$101     021240    3069   3076#
A$102     021316    3092   3099#
A$103     021346    3102   3108#
A$104     021374    3111   3117#
A$105     021424    3120   3126#
A$106     021572    3158   3165#
A$107     021624    3168   3175#
A$11      014654    2131   2138#
A$110     021656    3178   3185#
A$111     021710    3188   3195#
A$112     021744    3208   3210#
A$113     022136    3263   3270#
A$114     022176    3276   3283#
A$115     022226    3286   3293#
A$116     022256    3296   3303#
A$117     022356    3312   3328#
A$12      014726    2158   2164#
A$120     022532    3351   3358#
A$121     022564    3361   3368#
A$122     022614    3372   3379#
A$123     022704    3395   3402#
A$124     022736    3405   3412#
A$125     022770    3415   3422#
A$126     023022    3425   3432#
A$127     023106    3447   3454#
A$13      014744    2166   2170#
A$130     023140    3457   3464#
A$131     023170    3468   3475#
A$132     023260    3491   3498#
A$133     023312    3501   3508#
A$134     023344    3511   3518#
A$135     023376    3521   3528#
A$136     023466    3545   3552#
A$137     023520    3555   3562#
A$14      015036    2187   2194#
A$140     023552    3565   3572#
A$141     023604    3575   3582#
A$142     023674    3598   3605#
A$143     023726    3608   3615#
A$144     023760    3618   3625#
A$145     024012    3628   3635#
```

```
A$146   024150          3672    3679#
A$147   024210          3685    3692#
A$15    015070          2197    2204#
A$150   024242          3696    3703#
A$151   024274          3706    3713#
A$152   024366          3732    3739#
A$153   024420          3742    3749#
A$154   024450          3752    3759#
A$155   024502          3762    3769#
A$156   024554          3785    3791#
A$157   024634          3810    3817#
A$16    015122          2222    2226#
A$160   024666          3820    3827#
A$161   024720          3830    3837#
A$162   024752          3840    3847#
A$163   025050          3868    3875#
A$164   025102          3878    3885#
A$165   025134          3888    3895#
A$166   025166          3898    3905#
A$167   025216          3908    3914#
A$17    015140          2228    2232#
A$170   025332          3941    3948#
A$171   025364          3951    3958#
A$172   025430          3967    3974#
A$173   025460          3977    3984#
A$174   025574          4012    4019#
A$175   025626          4022    4029#
A$176   025660          4032    4039#
A$177   025712          4042    4049#
A$2     014202          2002    2010#
A$20    015444          2300    2307#
A$200   026026          4077    4084#
A$201   026060          4087    4094#
A$202   026110          4097    4104#
A$203   026142          4107    4114#
A$204   026202          4125    4132#
A$205   026264          4146    4160#
A$206   026340          4163    4177#
A$207   026372          4180    4187#
A$21    015476          2310    2317#
A$210   026446          4195    4208#
A$211   026520          4211    4224#
A$212   026552          4227    4234#
A$213   026724          4281    4288#
A$214   026760          4291    4298#
A$215   027014          4301    4308#
A$216   027052          4319    4326#
A$217   027106          4329    4336#
A$22    015526          2320    2327#
A$220   027154          4345    4352#
A$221   027206          4355    4362#
A$222   027562          4447#
A$223   027610          4453#
A$224   027662          4457    4465#
A$225   030010          4481    4497#
A$226   030026          4507    4509#
```

```
A$227   030110          4515    4523#
A$23    015556          2330    2337#
A$230   030162          4527    4535#
A$231   030250          4563    4565#
A$232   030260          4567    4570#
A$233   030272          4572    4574#
A$234   030350          4593    4595#
A$235   030402          4602    4606#
A$236   030422          4609    4611#
A$237   030632          4657    4659#
A$24    015662          2364    2373#
A$240   030770          4705    4711#
A$241   031142          4757    4762#
A$242   031206          4768    4775#
A$243   031272          4797    4800#
A$25    015700          2380#
A$26    016002          2413    2420#
A$27    016034          2423    2430#
A$3     014234          2013    2021#
A$30    016064          2433    2440#
A$31    016114          2443    2450#
A$32    016212          2473    2480#
A$33    016242          2483    2489#
A$34    016272          2492    2498#
A$35    016320          2501    2506#
A$36    016502          2537    2544#
A$37    016542          2549    2556#
A$4     014342          2042    2049#
A$40    016574          2559    2566#
A$41    016624          2569    2576#
A$42    016764          2600    2607#
A$43    017016          2610    2617#
A$44    017050          2620    2627#
A$45    017100          2630    2637#
A$46    017214          2663    2670#
A$47    017246          2673    2680#
A$5     014374          2052    2059#
A$50    017300          2683    2690#
A$51    017332          2693    2700#
A$52    017454          2725    2732#
A$53    017506          2735    2742#
A$54    017540          2745    2752#
A$55    017572          2755    2762#
A$56    017732          2790    2797#
A$57    017764          2800    2807#
A$6     014514          2087    2094#
A$60    020016          2810    2817#
A$61    020050          2820    2827#
A$62    020212          2854    2860#
A$63    020244          2862    2869#
A$64    020276          2872    2879#
A$65    020330          2882    2889#
A$66    020454          2917    2924#
A$67    020506          2927    2934#
A$7     014546          2097    2104#
A$70    020540          2937    2944#
```

```
A$71     020570           2947    2954#
A$72     020712           2982    2989#
A$73     020744           2992    2999#
A$74     020776           3002    3009#
A$75     021030           3012    3019#
A$76     021124           3039    3046#
A$77     021156           3049    3056#
B      = 031376           4561#   4569    4709#   4717    4743#   4748    4837#
BADOUT   031574           3930    3964    4919#
BBB      012210           1568#   1580    1582
BINWRD   004574           1099#
BIT0   = 000001            356#   2515    2519    2592    2596    2711    2776    2841    3139    4467    4468    4537    4538
                           4571   4706
BIT1   = 000002            355#   1933    2340
BIT10  = 002000            346#   1661    1705    1852    2153    2157    2221    3250    3323
BIT11  = 004000            345#    387    1810    1816    1839    1846    2233    2289
BIT12  = 010000            344#   2710    2840
BIT13  = 020000            343#   2775    2840
BIT14  = 040000            342#    875    1358    1390    1438    1485    1537    1578    1627
BIT15  = 100000            341#   1659    2165    2227    4514
BIT2   = 000004            354#    814    1673    1677    1695    1703
BIT3   = 000010            353#
BIT4   = 000020            352#   4601
BIT5   = 000040            351#   1563    1935    2342    3207    3273    3321    3682
BIT6   = 000100            350#    661    1779    1780    1794    1795    4508
BIT7   = 000200            349#   1663    3145    3311    3319    3325    4480
BIT8   = 000400            348#   4573    4967    4969    4984    4986
BIT9   = 001000            347#
BRER1    010454            768     783    1229#
BRSIZ    002732            750#    773
BRW      003656            820     894#
BRX      003660            821     895#
BUFTA8   027422           4374    4403#
CHKINT   031252           4246    4793#
CHRCNT   004572           1065#   1078#   1096#
CLOCK    004630           1120#   1121    1123
CNVRT  = 104413            542#    839     841     843     845    1160    1162    1164    1222    4613    4617
CONVRT= 104412             540#    791    1178
COR      001260            474#   4628
CORE     030636           4651    4661#
CORMAX   043216           5056#
CORSIZ= 040000            1296#   4656
CR       001270            478#
CSRER2   010415            697    1229#
CSRSIZ   002204            668#    670
CTR    = 030736           4700#   4704#   4710#   4716
CYCLE    010664            822     856     857    1253#
DATABP   005114           1149#   1152    1176    1179#
DATAHD   005102           1148#   1172    1175#
DEVADR   004316            974#   1010    1021#
DH1      042713           4998    5002    5006    5056#
DH2      042750           5018    5056#
DH3      043005           5022    5056#
DH4      043033           5026    5056#
DH5      043044           5034    5056#
DH6      043067           5038    5050    5056#
```

```
DISINT  031272          4803#
DISPLA  001200           439#    602*    836*
DISPRE  000174           430#    602
DT1     043122          4999    5003    5007    5056#
DT2     043140          5019    5056#
DT3     043156          5023    5056#
DT4     043170          5027    5056#
DT5     043176          5035    5056#
DT6     043210          5039    5051    5056#
EM1     042260          4997    5056#
EM2     042302          5001    5056#
EM3     042326          5005    5056#
ENAINT  031212          1977    4241    4377    4410    4785#
END1    011456          1378    1395#   1415
END2    011622          1443#   1461
END3    011762          1490#   1513
END4    012134          1542#   1551
END5    012266          1565    1583#   1594
END6    012454          1632#   1646
EOTTST  030012          2257    3316    3332    3348    3392    3444    3488    3542    3595    4506#
ERASE = 000016           383#   3329    4269
ERR     001252           471#   1356    1361*   1388    1393*   1436    1441*   1483    1488*   1535    1540*   1576    1581*
                        1624*   1625    1630*   4586*   4912*
ERRCNT  001232           459#    589*    869    1189*   4603    4605*   4619*   4913#
ERRFLG  001345           506#    586*    834*    887*   1137*   1150    1166*   1223*
ERRMSG  005070          1147*   1167    1170#
ERTABO  005214          1162    1198#
EVEN8   042044          4974#
EXITER  005150          1184    1189#
FIXBUF  030232          1362    4557#
FUNCTB  027414          4375    4403#
GOEOT = 001026           388#   2254    3138    3313
HALTS   005120          1133    1181#
HEADER  003140           663     751     781     784#
HILIM   004314           973*   1001    1020#
HOLD    001240           466#   1305    1326*   1327    1335*   1822    1830    1989*   1990    1998*   2234    2258    2266
                        2274    2291    2654    2716    2781    2845    2907    2972    3148    3226    3254    3662    3931
                        4001    4066
HTRCR   001404           551#   4943*
HTRSR   001410           553#   4947*
HTRWC   001414           555#
ICOUNT  001222           455#    885     890*
ILCTAB  014254          1978    2026#
ILC00 = 000000           376#
ILC03 = 000406           379#   2026
ILC05 = 001012           381#   2026
ILC06 = 001414           382#   2026
ILC11 = 000022           385#   2026
ILC12 = 000424           386#   2026
ILC14 = 001030           389#   2026
INBUF   010514           943     979    1238#
INIFLG  001344           505#
INPUT   035640          2359    2378    2471    2491    2493    2534    2619    2621    2906    2936    2938    2971    3001
                        3003    3443    3859*   3864    3887    3889    3907    3910    3911    4000    4031    4033    4065
                        4096    4403    4417    4576    4921#
INSTER= 104406           532#    995
```

```
INSTR =  104405            530#    628     640     651     1264    1309
INSTR2   004114            950     962#
INTFLG   031326           4506    4785#   4796    4804#   4811#   4814#   4910#
INTIN    031330           4787    4813#
LASTAD   030716           4654#   4656    4680#
LESS1    041770           4931#   4932#   4933#   4954#
LEVEL0=  000000            368#
LEVEL1=  000040            367#
LEVEL2=  000100            366#
LEVEL3=  000140            365#   4954
LEVEL4=  000200            364#   4953
LEVEL5=  000240            363#
LEVEL6=  000300            362#    583
LEVEL7=  000340            361#   1778
LF       001272            479#
LIMITS   004242            989    1001#
LOBITS   004320            975#   1005    1022#
LOC      001266            477#   4585#   4608    4615    4630
LOCK     001220            454#    889#    902     904    1156
LOKFLG   001346            507#
LOLIM    004312            972#   1003    1019#
LPCNT    001224            456#    884#    885     888#
LSTERR   001234            460#    590#    833#    874#   1134    1136#   1224#
LSTMSG   003766            926#    929#
MASTEK   005622           1158    1229#
MAXERR=  000062           1293#   4603
MAXTST=  000006           1294#
MBNXFL   007115           1229#   2730
MBUS     042455           5021    5056#
MCRLF    005320            664     666     785     823     915     1052    1154    1155    1165    1229#   1262    1280    4612
MCSRX    005552            838    1229#
MDATA    010622           1076    1086    1243#
MDATAE   042440           5017    5056#
MEOT     010203           1229#
MEPASS   005361            837    1229#
MEPSRF   010024           1229#   4017
MERFL    005731           1229#   4324    4334    4350    4360
MERRPC   005624           1161    1229#
MERRX    005577            844    1229#
MERR2    005416           1229#
MERR3    005465           1229#
MESTAD   012456           1610    1634#
METFL    007317           1229#   3163
MFAIL    010312            931    1229#
MIDSFL   006124           1229#   1798
MIENFL   006100           1229#   1783
MILFFL   006264           1229#   2007
MINSRG   006022           1229#   1697
MLOCK    005523            816    1229#
MMRSFL   007361           1229#   3268
MNTFLG   001350            509#
MNXBFL   007157           1229#   2795
MNXMFL   007227           1229#   2859
MOFLEF   006637           1229#   2230
MOLEFL   006520           1229#   2168
MON      003212            790     795#
```

```
MPARFL  010005  1229#  3946
MPASSX  005566   842   1163  1229#
MPCFL   006151  1229#  1865
MPFAIL  005323  1221   1229#
MQM     005314   960   1229#  1285
MR      005404   824   1229#
MRCFL   007301  1229#  2922
MRDERR  006704  1229#  2369
MPDFL   005700  1229#  4206  4222  4232
MRECF   042344  5009   5037  5056#
MREFET  007600  1229#  3550  3603
MREFFL  006771  1229#  2478
MREPFL  010061  1229#  4082
MRIBFL  007764  1229#  3873
MRMVRG  005742  1229#  1675
MRPETF  007476  1229#  3452
MRSTFL  005656  1229#  1323  1342
MRSTU   007336  1229#  3243
MRTOFL  007037  1229#  2605
MRWLPF  006232  1229#  1951
MRWOFL  006661  1229#  2305
MSREFF  006727  1229#  2418
MSRETF  007434  1229#  3400
MSRIFL  007727  1229#  3815
MSRLPF  007670  1229#  3737
MSRVFL  005710  1229#  4286  4296  4306
MTAB    030732  4686   4688#
MTBEX   010244  1229#
MTITLE  001000   437#   608
MTRANF  042405  5013   5056#
MTSTN   005610  1159   1229#  1265
MTSTPC  005511  1229#
MUNERR  010123  1229#  4518
MUNITR  042501  5025   5056#
MUNOFL  006603  1229#  2224
MUNRDY  010155  1229#  4462  4532
MUOLN   006435  1229#  2155
MVECX   005560   840   1229#
MWCFL   010105  1229#  4130
MWEFET  007527  1229#  3496
MWEFFL  006342  1229#  2092
MWELPF  006367  1229#  2126
MWENFL  007650  1229#  3677
MWILPF  006541  1229#  2192
MWIPLF  006170  1229#  1902
MWLPFL  006310  1229#  2047
MWNXFL  007060  1229#  2668
MWPETF  007401  1229#  3356
MWRTFL  005667  1229#  4158  4175  4185
MWTOFL  007015  1229#  2542
MWUNIT  042527  5029   5056#
NEXT    001216   453#   671#   700#   751#   891   1194  1375#  1426#  1473#  1524#  1562#  1586#  1605#
ODD8    041772  4957#
OFLINE= 001436   392#  2217
OUTPUT  031600  2039   2360  2517  2523  2528  2533  2558  2560  3331  3347  3660  3695  3697
                4245   4254  4271  4342  4403  4557  4920#
```

```
PARAM = 104407   534*  630   642   653   1266  1611
PARAM1  004162   978*  996
PARERR  004236   981   983   985   995*  1002  1004  1006
PASCNT  001230   458*  585*  835*  836   866
PASRAT= 000012  1295*
PASSUB= 031346  4827*  4833  4906*
PAST    031352  4832*
PASTOL= 031350  4829*  4832  4907*
PFTAB   005306  1222   1226*
POPRO = 012600   333*  1188
POP1SP= 005726   331*
POP2SP= 022626   335*   603   683   732   740   741   777   892   4655  4759
PRESET  031462  4892*
PRETST  011044  1303*  1360  4845
PRHLT   010323   691    728   734   774   1229*
PRIOR   030722  4684*
PS    = 177776   324*  583*  598*  715*  753*  810*  1778* 1801* 4786* 4793* 4905*
PUSHRO= 010046   332*  1185
PUSH1S= 005746   330*
PUSH2S= 024646   334*
PWRCLR= 004000   387*  1313  1314  1685  1686  1708  1709  1712  1713  1723  1724  1915  1916
                1925  1926  1937  1938  1965  1966  2022  2023  2061  2062  2171  2172  2243
                2244  2282  2283  2345  2346  2578  2579  2639  2640  2764  2765  2829  2830
                2891  2892  2956  2957  3021  3022  3135  3136  3204  3205  3333  3334  3654
                3655  3715  3716  3771  3772  3849  3850  3915  3916  3986  3987  4051  4052
                4520  4521
QV.FLG  001347   508*   587*  846*  882
READ  = 000004   378*  2469  2904  2969  3441  3592  3862  4063  4403  4415
REGIST  001244   468*  1321* 1340* 1351* 1726* 1863* 1875* 1900* 1910* 1949* 1959* 2005* 2016*
                2045* 2055* 2090* 2100* 2124* 2134* 2190* 2200* 2303* 2313* 2323* 2333* 2367*
                2416* 2426* 2436* 2446* 2476* 2486* 2495* 2504* 2540* 2552* 2562* 2572* 2603*
                2613* 2623* 2633* 2666* 2676* 2686* 2696* 2728* 2738* 2748* 2758* 2793* 2803*
                2813* 2823* 2857* 2865* 2875* 2885* 2920* 2930* 2940* 2950* 2985* 2995* 3005*
                3015* 3042* 3052* 3062* 3072* 3095* 3105* 3114* 3123* 3161* 3171* 3181* 3191*
                3266* 3279* 3289* 3299* 3354* 3364* 3375* 3398* 3408* 3418* 3428* 3450* 3460*
                3471* 3494* 3504* 3514* 3524* 3548* 3558* 3568* 3578* 3601* 3611* 3621* 3631*
                3675* 3688* 3699* 3709* 3735* 3745* 3755* 3765* 3788* 3813* 3823* 3833* 3843*
                3871* 3881* 3891* 3901* 3911* 3944* 3954* 3970* 3980* 4015* 4025* 4035* 4045*
                4080* 4090* 4100* 4110* 4128* 4156* 4173* 4183* 4204* 4220* 4230* 4284* 4294*
                4304* 4322* 4332* 4348* 4358* 4460* 4530* 5056
RESREG  005116  1177   1180*
RESTAR  005242  1209   1215*
RESTRT  003502   848    856*  4660
RESOS = 104411   538*  1180
RETURN  001214   452*   592*  670*  699*  750*  822*  825   856*  891*  893   1194* 1196  1225
                1279* 1287* 1288
REWIND= 001020   384*  1929  1941  2286
RO      001274   480*
RTA     012102  1532*
RTB     012234  1573*
RUNAS   024014  1402   3643*
RUNL    014414  1502   2070*
RUNU    015706  1503   2397*
RUNV    016116  1504   2457*
SAVE    031414  4600   4806  4834  4854*
SAVEO   001336   497*   667*  693   698*  742   747*  765   780   5056
```

```
SAVE1   001340            498#   681*   693    695    716*   738*   739*   742    744    749*   765    770*   771
                          778    780    5056
SAVLIN  001236            461#   5056
SAVPC   001334            496#   1028*  1200
SAVR0   001316            489#   1037*  1042
SAVR1   001320            490#   1036*  1043
SAVR2   001322            491#   1035*  1044
SAVR3   001324            492#   1034*  1045
SAVR4   001326            493#   1033*  1046   5056
SAVR5   001330            494#   1032*  1047   5056
SAVSP   001332            495#
SAVOS = 104410            536#   1138
SCOPE = 104400            520#   1395   1443   1490   1542   1587   1632
SCOP1 = 104401            522#
SET     031430            4863#
SETALL  031446            4808   4836   4877#
SIZE    010401            665    1229#
SIZER0  042546            5033   5056#
SIZER1  042567            5041   5056#
SIZER2  042626            5045   5056#
SIZER3  042663            5049   5056#
SIZTAB  027356            4370   4403#  4408
SPACER= 001410            380#   2409   2595   3034   3087   3389   3539   3726   3804   3998   4259   4403
SPACNT  004573            1097#
STACK = 001200            325#   582    668    701    752    811    1195   1217   1380   1428   1475   1526   1567
                          1607   4893
STAT    001246            469#
SUBTST  027430            4376   4403#
SVOS    004330            1032#
SWR     001202            440#   599    601*   611    661    814    875    900    1127   1132   1183   1190   1192
                          1260   1358   1373   1390   1438   1485   1537   1563   1578   1584   1627   4601   4706
                          4908*
SWREG   000176            431#   601
SW00  = 000001            228#   611
SW01  = 000002            227#   1260
SW02  = 000004            226#
SW03  = 000010            222#
SW04  = 000020            221#
SW05  = 000040            219#
SW06  = 000100            218#
SW07  = 000200            217#
SW08  = 000400            215#   1190
SW09  = 001000            214#   900
SW10  = 002000            213#   1192
SW11  = 004000            212#
SW12  = 010000            211#   1127
SW13  = 020000            210#   1132
SW14  = 040000            209#
SW15  = 100000            208#
TABLE1  011460            1379   1401#
TABLE2  011624            1427   1449#
TABLE3  011764            1474   1496#
TABLE4  012136            1525   1548#
TABLE5  012306            1566   1591#
TABLE6  012520            1606   1614   1617   1645#
TBA   = 164006            398#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0112

```
TC      001302            483#  1379*  1382   1383   1384*  1427*  1430   1431   1432*  1474*  1477   1478   1479*
                         1525*  1528   1529   1530*  1566*  1569   1570   1571*  1606*  1617*  1618   1619   1620*
TCR   = 164000            395#
TEMP    010560           1241#  1679*  1699*  1841*  2159*  2520*  2524*  2529*  2534*  3246*
TEMP1   001304            484#   748*   753    769*   787*   793*   795*  1122*  1231   4957   4967*  4969*  4974
                         4984*  4986*  4989*  5056
TEMP2   001306            485#   789*  1233
TEMP3   001310            486#
TEMP4   001312            487#
TEMP5   001314            488#
TESTA   012524           1655#  2597   3156   4476   4546
TESTAA  017612           2772#
TESTAB  020070           1458   2837#
TESTAC  020350           1506   2899#
TESTAD  020610           1507   2964#
TESTAE  021050           1508   3030#
TESTAF  021242           1509   3083#
TESTAG  021426           3134#
TESTAH  021712           2071   2514   2588   2649   2708   2773   2838   2900   2965   3031   3084   3203#  3925
TESTAJ  021752           1404   3217#
TESTAK  022024           1406   3241#
TESTAL  022260           1408   3310#  3387   3483
TESTAM  022616           1409   3386#
TESTAN  023024           1410   3440#
TESTAO  023172           1411   3482#  3537
TESTAP  023400           1412   3536#  3590
TESTAQ  023606           1413   3589#
TESTAS  024042           3653#
TESTAT  024314           1457   3723#
TESTAU  024522           1459   3781#
TESTAV  024562           1500   3800#
TESTAW  024772           1501   1592   3857#
TESTAX  025236           1510   3924#  3995
TESTAY  025500           1511   3994#  4060
TESTAZ  025732           1512   4059#
TESTB   012564           1401   1672#
TESTBA  026144           1460   4121#
TESTC   012646           1403   1450   1496   1548   1694#
TESTCA  026560           2589   4240#
TESTCK  026610           1889   2901   2966   3032   3085   4250#
TESTCX  026210           4145#  4247   4255   4403
TESTD   012760           1451   1722#
TESTDX  026374           4194#  4403   4418
TESTE   013146            672   1452   1778#
TESTEK  026630           2902   2967   4257#
TESTEX  026670           4262   4280#  4403
TESTF   013202            673   1453   1793#
TESTFO  026650           2590   2591   3211   4267#
TESTFX  027016           4272   4318#
TESTG   013234            674   1808#  2140
TESTH   013500           1499   1887#
TESTI   013626           14[    1449   1497   1591   1924#  1975   2035   2115   2181   3242   3724   3858   4493
                         4496
TESTJ   014034           1454   1974#
TESTJ1  014104           1984   1989#
TESTK   014270           1455   2034#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0113

```
TESTL   014446          2081#   4399
TESTM   014550          1456    2114#
TESTN   014664          1304    1711    2152#   2242    2344    4451    4513
TESTO   014764          1498    1549    1888    2180#   2581    3209    3801
TESTP   015072          1405    2215#
TESTQ   015214          1414    2253#
TESTS   015622          2359#   4236
TESTU   015734          2407#   4400
TESTV   016144          2467#   4401    4423
TESTW   016322          1505    2513#
TESTX   016650          2587#
TESTXX  027210          1550    1613    4369#   4402
TESTXY  027436          1593    4407#   4424
TESTY   017120          2648#
TESTZ   017334          2707#
TKCSR   001204           444#    878     945    4901#
TKDBR   001206           445#    880     947     953    4903#
TLAST = 012316          1283    5056#
TPCSR   001210           446#    916     951    1129    4725    4730    4902#
TPDBR   001212           447#    918#    953#   1131#   1681#   1701#   2161#   3248#   4727#   4904#
TRBA    001416           556#   2319    2322    2323    2432    2435    2436    2491    2494    2495    2517#   2523#   2528#
                        2533#   2558    2561    2562    2594#   2619    2622    2623    2682    2685    2686    2713#   2744
                        2747    2748    2778#   2809    2812    2813    2843#   2871    2874    2875    2936    2939    2940
                        3001    3004    3005    3058    3061    3062    3110    3113    3114    3140#   3177    3180    3181
                        3285    3288    3289    3414    3417    3418    3510    3513    3514    3564    3567    3568    3617
                        3620    3621    3695    3698    3699    3751    3754    3755    3783#   3784    3787    3788    3829
                        3832    3833    3887    3890    3891    3966    3969    3970    4031    4034    4035    4096    4099
                        4100    4344    4347    4348    4471#   4541#   4951#
TRBASE  001242           467#   1254#   4940
TRCR    001402           550#    860    1313#   1314    1317    1320    1321    1336    1339    1340    1659    1661    1663
                        1685#   1686    1705    1708#   1709    1712#   1713    1723#   1724    1726    1727#   1728    1734#
                        1735    1742#   1743    1750#   1751    1758#   1759    1765#   1766    1779#   1780    1794#   1795
                        1809#   1810#   1816    1839    1846    1857    1863    1896    1899    1900    1915#   1916    1925#
                        1926    1937#   1938    1945    1948    1949    1965#   1966    1999    2005    2022#   2023    2041
                        2044    2045    2061#   2062    2086    2089    2090    2120    2123    2124    2165    2171#   2172
                        2186    2189    2190    2227    2233#   2243#   2244    2282#   2283    2289#   2299    2302    2303
                        2345#   2346    2412    2415    2416    2472    2475    2476    2518#   2519#   2536    2539    2540
                        2578#   2579    2595#   2596#   2599    2602    2603    2639#   2640    2662    2665    2666    2709#
                        2710#   2714#   2724    2727    2728    2764#   2765    2774#   2775#   2779#   2789    2792    2793
                        2829#   2830    2839#   2840#   2844#   2853    2856    2857    2891#   2892    2916    2919    2920
                        2956#   2957    2981    2984    2985    3021#   3022    3038    3041    3042    3091    3094    3095
                        3135#   3136    3138#   3142#   3157    3160    3161    3204#   3205    3262    3265    3266    3321
                        3323    3325    3333#   3334    3350    3353    3354    3394    3397    3398    3446    3449    3450
                        3490    3493    3494    3544    3547    3548    3597    3600    3601    3654#   3655    3671    3674
                        3675    3715#   3716    3731    3734    3735    3771#   3772    3809    3812    3813    3849#   3850
                        3867    3870    3871    3915#   3916    3940    3943    3944    3986#   3987    4011    4014    4015
                        4051#   4052    4076    4079    4080    4145    4155    4156    4194    4203    4204    4280    4283
                        4284    4318    4321    4322    4454    4460    4474#   4514    4520#   4521    4524    4530    4544#
                        4789#   4941#
TRCRO   001500           568#    613#    633
TRLEV   041640          1257    4926#
TRLVO   001504           570#    615#    656
TRPOK   004610          1110#
TRPREG  042116          4989#
TRPRT   041766          1256#   4926#   4927#   4928#   4929#   4930#   4931    4953#
TRPIS   001422           559#   4788#   4936#
```

CROSS REFERENCE TABLE -- USER SYMBOLS


```
TRSR    001406   552* 1345 1351 1673 1677 1695 1703 1852 1869 1875 1906 1909 1910
                1933 1935 1955 1958 1959 2011 2016 2051 2054 2055 2096 2099 2100
                2130 2133 2134 2153 2157 2196 2199 2200 2221 2309 2312 2313 2340
                2342 2422 2425 2426 2482 2485 2486 2515* 2546 2551 2552 2592* 2609
                2612 2613 2672 2675 2676 2711* 2734 2737 2738 2776* 2799 2802 2803
                2841* 2861 2864 2865 2926 2929 2930 2991 2994 2995 3048 3051 3052
                3101 3104 3105 3139* 3145 3167 3170 3171 3207 3250 3272 3278 3279
                3311 3319 3360 3363 3364 3404 3407 3408 3456 3459 3460 3500 3503
                3504 3554 3557 3558 3607 3610 3611 3681 3687 3688 3741 3744 3745
                3819 3822 3823 3877 3880 3881 3950 3953 3954 4021 4024 4025 4086
                4089 4090 4147 4162 4164 4172 4173 4196 4210 4212 4219 4220 4290
                4293 4294 4328 4331 4332 4467* 4480 4537* 4945*
TRVCT   001420   558#  863 1255* 4787* 4934
TRVCO   001502   569#  614*  645
TRWC    001412   554# 2329 2332 2333 2442 2445 2446 2500 2503 2504 2516* 2522* 2527*
                2532* 2568 2571 2572 2593* 2629 2632 2633 2692 2695 2696 2712* 2754
                2757 2758 2777* 2819 2822 2823 2842* 2881 2884 2885 2946 2949 2950
                3011 3014 3015 3068 3071 3072 3119 3122 3123 3141* 3187 3190 3191
                3295 3298 3299 3371 3374 3375 3424 3427 3428 3467 3470 3471 3520
                3523 3524 3574 3577 3578 3627 3630 3631 3705 3708 3709 3761 3764
                3765 3839 3842 3843 3897 3900 3901 3976 3979 3980 4041 4044 4045
                4106 4109 4110 4123* 4124 4127 4172 4179 4182 4183 4226 4229 4230
                4300 4303 4304 4354 4357 4358 4470* 4540* 4949*
TRYIT   027546   1891 1928 1940 1981 2036 2082 2116 2182 2216 2285 2408 2468 2650
                2903 2968 3033 3086 3657 3725 3803 3861 3927 3997 4062 4242 4251
                4258 4268 4385 4414 4444#
TR.END  001506   572#  619
TR.MAP  001500   500  566# 588  617  667  695* 698  717* 724* 744* 747  754  757*
                764* 776* 778* 779* 788
TSA    = 000172  400#
TSR    = 164002  396#
TSTNO   001226   457#  591* 669* 784* 1203 1228 1269 1275 1277 1374* 1425* 1472* 1523*
                1561* 1604#
TSTPTR  001250   470# 1303* 1382* 1430* 1477* 1528* 1569* 1618* 4623
TST1    011336  1272 1287  1374# 1612
TST2    011516  1375 1425#
TST3    011656  1426 1472#
TST4    012030  1473 1523#
TST5    012146  1524 1561#
TST6    012316  1562 1604# 5056
TST7   = ****** U  1605
TTST    003556   817#  818*  820*  821*  876#
TTYEND  010556  1240#
TTYTAB  030724  4686#
TVA    = 000170  399#
TWC    = 164004  397#
TYPDAT  005104  1153 1173 1176#
TYPE   = 104402   524# 608  664  665  666  691  728  734  774  785  786  816  823
                824  837  838  840  842  844  915  928  930  941  960 1053 1086
                1154 1155 1158 1159 1161 1163 1165 1169 1174 1221 1262 1280 1285
                1675 1697 2155 2369 3243 4462 4607 4610 4686
TYPEF  = 104403   526#  697  746  768  783 1323 1342 1783 1798 1865 1902 1951 2007
                2047 2092 2126 2168 2192 2224 2230 2305 2418 2478 2542 2605 2668
                2730 2795 2859 2922 3163 3268 3356 3400 3452 3496 3550 3603 3677
                3737 3815 3873 3946 4017 4082 4130 4158 4175 4185 4206 4222 4232
                4286 4296 4306 4324 4334 4350 4360 4518 4532 4612
```

```
TYPEL = 104404          528#  1353  1877  1961  2018  2057  2102  2136  2202  2315  2325  2335  2428
                       2438  2448  2554  2564  2574  2615  2625  2635  2678  2688  2698  2740  2750
                       2760  2805  2815  2825  2867  2877  2887  2932  2942  2952  2987  2997  3007
                       3017  3044  3054  3064  3074  3097  3173  3183  3193  3281  3291  3301  3366
                       3377  3410  3420  3430  3462  3473  3506  3516  3526  3560  3570  3580  3613
                       3623  3633  3690  3701  3711  3747  3757  3767  3825  3835  3845  3883  3893
                       3903  3913  3956  3972  3962  4027  4037  4047  4092  4102  4112
TYPMSG  004774         1151  1154#
TYPSTK  031012         4723#
TYPTEN  030752         4706#  4835
USEA    027536         2254#  3313#  3329#  3345#  3389#  3441#  3485#  3539#  3592#  4439#  4444#  4448  4468#
                       4474   4508#  4510   4538#  4544
USEAO   027526         4434#  4448#  4510#
USEB    027540         2255#  2361   3314#  3330#  3346#  3390#  3442#  3486#  3540#  3593#  3960  4300  4302
                       4338   4440#  4445#  4449   4470   4511   4540
USEBO   027530         4435#  4449#  4511#
USEC    027542         2256#  3315#  3331#  3347#  3391#  3443#  3487#  3541#  3594#  4441#  4446#  4450  4471
                       4512   4541
USECO   027532         4436#  4450#  4512#
USED    027544         4442#  4454#  4455#  4456  4459  4524#  4525#  4526  4529
USEDO   027534         4437#
VECADD  030722         4682#
VECER3  010433          746   1229#
VECSIZ  002404          699#
VECSTA  030722         4683#
VECSTR  031210         4592   4758#  4763#  4767  4769  4773#  4777#  4909#
VECTOR  031114         4750   4754#  4807
VECTRX  031152         4594   4765#
WEOF  = 001034          391#  2083   2117  3485
WHED1   030500         4607   4631#
WHED2   030537         4610   4637#
WIDB  = 000432          390#  1892   2183
WRDATA  030454         4614   4621#
WRDAT2  030466         4618   4626#
WRDCNT  004570         1061#  1087#  1095#
WRITE = 000402          377#  2037   2518  2651  2709  2774  2839  3345  3658  3928  4243  4252  4403
WRONG   030304         4585#  4752   4761
WRONGX  030352         4599#  4766
WRONGY  030446         4604   4616   4619#
WRTFLG  001264          476#  4371#  4409#  4482  4484  4492#
WTBS.F  005072         1168   1171#
X     = 030734         4652#  4661   4668#  4672  4698#  4707
XBX     004666         1128   1130   1132#
XCSR    003514          839    858#
XERR    003536          845    867#
XHEAD   005631          786   1229#
XPASS   003530          843    864#  1164
XPC     001254          472#  4589#  4624
XSR     001256          473#  4590#  4625
XSTATQ  010502          792   1229#
XTEST   001276          481#  4846
XTSTN   005222         1160   1201#
XVEC    003522          841    861#
XX    = 164010          567#   571#
YY    = 000200          567#   571#
ZERO    001300          482#
```

```
ZZ      = 000001   567#   571#
$A$     = 031026  4724#  4726   4729#  4731
$CRAP   = 177777     1#  1366#  1368#  1417#  1419#  1464#  1466#  1515#  1517#  1553#  1555#  1596#  1598#
$E      = 000010     1#  1375   1376#  1426   1427#  1473   1474#  1524   1525#  1562   1563#  1605   1606#
$ENDAD  003470   425    606    850#  1181
$N      = 000006     1#  1366   1369   1376#  1417   1420   1427#  1464   1467   1474#  1515   1518   1525#
                  1553   1556   1563#  1596   1599   1606#  5056#
$TAGA$= 000244   1293#  1946   1953#  1956   1963#  2002   2010#  2013   2021#  2042   2049#  2052   2059#
                 2087   2094#  2097   2104#  2121   2128#  2131   2138#  2158   2164#  2166   2170#  2187
                 2194#  2197   2204#  2222   2226#  2228   2232#  2300   2307#  2310   2317#  2320   2327#
                 2330   2337#  2364   2373#  2380#  2413   2420#  2423   2430#  2433   2440#  2443   2450#
                 2473   2480#  2483   2489#  2492   2498#  2501   2506#  2537   2544#  2549   2556#  2559
                 2566#  2569   2576#  2600   2607#  2610   2617#  2620   2627#  2630   2637#  2663   2670#
                 2673   2680#  2693   2690#  2693   2700#  2725   2732#  2735   2742#  2745   2752#  2755
                 2762#  2790   27 7#  2800   2807#  2810   2817#  2820   2827#  2854   2860#  2862   2869#
                 2872   2879#  2882   2889#  2917   2924#  2927   2934#  2937   2944#  2947   2954#  2982
                 2989#  2992   2995#  300?   3009#  3012   3019#  3039   3046#  3049   3056#  3059   3066#
                 3069   3076#  3092   3099#  3102   3108#  3111   3117#  3120   3126#  3158   3165#  3168
                 3175#  3178   3185#  3188   3195#  3208   3210#  3263   3270#  3276   3283#  3286   3293#
                 3296   3303#  3312   3328#  3351   3358#  3361   3368#  3372   3379#  3395   3402#  3405
                 3412#  3415   3422#  3425   3432#  3447   3454#  3457   3464#  3468   3475#  3491   3498#
                 3501   3508#  3511   3518#  3521   3528#  3545   3552#  3555   3562#  3565   3572#  3575
                 3582#  3598   3605#  3608   3615#  3618   3625#  3628   3635#  3672   3679#  3685   3692#
                 3696   3703#  3706   3713#  3732   3739#  3742   3749#  3752   3759#  3762   3769#  3785
                 3791#  3810   3817#  3820   3827#  3830   3837#  3840   3847#  3868   3875#  3878   3885#
                 3888   3895#  3898   3905#  3908   3914#  3941   3948#  3951   3958#  3967   3974#  3977
                 3984#  4012   4019#  4022   4029#  4032   4039#  4042   4049#  4077   4084#  4087   4094#
                 4097   4104#  4107   4114#  4125   4132#  4146   4160#  4163   4177#  4180   4187#  4195
                 4208#  4211   4224#  4227   4234#  4281   4288#  4291   4298#  4301   4308#  4319   4326#
                 4329   4336#  4345   43 2#  4355   4362#  4447#  4453#  4457   4465#  4481   4497#  4507
                 4509#  4515   4523#  4527   4535#  4563   4565#  4567   4570#  4572   4574#  4593   4595#
                 4602   4606#  4609   4611#  4657   4659#  4705   4711#  4757   4762#  4768   4775#  4797
                 4800#
$TAGC$= 000000   1293#
$Y      = 000014    512#   520    522#   524#   526#   528#   530#   532#   534#   536#   538#   540#   542#
                    544#
.       = 043216    410#   411    414#   421#   426#   429#   432#   436#   438#   511#   565#   568#   569#
                    570#   661#  1211   1229#  1239#  1242#  1244#  1640#  1932   2150   2156   2339   2362
                   2381   2521   2526   2531   3144   3245   3318   3782   4122   4452   4516   4558   4561
                   4652   4664   4668   4698   4700   4702   4709   4722   4724   4729   4740   4743   4760
                   4771   4827   4829   4831   4837   4921#  5056#

.BEGIN  003254   810#
.CNVRT  004420   543   1054#
.CONVR  004414   541   1053#
.EOP    003364   832#  1586   1605
.EROR   004642   417   1127#
.ERRTA  042126  1146   4993#
.INSTE  004102   533    958#
.INSTR  003776   531    937#
.INST1  004016   941#   961
.MSG    004020   939*   942#
.PARAM  004122   535    969#
.PFAIL  005230   415    584   1208#  1216
.RESOS  004362   539   1042#
.SAVOS  004322   537   1028#
.SCOPE  003544   521    874#
```

```
.SCOP1  003662            523     900#
.START  001510            433     581#    592
.TRPSR  004576            419    1107#
.TRPTA  001352            519#   1112
.TYPE   003706            525     910#
.TYPEF  003752            527     926#
.TYPEL  003764            529     928#
```

CROSS REFERENCE TABLE -- MACRO NAMES


```
ADDTST      1#
ASKOPR      1#
AUTRES      1#
BELL        1#  1679  1699  2159  3246
BREAK       1#  1386  1434  1481  1532  1573  1622  4472  4477  4542  4547
CHKINT      1#  4246
CRLF        1#  4612
DEROR       1#  1319  1338  1349  1861  1873  1898  1908  1947  1957  2003  2014  2043  2053  2088
            2098  2122  2132  2188  2198  2301  2311  2321  2331  2365  2414  2424  2434  2444  2474
            2484  2493  2502  2538  2550  2560  2570  2601  2611  2621  2631  2664  2674  2684  2694
            2726  2736  2746  2756  2791  2801  2811  2821  2855  2863  2873  2883  2918  2928  2938
            2948  2983  2993  3003  3013  3040  3050  3060  3070  3093  3103  3112  3121  3159  3169
            3179  3189  3264  3277  3287  3297  3352  3362  3373  3396  3406  3416  3426  3448  3458
            3469  3492  3502  3512  3522  3546  3556  3566  3576  3599  3609  3619  3629  3673  3686
            3697  3707  3733  3743  3753  3763  3786  3811  3821  3831  3841  3869  3879  3889  3899
            3909  3942  3952  3968  3978  4013  4023  4033  4043  4078  4088  4098  4108  4126  4153
            4170  4181  4201  4217  4229  4282  4292  4302  4320  4330  4346  4356  4458  4528
DISINT      1#
DO          1#  4806  4834
ENAINT      1#  1977  4240  4377  4410
EOTFUN      1#  2253  3313  3329  3345  3389  3440  3485  3539  3592
ERROR     336#   690   696   727   733   745   767   782  1322  1341  1352  1730  1737  1745  1753
            1761  1768  1782  1797  1818  1848  1864  1876  1901  1911  1950  1960  2006  2017  2046
            2056  2091  2101  2125  2135  2167  2191  2201  2223  2229  2304  2314  2324  2334  2368
            2417  2427  2437  2447  2477  2487  2496  2505  2541  2553  2563  2573  2604  2614  2624
            2634  2667  2677  2687  2697  2729  2739  2749  2759  2794  2804  2814  2824  2858  2866
            2876  2886  2921  2931  2941  2951  2986  2996  3006  3016  3043  3053  3063  3073  3096
            3106  3115  3124  3162  3172  3182  3192  3267  3280  3290  3300  3355  3365  3376  3399
            3409  3419  3429  3451  3461  3472  3495  3505  3515  3525  3549  3559  3569  3579  3602
            3612  3622  3632  3676  3689  3700  3710  3736  3746  3756  3766  3789  3814  3824  3834
            3844  3872  3882  3892  3902  3912  3945  3955  3971  3981  4016  4026  4036  4046  4081
            4091  4101  4111  4129  4157  4174  4184  4205  4221  4231  4285  4295  4305  4323  4333
            4349  4359  4461  4517  4531  4658  4798  4990
GETADD      1#
GETTST      1#
GO          1#  1304  1362  1711  1887  1889  1974  1977  2034  2070  2114  2140  2180  2242  2257
            2344  2513  2581  2587  2589  2590  2591  2597  2648  2707  2772  2837  2899  2901  2902
            2964  2966  2967  3030  3032  3083  3085  3156  3209  3211  3241  3316  3332  3348  3386
            3392  3444  3482  3488  3536  3542  3589  3595  3723  3800  3857  3924  3994  4059  4236
            4241  4246  4247  4255  4262  4272  4377  4399  4400  4401  4410  4423  4451  4476  4493
            4495  4513  4546  4765  4807  4835  4845
HERE        1#  1953  1963  2010  2021  2049  2059  2094  2104  2128  2138  2164  2170  2194  2204
            2226  2232  2307  2317  2327  2337  2373  2380  2420  2430  2440  2450  2480  2489  2498
            2506  2544  2556  2566  2576  2607  2617  2627  2637  2670  2680  2690  2700  2732  2742
            2752  2762  2797  2807  2817  2827  2860  2869  2879  2889  2924  2934  2944  2954  2989
            2999  3009  3019  3046  3056  3066  3076  3099  3108  3117  3126  3165  3175  3185  3195
            3210  3270  3283  3293  3303  3328  3358  3368  3379  3402  3412  3422  3432  3454  3464
            3475  3498  3508  3518  3528  3552  3562  3572  3582  3605  3615  3625  3635  3679  3692
            3703  3713  3739  3749  3759  3769  3791  3817  3827  3837  3847  3875  3885  3895  3905
            3914  3948  3958  3974  3984  4019  4029  4039  4049  4084  4094  4104  4114  4132  4160
            4177  4187  4208  4224  4234  4288  4298  4308  4326  4336  4352  4362  4447  4453  4465
            4497  4509  4523  4535  4565  4570  4574  4595  4606  4611  4659  4711  4762  4775  4800
JUMP        1#
LOOP        1#  1356  1388  1436  1483  1535  1576  1625
PASNOW      1#
PASS        1#
```

```
PAUSE       1#
PRESET      1#
PRGEND      1#    826
PRGFRT      1#    186
PRINTM      1#
READY       1#   4723    4729
RESTRT      1#   4660
RETURN      1#   1364    1666    1688    1715    1771    1786    1802    1854    1881    1918    1968    1987    2025    2064
          2075   2107    2141    2151    2174    2207    2246    2348    2384    2401    2452    2461    2508    2582    2642
          2702   2767    2832    2894    2959    3024    3078    3128    3197    3212    3221    3234    3305    3340    3381
          3434   3477    3530    3584    3637    3647    3718    3774    3794    3852    3919    3989    4054    4116    4135
          4188   4237    4248    4256    4263    4273    4310    4364    4550    4578    4620    4676    4687    4699    4732
          4753   4790    4809    4838    4847    4914
RUN         1#   2072    2397    2457    3217    3337    3643
SAVE        1#   4599    4806    4834
SET         1#
SETALL      1#   4808    4836
SKIP        1#   1946    1956    2002    2013    2042    2052    2087    2097    2121    2131    2158    2166    2187    2197
          2222   2228    2300    2310    2320    2330    2364    2413    2423    2433    2443    2473    2483    2492    2501
          2537   2549    2559    2569    2600    2610    2620    2630    2663    2673    2683    2693    2725    2735    2745
          2755   2790    2800    2810    2820    2854    2862    2872    2882    2917    2927    2937    2947    2982    2992
          3002   3012    3039    3049    3059    3069    3092    3102    3111    3120    3158    3168    3178    3188    3208
          3263   3276    3286    3296    3312    3351    3361    3372    3395    3405    3415    3425    3447    3457    3468
          3491   3501    3511    3521    3545    3555    3565    3575    3598    3608    3618    3628    3672    3685    3696
          3706   3732    3742    3752    3762    3785    3810    3820    3830    3840    3868    3878    3888    3898    3908
          3941   3951    3967    3977    4012    4022    4032    4042    4077    4087    4097    4107    4125    4146    4163
          4180   4195    4211    4227    4281    4291    4301    4319    4329    4345    4355    4457    4481    4507    4515
          4527   4563    4567    4572    4593    4602    4609    4657    4705    4757    4768    4797
TAPFUN      1#   1891    1928    1940    2036    2081    2116    2182    2215    2285    2407    2467    2650    2903    2968
          3033   3086    3657    3725    3803    3861    3927    3997    4062    4242    4250    4257    4267
TBGL        1#
TBUSY       1#   3321
TEOF        1#
TEOT        1#   3145    3310    3319    4480
TERR        1#   1659    2165    2227    4514
TFP         1#   1672    1677    1694    1703
TICMD       1#
TIDB        1#
TIMOUT      1#
TINH        1#   2515    2592    2711    2776    2841    4467    4537
TINT        1#   1779    1780    1793    1795
TLDP        1#   1935    2342    3207
TMTE        1#
TNXM        1#
TOL         1#   1852    2152    2157    2221    3250
TPCLR       1#   1810    1816    1839    1846    2233    2289
TRCRDY      1#   1663    3325
TRDCNT      1#
TRPE        1#
TRWS        1#   1933    2340
TSTE        1#
TTRDY       1#   1661    1705    3323
TUAE0       1#   2710
TUAE1       1#   2775
TWPE        1#
TYPLET      1#
```

```
TYPTAB      1#
TYPTEN      1#  4834
VECTOR      1#  4806
WRONG       1#
WRONGX      1#
$BUFFE      1#  1235
$CONFI      1#  4925
$CYCLE      1#  1245
$DELAY      1#  1305  1327  1822  1830  1990  2234  2258  2266  2274  2291  2654  2716  2781  2845
           2907  2972  3148  3226  3254  3662  3931  4001  4066
$DPOIN      1#
$EOP        1#   826
$ERTAB      1#  4992
$FINI       1#  5056
$GETPA      1#   627   639   651  1264  1609
$HEADE      1#   186
$HERES      1#  1953  1963  2010  2021  2049  2059  2094  2104  2128  2138  2164  2170  2194  2204
           2226  2232  2307  2317  2327  2337  2373  2380  2420  2430  2440  2450  2480  2489  2498
           2506  2544  2556  2566  2576  2607  2617  2627  2637  2670  2680  2690  2700  2732  2742
           2752  2762  2797  2807  2817  2827  2860  2869  2879  2889  2924  2934  2944  2954  2989
           2999  3009  3019  3046  3056  3066  3076  3099  3108  3117  3126  3165  3175  3185  3195
           3210  3270  3283  3293  3303  3328  3358  3368  3379  3402  3412  3422  3432  3454  3464
           3475  3498  3508  3518  3528  3552  3562  3572  3582  3605  3615  3625  3635  3679  3692
           3703  3713  3739  3749  3759  3769  3791  3817  3827  3837  3847  3875  3885  3895  3905
           3914  3948  3958  3974  3984  4019  4029  4039  4049  4084  4094  4104  4114  4132  4160
           4177  4187  4208  4224  4234  4288  4298  4308  4326  4336  4352  4362  4447  4453  4465
           4497  4509  4523  4535  4565  4570  4574  4595  4606  4611  4659  4711  4762  4775  4800
$INTER      1#
$INTSE      1#
$JUNK       1#   567
$LINEU      1#
$LVLTS      1#
$MRESE      1#  1313  1685  1708  1712  1722  1915  1924  1937  1965  2022  2061  2171  2243  2282
           2345  2578  2639  2764  2829  2891  2956  3021  3134  3203  3333  3653  3715  3771  3849
           3915  3986  4051  4520
$MRR        1#
$MRRD       1#
$MRRW       1#
$MRRWR      1#
$MRWD       1#
$MSG        1#  1229
$NOSLA      1#  4989
$PAR8       1#  4956
$PFAIL      1#  1204
$PRINT      1#
$QUEST      1#   594
$RESTO      1#
$SCOPE      1#   870
$SKIP$      1#  1946  1956  2002  2013  2042  2052  2087  2097  2121  2131  2158  2166  2187  2197
           2222  2228  2300  2310  2320  2330  2364  2413  2423  2433  2443  2473  2483  2492  2501
           2537  2549  2559  2569  2600  2610  2620  2630  2663  2673  2683  2693  2725  2735  2745
           2755  2790  2800  2810  2820  2854  2862  2872  2882  2917  2927  2937  2947  2982  2992
           3002  3012  3039  3049  3059  3069  3092  3102  3111  3120  3158  3168  3178  3188  3208
           3263  3276  3286  3296  3312  3351  3361  3372  3395  3405  3415  3425  3447  3457  3468
           3491  3501  3511  3521  3545  3555  3565  3575  3598  3608  3618  3628  3672  3685  3696
           3706  3732  3742  3752  3762  3785  3810  3820  3830  3840  3868  3878  3888  3898  3908
```

```
CZTRABO MACY11 30(1046)  14-DEC-77  12:32  PAGE 124
CZTRAB.HED    14-DEC-77 12:19          CROSS REFERENCE TABLE -- MACRO NAMES                                    SEQ 0121

          3941    3951    3967    3977    4012    4022    4032    4042    4077    4087    4097    4107    4125    4146    4163
          4180    4195    4211    4227    4281    4291    4301    4319    4329    4345    4355    4457    4481    4507    4515
          4527    4563    4567    4572    4593    4602    4609    4657    4705    4757    4768    4797
$TRPDE      1#     520     522     524     526     528     530     532     534     536     538     540     542
$TR79       1#    1292
$TSTN       1#    1369    1420    1467    1518    1556    1599
$TUNIT      1#
$UNIBU      1#
$VARIA      1#     435
$XZ         1#    1366    1368    1417    1419    1464    1466    1515    1517    1553    1555    1596    1598
$$$$$B      1#

. ABS.  043216     000

 ERRORS DETECTED:  0

 CZTRAB.BIN,CZTRAB.LST/CRF/SOL/NL:TOC=CZTRAB.P11,CZTRAB.HED
 RUN-TIME:  13 18 2 SECONDS
 RUN-TIME RATIO: 165/34=4.7
 CORE USED:  37K  (73 PAGES)
```

F10