KW11-L

LINE FREQUENCY CLOCK TEST

MD-11-DZKWA-E

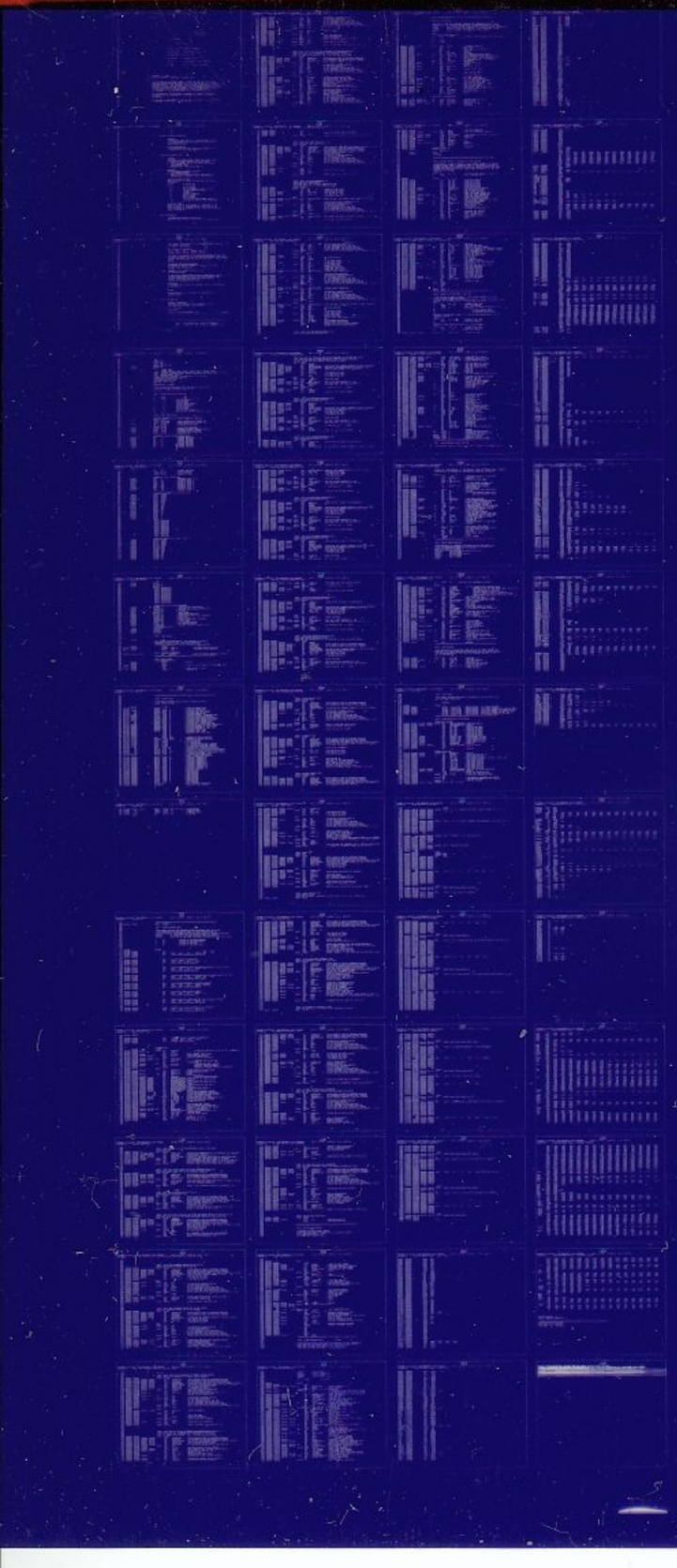
EP DZKWA-E-DL-A

OCT 1976

FICHE 1 OF 1

digital

Made in U.S.A.



.REM

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZKWA-E-D
PRODUCT NAME: LINE FREQUENCY CLOCK TEST
DATE REVISED: FEB. 21, 1976
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: J RODENHISER/J LACEY/J COMEAU
REVISED: FEB. , 1976 BY B. BURGESS

1.0 GENERAL PROGRAM INFORMATION

1.1 ABSTRACT
THIS PROGRAM TESTS THE KW11L LINE FREQUENCY CLOCK. IT VALIDATES PROPER OPERATION UNDER BOTH INTERRUPT AND NON-INTERRUPT MODES.

1.2 SYSTEM REQUIREMENTS
THIS PROGRAM IS DESIGNED TO RUN ON ANY PDP-11 WITH 4K OF MEMORY AND A KW11 LINE FREQUENCY CLOCK.

2.0 OPPERATING INSTRUCTIONS

2.1 LOADING
PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED
A ABSOLUTE LOADER PROGRAM MUST BE IN MEMORY
B PLACE THE BINARY TAPE IN THE PAPER TAPE READER
C LOAD ADDRESS 17500
D DEPRESS START (TAPE SHOULD READ IN)

2.2 STARTING
PROGRAM STARTING ADDRESS IS 000200
A LOAD ADDRESS 000200
B SELECT SWITCH REGISTER OPTIONS (SEE SECTION 2.3)
C DEPRESS START (PROGRAM SHOULD START RUNNING)

2.3 SWITCH REGISTER OPTIONS
HERE IS A LIST OF CONSOLE SWITCHES AND THEIR EFFECT ON THE PROGRAM...

SWITCH	ACTION IF SET
15	HALT ON ERROR
14	LOOP ON CURRENTLY EXECUTING TEST
13	INHIBIT ERROR PRINTOUTS
12	(UNUSED)
11	INHIBIT ITERATIONS
10	BELL ON ERROR
9	LOOP ON ERROR
8	LOOP ON TEST SPECIFIED IN SWR<7:0>
7-0	# OF TEST TO LOOP ON (ONLY WHEN SWR9 = 1)

2.4 EXECUTION TIMES
EXECUTION TIME FOR THIS PROGRAM IS DEPENDENT ON THE MODEL OF PDP-11 IT IS BEING RUN ON. FOR A PDP-11 40 ABOUT 5 SECONDS IS NECESSARY TO DO 1 PASS OF THE PROGRAM WITHOUT ITERATIONS.

3.0 ERROR INFORMATION

3.1 STANDARD ERROR REPORTING PROCEDURES
ERROR PRINTOUTS CONSIST OF FROM 4 TO 9 COLUMNS OF DATA. A

DATA HEADER, AND POSSIBLY A SHORT ERROR MESSAGE DESCRIBING
THE ERROR. FOR EXAMPLE...

```
CLOCK FAILED TO INTERRUPT
PC      PS      SP      TEST#   LKS
002262  000344  000764  000007  000300
```

THE FIRST 4 COLUMNS OF OF THE ERROR MESSAGE ALWAYS SHOW THE
CONTENTS OF THE PC, PS, SP, AND THE TEST NUMBER. MORE COLUMNS
OF DATA ARE ADDED WHERE THEY MIGHT BE RELAVENT TO A PARTICULAR
ERROR.

3.2 UNEXPECTED TRAP ERROR REPORTING

AN UNEXPECTED TRAP TO ADDRESS 4 CAUSES THE FOLLOWING
MESSAGE TO BE PRINTED OUT...

```
TRAPPED TO LOC 4 FROM LOCATION "XXXXXX"
RESTARTING PROGRAM
```

IN THE ACTUAL MESSAGE THE "XXXXXX" IS REPLACED BY THE PC
ADDRESS PUSHED ONTO THE STACK WHEN THE UNEXPECTED TRAP
OCCURS. THE PROGRAM THEN TRYS TO RESTART ITSELF
DESPITE SWITCH REGISTER SETTINGS.

3.3 POWER FAIL

IF A POWER FAIL CONDITION IS DETECTED THE FOLLOWING MESSAGE
IS PRINTED...

```
POWER
```

AFTER PRINTING OUT THE MESSAGE THE PROGRAM TRYS TO
RESTART ITSELF.

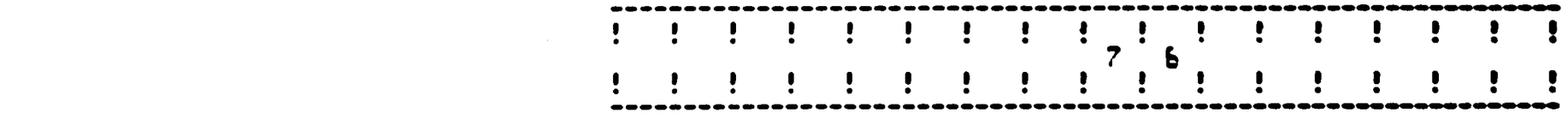
5.0 DEVICE INFORMATION

5.1 GENERAL INFORMATION

THE LINE CLOCK INTERRUPT VECTOR ADDRESS IS 100
THE LINE CLOCK PRIORITY LEVEL IS BR6

5.2 REGISTERS

LINE CLOCK STATUS REGISTER (LKS) 777546



BIT6 IF SET MONITOR=1 CAUSES AN INTERRUPT
BIT7 MONITOR BIT. SET BY CLOCK, CLEARED BY USER

7.0 FLOW CHARTS

```
                                        %
                                .ABS
                                .ENABL AMA
                                .LIST  ME
                                .NLIST  MC, MD, CND
167400                          $SWR=167400
000000                          $TN=0
                                        .ENABL  ABS
                                .MCALL  .HEADER,.$CATCH,.$EOP,.EQUAT,.SWRHI,.SWRLO,.$SCOPE,.SETUP
                                .MCALL  .$TYPOCT,.$TYPDEC,.$TRAP,.$POWER,.$ERROR,.$TYPE,STARS,.$ERRTYP,.$CMTAG
                                .MCALL  SETPRI,GETPRI,.$ACT11
                                .TITLE  MAINDEC-11-DZKWA-E  LINE FREQUENCY CLOCK PROGRAM
                                ;*COPYRIGHT (C) 1970,1972,1975,1976
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY J. COMEAU
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZGAC-B1),AUG 29,1975.
                                ;*

                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          --------------------
                                ;*        15            HALT ON ERROR
                                ;*        14            LOOP ON TEST
                                ;*        13            INHIBIT ERROR TYPEOUTS
                                ;*        11            INHIBIT ITERATIONS
                                ;*        10            BELL ON ERROR
                                ;*                      LOOP ON ERROR
                                ;*         8            LOOP ON TEST IN SWR<7:0>
                                ;*       7-0            #OF TEST TO LOOP ON IF SWR<9> IS SET

                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100
                                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                                .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
177776                          PS=     177776          ;;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                          DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                                ;*GENERAL PURPOSE 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                          R6=     %5              ;;GENERAL REGISTER
```

```
                        R7=     %7              ;;GENERAL REGISTER
                        .EQUIV  R6,SP           ;;STACK POINTER
                        .EQUIV  R7,PC           ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
000040                  PR1=    40              ;;PRIORITY LEVEL 1
000100                  PR2=    100             ;;PRIORITY LEVEL 2
000140                  PR3=    140             ;;PRIORITY LEVEL 3
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  PR7=    340             ;;PRIORITY LEVEL 7

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
000002                  SW01=   2
000001                  SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                  BIT15=  100000
040000                  BIT14=  40000
020000                  BIT13=  20000
010000                  BIT12=  10000
004000                  BIT11=  4000
002000                  BIT10=  2000
001000                  BIT09=  1000
000400                  BIT08=  400
000200                  BIT07=  200
000100                  BIT06=  100
000040                  BIT05=  40
000020                  BIT04=  20
000010                  BIT03=  10
```

```
                        BIT02=  4
                        BIT01=  2
                        BIT00=  1
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                  TBITVEC=14              ;;"T" BIT
000014                  TRTVEC= 14              ;;TRACE TRAP
000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                  PWRVEC= 24              ;;POWER FAIL
000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034                  TRAPVEC=34              ;;"TRAP" TRAP
000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
                        ;MISCELANHUS EQUATES
177546                  LKS=177546
000240                  NOP=240
000774                  BUF2=774
000776                  BUF1=776

                        .SBTTL  TRAP CATCHER

000000                          .=0
                        ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                        ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                        ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
000174                          .=174
000174  000000                  DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000                  SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER

                        .SBTTL  STARTING ADDRESS(ES)
000200  000137  001400          JMP     @#KSTART                ;;JUMP TO STARTING ADDRESS OF PROGRAM
                        ;*****************************************************************

                        .SBTTL  ACT11 HOOKS
                        ;HOOKS REQUIRED BY ACT11
000204                          $SVPC=.                 ;SAVE PC
000046                          .=46
000046  006166                  $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
000052                          .=52
000052  000000                  .WORD   0               ;;2)SET LOC.52 TO ZERO
000204                          .=$SVPC                 ;; RESTORE PC
```

```
                                ;*********************************************************

                                .SBTTL  COMMON TAGS

                                ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                ;*USED IN THE PROGRAM.

                      001100            .=1100
001100                          $CMTAG:                 ;;START OF COMMON TAGS
001100  000000          $PASS:  .WORD   0               ;;CONTAINS PASS COUNT
001102     000          $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103     000          $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114     000          $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115     001          $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                  .WORD   0
001134  000000                  .WORD   0
001136  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001140  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001142  177560          $TKS:   177560                  ;;TTY KBD STATUS
001144  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001146  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001150  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001152     000          $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001153     002          $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001154     012          $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001155     000          $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001156  000000          $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($REG0) WAS OBTAINED
001160  000000          $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001162  000000          $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001164  000000          $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001166  000000          $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001170  000000          $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001172  000000          $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001174  000000          $REG6:  .WORD   0               ;;CONTAINS (($REGAD)+14)
001176  000000          $REG7:  .WORD   0               ;;CONTAINS (($REGAD)+16)
001200  000000          $TMP0:  .WORD   0               ;;USER DEFINED
001202  000000          $TMP1:  .WORD   0               ;;USER DEFINED
001204  000000          $TMP2:  .WORD   0               ;;USER DEFINED
001206  000000          $TMP3:  .WORD   0               ;;USER DEFINED
001210  000000          $TMP4:  .WORD   0               ;;USER DEFINED
001212  000000          $TMP5:  .WORD   0               ;;USER DEFINED
001214  000000          $TMP6:  .WORD   0               ;;USER DEFINED
001216  000000          $TMP7:  .WORD   0               ;;USER DEFINED
001220  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001222  000000          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
```

```
   390  001224  177607  000377          $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
   391  001230     077                  $QUES:  .ASCII  /?/             ;;QUESTION MARK
   392  001231     015                  $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
   393  001232  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
```

```
                                ;*****************************************************************
001234  000000                  WORD:   000000

                                .SBTTL  ERROR POINTER TABLE

                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*      DF              ;;POINTS TO THE DATA FORMAT

001236                          $ERRTB:
001236  010424                          EM1     ;"                                      LKS     LKS     "
001240  010505                          DH1     ;"PC      PS      SP      TEST#   WAS     S/B     "
001242  010566                          DT1     ;$ERRPC,$REG7,$REG6,$REG5,LKS,$GDDAT
001244  000000                          0

001246  010604                          EM2     ;"CLOCK FAILED TO INTERRUPT"
001250  010636                          DH2     ;"PC      PS      SP      TEST#   LKS     "
001252  010710                          DT2     ;$ERRPC,$REG7,$REG6,$REG5,LKS
001254  000000                          0

001256  010724                          EM3     ;"CLOCK INTERRUPTED WHEN THE PROCESSOR PRIORITY WAS TOO HIGH"
001260  011017                          DH3     ;"PC      PS      SP      TEST#   LKS     "
001262  011070                          DT3     ;$ERRPC,$REG7,$REG6,$REG5,LKS
001264  000000                          0

001266  011104                          EM4     ;"CLOCK GIVES UNEQUAL # OF PULSES OVER TWO EQUAL PERIODS OF TIME"
001270  011203                          DH4     ;"PC      PS      SP      TEST#   1ST     2ND"
001272  011340                          DT4     ;$ERRPC,$REG7,$REG6,$REG5,$REG1,$REG0
001274  000000                          0

001276  011356                          EM5     ;"LKS REGISTER RESPONDS TO ANOTHER ADDRESS"
001300  011427                          DH5     ;"PC      PS      SP      TEST#   ADDRESS"
001302  011500                          DT5     ;$ERRPC,$REG7,$REG6,$REG5,$GDADR
001304  000000                          0

001306  011514                          EM6     ;"A NO SACK TIMEOUT HAS OCCURED"
001310  011552                          DH6     ;"PC      PS      SP      TEST#   LKS     "
001312  011624                          DT6     ;$ERRPC,$REG7,$REG6,$REG5,LKS
001314  000000                          0

001316  011640                          EM7     ;"WRONG CONDITION CODES WERE PUT ONTO STACK BY INTERRUPT"
001320  011727                          DH7     ;"PC      PS      SP      TEST#   CC      CC"
001322  012006                          DT7     ;$ERRPC,$REG7,$REG6,$REG5,BUF1,$GDDAT
001324  000000                          0

001326  012024                          EM10    ;"WRONG PC PUT ONTO THE STACK BY AN INTERRUPT"
001330  012100                          DH10    ;"PC      PS      SP      TEST#   "
001332  012164                          DT10    ;$ERRPC,$REG7,$REG6,$REG5,BUF2,$GDDAT
```

```
 450  001334  000000                          0
 451
 452  001336  012202                          EM11    ;"TRAPPED TRYING TO ACCESS LKS REGISTER"
 453  001340  012260                          DH11    ;"(PC)    (PS)    (SP)    TEST#"
 454  001342  012316                          DT11    ;$ERRPC,$REG7,$REG6,$REG5
 455  001344  000000                  0
 456
 457
 458                                  ;STARTUP CODE
 459          001400                  .=1400
 460  001400  012706  001000          KSTART: MOV     #1000,SP                ;INITIALIZE THE STACK SO WE CAN CALL THE TYPEOUT
 461  001404  005046                          CLR     -(SP)           ;;
 462  001406  013746  000034                  MOV     34,-(SP)        ;;SAVE CURRENT TRAP VECTOR
 463  001412  012737  001422  000034          MOV     #64$,34         ;;SETUP NEW TRAP VECTOT
 464  001420  104400                          TRAP                    ;;PUSH OLD PSW AN PCON STACK
 465  001422  016666  000002  000006  64$:    MOV     2(SP),6(SP)     ;;
 466  001430  012716  001436                  MOV     #65$,(SP)               ;;REPLACE OLD PC WITH NEW
 467  001434  000002                          RTI                     ;;RESTORE PSW
 468  001436  012637  000034          65$:    MOV     (SP)+,34                ;;RESTORE OLD TRAP  VECTOR
 469  001442  004737  006546                  JSR     PC,$TYPE                ;PRINTOUT STARTUP MESSAGE
 470  001446  010256                          STMES                   ;ADDRESS OF MESSAGE "MAINDEC-11-DZKWA-E"
 471  001450                          START:
 472  001450  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
 473  001454  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
 474  001456  022706  001126                  CMP     #$BDDAT,R6      ;;DONE?
 475  001462  001374                          BNE     .-6             ;;LOOP BACK IF NO
 476  001464  012706  001000                  MOV     #1000,SP                ;;SETUP THE STACK POINTER
 477  001470  012737  006274  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
 478  001476  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
 479  001504  012737  007570  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
 480  001512  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
 481  001520  012737  007774  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
 482  001526  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
 483  001534  012737  010030  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
 484  001542  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
 485  001550  013737  006134  006126          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
 486  001556  005037  001220                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
 487  001562  005037  001222                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
 488  001566  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
 489  001574  012737  001574  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
 490  001602  012737  001602  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
 491  001610  013746  000004                  MOV     @#4,-(SP)       ;;SAVE ERROR VECTOR
 492  001614  013746  000006                  MOV     @#6,-(SP)
 493  001620  012737  001634  000004          MOV     #64$,4          ;;SET UP TIME OUT VECTOR
 494  001626  005777  177304                  TST     @SWR            ;;TRY TO REFERENCE HARDWARE SWR
 495  001632  000407                          BR      65$             ;;BRANCH IF NO TIMEOUT TRAP OCCURS
 496  001634  012737  000176  001136  64$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
 497  001642  012737  000174  001140          MOV     #DISPREG,DISPLAY ;;POINT TO SOFTWARE DISPLAY REG
 498  001650  022626                          CMP     (SP)+,(SP)+     ;;RESTORE STACK
 499  001652  012637  000006          65$:    MOV     (SP)+,@#6       ;;RESTORE ERROR VECTOR
 500  001656  012637  000004                  MOV     (SP)+,@#4
 501  001662  005737  000042                  TST     42              ;LOADED BY A MONITOR
 502  001666  001401                          BEQ     T0001           ;BR IF NO
 503  001670  000005                          RESET                   ;YES--GENERATE AN INIT
 504
 505
```

```
                                        .SBTTL  TEST THAT THE LKS CAN BE REFERENCED WITHOUT A BUS ERROR
                                        ;LKS ACCESS TEST
001672  000004                  T0001:  SCOPE
001674  012737  001730  000004          MOV     #E0001,@#4      ;PREPARE FOR ADDRESSING THE LKS REGISTER. BAD HARDWARE
001702  012737  000340  000006          MOV     #340,@#6        ;COULD CAUSE A TRAP TO 4
001710  012737  001716  001106          MOV     #R0001,$LPADR   ;TIGHTEN UP THE SCOPE LOOP A BIT IN CASE OF AN ERROR
001716  012706  001000          R0001:  MOV     #1000,SP        ;SETUP THE STACK POINTER IN CASE OF AN ERROR
001722  005037  177546          I0001:  CLR     @#LKS           ;JUST REFERENCE LKS. DONT WORRY IF IT DIDNT CLEAR YET
001726  000401                          BR      T0002           ;WE DIDNT TRAP IF WE REACH HERE. GO ON TO NEXT TEST
001730  104011          E0001:  ERROR 11                        ;ERROR:::TRAPED TRYING TO ACCESS THE LKS REGISTER


                                        .SBTTL  TEST THAT START CLEARS LINE CLOCK INTERRUPT ENABLE BIT
                                        ;TEST THAT START CLEARS LINE CLOCK INTERRUPT ENABLE BIT
001732  000004                  T0002:  SCOPE
001734  012737  006222  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
001742  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
001750  012737  001756  001106          MOV     #R0002,$LPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
001756  000005          R0002:  RESET
001760  012737  000200  001124          MOV     #200,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
001766  032737  000100  177546          BIT     #100,LKS        ;TEST THE INTERRUPT ENABLE BIT
001774  001401                          BEQ     T0003
001776  104001          E0002:  ERROR 1                         ;ERROR, CLOCK INTERRUPT ENABLE NOT CLEARED BY INIT


                                        .SBTTL  TEST THAT START SETS CLOCK FLAG
                                        ;TEST THAT START SETS CLOCK FLAG
002000  000004                  T0003:  SCOPE
002002  012737  006222  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
002010  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
002016  012737  000200  001124          MOV     #200,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
002024  012737  002032  001106          MOV     #R0003,$LPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
002032  000005          R0003:  RESET                           ;SHOULD SET THE  CLOCK FLAG
002034  105737  177546                  TSTB    LKS             ;FIND OUT IF IT DID
002040  100401                          BMI     T0004           ;GO ON TO THE NEXT TEST IF IT SET THE CLOCK FLAG
002042  104001          E0003:  ERROR 1                         ;ERROR, CLOCK FLAG NOT SET BY INIT


                                        .SBTTL  TEST THAT CLOCK FLAG WILL SET AFTER SUFFICIENT PERIOD OF TIME (20 MS MIN)
                                        ;TEST THAT CLOCK FLAG WILL SET AFTER SUFFICIENT PERIOD OF TIME (20 MS MIN)
002044  000004                  T0004:  SCOPE
002046  012737  006222  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
002054  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
002062  012737  002070  001106          MOV     #R0004,$LPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
002070  012737  000200  001124  R0004:  MOV     #200,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
002076  005037  177546                  CLR     LKS             ;CLEAR THE CLOCK FLAG
002102  005000                          CLR     R0              ;AND A TIMER LOCATION
002104  105737  177546          A0004:  TSTB    LKS             ;IS CLOCK FLAG SET
002110  100403                          BMI     T0005
002112  005200                          INC     R0              ;NO, INCREMENT COUNT 003 WAIT FOR SOMEMORE
002114  001373                          BNE     A0004           ;WAIT SUFFICIENT AMOUNT OF TIME FOR CLOCK
002116  104001          E0004:  ERROR 1                         ;ERROR, CLOCK FLAG FAILED TO SET
```

```
                                        .SBTTL  TEST THAT INTERRUPT ENABLE BIT MAY BE SET
                                        ;TEST THAT INTERRUPT ENABLE BIT MAY BE SET
002120  000004                          T0005:  SCOPE
002122  012737  006222  000004                  MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
002130  012737  000340  000006                  MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
002136  012737  000100  001124                  MOV     #100,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
002144  012737  002164  001106                  MOV     #R0005,$LPADR   ;SETUP LOOP BACK ADDRESS IN CASE OF ERROR
002152  013746  000340                          MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
002156  012746  002164                          MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
002162  000002                                  RTI                     ;; POP NEW PC AND PS
002164                                  64$:
002164  005037  177546              R0005:  CLR     LKS
002170  005003                              CLR     R3              ;INITIALIZE A COUNTER LOCATION
002172  105737  177546              A0005:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
002176  100404                              BMI     B0005           ;IF SO, CONTINUE ON WITH THE TEST
002200  005203                              INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
002202  001373                              BNE     A0005           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
002204  104001                  E0005.  ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
002206  000410                              BR      T0006
002210                                  B0005:
002210  012737  000100  177546              MOV     #100,LKS        ;CLEAR CLOCK FLAG AND SET INTERRUPT ENABLE
002216  032737  000100  177546              BIT     #100,LKS        ;IS INTERRUPT ENABLE SET?
002224  001001                              BNE     T0006
002226  104001                  E1005:  ERROR 1                 ;ERROR INTERRUPT ENABLE NOT SET
```

```
                                        .SBTTL  TEST THAT INTERRUPT ENABLE BIT MAY BE CLEARED
                                        ;TEST THAT INTERRUPT ENABLE BIT MAY BE CLEARED
002230  000004                          T0006:  SCOPE
002232  012737  006222  000004                  MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
002240  012737  000340  000006                  MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
002246  012737  000000  001124                  MOV     #0,$GDDAT
002254  012737  002274  001106                  MOV     #R0006,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
002262  013746  000340                          MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
002266  012746  002274                          MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
002272  000002                                  RTI                     ;; POP NEW PC AND PS
002274                                  64$:
002274  005037  177546              R0006:  CLR     LKS
002300  005003                              CLR     R3              ;INITIALIZE A COUNTER LOCATION
002302  105737  177546              A0006:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
002306  100404                              BMI     B0006           ;IF SO, CONTINUE ON WITH THE TEST
002310  005203                              INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
002312  001373                              BNE     A0006           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
002314  104001                  E0006:  ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
002316  000412                              BR      T0007
002320                                  B0006:
002320  012737  000100  177546              MOV     #100,LKS        ;CLEAR CLOCK FLAG AND SET INTERRUPT ENABLE
002326  005037  177546                      CLR     LKS             ;CLEAR INTERRUPT ENABLE
002332  032737  000100  177546              BIT     #100,LKS        ;TEST THE INTERRUPT ENABLE BIT
002340  001401                              BEQ     T0007           ;IS INTERRUPT ENABLE CLEARED
002342  104001                  E10006: ERROR 1                 ;ERROR, ERROR INTERRUPT BIT CAN NOT BE CLEARED
```

```
                                        .SBTTL  TEST THAT CLOCK INTERRUPTS TO CORRECT VECTOR ADDRESS
                                        ;TEST THAT CLOCK INTERRUPTS TO CORRECT VECTOR ADDRESS
002344  000004                  T0007:  SCOPE
002346  012737  006222  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
002354  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
002362  012737  000300  001124          MOV     #300,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
002370  012737  002412  001106          MOV     #R0007,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
002376  012737  002474  000100          MOV     #D0007,100      ;SET UP VECTOR RETURN POINTER
002404  012737  000340  000102          MOV     #340,@#102      ;1 INTERRUPT IS ENOUGH
002412  012706  001000          R0007:  MOV     #1000,SP        ;GET STACK READY FOR INTERRUPTS
002416  005046                          CLR     -(SP)           ;DROP CPU PRIORITY LEVEL
002420  012746  002426                  MOV     #1$,-(SP)       ;SET RETURN ADDRESS ON STACK FROM 'RTI'
002424  000002                          RTI                     ;RETURN TO NEXT INSTRUCTION
002426  005037  177546          1$:     CLR     LKS
002432  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
002434  105737  177546          A0007:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
002440  100404                          BMI     B0007           ;IF SO, CONTINUE ON WITH THE TEST
002442  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
002444  001373                          BNE     A0007           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
002446  104001          E0007:  ERROR 1                         ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
002450  000415                          BR      T0010
002452                          B0007:
002452  012737  000100  177546          MOV     #100,LKS        ;ENABLE INTERRUPT
002460  005000                          CLR     R0
002462  005200          C0007:  INC     R0
002464  000240                          NOP                     ;STALL FOR TIME
002466  001375                          BNE     C0007           ;WAIT FOR INTERRUPT
002470  104002          E10007: ERROR 2                         ;ERROR, DIDNT GET INTERRUPT
002472  000404                          BR      T0010
002474  105737  177546          D0007:  TSTB    LKS             ;ENTER HERE IF INTERRUPTED
002500  100401                          BMI     T0010
002502  104001          E20007: ERROR 1                         ;ERROR, INTERRUPT NOT CAUSED BY CLOCK


                                        .SBTTL  TEST THAT CLOCK WILL INTERRUPT WITH PROCESSOR AT PRIORITY 5
                                        ;TEST THAT CLOCK WILL INTERRUPT WITH PROCESSOR AT PRICRITY 5
                                        ;NOTE: IF THERE IS NO HARDWARE 'PSW' THIS TEST WILL BE SKIPPED
002504  000004                  T0010:  SCOPE
002506  012737  006222  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
002514  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
002522  012737  002544  001106          MOV     #R0010,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
002530  012737  002654  000100          MOV     #D0010,100      ;SET UP VECTOR RETURN POINTER
002536  012737  000340  000102          MOV     #340,@#102      ;NO INTERRUPTS ALLOWED AFTER THE FIRST ONE
002544  005037  177546          R0010:  CLR     LKS
002550  013746  000004                  MOV     4,-(SP)         ;SAVE BUS TIMEOUT VECTOR CONTENTS
002554  012737  002572  000004          MOV     #1$,4           ;SET A SERVICE 'PC' IN CASE TIMEOUT OCCURS
002562  012737  000240  177776          MOV     #PR5,PS         ;DO WE HAVE A HARDWARE 'PSW'?
002570  000404                          BR      2$              ;BRANCH IF YES
002572  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE STACK FROM TIMEOUT
002574  012637  000004                  MOV     (SP)+,4         ;RESTORE BUS TIMEOUT VECTOR CONTENTS
002600  000431                          BR      T0011
002602  012637  000004          2$:     MOV     (SP)+,4         ;RESTORE BUS TIMEOUT VECTOR CONTENTS
002606  012706  001000                  MOV     #1000,SP        ;INITIALIZE THE STACK POINTER
```

```
002612  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
002614  105737  177546          A0010:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
002620  100404                          BMI     B0010           ;IF SO, CONTINUE ON WITH THE TEST
002622  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
002624  001373                          BNE     A0010           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
002626  104001          E0010:  ERROR 1                         ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
002630  000415                  BR      T0011
002632                  B0010:
002632  012737  000100  177546          MOV     #100.,LKS       ;ENABLE INTERRUPT
002640  005000                          CLR     R0
002642  005200          C0010:  INC     R0
002644  000240                          NOP                     ;STALL FOR SOME TIME
002646  001375                          BNE     C0010           ;WAIT FOR INTERRUPT
002650  104002          E1010:  ERROR 2                         ;ERROR, INTERRUPT FAILED TO OCCUP
002652  000404                  BR      T0011
002654  105737  177546          D0010:  TSTB    LKS             ;ENTER HERE IF INTERRUPTED
002660  100401                          BMI     T0011
002662  104001          E2010:  ERROR 1                         ;ERROR, INTERRUPTDID NOT CLEAR THE CLOCK FLAG


                        .SBTTL  TEST THAT CLOCK WILL NOT INTERRUPT WITH PROCESSOR PRIORITY 6
                        ;TEST THAT CLOCK WILL NOT INTERRUPT WITH PROCESSOR PRIORITY 6
                        ;NOTE: IF THERE IS NO HARDWARE 'PSW' THIS TEST WILL BE SKIPPED
002664  000004          T0011:  SCOPE
002666  012737  006222  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
002674  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
002702  012737  002710  001136          MOV     #R0011,$LPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
002710  005037  177546  R0011:  CLR     LKS
002714  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
002716  105737  177546          A0011:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
002722  100404                          BMI     B0011           ;IF SO, CONTINUE ON WITH THE TEST
002724  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
002726  001373                          BNE     A0011           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
002730  104001          E0011:  ERROR 1                         ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
002732  000443                  BR      T0012
002734                  B0011:
002734  012706  001000                  MOV     #1000,SP        ;INITIALIZE THE STACK POINTER
002740  013746  000004                  MOV     4,-(SP)         ;SAVE BUS TIMEOUT VECTOR CONTENTS
002744  012737  002762  000004          MOV     #1$,4           ;SET SERVICE 'PC' IN CASE TIMEOUT OCCURS
002752  012737  000300  177776          MOV     #PR6,PS         ;DO WE HAVE A HARDWARE 'PSW'?
002760  000404                          BR      2$              ;BRANCH IF YES
002762  022626          1$:     CMP     (SP)+,(SP)+             ;RESTORE STACK FROM BUS TIMEOUT
002764  012637  000004                  MOV     (SP)+,4         ;RESTORE BUS TIMEOUT VECTOR CONTENTS
002770  000424                          BR      T0012
002772  012637  000004          2$:     MOV     (SP)+,4         ;RESTORE BUS TIMEOUT VECTOR CONTENTS
002776  012737  003026  000100          MOV     #E1011,100      ;SET UP VECTOR RETURN
003004  012737  000100  177546          MOV     #100,LKS        ;ENABLE INTERRUPT
003012  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
003014  105737  177546          C0011:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
003020  100404                          BMI     D0011           ;IF SO, CONTINUE ON WITH THE TEST
003022  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
003024  001373                          BNE     C0011           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
003026  104001          E1011:  ERROR 1                         ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
003030  000715                  BR      T0011
003032                  D0011:
```

```
                                        NOP
                                        NOP                     ;GIVE CLOCK EXTRA TIME TO INTERRUPT
                                        BR      T0012
                                E20011: ERROR 3                 ;ERROR, CLOCK INTERRUPTED WITHOUT HAVING PRIORITY



                                .SBTTL  TEST THAT RESET SETS CLOCK FLAG
                                ;TEST THAT RESET SETS CLOCK FLAG
                                T0012:  SCOPE
003042  000004
003044  012737  000200  001124          MOV     #200,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPECUT IN CASE OF AN EPR
003052  012737  006222  000004          MOV     #TRAP0 @#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
003060  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
003066  012737  003074  001106          MOV     #R0012,$LPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
003074  005037  177546        R0012:    CLR     LKS             ;CLEAR CLOCK FLAG
003100  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
003102  105737  177546        A0012:    TSTB    LKS             ;IS THE CLOCK FLAG SET?
003106  100404                          BMI     B0012           ;IF SO, CONTINUE ON WITH THE TEST
003110  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
003112  001373                          BNE     A0012           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
003114  104001                E0012:    ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
003116  000407                          BR      T0013
003120                        B0012:
003120  005037  177546                  CLR     LKS
003124  000005                          RESET                   ;SHOULD SET CLOCK FLAG
003126  105737  177546                  TSTB    LKS
003132  100401                          BMI     T0013
003134  104001                E10012:   ERROR 1                 ;ERROR, RESET DIDN'T SET CLOCK FLAG



                                .SBTTL  TEST LINE CLOCK REPEATABILITY
                                ;TEST LINE CLOCK REPEATABILITY
                                ;MAKE SURE THAT OVER TWO EQUAL PERIODS OF TIME
                                ;THE CLOCK PUTS OUT THE SAME NUMBER OF PULSES
003136  000004                  T0013:  SCOPE
003140  005000        R0013:    CLR     R0              ;CLEAR 1ST TIME COUNT
003142  005001                  CLR     R1              ;CLEAR 1ST CLOCK COUNT
003144  013746  000340          MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
003150  012746  003156          MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
003154  000002                  RTI                     ;; POP NEW PC AND PS
003156                        64$:
003156  012737  003156  001106  P1013:   MOV     #R1013,$LPADR   ;ERROR IN NEXT  FEW INSTRUCTIONS CAUSES A SHORT SCOPE LO
003164  005037  177546                  CLR     LKS
                                                                ;SYNC ON CLOCK FLAG A COUPLE OF TIMES
003170  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
003172  105737  177546        A0013:    TSTB    LKS             ;IS THE CLOCK FLAG SET?
003176  100404                          BMI     B0013           ;IF SO, CONTINUE ON WITH THE TEST
003200  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
003202  001373                          BNE     A0013           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
003204  104001                E0013:    ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
003206  000510                          BR      T0014
003210                        B0013:
003210  012737  003210  001106  R2013:   MOV     #R2013,$LPADR   ;MAKE SCOPE LOOP SHORT IN CASE OF AN ERROR
003216  005037  177546                  CLR     LKS
003222  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
```

```
 796  003224  105737  177546          C0013:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
 797  003230  100404                          BMI     D0013           ;IF SO, CONTINUE ON WITH THE TEST
 798  003232  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
 799  003234  001373                          BNE     C0013           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
 800  003236  104001          E10013: ERROR 1                         ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
 801  003240  000473                          BR      T0014
 802  003242                  D0013:
 803  003242  005037  177546                  CLR     LKS
 804  003246  105737  177546          F0013:  TSTB    LKS             ;IS CLOCK FLAG SET
 805  003252  100003                          BPL     G0013           ;NO
 806  003254  005201                          INC     R1              ;+1 TO CLOCK COUNT
 807  003256  005037  177546                  CLR     LKS             ;CLEAR CLOCK IF SET
 808  003262  005200          G0013:  INC     R0              ;+1 TO TIME COUNT
 809  003264  001370                          BNE     F0013           ;REPEAT UNTIL R0=0
 810  003266  005000                          CLR     R0              ;CLEAR 2ND TIME COUNT
 811  003270  005002                          CLR     R2              ;CLEAR 2ND CLOCK COUNT
 812  003272  012737  003272  001106  R3013:  MOV     #R3013,$LPADR   ;INSURE A SHORT SCOPE LOOP
 813  003300  005037  177546                  CLR     LKS
 814                                                                  ;SYNC ON CLOCK FLAG TWICE
 815  003304  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
 816  003306  105737  177546          H0013:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
 817  003312  100404                          BMI     J0013           ;IF SO, CONTINUE ON WITH THE TEST
 818  003314  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
 819  003316  001373                          BNE     H0013           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
 820  003320  104001          E20013: ERROR 1                         ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
 821  003322  000442                          BR      T0014
 822  003324                  J0013:
 823  003324  012737  003324  001106  R4013:  MOV     #R4013,$LPADR   ;INSURE A SHORT SCOPE LOOP
 824  003332  005037  177546                  CLR     LKS
 825  003336  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
 826  003340  105737  177546          K0013:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
 827  003344  100404                          BMI     L0013           ;IF SO, CONTINUE ON WITH THE TEST
 828  003346  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
 829  003350  001373                          BNE     K0013           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
 830  003352  104001          E30013: ERROR 1                         ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
 831  003354  000425                          BR      T0014
 832  003356                  L0013:
 833  003356  012737  003140  001106          MOV     #R0013,$LPADR   ;MUST LOOP BACK TO BEGINING OF TEST IF EEROR COMES NOW
 834  003364  005037  177546                  CLR     LKS
 835  003370  105737  177546          M0013:  TSTB    LKS             ;IS CLOCK FLAG SET
 836  003374  100003                          BPL     N0013           ;NO
 837  003376  005202                          INC     R2              ;+1 TO CLOCK COUNT
 838  003400  005037  177546                  CLR     LKS             ;CLEAR CLOCK IF SET
 839  003404  005200          N0013:  INC     R0              ;+1 TO TIME COUNT
 840  003406  001370                          BNE     M0013           ;REPEAT UNTIL R0=0
 841  003410  020102                          CMP     R1,R2           ;IS 1ST CLOCK COUNT EQUAL TO 2ND CLOCK COUNT?
 842  003412  001406                          BEQ     T0014           ;YES
 843  003414  010137  001162          E40013: MOV     R1,$REG1        ;GET R1 READY FOR PRINTOUT
 844  003420  010237  001164                  MOV     R2,$REG2        ;GET R2 READY FOR PRINTOUT
 845  003424  104004                          ERROR 4                 ;ERROR, CLOCK FLAG OCCURRED DIFFERENT
 846  003426  000240                          NOP                     ;NUMBER OF TIMES IN EQUAL PERIODS
 847
 848
 849
 850                                  .SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
 851                                  ;LINE CLOCK REGISTER ADDRESSING TEST
```

```
                                        ;TEST THAT THE "LKS" REGISTER CAN NOT BE ADDRESSED AS ANYTHING BUT 177546
                                        ;SET A LOCATION THAT IS CLOSE(DIFFERS BY 1 BIT) TO THE LKS REGISTER
                                        ;TO 100. IF THE LKS ALSO CHANGES, THEN SIGNAL AN ERROR
003430  000004                  T0014:  SCOPE
003432  005037  001124                  CLR     $GDDAT
003436  012737  003500  001106          MOV     #R0014,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
003444  012737  157546  001120          MOV     #157546,$GDADR  ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 13 CLEAR
003452  012737  003516  000004          MOV     #A0014,4        ;SETUP VECTOR IN CASE IT IS NONEXISTANT
003460  012737  000340  000006          MOV     #340,6
003466  013746  000340                  MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
003472  012746  003500                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
003476  000002                          RTI                     ;; POP NEW PC AND PS
003500                          64$:
003500  012706  001000          R0014:  MOV     #1000,SP        ;SETUP THE STACK
003504  005037  177546                  CLR     LKS
003510  012777  000100  175402  I0014:  MOV     #100,@$GDADR    ;SET THE "CLOSE" ADDRESS TO = 100
003516  032737  000100  177546  A0014:  BIT     #100,LKS        ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
003524  001401                  E0014:  BEQ     B0014
003526  104005                          ERROR 5                 ;IT AFFECTED "LKS"  --  ERROR
003530  005037  177546          B0014:  CLR     LKS
003534  012737  003546  000004          MOV     #T0015  ,4
003542  005077  175352                  CLR     @$GDADR         ;CLEAR OUT THE "CLOSE" ADDRESS


                                        .SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
                                        ;LINE CLOCK REGISTER ADDRESSING TEST
003546  000004                  T0015:  SCOPE
003550  012737  003634  000004          MOV     #A0015,4        ;SETUP VECTOR IN CASE IT IS NONEXISTANT
003556  012737  000340  000006          MOV     #340,6
003564  005037  001124                  CLR     $GDDAT
003570  012737  003616  001106          MOV     #R0015,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
003576  012737  177146  001120          MOV     #177146,$GDADR  ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 9 CLEAR
003604  013746  000340                  MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
003610  012746  003616                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
003614  000002                          RTI                     ;; POP NEW PC AND PS
003616                          64$:
003616  012706  001000          R0015:  MOV     #1000,SP        ;SETUP THE STACK
003622  005037  177546                  CLR     LKS
003626  012777  000100  175264  I0015:  MOV     #100,@$GDADR    ;SET THE "CLOSE" ADDRESS TO = 100
003634  032737  000100  177546  A0015:  BIT     #100,LKS        ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
003642  001401                  E0015:  BEQ     B0015
003644  104005                          ERROR 5                 ;IT AFFECTED "LKS"  --  ERROR
003646  005037  177546          B0015:  CLR     LKS
003652  012737  003664  000004          MOV     #T0016  ,4
003660  005077  175234                  CLR     @$GDADR         ;CLEAR OUT THE "CLOSE" ADDRESS


                                        .SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
                                        ;LINE CLOCK REGISTER ADDRESSING TEST
003664  000004                  T0016:  SCOPE
003666  012737  003752  000004          MOV     #A0016,4        ;SETUP VECTOR IN CASE IT IS NONEXISTANT
003674  012737  000340  000006          MOV     #340,6
003702  005037  001124                  CLR     $GDDAT
003708  012737  003734  001106          MOV     #R0016,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
```

```
   898  003714  012737  177446  001120          MOV     #177446,$GDADR  ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 6 CLEAR
   899  003722  013746  000340                  MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
   900  003726  012746  003734                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
   901  003732  000002                          RTI                     ;; POP NEW PC AND PS
   902  003734                          64$:
   903  003734  012706  001000          R0016:  MOV     #1000,SP        ;SETUP THE STACK
   904  003740  005037  177546                  CLR     LKS
   905  003744  012777  000100  175146  I0016:  MOV     #100,@$GDADR    ;SET THE "CLOSE" ADDRESS TO = 100
   906  003752  032737  000100  177546  A0016:  BIT     #100,LKS        ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
   907  003760  001401                          BEQ     B0016
   908  003762  104005                  E0016:  ERROR 5                 ;IT AFFECTED "LKS"  --  ERROR
   909  003764  005037  177546          B0016:  CLR     LKS
   910  003770  012737  004002  000004          MOV     #T0017  ,4
   911  003776  005077  175116                  CLR     @$GDADR         ;CLEAR OUT THE "CLOSE" ADDRESS
   912
   913
   914
   915                                  .SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
   916                                  ;LINE CLOCK REGISTER ADDRESSING TEST
   917  004002  000004                  T0017:  SCOPE
   918  004004  012737  004070  000004          MOV     #A0017,4        ;SETUP VECTOR IN CASE IT IS NONEXISTANT
   919  004012  012737  000340  000006          MOV     #340,6
   920  004020  005037  001124                  CLR     $GDDAT
   921  004024  012737  004052  001106          MOV     #R0017,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
   922  004032  012737  177556  001120          MOV     #177556,$GDADR  ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 3 SET
   923  004040  013746  000340                  MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
   924  004044  012746  004052                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
   925  004050  000002                          RTI                     ;; POP NEW PC AND PS
   926  004052                          64$:
   927  004052  012706  001000          R0017:  MOV     #1000,SP        ;SETUP THE STACK
   928  004056  005037  177546                  CLR     LKS
   929  004062  012777  000100  175030  I0017:  MOV     #100,@$GDADR    ;SET THE "CLOSE" ADDRESS TO = 100
   930  004070  032737  000100  177546  A0017:  BIT     #100,LKS        ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
   931  004076  001401                          BEQ     B0017
   932  004100  104005                  E0017:  ERROR 5                 ;IT AFFECTED "LKS"  --  ERROR
   933  004102  005037  177546          B0017:  CLR     LKS
   934  004106  012737  004120  000004          MOV     #T0020  ,4
   935  004114  005077  175000                  CLR     @$GDADR         ;CLEAR OUT THE "CLOSE" ADDRESS
   936
   937
   938
   939                                  .SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
   940                                  ;LINE CLOCK REGISTER ADDRESSING TEST
   941  004120  000004                  T0020:  SCOPE
   942  004122  012737  004206  000004          MOV     #A0020,4        ;SETUP VECTOR IN CASE IT IS NONEXISTANT
   943  004130  012737  000340  000006          MOV     #340,6
   944  004136  005037  001124                  CLR     $GDDAT
   945  004142  012737  004170  001106          MOV     #R0020,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
   946  004150  012737  177566  001120          MOV     #177566,$GDADR  ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 4 SET
   947  004156  013746  000340                  MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
   948  004162  012746  004170                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
   949  004166  000002                          RTI                     ;; POP NEW PC AND PS
   950  004170                          64$:
   951  004170  012706  001000          R0020:  MOV     #1000,SP        ;SETUP THE STACK
   952  004174  005037  177546                  CLR     LKS
   953  004200  012777  000100  174712  I0020:  MOV     #100,@$GDADR    ;SET THE "CLOSE" ADDRESS TO = 100
```

```
                                ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
                                ;IT AFFECTED "LKS"  --  ERROR
                                ;CLEAR OUT THE "CLOSE" ADDRESS


                        .SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
                        ;LINE CLOCK REGISTER ADDRESSING TEST
                                ;SETUP VECTOR IN CASE IT IS NONEXISTANT
                                ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
                                ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 7 SET
                                ;;PUT NEW PS ON STACK
                                ;;PUT NEW PC ON  STACK
                                ;; POP NEW PC AND PS
                                ;SETUP THE STACK
                                ;SET THE "CLOSE" ADDRESS TO = 100
                                ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
                                ;IT AFFECTED "LKS"  --  ERROR
                                ;CLEAR OUT THE "CLOSE" ADDRESS


                        .SBTTL  LINE CLOCK REGISTER HIGH BYTE TEST
                        ;LINE CLOCK REGISTER HIGH BYTE TEST
                        ;MAKE SURE THE LKS REGISTER LOW BYTE RESPONDS TO THE HIGH BYTES ADDRESS
                                ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
                                ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
                                ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
                                ;HIGH BYTE OF THE LKS REGISTER
                                ;;PUT NEW PS ON STACK
                                ;;PUT NEW PC ON  STACK
                                ;; POP NEW PC AND PS
                                ;SETUP THE STACK
                                ;SET THE HIGH BYTE ADDRESS TO = 100
                                ;MAKE SURE THAT "LKS" LOW BYTE WAS AFFECTED
                                ;SHOULD HAVE SET BIT 7.  ERROR
```

```
1010
1011
1012                                        .SBTTL  CLOCK FLAG BIT TEST
1013                                        ;CLOCK FLAG BIT TEST
1014  004470  000004                       T0023:  SCOPE
1015  004472  012737  006222  000004               MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
1016  004500  012737  000340  000006               MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
1017  004506  012737  000200  001124               MOV     #200,$GDDAT
1018  004514  012737  004522  001106               MOV     #R0023,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
1019  004522  005037  177546            R0023:     CLR     LKS
1020  004526  005003                              CLR     R3              ;INITIALIZE A COUNTER LOCATION
1021  004530  105737  177546            A0023:     TSTB    LKS             ;IS THE CLOCK FLAG SET?
1022  004534  100404                              BMI     B0023           ;IF SO, CONTINUE ON WITH THE TEST
1023  004536  005203                              INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
1024  004540  001373                              BNE     A0023           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
1025  004542  104001            E0023:     ERROR 1                         ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
1026  004544  000410                              BR      T0024
1027  004546                    B0023:
1028  004546  012737  000200  177546     I0023:     MOV     #200,LKS        ;MOVE A 1 INTO THE CLOCK FLAG BIT
1029  004554  023737  177546  001124               CMP     LKS,$GDDAT      ;SHOULD NOT AFFECT THE FLAG BIT
1030  004562  001401                              BEQ     T0024
1031  004564  104001            E10023:    ERROR 1                         ;CLOCK FLAG DID NOT CLEAR
1032
1033
1034
1035                                        .SBTTL  INTERRUPT TEST
1036  004566  000004                       T0024:  SCOPE
1037  004570  012737  006222  000004               MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
1038  004576  012737  000340  000006               MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
1039  004604  012737  000200  001124               MOV     #200,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
1040  004612  012737  004632  001106               MOV     #R0024,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
1041  004620  012737  004672  000100               MOV     #E0024,100
1042  004626  005037  177546                      CLR     LKS             ;ALLOW CLOCK INTERRUPTS
1043  004632                    R0024:
1044  004632  013746  000340                      MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
1045  004636  012746  004644                      MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
1046  004642  000002                              RTI                     ;; POP NEW PC AND PS
1047  004644                    64$:
1048  004644  012737  000300  177546               MOV     #300,LKS
1049  004652  005227  000000            A0024:     INC     #0              ;WAIT FOR 20+ MS
1050  004656  001375                              BNE     A0024           ;LOOP BACK IF NOT DONE WAITING
1051  004660  000005                              RESET                   ;RESET SHOULD CLEAR INTERRUPT ENABLE
1052  004662  023737  001124  177546               CMP     $GDDAT,LKS      ;AND LEAVE THE CLOCK FLAG SET
1053  004670  001401                              BEQ     T0025           ;GO ON TO THE NEXT TEST IF IT DID
1054  004672  104001            E0024:     ERROR 1                         ;RESET SET INTERRUPT ENABLE OR CLEARED CLOCK FLAG
1055
1056
1057
1058                                        .SBTTL  NO SACK TIMEOUT TEST
1059                                        ;NO SACK TIMEOUT TEST
1060  004674  000004                       T0025:  SCOPE
1061  004676  012737  006222  000004               MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
1062  004704  012737  000340  000006               MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
1063  004712  012737  000300  001124               MOV     #300,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
1064  004720  012737  004742  001106               MOV     #R0025,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
1065  004726  012737  000340  000102               MOV     #340,102        ;NO INTERRUPTS AFTER THE FIRST ONE
```

```
1066  004734  012737  005024  000100          MOV     #C0025,100
1067  004742  005037  177546          R0025:   CLR     LKS
1068  004746  013746  000340                   MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
1069  004752  012746  004760                   MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
1070  004756  000002                           RTI                     ;; POP NEW PC AND PS
1071  004760                          64$:
1072  004760  005003                           CLR     R3              ;INITIALIZE A COUNTER LOCATION
1073  004762  105737  177546          A0025:   TSTB    LKS             ;IS THE CLOCK FLAG SET?
1074  004766  100404                           BMI     B0025           ;IF SO, CONTINUE ON WITH THE TEST
1075  004770  005203                           INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
1076  004772  001373                           BNE     A0025           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
1077  004774  104001          E0025:   ERROR 1                         ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS.
1078  004776  000417                           BR      T0026
1079  005000                          B0025:
1080  005000  005046                           CLR     -(SP)           ;DROP CPU PRIORITY LEVEL
1081  005002  012746  005010                   MOV     #1$,-(SP)       ;SET RETURN 'PC' FROM THE RTI
1082  005006  000002                           RTI                     ;RETURN TO NEXT INSTRUCTION
1083  005010  012737  000100  177546  1$:      MOV     #100,LKS        ;ENABLE CLOCK INTERRUPTS
1084  005016  000001                           WAIT                    ;THE ONLY WAY TO LEAVE HERE WITHOUT ERROR IS TO INTERRUP
1085  005020  104006          E1O025:  ERROR 6                         ;IF IT ERROR IS HERE ITS BECAUSE OF A "NO-SACK" TIMEOUT
1086  005022  000405                           BR      T0026
1087  005024  022737  000300  177546  C0025:   CMP     #300,LKS
1088  005032  001401                           BEQ     T0026           ;FIND OUT WHAT THE INTERRUPT DID TO THE CLOCK STATUS REG
1089  005034  104001          E2O025:  ERROR 1                         ;IT CLEARED THE INTERRUPT ENABLE OR THE FLAG BIT
1090
1091
1092
1093                                    .SBTTL  RESET TEST
1094                                    ;RESET TEST
1095  005036  000004          T0026:   SCOPE
1096  005040  012737  006222  000004           MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
1097  005046  012737  000340  000006           MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
1098  005054  012737  000200  001124           MOV     #200,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
1099  005062  012737  005116  001106           MOV     #R0026,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
1100  005070  013746  000340                   MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
1101  005074  012746  005102                   MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
1102  005100  000002                           RTI                     ;; POP NEW PC AND PS
1103  005102                          64$:
1104  005102  012737  005154  000100           MOV     #E0026,100
1105  005110  012737  000140  000102           MOV     #140,102        ;SETUP STATUS FOR AFTER THE INTERRUPT
1106  005116  005037  177546          R0026:   CLR     LKS
1107  005122  012706  001000                   MOV     #1000,SP        ;SETUP THE STACK
1108  005126  012737  000100  177546           MOV     #100,LKS        ;SET INTERRUPT ENABLE BIT NOW
1109  005134  005227  000000          A0026:   INC     #0              ;WAIT FOR CLOCK FLAG TO SET
1110  005140  001375                           BNE     A0026
1111  005142  000005          I0026:   RESET                           ;RESET SHOULD NOT CLEAR THE FLAG
1112  005144  023737  177546  001124           CMP     LKS,$GDDAT      ;FIND OUT IF ID DIT DID OR NOT
1113  005152  001401                           BEQ     T0027
1114  005154  104001          E0026:   ERROR 1                         ;RESET DID NOT INITIALIZE THE LKS WORD CORRECTLY
1115
1116
1117
1118                                    .SBTTL  CLOCK FLAG BIT TEST
1119                                    ;CLOCK FLAG BIT TEST
1120                                    ;MAKE SURE IT DOESNT CLEAR WHEN YOU TRY TO SET IT VIA A 'MOV' INSTRUCTION
1121  005156  000004          T0027:   SCOPE
```

```
1122  005160  012737  006222  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
1123  005166  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
1124  005174  012737  000200  001124          MOV     #200,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
1125  005202  012737  005222  001106          MOV     #R0027,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
1126  005210  005037  000102                  CLR     102
1127  005214  012737  005304  000100          MOV     #T0030  ,100
1128  005222  005037  177546          R0027:  CLR     LKS
1129  005226  013746  000340                  MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
1130  005232  012746  005240                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
1131  005236  000002                          RTI                     ;; POP NEW PC AND PS
1132  005240                          64$:
1133  005240  012706  001000                  MOV     #1000,SP        ;SETUP THE STACK
1134  005244  005037  001234                  CLR     WORD            ;SETUP A COUNTER LOCATION TO = 0
1135  005250  005237  001234          A0027:  INC     WORD
1136  005254  001375                          BNE     A0027           ;WASTE TIME LOOPING UNTIL THE COUNTER REACHES 0
1137  005256  012737  000300  177546          MOV     #300,LKS        ;SET INTERRUPT ENABLE AND TRY TO SET THE CLOCK FLAG
1138  005264  012737  000200  177546          MOV     #200,LKS        ;TRY TO SET IT AGAIN
1139  005272  023737  177546  001124          CMP     LKS,$GDDAT      ;DID THE CLOCK FLAG STAY SET?
1140  005300  001401                          BEQ     T0030           ;IF NOT GO ON TO THE NEXT TEST
1141  005302  104001                  E0027:  ERROR 1                 ;ERROR - MOVED A '1' INTO THE CLOCK FLAG BIT AND IT STAY
1142
1143
1144
1145                                  .SBTTL  CLOCK FLAG AFTER INTERRUPT TEST
1146                                  ;SEE IF AN INTERRUPT CLEARS THE CLOCK FLAG
1147  005304  000004                  T0030:  SCOPE
1148  005306  012737  006222  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
1149  005314  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
1150  005322  005037  177546                  CLR     LKS             ;NO CLOCK INTERRUPTS BEFORE WE ARE READY
1151  005326  012737  000100  001124          MOV     #100,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
1152  005334  012737  005342  001106          MOV     #R0030,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
1153  005342  012737  005410  000100  R0030:  MOV     #A0030,100      ;SETUP CLOCK INTERRUPT VECTOR
1154  005350  005037  000102                  CLR     102             ;PRIORITY LEVEL WILL ALLOW FURTHER INTERRUPTS
1155  005354  005046                          CLR     -(SP)           ;DROP CPU PRIORITY LEVEL
1156  005356  012746  005364                  MOV     #1$,-(SP)       ;SET RETURN 'PC' FROM THE RTI
1157  005362  000002                          RTI                     ;RETURN TO NEXT INSTRUCTION
1158  005364  012706  001000          1$:     MOV     #1000,SP        ;SETUP THE STACK
1159  005370  005037  177546                  CLR     LKS
1160  005374  105037  001234                  CLRB    WORD            ;CLEAR OUT A COUNTER LOCATION
1161  005400  012737  000100  177546          MOV     #100,LKS        ;ENABLE CLOCK INTERRUPTS NOW
1162  005406  000001                          WAIT                    ;WAIT FOR AN INTERRUPT
1163  005410  012737  005442  000100  A0030:  MOV     #E0030,100      ;ERROR IF WE INTERRUPT AGAIN
1164  005416  005046                          CLR     -(SP)           ;DROP CPU PRIORITY LEVEL
1165  005420  012746  005426                  MOV     #B0030,-(SP)    ;SET RETURN 'PC' FROM THE RTI
1166  005424  000002                          RTI                     ;RETURN TO NEXT INSTRUCTION
1167  005426  105237  001234          B0030:  INCB    WORD            ;DO A NOTHING LOOP FOR A VERY SHORT PERIOD OF TIME
1168  005432  001375                          BNE     B0030           ;WE SHOULD INCREMENT TO 0 LONG BEFORE AN INTERRUPT COMES
1169  005434  005037  177546                  CLR     LKS
1170  005440  000401                          BR      T0031
1171  005442  104001                  E0030:  ERROR 1                 ;INTERRUPT DID NOT CLEAR THE CLOCK FLAG
1172
1173
1174
1175                                  .SBTTL  NO INTERRUT AT PRIORITY 7 TEST
1176                                  ;TEST THAT CLOCK WILL NOT INTERRUPT WITH PROCESSR AT PRIORITY 7
1177  005444  000004                  T0031:  SCOPE
```

```
1178  005446  012737  006222  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
1179  005454  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
1180  005462  012737  005470  001106          MOV     #R0031,$LPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
1181  005470  005037  177546          R0031:  CLR     LKS
1182  005474  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
1183  005476  105737  177546          A0031:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
1184  005502  100404                          BMI     B0031           ;IF SO, CONTINUE ON WITH THE TEST
1185  005504  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
1186  005506  001373                          BNE     A0031           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
1187  005510  104001          E0031:  ERROR 1                         ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
1188  005512  000431                          BR      T0032
1189  005514                          B0031:
1190  005514  012706  001000                  MOV     #1000,SP        ;INITIALIZE THE STACK POINTER
1191  005520  013746  000340                  MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
1192  005524  012746  005532                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
1193  005530  000002                          RTI                     ;; POP NEW PC AND PS
1194  005532                          64$:
1195  005532  012737  005510  000100          MOV     #E0031,100      ;SET UP VECTOR RETURN
1196  005540  012737  000100  177546          MOV     #100,LKS        ;ENABLE INTERRUPT
1197  005546  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
1198  005550  105737  177546          C0031:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
1199  005554  100404                          BMI     D0031           ;IF SO, CONTINUE ON WITH THE TEST
1200  005556  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
1201  005560  001373                          BNE     C0031           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
1202  005562  104001          E10031: ERROR 1                         ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
1203  005564  000404                          BR      T0032
1204  005566                          D0031:
1205  005566  000240                          NOP
1206  005570  000240                          NOP                     ;GIVE CLOCK EXTRA TIME TO INTERRUPT
1207  005572  000401                          BR      T0032
1208  005574  104003          E20031: ERROR 3                         ;ERROR, CLOCK INTERRUPTED WITHOUT HAVING PRIORITY
1209
1210
1211
1212                                    .SBTTL  CC PUSH TEST FOR CLOCK INTERRUPTS
1213                                    ;TEST THAT CLOCK INTERRUPT PUSHES CONDITION CODES ONTO STACK
1214  005576  000004          T0032:  SCOPE
1215  005600  012737  006222  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
1216  005606  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
1217  005614  012737  005622  001106          MOV     #R0032,$LPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
1218  005622  005037  177546          R0032:  CLR     LKS
1219  005626  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
1220  005630  105737  177546          A0032:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
1221  005634  100404                          BMI     B0032           ;IF SO, CONTINUE ON WITH THE TEST
1222  005636  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
1223  005640  001373                          BNE     A0032           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
1224  005642  104001          E0032:  ERROR 1                         ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
1225  005644  000433                          BR      T0033
1226  005646                          B0032:
1227  005646  012706  001000                  MOV     #1000,SP        ;INITIALIZE THE STACK POINTER
1228  005652  005037  000776                  CLR     BUF1
1229  005656  005037  000774                  CLR     BUF2
1230  005662  012737  005712  000100          MOV     #C0032,100      ;SET UP VECTOR RETURN
1231  005670  012737  000100  177546          MOV     #100,LKS        ;ENABLE INTERRUPT
1232  005676  005046                          CLR     -(SP)           ;DROP CPU PRIORITY LEVEL
1233  005700  012746  005706                  MOV     #1$,-(SP)       ;SET RETURN 'PC' FROM THE RTI
```

```
1234  005704  000002                          RTI                     ;RETURN TO NEXT INSTRUCTION
1235  005706  000277                  1$:     +SEC!SEV!SEZ!SEN        ;SET ALL CONDITION CODES
1236  005710  000001                          WAIT                    ;WAIT FOR INTERRUPT
1237  005712  022737  000017  000776  C0032:  CMP     #17,BUF1
1238  005720  001405                          BEQ     T0033
1239  005722  012737  000017  001124          MOV     #17,$GDDAT
1240  005730  104007                          ERROR 7                 ;ERROR DID NOT PUSH CORRECT PS ONTO STACK
1241  005732  000721                          BR      T0032
1242
1243
1244
1245                                  .SBTTL  PC PUSH TEST FOR CLOCK INTERRUPTS
1246                                  ;TEST THAT CLOCK INTERRUPT PUSHES THE PROGRAM COUNTER ONTO STACK
1247  005734  000004                  T0033:  SCOPE
1248  005736  012737  006222  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
1249  005744  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
1250  005752  012737  005760  001106          MOV     #R0033,$LPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
1251  005760  005037  177546          R0033:  CLR     LKS
1252  005764  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
1253  005766  105737  177546          A0033:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
1254  005772  100404                          BMI     B0033           ;IF SO, CONTINUE ON WITH THE TEST
1255  005774  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
1256  005776  001373                          BNE     A0033           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
1257  006000  104001          E0033:  ERROR 1                         ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
1258  006002  000432                          BR      T0034
1259  006004                          B0033:
1260  005004  012706  001000                  MOV     #1000,SP        ;INITIALIZE THE STACK POINTER
1261  006010  005037  000776                  CLR     BUF1
1262  006014  005037  000774                  CLR     BUF2
1263  006020  012737  006050  000100          MOV     #C0033,100      ;SET UP VECTOR RETURN
1264  006026  012737  000100  177546          MOV     #100,LKS        ;ENABLE INTERRUPT
1265  006034  005046                          CLR     -(SP)           ;DROP CPU PRIORITY LEVEL
1266  006036  012746  006044                  MOV     #1$,-(SP)       ;SET RETURN 'PC' FROM THE RTI
1267  006042  000002                          RTI                     ;RETURN TO NEXT INSTRUCTION
1268  006044  000277                  1$:     +SEC!SEV!SEZ!SEN        ;SET ALL CONDITION CODES
1269  006046  000001                          WAIT                    ;WAIT FOR INTERRUPT
1270  006050  022737  006050  000774  C0033:  CMP     #C0033,BUF2
1271  006056  001404                          BEQ     T0034
1272  006060  012737  006050  001124  E10033: MOV     #C0033,$GDDAT
1273  006066  104010                          ERROR 10                ;ERROR, DID NOT PUSH CORRECT PC ONTO STACK
1274
1275
1276
1277                                  .SBTTL  END OF PASS INDICATING
1278  006070  000004                  T0034:  SCOPE
1279  006072  005037  177546                  CLR     LKS             ;TURN THE CLOCK OFF
1280  006076  000005                          RESET                   ;TURN EVERYTHING OFF
1281
1282                                  ;*****************************************************************
1283
1284                                  .SBTTL  END OF PASS ROUTINE
1285
1286                                  ;*INCREMENT THE PASS NUMBER ($PASS)
1287                                  ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
1288                                  ;*IF THERES A MONITOR GO TO IT
1289                                  ;*IF THERE ISN'T JUMP TO T0001
```

```
1290
1291  006100                                  $EOP:
1292  006100  000004                                 SCOPE
1293  006102  005037  001102                         CLR     $TSTNM          ;;ZERO THE TEST NUMBER
1294  006106  005037  001220                         CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
1295  006112  005237  001100                         INC     $PASS           ;;INCREMENT THE PASS NUMBER
1296  006116  042737  100000  001100                 BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
1297  006124  005327                                 DEC     (PC)+           ;;LOOP?
1298  006126  000001                          $EOPCT: .WORD   1
1299  006130  003022                                 BGT     $DOAGN          ;;YES
1300  006132  012737                                 MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
1301  006134  000001                          $ENDCT: .WORD   1
1302  006136  006126                                 $EOPCT
1303  006140  104400  006202                         TYPE    ,$ENDMG         ;;TYPE "END PASS #"
1304  006144  013746  001100                         MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
1305  006150  104404                                 TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
1306  006152  104400  006217                         TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
1307  006156                                  $GET42:
1308
1309  006156  013700  000042                         MOV     @#42,R0         ;;GET MONITOR ADDRESS
1310  006162  001405                                 BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
1311  006164  000005                                 RESET                   ;;CLEAR THE WORLD
1312  006166  004710                          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
1313  006170  000240                                 NOP                     ;;SAVE ROOM
1314  006172  000240                                 NOP                     ;;FOR
1315  006174  000240                                 NOP                     ;;ACT11
1316  006176                                  $DOAGN:
1317  006176  000137  001672                         JMP     @#T0001         ;;RETURN
1318  006202  005015  047105  020104  $ENDMG: .ASCIZ  <15><12>/END PASS #/
1319  006210  040520  051523  021440
1320  006216     000
1321  006217     377     377     000  $ENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
1322  006222  005046                          TRAP0:  CLR     -(SP)
1323  006224  004737  006546                         JSR     PC,$TYPE        ;PRINTOUT"TRAPPED TO 4 FROM "
1324  006230  010333                                 TRPMES                  ;ADDRESS OF THE MESSAGE
1325  006232  011646                                 MOV     (SP),-(SP)      ;GET THE ADDRESS WHERE THE TRAP OCCURED
1326  006234  162716  000002                         SUB     #2,(SP)         ;MAKE IT RIGHT
1327  006240  104401                                 TYPOC                   ;TYPE OUT ADDRESS IN OCTAL
1328  006242  104400  001231                         TYPE    ,$CRLF          ;PRINTOUT A CARRIAGE RETURN-LINE FEED
1329  006246  005046                                 CLR     -(SP)
1330  006250  004737  006546                         JSR     PC,$TYPE        ;PRINTOUT RESTARTING MESSAGE
1331  006254  010374                                 TRPM2S                  ;ADDRESS OF RESTART MESSAGE
1332  006256  000240                                 NOP
1333  006260  000240                                 NOP
1334  006262  000240                                 NOP
1335  006264  000240                                 NOP
1336  006266  000005                                 RESET
1337  006270  000137  001450                         JMP     START
1338                                  ;*****************************************************************
1339
1340                                  .SBTTL  SCOPE HANDLER ROUTINE
1341
1342                                  ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
1343                                  ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
1344                                  ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
1345                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
```

```
1346                                        :*SW14=1        LOOP ON TEST
1347                                        :*SW11=1        INHIBIT ITERATIONS
1348                                        :*SW09=1        LOOP ON ERROR
1349                                        :*SW08=1        LOOP ON TEST IN SWR<7:0>
1350                                        :*CALL
1351                                        :*      SCOPE           ;;SCOPE=IOT
1352
1353  006274                                $SCOPE:
1354  006274  000240                               NOP
1355  006276  032777  040000  172632  1$:   BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
1356  006304  001111                               BNE     $OVER           ;;YES IF SW14=1
1357                                        ;*****START OF CODE FOR THE XOR TESTER*****
1358  006306  000416                        $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
1359                                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
1360  006310  013746  000004                       MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
1361  006314  012737  006334  000004               MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
1362  006322  005737  177060                       TST     @#177060        ;;TIME OUT ON > 2?
1363  006326  012637  000004                       MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
1364  006332  000463                               BR      $SVLAD          ;;GO TO THE NEXT TEST
1365  006334  022626                        5$:    CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
1366  006336  012637  000004                       MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
1367  006342  000423                               BR      7$              ;;LOOP ON THE PRESENT TEST
1368  006344                                6$:;*****END OF CODE FOR THE XOR TESTER*****
1369  006344  032777  000400  172564               BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
1370  006352  001404                               BEQ     2$              ;;BR IF NO
1371  006354  127737  172556  001102               CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
1372  006362  001462                               BEQ     $OVER           ;;BR IF YES
1373  006364  105737  001103                2$:    TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
1374  006370  001421                               BEQ     3$              ;;BR IF NO
1375  006372  123737  001115  001103               CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
1376  006400  101015                               BHI     3$              ;;BR IF NO
1377  006402  032777  001000  172526               BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
1378  006410  001404                               BEQ     4$              ;;BR IF NO
1379  006412  013737  001110  001106  7$:    MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
1380  006420  000443                               BR      $OVER
1381  006422  105037  001103                4$:    CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
1382  006426  005037  001220                       CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
1383  006432  000415                               BR      1$              ;;ESCAPE TO THE NEXT TEST
1384  006434  032777  004000  172474  3$:    BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
1385  006442  001011                               BNE     1$              ;;BR IF YES
1386  006444  005737  001100                       TST     $PASS           ;;IF FIRST PASS OF PROGRAM
1387  006450  001406                               BEQ     1$              ;;      INHIBIT ITERATIONS
1388  006452  005237  001104                       INC     $ICNT           ;;INCREMENT ITERATION COUNT
1389  006456  023737  001220  001104               CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
1390  006464  002021                               BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
1391  006466  012737  000001  001104  1$:    MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
1392  006474  013737  006544  001220               MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
1393  006502  105237  001102                $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
1394  006506  011637  001106                       MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
1395  006512  011637  001110                       MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
1396  006516  005037  001222                       CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
1397  006522  112737  000001  001115               MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
1398  006530  013777  001102  172402  $OVER: MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
1399  006536  013716  001106                       MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
1400  006542  000002                               RTI                     ;;FIXES PS
1401  006544  000010                        $MXCNT: 10                      ;;MAX. NUMBER OF ITERATIONS
```

```
                                ;*****************************************************************

                                .SBTTL  TYPE ROUTINE

                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                ;*
                                ;*CALL:
                                ;*1) USING A TRAP INSTRUCTION
                                ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                ;*OR
                                ;*      TYPE
                                ;*      MESADR
                                ;*

006546  105737  001155          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
006552  100002                          BPL     1$              ;;BR IF YES
006554  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
006556  000407                          BR      3$              ;;LEAVE
006560  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
006562  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
006566  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
006570  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
006572  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
006574  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
006576  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
006602  000002                          RTI                     ;;RETURN
006604  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
006610  001426                          BEQ     8$
006612  122716  000200                  CMPB    #^CRLF,(SP)     ;;BRANCH IF NOT <CRLF
006616  001004                          BNE     5$
006620  005726                          TST     (SP)+           ;;POP <CR><LF EQUIV
006622  104400                          TYPE                    ;;TYPE A CR AND LF
006624  001233                          $CRLF
006626  000757                          BR      2$              ;;GET NEXT CHARACTER
006630  004737  006712          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
006634  123726  001154          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
006640  001352                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
006642  013746  001152                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
006646  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
006652  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
006654  004737  006                     JSR     PC,$TYPEC       ;;GO TYPE A NULL
006660  105337  006                     DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
006664  000770                          BR      7$              ;;LOOP

                                ;HORIZONTAL TAB PROCESSOR

006666  112716  000040          8$:     MOVB    #40,(SP)        ;;REPLACE TAB WITH SPACE
006672  004737  006712          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
006676  132737  000007  006756          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
006704  001372                          BNE     9$              ;;TAB STOP
006706  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
```

```
                                        BR      2$              ;;GET NEXT CHARACTER
                        172230  $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
                                        BPL     $TYPEC
                        172232          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
                000002                  CMPB    #15,2(SP)       ;;BRANCH IF
                000015  000002          BNE     1$              ;;NOT <CR>
                                        CLRB    $CHARCNT        ;;
                006756                  BR      $TYPEX          ;;EXIT
                000406
                000012  000002  1$:     CMPB    #12,2(SP)       ;;BRANCH IF
                                        BGE     $TYPEX          ;;<LF>
                                        INCB    (PC)+           ;;INC SPACE
                000000          $CHARCNT:.WORD  0               ;;COUNT
                000207          $TYPEX: RTS     PC
                                ;;      EQUATES
                000011                  TAT=11
                000200                  TCRLF=200

                                ;*******************************************************************

                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                                ;*REPLACED WITH SPACES.
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                                ;*      TYPDS                   ;;GO TO THE ROUTINE

006762                          $TYPDS:
006762  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
006764  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
006766  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
006770  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
006772  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
006774  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
007000  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
007004  100004                          BPL     1$              ;;BR IF INPUT IS POS.
007006  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
007010  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
007016  005000          1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
007020  012703  007176                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
007024  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
007030  005002          2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
007032  016001  007166                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
007036  160105          3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
007040  002402                  BLT     4$              ;;BR IF DONE
007042  005202                  INC     R2              ;;INCREASE THE BCD DIGIT BY 1
007044  000774                  BR      3$
007046  060105          4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
007050  005702                  TST     R2              ;;CHECK IF BCD DIGIT=0
007052  001002                  BNE     5$              ;;FALL THROUGH IF 0
007054  105716                  TSTB    (SP)            ;;STILL DOING LEADING 0'S
007056  100407                  BMI     7$              ;;BR IF YES
007060  106316          5$:     ASLB    (SP)            ;;MSD
```

```
 1514  007062  103003                        BCC     6$              ;;BR IF NO
 1515  007064  116663  000001  177777        MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
 1516  007072  052702  000060          6$:   BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
 1517  007076  052702  000040          7$:   BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
 1518  007102  110233                        MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
 1519  007104  005720                        TST     (R0)+           ;;JUST INCREMENTING
 1520  007106  020027  000010                CMP     R0,#10          ;;CHECK THE TABLE INDEX
 1521  007112  002746                        BLT     2$              ;;GO DO THE NEXT DIGIT
 1522  007114  003002                        BGT     8$              ;;GO TO EXIT
 1523  007116  010502                        MOV     R5,R2           ;;GET THE LSD
 1524  007120  000764                        BR      6$              ;;GO CHANGE TO ASCII
 1525  007122  105726                  8$:   TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
 1526  007124  100003                        BPL     9$              ;;BR IF NO
 1527  007126  116663  177777  177776        MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
 1528  007134  105013                  9$:   CLRB    (R3)            ;;SET THE TERMINATOR
 1529  007136  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
 1530  007140  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
 1531  007142  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
 1532  007144  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 1533  007146  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 1534  007150  104400  007176                TYPE    $DBLK           ;;NOW TYPE THE NUMBER
 1535  007154  016666  000002  000004        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 1536  007162  012616                        MOV     (SP)+,(SP)
 1537  007164  000002                        RTI                     ;;RETURN TO USER
 1538  007166  023420                  $DTB:: 10000.
 1539  007170  001750                        1000.
 1540  007172  000144                        100.
 1541  007174  000012                        10.
 1542  007176  000004                  $DBLK:  .BLKW   4
 1543                                  ;*****************************************************************
 1544
 1545                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
 1546
 1547                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
 1548                                  ;*OCTAL (ASCII) NUMBER AND TYPE IT.
 1549                                  ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
 1550                                  ;*CALL:
 1551                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 1552                                  ;*      TYPOS                   ;;CALL FOR TYPEOUT
 1553                                  ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
 1554                                  ;*      .BYTE   M               ;;M=1 OR 0
 1555                                  ;*                                      ;;1=TYPE LEADING ZEROS
 1556                                  ;*                                      ;;0=SUPPRESS LEADING ZEROS
 1557                                  ;*
 1558                                  ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
 1559                                  ;*$TYPOS OR $TYPOC
 1560                                  ;*CALL:
 1561                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 1562                                  ;*      TYPON                   ;;CALL FOR TYPEOUT
 1563                                  ;*
 1564                                  ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
 1565                                  ;*CALL:
 1566                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 1567                                  ;*      TYPOC                   ;;CALL FOR TYPEOUT
 1568
 1569  007200  017646  000000          $TYPOS: MOV     @(SP),-(SP      ;;PICKUP THE MODE
```

```
1570  007212  116637  000001  007431          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
1571  007220  112637  007433                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
1572  007224  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
1573  007230  000406                          BR      $TYPON
1574  007232  112737  000001  007431  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
1575  007240  112737  000006  007433          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
1576  007246  112737  000005  007430  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
1577  007254  010346                          MOV     R3,-(SP)        ;;SAVE R3
1578  007256  010446                          MOV     R4,-(SP)        ;;SAVE R4
1579  007260  010546                          MOV     R5,-(SP)        ;;SAVE R5
1580  007262  113704  007433                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
1581  007266  005404                          NEG     R4
1582  007270  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
1583  007274  110437  007432                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
1584  007300  113704  007431                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
1585  007304  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
1586  007310  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
1587  007312  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
1588  007314  000404                          BR      3$              ;;GO DO MSB
1589  007316  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
1590  007320  006105                          ROL     R5
1591  007322  006105                          ROL     R5
1592  007324  010503                          MOV     R5,R3
1593  007326  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
1594  007330  105337  007432                  DECB    $OMODE          ;;TYPE THIS DIGIT?
1595  007334  100016                          BPL     7$              ;;BR IF NO
1596  007336  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
1597  007342  001002                          BNE     4$              ;;TEST FOR 0
1598  007344  005704                          TST     R4              ;;SUPPRESS THIS 0?
1599  007346  001403                          BEQ     5$              ;;BR IF YES
1600  007350  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
1601  007352  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
1602  007356  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
1603  007362  110337  007426                  MOVB    R3,8$           ;;SAVE FOR TYPING
1604  007366  104400  007426                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
1605  007372  105337  007430          7$:     DECB    $OCNT           ;;COUNT BY 1
1606  007376  003347                          BGT     2$              ;;BR IF MORE TO DO
1607  007400  002402                          BLT     6$              ;;BR IF DONE
1608  007402  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
1609  007404  000744                          BR      2$              ;;GO DO THE LAST DIGIT
1610  007406  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
1611  007410  012604                          MOV     (SP)+,R4        ;;RESTORE R4
1612  007412  012603                          MOV     (SP)+,R3        ;;RESTORE R3
1613  007414  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
1614  007422  012616                          MOV     (SP)+,(SP)
1615  007424  000002                          RTI                     ;;RETURN
1616  007426  000                     8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
1617  007427  000                             .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
1618  007430  000                     $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
1619  007431  000                     $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
1620  007432  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
1621                                  ;******************************************************************
1622
1623                                  .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
1624
1625                                  ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
```

```
                                        ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
                                        ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.

007434                                  $ERRTYP:
007434  104400  001231                        TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
007440  010046                                MOV     R0,-(SP)        ;;SAVE R0
007442  005000                                CLR     R0              ;;PICKUP THE ITEM INDEX
007444  153700  001114                        BISB    @#$ITEMB,R0
007450  001004                                BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                                                        ;;TYPE THE PC OF THE ERROR
007452  013746  001116                        MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                                        ;;ERROR ADDRESS
007456  104401                                TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
007460  000426                                BR      6$              ;;GET OUT
007462  005300                          1$:   DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
007464  006300                                ASL     R0              ;;      WORK FOR THE ERROR TABLE
007466  006300                                ASL     R0
007470  006300                                ASL     R0
007472  062700  001236                        ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
007476  012037  007506                        MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
007502  001404                                BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
007504  104400                                TYPE                    ;;TYPE THE "ERROR MESSAGE"
007506  000000                          2$:   .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
007510  104400  001231                        TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
007514  012037  007524                  3$:   MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
007520  001404                                BEQ     5$              ;;SKIP TYPEOUT IF 0
007522  104400                                TYPE                    ;;TYPE THE "DATA HEADER"
007524  000000                          4$:   .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
007526  104400  001231                        TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
007532  011000                          5$:   MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
007534  001004                                BNE     7$              ;;GO TYPE THE DATA
007536  012600                          6$:   MOV     (SP)+,R0        ;;RESTORE R0
007540  104400  001231                        TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
007544  000207                                RTS     PC              ;;RETURN
007546                                  7$:
007546  013046                                MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
007550  104401                                TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
007552  005710                                TST     (R0)            ;;IS THERE ANOTHER NUMBER?
007554  001770                                BEQ     6$              ;;BR IF NO
007556  104400  007564                        TYPE    ,8$             ;;TYPE TWO(2) SPACES
007562  000771                                BR      7$              ;;LOOP
007564  020040      000               8$:   .ASCIZ    /  /            ;;TWO(2) SPACES
007570                                        .EVEN
                                        ;*****************************************************************

                                        .SBTTL  ERROR HANDLER ROUTINE

                                        ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                        ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                        ;*AND GO TO $ERRTYP ON ERROR
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW15=1         HALT ON ERROR
                                        ;*SW13=1         INHIBIT ERROR TYPEOUTS
                                        ;*SW10=1         BELL ON ERROR
                                        ;*SW09=1         LOOP ON ERROR
                                        ;*CALL
```

```
                                        ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
                                        $ERROR:
007570  010637  001174                          MOV     SP,$REG6        ;GET THE CURRENT STACK POINTER VALUE
007574  162737  000004  001174                  SUB     #4,$REG6                ;RESTORE IT TO ITS "PRE ERROR TRAP" VALUE FOR PR
007602  016637  000002  001176                  MOV     2(SP),$REG7             ;GET THE PS OFF OF THE STACK
007610  005037  001172                          CLR     $REG5                   ;PREPARE "$REG5" TO HOLD THE TEST #
007614  113737  001102  001172                  MOVB    $TSTNM,$REG5            ;TEST # IS HELD IN THE LOW BYTE OF "TSTNM"
007622  010037  001160                          MOV     R0,$REG0                ;MOST OF THE TIME R0 HAS GOOD STUFF IN IT ALSO
007626  105237  001103                  7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
007632  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
007634  013777  001102  171276          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
007642  032777  002000  171266          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
007650  001402                          BEQ     1$              ;;NO - SKIP
007652  104400  001224                  TYPE    ,$BELL          ;;RING BELL
007656  005237  001112                  1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
007662  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
007666  162737  000002  001116          SUB     #2,$ERRPC
007674  117737  171216  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
007702  032777  020000  171226          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
007710  001004                          BNE     20$             ;;SKIP TYPEOUTS
007712  004737  007434                  JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
007716  104400  001231                  TYPE    ,$CRLF
007722                                  20$:
007722  005777  171210                  2$:     TST     @SWR            ;;HALT ON ERROR
007726  100006                          BPL     3$              ;;SKIP IF CONTINUE
007730  000000                          HALT                    ;;HALT ON ERROR!
007732  022737  006166  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
007740  001001                          BNE     3$              ;;BRANCH IF NO
007742  000000                          HALT                    ;;YES
007744  032777  001000  171164  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
007752  001402                          BEQ     4$              ;;BR IF NO
007754  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
007760  005737  001222          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
007764  001402                          BEQ     5$              ;;BR IF NONE
007766  013716  001222                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
007772                                  5$:
007772  000002                          RTI                     ;;RETURN
                                        ;*******************************************************************

                                        .SBTTL  TRAP DECODER

                                        ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                                        ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                        ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                                        ;*GO TO THAT ROUTINE.

007774  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
007776  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
010002  005740                          TST     -(R0)           ;;BACKUP BY 2
010004  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
010006  006300                          ASL     R0              ;;POSITION FOR INDEXING
010010  016000  010016                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
010014  000200                          RTS     R0              ;;GO TO ROUTINE
```

```
                                        .SBTTL  TRAP TABLE
                                        ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                        ;*BY THE "TRAP" INSTRUCTION.
                                        ;
                                        ;       ROUTINE
                                        ;       -------
010016                                  $TRPAD:
010016  006546                                  $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
010020  007232                                  $TYPOC  ;;CALL=TYPOC    TRAP+1(104401)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
010022  007206                                  $TYPOS  ;;CALL=TYPOS    TRAP+2(104402)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
010024  007246                                  $TYPON  ;;CALL=TYPON    TRAP+3(104403)  TYPE OCTAL NUMBER (AS PER LAST CALL)
010026  006762                                  $TYPDS  ;;CALL=TYPDS    TRAP+4(104404)  TYPE DECIMAL NUMBER (WITH SIGN)
                                        ;******************************************************************
                                        .SBTTL  POWER DOWN AND UP ROUTINES
                                        ;POWER DOWN ROUTINE
010030  012737  010172  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
010036  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
010044  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
010046  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
010050  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
010052  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
010054  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
010056  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
010060  013746  010420                  MOV     POWPUS,-(SP)    ;;PUSH POWPUS ON STACK
010064  013746  010422                  MOV     POWPOP,-(SP)    ;;PUSH POWPOP ON STACK
010070  013746  010210                  MOV     POWMES,-(SP)    ;;PUSH POWMES ON STACK
010074  013746  001672                  MOV     T0001,-(SP)     ;;PUSH T0001 ON STACK
010100  010637  010176                  MOV     SP,$SAVR6       ;;SAVE SP
010104  012737  010116  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
010112  000000                          HALT
010114  000776                          BR      .-2             ;;HANG UP
                                        ;POWER UP ROUTINE
010116  013706  010176          $PWRUP: MOV     $SAVR6,SP       ;;GET SP
010122  005037  010176                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
010126  005237  010176          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
010132  001375                          BNE     1$              ;;OF <POWPUS>,<POWPOP>,<POWMES>,<T0001> WORD
010134  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
010136  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
010140  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
010142  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
010144  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
010146  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
010150  012737  010030  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
010156  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
010164  104400                          TYPE                    ;REPORT THE POWER FAILURE
010166  010200          $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
010170  000002                          RTI
010172  000000          $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
010174  000776                          BR      .-2             ;;  BEFORE THE POWER DOWN WAS COMPLETE
010176  000000          $SAVR6: 0                       ;;PUT THE SP HERE
010200  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
010206  000122
```

```
1794                                                        .EVEN
1795  010210  005015  042522  052123  POWMES: .ASCIZ <15> <12> "RESTARTING AFTER A POWER FIALURE" <15> <12> <12>
1796  010216  051101  044524  043516
1797  010224  040440  052106  051105
1798  010232  040440  050040  053517
1799  010240  051105  043040  040511
1800  010246  052514  042522  005015
1801  010254  000012
1802  010256  005015  046412  026504  STMES:  .ASCIZ  <15><12><12>"MD-11-DZKWA-E LINE FREQUENCY CLOCK TEST"<15><12>
1803  010264  030461  042055  045532
1804  010272  040527  042455  046040
1805  010300  047111  020105  051106
1806  010306  050505  042525  041516
1807  010314  020131  046103  041517
1808  010322  020113  042524  052123
1809  010330  005015     000
1810
1811  010333     124  040522  050120  TRPMES: .ASCIZ  "TRAPPED TO LOC 4 FROM LOCATION  "
1812  010340  042105  052040  020117
1813  010346  047514  020103  020064
1814  010354  051106  046517  046040
1815  010362  041517  052101  047511
1816  010370  020116  000040
1817  010374  042522  052123  051101  TRPM2S: .ASCIZ  "RESTARTING PROGRAM"
1818  010402  044524  043516  050040
1819  010410  047522  051107  046501
1820  010416     000
1821          010420                  .EVEN
1822  010420  001234                  POWPUS: WORD
1823  010422  001234                  POWPOP: WORD
1824  010424  020040  020040  020040  EM1:    .ASCIZ  "                                    LKS     LKS     "
1825  010432  020040  020040  020040
1826  010440  020040  020040  020040
1827  010446  020040  020040  020040
1828  010454  020040  020040  020040
1829  010462  020040  045514  020123
1830  010470  020040  020040  045514
1831  010476  020123  020040  020040
1832  010504     000
1833  010505     050  041520  020051  DH1:    .ASCIZ  "(PC)    (PS)    (SP)    TEST#   WAS     S/B     "
1834  010512  020040  024040  051520
1835  010520  020051  020040  024040
1836  010526  050123  020051  020040
1837  010534  052040  051505  021524
1838  010542  020040  053440  051501
1839  010550  020040  020040  051440
1840  010556  041057  020040  020040
1841  010564  000040
1842                                  .EVEN
1843  010566  001116  001176  001174  DT1:    $ERRPC,$REG7,$REG6,$REG5,LKS,$GDDAT
1844  010574  001172  177546  001124
1845  010602  000000                  0
1846  010604  046103  041517  020113  EM2:    .ASCIZ  "CLOCK FAILED TO INTERRUPT"
1847  010612  040506  046111  042105
1848  010620  052040  020117  047111
1849  010626  042524  051122  050125
```

```
1850  010634  000124
1851  010636  050050  024503  020040  DH2:    .ASCIZ  "(PC)    (PS)    (SP)    TEST#   (LKS)   "
1852  010644  020040  050050  024523
1853  010652  020040  020040  051450
1854  010660  024520  020040  020040
1855  010666  042524  052123  020043
1856  010674  020040  046050  051513
1857  010702  020051  020040     000
1858          010710                  .EVEN
1859  010710  001116  001176  001174  DT2:    $ERRPC,$REG7,$REG6,$REG5,LKS
1860  010716  001172  177546
1861  010722  000000                  0
1862  010724  046103  041517  020113  EM3:    .ASCIZ  "CLOCK INTERRUPTED WHEN THE PROCESSOR PRIORITY WAS TOO HIGH"
1863  010732  047111  042524  051122
1864  010740  050125  042524  020104
1865  010746  044127  047105  052040
1866  010754  042510  050040  047522
1867  010762  042503  051523  051117
1868  010770  050040  044522  051117
1869  010776  052111  020131  040527
1870  011004  020123  047524  020117
1871  011012  044510  044107     000
1872  011017     050  041520  020051  DH3:    .ASCIZ  "(PC)    (PS)    (SP)    TEST#   (LKS)   "
1873  011024  020040  024040  051520
1874  011032  020051  020040  024040
1875  011040  050123  020051  020040
1876  011046  052040  051505  021524
1877  011054  020040  024040  045514
1878  011062  024523  020040  000040
1879                          .EVEN
1880  011070  001116  001176  001174  DT3:    $ERRPC,$REG7,$REG6,$REG5,LKS
1881  011076  001172  177546
1882  011102  000000                  0
1883  011104  046103  041517  020113  EM4:    .ASCIZ  "CLOCK GIVES UNEQUAL # OF PULSES OVER TWO EQUAL PERIODS OF TIME"
1884  011112  044507  042526  020123
1885  011120  047125  050505  040525
1886  011126  020114  020043  043117
1887  011134  050040  046125  042523
1888  011142  020123  053117  051105
1889  011150  052040  047527  042440
1890  011156  052521  046101  050040
1891  011164  051105  047511  051504
1892  011172  047440  020106  044524
1893  011200  042515     000
1894  011203     050  041520  020051  DH4:    .ASCIZ  "(PC)    (PS)    (SP)    TEST#   1ST     2ND"<15><12>"
1895  011210  020040  024040  051520
1896  011216  020051  020040  024040
1897  011224  050123  020051  020040
1898  011232  052040  051505  021524
1899  011240  020040  030440  052123
1900  011246  020040  020040  031040
1901  011254  042116  005015  020040
1902  011262  020040  020040  020040
1903  011270  020040  020040  020040
1904  011276  020040  020040  020040
1905  011304  020040  020040  020040
```

```
 1906  011312  020040  020040  020040
 1907  011320  042520  044522  042117
 1908  011326  020040  042520  044522
 1909  011334  042117     000
 1910          011340                  .EVEN
 1911  011340  001116  001176  001174  DT4:    $ERRPC,$REG7,$REG6,$REG5,$REG1,$REG2
 1912  011346  001172  001162  001164
 1913  011354  000000                  0
 1914  011356  045514  020123  042522  EM5:    .ASCIZ  "LKS REGISTER RESPONDS TO ANOTHER ADDRESS"
 1915  011364  044507  052123  051105
 1916  011372  051040  051505  047520
 1917  011400  042116  020123  047524
 1918  011406  040440  047516  044124
 1919  011414  051105  040440  042104
 1920  011422  042522  051523     000
 1921  011427     050  041520  020051  DH5:    .ASCIZ  "(PC)    (PS)    (SP)    TEST#   ADDRESS"
 1922  011434  020040  024040  051520
 1923  011442  020051  020040  024040
 1924  011450  050123  020051  020040
 1925  011456  052040  051505  021524
 1926  011464  020040  040440  042104
 1927  011472  042522  051523     000
 1928          011500                  .EVEN
 1929  011500  001116  001176  001174  DT5:    $ERRPC,$REG7,$REG6,$REG5,$GDADR
 1930  011506  001172  001120
 1931  011512  000000                  C
 1932  011514  020101  047516  051440  EM6:    .ASCIZ  "A NO SACK TIMEOUT HAS OCCURED"
 1933  011522  041501  020113  044524
 1934  011530  042515  052517  020124
 1935  011536  040510  020123  041517
 1936  011544  052503  042522  000104
 1937  011552  050050  024503  020040  DH6:    .ASCIZ  "(PC)    (PS)    (SP)    TEST#   (LKS)   "
 1938  011560  020040  050050  024523
 1939  011566  020040  020040  051450
 1940  011574  024520  020040  020040
 1941  011602  042524  052123  020043
 1942  011610  020040  046050  051513
 1943  011616  020051  020040     000
 1944          011624                  .EVEN
 1945  011624  001116  001176  001174  DT6:    $ERRPC,$REG7,$REG6,$REG5,LKS
 1946  011632  001172  177546
 1947  011636  000000                  0
 1948  011640  051127  047117  020107  EM7:    .ASCIZ  "WRONG CONDITION CODES WERE PUT ONTO STACK BY INTERRUPT"
 1949  011646  047503  042116  052111
 1950  011654  047511  020116  047503
 1951  011662  042504  020123  042527
 1952  011670  042522  050040  052125
 1953  011676  047440  052116  020117
 1954  011704  052123  041501  020113
 1955  011712  054502  044440  052116
 1956  011720  051105  052522  052120
 1957  011726     000
 1958  011727     050  041520  020051  DH7:    .ASCIZ  "(PC)    (PS)    (SP)    TEST#   CC WAS  CC S/B"
 1959  011734  020040  024040  051520
 1960  011742  020051  020040  024040
 1961  011750  050123  020051  020040
```

```
 1962  011756  052040  051505  021524
 1963  011764  020040  041440  020103
 1964  011772  040527  020123  041440
 1965  012000  020103  027523  000102
 1966                                  EVEN
 1967  012006  001116  001176  001174  DT7:    $ERRPC,$REG7,$REG6,$REG5,BUF1,$GDDAT
 1968  012014  001172  000776  001124
 1969  012022  000000                  0
 1970  012024  051127  047117  020107  EM10:   .ASCIZ  "WRONG PC PUT ONTO THE STACK BY AN INTEPRUPT"
 1971  012032  041520  050040  052125
 1972  012040  047440  052116  020117
 1973  012046  044124  020105  052123
 1974  012054  041501  020113  054502
 1975  012062  040440  020116  047111
 1976  012070  042524  051122  050125
 1977  012076  000124
 1978  012100  050050  024503  020040  DH10:   .ASCIZ  "(PC)    (PS)    (SP)    TEST#  @(SP)WAS @(SP)S/B  "
 1979  012106  020040  050050  024523
 1980  012114  020040  020040  051450
 1981  012122  024520  020040  020040
 1982  012130  042524  052123  020043
 1983  012136  040040  051450  024520
 1984  012144  040527  020123  024100
 1985  012152  050123  051451  041057
 1986  012160  020040     000
 1987          012164                  .EVEN
 1988  012164  001116  001176  001174  DT10:   $ERRPC,$REG7,$REG6,$REG5,BUF2,$GDDAT
 1989  012172  001172  000774  001124
 1990  012200  000000                  0
 1991  012202  051124  042531  020104  EM11:   .ASCIZ  "TRYED TO ACCESS THE LKS REGISTER, AND TRAPPED"
 1992  012210  047524  040440  041503
 1993  012216  051505  020123  044124
 1994  012224  020105  045514  020123
 1995  012232  042522  044507  052123
 1996  012240  051105  020054  047101
 1997  012246  020104  051124  050101
 1998  012254  042520  000104
 1999  012260  050050  024503  020040  DH11:   .ASCIZ  "(PC)    (PS)    (SP)    TEST#"
 2000  012266  020040  050050  024523
 2001  012274  020040  020040  051450
 2002  012302  024520  020040  020040
 2003  012310  042524  052123  000043
 2004                                  .EVEN
 2005  012316  001116  001176  001174  DT11:   $ERRPC,$REG7,$REG6,$REG5
 2006  012324  001172
 2007  012326  000000                  0
 2008          000001                  .END
```

```
A0004   002104            557*    560
A0005   002172            578*    581
A0006   002302            605*    608
A0007   002434            635*    638
A0010   002614            675*    678
A0011   002716            704*    707
A0012   003102            746*    749
A0013   003172            776*    779
A0014   003516            849     858*
A0015   003634            870     882*
A0016   003752            894     906*
A0017   004070            918     930*
A0020   004206            942     954*
A0021   004324            966     978*
A0022   004462           1005    1007*
A0023   004530           1021*   1024
A0024   004652           1049*   1050
A0025   004762           1073*   1076
A0026   005134           1109*   1110
A0027   005250           1135*   1136
A0030   005410           1153    1163*
A0031   005476           1183*   1186
A0032   005630           1220*   1223
A0033   005766           1253*   1256
BIT0  = 000001            291*
BIT00 = 000001            281*    291
BIT01 = 000002            280*    290
BIT02 = 000004            279*    289
BIT03 = 000010            278*    288
BIT04 = 000020            277*    287
BIT05 = 000040            276*    286
BIT06 = 000100            275*    285
BIT07 = 000200            274*    284
BIT08 = 000400            273*    283    1369
BIT09 = 001000            272*    282    1377    1712
BIT1  = 000002            290*
BIT10 = 002000            271*   1694
BIT11 = 004000            270*   1384
BIT12 = 010000            269*
BIT13 = 020000            268*   1701
BIT14 = 040000            267*   1355
BIT15 = 100000            266*
BIT2  = 000004            289*
BIT3  = 000010            288*
BIT4  = 000020            287*
BIT5  = 000040            286*
BIT6  = 000100            285*
BIT7  = 000200            284*
BIT8  = 000400            283*
BIT9  = 001000            282*
BPTVEC= 000014            298*
BUF1  = 000776            310*   1228*   1237    1261*   1967
BUF2  = 000774            309*   1229*   1262*   1270    1988
B0005   002210            579     584*
B0006   002320            606     611*
B0007   002452            636     641*
```

```
B0010   002632          676     681*
B0011   002734          705     710*
B0012   003120          747     752*
B0013   003210          777     782*
B0014   003530          859     861*
B0015   003646          883     885*
B0016   003764          907     909*
B0017   004102          931     933*
B0020   004220          955     957*
B0021   004336          979     981*
B0023   004546         1022    1027*
B0025   005000         1074    1079*
B0030   005426         1165    1167*   1168
B0031   005514         1184    1189*
B0032   005646         1221    1226*
B0033   006004         1254    1259*
C0007   002462          644*    646
C0010   002642          684*    686
C0011   003014          723*    726
C0013   003224          786*    789
C0025   005024         1066    1087*
C0031   005550         1198*   1201
C0032   005712         1230    1237*
C0033   006050         1263    1270*   1272
DDISP = 177570          213*    361
DH1     010505          413    1833*
DH10    012100          448    1978*
DH11    012260          453    1999*
DH2     010636          418    1851*
DH3     011017          423    1872*
DH4     011203          428    1894*
DH5     011427          433    1921*
DH6     011552          438    1937*
DH7     011727          443    1958*
DISPLA  001140          361*    497*   1398*   1693*
DISPRE  000174          319*    497
DSWR  = 177570          212*    360
DT1     010566          414    1843*
DT10    012164          449    1988*
DT11    012316          454    2005*
DT2     010710          419    1859*
DT3     011070          424    1880*
DT4     011340          429    1911*
DT5     011500          434    1929*
DT6     011624          439    1945*
DT7     012006          444    1967*
D0007   002474          627     649*
D0010   002654          662     689*
D0011   003032          724     729*
D0013   003242          787     792*
D0031   005566         1199    1204*
EMTVEC= 000030          301*    479*    480*
EM1     010424          412    1824*
EM10    012024          447    1970*
EM11    012202          452    1991*
EM2     010604          417    1846*
```

```
000443  010724            1422    1962#
000444  011024            1432    1983#
000445  011356            1432    1914#
000446  011514            1437    1932#
000447  011640            1443    1948#
000000= 000004            294#    1360    1361*   1363*   1366*
000001  001730             510     516#
000002  001776             530#
000003  002042             544#
000004  002116             561#
000005  002224             586#
000006  002314             609#
000007  002446             639#
000010  002626             698#
000011  003000             768#
000012  003114             800#
000013  003204             830#
000014  003526             960#
000015  003644             994#
000016  003762            1008#
000017  004100            1032#
000020  004216            1056#
000021  004334            1080#
000022  004460            1006#
000023  004540            1025#
000024  004672            1041    1054#
000025  004774            1067#
000026  005154            1104    1114#
000027  005302            1141#
000030  005442            1163    1171#
000031  005510            1197#   1195
000032  005642            1224#
000033  006000            1257#
000036  002342             616#
000037  002470             647#
000040  002650             687#
000041  003134             757#
000042  003336             790#
000043  004464            1031#
000044  005020            1085#
000045  005202            1132#
000046  006060            1272#
000206  002226             588#
000211  003026             720     727#
000207  002502             651#
000210  002662             691#
000211  003040             733#
000212  003320             810#
000225  005034            1089#
000231  005574            1208#
000213  003352             820#
000214  003414             833#
000213  003246             794#    799
000215  ******             318    1746    1747    1748    1749    1750
000216  003262             795     798#
000217  003306             806#    809
```

D2KWAE.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
IOTVEC= 000020           299*    477*    478*
I0001   001722           514*
I0014   003510           857*
I0015   003626           881*
I0016   003744           905*
I0017   004062           929*
I0020   004200           953*
I0021   004316           977*
I0022   004442          1003*
I0023   004546          1028*
I0026   005142          1111*
J0013   003324           807     912*
KSTART  001400           323     460*
K0013   003340           816*    819
LKS   = 177546           307*    514*    528     542     555*    557     576*    578     585*    596     603*    605     612*
                         613*    614     633*    635     642*    649     664*    675     692*    689     702*    704     721*
                         723     744*    746     753*    755     773*    776     784*    786     793*    794     797*    803*
                         906     814*    816     824*    825     828*    856*    958     861*    880*    982     885*    904*
                         906     909*    928*    930     933*    952*    954     957*    976*    979     991*    1002*   1004
                        1007*   1019*   1021    1028*   1029    1042*   1048*   1052    1067*   1073    1083*   1097    1106*
                        1108*   1112    1126*   1137*   1138*   1139    1150*   1159*   1161*   1169*   1181*   1183    1196*
                        1198    1218*   1220    1231*   1251*   1253    1264*   1279*   1843    1859    1880    1945
L0013   003356           817     822*
M0013   003370           825*    830
NOP   = 000240           308*
N0013   003404           826     829*
PC    = 000007           225*    469*    1297*   1300*   1312*   1323*   1330*   1440*   1447*   1454*   1468*   1470*   1659*
                        1703*
PIRQ  = 177772           211*
PIRQVE= 000240           305*
PCMES   010210          1766    1795*
PCMPOP  010422          1765    1823*
PCMPUS  010420          1764    1822*
PR0   = 000000           228*
PR1   = 000040           229*
PR2   = 000100           230*
PR3   = 000140           231*
PR4   = 000200           232*
PR5   = 000240           233*    667
PR6   = 000300           234*    714
PR7   = 000340           235*    572     599     768     851     975     899     923     947     971     997     1044    1069
                        1100    1129    1191
PS    = 177776           208*    209     667*    714*
PSW   = 177776           209*
PWRVEC= 000024           300*    483*    484*    1756*   1757*   1769*   1784*   1785*
RESVEC= 000010           295*
R0    = 000000           216*    556*    559*    643*    644*    683*    684*    766*    798*    800*    829*    1309*   1312
                        1424    1425*   1426    1429*   1489    1499*   1503    1519    1520    1533*   1631    1632*   1633*
                        1640*   1641*   1642*   1643*   1644*   1645    1650    1655*   1657*   1661    1663    1690    1729
                        1730*   1731    1732*   1733*   1734*   1735*   1758    1783*
R0001   001716           512     513*
R0002   001756           525     526*
R0003   002032           540     541*
R0004   002070           553     554*
R0005   002164           571     576*
R0006   002274           598     603*
```

```
R0007   002412          626     629*
R0010   002544          661     664*
R0011   002710          701     702*
R0012   003074          743     744*
R0013   003140          766*    823
R0014   003500          947     955*
R0015   003616          873     879*
R0016   003734          897     903*
R0017   004052          921     927*
R0020   004170          945     951*
R0021   004306          969     975*
R0022   004432          995    1001*
R0023   004522         1018    1019*
R0024   004632         1040    1043*
R0025   004742         1064    1067*
R0026   005116         1099    1106*
R0027   005222         1125    1128*
R0030   005342         1152    1153*
R0031   005470         1180    1181*
R0032   005622         1217    1218*
R0033   005760         1250    1251*
R1      =.000001      217*     767*    796*    831     833     1490    1503*   1504    1508    1532*   1759    1782*
R1013     003156      772*
R2      =.000002      218*     801*    827*    831     834     1491    1502*   1506*   1509    1516*   1517*   1518    1523*
                     1531*    1760    1781*
R2013     003210      783*
R3      =.000003      219*     577*    580*    604*    607*    634*    637*    674*    677*    703*    706*    722*    725*
                      745*     748*    775*    778*    785*    788*    805*    808*    815*    818*   1020*   1023*   1072*
                     1075*    1182*   1185*   1197*   1200*   1219*   1222*   1252*   1255*   1492    1500*   1501*   1515*
                     1518*    1527*   1528*   1530*   1577    1586*   1592*   1593*   1596*   1601*   1602*   1603    1612*
                     1761     1780*
R3013     003272      802*
R4      =.000004      220*    1578     1580*   1581*   1582*   1583    1594*   1598    1600*   1608*   1611*   1762    1779*
R4013     003324      813*
R5      =.000005      221*    1493     1495*   1497*   1504*   1508*   1523    1529*   1579    1585*   1587*   1589*   1590*
                     1591*    1592     1610*   1763    1778*
R6      =.000006      222*     224      472*    473*    474
R7      =.000007      223*     225
SP      =.000006      224*     460*     461*    462*    465*    466*    468     476*    491*    492*    498     499     500
                      513*     572*     573*    599*    600*    629*    630*    631*    665*    669     670     672     673*
                      711*     712*     716     717     719     768*    769*    951*    952*    855*    975*    876*    979*
                      899*     900*     903*    923*    924*    927*    947*    948*    951*    971*    972*    975*    997*
                      998*    1001*    1044*   1045*   1068*   1069*   1080*   1081*   1100*   1101*   1107*   1129*   1130*
                     1133*    1155*    1156*   1158*   1164*   1165*   1190*   1191*   1192*   1227*   1232*   1233*   1260*
                     1265*    1266*    1304*   1322*   1325*   1326*   1329*   1360*   1363    1365    1366    1394    1395
                     1399*    1424*    1425    1426*   1428    1429    1430*   1432    1434    1436    1441    1443*   1445*
                     1453*    1457     1461    1462    1466    1489*   1490*   1491*   1492*   1493*   1494*   1495    1498*
                     1511     1513*    1515    1525    1527    1529    1530    1531    1532    1533    1535*   1536*   1569*
                     1570     1571     1572*   1577*   1578*   1579*   1585    1610    1611    1612    1613*   1614*   1631*
                     1636*    1657     1661*   1685    1687    1698    1714*   1717*   1729*   1730    1758*   1759*   1760*
                     1761*    1762*    1763*   1764*   1765*   1766*   1767*   1768    1774*   1778    1779    1790    1791
                     1782     1783
STACK = 001100       205*
START   001450       471*    1337
SV_MT = 177774       210*
SYSES   013256       470     1802*
```

```
SWR     001136            360*    494     496*   1355    1369    1371    1377    1384    1694    1701    1706    1712
SWREG   000176            320*    496
SW0   = 000001            263*
SW00  = 000001            253*    263
SW01  = 000002            252*    262
SW02  = 000004            251*    261
SW03  = 000010            250*    260
SW04  = 000020            249*    259
SW05  = 000040            248*    258
SW06  = 000100            247*    257
SW07  = 000200            246*    256
SW08  = 000400            245*    255
SW09  = 001000            244*    254
SW1   = 000002            262*
SW10  = 002000            243*
SW11  = 004000            242*
SW12  = 010000            241*
SW13  = 020000            240*
SW14  = 040000            239*
SW15  = 100000            238*
SW2   = 000004            261*
SW3   = 000010            260*
SW4   = 000020            259*
SW5   = 000040            258*
SW6   = 000100            257*
SW7   = 000200            256*
SW8   = 000400            255*
SW9   = 001000            254*
TBITVE= 000014            296*
TCRLF = 000200           1434    1473*
THT   = 000011           1432    1472*
TKVEC = 000060            303*
TPVEC = 000064            304*
TRAPVE= 000034            302*    481*    482*
TRAPC   006222            523     537     551     568     595     623     659     699     741     992    1015    1037    1061
                         1096    1122    1148    1178    1215    1248    1322*
TRPMES  010333           1324    1811*
TRPM2S  010374           1331    1817*
TRTVEC= 000014            297*
TYPOS = 104404           1305    1750*
TYPE  = 104400           1303    1306    1328    1437    1534    1604    1630    1647    1649    1652    1654    1658    1665
                         1696    1704    1746*   1786
TYPOC = 104401           1327    1638    1662    1747*
TYPON = 104403           1749*
TYPOS = 104402           1748*
T0001   001672            502     509*   1317    1767
T0002   001732            515     522*
T0003   002000            529     536*
T0004   002044            543     550*
T0005   002120            558     567*
T0006   002230            583     587     594*
T0007   002344            610     615     622*
T0010   002504            640     648     650     658*
T0011   002664            671     680     688     690     698*    728
T0012   003042            709     718     732     739*
T0013   003136            751     756     765*
```

```
T0014   003430            781     791     811     821     832     845#
T0015   003546            862     869#
T0016   003664            886     893#
T0017   004002            910     917#
T0020   004120            934     941#
T0021   004236            958     965#
T0022   004354            982     990#
T0023   004470            991    1014#
T0024   004566           1026    1030    1036#
T0025   004674           1053    1060#
T0026   005036           1078    1086    1088    1095#
T0027   005156           1113    1121#
T0030   005304           1127    1140    1147#
T0031   005444           1170    1177#
T0032   005576           1188    1203    1207    1214#   1241
T0033   005734           1225    1239    1247#
T0034   006070           1258    1271    1278#
WORD    001234            395#   1134#   1135#   1160#   1167#   1822    1823
SBDADR  001122            354#
SBDDAT  001126            356#    474
SBELL   001224            390#   1696    1720
SCHARC  006756           1448#   1455    1464#   1469#
SCMTAG  001100            342#    472     479     485     486     487
SCM1    = 000010          372#    373#    374#    375#    376#    377#    378#    379#    380#
SCM2    = 000020          372#    373#    374#    375#    376#    377#    378#    379#    380#
SCM3    = 000010          370#    372
SCM4    = 000010          380#    381#    382#    383#    384#    385#    386#    387#    388#
SCRLF   001231            392#   1328    1438    1475    1630    1649    1654    1658    1704    1720
SDBLK   007176           1500    1534    1542#
SDDAGN  006176           1299    1310    1316#
SDTBL   007166           1503    1538#
SENDAD  006166            330    1312#   1709
SENDCT  006134            485    1301#
SENDMG  006202           1303    1318#
SENULL  006217           1306    1321#
SEOP    006100           1291#
SEOPCT  006126            485#   1298#   1302
SERFLG  001103            345#   1344    1373    1375    1381#   1402    1691#   1720
SERMAX  001115            351#    488#   1375    1397#   1402
SERROR  007570            479    1684#
SERRPC  001116            352#   1636    1698#   1699#   1700    1720    1843    1859    1880    1911    1929    1945    1967
                         1988    2005
SERRTB  001236            411#   1644
SERRTY  007434           1629#   1703
SERTTL  001112            349#   1697#   1720
SESCAP  001222            389#    487#   1396#   1715    1717    1720
SFILLC  001154            368#   1441    1475
SFILLS  001153            367#   1475
SGDADR  001120            353#    848#    857#    863#    874#    881#    887#    898#    905#    911#    922#    929#    935#
                          946#    953#    959#    970#    977#    983#    996#   1003#   1929
SGDDAT  001124            355#    527#    539#    554#    570#    597#    625#    740#    846#    972#    896#    920#    944#
                          968#    994#   1017#   1029    1039#   1052    1063#   1098#   1112    1124#   1139    1151#   1239#
                         1272#   1843    1967    1988
SGET42  006156           1307#
SHD     = 000000          188
SICNT   001104            346#   1388#   1389    1391#   1401
```

```
$ILLUP  010172          1756    1789#
$ITEMB  001114           350#   1633    1700*   1720
$LF     001232           393#   1475    1720
$LPADR  001106           347#    489*    512*    525*    540*    553*    571*    598*    626*    661*    701*    743*    772*
                         783*    802*    813*    823*    847*    873*    897*    921*    945*    969*    995*   1018*   1040*
                        1064*   1099*   1125*   1152*   1180*   1217*   1250*   1379*   1394*   1399    1401
$LPERR  001110           348#    490*   1379    1395*   1401    1714
$MAIL = ****** U         501    1394    1426    1706
$MXCNT  006544          1392    1401#
$NULL   001152           366#   1443    1475
$OCNT   007430          1576*   1605*   1618#
$OMODE  007432          1571*   1575*   1580    1583*   1594*   1620#
$OVER   006530          1356    1372    1380    1390    1398#
$PASS   001100           343#   1295*   1296*   1304    1318    1386    1402
$POWER  010200          1787    1792#
$PWRDN  010030           483    1756#   1784
$PWRMG  010166          1787#
$PWRUP  010116          1769    1774#
$QUES   001230           391#   1475    1720
$RDCHR= ****** U        1751
$RDDEC= ****** U        1751
$RDLIN= ****** U        1751
$RDOCT= ****** U        1751
$REGAD  001156           370#
$REG0   001160           372#   1690#
$REG1   001162           373#    833    1911
$REG2   001164           374#    834    1911
$REG3   001166           375#
$REG4   001170           376#
$REG5   001172           377#   1688*   1689*   1843    1859    1880    1911    1929    1945    1967    1988    2005
$REG6   001174           378#   1685*   1686*   1843    1859    1880    1911    1929    1945    1967    1988    2005
$REG7   001176           379#   1687*   1843    1859    1880    1911    1929    1945    1967    1988    2005
$R2A  = ****** U        1751
$SAVRE= ****** U        1751
$SAVR6  010176          1768*   1774    1775*   1776*   1791#
$SCOPE  006274           477    1353#
$SETUP= 000037           178#    477     479     481     483     485     486     487     499    1293    1709
$STUP = 177777           178#
$SVLAD  006502          1364    1393#
$SVPC = 000204           328#    333
$SWR  = 167400           172#    185     193     194     195     196     197     198     199     388     389     390     486
                         487     489     490    1288    1294    1308    1317    1318    1345    1346    1347    1348    1349
                        1355    1367    1369    1370    1373    1374    1375    1382    1383    1384    1395    1398    1401
                        1676    1677    1678    1679    1680    1694    1701    1706    1712    1720    1788
$SWRMK= 000000           199     200    1349    1350    1371
$TIMES  001220           398#    486*   1294*   1382*   1389    1392*   1401
$TKB    001144           363#
$TKS    001142           362#
$TMP0   001200           380#
$TMP1   001202           381#
$TMP2   001204           382#
$TMP3   001206           383#
$TMP4   001210           384#
$TMP5   001212           385#
$TMP6   001214           386#
$TMP7   001216           387#
```

```
$TN    = 000000            173#   188
$TPB     001150            365#  1461*  1475
$TPFLG   001155            369#  1420   1475
$TPS     001146            364#  1459   1475
$TRAP    007774            481   1729#
$TRP   = 000005           1737#  1747#  1748#  1749#  1750#  1751#
$TPPAD   010016           1734   1745#
$TSTNM   001102            344#  1293*  1344   1371   1393*  1398   1402   1689   1693   1720
$TYPBN=  ****** U         1751
$TYPDS   006762           1488#  1750
$TYPE    006546            469   1323   1330   1420#  1737   1746
$TYPEC   006712           1440   1447   1454   1459#  1460
$TYPEX   006760           1465   1467   1470#
$TYPOC   007232           1574#  1747
$TYPON   007246           1573   1576#  1749
$TYPOS   007206           1569#  1748
$XTSTR   006306           1358#
$DFILL   007431           1570*  1574*  1584   1619#
$40CAT=  ****** U         1355   1703
.      = 012330            314#   318#   328    329#   331#   333#   341#   334    459#   475    489    490    1318
                          1322   1401   1402   1475   1542#  1668#  1720   1771   1790   1821#  1858#  1910#  1928#
                          1944#  1987#
```

```
ADITAG    334*    395
COMMEN      1*    306*
ENDCOM      1*    306*
ERROR     206*    516     530     544     561     582     588     609     616     639     647     651     679     687     691
          708     727     733     750     757     780     790     810     820     835     860     884     908     932     956
          980    1006    1025    1031    1054    1077    1085    1089    1114    1141    1171    1197    1202    1208    1224
         1240    1257    1273
ESCAPE      1*    306*
GETPRI      1*    177*    461
MULT        1*    306*
NEWTST      1*    306*
POP         1*    306*    1529    1778
PUSH        1*    306*    1488    1758    1764
REPORT      1*
SAVE     1669*   1685
SCOPE     207*    509     522     536     550     567     594     622     658     698     739     765     845     869     893
          917     941     965     990    1014    1036    1060    1095    1121    1147    1177    1214    1247    1278    1292
SETPRI      1*    177*    572     599     768     851     875     899     923     947     971     997    1043    1068    1100
         1129    1191
SETRAP    178*    523     537     551     568     595     623     659     699     741     992    1015    1037    1061    1096
         1122    1148    1178    1215    1248
SETTRA   1737*   1747    1748    1749    1750
SETUP       1*    306*    471
SKIP        1*    306*
SLASH       1*    306*
SPACE     306*
STARS       1*    176*    306*    324     334     394    1282    1338    1402    1475    1543    1621    1669    1720    1751
SWRSU       1*    491*
TRMTRP   1737*
TYPBIN      1*    306*
TYPDEC      1*    306*    1304
TYPNAM      1*
TYPNUM      1*    306*
TYPOCS      1*    306*
TYPOCT      1*    306*    1636    1660
TYPTXT      1*    306*
WAITLK    178*    577     604     634     674     703     722     745     775     785     805     815    1020    1072    1182
         1197    1219    1252
$$CMRE    334*    372     373     374     375     376     377     378     379
$$CMTM    334*    380     381     382     383     384     385     386     387
$$ESCA      1*    306*
$$NEWT      1*    306*
$$SET    1737*   1747    1748    1749    1750
$$SKIP      1*    306*
.EQUAT      1*    175*    201
.HEADE      1*    175*    178
.KT11       1*
.SETUP      1*    175*    178
.SWRHI      1*    175*    188
.SWRLO    175*    200*
.$ACTI      1*    177*    324
.$APTB      1*
.$APTH      1*
.$APTY      1*
.$ASTA      1*
.$CATC      1*    175*    311
```

```
.$CMTA      1#    176#    334
.$DB2D      1#
.$DB2O      1#
.$DIV       1#
.$EOP       1#    175#   1282
.$ERRO      1#    176#   1669
.$ERRT      1#    176#   1621
.$MULT      1#
.$POWE      1#    176#   1751
.$RAND      1#
.$RDDE      1#
.$RDOC      1#
.$READ      1#
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#    175#   1338
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#    176#   1720
.$TYPB      1#
.$TYPD      1#    176#   1475
.$TYPE      1#    176#   1402
.$TYPO      1#    176#   1543
.$40CA      1#
```

```
ADD     1430  1508  1572  1582  1644
ASL     1641  1642  1643  1733
ASLB    1513
BCC     1514
BEQ      502   529   615   832   859   893   907   931   955   979  1030  1053  1088  1113  1140
        1238  1271  1310  1370  1372  1374  1378  1387  1433  1599  1646  1651  1664  1692  1695
        1713  1716
BGE     1390  1467
BGT     1299  1522  1606
BHI     1376
BIC     1296  1596
BIS     1516  1517  1601  1602
BISB    1633
BIT      528   586   614   858   882   906   930   954   978  1004  1355  1369  1377  1384  1694
        1701  1712
BITB    1455
BLT     1446  1505  1521  1607
BMI      543   558   579   606   636   650   676   690   705   724   747   756   777   787   807
         817  1022  1074  1184  1199  1221  1254  1512
BNE      475   560   581   587   608   638   646   678   686   707   726   749   779   799   799
         809   819   830  1005  1024  1050  1076  1110  1136  1168  1186  1201  1223  1256  1356
        1385  1427  1435  1442  1456  1463  1510  1597  1634  1656  1702  1710  1777
BPL      795   826  1421  1460  1496  1526  1595  1707
BR       495   515   583   610   640   648   668   671   680   688   709   715   718   728   732
         751   781   791   811   821  1026  1078  1086  1170  1188  1203  1207  1225  1241  1258
        1358  1364  1367  1380  1383  1423  1439  1449  1458  1465  1507  1524  1573  1588  1609
        1639  1666  1771  1790
CLR      461   473   486   487   514   555   556   576   577   603   604   613   630   633   634
         643   664   674   683   702   703   722   744   745   753   766   767   773   775   784
         785   793   797   800   801   803   805   814   815   824   828   846   856   861   863
         872   880   885   887   896   904   909   911   920   928   933   935   944   952   957
         959   968   976   981   983  1002  1007  1019  1020  1042  1067  1072  1080  1106  1126
        1128  1134  1150  1154  1155  1159  1164  1169  1181  1182  1197  1218  1219  1228  1229
        1232  1251  1252  1261  1262  1265  1279  1293  1294  1322  1329  1382  1396  1499  1502
        1586  1632  1688  1775
CLRB    1160  1381  1464  1528
CMP      474   498   669   716   831  1029  1052  1087  1112  1139  1237  1270  1365  1389  1520
        1709
CMPB    1371  1375  1432  1434  1441  1462  1466
DEC     1297  1640
DECB    1445  1448  1594  1605
EMT      206
HALT     318  1422  1708  1711  1770  1789
INC      559   580   607   637   644   677   684   706   725   748   778   788   796   798   808
         818   827   829  1023  1049  1075  1109  1135  1185  1200  1222  1255  1295  1388  1506
        1600  1608  1697  1776
INCB    1167  1393  1468  1691
IOT      207
JMP      323   991  1317  1337
JSR      469  1312  1323  1330  1440  1447  1454  1703
MOV      460   462   463   465   466   468   472   476   477   478   479   480   481   482   483
         484   485   489   490   491   492   493   496   497   499   500   510   511   512   513
         523   524   525   527   537   538   539   540   551   552   553   554   568   569   570
         571   572   573   595   595   596   597   598   599   600   612   623   624   625   626
         627   628   629   631   642   659   660   661   662   663   665   666   667   670   672
         673   682   699   700   701   711   712   713   714   717   719   720   721   740   741
```

```
            742   743   768   769   772   783   802   813   823   833   834   847   848   849   850
            851   852   855   857   862   870   871   873   874   875   876   879   881   886   894
            895   897   898   899   900   903   905   910   918   919   921   922   923   924   927
            929   934   942   943   945   946   947   948   951   953   958   966   967   969   970
            971   972   975   977   982   992   993   994   995   996   997   998  1001  1015  1016
           1017  1018  1028  1037  1038  1039  1040  1041  1044  1045  1048  1061  1062  1063  1064
           1065  1066  1068  1069  1081  1083  1096  1097  1098  1099  1100  1101  1104  1105  1107
           1108  1122  1123  1124  1125  1127  1129  1130  1133  1137  1138  1148  1149  1151  1152
           1153  1156  1158  1161  1163  1165  1178  1179  1180  1190  1191  1192  1195  1196  1215
           1216  1217  1227  1230  1231  1233  1239  1248  1249  1250  1260  1263  1264  1266  1272
           1300  1304  1309  1325  1360  1361  1363  1366  1379  1391  1392  1394  1395  1398  1399
           1424  1425  1429  1443  1489  1490  1491  1492  1493  1494  1495  1500  1503  1523  1529
           1530  1531  1532  1533  1535  1536  1569  1577  1578  1579  1585  1592  1610  1611  1612
           1613  1614  1631  1636  1645  1650  1655  1657  1661  1685  1687  1690  1693  1698  1714
           1717  1729  1730  1734  1756  1757  1758  1759  1760  1761  1762  1763  1764  1765  1766
           1767  1768  1769  1774  1778  1779  1780  1781  1782  1783  1784  1785
MOVB        488  1003  1397  1426  1453  1461  1498  1501  1515  1518  1527  1570  1571  1574  1575
           1576  1580  1583  1584  1603  1689  1700  1732
NEG        1497  1581
NOP         645   685   730   731   836  1205  1206  1313  1314  1315  1332  1333  1334  1335  1354
RESET       503   526   541   754  1006  1051  1111  1280  1311  1336
ROL        1587  1589  1590  1591  1593
RTI         467   574   601   632   770   853   877   901   925   949   973   999  1046  1070  1082
           1102  1131  1157  1166  1193  1234  1267  1400  1431  1537  1615  1719  1788
RTS        1470  1659  1735
SEC        1235  1268
SEN        1235  1268
SEV        1235  1268
SEZ        1235  1268
SUB        1326  1504  1686  1699
TRAP        464  1737  1747  1748  1749  1750
TST         494   501  1362  1386  1428  1436  1457  1509  1519  1598  1663  1706  1715  1731
TSTB        542   557   578   605   635   649   675   689   704   723   746   755   776   786   794
            806   816   825  1021  1073  1183  1198  1220  1253  1373  1420  1459  1511  1525
WAIT       1084  1162  1236  1269
.ABS        168
.ASCII      391   392
.ASCIZ      390   393  1318  1667  1792  1795  1802  1811  1817  1824  1833  1846  1851  1862  1872
           1883  1894  1914  1921  1932  1937  1948  1958  1970  1978  1991  1999
.BLKW      1542
.BYTE       344   345   350   351   366   367   368   369  1321  1616  1617  1618  1619
.ENABL        1   169   174
.END          8
.ENDC       178   183   196   198   199   200   206   292   306   324   325   331   333   335   342
            344   370   380   388   389   390   391   395   396   476   477   479   481   483   485
            486   487   489   491   501  1283  1286  1287  1288  1290  1293  1299  1302  1303  1307
           1308  1317  1318  1321  1322  1339  1345  1350  1355  1357  1368  1371  1372  1373  1375
           1377  1384  1388  1393  1394  1398  1401  1402  1403  1426  1476  1544  1622  1640  1669
           1670  1676  1691  1698  1703  1704  1705  1706  1712  1719  1720  1721  1730  1733  1746
           1747  1748  1749  1750  1751  1752  1768  1778  1788  1795
.EQUIV      206   207   209   224   225   254   255   256   257   258   259   260   261   262   263
            282   283   284   285   286   287   288   289   290   291
.EVEN      1668  1794  1821  1842  1858  1879  1910  1928  1944  1966  1987  2004
.IF         178   179   196   197   198   199   200   204   264   292   321   324   329   331   334
            341   343   370   380   388   389   390   394   395   472   476   477   479   481   483
            485   486   487   489   501  1282  1286  1287  1288  1289  1290  1292  1298  1301  1303
```

DZKWAE.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
                1307  1308  1317  1318  1338  1344  1349  1354  1355  1367  1369  1370  1371  1373  1374
                1375  1384  1386  1394  1395  1400  1401  1402  1426  1475  1543  1621  1639  1655  1669
                1675  1685  1694  1701  1703  1704  1706  1709  1712  1719  1720  1729  1733  1737  1747
                1748  1749  1750  1751  1764  1778  1786  1788  1792
.IFF             196   198   199   200   204   325   331   333   335   342   343   370   395   477  1283
                1289  1293  1299  1302  1318  1339  1368  1371  1372  1375  1401  1403  1476  1544  1622
                1640  1669  1670  1675  1694  1719  1720  1721  1730  1752  1788
.IFT            1383  1704
.IFTF           1301  1703
.IIF             178   183   188   193   194   195   196   199   200   318   394   477   479   485   488
                 487   489   490  1287  1293  1294  1305  1318  1322  1345  1346  1347  1348  1349  1350
                1382  1383  1398  1401  1402  1475  1637  1662  1676  1677  1678  1679  1680  1709  1720
                1746  1747  1748  1749  1750
.IRP             178   395  1354  1489  1529  1685  1758  1764  1778
.LIST              1   170   178   199   306   318   370   372   373   374   375   376   377   378   379
                 380   381   382   383   384   385   386   387   388   491  1293  1349  1709  1737  1746
                1747  1748  1749  1750  1751
.MACRO             1   178   200   334  1669  1737
.MCALL           175   176   177   306   491
.NLIST             1   171   179   199   306   318   370   372   373   374   375   376   377   378   379
                 380   381   382   383   384   385   386   387   388   491  1293  1349  1709  1737  1746
                1747  1748  1749  1750  1751
.PAGE            334   394
.REM               1
.REPT            318   372   380
.SBTTL           189   202   312   322   326   336   397   507   520   534   548   565   592   620   655
                 695   737   761   840   867   891   915   939   963   987  1012  1035  1058  1093  1118
                1145  1175  1212  1245  1277  1284  1340  1404  1477  1545  1625  1671  1722  1738  1753
.TITLE           178
.WORD            318   319   320   332   343   346   347   348   349   352   353   354   355   356   357
                 358   359   360   361   370   372   373   374   375   376   377   378   379   380   381
                 382   383   384   385   386   387  1298  1301  1469  1620  1648  1653  1787
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*DZKWAE,DZKWAE.SEQ/SOL/CRF/DS:ERFZ/EN:ABS=DSKM:SYSMAC.SML,DSKM:DZKWAE.P11
RUN-TIME: 41 33 4 SECONDS
RUN-TIME RATIO: 138/80=1.7
CORE USED: 29K (57 PAGES)

Spooler runtime 0 Seconds, 37 KCS, 235 disk reads, 3 disk writes, 51 pages