KW11-L

LINE FREQUENCY CLOCK TEST

MD-11-DDKWA-A

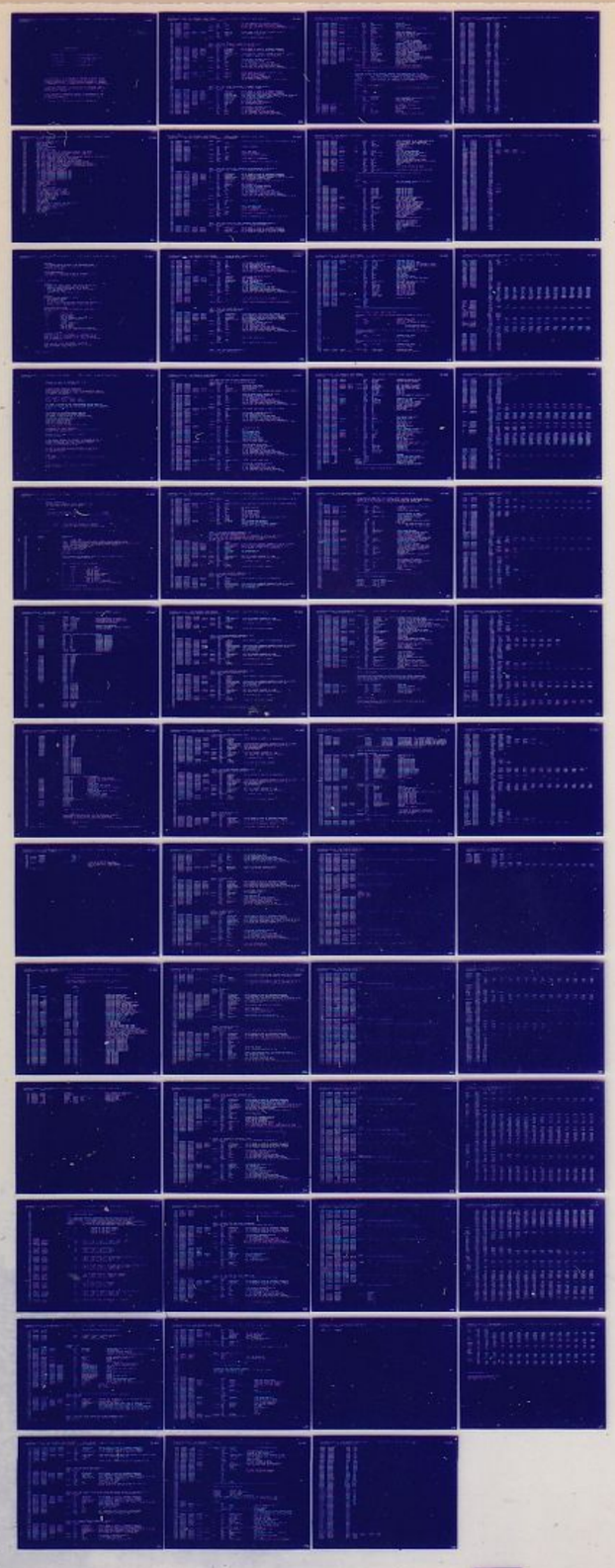
EP DDKWA A DL

FICHE 1 OF 1

JAN 1978

digital

MADE IN USA



KW11

LINE FREQUENCY CLOCK TEST

MD-11-DDKWA-A

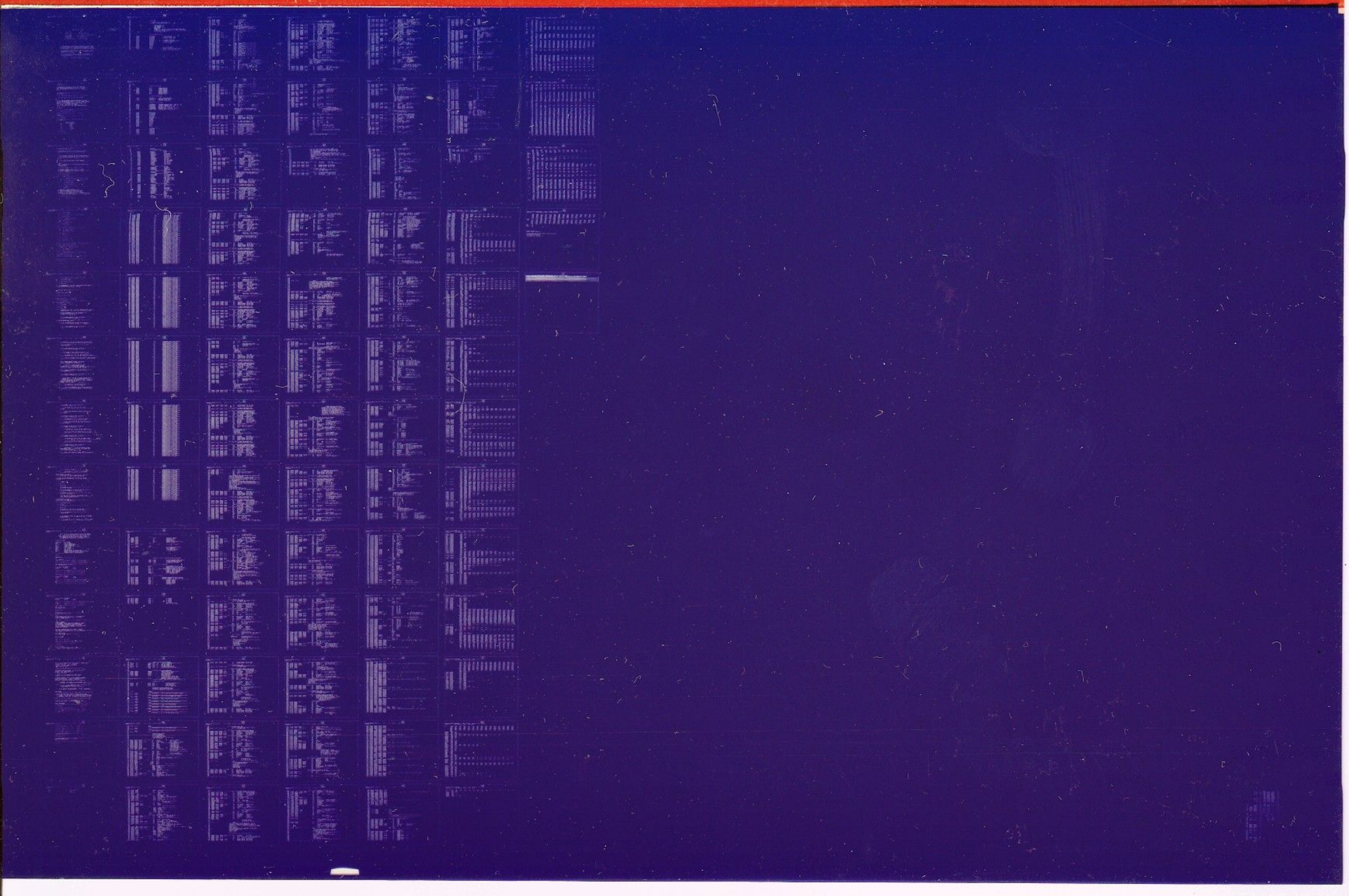
EP-DDKWA-A-DL-A

OCT 1976

digital

Made in U.S.A.

FICHE 1 OF 1



```
1
2
3
```

```
.REM {
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DDKWA-A-D
PRODUCT NAME:	LINE FREQUENCY CLOCK TEST
DATE RELEASED:	21 DECEMBER 1979
MAINTAINER:	DIAGNOSTIC GROUP

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 SET SW15 = 1 (UP)
C SELECT OTHER SWITCH REGISTER OPTIONS (SEE SECTION 2,3)
D 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 SWR8 = 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.

THERE WILL BE A 5 ON THE DISPLAY LIGHTS FOR ABOUT 20 SECONDS AFTER COMPLETION OF EATCH PASS, (COUPLE OF SECS, IF SW11 IS UP)

3,0 ERROR INFORMATION

PROGRAM WILL HALT AT LOCATION 7272 (SHOWN
ON DISPLAY LIGHTS) ON AN ERROR.

3.1 STANDARD ERROR REPORTING PROCEDURES
ERROR PRINTOUTS CONSIST OF FROM 4 TO 8 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 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 RELEVENT TO A PARTICULAR
ERROR.

THE POINTER TO THE ERROR MESSAGE AND THE
FIRST FOUR COLUMNS OF THE ERROR MESSAGE
ARE STORED IN CORE STARTING AT LOCATION 17400
FOR EXAMPLE, 17400 WILL CONTAIN POINTER TO EMXX
17402 WILL CONTAIN ERROR PC
17404 WILL CONTAIN THE PS
17406 WILL CONTAIN THE SP
17410 WILL CONTAIN THE TEST # THAT FAILED.

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.

POINTER TO THE ERROR MESSAGE IS STORED IN LOC. 17400
AND THE ADDRESS WHERE THE TRAP OCCURED IS STORED
IN LOCATION 17402.

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
                        -----------------------------------------------------------------
                        !   !   !   !   !   !   !   !   !   !   !   !   !   !   !   !   !
                                                          7   6
                        !   !   !   !   !   !   !   !   !   !   !   !   !   !   !   !   !
                        -----------------------------------------------------------------

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


        7.0     FLOW CHARTS
                                        C

        167400                  $SWR=167400
        000000                  $TN=0
                                        .ENABL  ABS
                                .MCALL  .HEADER,.SCATCH,.SEOP,.EQUAT,.SWRHI,.SWRLO,.SSCOPE,.SETUP
                                .MCALL  .STYPOCT,.STYPDEC,.STRAP,.SPOWER,.SERROR,.STYPE,STARS,.SERRTYP,.SCNTAG
                                .TITLE  MAINDEC-11-DDKWA-A  LINE FREQUENCY CLOCK PROGRAM
                                ;*COPYRIGHT (C) 1970,1972,1975
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY J. COMEAU
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-A1).
                                ;*

                                .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
                                ;*         9                    LOOP ON ERROR
                                ;*         8                    LOOP ON TEST IN SWR<7:0>
                                ;*        7-0                   #OF TEST TO LOOP ON IF SWR<8> 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          SWR=    177570                  ;SWITCH REGISTER
177570          DISPLAY=SWR

                ;*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=     %6                      ;GENERAL REGISTER
000007          R7=     %7                      ;GENERAL REGISTER
                .EQUIV  R6,SP                   ;STACK POINTER
                .EQUIV  R7,PC                   ;PROGRAM COUNTER

                ;*"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
                        000004          BIT02=  4
                        000002          BIT01=  2
                        000001          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

                                        .SBTTL  STARTING ADDRESS(ES)
                        000200                  .=200

000200  000137  001430                  JMP     @#KSTART                ;JUMP TO STARTING ADDRESS OF PROGRAM
```

```
                    000050                          .=50
000050  000000                        .WORD   0
000052  000000                        .WORD   0
                    000250                          .=250
000250  000000              EOPHLT: HALT            ;THIS IS AN END OF PASS HALT;
                                                    ;NOT AN ERROR HALT, THIS HAPPENS
                                                    ;ONLY IF SW0 IS UP, PRESS CONTINUE FOR ANOTHER
                                                    ;PASS.
000252  000207                      RTS     PC
```

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

                                        .SBTTL  COMMON TAGS

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

          000046                                .=46
 000046   005536                                $ENDAD                          ;LOGICAL END OF 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 1100
 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 1100 OF 'GOOD' DATA
 001122   000000                        $BDADR: .WORD   0                       ;CONTAINS 1100 OF 'BAD' DATA
 001124   000000                        $GDDAT: .WORD   0                       ;CONTAINS 'GOOD' DATA
 001126   000000                        $BDDAT: .WORD   0                       ;CONTAINS 'BAD' DATA
 001130   000000  000000  000000                .WORD   0,0,0                   ;RESERVED--NOT TO BE USED
 001136   177560                        $TKS:   177560                          ;TTY KBD STATUS
 001140   177562                        $TKB:   177562                          ;TTY KBD BUFFER
 001142   177564                        $TPS:   177564                          ;TTY PRINTER STATUS REG. 1100
 001144   177566                        $TPB:   177566                          ;TTY PRINTER BUFFER REG. 1100
 001146      000                        $NULL:  .BYTE   0                       ;CONTAINS NULL CHARACTER FOR FILLS
 001147      002                        $FILLS: .BYTE   2                       ;CONTAINS # OF FILLER CHARACTERS REQUIRED
 001150      012                        $FILLC: .BYTE   12                      ;INSERT FILL CHARS. AFTER A "LINE FEED"
 001151      000                        $TPFLG: .BYTE   0                       ;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
 001152   000000                        $REGAD: .WORD   0                       ;CONTAINS THE 1100 FROM
                                                                                ;WHICH  ($REG0) WAS OBTAINED
 001154   000000                        $REG0:  .WORD   0                       ;CONTAINS (($REGAD)+0)
 001156   000000                        $REG1:  .WORD   0                       ;CONTAINS (($REGAD)+2)
 001160   000000                        $REG2:  .WORD   0                       ;CONTAINS (($REGAD)+4)
 001162   000000                        $REG3:  .WORD   0                       ;CONTAINS (($REGAD)+6)
 001164   000000                        $REG4:  .WORD   0                       ;CONTAINS (($REGAD)+10)
 001166   000000                        $REG5:  .WORD   0                       ;CONTAINS (($REGAD)+12)
 001170   000000                        $REG6:  .WORD   0                       ;CONTAINS (($REGAD)+14)
 001172   000000                        $REG7:  .WORD   0                       ;CONTAINS (($REGAD)+16)
 001174   000000                        $TMP0:  .WORD   0                       ;USER DEFINED
 001176   000000                        $TMP1:  .WORD   0                       ;USER DEFINED
 001200   000000                        $TMP2:  .WORD   0                       ;USER DEFINED
 001202   000000                        $TMP3:  .WORD   0                       ;USER DEFINED
 001204   000000                        $TMP4:  .WORD   0                       ;USER DEFINED
 001206   000000                        $TMP5:  .WORD   0                       ;USER DEFINED
 001210   000000                        $TMP6:  .WORD   0                       ;USER DEFINED
 001212   000000                        $TMP7:  .WORD   0                       ;USER DEFINED
```

```
388  001214  000000                STIMES: 0                       ;MAX. NUMBER OF ITERATIONS
389  001216  000000                SESCAPE:0                       ;ESCAPE ON ERROR 1100
390  001220  177607  000377        SBELL:  .ASCIZ  <207><377><377> ;CODE FOR BELL
391  001224     077                SQUES:  .ASCII  /?/             ;QUESTION MARK
392  001225     015                SCRLF:  .ASCII  <15>            ;CARRIAGE RETURN
393  001226  000012                SLF:    .ASCIZ  <12>            ;LINE FEED
394  001230  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 SITEMB, THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF SITEMB IS 0 THE ONLY PERTINENT DATA IS (SERRPC).
                                ;*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
                                ;
                                ;
001232                          SERRTB:
001232  007666                          EM1     ;"                                      LKS     LKS     "
001234  007747                          DH1     ;"PC      PS      SP      TEST#   WAS     S/B     "
001236  012030                          DT1     ;SERRPC,SREG7,SREG6,SREG5,LKS,SGDDAT
001240  000000                          0
                                ;
001242  012046                          EM2     ;"CLOCK FAILED TO INTERRUPT"
001244  012100                          DH2     ;"PC      PS      SP      TEST#   LKS     "
001246  012152                          DT2     ;SERRPC,SREG7,SREG6,SREG5,LKS
001250  000000                          0
                                ;
001252  012166                          EM3     ;"CLOCK INTERRUPTED WHEN THE PROCESSOR PRIORITY WAS TOO HIGH"
001254  012261                          DH3     ;"PC      PS      SP      TEST#   LKS     "
001256  012332                          DT3     ;SERRPC,SREG7,SREG6,SREG5,LKS
001260  000000                          0
                                ;
001262  012346                          EM4     ;"CLOCK GIVES UNEQUAL # OF PULSES OVER TWO EQUAL PERIODS OF TIME"
001264  012445                          DH4     ;"PC      PS      SP      TEST#   1ST     2ND"
001266  012602                          DT4     ;SERRPC,SREG7,SREG6,SREG5,SREG1,SREG0
001270  000000                          0
                                ;
001272  012620                          EM5     ;"LKS REGISTER RESPONDS TO ANOTHER ADDRESS"
001274  012671                          DH5     ;"PC      PS      SP      TEST#   ADDRESS"
001276  012742                          DT5     ;SERRPC,SREG7,SREG6,SREG5,SGDADR
001300  000000                          0
                                ;
001302  012756                          EM6     ;"A NO SACK TIMEOUT WAS OCCURED"
001304  011014                          DH6     ;"PC      PS      SP      TEST#   LKS     "
001306  011066                          DT6     ;SERRPC,SREG7,SREG6,SREG5,LKS
001310  000000                          0
                                ;
001312  011102                          EM7     ;"WRONG CONDITION CODES WERE PUT ONTO STACK BY INTERRUPT"
001314  011171                          DH7     ;"PC      PS      SP      TEST#   CC      CC"
001316  011250                          DT7     ;SERRPC,SREG7,SREG6,SREG5,BUF1,SGDDAT
001320  000000                          0
                                ;
001322  011266                          EM10    ;"WRONG PC PUT ONTO THE STACK BY AN INTERRUPT"
001324  011342                          DH10    ;"PC      PS      SP      TEST#  "
```

```
                001326  011426                          DT1B    ;SERRPC,SREG7,SREG6,SREG5,SUP2,SCODAT
                001330  000000                          0
                
                001332  011444                          EM11    ;"TRAPPED TRYING TO ACCESS LKS REGISTER"
                001334  011522                          DH11    ;"(PC)    (PS)    (SP)    TEST#"
                001336  011560                          DT11    ;SERRPC,SREG7,SREG6,SREG5
                001340  000000              0
                
                
                                            ;STARTUP CODE
                        001400              .=1400
                001400  012706  001000      KSTART: MOV     #1000,SP                ;INITIALIZE THE STACK SO WE CAN CALL THE TYPEOUT
                001404  013746  177776              MOV     PS,-(SP)                ;SAVE STATUS
                001410  004737  006132              JSR     PC,$TYPE                ;PRINTOUT STARTUP MESSAGE
                001414  007520                      $TMES                           ;ADDRESS OF MESSAGE "MAINDEC-11-DDKWA-A"
                001416                      START:
                001416  012706  001100              MOV     #SCMTAG,R6              ;FIRST LOCATION TO BE CLEARED
                001422  005026                      CLR     (R6)+                   ;CLEAR MEMORY LOCATION
                001424  022706  001136              CMP     #$TKS,R6                ;DONE?
                001430  001374                      BNE     .-6                     ;LOOP BACK IF NO
                001432  012706  001000              MOV     #1000,SP                        ;SETUP THE STACK POINTER
                001436  012737  005662  000020      MOV     #$SCOPE,@#IOTVEC        ;IOT VECTOR FOR SCOPE ROUTINE
                001444  012737  000340  000022      MOV     #340,@#IOTVEC+2         ;LEVEL 7
                001452  012737  007346  000030      MOV     #SERROR,@#EMTVEC        ;EMT VECTOR FOR ERROR ROUTINE
                001460  012737  000340  000032      MOV     #340,@#EMTVEC+2         ;LEVEL 7
                001466  012737  007240  000034      MOV     #$TRAP,@#TRAPVEC        ;TRAP VECTOR FOR TRAP CALLS
                001474  012737  000340  000036      MOV     #340,@#TRAPVEC+2        ;LEVEL 7
                001502  012737  007272  000024      MOV     #$PWRDN,@#PWRVEC        ;POWER FAILURE VECTOR
                001510  012737  000340  000026      MOV     #340,@#PWRVEC+2         ;LEVEL 7
                001516  013737  005500  005500      MOV     $ENDCT,$EOPCT           ;SETUP END-OF-PROGRAM COUNTER
                001524  112737  000001  001115      MOVB    #1,$ERMAX               ;ALLOW ONE ERROR PER TEST
                001532  012737  001532  001106      MOV     #.,SLPADR               ;INITIALIZE THE LOOP ADDRESS FOR SCOPE
                001540  005737  000042              TST     42                      ;LOADED BY A MONITOR
                001544  001401                      BEQ     T0001                   ;BR IF NO
                001546  000005                      RESET                           ;YES--GENERATE AN INIT
                
                
                
                                            .SBTTL  TEST THAT THE LKS CAN BE REFERENCED WITHOUT A BUS ERROR
                                            ;LKS ACCESS TEST
                001550  002004              T0001:  SCOPE
                001552  012737  001606  000004              MOV     #E0001,@#4      ;PREPARE FOR ADDRESSING THE LKS REGISTER, BAD HARDWARE
                001560  012737  000340  000006              MOV     #340,@#6        ;COULD CAUSE A TRAP TO 4
                001566  012737  001574  001106              MOV     #R0001,SLPADR   ;TIGHTEN UP THE SCOPE LOOP A BIT IN CASE OF AN ERROR
                001574  012706  001000      R0001:  MOV     #1000,SP        ;SETUP THE STACK POINTER IN CASE OF AN ERROR
                001600  005037  177546      I0001:  CLR     @#LKS           ;JUST REFERENCE LKS, DONT WORRY IF IT DIDNT CLEAR YET
                001604  000421                      BR      T0002           ;WE DIDNT TRAP IF WE REACH HERE, GO ON TO NEXT TEST
                001606  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
                001610  002004              T0002:  SCOPE
```

```
                001612  012737  005572  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
                001620  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
                001626  012737  001634  001106          MOV     #R0002,SLPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
                001634  000005                  R00021: RESET
                001636  012737  000200  001124          MOV     #200,SGDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
                001644  032737  000100  177546          BIT     #100,LKS        ;TEST THE INTERRUPT ENABLE BIT
                001652  001421                          BEQ     T0003
                001654  104001                  E00021: ERROR 1                 ;ERROR, CLOCK INTERRUPT ENABLE NOT CLEARED BY INIT

                                                ;SBTTL  TEST THAT START SETS CLOCK FLAG
                                                ;TEST THAT START SETS CLOCK FLAG
                001656  000004                  T0003:  SCOPE
                001660  012737  005572  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
                001666  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
                001674  012737  000200  001124          MOV     #200,SGDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
                001702  012737  001710  001106          MOV     #R0003,SLPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
                001710  000005                  R00031: RESET                   ;SHOULD SET THE  CLOCK FLAG
                001712  105737  177546                  TSTB    LKS             ;FIND OUT IF IT DID
                001716  100401                          BMI     T0004           ;GO ON TO THE NEXT TEST IF IT SET THE CLOCK FLAG
                001720  104001                  E00031: 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)
                001722  000004                  T0004:  SCOPE
                001724  012737  005572  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
                001732  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
                001740  012737  001746  001106          MOV     #R0004,SLPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
                001746  012737  000200  001124  R00041: MOV     #200,SGDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
                001754  005037  177546                  CLR     LKS             ;CLEAR THE CLOCK FLAG
                001760  005000                          CLR     R0              ;AND A TIMER LOCATION
                001762  105737  177546          A00041: TSTB    LKS             ;IS CLOCK FLAG SET
                001766  100403                          BMI     T0005
                001770  005200                          INC     R0              ;NO, INCREMENT COUNT @03 WAIT FOR SOMEMORE
                001772  001373                          BNE     A0004           ;WAIT SUFFICIENT AMOUNT OF TIME FOR CLOCK
                001774  104001                  E00041: 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
                001776  000004                  T0005:  SCOPE
                002000  012737  005572  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
                002006  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
                002014  012737  000100  001124          MOV     #100,SGDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
                002022  012737  002030  001106          MOV     #R0005,SLPADR   ;SETUP LOOP BACK ADDRESS IN CASE OF ERROR
                002030  012737  000340  177776          MOV     #340,PS         ;SET PRIORITY TO LEVEL 7, NO INTERRUPTS
                002036  005037  177546          R00051: CLR     LKS
                002042  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
                002044  105737  177546          A00051: TSTB    LKS             ;IS THE CLOCK FLAG SET?
                002050  100404                          BMI     B0005           ;IF SO, CONTINUE ON WITH THE TEST
```

```
                                                        INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
                                                        BNE     A00005          ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
                                                E00005: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
                                                        BR      T00006
                                                B00005:
                                                        MOV     #100,LKS        ;CLEAR CLOCK FLAG AND SET INTERRUPT ENABLE
                                                        BIT     #100,LKS        ;IS INTERRUPT ENABLE SET?
                                                        BNE     T00006
                                                E10005: ERROR 1                 ;ERROR INTERRUPT ENABLE NOT SET


                                                .SBTTL  TEST THAT INTERRUPT ENABLE BIT MAY BE CLEARED
                                                ;TEST THAT INTERRUPT ENABLE BIT MAY BE CLEARED
                                                T00201: SCOPE
                                                        MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
                                                        MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
                                                        MOV     #0,SCOPAT
                                                        MOV     #R00006,SLPADR  ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
                                                        MOV     #340,PS         ;SET PRIORITY LEVEL TO 7, NO INTERRUPTS
                                                R00201: CLR     LKS
                                                        CLR     R3              ;INITIALIZE A COUNTER LOCATION
                                                A00201: TSTB    LKS             ;IS THE CLOCK FLAG SET?
                                                        BMI     B00006          ;IF SO, CONTINUE ON WITH THE TEST
                                                        INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
                                                        BNE     A00006          ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
                                                E00201: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
                                                        BR      T00007
                                                B00201:
                                                        MOV     #100,LKS        ;CLEAR CLOCK FLAG AND SET INTERRUPT ENABLE
                                                        CLR     LKS             ;CLEAR INTERRUPT ENABLE
                                                        BIT     #100,LKS        ;TEST THE INTERRUPT ENABLE BIT
                                                        BEQ     T00007          ;IS INTERRUPT ENABLE CLEARED
                                                E10201: 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
                                                T00271: SCOPE
                                                        MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
                                                        MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
                                                        MOV     #300,SCOPAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
                                                        MOV     #R00207,SLPADR  ;INITIALIBE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
                                                        MOV     #D00207,100     ;SET UP VECTOR RETURN POINTER
                                                        MOV     #340,@#102      ;1 INTERRUPT IS ENOUGH
                                                R00271: MOV     #1000,SP        ;GET STACK READY FOR INTERRUPTS
                                                        MOV     #200,PS         ;SET PROCESSOR PRIORITY 4
                                                        CLR     LKS
                                                        CLR     R3              ;INITIALIBE A COUNTER LOCATION
                                                A00271: TSTB    LKS             ;IS THE CLOCK FLAG SET?
                                                        BMI     B00007          ;IF SO, CONTINUE ON WITH THE TEST
                                                        INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
                                                        BNE     A00007          ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
```

A2

```
000312  104001                  E00007: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
000314  000415                          BR      T00010
000316                          000007:
000316  012737  000100  177546          MOV     #100,LKS        ;ENABLE INTERRUPT
000324  005000                          CLR     R0
000326  005200                  C00007: INC     R0
000330  000240                          NOP                     ;STALL FOR TIME
000332  001375                          BNE     C00007          ;WAIT FOR INTERRUPT
000334  104002                  E100007: ERROR 2                ;ERROR, DIDNT GET INTERRUPT
000336  000404                          BR      T00010
000340  105737  177546          D00007: TSTB    LKS             ;ENTER HERE IF INTERRUPTED
000344  100401                          BMI     T00010
000346  104001                  E200007: 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 PRIORITY 5
000350  000004                  T00101: SCOPE
000352  012737  000572  000024          MOV     #TRAP0,#4       ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
000360  012737  000340  000006          MOV     #340,#6         ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
000366  012737  002410  001106          MOV     #R0010,SLPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
000374  012737  002470  000100          MOV     #D0010,100      ;SET UP VECTOR RETURN POINTER
000402  012737  000340  000102          MOV     #340,#0102      ;NO INTERRUPTS ALLOWED AFTER THE FIRST ONE
000410  005037  177546          R00101: CLR     LKS
000414  012737  000240  177776          MOV     #240,PS         ;SET PRIORITY 5
000422  012706  001000                  MOV     #1000,SP        ;INITIALIZE THE STACK POINTER
000426  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
000430  105737  177546          A00101: TSTB    LKS             ;IS THE CLOCK FLAG SET?
000434  100404                          BMI     B00010          ;IF SO, CONTINUE ON WITH THE TEST
000436  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
000440  001373                          BNE     A00101          ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS
000442  104001                  E00101: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
000444  000415                          BR      T00011
000446                          000101:
000446  012737  000100  177546          MOV     #100,LKS        ;ENABLE INTERRUPT
000454  005000                          CLR     R0
000456  005200                  C00101: INC     R0
000460  000240                          NOP                     ;STALL FOR SOME TIME
000462  001375                          BNE     C00010          ;WAIT FOR INTERRUPT
000464  104002                  E100101: ERROR 2                ;ERROR, INTERRUPT FAILED TO OCCUR
000466  000404                          BR      T00011
000470  105737  177546          D00101: TSTB    LKS             ;ENTER HERE IF INTERRUPTED
000474  100401                          BMI     T00011
000476  104001                  E200101: 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
000502  000004                  T00111: SCOPE
000502  012737  000572  000024          MOV     #TRAP0,#4       ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
000510  012737  000340  000006          MOV     #340,#6         ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
000516  012737  002524  001106          MOV     #R0011,SLPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
```

```
 665  002524  005037  177546          R00111: CLR     LKS
 666  002530  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
 667  002532  105737  177546          A00111: TSTB    LKS             ;IS THE CLOCK FLAG SET?
 668  002536  100404                          BMI     B0011           ;IF SO, CONTINUE ON WITH THE TEST
 669  002540  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
 670  002542  001373                          BNE     A0011           ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
 671  002544  104001                  E00111: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
 672  002546  000427                          BR      T0012
 673  002550                          B00111:
 674  002552  012706  001000                  MOV     #1000,SP        ;INITIALIZE THE STACK POINTER
 675  002554  012737  000300  177776          MOV     #300,PS         ;SET PRIORITY 6
 676  002562  012737  002012  000100          MOV     #E1011,100      ;SET UP VECTOR RETURN
 677  002570  012737  000100  177546          MOV     #100,LKS        ;ENABLE INTERRUPT
 678  002576  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
 679  002600  105737  177546          C00111: TSTB    LKS             ;IS THE CLOCK FLAG SET?
 680  002604  100404                          BMI     D0011           ;IF SO, CONTINUE ON WITH THE TEST
 681  002606  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
 682  002610  001373                          BNE     C0011           ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
 683  002612  104001                  E10111: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
 684  002614  000731                          BR      T0011
 685  002616                          D00111:
 686  002616  000240                          NOP
 687  002620  000240                          NOP                     ;GIVE CLOCK EXTRA TIME TO INTERRUPT
 688  002622  000401                          BR      T0012
 689  002624  104003                  E20011: ERROR 3                 ;ERROR, CLOCK INTERRUPTED WITHOUT HAVING PRIORITY
 690
 691
 692
 693                                  ;SBTTL  TEST THAT RESET SETS CLOCK FLAG
 694                                  ;TEST THAT RESET SETS CLOCK FLAG
 695  002626  000004          T00121: SCOPE
 696  002630  012737  000200  001124          MOV     #200,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
 697  002636  012737  005572  000004          MOV     #TRAP0,4        ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
 698  002644  012737  000340  000006          MOV     #340,6          ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
 699  002652  012737  002660  001106          MOV     #R0012,$LPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
 700  002660  005037  177546          R00121: CLR     LKS             ;CLEAR CLOCK FLAG
 701  002664  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
 702  002666  105737  177546          A00121: TSTB    LKS             ;IS THE CLOCK FLAG SET?
 703  002672  100404                          BMI     B0012           ;IF SO, CONTINUE ON WITH THE TEST
 704  002674  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
 705  002676  001373                          BNE     A0012           ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
 706  002700  104001                  E00121: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
 707  002702  000427                          BR      T0013
 708  002704                          B00121:
 709  002704  005037  177546                  CLR     LKS
 710  002710  000025                          RESET                   ;SHOULD SET CLOCK FLAG
 711  002712  105737  177546                  TSTB    LKS
 712  002716  100401                          BMI     T0013
 713  002720  104021                  E10012: ERROR 1                 ;ERROR, RESET DIDN'T SET CLOCK FLAG
 714
 715
 716
 717                                  ;SBTTL  TEST LINE CLOCK REPEATABILITY
 718                                  ;TEST LINE CLOCK REPEATABILITY
```

```
                                        ;MAKE SURE THAT OVER TWO EQUAL PERIODS OF TIME
                                        ;THE CLOCK PUTS OUT THE SAME NUMBER OF PULSES
002722  000024                  T00131: SCOPE
002724  005000                  R00131: CLR     R0              ;CLEAR 1ST TIME COUNT
002726  005001                          CLR     R1              ;CLEAR 1ST CLOCK COUNT
002730  012737  000340  177776          MOV     #340,PS         ;SET PRIORITY 7
002736  012737  002736  001106  R10131: MOV     #R1013,SLPADR   ;ERROR IN NEXT  FEW INSTRUCTIONS CAUSES A SHORT SCOPE LO
002744  005037  177546                  CLR     LKS
                                                                ;SYNC ON CLOCK FLAG A COUPLE OF TIMES
002750  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
002752  105737  177546          A00131: TSTB    LKS             ;IS THE CLOCK FLAG SET?
002756  100404                          BMI     B0013           ;IF SO, CONTINUE ON WITH THE TEST
002760  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
002762  001373                          BNE     A0013           ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
002764  104001                  E00131: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 2S MS
002766  000510                          BR      T0014
002770                          B00131:
002770  012737  002770  001106  R20131: MOV     #R2013,SLPADR   ;MAKE SCOPE LOOP SHORT IN CASE OF AN ERROR
002776  005037  177546                  CLR     LKS
003002  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
003004  105737  177546          C00131: TSTB    LKS             ;IS THE CLOCK FLAG SET?
003010  100404                          BMI     D0013           ;IF SO, CONTINUE ON WITH THE TEST
003012  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
003014  001373                          BNE     C0013           ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
003016  104001                  E10013: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 2S MS
003020  000473                          BR      T0014
003022                          D00131:
003022  005037  177546                  CLR     LKS
003026  105737  177546          F00131: TSTB    LKS             ;IS CLOCK FLAG SET
003032  100003                          BPL     G0013           ;NO
003034  005201                          INC     R1              ;+1 TO CLOCK COUNT
003036  005037  177546                  CLR     LKS             ;CLEAR CLOCK IF SET
003042  005200                  G00131: INC     R0              ;+1 TO TIME COUNT
003044  001370                          BNE     F0013           ;REPEAT UNTIL R0=0
003046  005000                          CLR     R0              ;CLEAR 2ND TIME COUNT
003050  005002                          CLR     R2              ;CLEAR 2ND CLOCK COUNT
003052  012737  003052  001106  R30131: MOV     #R3013,SLPADR   ;INSURE A SHORT SCOPE LOOP
003060  005037  177546                  CLR     LKS
                                                                ;SYNC ON CLOCK FLAG TWICE
003064  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
003066  105737  177546          H00131: TSTB    LKS             ;IS THE CLOCK FLAG SET?
003072  100404                          BMI     J0013           ;IF SO, CONTINUE ON WITH THE TEST
003074  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
003076  001373                          BNE     H0013           ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
003100  104001                  E20013: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 2S MS
003102  000442                          BR      T0014
003104                          J00131:
003104  012737  003134  001106  R40131: MOV     #R4013,SLPADR   ;INSURE A SHORT SCOPE LOOP
003112  005037  177546                  CLR     LKS
003116  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
003120  105737  177546          K00131: TSTB    LKS             ;IS THE CLOCK FLAG SET?
003124  100404                          BMI     L0013           ;IF SO, CONTINUE ON WITH THE TEST
003126  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
003130  001373                          BNE     K0013           ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
```

```
                                        E30013: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
                                                BR      T0014
                                        L00131:
                000132  012737  002724  001106        MOV     #R0013,SLPADR   ;MUST LOOP BACK TO BEGINING OF TEST IF ERROR COMES NOW
                000140  005037  177546                CLR     LKS
                000144  105737  177546        M00131: TSTB    LKS             ;IS CLOCK FLAG SET
                000150  100003                        BPL     N0013           ;NO
                000152  005202                        INC     R2              ;+1 TO CLOCK COUNT
                000154  005037  177546                CLR     LKS             ;CLEAR CLOCK IF SET
                000160  005200                N00131: INC     R0              ;+1 TO TIME COUNT
                000162  001370                        BNE     M0013           ;REPEAT UNTIL R0=0
                000164  020102                        CMP     R1,R2           ;IS 1ST CLOCK COUNT EQUAL TO 2ND CLOCK COUNT?
                000166  001406                        BEQ     T0014           ;YES
                000170  010137  001156        E40013: MOV     R1,$REG1        ;GET R1 READY FOR PRINTOUT
                000174  010237  001160                MOV     R2,$REG2        ;GET R2 READY FOR PRINTOUT
                000200  104024                        ERROR 4                 ;ERROR, CLOCK FLAG OCCURRED DIFFERENT
                000202  000240                        NOP                     ;NUMBER OF TIMES IN EQUAL PERIODS


                                        .SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
                                        ;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
                000210  000004        T00141: SCOPE
                000212  005037  001124                CLR     $GDDAT
                000216  012737  003254  001106        MOV     #R0014,SLPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
                000224  012737  157546  001120        MOV     #157546,$GDADR  ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 13 CLEAR
                000232  012737  003272  000004        MOV     #A0014,4        ;SETUP VECTOR IN CASE IT IS NONEXISTANT
                000240  012737  000340  000006        MOV     #340,6
                000246  012737  000340  177776        MOV     #340,@#PS       ;NO INTERRUPTS NOW
                000254  012706  001000        R00141: MOV     #1000,SP        ;SETUP THE STACK
                000260  005037  177546                CLR     LKS
                000264  012777  000100  175626  I00141: MOV    #100,@$GDADR    ;SET THE "CLOSE" ADDRESS TO = 100
                000272  032737  000100  177546  A00141: BIT    #100,LKS        ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
                000300  001401                        BEQ     B0014
                000302  104005                E20141: ERROR 5                 ;IT AFFECTED "LKS"  ==  ERROR
                000304  005037  177546        B00141: CLR     LKS
                000310  012737  003322  000004        MOV     #T0015  ,4
                000316  005077  175576                CLR     @$GDADR         ;CLEAR OUT THE "CLOSE" ADDRESS


                                        .SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
                                        ;LINE CLOCK REGISTER ADDRESSING TEST
                000322  000004        T00151: SCOPE
                000324  012737  003404  000004        MOV     #A0015,4        ;SETUP VECTOR IN CASE IT IS NONEXISTANT
                000332  012737  000340  000006        MOV     #340,6
                000340  005037  001124                CLR     $GDDAT
                000344  012737  003366  001106        MOV     #R0015,SLPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
                000352  012737  177146  001120        MOV     #177146,$GDADR  ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 0 CLEAR
                000360  012737  000340  177776        MOV     #340,@#PS       ;NO INTERRUPTS NOW
                000366  012706  001000        R00151: MOV     #1000,SP        ;SETUP THE STACK
```

```
003372 005037 177546                          CLR    LKS
003376 012777 000100 175514         I00015:   MOV    #100,@SGDADR      ;SET THE "CLOSE" ADDRESS TO = 100
003404 032737 000100 177546         A00015:   BIT    #100,LKS          ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
003412 001401                                 BEQ    B0015
003414 104005                       E00015:   ERROR 5                  ;IT AFFECTED "LKS"  --  ERROR
003416 005037 177546                B00015:   CLR    LKS
003422 012737 003434 000004                   MOV    #T0016  ,4
003430 005077 175464                          CLR    @SGDADR           ;CLEAR OUT THE "CLOSE" ADDRESS

                                    .SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
                                    ;LINE CLOCK REGISTER ADDRESSING TEST
003434 000004                       T00016:   SCOPE
003436 012737 003516 000004                   MOV    #A0016,4          ;SETUP VECTOR IN CASE IT IS NONEXISTANT
003444 012737 000340 000006                   MOV    #340,6
003452 005037 001124                          CLR    SGDDAT
003456 012737 003500 001106                   MOV    #R0016,SLPADR     ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
003464 012737 177446 001120                   MOV    #177446,SGDADR    ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 6 CLEAR
003472 012737 000340 177776                   MOV    #340,@#PS         ;NO INTERRUPTS NOW
003500 012706 001000                R00016:   MOV    #1000,SP          ;SETUP THE STACK
003504 005037 177546                          CLR    LKS
003510 012777 000100 175402         I00016:   MOV    #100,@SGDADR      ;SET THE "CLOSE" ADDRESS TO = 100
003516 032737 000100 177546         A00016:   BIT    #100,LKS          ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
003524 001401                                 BEQ    B0016
003526 104005                       E00016:   ERROR 5                  ;IT AFFECTED "LKS"  --  ERROR
003530 005037 177546                B00016:   CLR    LKS
003534 012737 003546 000004                   MOV    #T0017  ,4
003542 005077 175352                          CLR    @SGDADR           ;CLEAR OUT THE "CLOSE" ADDRESS

                                    .SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
                                    ;LINE CLOCK REGISTER ADDRESSING TEST
003546 000004                       T00017:   SCOPE
003550 012737 003630 000004                   MOV    #A0017,4          ;SETUP VECTOR IN CASE IT IS NONEXISTANT
003556 012737 000340 000006                   MOV    #340,6
003564 005037 001124                          CLR    SGDDAT
003570 012737 003612 001106                   MOV    #R0017,SLPADR     ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
003576 012737 177556 001120                   MOV    #177556,SGDADR    ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 3 SET
003604 012737 000340 177776                   MOV    #340,@#PS         ;NO INTERRUPTS NOW
003612 012706 001000                R00017:   MOV    #1000,SP          ;SETUP THE STACK
003616 005037 177546                          CLR    LKS
003622 012777 000100 175270         I00017:   MOV    #100,@SGDADR      ;SET THE "CLOSE" ADDRESS TO = 100
003630 032737 000100 177546         A00017:   BIT    #100,LKS          ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
003636 001401                                 BEQ    B0017
003640 104005                       E00017:   ERROR 5                  ;IT AFFECTED "LKS"  --  ERROR
003642 005037 177546                B00017:   CLR    LKS
003646 012737 003660 000004                   MOV    #T0020  ,4
003654 005077 175240                          CLR    @SGDADR           ;CLEAR OUT THE "CLOSE" ADDRESS

                                    .SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
```

F2

```
                                                ;LINE CLOCK REGISTER ADDRESSING TEST
003660  000004                                  T00201: SCOPE
003662  012737  003742  000004                          MOV     #A00201,4       ;SETUP VECTOR IN CASE IT IS NONEXISTANT
003670  012737  000340  000006                          MOV     #340,6
003676  005037  001124                                  CLR     SGDDAT
003702  012737  003724  001106                          MOV     #R00201,SLPADR  ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
003710  012737  177566  001120                          MOV     #177566,SGDADR  ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 4 SET
003716  012737  000340  177776                          MOV     #340,@#PS       ;NO INTERRUPTS NOW
003724  012706  001000                        R00201:   MOV     #1000,SP        ;SETUP THE STACK
003730  005037  177546                                  CLR     LKS
003734  012777  000100  175156                I00201:   MOV     #100,@SGDADR    ;SET THE "CLOSE" ADDRESS TO = 100
003742  032737  000100  177546                A00201:   BIT     #100,LKS        ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
003750  001401                                          BEQ     B00201
003752  104005                                E00201:   ERROR 5                 ;IT AFFECTED "LKS"  --  ERROR
003754  005037  177546                        B00201:   CLR     LKS
003760  012737  003772  000004                          MOV     #T00211 ,4
003766  005077  175120                                  CLR     @SGDADR         ;CLEAR OUT THE "CLOSE" ADDRESS


                                                ;.SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
                                                ;LINE CLOCK REGISTER ADDRESSING TEST
004000  000004                                  T00211: SCOPE
004002  012737  004034  000004                          MOV     #A00211,4       ;SETUP VECTOR IN CASE IT IS NONEXISTANT
004010  012737  000340  000006                          MOV     #340,6
004016  005037  001124                                  CLR     SGDDAT
004022  012737  004036  001106                          MOV     #R00211,SLPADR  ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
004030  012737  177746  001120                          MOV     #177746,SGDADR  ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 7 SET
004036  012737  000340  177776                          MOV     #340,@#PS       ;NO INTERRUPTS NOW
004044  012706  001000                        R00211:   MOV     #1000,SP        ;SETUP THE STACK
004050  005037  177546                                  CLR     LKS
004054  012777  000100  175044                I00211:   MOV     #100,@SGDADR    ;SET THE "CLOSE" ADDRESS TO = 100
004062  032737  000100  177546                A00211:   BIT     #100,LKS        ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
004070  001401                                          BEQ     B00211
004072  104005                                E00211:   ERROR 5                 ;IT AFFECTED "LKS"  --  ERROR
004074  005037  177546                        B00211:   CLR     LKS
004100  012737  004136  000004                          MOV     #T00231 ,4
004106  005077  175014                                  CLR     @SGDADR         ;CLEAR OUT THE "CLOSE" ADDRESS


004114  000004                                  T00221: SCOPE


                                                ;.SBTTL  CLOCK FLAG BIT TEST
                                                ;CLOCK FLAG BIT TEST
004120  000004                                  T00231: SCOPE
004122  012737  005572  000004                          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
004130  012737  000340  000006                          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
004136  012737  000200  001124                          MOV     #200,SGDDAT
004144  012737  004140  001106                          MOV     #R00231,SLPADR  ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERR.
004152  005037  177546                        R00231:   CLR     LKS
004156  005003                                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
```

```
                                                A0023:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
                                                        BMI     B0023           ;IF SO, CONTINUE ON WITH THE TEST
                                                        INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
                                                        BNE     A0023           ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
                                                E0023:  ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
                                                        BR      T0024
                                                B0023:
                                                I0023:  MOV     #200,LKS        ;MOVE A 1 INTO THE CLOCK FLAG BIT
                                                        CMP     LKS,$GDDAT      ;SHOULD NOT AFFECT THE FLAG BIT
                                                        BEQ     T0024
                                                E10023: ERROR 1                 ;CLOCK FLAG DID NOT CLEAR


                                                .SBTTL  INTERRUPT TEST
                                                T0024:  SCOPE
                                                        MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
                                                        MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
                                                        MOV     #200,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
                                                        MOV     #R0024,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
                                                        MOV     #E0024,100
                                                        CLR     LKS             ;ALLOW CLOCK INTERRUPTS
                                                R0024:  MOV     #340,@#PS       ;NO INTERRUPTS NOW
                                                        MOV     #300,LKS
                                                A0024:  INC     @0              ;WAIT FOR 20+ MS
                                                        BNE     A0024           ;LOOP BACK IF NOT DONE WAITING
                                                        RESET                   ;RESET SHOULD CLEAR INTERRUPT ENABLE
                                                        CMP     $GDDAT,LKS      ;AND LEAVE THE CLOCK FLAG SET
                                                        BEQ     T0025           ;GO ON TO THE NEXT TEST IF IT DID
                                                E0024:  ERROR 1                 ;RESET SET INTERRUPT ENABLE OR CLEARED CLOCK FLAG


                                                .SBTTL  NO SACK TIMEOUT TEST
                                                ;NO SACK TIMEOUT TEST
                                                T0025:  SCOPE
                                                        MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
                                                        MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
                                                        MOV     #300,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
                                                        MOV     #R0025,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
                                                        MOV     #340,102        ;NO INTERRUPTS AFTER THE FIRST ONE
                                                        MOV     #C0025,100
                                                R0025:  CLR     LKS
                                                        MOV     #340,PS
                                                        CLR     R3              ;INITIALIZE A COUNTER LOCATION
                                                A0025:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
                                                        BMI     B0025           ;IF SO, CONTINUE ON WITH THE TEST
                                                        INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
                                                        BNE     A0025           ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
                                                E0025:  ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
                                                        BR      T0026
                                                B0025:
                                                        CLR     PS              ;MAKE ALL INTERRUPTS OK
                                                        MOV     #100,LKS        ;ENABLE CLOCK INTERRUPTS
```

```
  989  004420  000001                          WAIT                    ;THE ONLY WAY TO LEAVE HERE WITHOUT ERROR IS TO INTERRUP
  990  004422  104006                E10025:   ERROR 6                 ;IF IT ERROR IS HERE ITS BECAUSE OF A "NO-SACK" TIMEOUT
  991  004424  000405                          BR      T0026
  992  004426  022737  000300 177546 C00025:   CMP     #300,LKS
  993  004434  001401                          BEQ     T0026           ;FIND OUT WHAT THE INTERRUPT DID TO THE CLOCK STATUS REG
  994  004436  104001                E20025:   ERROR 1                 ;IT CLEARED THE INTERRUPT ENABLE OR THE FLAG BIT
  995
  996
  997
  998                                         .SBTTL  RESET TEST
  999                                         ;RESET TEST
 1000  004440  000004                T00261:   SCOPE
 1001  004442  012737  005572 000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
 1002  004450  012737  000340 000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
 1003  004456  012737  000200 001124          MOV     #200,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
 1004  004464  012737  004514 001106          MOV     #R0020,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
 1005  004472  012737  000340 177776          MOV     #340,@#PS       ;NO INTERRUPTS NOW
 1006  004500  012737  004552 000100          MOV     #E0020,100
 1007  004506  012737  000140 000102          MOV     #140,102        ;SETUP STATUS FOR AFTER THE INTERRUPT
 1008  004514  005037  177546       R00261:   CLR     LKS
 1009  004520  012706  001300                  MOV     #1000,SP        ;SETUP THE STACK
 1010  004524  012737  000100 177546          MOV     #100,LKS        ;SET INTERRUPT ENABLE BIT NOW
 1011  004532  005227  000000       A00261:   INC     #0              ;WAIT FOR CLOCK FLAG TO SET
 1012  004536  001375                          BNE     A0026
 1013  004540  000005                I00261:   RESET                   ;RESET SHOULD NOT CLEAR THE FLAG
 1014  004542  023737  177546 001124          CMP     LKS,$GDDAT      ;FIND OUT IF ID DIT DID OR NOT
 1015  004550  001401                          BEQ     T0027
 1016  004552  104001                E00261:   ERROR 1                 ;RESET DID NOT INITIALIZE THE LKS WORD CORRECTLY
 1017
 1018
 1019
 1020                                         .SBTTL  CLOCK FLAG BIT TEST
 1021                                         ;CLOCK FLAG BIT TEST
 1022                                         ;MAKE SURE IT DOESNT CLEAR WHEN YOU TRY TO SET IT VIA A 'MOV' INSTRUCTION
 1023  004554  000004                T00271:   SCOPE
 1024  004556  012737  005572 000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
 1025  004564  012737  000340 000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
 1026  004572  012737  000200 001124          MOV     #200,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
 1027  004600  012737  004620 001106          MOV     #R0027,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
 1028  004606  005037  000102                  CLR     102
 1029  004612  012737  004676 000100          MOV     #T0030  ,100
 1030  004620  005037  177546       R00271:   CLR     LKS
 1031  004624  012737  000340 177776          MOV     #340,PS
 1032  004632  012706  001000                  MOV     #1000,SP        ;SETUP THE STACK
 1033  004636  005037  001230                  CLR     WORD            ;SETUP A COUNTER LOCATION TO = 0
 1034  004642  005237  001230       A00271:   INC     WORD
 1035  004646  001375                          BNE     A0027           ;WASTE TIME LOOPING UNTIL THE COUNTER REACHES 0
 1036  004650  012737  000300 177546          MOV     #300,LKS        ;SET INTERRUPT ENABLE AND TRY TO SET THE CLOCK FLAG
 1037  004656  012737  000200 177546          MOV     #200,LKS        ;TRY TO SET IT AGAIN
 1038  004664  023737  177546 001124          CMP     LKS,$GDDAT      ;DID THE CLOCK FLAG STAY SET?
 1039  004672  001401                          BEQ     T0030           ;IF NOT GO ON TO THE NEXT TEST
 1040  004674  104001                E00271:   ERROR 1                 ;ERROR = MOVED A '1' INTO THE CLOCK FLAG BIT AND IT STAY
 1041
 1042
```

```
                                        .SBTTL  CLOCK FLAG AFTER INTERRUPT TEST
                                        ;SEE IF AN INTERRUPT CLEARS THE CLOCK FLAG
 004676  002004                         T00301: SCOPE
 004700  012737  005572  000004                 MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
 004706  012737  000340  000006                 MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
 004714  005037  177546                         CLR     LKS             ;NO CLOCK INTERRUPTS BEFORE WE ARE READY
 004720  012737  000100  001124                 MOV     #100,SGDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
 004726  012737  004734  001106                 MOV     #R0030,SLPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
 004734  012737  004776  000100  R00301:        MOV     #A0030,100      ;SETUP CLOCK INTERRUPT VECTOR
 004742  005037  000102                         CLR     102             ;PRIORITY LEVEL WILL ALLOW FURTHER INTERRUPTS
 004746  005037  177776                         CLR     PS
 004752  012706  001000                         MOV     #1000,SP        ;SETUP THE STACK
 004756  005037  177546                         CLR     LKS
 004762  105037  001230                         CLRB    WORD            ;CLEAR OUT A COUNTER LOCATION
 004766  012737  000100  177546                 MOV     #100,LKS        ;ENABLE CLOCK INTERRUPTS NOW
 004774  000001                                 WAIT                    ;WAIT FOR AN INTERRUPT
 004776  012737  005024  000100  A00301:        MOV     #E0030,100      ;ERROR IF WE INTERRUPT AGAIN
 005004  005037  177776                         CLR     PS              ;LET INTERRUPTS HAPPEN NOW
 005010  105237  001230          B00301:        INCB    WORD            ;DO A NOTHING LOOP FOR A VERY SHORT PERIOD OF TIME
 005014  001375                                 BNE     B0030           ;WE SHOULD INCREMENT TO 0 LONG BEFORE AN INTERRUPT COMES
 005016  005037  177546                         CLR     LKS
 005022  000401                                 BR      T0031
 005024  104001                 E00301:        ERROR 1                 ;INTERRUPT DID NOT CLEAR THE CLOCK FLAG


                                        .SBTTL  NO INTERRUT AT PRIORITY 7 TEST
                                        ;TEST THAT CLOCK WILL NOT INTERRUPT WITH PROCESSR AT PRIORITY 7
 005026  000004                         T00311: SCOPE
 005030  012737  005572  000004                 MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
 005036  012737  000340  000006                 MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
 005044  012737  005052  001106                 MOV     #R0031,SLPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
 005052  005037  177546          R00311:        CLR     LKS
 005056  005003                                 CLR     R3              ;INITIALIZE A COUNTER LOCATION
 005060  105737  177546          A00311:        TSTB    LKS             ;IS THE CLOCK FLAG SET?
 005064  100404                                 BMI     B0031           ;IF SO, CONTINUE ON WITH THE TEST
 005066  005203                                 INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
 005070  001373                                 BNE     A0031           ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
 005072  104001                 E00311:        ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
 005074  000427                                 BR      T0032
 005076                         B00311:
 005076  012706  001000                         MOV     #1000,SP        ;INITIALIZE THE STACK POINTER
 005102  012737  000340  177776                 MOV     #340,PS         ;SET PRIORITY 7
 005110  012737  005072  000100                 MOV     #E0031,100      ;SET UP VECTOR RETURN
 005116  012737  000100  177546                 MOV     #100,LKS        ;ENABLE INTERRUPT
 005124  005003                                 CLR     R3              ;INITIALIZE A COUNTER LOCATION
 005126  105737  177546          C00311:        TSTB    LKS             ;IS THE CLOCK FLAG SET?
 005132  100404                                 BMI     D0031           ;IF SO, CONTINUE ON WITH THE TEST
 005134  005203                                 INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
 005136  001373                                 BNE     C0031           ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
 005140  104001                 E100311:       ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
 005142  000404                                 BR      T0032
 005144                         D00311:
```

J2

```
1097  005144  012240                          NOP
1098  005146  012240                          NOP                     ;GIVE CLOCK EXTRA TIME TO INTERRUPT
1099  005150  000401                          BR      T0032
1100  005152  104003                  E20031: ERROR 3                 ;ERROR, CLOCK INTERRUPTED WITHOUT HAVING PRIORITY
1101
1102
1103
1104                                   ;SBTTL  CC PUSH TEST FOR CLOCK INTERRUPTS
1105                                   ;TEST THAT CLOCK INTERRUPT PUSHES CONDITION CODES ONTO STACK
1106  005154  000004                  T0032:  SCOPE
1107  005156  012737  005572  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
1108  005164  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
1109  005172  012737  005200  001106          MOV     #R0032,SLPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
1110  005200  005037  177546          R0032:  CLR     LKS
1111  005204  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
1112  005206  105737  177546          A0032:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
1113  005212  100404                          BMI     B0032           ;IF SO, CONTINUE ON WITH THE TEST
1114  005214  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
1115  005216  001373                          BNE     A0032           ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
1116  005220  104001                  E0032:  ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
1117  005222  000432                          BR      T0033
1118  005224                          B0032:
1119  005224  012706  001000                  MOV     #1000,SP        ;INITIALIZE THE STACK POINTER
1120  005230  005037  000776                  CLR     BUF1
1121  005234  005037  000774                  CLR     BUF2
1122  005240  012737  005266  000100          MOV     #C0032,100      ;SET UP VECTOR RETURN
1123  005246  012737  000100  177546          MOV     #100,LKS        ;ENABLE INTERRUPT
1124  005254  012737  000200  177776          MOV     #200,PS         ;SET PRIORITY 4
1125  005262  000277                          .SEC!SEV!SEZ!SEN        ;SET ALL CONDITION CODES
1126  005264  000001                          WAIT                    ;WAIT FOR INTERRUPT
1127  005266  022737  000217  000776  C0032:  CMP     #217,BUF1
1128  005274  001405                          BEQ     T0033
1129  005276  012737  000017  001124          MOV     #17,SGDDAT
1130  005304  104007                          ERROR 7                 ;ERROR DID NOT PUSH CORRECT PS ONTO STACK
1131  005306  000722                          BR      T0032
1132
1133
1134
1135                                   ;SBTTL  PC PUSH TEST FOR CLOCK INTERRUPTS
1136                                   ;TEST THAT CLOCK INTERRUPT PUSHES THE PROGRAM COUNTER ONTO STACK
1137  005310  000004                  T0033:  SCOPE
1138  005312  012737  005572  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
1139  005320  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
1140  005326  012737  005334  001106          MOV     #R0033,SLPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
1141  005334  005037  177546          R0033:  CLR     LKS
1142  005340  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
1143  005342  105737  177546          A0033:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
1144  005346  100404                          BMI     B0033           ;IF SO, CONTINUE ON WITH THE TEST
1145  005350  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
1146  005352  001373                          BNE     A0033           ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
1147  005354  104001                  E0033:  ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
1148  005356  000431                          BR      T0034
1149  005362                          B0033:
1150  005360  012706  001000                  MOV     #1000,SP        ;INITIALIZE THE STACK POINTER
```

```
005364  005037  000776                  CLR     BUF1
005370  005037  000774                  CLR     BUF2
005374  012737  005422  000100          MOV     #C0033,100              ;SET UP VECTOR RETURN
005402  012737  000100  177546          MOV     #100,LKS                ;ENABLE INTERRUPT
005410  012737  000200  177776          MOV     #200,PS                 ;SET PRIORITY 4
005416  000277                          +SEC!SEV!SEZ!SEN                ;SET ALL CONDITION CODES
005420  000001                          WAIT                            ;WAIT FOR INTERRUPT
005422  022737  005422  000774  C0033:  CMP     #C0033,BUF2
005430  001404                          BEQ     T0034
005432  012737  005422  001124  E10033: MOV     #C0033,SGDDAT
005440  104010                          ERROR 10                        ;ERROR, DID NOT PUSH CORRECT PC ONTO STACK



                                .SBTTL  END OF PASS INDICATING
005442  000004                  T0034:  SCOPE
005444  005037  177546                  CLR     LKS                     ;TURN THE CLOCK OFF
005450  000005                          RESET                           ;TURN EVERYTHING OFF

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

                                .SBTTL  END OF PASS ROUTINE

                                ;*INCREMENT THE PASS NUMBER (SPASS)
                                ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                                ;*IF THERES A MONITOR GO TO IT
                                ;*IF THERE ISN'T JUMP TO T0001

005452                          SEOP:
005452  000004                          SCOPE
005454  005037  001102                  CLR     STSTNM                          ;ZERO THE TEST NUMBER
005460  005037  001214                  CLR     STIMES                          ;ZERO THE NUMBER OF ITERATIONS
005464  005237  001100                  INC     SPASS                           ;INCREMENT THE PASS NUMBER
005470  042737  100000  001100          BIC     #100000,SPASS                   ;DON'T ALLOW A NEG. NUMBER
005476  005327                          DEC     (PC)+                           ;LOOP?
005500  000001                  SEOPCT: .WORD   1
005502  003021                          BGT     SDOAGN                          ;YES
005504  012737                          MOV     (PC)+,@(PC)+                    ;RESTORE COUNTER
005506  000001                  SENDCT: .WORD   1
005510  005500                          SEOPCT
005512  104400  005552                  TYPE    ,SENDMG                         ;TYPE "END PASS #"
005516  013746  001100                  MOV     SPASS,-(SP)                     ;SAVE SPASS FOR TYPEOUT
005522  104410                          TYPDS                                   ;GO TYPE--DECIMAL ASCII WITH SIGN
005524  104400  005567                  TYPE    ,SENULL                         ;TYPE A NULL CHARACTER
005530  013700  000042          SGET42: MOV     @#42,R0                         ;GET MONITOR ADDRESS
005534  001404                          BEQ     SDOAGN                          ;IF NONE
005536  004710                  SENDAD: JSR     PC,(R0)                         ;GO TO MONITOR
005540  000240                          NOP                                     ;SAVE ROOM
005542  000240                          NOP                                     ;FOR
005544  000240                          NOP                                     ;ACT11
005546  000137  001550          SDOAGN: JMP     @#T0001                         ;RETURN
005552  005015  047105  020104  SENDMG: .ASCIZ  <15><12>/END PASS #/
005560  040520  051523  021440
005566     000
```

L1

L2

```
1205  005567  000377  000377  000  $ENULL: .BYTE   -1,-1,0                 ;NULL CHARACTER STRING
1206  005572  005046                TRAP3:  CLR     -(SP)
1207  005574  004737  006132              JSR     PC,$TYPE                ;PRINTOUT"TRAPPED TO 4 FROM "
1208  005600  007575                      TRPMES                          ;ADDRESS OF THE MESSAGE
1209  005602  012737  007575  017400      MOV     #TRPMES,ERRST           ;STORE POINTER TO TRAP MESSAGE IN CORE
1210  005610  011646                      MOV     (SP),-(SP)              ;GET THE ADDRESS WHERE THE TRAP OCCURED
1211  005612  162716  000002              SUB     #2,(SP)                 ;MAKE IT RIGHT
1212  005616  104402                      TYPOC                           ;TYPE OUT ADDRESS IN OCTAL
1213  005620  104400  001225              TYPE    ,$CRLF                  ;PRINTOUT A CARRIAGE RETURN-LINE FEED
1214  005624  011637  017402              MOV     (SP),ERRST+2            ;STORE ADDRESS WHERE TRAP OCCURED
1215  005630  005037  017404              CLR     ERRST+4
1216  005634  005046                      CLR     -(SP)
1217  005636  004737  006132              JSR     PC,$TYPE                ;PRINTOUT RESTARTING MESSAGE
1218  005642  007636                      TRPM2S                          ;ADDRESS OF RESTART MESSAGE
1219  005644  000240                      NOP
1220  005646  000240                      NOP
1221  005650  000240                      NOP
1222  005652  000240                      NOP
1223  005654  000005                      RESET
1224  005656  000137  001416              JMP     START
1225                                ;;*****************************************************************
1226
1227                                .SBTTL  SCOPE HANDLER ROUTINE
1228
1229                                ;-SW14=1        LOOP ON TEST
1230                                ;-SW11=1        INHIBIT ITERATIONS
1231                                ;-SW09=1        LOOP ON ERROR
1232                                ;-SW08=1        LOOP ON TEST IN SWR<7:0>
1233                                ;-THE TEST NUMBER (STSTNM) IS INCREMENTED AND DISPLAYED IN DISPLAY<7:0>
1234                                ;-AND THE ERROR FLAG (SERFLG) IS DISPLAYED IN DISPLAY<15:08>
1235
1236  005662                        $SCOPE:
1237  005662  000240                      NOP
1238  005664  006137  177570              ROL     @#SWR                   ;LOOP ON PRESENT TEST?
1239  005670  100511                      BMI     $OVER                   ;YES IF SW14=1
1240                                ;#####START OF CODE FOR THE XOR TESTER#####
1241  005672  000416                $XTSTR: BR      6$                      ;IF RUNNING ON THE "XOR" TESTER CHANGE
1242                                                                        ;THIS INSTRUCTION TO A "NOP" (NOP=240)
1243  005674  013746  000004              MOV     @#ERRVEC,-(SP)          ;SAVE THE CONTENTS OF THE ERROR VECTOR
1244  005700  012737  005720  000004      MOV     #5$,@#ERRVEC            ;SET FOR TIMEOUT
1245  005706  005737  177060              TST     @#177060                ;TIME OUT ON XOR?
1246  005712  012637  000004              MOV     (SP)+,@#ERRVEC          ;RESTORE THE ERROR VECTOR
1247  005716  000463                      BR      $SVLAD                  ;GO TO THE NEXT TEST
1248  005720  022626                5$:   CMP     (SP)+,(SP)+             ;CLEAR THE STACK AFTER A TIME OUT
1249  005722  012637  000004              MOV     (SP)+,@#ERRVEC          ;RESTORE THE ERROR VECTOR
1250  005726  000423                      BR      7$                      ;LOOP ON THE PRESENT TEST
1251  005730                        6$:   #####END OF CODE FOR THE XOR TESTER#####
1252  005730  032737  000400  177570      BIT     #SW08,@#SWR             ;LOOP ON SPEC. TEST?
1253  005736  001404                      BEQ     2$                      ;BR IF NO
1254  005740  123737  177570  001102      CMPB    @#SWR,STSTNM            ;ON THE RIGHT TEST?     SWR<7:0>
1255  005746  001462                      BEQ     $OVER                   ;BR IF YES
1256  005750  105737  001103        2$:   TSTB    $ERFLG                  ;HAS AN ERROR OCCURRED?
1257  005754  001421                      BEQ     3$                      ;BR IF NO
1258  005756  123737  001115  001103      CMPB    $ERMAX,$ERFLG           ;MAX. ERRORS FOR THIS TEST OCCURRED?
```

```
1259  000764  101015                        BHI     3$                      ;BR IF NO
1260  000766  032737  001000  177570        BIT     #SW09,@#SWR             ;LOOP ON ERROR?
1261  000774  001404                        BEQ     4$                      ;BR IF NO
1262  000776  013737  001110  001106  7$:   MOV     $LPERR,$LPADR           ;SET LOOP ADDRESS TO LAST SCOPE
1263  001004  000443                        BR      $OVER
1264  001006  105037  001103          4$:   CLRB    $ERFLG                  ;ZERO THE ERROR FLAG
1265  001012  005037  001214                CLR     $TIMES                  ;CLEAR THE NUMBER OF ITERATIONS TO MAKE
1266  001016  000415                        BR      1$                      ;ESCAPE TO THE NEXT TEST
1267  001022  032737  004000  177570  3$:   BIT     #SW11,@#SWR             ;INHIBIT ITERATIONS?
1268  001026  001011                        BNE     1$                      ;BR IF YES
1269  001030  005737  001100                TST     $PASS                   ;IF FIRST PASS OF PROGRAM
1270  001034  001406                        BEQ     1$                      ;       INHIBIT ITERATIONS
1271  001036  005237  001104                INC     $ICNT                   ;INCREMENT ITERATION COUNT
1272  001042  023737  001214  001104        CMP     $TIMES,$ICNT            ;CHECK THE NUMBER OF ITERATIONS MADE
1273  001050  002021                        BGE     $OVER                   ;BR IF MORE ITERATION REQUIRED
1274  001052  012737  000001  001104  1$:   MOV     #1,$ICNT                ;REINITIALIZE THE ITERATION COUNTER
1275  001060  013737  000130  001214        MOV     $MXCNT,$TIMES           ;SET NUMBER OF ITERATIONS TO DO
1276  001066  105237  001102          $SVLAD: INCB    $TSTNM                  ;COUNT TEST NUMBERS
1277  001072  011637  001106                MOV     (SP),$LPADR             ;SAVE SCOPE LOOP ADDRESS
1278  001076  011637  001110                MOV     (SP),$LPERR             ;SAVE ERROR LOOP ADDRESS
1279  001102  005037  001216                CLR     $ESCAPE                 ;CLEAR THE ESCAPE FROM ERROR ADDRESS
1280  001106  112737  000001  001115        MOVB    #1,$ERMAX               ;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
1281  001114  013737  001102  177570  $OVER: MOV     $TSTNM,@#DISPLAY        ;DISPLAY TEST NUMBER
1282  001122  013716  001100                MOV     $LPADR,(SP)             ;FUDGE RETURN ADDRESS
1283  001126  000002                        RTI                             ;FIXES PS
1284  001130  000010                $MXCNT: 10                              ;MAX, NUMBER OF ITERATIONS
1285                                        ;;*******************************************************************
1286
1287                                        .SBTTL  TYPE ROUTINE
1288
1289                                        ;*ROUTINE TO TYPE ASCIZ MESSAGE, MESSAGE MUST TERMINATE WITH A 0 BYTE.
1290                                        ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
1291                                        ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
1292                                        ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
1293                                        ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
1294                                        ;*
1295                                        ;*CALL:
1296                                        ;*1) USING A TRAP INSTRUCTION
1297                                        ;*      TYPE    ,MESADR                 ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
1298                                        ;*OR
1299                                        ;*      TYPE
1300                                        ;*      MESADR
1301                                        ;*
1302                                        ;*2) USING A JSR INSTRUCTION
1303                                        ;*      MOV     PS,-(SP)                ;PUSH PROCESSOR STATUS WORD ON THE STACK
1304                                        ;*      JSR     PC,$TYPE                ;CALL TYPE ROUTINE
1305                                        ;*      MESADDR                         ;FIRST ADRESS OF MESSAGE
1306
1307  000132  105737  001151          $TYPE: TSTB    $TPFLG                  ;IS THERE A TERMINAL?
1308  000136  100002                        BPL     1$                      ;BR IF YES
1309  000140  000078                        HALT                            ;HALT HERE IF NO TERMINAL
1310  000142  000407                        BR      3$                      ;LEAVE
1311  000144  010046                1$:     MOV     R0,-(SP)                ;SAVE R0
1312  000146  017600  000002                MOV     @2(SP),R0               ;GET ADDRESS OF ASCIZ STRING
```

```
1313  006192  112046                  251     MOVB    (R0)+,-(SP)     ;PUSH CHARACTER TO BE TYPED ONTO STACK
1314  006194  001005                          BNE     4$              ;BR IF IT ISN'T THE TERMINATOR
1315  006196  005726                          TST     (SP)+           ;IF TERMINATOR POP IT OFF THE STACK
1316  006200  012600                          MOV     (SP)+,R0        ;RESTORE R0
1317  006202  062716  000002                3$:     ADD     #2,(SP)         ;ADJUST RETURN PC
1318  006206  000002                          RTI                     ;RETURN
1319  006210  004737  006222                4$:     JSR     PC,7$           ;GO TYPE THIS CHARACTER
1320  006214  123726  001150                5$:     CMPB    $FILLC,(SP)+    ;IS IT TIME FOR FILLER CHARS.?
1321  006220  001364                          BNE     2$              ;IF NO GO GET NEXT CHAR.
1322  006222  013746  001146                        MOV     $NULL,-(SP)     ;GET # OF FILLER CHARS. NEEDED
1323                                                                  ;AND THE NULL CHAR.
1324  006226  105366  000001                6$:     DECB    1(SP)           ;DOES A NULL NEED TO BE TYPED?
1325  006232  002770                          BLT     5$              ;BR IF NO--GO POP THE NULL OFF OF STACK
1326  006234  004737  006222                        JSR     PC,7$           ;GO TYPE A NULL
1327  006240  000772                          BR      6$              ;LOOP
1328  006242  105777  172714                7$:     TSTB    @$TPS           ;WAIT UNTIL PRINTER IS READY
1329  006246  100375                          BPL     7$
1330  006250  116677  000002  172706                MOVB    2(SP),@$TPB     ;LOAD CHAR TO BE TYPED INTO DATA REG.
1331  006256  000207                          RTS     PC
1332                                        ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
1333
1334                                        .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
1335
1336                                        ;+CALL:
1337                                        ;+      MOV     NUM,-(SP)       ;PUT THE BINARY NUMBER ON THE STACK
1338                                        ;+      TYPDS                   ;GO TO THE ROUTINE
1339
1340  006260                                $TYPDS:
1341  006260  010046                                MOV     R0,-(SP)        ;PUSH R0 ON STACK
1342  006262  010146                                MOV     R1,-(SP)        ;PUSH R1 ON STACK
1343  006264  010246                                MOV     R2,-(SP)        ;PUSH R2 ON STACK
1344  006266  010346                                MOV     R3,-(SP)        ;PUSH R3 ON STACK
1345  006270  010546                                MOV     R5,-(SP)        ;PUSH R5 ON STACK
1346  006272  012746  020200                        MOV     #20200,-(SP)    ;SET BLANK SWITCH AND SIGN
1347  006276  016605  000020                        MOV     20(SP),R5       ;GET THE INPUT NUMBER
1348  006302  100004                                BPL     1$              ;BR IF INPUT IS POS.
1349  006304  005405                                NEG     R5              ;MAKE THE BINARY NUMBER POS.
1350  006306  112766  000055  000001                MOVB    #'-,1(SP)       ;MAKE THE ASCII NUMBER NEG.
1351  006314  005000                          1$:   CLR     R0              ;ZERO THE CONSTANTS INDEX
1352  006316  012703  006454                        MOV     #$DBLK,R3       ;SETUP THE OUTPUT POINTER
1353  006322  112723  000040                        MOVB    #' ,(R3)+       ;SET THE FIRST CHARACTER TO A BLANK
1354  006326  005002                          2$:   CLR     R2              ;CLEAR THE BCD NUMBER
1355  006330  016001  006444                3$:   MOV     $DTBL(R0),R1    ;GET THE CONSTANT
1356  006334  160105                          4$:   SUB     R1,R5           ;FORM THIS BCD DIGIT
1357  006336  002402                                BLT     4$              ;BR IF DONE
1358  006340  005202                                INC     R2              ;INCREASE THE BCD DIGIT BY 1
1359  006342  000774                                BR      3$
1360  006344  060105                          4$:   ADD     R1,R5           ;ADD BACK THE CONSTANT
1361  006346  005702                                TST     R2              ;CHECK IF BCD DIGIT=0
1362  006350  001002                                BNE     5$              ;FALL THROUGH IF 0
1363  006352  105716                                TSTB    (SP)            ;STILL DOING LEADING 0'S?
1364  006354  100427                                BMI     7$              ;BR IF YES
1365  006356  106316                          5$:   ASLB    (SP)            ;MSD?
1366  006360  103003                                BCC     6$              ;BR IF NO
```

```
1367  006342  116663  000001  177777          MOVB    1(SP),-1(R3)            ;YES--SET THE SIGN
1368  006350  052702  000060          6$:     BIS     #'0,R2                  ;MAKE THE BCD DIGIT ASCII
1369  006354  052702  000040          7$:     BIS     #' ,R2                  ;MAKE IT A SPACE IF NOT ALREADY A DIGIT
1370  006360  112223                          MOVB    R2,(R3)+                ;PUT THIS CHARACTER IN THE OUTPUT BUFFER
1371  006362  005720                          TST     (R0)+                   ;JUST INCREMENTING
1372  006364  020027  000010                  CMP     R0,#10                  ;CHECK THE TABLE INDEX
1373  006370  002746                          BLT     2$                      ;GO DO THE NEXT DIGIT
1374  006372  003002                          BGT     8$                      ;GO TO EXIT
1375  006374  010502                          MOV     R5,R2                   ;GET THE LSD
1376  006376  000764                          BR      6$                      ;GO CHANGE TO ASCII
1377  006400  105726                  8$:     TSTB    (SP)+                   ;WAS THE LSD THE FIRST NON-ZERO?
1378  006402  100003                          BPL     9$                      ;BR IF NO
1379  006404  116663  177777  177776          MOVB    -1(SP),-2(R3)           ;YES--SET THE SIGN FOR TYPING
1380  006412  105013                  9$:     CLRB    (R3)                    ;SET THE TERMINATOR
1381  006414  012605                          MOV     (SP)+,R5                ;POP STACK INTO R5
1382  006416  012603                          MOV     (SP)+,R3                ;POP STACK INTO R3
1383  006420  012602                          MOV     (SP)+,R2                ;POP STACK INTO R2
1384  006422  012601                          MOV     (SP)+,R1                ;POP STACK INTO R1
1385  006424  012600                          MOV     (SP)+,R0                ;POP STACK INTO R0
1386  006426  104400  006454                  TYPE    ,SDBLK                  ;NOW TYPE THE NUMBER
1387  006432  016666  000002  000004          MOV     2(SP),4(SP)             ;ADJUST THE STACK
1388  006440  012616                          MOV     (SP)+,(SP)
1389  006442  000002                          RTI                             ;RETURN TO USER
1390  006444  023420                  SDTBL:  10000.,
1391  006446  001750                          1000.,
1392  006450  000144                          100.,
1393  006452  000012                          10.,
1394  006454  000004                  SDBLK:  .BLKW   4
1395                                  ;;=====================================================================
1396
1397                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
1398
1399                                  ;+STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
1400                                  ;-CALL:
1401                                  ;-      MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
1402                                  ;-      TYPOS                           ;CALL FOR TYPEOUT
1403                                  ;-      .BYTE   N                       ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
1404                                  ;-      .BYTE   M                       ;M=1 OR 0
1405                                  ;-                                      ;1=TYPE LEADING ZEROS
1406                                  ;-                                      ;0=SUPPRESS LEADING ZEROS
1407                                  ;-
1408                                  ;+STYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
1409                                  ;-STYPOS OR STYPOC
1410                                  ;-CALL:
1411                                  ;-      MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
1412                                  ;-      TYPON                           ;CALL FOR TYPEOUT
1413                                  ;-
1414                                  ;+STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
1415                                  ;-CALL:
1416                                  ;-      MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
1417                                  ;-      TYPOC                           ;CALL FOR TYPEOUT
1418
1419  006464  017646  000000          STYPOS: MOV     @(SP),-(SP)             ;PICKUP THE MODE
1420  006472  116637  000001  006727          MOVB    1(SP),$BFILL            ;LOAD ZERO FILL SWITCH
```

```
006476  112637  006711                    MOVB    (SP)+,$OMODE+1          ;NUMBER OF DIGITS TO TYPE
006502  062716  000002                    ADD     #2,(SP)                 ;ADJUST RETURN ADDRESS
006506  000406                            BR      $TYPON
006510  112737  000001  006707  $TYPOC:   MOVB    #1,$0FILL               ;SET THE ZERO FILL SWITCH
006516  112737  000006  006711            MOVB    #6,$OMODE+1             ;SET FOR SIX(6) DIGITS
006524  112737  000005  006706  $TYPON:   MOVB    #5,$OCNT                ;SET THE ITERATION COUNT
006532  010346                            MOV     R3,-(SP)                ;SAVE R3
006534  010446                            MOV     R4,-(SP)                ;SAVE R4
006536  010546                            MOV     R5,-(SP)                ;SAVE R5
006540  113704  006711                    MOVB    $OMODE+1,R4             ;GET THE NUMBER OF DIGITS TO TYPE
006544  005404                            NEG     R4
006546  062704  000006                    ADD     #6,R4                   ;SUBTRACT IT FOR MAX. ALLOWED
006552  110437  006710                    MOVB    R4,$OMODE               ;SAVE IT FOR USE
006556  113704  006707                    MOVB    $0FILL,R4               ;GET THE ZERO FILL SWITCH
006562  016605  000012                    MOV     12(SP),R5               ;PICKUP THE INPUT NUMBER
006566  005003                            CLR     R3                      ;CLEAR THE OUTPUT WORD
006570  006105                  1$:       ROL     R5                      ;ROTATE MSB INTO "C"
006572  000404                            BR      3$                      ;GO DO MSB
006574  006105                  2$:       ROL     R5                      ;FORM THIS DIGIT
006576  006105                            ROL     R5
006600  006105                            ROL     R5
006602  010503                            MOV     R5,R3
006604  006103                  3$:       ROL     R3                      ;GET LSB OF THIS DIGIT
006606  105337  006710                    DECB    $OMODE                  ;TYPE THIS DIGIT?
006612  100016                            BPL     7$                      ;BR IF NO
006614  042703  177770                    BIC     #177770,R3              ;GET RID OF JUNK
006620  001002                            BNE     4$                      ;TEST FOR 0
006622  005704                            TST     R4                      ;SUPPRESS THIS 0?
006624  001403                            BEQ     5$                      ;BR IF YES
006626  005204                  4$:       INC     R4                      ;DON'T SUPPRESS ANYMORE 0'S
006630  052703  000060                    BIS     #'0,R3                  ;MAKE THIS DIGIT ASCII
006634  052703  000040          5$:       BIS     #' ,R3                  ;MAKE ASCII IF NOT ALREADY
006640  110337  006704                    MOVB    R3,8$                   ;SAVE FOR TYPING
006644  104420  006704                    TYPE    ,8$                     ;GO TYPE THIS DIGIT
006650  105337  006706          7$:       DECB    $OCNT                   ;COUNT BY 1
006654  003347                            BGT     2$                      ;BR IF MORE TO DO
006656  002402                            BLT     6$                      ;BR IF DONE
006660  005204                            INC     R4                      ;INSURE LAST DIGIT ISN'T A BLANK
006662  000744                            BR      2$                      ;GO DO THE LAST DIGIT
006664  012605                  6$:       MOV     (SP)+,R5                ;RESTORE R5
006666  012604                            MOV     (SP)+,R4                ;RESTORE R4
006670  012603                            MOV     (SP)+,R3                ;RESTORE R3
006672  016666  000002  000004            MOV     2(SP),4(SP)             ;SET THE STACK FOR RETURNING
006700  012616                            MOV     (SP)+,(SP)
006702  000002                            RTI                             ;RETURN
006704     000                  8$:       .BYTE   0                       ;STORAGE FOR ASCII DIGIT
006705     000                            .BYTE   0                       ;TERMINATOR FOR TYPE ROUTINE
006706     000                  $OCNT:    .BYTE   0                       ;OCTAL DIGIT COUNTER
006707     000                  $0FILL:   .BYTE   0                       ;ZERO FILL SWITCH
006710  000000                  $OMODE:   0                               ;NUMBER OF DIGITS TO TYPE
                                ;;*************************************************************


                                .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE

```

D3

```
                                ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" (SITEMB) TO DETERMINE WHICH
                                ;*ERROR IS TO BE REPORTED, IT THEN OBTAINS, FROM THE "ERROR TABLE" (SERRTB),
                                ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR,

006712                          SERRTYP:
006712  104420  001225                TYPE    ,SCRLF                  ;"CARRIAGE RETURN" & "LINE FEED"
006716  010046                        MOV     R0,-(SP)                ;SAVE R0
006720  005000                        CLR     R0                      ;PICKUP THE ITEM INDEX
006722  153700  001114                BISB    @#SITEMB,R0
006726  001004                        BNE     1$                      ;IF ITEM NUMBER IS ZERO, JUST
                                                                        ;TYPE THE PC OF THE ERROR
006730  013746  001116                MOV     SERRPC,-(SP)            ;SAVE SERRPC FOR TYPEOUT
                                                                        ;ERROR ADDRESS
006734  104402                        TYPOC                           ;GO TYPE--OCTAL ASCII(ALL DIGITS)
006736  000426                        BR      6$                      ;GET OUT
006740  005300                1$:     DEC     R0                      ;ADJUST THE INDEX SO THAT IT WILL
006742  006300                        ASL     R0                      ;       WORK FOR THE ERROR TABLE
006744  006300                        ASL     R0
006746  006300                        ASL     R0
006750  062700  001232                ADD     #SERRTB,R0              ;FORM TABLE POINTER
006754  012037  006764                MOV     (R0)+,2$                ;PICKUP "ERROR MESSAGE" POINTER
006760  001404                        BEQ     3$                      ;SKIP TYPEOUT IF NO POINTER
006762  104403                        TYPE                            ;TYPE THE "ERROR MESSAGE"
006764  000000                2$:     .WORD   0                       ;"ERROR MESSAGE" POINTER GOES HERE
006766  104420  001225                TYPE    ,SCRLF                  ;"CARRIAGE RETURN" & "LINE FEED"
006772  012037  007002        3$:     MOV     (R0)+,4$                ;PICKUP "DATA HEADER" POINTER
006776  001404                        BEQ     5$                      ;SKIP TYPEOUT IF 0
007000  104400                        TYPE                            ;TYPE THE "DATA HEADER"
007002  000000                4$:     .WORD   0                       ;"DATA HEADER" POINTER GOES HERE
007004  104420  001225                TYPE    ,SCRLF                  ;"CARRIAGE RETURN" & "LINE FEED"
007010  011000                5$:     MOV     (R0),R0                 ;PICKUP "DATA TABLE" POINTER
007012  001004                        BNE     7$                      ;GO TYPE THE DATA
007014  012600                6$:     MOV     (SP)+,R0                ;RESTORE R0
007016  104420  001225                TYPE    ,SCRLF                  ;"CARRIAGE RETURN" & "LINE FEED"
007022  000207                        RTS     PC                      ;RETURN
007024                          7$:
007024  013046                        MOV     @(R0)+,-(SP)            ;SAVE @(R0)+ FOR TYPEOUT
007026  104402                        TYPOC                           ;GO TYPE--OCTAL ASCII(ALL DIGITS)
007030  005710                        TST     (R0)                    ;IS THERE ANOTHER NUMBER?
007032  001770                        BEQ     6$                      ;BR IF NO
007034  104403  007042                TYPE    ,8$                     ;TYPE TWO(2) SPACES
007040  000771                        BR      7$                      ;LOOP
007042  020040          000   8$:     .ASCIZ  /  /                    ;TWO(2) SPACES
007046                                .EVEN
                                ;;*****************************************************************

                                .SBTTL  ERROR HANDLER ROUTINE

                                ;*SW15=1        HALT ON ERROR
                                ;*SW13=1        INHIBIT ERROR TYPEOUTS
                                ;*SW10=1        BELL ON ERROR
                                ;*SW09=1        LOOP ON ERROR
                                ;*GO TO SERRTYP ON ERROR
```

```
1529  007046                          $ERROR:
1530  007046  010637  001170                    MOV     SP,$REG6        ;GET THE CURRENT STACK POINTER VALUE
1531  007052  162737  000004  001170            SUB     #4,$REG6                ;RESTORE IT TO ITS "PRE ERROR TRAP" VALUE FOR PR
1532  007060  016637  000002  001172            MOV     2(SP),$REG7             ;GET THE PS OFF OF THE STACK
1533  007066  005037  001166                    CLR     $REG5                   ;PREPARE "$REG5" TO HOLD THE TEST #
1534  007072  113737  001132  001166            MOVB    $TSTNM,$REG5            ;TEST # IS HELD IN THE LOW BYTE OF "TSTNM"
1535  007100  010037  001154                    MOV     R0,$REG0                ;MOST OF THE TIME R0 HAS GOOD STUFF IN IT ALSO
1536  007104  105237  001153            7$:     INCB    $ERFLG                  ;SET THE ERROR FLAG
1537  007110  001775                          BEQ     7$                      ;DON'T LET THE FLAG GO TO ZERO
1538  007112  013737  001132  177570            MOV     $TSTNM,@#DISPLAY        ;DISPLAY TEST NUMBER AND ERROR FLAG
1539  007120  032737  002000  177570            BIT     #SW10,@#SWR             ;BELL ON ERROR?
1540  007126  001402                          BEQ     1$                      ;NO = SKIP
1541  007130  104420  001220                    TYPE    ,$BELL                  ;RING BELL
1542  007134  005237  001112            1$:     INC     $ERTTL                  ;COUNT THE NUMBER OF ERRORS
1543  007140  011637  001116                    MOV     (SP),$ERRPC             ;GET ADDRESS OF ERROR INSTRUCTION
1544  007144  162737  000002  001116            SUB     #2,$ERRPC
1545  007152  117737  171748  001114            MOVB    @$ERRPC,$ITEMB          ;STRIP AND SAVE THE ERROR ITEM CODE
1546  007160  032737  020000  177570            BIT     #SW13,@#SWR             ;SKIP TYPEOUT IF SET
1547  007166  001004                          BNE     2$                      ;SKIP TYPEOUTS
1548  007170  004737  006712                    JSR     PC,@#$ERRTYP            ;GO TO USER ERROR ROUTINE
1549  007174  104400  001225                    TYPE    ,$CRLF
1550  007200  005737  177570            2$:     TST     @#SWR                   ;HALT ON ERROR
1551  007204  100001                          BPL     3$                      ;SKIP IF CONTINUE
1552  007206  000000                          HALT                            ;HALT ON ERROR!
1553  007210  032737  001000  177570    3$:.    BIT     #SW09,@#SWR             ;LOOP ON ERROR SWITCH SET?
1554  007216  001402                          BEQ     4$                      ;BR IF NO
1555  007220  013716  001110                    MOV     $LPERR,(SP)             ;FUDGE RETURN FOR LOOPING
1556  007224  005737  001216            4$:     TST     $ESCAPE                 ;CHECK FOR AN ESCAPE ADDRESS
1557  007230  001402                          BEQ     5$                      ;BR IF NONE
1558  007232  013716  001216                    MOV     $ESCAPE,(SP)            ;FUDGE RETURN ADDRESS FOR ESCAPE
1559  007236  000002                  5$:     RTI                             ;RETURN
1560                                  ;;*****************************************************************
1561
1562                                  .SBTTL  TRAP DECODER
1563
1564                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
1565                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
1566                                  ;*OF THE DESIRED ROUTINE, THEN USING THE ADDRESS OBTAINED IT WILL
1567                                  ;*GO TO THAT ROUTINE.
1568
1569  007240  010046                  $TRAP:  MOV     R0,-(SP)                ;SAVE R0
1570  007242  016600  000002                  MOV     2(SP),R0                ;GET TRAP ADDRESS
1571  007246  005740                          TST     -(R0)                   ;BACKUP BY 2
1572  007250  111003                          MOVB    (R0),R3                 ;GET RIGHT BYTE OF TRAP
1573  007252  006000  007260                  MOV     $TRPAD(R0),R0           ;INDEX TO TABLE
1574  007256  000200                          RTS     R0                      ;GO TO ROUTINE
1575
1576
1577                                  .SBTTL  TRAP TABLE
1578
1579                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
1580                                  ;*BY THE "TRAP" INSTRUCTION.
1581
1582                                  ;       ROUTINE
```

```
                                         ;       -------
007260                                   $TRPAD:
007260  006132                                   $TYPE          ;CALL=TYPE       TRAP+0(104400)  TTY TYPEOUT ROUTINE
007262  006510                                   $TYPOC         ;CALL=TYPOC      TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING
007264  006464                                   $TYPOS         ;CALL=TYPOS      TRAP+4(104404)  TYPE OCTAL NUMBER (NO LEADING ZE
007266  006524                                   $TYPON         ;CALL=TYPON      TRAP+6(104406)  TYPE OCTAL NUMBER (AS PER LAST C
007270  006240                                   $TYPDS         ;CALL=TYPDS      TRAP+10(104410) TYPE DECIMAL NUMBER (WITH SIGN)
                                         ;;-----------------------------------------------------------------------

                                         .SBTTL  POWER DOWN AND UP ROUTINES

                                         ;POWER DOWN ROUTINE
007272  012737  007434  000024   $PWRDN: MOV     #$ILLUP,@#PWRVEC            ;SET FOR FAST UP
007300  012737  000340  000026           MOV     #340,@#PWRVEC+2             ;PRIO:7
007306  010046                           MOV     R0,-(SP)                    ;PUSH R0 ON STACK
007310  010146                           MOV     R1,-(SP)                    ;PUSH R1 ON STACK
007312  010246                           MOV     R2,-(SP)                    ;PUSH R2 ON STACK
007314  010346                           MOV     R3,-(SP)                    ;PUSH R3 ON STACK
007316  010446                           MOV     R4,-(SP)                    ;PUSH R4 ON STACK
007320  010546                           MOV     R5,-(SP)                    ;PUSH R5 ON STACK
007322  013746  007662                   MOV     POWPUS,-(SP)                ;PUSH POWPUS ON STACK
007326  013746  007664                   MOV     POWPOP,-(SP)                ;PUSH POWPOP ON STACK
007332  013746  007452                   MOV     POWMES,-(SP)                ;PUSH POWMES ON STACK
007336  013746  001550                   MOV     T0001,-(SP)                 ;PUSH T0001 ON STACK
007342  010637  007440                   MOV     SP,$SAVR6                   ;SAVE SP
007346  012737  007360  000024           MOV     #$PWRUP,@#PWRVEC            ;SET UP VECTOR
007354  000000                           HALT
007356  000776                           BR      .-2                         ;HANG UP

                                         ;POWER UP ROUTINE
007360  013706  007440           $PWRUP: MOV     $SAVR6,SP                   ;GET SP
007364  005037  007440                   CLR     $SAVR6                      ;WAIT LOOP FOR THE TTY
007370  005237  007440           1$:     INC     $SAVR6                      ;WAIT FOR THE INC
007374  001375                           BNE     1$                          ;OF <POWPUS>,<POWPOP>,<POWMES>,<T0001> WORD
007376  012605                           MOV     (SP)+,R5                    ;POP STACK INTO R5
007400  012604                           MOV     (SP)+,R4                    ;POP STACK INTO R4
007402  012603                           MOV     (SP)+,R3                    ;POP STACK INTO R3
007404  012602                           MOV     (SP)+,R2                    ;POP STACK INTO R2
007406  012601                           MOV     (SP)+,R1                    ;POP STACK INTO R1
007410  012600                           MOV     (SP)+,R0                    ;POP STACK INTO R0
007412  012737  007272  000024           MOV     #$PWRDN,@#PWRVEC            ;SET UP THE POWER DOWN VECTOR
007420  012737  000340  000026           MOV     #340,@#PWRVEC+2             ;PRIO:7
007426  104400  007442                   TYPE    ,$POWER                     ;POWER FAIL MESSAGE
007432  000002                           RTI
007434  000000                   $ILLUP: HALT                                ;THE POWER UP SEQUENCE WAS STARTED
007436  000776                           BR      .-2                         ; BEFORE THE POWER DOWN WAS COMPLETE
007440  000000                   $SAVR6: 0                                   ;PUT THE SP HERE
007442  005015  047520  042527   $POWER: .ASCIZ  <15><12>"POWER"
007450  000122
                                         .EVEN
007452  005015  042522  052123   POWMES: .ASCIZ <15> <12> "RESTARTING AFTER A POWER FIALURE" <15> <12> <12>
007460  051101  044524  043516
007466  040440  052106  051105
007474  040440  050040  053517
```

```
1637  007502  051105  043040  040511
1638  007510  052514  042522  205015
1639  007516  000012
1640  007522  005015  046412  026504  STMES1:  .ASCIZ  <15><12><12>"MD-11-DDKWA-A LINE FREQUENCY CLOCK TEST"<15><12>
1641  007526  030461  042055  045504
1642  007534  040527  040455  046040
1643  007542  047111  020105  051106
1644  007550  050505  042525  041516
1645  007556  020131  046103  041517
1646  007564  020113  042524  052123
1647  007572  005015     000
1648
1649  007575     124  040522  050120  TRPMES1:  .ASCIZ  "TRAPPED TO LOC 4 FROM LOCATION  "
1650  007602  042105  052040  020117
1651  007610  047514  020103  020064
1652  007616  051106  046517  046040
1653  007624  041517  052101  047511
1654  007632  020116  000040
1655  007636  042522  052123  051101  TRPM2S1:  .ASCIZ  "RESTARTING PROGRAM"
1656  007644  044524  043516  050040
1657  007652  047522  051107  046501
1658  007660     000
1659          007662                  .EVEN
1660  007662  001230                  POWPUS1:  WORD
1661  007664  001230                  POWPOP1:  WORD
1662  007666  020040  020040  020040  EM11:  .ASCIZ  "                                LKS     LKS     "
1663  007674  020040  020040  020040
1664  007702  020040  020040  020040
1665  007710  020040  020040  020040
1666  007716  020040  020040  020040
1667  007724  020040  045514  020123
1668  007732  020040  020040  045514
1669  007740  020123  020040  020040
1670  007746     000
1671  007747     050  041520  020051  DH11:  .ASCIZ  "(PC)    (PS)    (SP)    TEST#   WAS     S/B     "
1672  007754  020040  024040  051520
1673  007762  020051  020040  024040
1674  007770  050123  020051  020040
1675  007776  052040  051505  021524
1676  010004  020040  053440  051501
1677  010012  020040  020040  051440
1678  010020  041057  020040  020040
1679  010026  000040
1680                          .EVEN
1681  010030  001116  001172  001170  DT11:  SERRPC,SREG7,SREG6,SREG5,LKS,SGDDAT
1682  010036  001166  177546  001124
1683  010044  000000          0
1684  010046  046103  041517  020113  EM21:  .ASCIZ  "CLOCK FAILED TO INTERRUPT"
1685  010054  042506  046111  042105
1686  010062  052040  020117  047111
1687  010070  042524  051122  050125
1688  010076  000124
1689  010103  050050  024503  020040  DH21:  .ASCIZ  "(PC)    (PS)    (SP)    TEST#   (LKS)   "
1690  010106  020040  050050  024523
```

```
1691  010114  020040  020040  051450
1692  010122  024520  020040  020040
1693  010130  042524  052123  020043
1694  010136  020040  046050  051513
1695  010144  020051  020040     000
1696          010152                  .EVEN
1697  010152  001116  001172  001170  DT2:    SERRPC,SREG7,SREG6,SREG5,LKS
1698  010160  001166  177546
1699  010164  000000                  0
1700  010166  046103  041517  020113  EM3:    .ASCIZ  "CLOCK INTERRUPTED WHEN THE PROCESSOR PRIORITY WAS TOO HIGH"
1701  010174  047111  042524  051122
1702  010202  050125  042524  020104
1703  010210  044127  047105  052040
1704  010216  042510  050040  047522
1705  010224  042503  051523  051117
1706  010232  050040  044522  051117
1707  010240  052111  020131  040527
1708  010246  020123  047524  020117
1709  010254  044510  044107     000
1710  010261     050  041520  020051  DH3:    .ASCIZ  "(PC)    (PS)    (SP)    TST#   (LKS)   "
1711  010266  020040  024040  051520
1712  010274  020051  020040  024040
1713  010302  052123  020051  020040
1714  010310  052040  051505  021524
1715  010316  020040  024040  045514
1716  010324  024523  020040  000040
1717                          .EVEN
1718  010332  001116  001172  001170  DT3:    SERRPC,SREG7,SREG6,SREG5,LKS
1719  010340  001166  177546
1720  010344  000000                  0
1721  010346  046103  041517  020113  EM4:    .ASCIZ  "CLOCK GIVES UNEQUAL # OF PULSES OVER TWO EQUAL PERIODS OF TIME"
1722  010354  044527  042526  020123
1723  010362  047125  050505  040525
1724  010370  022114  020043  043117
1725  010376  050040  046125  042523
1726  010404  020123  053117  051105
1727  010412  052040  047527  042440
1728  010422  052521  046101  050040
1729  010426  051105  047511  051504
1730  010434  047440  020106  044524
1731  010442  042515     000
1732  010445     050  041520  020051  DH4:    .ASCIZ  "(PC)    (PS)    (SP)    TST#   1ST     2ND"<15><12>"
1733  010452  020040  024040  051520
1734  010460  020051  020040  024040
1735  010466  052123  020051  020040
1736  010474  052040  051505  021524
1737  010502  020040  030440  052123
1738  010510  020040  020040  031040
1739  010516  042116  005015  020040
1740  010524  020040  020040  020040
1741  010532  020040  020040  020040
1742  010540  020040  020040  020040
1743  010546  020040  020040  020040
1744  010554  020040  020040  020040
```

```
1745  010562  042520  044522  042117
1746  010570  020040  042520  044522
1747  010576  042117     000
1748          010602                  .EVEN
1749  010602  001116  001172  001170  DT4:    SERRPC,SREG7,SREG6,SREG5,SREG1,SREG2
1750  010610  001166  001156  001160
1751  010616  000000                  0
1752  010620  045514  020123  042522  EM5:    .ASCIZ  "LKS REGISTER RESPONDS TO ANOTHER ADDRESS"
1753  010626  044507  052123  051105
1754  010634  051040  051505  047520
1755  010642  042116  020123  047524
1756  010650  040440  047516  044124
1757  010656  051105  040440  042104
1758  010664  042522  051523     000
1759  010671     050  041520  020051  DH5:    .ASCIZ  "(PC)    (PS)    (SP)    TEST#   ADDRESS"
1760  010676  020040  024040  051520
1761  010704  020051  020040  024040
1762  010712  050123  020051  020040
1763  010720  052040  051505  021524
1764  010726  020040  040440  042104
1765  010734  042522  051523     000
1766          010742                  .EVEN
1767  010742  001116  001172  001170  DT5:    SERRPC,SREG7,SREG6,SREG5,SGDADR
1768  010750  001166  001120
1769  010754  000000                  0
1770  010756  020101  047516  051440  EM6:    .ASCIZ  "A NO SACK TIMEOUT HAS OCCURED"
1771  010764  041501  020113  044524
1772  010772  042515  052517  020124
1773  011000  040510  020123  041517
1774  011006  052503  042522  000104
1775  011014  050050  024503  020040  DH6:    .ASCIZ  "(PC)    (PS)    (SP)    TEST#   (LKS)   "
1776  011022  020040  050050  024523
1777  011030  020040  020040  051450
1778  011036  024520  020040  020040
1779  011044  042524  052123  020043
1780  011052  020040  046050  051513
1781  011060  020051  020040     000
1782          011066                  .EVEN
1783  011066  001116  001172  001170  DT6:    SERRPC,SREG7,SREG6,SREG5,LKS
1784  011074  001166  177546
1785  011100  000000                  0
1786  011102  051127  047117  020107  EM7:    .ASCIZ  "WRONG CONDITION CODES WERE PUT ONTO STACK BY INTERRUPT"
1787  011110  047523  042116  052111
1788  011116  047511  020116  047503
1789  011124  042504  020123  042527
1790  011132  042522  050040  052125
1791  011140  044440  052116  020117
1792  011146  052123  041501  020113
1793  011154  054502  044440  052116
1794  011162  051105  052522  052120
1795  011170     000
1796  011171     050  041520  020051  DH7:    .ASCIZ  "(PC)    (PS)    (SP)    TEST#   CC WAS  CC S/B"
1797  011176  020040  024040  051520
1798  011204  020051  020040  024040
```

```
1799  011212  000123  020051  020040
1800  011220  052040  051505  021524
1801  011226  020040  041440  020103
1802  011234  042527  020123  041440
1803  011242  020103  027523  000102
1804                          .EVEN
1805  011250  001116  001172  001170  DT7:   SERRPC,$REG7,$REG6,$REG5,BUF1,$GOODAT
1806  011256  001166  000776  001124
1807  011264  000000                  0
1808  011266  051127  047117  020107  EM10:  .ASCIZ  "WRONG PC PUT ONTO THE STACK BY AN INTERRUPT"
1809  011274  041520  050040  052125
1810  011302  047440  052116  020117
1811  011310  044124  020105  052123
1812  011316  041501  020113  054502
1813  011324  040440  020116  047111
1814  011332  042524  051122  050125
1815  011340  000124
1816  011342  050050  024503  020040  DH10:  .ASCIZ  "(PC)    (PS)    (SP)    TEST#  @(SP)WAS @(SP)S/B  "
1817  011352  020040  050050  024523
1818  011356  020040  020040  051490
1819  011364  024520  020040  020040
1820  011372  042524  052123  020043
1821  011400  040040  051450  024520
1822  011406  042527  020123  024100
1823  011414  050123  051451  041057
1824  011422  020040     000
1825          011426                  .EVEN
1826  011426  001116  001172  001170  DT10:  SERRPC,$REG7,$REG6,$REG5,BUF2,$GOODAT
1827  011434  001166  000774  001124
1828  011442  000000                  0
1829  011444  051124  042531  020104  EM11:  .ASCIZ  "TRYED TO ACCESS THE LKS REGISTER, AND TRAPPED"
1830  011452  047524  040440  041503
1831  011460  051505  020123  044124
1832  011466  020125  045514  020123
1833  011474  042522  044507  052123
1834  011502  051105  020054  047101
1835  011510  020104  051124  050101
1836  011516  042520  000104
1837  011522  050050  024503  020040  DH11:  .ASCIZ  "(PC)    (PS)    (SP)    TEST#"
1838  011530  020040  050050  024523
1839  011536  020040  020040  051490
1840  011544  024520  020040  020040
1841  011552  042524  052123  000043
1842                          .EVEN
1843  011562  001116  001172  001170  DT11:  SERRPC,$REG7,$REG6,$REG5
1844  011566  001166
1845  011570  000000                  0
1846
1847          017400                  .=17400
1848  017400  000000          ERRST:  .WORD 0
1849  017402  000000                  .WORD 0
1850  017404  000000                  .WORD 0
1851  017406  000000                  .WORD 0
1852  017410  000000                  .WORD 0
```

```
1853  017412  000000                          ,WORD 0
1854          000001                  ,END
```

DDKWAA.P11 CROSS REFERENCE TABLE

```
A0004   000762            537#    540
A0005   001044            555#    558
A0006   001150            579#    582
A0007   001300            607#    610
A0010   001430            639#    642
A0011   001532            667#    670
A0012   001666            702#    705
A0013   001752            729#    732
A0014   002272            802     808#
A0015   002404            820     829#
A0016   002516            841     850#
A0017   002630            862     871#
A0020   002742            883     892#
A0021   003054            904     913#
A0023   003146            935#    938
A0024   003264            959#    960
A0025   003370            980#    983
A0026   003532           1011#   1012
A0027   003642           1034#   1035
A0030   003776           1052    1060#
A0031   004060           1078#   1081
A0032   004206           1112#   1115
A0033   004342           1143#   1146
BIT0    =  000001            293#
BIT00   =  000001            283#    293
BIT01   =  000002            282#    292
BIT02   =  000004            281#    291
BIT03   =  000010            280#    290
BIT04   =  000020            279#    289
BIT05   =  000040            278#    288
BIT06   =  000100            277#    287
BIT07   =  000200            276#    286
BIT08   =  000400            275#    285
BIT09   =  001000            274#    284
BIT1    =  000002            292#
BIT10   =  002000            273#
BIT11   =  004000            272#
BIT12   =  010000            271#
BIT13   =  020000            270#
BIT14   =  040000            269#
BIT15   =  100000            268#
BIT2    =  000004            291#
BIT3    =  000010            290#
BIT4    =  000020            289#
BIT5    =  000040            288#
BIT6    =  000100            287#
BIT7    =  000200            286#
BIT8    =  000400            285#
BIT9    =  001000            284#
BPTVEC= 000014               300#
BUF1    =  000776            312#   1120*   1127    1151*   1805
BUF2    =  000774            311#   1121*   1152*   1158    1826
B0005       002062            556     561#
B0006       002166            559     585#
```

```
B0007   002316          638     613#
B0010   002446          643     645#
B0011   002552          668     673#
B0012   002704          703     708#
B0013   002770          730     735#
B0014   003004          809     811#
B0015   003416          830     832#
B0016   003530          851     853#
B0017   003642          872     874#
B0020   003754          893     895#
B0021   004066          914     916#
B0023   004164          936     941#
B0025   004406          961     966#
B0030   005010         1002#   1003
B0031   005076         1079    1084#
B0032   005224         1113    1118#
B0033   005362         1144    1149#
C0007   002326          616#    618
C0010   002456          648#    650
C0011   002600          679#    682
C0013   003004          739#    742
C0025   004426          976     992#
C0031   005126         1090#   1093
C0032   005266         1122    1127#
C0033   005422         1153    1158#   1160
DH1     007747          413    1071#
DH10    011342          448    1016#
DH11    011522          453    1037#
DH2     010100          418    1009#
DH3     010261          423    1710#
DH4     010445          428    1732#
DH5     010671          433    1759#
DH6     011014          438    1775#
DH7     011171          443    1796#
DISPLA= 177570          225#   1201=   1538=
DT1     010030          414    1001#
DT10    011426          449    1026#
DT11    011560          454    1043#
DT2     010152          419    1007#
DT3     010332          424    1710#
DT4     010502          429    1749#
DT5     010742          434    1707#
DT6     011006          439    1703#
DT7     011250          444    1005#
D0007   002340          601     621#
D0010   002470          633     653#
D0011   002616          660     665#
D0013   003022          740     745#
D0031   005144         1091    1096#
EMTVEC= 000030          303#    472=    473=
EM1     007066          412    1002#
EM10    011266          447    1008#
EM11    011444          452    1029#
EM2     010046          417    1004#
```

EM3	010166	422	1700#			
EM4	010346	427	1721#			
EM5	010620	432	1752#			
EM6	010756	437	1770#			
EM7	011102	442	1786#			
EOPHLT	000256	329#				
ERRST	017400	1209*	1214*	1215*	1848#	
ERRVEC=	000004	296#	1243	1244*	1246*	1249*
E0001	001606	493	496#			
E0002	001654	510#				
E0003	001720	524#				
E0004	001774	541#				
E0005	002056	559#				
E0006	002162	583#				
E0007	002312	611#				
E0010	002442	643#				
E0011	002544	671#				
E0012	002700	706#				
E0013	002764	733#				
E0014	003302	810#				
E0015	003414	831#				
E0016	003526	852#				
E0017	003640	873#				
E0020	003752	894#				
E0021	004064	915#				
E0023	004160	939#				
E0024	004304	955	964#			
E0025	004402	984#				
E0026	004552	1006	1016#			
E0027	004674	1040#				
E0030	005024	1060	1066#			
E0031	005072	1082#	1087			
E0032	005220	1116#				
E0033	005354	1147#				
E10006	002210	590#				
E10007	002334	619#				
E10010	002464	651#				
E10012	002720	713#				
E10013	003016	743#				
E10023	004202	945#				
E10025	004422	990#				
E10031	005142	1094#				
E10033	005432	1160#				
E1005	002100	565#				
E1011	002612	676	683#			
E20007	002346	623#				
E20010	002476	655#				
E20011	002624	689#				
E20013	003100	763#				
E20025	004436	994#				
E20031	005152	1103#				
E30013	003132	773#				
E40013	003174	786#				
F0013	003026	747#	752			

```
GNS     = ****** U     323     1585    1586    1587    1588    1589
G0013     003042       748     751#
H0013     003066       759#    762
IOTVEC= 000020         301#    470*    471*
I0001     001600       494#
I0014     003264       807#
I0015     003376       828#
I0016     003510       849#
I0017     003622       870#
I0020     003734       891#
I0021     004046       912#
I0023     004164       942#
I0026     004540       1013#
J0013     003104       760     765#
KSTART    001400       324     400#
K0013     003122       769#    772
LKS     = 177546       309#    494*    508     522     535*    537     553*    555     562*    563     577*    579     586*
                       587*    588     605*    607     614*    621     635*    639     646*    653     665*    667     677*
                       679     700*    702     709*    711     726*    729     737*    739     746*    747     750*    756*
                       759     767*    769     777*    778     781*    806*    808     811*    827*    829     832*    840*
                       852     853*    869*    871     874*    890*    892     895*    911*    913     916*    933*    935
                       942*    943     956*    958*    962     977*    983     988*    992     1000*   1010*   1014    1030*
                       1036*   1037*   1038    1049*   1056*   1058*   1064*   1076*   1078    1080*   1090    1110*   1112
                       1123*   1141*   1143    1154*   1167*   1681    1697    1710    1783
L0013     003136       770     775#
M0013     003150       778#    783
NOP     = 000240       310#
N0013     003164       779     782#
PC      =%000007       237#    333*    462*    1185*   1188*   1197*   1207*   1217*   1319*   1320*   1331*   1509*   1540*
PIRQ    = 177772       223#
PIRQVE= 000240         307#
POWMES    007452       1025    1633#
POWPOP    007664       1024    1601#
POWPUS    007662       1003    1600#
PS      = 177776       220#    221     461     552*    576*    604*    636*    675*    724*    804*    825*    846*    867*
                       888*    909*    957*    978*    987*    1005*   1031*   1054*   1061*   1086*   1124*   1155*
PSW     = 177776       221#
PWRVEC= 000024         302#    476*    477*    1595*   1596*   1600*   1623*   1624*
RESVEC= 000010         297#
R0      =%000000       228#    536*    539*    615*    616*    647*    649*    722*    751*    753*    782*    1195*   1197
                       1311    1312*   1313    1316*   1341    1351*   1355    1371    1372    1385*   1401    1402*   1403*
                       1490*   1491*   1492*   1493*   1494*   1495    1500    1505*   1507*   1511    1513    1535    1569
                       1570*   1571    1572*   1573*   1574*   1597    1622*
R0001     001574       492     493#
R0002     001634       505     506#
R0003     001712       520     521#
R0004     001746       533     534#
R0005     002036       551     553#
R0006     002142       575     577#
R0007     002260       600     603#
R0010     002410       632     635#
R0011     002524       664     665#
R0012     002660       699     700#
R0013     002724       722#    776
```

C3

C4

```
R0014   001254          880     885#
R0015   001366          823     826#
R0016   001502          844     847#
R0017   001612          865     868#
R0020   001724          886     889#
R0021   002036          907     910#
R0022   002140          932     935#
R0023   002250          954     957#
R0024   002394          974     977#
R0025   002514         1004    1008#
R0026   002622         1027    1030#
R0027   002734         1051    1052#
R0030   003052         1075    1076#
R0031   003200         1109    1110#
R0032   003334         1140    1141#
R1      =%000001      229#     723#    749#    784     786     1342    1355#   1356    1360    1384#   1398    1021#
R1013   002736        725#
R2      =%000002      230#     754#    788#    784     787     1343    1354#   1358#   1361    1368#   1369#   137#    1379#
                     1383#    1399    1425#
R2013   002770        736#
R3      =%000003      231#     554#    557#    578#    581#    686#    689#    638#    641#    666#    669#    678#    681#
                      701#     704#    728#    731#    738#    741#    758#    761#    768#    771#    934#    937#    979#
                      982#    1077#   1088#   1089#   1092#   1111#   1114#   1142#   1145#   1344    1352#   1353#   1367#
                     1378#    1379#   1388#   1382#   1427    1436#   1442#   1443#   1446#   1451#   1452#   1453    1462#
                     1600    1619#
R3013   003052        755#
R4      =%000004      232#    1428     1438#   1431#   1432#   1433    1434#   1448    1458#   1458#   1461#   1481    1618#
R4013   003104        766#
R5      =%000005      233#    1345     1347#   1349#   1356#   1362#   1375    1381#   1429    1435#   1437#   1439#   1448#
                     1441#    1442    1468#   1682    1617#
R6      =%000006      234#     236      465#    466#    467
R7      =%000007      235#     237
SP      =%000006      236#     468#    461#    469#    493#    683#    637#    674#    885#    826#    847#    868#    889#
                      910#    1889#   1832#   1955#   1885#   1119#   1152#   1192#   1286#   1218#   1211#   1214    1216#
                     1243#    1246    1248    1249    1277    1278    1282#   1311#   1312    1313#   1315    1316    1317#
                     1320     1322#   1324#   1338    1341#   1342#   1343#   1344#   1345#   1346#   1347    1358#   1363
                     1365#    1367    1377    1379    1381    1382    1383    1384    1385    1387#   1388#   1419#   1428
                     1421     1422#   1427#   1428#   1429#   1435    1402    1401    1462    1463#   1464#   1481#   1486#
                     1537     1511#   1538    1532    1543    1555#   1558#   1569#   1578    1597#   1598#   1599#   1688#
                     1681#    1682#   1683#   1684#   1685#   1686#   1687    1613#   1617    1618    1619    1628    1621
                     1622
STACK = 001183       217#
START   001416       464#    1224
STKLMT= 177774       222#
STMES   007520       463     1648#
SWR   = 177570       224#     225     1238#   1252    1254    1268    1267    1539    1546    1558    1553
SW8   = 000001       265#
SW00  = 000001       255#     265
SW01  = 000002       254#     264
SW02  = 000004       253#     263
SW03  = 000010       252#     262
SW04  = 000020       251#     261
SW05  = 000040       250#     260
SW06  = 000100       249#     259
```

```
SW07  = 000200          248#    258
SW08  = 000400          247#    257    1232    1252
SW09  = 001000          246#    256     389     479    1231    1257    1253    1256    1258    1260    1278    1281    1284
                       1526    1553    1568
SW1   = 000002          264#
SW10  = 002000          245#    393    1525    1539    1560
SW11  = 004000          244#    388     479    1182    1238    1253    1256    1257    1258    1265    1266    1267    1281
                       1284
SW12  = 010000          243#    480    1176    1195    1282
SW13  = 020000          242#   1524    1546
SW14  = 040000          241#   1229    1238
SW15  = 100000          240#   1523    1558
SW2   = 000004          263#
SW3   = 000010          262#
SW4   = 000020          261#
SW5   = 000040          260#
SW6   = 000100          259#
SW7   = 000200          258#
SW8   = 000400          257#
SW9   = 001000          256#
TBITVE= 000014          298#
TKVEC = 000060          305#
TPVEC = 000064          306#
TRAPVE= 000034          304#    474*    475*
TRAPS   005572          503     517     531     548     572     597     632     662     697     929     951     971    1001
                       1024    1047    1073    1137    1138    1286#
TRPMES  007575          1238    1289    1649#
TRPM2S  007636          1218    1655#
TRTVEC= 000014           299#
TYPDS = 104412          1193    1589#
TYPE  = 104402          1191    1194    1213    1386    1454    1487    1497    1499    1502    1504    1508    1515    1541
                       1549    1585#   1625
TYPOC = 104402          1212    1488    1512    1586#
TYPOW = 104406          1588#
TYPOS = 104404          1587#
T0001   001550          482     489#    1281    1636
T0002   001610          495     502#
T0003   001656          509     516#
T0004   001722          523     530#
T0005   001776          538     547#
T0006   002102          560     564     571#
T0007   002212          584     589     596#
T0010   002350          612     620     622     629#
T0011   002500          644     652     654     661#    684
T0012   002626          672     688     695#
T0013   002722          707     712     721#
T0014   003210          734     744     764     774     785     798#
T0015   003322          812     819#
T0016   003434          833     840#
T0017   003546          854     861#
T0020   003662          875     882#
T0021   003772          896     903#
T0022   004104          921#
T0023   004106          917     928#
```

```
T0024   004204   943    944    953#
T0025   004306   963    978#
T0026   004442   985    991    993    1303#
T0027   004554   1015   1023#
T0030   004676   1029   1039   1046#
T0031   005026   1065   1272#
T0032   005154   1083   1095   1099   1106#  1131
T0033   005310   1117   1128   1137#
T0034   005442   1148   1159   1166#
WORD    001232   394#   1033*  1034*  1057*  1062*  1066#  1601
SBDADR  001122   358#
SBDDAT  001126   362#
SBELL   001222   390#   1541   1568
SCNTAG  001102   346#   465    472    478    479
SCH1  * 000010   372#   373#   374#   375#   376#   377#   378#   379#   380#
SCH2  * 000020   372#   373#   374#   375#   376#   377#   378#   379#   380#
SCH3  * 000010   370#   372
SCH4  * 000010   380#   381#   382#   383#   384#   385#   386#   387#   388#
SCRLF   001225   392#   1213   1488   1499   1504   1508   1549   1568
SDBLK   006454   1352   1386   1394#
SDOAGN  005546   1187   1196   1201#
SDTBL   006444   1355   1390#
SENDAD  005536   342    1197#
SENDCT  005586   478    1189#
SENDMG  005552   1191   1202#
SENULL  005567   1194   1209#
SEOP    005452   1179#
SEOPCT  005502   478*   1186#  1198
SERFLG  001103   349#   1234   1256   1258   1284*  1285   1536*  1568
SERMAX  001115   355#   479*   1258   1288*  1285
SERROR  007046   472    1529#
SERRPC  001116   356#   1486   1543*  1544*  1545   1567   1601   1697   1718   1749   1767   1783   1889
                 1826   1843
SERRTB  001232   411#   1494
SERRTY  006712   1479#  1548
SERTTL  001112   353#   1542*  1563
SESCAP  001216   389#   1279*  1556   1558   1560
SFILLC  001150   368#   1328   1332
SFILLS  001147   367#   1332
SGDADR  001120   357#   881*   887*   813*   824*   828*   834*   845*   849*   855*   866*   878*   876*
                 887*   891*   897*   988*   912*   918*   1707
SGDDAT  001124   359#   507*   519*   534*   553*   574*   599*   696*   799*   822*   843*   864*   885*
                 906*   931*   943    953*   962    973*   1883*  1814   1826*  1838   1858*  1129*  1168*
                 1681   1889   1826
SGET42  005530   1195#
SHD   * 002003   233
SICNT   001104   358#   1271*  1272   1274*  1284
SILLUP  007434   1595   1627#
SITEMB  001114   354#   1483   1545*  1563
SLF     001226   393#   1568
SLPADR  001106   351#   488*   492*   585*   528*   533*   551*   575*   683*   632*   664*   699*   725*
                 736*   755*   766*   776*   882*   923*   844*   865*   886*   987*   932*   954*   974*
                 1084*  1827*  1351*  1875*  1189*  1148*  1202*  1277*  1282   1284
SLPERR  001110   352#   1262   1278*  1284   1555
```

```
$MXCNT  000130          1275    1284#
$NULL   001146           366#   1322    1332
$OCNT   000706          1426*   1455*   1468#
$OMODE  000710          1421*   1429*   1430    1433*   1444*   1470#
$OVER   000114          1239    1255    1263    1273    1281#
$PASS   001100           347#   1183*   1184*   1192    1202    1269    1285
$POWER  007442          1625    1630#
$PWRDN  007272           476    1595#   1623
$PWRUP  007360          1608    1613#
$QUES   001224           391#   1568
$RDCHR* ****** U        1598
$RDDEC* ****** U        1598
$RDLIN* ****** U        1598
$RDOCT* ****** U        1598
$REGAD  001152           370#
$REG0   001154           372#   1539*
$REG1   001156           373#    706*   1749
$REG2   001160           374#    707*   1749
$REG3   001162           375#
$REG4   001164           376#
$REG5   001166           377#   1533*   1534*   1681    1697    1718    1749    1767    1783    1805    1826    1843
$REG6   001170           378#   1530*   1531*   1681    1697    1718    1749    1767    1783    1805    1826    1843
$REG7   001172           379#   1532*   1681    1697    1718    1749    1767    1783    1805    1826    1843
$SAVRE* ****** U        1593
$SAVR6  007440          1607*   1613    1614*   1615*   1629#
$SCOPE  005662           473    1236#
$SETUP* 002037           190#    478     472     474     476     478     479     489    1181
$STUP * 177777           193#
$SVLAD  006266          1247    1276#
$SWR  * 167400           185#    208     205     206     207     208     209     210     211     388     389     398     479
                         488    1176    1182    1195    1202    1229    1237    1231    1232    1238    1257    1252    1253
                        1256    1257    1258    1265    1266    1267    1278    1281    1284    1523    1524    1525    1526
                        1539    1546    1553    1553    1560
$TIMES  001214           388#   1182*   1265*   1272    1279*   1284
$TKB    001140           363#
$TKS    001136           362#    467
$TMP0   001174           383#
$TMP1   001176           381#
$TMP2   001200           382#
$TMP3   001202           383#
$TMP4   001204           384#
$TMP5   001206           385#
$TMP6   001210           386#
$TMP7   001212           387#
$TN   * 000000           186#    208
$TPB    001144           365#   1338*   1332
$TPFLG  001151           369#   1387    1332
$TPS    001142           364#   1328    1332
$TRAP   007242           474    1569#
$TRP  * 000012          1576#   1586#   1587#   1588#   1589#   1590#
$TRPAD  007262          1573    1584#
$TSTIM  001102           348#   1181*   1234    1254    1276*   1281    1285    1534    1538    1568
$TYPBN* ****** U        1593
$TYPDS  006240          1340#   1589
```

DDKWAA.P11 CROSS REFERENCE TABLE

```
STYPE   000132            462    1207    1217    1387#   1576    1985
STYPOC  000510           1424#   1586
STYPON  000524           1423    1426#   1588
STYPCS  000464           1419#   1587
SXTSTR  000672           1241#
SOFILL  000707           1420.   1424.   1434    1469#
.       = 017414          316#    328     322#    325#    328#    341#    344#    394     459#    468     482     1202    1206
                         1284    1285    1332    1394#   1518#   1567    1617    1628    1659#   1696#   1748#   1766#   1782#
                         1825#   1847#
```

```
ADITAG    525#  394
COMMEN    308#
ENDCOM    308#
ERROR     218#  496  510  524  541  559  565  583  597  611  619  623  643  651  655
           671  683  689  706  713  733  743  763  773  788  810  831  852  873  894
           915  939  945  964  984  998  994 1016 1047 1066 1082 1094 1108 1116 1138
          1147 1161
ESCAPE    308#
MULT      308#
NEWTST    308#
POP       308#  1381 1617
PUSH      308#  1340 1597 1603
SAVE     1519#  1530
SCOPE     219#  489  502  516  530  547  571  596  629  661  695  721  798  819  840
           861  882  903  921  928  950  978 1008 1023 1046 1072 1106 1137 1166 1186
SETRAP    198#  503  517  531  548  572  597  638  662  697  929  951  971 1001 1024
          1047 1073 1107 1138
SETTRA   1576#  1586 1587 1588 1589
SETUP     308#  464
SKIP      308#
SLASH     308#
SPACE     308#
STARS     189#  308#  334  349 1178 1225 1285 1332 1395 1471 1519 1568 1598
TRMTRP   1576#
TYPBIN    308#
TYPDEC    308#  1192
TYPNUM    308#
TYPOCS    308#
TYPOCT    308#  1486 1510
TYPTXT    308#
WAITLK    198#  554  578  606  638  666  678  701  728  738  758  768  934  979 1077
          1089 1111 1142
$SCMRE    334#  372  373  374  375  376  377  378  379
$SCMTM    334#  380  381  382  383  384  385  386  387
$SESCA    308#
$SNEWT    308#
$SSET    1576#  1586 1587 1588 1589
$SSKIP    308#
.EQUAT    188#  213
.HEADE    188#  190
.SETUP    188#  190
.SWRHI    188#  200
.SWRLO    188#  212#
.SCATC    188#  313
.SCMTA    189#  334
.SEOP     188#  1170
.SERRO    189#  1519
.SERRT    189#  1471
.SPOWE    189#  1590
.SSCOP    188#  1225
.STRAP    189#  1560
.STYPD    189#  1332
.STYPE    189#  1285
.STYPO    189#  1395
```

DDKWAA;P11 CROSS REFERENCE TABLE

```
                955     957     958     971     972     973     974     975     976     978     980    1001    1002    1003    1004
               1005    1006    1007    1009    1010    1024    1025    1026    1027    1029    1031    1032    1036    1037    1047
               1048    1050    1051    1052    1055    1058    1060    1073    1074    1075    1085    1086    1087    1088    1107
               1108    1109    1119    1122    1123    1124    1129    1138    1139    1140    1150    1153    1154    1155    1160
               1188    1192    1195    1209    1210    1214    1243    1244    1246    1249    1262    1274    1275    1277    1278
               1281    1282    1311    1312    1316    1322    1341    1342    1343    1344    1345    1346    1347    1392    1399
               1375    1381    1382    1383    1384    1385    1387    1388    1419    1427    1428    1429    1435    1442    1460
               1461    1462    1463    1464    1481    1486    1495    1500    1505    1507    1511    1530    1532    1535    1538
               1543    1555    1558    1569    1570    1573    1595    1596    1597    1598    1599    1600    1601    1602    1603
               1604    1605    1606    1607    1608    1613    1617    1618    1619    1620    1621    1622    1623    1624
MOVB            479    1280    1313    1338    1352    1353    1367    1370    1379    1420    1421    1424    1425    1426    1430
               1433    1434    1453    1534    1545    1572
NEG            1349    1431
NOP             617     649     686     687     789    1097    1098    1198    1199    1200    1219    1220    1221    1222    1237
RESET           483     506     521     710     961    1313    1100    1223
ROL            1238    1437    1439    1440    1441    1443
RTI            1283    1310    1389    1405    1559    1626
RTS             333    1331    1509    1574
SEC            1125    1156
SEN            1125    1156
SEV            1125    1156
SEZ            1125    1156
SUB            1211    1356    1531    1944
TRAP           1576    1586    1587    1588    1589
TST             481    1245    1269    1315    1361    1371    1448    1513    1598    1556    1571
TSTB            522     537     555     579     607     621     639     653     667     679     702     711     729     739     747
                759     769     778     935     983    1078    1093    1112    1143    1256    1307    1320    1363    1377
WAIT            989    1059    1126    1157
.ABS            185
.ASCII          391     392
.ASCIZ          390     393    1202    1517    1638    1633    1642    1649    1695    1662    1671    1684    1689    1700    1710
               1721    1732    1752    1759    1773    1775    1786    1796    1808    1816    1829    1837
.BLKW           1394
.BYTE           348     349     354     355     366     367     368     369    1205    1466    1467    1468    1469
.ENABL          185     187
.END            1854
.ENDC           190     195     200     210     211     212     218     294     300     325     335     345     370     380     388
                389     390     391     395     396     469     470     472     474     476     478     479     480    1171    1174
               1175    1176    1178    1181    1187    1190    1191    1195    1202    1205    1206    1220    1235    1238    1240
               1251    1254    1256    1258    1260    1267    1271    1276    1281    1284    1285    1286    1333    1396    1472
               1490    1519    1520    1528    1536    1543    1549    1550    1562    1561    1570    1573    1585    1586    1587
               1588    1589    1590    1591    1607    1617    1626    1633
.EQUIV          218     219     221     236     237     256     257     258     259     260     261     262     263     264     269
                284     285     286     287     288     289     290     291     292     293
.EVEN          1518    1632    1659    1680    1696    1717    1740    1706    1702    1804    1825    1842
.IF             190     191     200     209     213     211     212     216     266     294     320     334     344     370     380
                388     389     390     394     395     465     469     470     472     474     476     478     479     480    1170
               1174    1175    1176    1177    1178    1180    1186    1189    1191    1195    1201    1202    1225    1234    1237
               1238    1250    1252    1253    1256    1257    1258    1267    1269    1270    1284    1285    1332    1395    1471
               1489    1505    1519    1527    1530    1539    1546    1548    1549    1550    1553    1560    1569    1573    1576
               1586    1587    1588    1589    1590    1603    1617    1625    1626    1630
.IFF            220     210     211     212     216     335     345     370     396     470    1171    1177    1181    1187    1190
               1222    1226    1251    1254    1258    1284    1286    1333    1396    1472    1490    1519    1520    1527    1539
               1560    1561    1570    1591    1626
```

DDKWAA,P11 CROSS REFERENCE TABLE

```
.IFT      1266    1549
.IFTF     1264    1548
.IIF       190     195     200     205     206     207     208     320     394     478     472     478     479     488    1175
          1181    1182    1193    1202    1226    1229    1230    1231    1232    1265    1266    1281    1284    1285    1332
          1487    1512    1523    1524    1525    1526    1560    1585    1586    1587    1588    1589
.IRP       190     394    1237    1341    1381    1532    1597    1603    1617
.LIST        4     185     190     308     320     370     372     373     374     375     376     377     378     379     380
           381     382     383     384     385     386     387     388    1576    1585    1586    1587    1588    1589    1590
.MACRO     190     212     325     334    1519    1576
.MCALL     188     189     308
.NLIST       4     185     190     308     320     370     372     373     374     375     376     377     378     379     380
           381     382     383     384     385     386     387     388    1576    1585    1586    1587    1588    1589    1590
.PAGE      334     395
.REM         5
.REPT      320     372     380
.SBTTL     201     214     314     321     336     397     487     500     514     528     545     569     594     627     659
           693     717     793     817     838     859     880     901     926     949     968     998    1020    1044    1070
          1104    1135    1165    1172    1227    1287    1334    1397    1473    1521    1562    1577    1592
.TITLE     190
.WORD      320     326     327     347     350     351     352     353     356     357     358     359     360     361     370
           372     373     374     375     376     377     378     379     380     381     382     383     384     385     386
           387    1186    1189    1498    1503    1848    1849    1850    1851    1852    1853
```

ERRORS DETECTED: 0

```
.DDKWAA,DDKWAA/SOL/CRF=DDKWAA,P11
RUN-TIME:  25 16 3 SECONDS
CORE USED:  16K
```