KW11-L

LINE FREQUENCY CLOCK TEST

MD-11-DDKWA-A

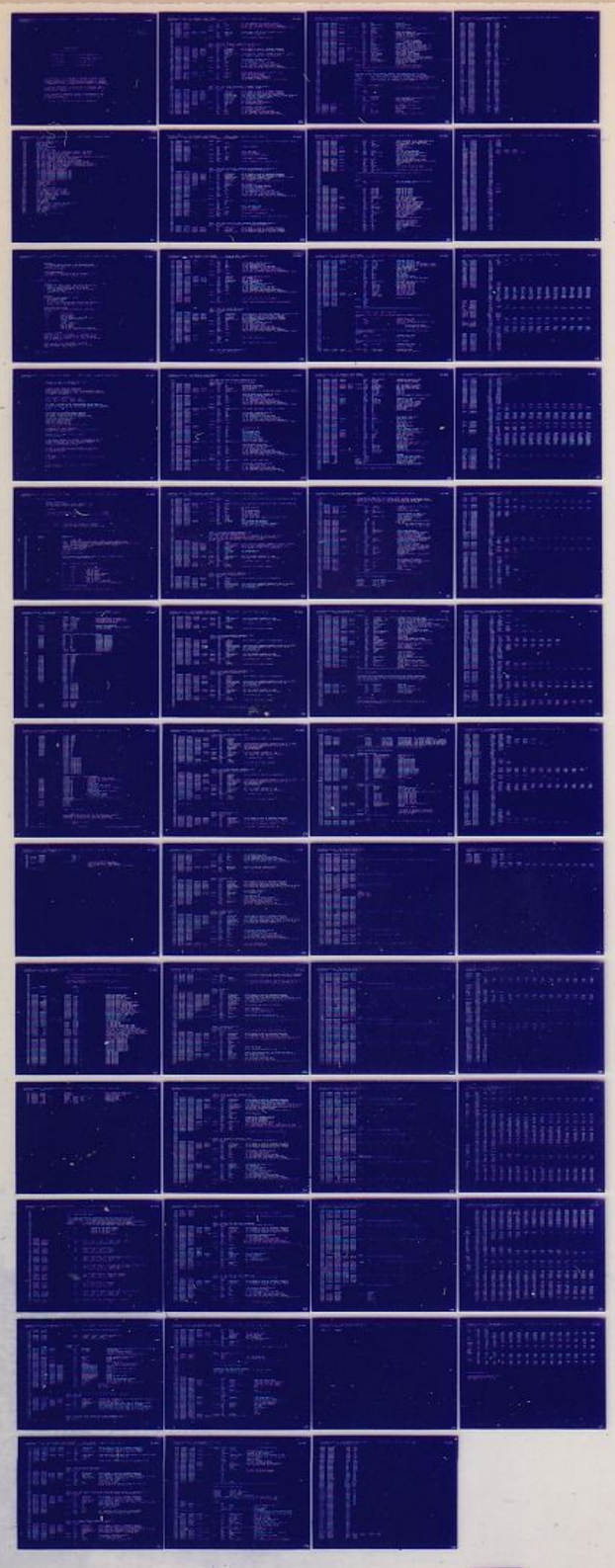
EP DDKWA A DL

FICHE 1 OF 1

JAN 1978

digital

MADE IN USA



KW11

LINE FREQUENCY CLOCK TEST

MD-11-DDKWA-A

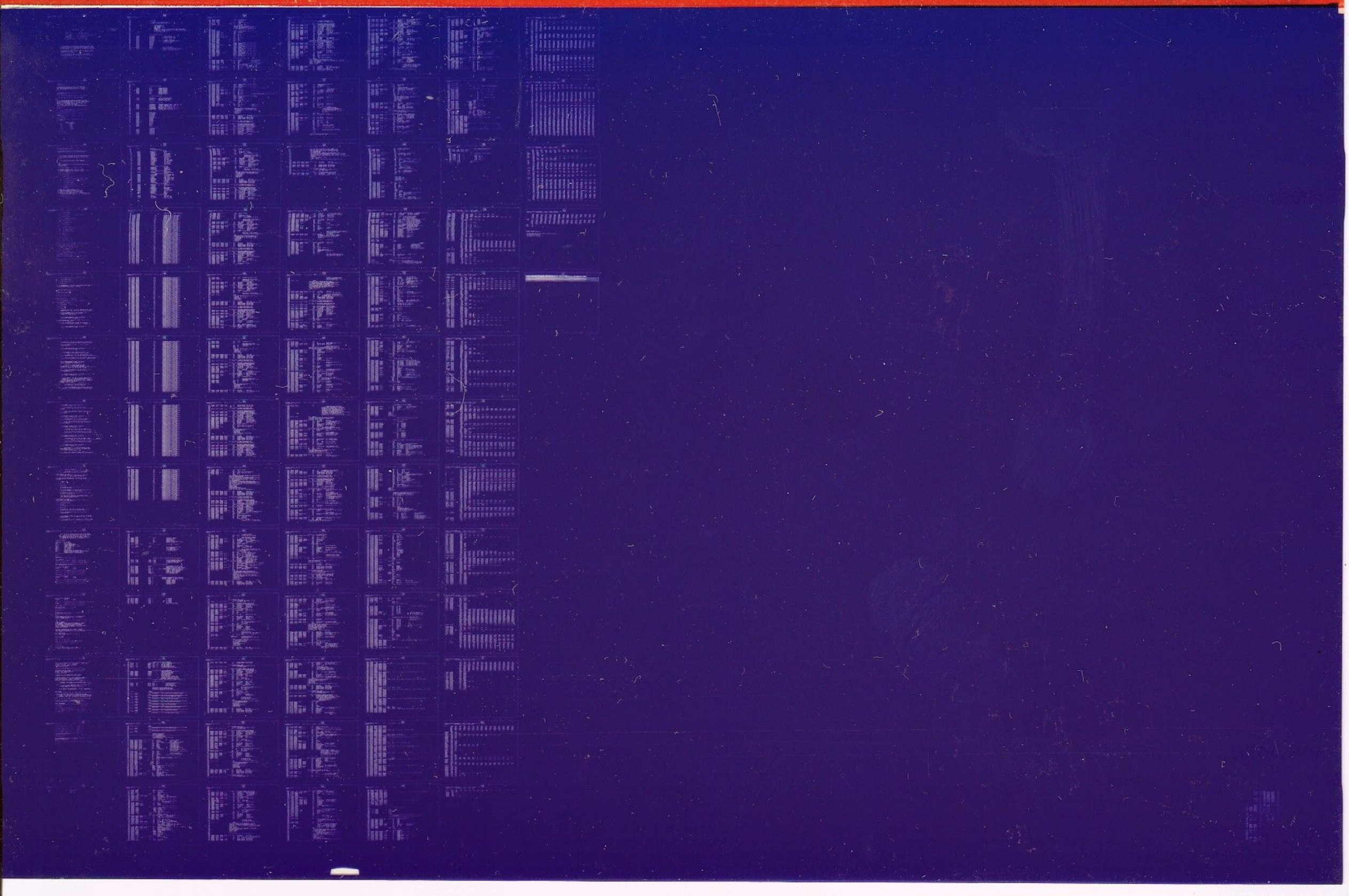
EP-DDKWA-A-DL-A

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 OPERATING 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
```

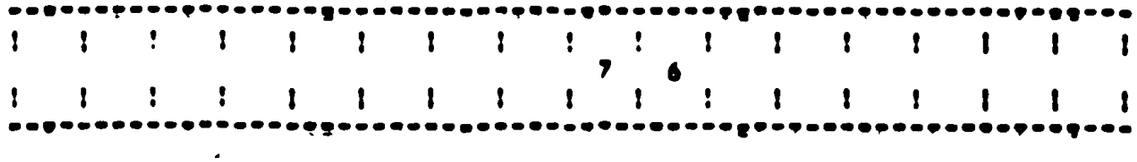
AFTER PRINTING OUT THE MESSAGE THE PROGRAM TRYS TO
RESTART ITSELF.

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



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

7.0 FLOW CHARTS

```
185          167400                  $SWR=167400
186          000000                  $TN=0
187                                          .ENABL  ABS
188                                  .MCALL  .HEADER,.SCATCH,.SEOP,.EQUAT,.SWRHI,.SWRLO,.SSCOPE,.SETUP
189                                  .MCALL  .STYPOCT,.STYPDEC,.STRAP,.SPOWER,.SERROR,.STYPE,STARS,.SERRTYP,.SCMTAG
190                                  .TITLE  MAINDEC-11-DDKWA-A  LINE FREQUENCY CLOCK PROGRAM
191                                  ;*COPYRIGHT (C) 1978,1972,1975
192                                  ;*DIGITAL EQUIPMENT CORP,
193                                  ;*MAYNARD, MASS, 01754
194                                  ;*
195                                  ;*PROGRAM BY J. COMEAU
196                                  ;*
197                                  ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
198                                  ;*PACKAGE (MAINDEC-11-DZQAC-A1),
199                                  ;*
200
201                                  .SBTTL  OPERATIONAL SWITCH SETTINGS
202                                  ;*
203                                  ;*      SWITCH                  USE
204                                  ;*      ******          ********************
205                                  ;*        15            HALT ON ERROR
206                                  ;*        14            LOOP ON TEST
207                                  ;*        13            INHIBIT ERROR TYPEOUTS
208                                  ;*        11            INHIBIT ITERATIONS
209                                  ;*        10            BELL ON ERROR
210                                  ;*         9            LOOP ON ERROR
211                                  ;*         8            LOOP ON TEST IN SWR<7:0>
212                                  ;*       7-0            #OF TEST TO LOOP ON IF SWR<8> IS SET
213
214                                  .SBTTL  BASIC DEFINITIONS
215
216                                  ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
```

```
                        177120          STACK=  1120
                                        .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 DEFINITIO   (BIT00 TO BIT15)
                        100000          BIT15=  100000
                        040000          BIT14=  40000
                        020000          BIT13=  20000
```

```
                        000020          BIT12=  10000
                        000010          BIT11=  4000
                        000008          BIT10=  2000
                        000004          BIT09=  1000
                        000002          BIT08=  400
                        000001          BIT07=  200
                        000000          BIT06=  100
                        000000          BIT05=  40
                        000000          BIT04=  20
                        000000          BIT03=  10
                        000000          BIT02=  4
                        000000          BIT01=  2
                        000000          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
                                        ;MISCELANEOUS 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)
                        001000                  .=1000

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

```
                    000050                          .=50
000050  000000      000000                          .WORD   0
000052  000000      000000                          .WORD   0
                    000250                          .=250
000250  000000      000000          EOPHLT: HALT            ;THIS IS AN END OF PASS HALT!
                                                    ;NOT AN ERROR HALT, THIS HAPPENS
                                                    ;ONLY IF SW6 IS UP, PRESS CONTINUE FOR ANOTHER
                                                    ;PASS.
000252  000207      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 ADDR
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 ADDR OF 'GOOD' DATA
001122  000000          $BDADR: .WORD   0                       ;CONTAINS ADDR 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
001142  177562          $TKB:   177562                          ;TTY KBD BUFFER
001142  177564          $TPS:   177564                          ;TTY PRINTER STATUS REG. ADDR
001144  177566          $TPB:   177566                          ;TTY PRINTER BUFFER REG. ADDR
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          $REGAD: .WORD   0                       ;CONTAINS THE ADDR FROM
                                ;                               ;WHICH  ($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
```

```
                001214  000000          STIMES: 0                       ;MAX. NUMBER OF ITERATIONS
                001216  000000          SESCAPE:0                       ;ESCAPE ON ERROR 1100
                001220  177607  000377  SBELL:  .ASCIZ  <207><377><377> ;CODE FOR BELL
                001224     077          SQUES:  .ASCII  /?/             ;QUESTION MARK
                001225     015          SCRLF:  .ASCII  <15>            ;CARRIAGE RETURN
                001226  000012          SLF:    .ASCIZ  <12>            ;LINE FEED
                00123C  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 SITEM3, 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  010038                          DT1     ;SERRPC,SREG7,SREG6,SREG5,LKS,SGDDAT
001240  000000                          0
                                ;
001242  010046                          EM2     ;"CLOCK FAILED TO INTERRUPT"
001244  010120                          DH2     ;"PC      PS      SP      TEST#   LKS     "
001246  010152                          DT2     ;SERRPC,SREG7,SREG6,SREG5,LKS
001250  000000                          0
                                ;
001252  010166                          EM3     ;"CLOCK INTERRUPTED WHEN THE PROCESSOR PRIORITY WAS TOO HIGH"
001254  010261                          DH3     ;"PC      PS      SP      TEST#   LKS     "
001256  010332                          DT3     ;SERRPC,SREG7,SREG6,SREG5,LKS
001260  000000                          0
                                ;
001262  010346                          EM4     ;"CLOCK GIVES UNEQUAL # OF PULSES OVER TWO EQUAL PERIODS OF TIME"
001264  010445                          DH4     ;"PC      PS      SP      TEST#   1ST     2ND"
001266  010672                          DT4     ;SERRPC,SREG7,SREG6,SREG5,SREG1,SREG0
001270  000000                          0
                                ;
001272  010620                          EM5     ;"LKS REGISTER RESPONDS TO ANOTHER ADDRESS"
001274  010671                          DH5     ;"PC      PS      SP      TEST#   ADDRESS"
001276  010742                          DT5     ;SERRPC,SREG7,SREG6,SREG5,SGDADR
001300  000000                          0
                                ;
001302  010756                          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  011172                          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#   "
```

```
                                                DT18    ;SERRPC,SREG7,SREG6,SREG5,SUP2,SCODAT
                                                0
                                                EM11    ;"TRAPPED TRYING TO ACCESS LKS REGISTER"
                                                DH11    ;"(PC)    (PS)    (SP)    TEST#"
                                                DT11    ;SERRPC,SREG7,SREG6,SREG5
                                        0
                                        ;STARTUP CODE
                                        .=1000
                                        KSTART: MOV     #1000,SP                ;INITIALIZE THE STACK SO WE CAN CALL THE TYPEOUT
                                                MOV     PS,-(SP)                ;SAVE STATUS
                                                JSR     PC,STYPE                ;PRINTOUT STARTUP MESSAGE
                                                STMES                           ;ADDRESS OF MESSAGE "MAINDEC-11-DDKWA-A"
                                        START:
                                                MOV     #SCMTAG,R0              ;FIRST LOCATION TO BE CLEARED
                                                CLR     (R0)+                   ;CLEAR MEMORY LOCATION
                                                CMP     #STKS,R0                ;DONE?
                                                BNE     .-6                     ;LOOP BACK IF NO
                                                MOV     #1000,SP                        ;SETUP THE STACK POINTER
                                                MOV     #$SCOPE,@#IOTVEC        ;IOT VECTOR FOR SCOPE ROUTINE
                                                MOV     #340,@#IOTVEC+2         ;LEVEL 7
                                                MOV     #SERROR,@#EMTVEC        ;EMT VECTOR FOR ERROR ROUTINE
                                                MOV     #340,@#EMTVEC+2         ;LEVEL 7
                                                MOV     #STRAP,@#TRAPVEC        ;TRAP VECTOR FOR TRAP CALLS
                                                MOV     #340,@#TRAPVEC+2        ;LEVEL 7
                                                MOV     #SPWRUN,@#PWRVEC        ;POWER FAILURE VECTOR
                                                MOV     #340,@#PWRVEC+2         ;LEVEL 7
                                                MOV     SENDCT,SCOPCT           ;SETUP END-OF-PROGRAM COUNTER
                                                MOVB    #1,SERMAX               ;ALLOW ONE ERROR PER TEST
                                                MOV     #.,SLPADR               ;INITIALIZE THE LOOP ADDRESS FOR SCOPE
                                                TST     42              ;LOADED BY A MONITOR
                                                BEQ     T0001           ;BR IF NO
                                                RESET                   ;YES--GENERATE AN INIT
                                        .SBTTL  TEST THAT THE LKS CAN BE REFERENCED WITHOUT A BUS ERROR
                                        ;LKS ACCESS TEST
                                        T00011: SCOPE
                                                MOV     #E0001,@#4      ;PREPARE FOR ADDRESSING THE LKS REGISTER, BAD HARDWARE
                                                MOV     #340,@#6        ;COULD CAUSE A TRAP TO 4
                                                MOV     #R0001,SLPADR   ;TIGHTEN UP THE SCOPE LOOP A BIT IN CASE OF AN ERROR
                                        R00011: MOV     #1000,SP        ;SETUP THE STACK POINTER IN CASE OF AN ERROR
                                        100011: CLR     @#LKS           ;JUST REFERENCE LKS, DONT WORRY IF IT DIDNT CLEAR YET
                                                BR      T0002           ;WE DIDNT TRAP IF WE REACH HERE, GO ON TO NEXT TEST
                                        E00011: 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
                                        T00021: SCOPE
```

```
001612  012737  003572  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                  R00002: RESET
001636  012737  000200  001124          MOV     #200,$GOODAT    ;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  001401                          BEQ     T0003
001654  104001                  E00002: ERROR 1                 ;ERROR, CLOCK INTERRUPT ENABLE NOT CLEARED BY INIT


                                ;SBTTL  TEST THAT STANT SETS CLOCK FLAG
                                ;TEST THAT START SETS CLOCK FLAG
001656  000004                  T0003:  SCOPE
001660  012737  003572  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,$GOODAT    ;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                  R0003:  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                  E0003:  ERROR 1                 ;ERROR, CLOCK FLAG NOT SET BY INIT


                                ;SBTTL  TEST THAT CLOCK FLAG WILL SET AFTER SUFFICIENT PERIOD OF TIME (20 MS MIN)
                                ;TEST THAT CLOCK FLAG WILL SET AFTER SUFFICIENT PERIOD OF TIME (20 MS MIN)
001722  000004                  T0004:  SCOPE
001724  012737  003572  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  R0004:  MOV     #200,$GOODAT    ;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          A0004:  TSTB    LKS             ;IS CLOCK FLAG SET
001766  100403                          BMI     T0005
001770  005200                          INC     R0              ;NO, INCREMENT COUNT AND WAIT FOR SOMEMORE
001772  001373                          BNE     A0004           ;WAIT SUFFICIENT AMOUNT OF TIME FOR CLOCK
001774  104001                  E0004:  ERROR 1                 ;ERROR, CLOCK FLAG FAILED TO SET


                                ;SBTTL  TEST THAT INTERRUPT ENABLE BIT MAY BE SET
                                ;TEST THAT INTERRUPT ENABLE BIT MAY BE SET
001776  000004                  T0005:  SCOPE
002000  012737  003572  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,$GOODAT    ;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          R0005:  CLR     LKS
002042  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
002044  105737  177546          A0005:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
002050  100404                          BMI     B0005           ;IF SO, CONTINUE ON WITH THE TEST
```

```
557  002252  005203                    INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
558  002254  001373                    BNE     A0005           ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
559  002256  104001              E00005: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
560  002260  000412                    BR      T0006
561  002262                       B00005:
562  002262  012737  000100  177546      MOV     #100,LKS        ;CLEAR CLOCK FLAG AND SET INTERRUPT ENABLE
563  002270  032737  000100  177546      BIT     #100,LKS        ;IS INTERRUPT ENABLE SET?
564  002276  001001                    BNE     T0006
565  002300  104001              E10005: ERROR 1                 ;ERROR INTERRUPT ENABLE NOT SET
566
567
568
569                                  .SBTTL  TEST THAT INTERRUPT ENABLE BIT MAY BE CLEARED
570                                  ;TEST THAT INTERRUPT ENABLE BIT MAY BE CLEARED
571  002302  005024              T00006: SCOPE
572  002304  012737  000572  000004      MOV     #TRAP8,#4       ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
573  002312  012737  000340  000006      MOV     #340,#6         ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
574  002320  012737  000000  001124      MOV     #0,SCOUNT
575  002326  012737  002142  001106      MOV     #ERR006,SLPADR  ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
576  002334  012737  000340  177776      MOV     #340,PS         ;SET PRIORITY LEVEL TO 7, NO INTERRUPTS
577  002342  005037  177546        A00006: CLR     LKS
578  002346  005003                    CLR     R3              ;INITIALIZE A COUNTER LOCATION
579  002350  105737  177546        A00006: TSTB    LKS             ;IS THE CLOCK FLAG SET?
580  002354  100404                    BMI     B00006          ;IF SO, CONTINUE ON WITH THE TEST
581  002356  005203                    INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
582  002360  001373                    BNE     A00006          ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
583  002362  104001              E00006: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
584  002364  000412                    BR      T0007
585  002366                       B00006:
586  002366  012737  000100  177546      MOV     #100,LKS        ;CLEAR CLOCK FLAG AND SET INTERRUPT ENABLE
587  002374  005037  177546            CLR     LKS             ;CLEAR INTERRUPT ENABLE
588  002400  032737  000100  177546      BIT     #100,LKS        ;TEST THE INTERRUPT ENABLE BIT
589  002406  001401                    BEQ     T0007           ;IS INTERRUPT ENABLE CLEARED
590  002410  104001              E10006: ERROR 1                 ;ERROR, ERROR INTERRUPT BIT CAN NOT BE CLEARED
591
592
593
594                                  .SBTTL  TEST THAT CLOCK INTERRUPTS TO CORRECT VECTOR ADDRESS
595                                  ;TEST THAT CLOCK INTERRUPTS TO CORRECT VECTOR ADDRESS
596  002412  005024              T00007: SCOPE
597  002414  012737  000572  000004      MOV     #TRAP8,#4       ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
598  002422  012737  000340  000006      MOV     #340,#6         ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
599  002430  012737  000000  001124      MOV     #0,SCOUNT       ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
600  002436  012737  002452  001106      MOV     #ERR007,SLPADR  ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
601  002444  012737  002460  000100      MOV     #D00007,100     ;SET UP VECTOR RETURN POINTER
602  002452  012737  000340  000102      MOV     #340,#102       ;1 INTERRUPT IS ENOUGH
603  002460  012706  001000        R00007: MOV     #1000,SP        ;GET STACK READY FOR INTERRUPTS
604  002464  012737  000200  177776      MOV     #200,PS         ;SET PROCESSOR PRIORITY 4
605  002472  005037  177546            CLR     LKS
606  002476  005003                    CLR     R3              ;INITIALIZE A COUNTER LOCATION
607  002500  105737  177546        A00007: TSTB    LKS             ;IS THE CLOCK FLAG SET?
608  002504  100404                    BMI     B00007          ;IF SO, CONTINUE ON WITH THE TEST
609  002506  005203                    INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
610  002510  001373                    BNE     A00007          ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
```

```
000312  104001                  E00071: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
000314  000415                          BR      T00010
000316                          000071:
000316  012737  000100  177546          MOV     #100,LKS        ;ENABLE INTERRUPT
000324  005000                          CLR     R0
000326  005200                  C00071: INC     R0
000330  000240                          NOP                     ;STALL FOR TIME
000332  001375                          BNE     C00071          ;WAIT FOR INTERRUPT
000334  104002                  E100071: ERROR 2                ;ERROR, DIDNT GET INTERRUPT
000336  000404                          BR      T00010
000340  105737  177546          D00071: TSTB    LKS             ;ENTER HERE IF INTERRUPTED
000344  100401                          BMI     T00012
000346  104001                  E200071: 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
000352  000004                  T00101: SCOPE
000352  012737  000572  000324          MOV     #TRAP5,#04      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
000360  012737  000340  000006          MOV     #340,#06        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
000366  012737  000410  001106          MOV     #ERR10,SLPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
000374  012737  000420  000102          MOV     #DON10,1P5      ;SET UP VECTOR RETURN POINTER
000402  012737  000340  000102          MOV     #340,#00102     ;NO INTERRUPTS ALLOWED AFTER THE FIRST ONE
000410  005037  177546          ERR101: CLR     LKS
000414  012737  000240  177776          MOV     #240,PS         ;SET PRIORITY 5
000422  012706  001000                  MOV     #1000P,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     B00101          ;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     C00110          ;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, INTERRUPTOID 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  000324          MOV     #TRAP5,#04      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
000510  012737  000340  000006          MOV     #340,#06        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
000516  012737  002524  001106          MOV     #ER011,SLPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
```

```
002524  005037  177546                  R00111: CLR     LKS
002530  005003                                  CLR     R3              ;INITIALIZE A COUNTER LOCATION
002532  105737  177546                  A00111: TSTB    LKS             ;IS THE CLOCK FLAG SET?
002536  100404                                  BMI     B0011           ;IF SO, CONTINUE ON WITH THE TEST
002540  005203                                  INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
002542  001373                                  BNE     A0011           ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
002544  104001                          E00111: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
002546  000427                                  BR      T0012
002550                                  B00111:
002550  012706  001000                          MOV     #1000,SP        ;INITIALIZE THE STACK POINTER
002554  012737  000300  177776                  MOV     #300,PS         ;SET PRIORITY 6
002562  012737  002012  000100                  MOV     #E1011,100      ;SET UP VECTOR RETURN
002570  012737  000100  177546                  MOV     #100,LKS        ;ENABLE INTERRUPT
002576  005003                                  CLR     R3              ;INITIALIZE A COUNTER LOCATION
002600  105737  177546                  C00111: TSTB    LKS             ;IS THE CLOCK FLAG SET?
002604  100404                                  BMI     D0011           ;IF SO, CONTINUE ON WITH THE TEST
002606  005203                                  INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
002610  001373                                  BNE     C0011           ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
002612  104001                          E10111: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
002614  000731                                  BR      T0011
002616                                  D00111:
002616  000240                                  NOP
002620  000240                                  NOP                     ;GIVE CLOCK EXTRA TIME TO INTERRUPT
002622  000401                                  BR      T0012
002624  104003                          E200111: ERROR 3                ;ERROR, CLOCK INTERRUPTED WITHOUT HAVING PRIORITY



                                        .SBTTL  TEST THAT RESET SETS CLOCK FLAG
                                        ;TEST THAT RESET SETS CLOCK FLAG
002626  000004                          T00121: SCOPE
002630  012737  000200  001124                  MOV     #200,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
002636  012737  005572  000004                  MOV     #TRAP0,004      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
002644  012737  000340  000006                  MOV     #340,006        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
002652  012737  002660  001136                  MOV     #R0012,$LPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
002660  005037  177546                  R00121: CLR     LKS             ;CLEAR CLOCK FLAG
002664  005003                                  CLR     R3              ;INITIALIZE A COUNTER LOCATION
002666  105737  177546                  A00121: TSTB    LKS             ;IS THE CLOCK FLAG SET?
002672  100404                                  BMI     B0012           ;IF SO, CONTINUE ON WITH THE TEST
002674  005203                                  INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
002676  001373                                  BNE     A0012           ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
002700  104001                          E00121: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
002702  000427                                  BR      T0013
002704                                  B00121:
002704  005037  177546                          CLR     LKS
002710  000005                                  RESET                   ;SHOULD SET CLOCK FLAG
002712  105737  177546                          TSTB    LKS
002716  100401                                  BMI     T0013
002720  104001                          E100121: ERROR 1                ;ERROR, RESET DIDN'T SET CLOCK FLAG



                                        .SBTTL  TEST LINE CLOCK REPEATABILITY
                                        ;TEST LINE CLOCK REPEATABILITY
```

C2

```
                                        ;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 > 20 MS
002766  000510                          BR      T0014
002770                          B00131:
002770  012737  002776  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 > 20 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 > 20 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 G. TEST THE CLOCK FLAG AGAIN, UNLESS...
```

```
 773  003132  104001                  E300131: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 30 MS
 774  003134  000425                           BR      .+14
 775  003136                          L300131:
 776  003136  012737  002724  001106           MOV     #RB013,SLPADR    ;MUST LOOP BACK TO BEGINING OF TEST IF ERROR COMES NOW
 777  003144  005037  177546                   CLR     LKS
 778  003150  105737  177546          M00131:  TSTB    LKS              ;IS CLOCK FLAG SET
 779  003154  100003                           BPL     N0013            ;NO
 780  003156  005202                           INC     R2               ;+1 TO CLOCK COUNT
 781  003160  005037  177546                   CLR     LKS              ;CLEAR CLOCK IF SET
 782  003164  005200                  N00131:  INC     R0               ;+1 TO TIME COUNT
 783  003166  001370                           BNE     M0013            ;REPEAT UNTIL R0=0
 784  003170  022102                           CMP     R1,R2            ;IS 1ST CLOCK COUNT EQUAL TO 2ND CLOCK COUNT?
 785  003172  001406                           BEQ     T0014            ;YES
 786  003174  010137  001156          E400131: MOV     R1,$REG1         ;GET R1 READY FOR PRINTOUT
 787  003200  010237  001160                   MOV     R2,$REG2         ;GET R2 READY FOR PRINTOUT
 788  003204  104004                           ERROR 4                  ;ERROR, CLOCK FLAG OCCURRED DIFFERENT
 789  003206  000240                           NOP                      ;NUMBER OF TIMES IN EQUAL PERIODS
 790
 791
 792
 793                                  .SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
 794                                  ;LINE CLOCK REGISTER ADDRESSING TEST
 795                                  ;TEST THAT THE "LKS" REGISTER CAN NOT BE ADDRESSED AS ANYTHING BUT 177546
 796                                  ;SET A LOCATION THAT IS CLOSE(DIFFERS BY 1 BIT) TO THE LKS REGISTER
 797                                  ;TO 100, IF THE LKS ALSO CHANGES, THEN SIGNAL AN ERROR
 798  003210  000004                  T00140:  SCOPE
 799  003212  005037  001124                   CLR     SGDDAT
 800  003216  012737  003254  001106           MOV     #RB014,SLPADR    ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
 801  003224  012737  157546  001120           MOV     #157546,SGDADR   ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 13 CLEAR
 802  003232  012737  003272  000004           MOV     #AB014,4         ;SETUP VECTOR IN CASE IT IS NONEXISTANT
 803  003240  012737  000340  000006           MOV     #340,6
 804  003246  012737  000340  177776           MOV     #340,@#PS        ;NO INTERRUPTS NOW
 805  003254  012706  001000          R00140:  MOV     #1000,SP         ;SETUP THE STACK
 806  003260  005037  177546                   CLR     LKS
 807  003264  012777  000100  175026  100140:  MOV     #100,@SGDADR     ;SET THE "CLOSE" ADDRESS TO = 100
 808  003272  032737  000100  177546  A00140:  BIT     #100,LKS         ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
 809  003300  001401                           BEQ     B0014
 810  003302  104005                  E00140:  ERROR 5                  ;IT AFFECTED "LKS"  ==  ERROR
 811  003304  005037  177546          B00140:  CLR     LKS
 812  003310  012737  003322  000004           MOV     #T0015  ,4
 813  003316  005077  175570                   CLR     @SGDADR          ;CLEAR OUT THE "CLOSE" ADDRESS
 814
 815
 816
 817                                  .SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
 818                                  ;LINE CLOCK REGISTER ADDRESSING TEST
 819  003322  000004                  T00150:  SCOPE
 820  003324  012737  003434  000004           MOV     #A0015,4         ;SETUP VECTOR IN CASE IT IS NONEXISTANT
 821  003332  012737  000340  000006           MOV     #340,6
 822  003340  005037  001124                   CLR     SGDDAT
 823  003344  012737  003366  001106           MOV     #RB015,SLPADR    ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
 824  003352  012737  177146  001120           MOV     #177146,SGDADR   ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 8 CLEAR
 825  003360  012737  000340  177776           MOV     #340,@#PS        ;NO INTERRUPTS NOW
 826  003366  012706  001000          R00150:  MOV     #1000,SP         ;SETUP THE STACK
```

```
                                        CLR     LKS
                        012777  000100  175514  100151  MOV     #100,@SGDADR    ;SET THE "CLOSE" ADDRESS TO = 100
                        032737  000100  177546  A00151  BIT     #100,LKS        ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
                                                        BEQ     B00151
                                                E00151  ERROR 5                 ;IT AFFECTED "LKS"  -- ERROR
                        005037  177546          B00151  CLR     LKS
                        012737  003434  000004          MOV     #T00110  ,4
                        005077  175464                  CLR     @SGDADR         ;CLEAR OUT THE "CLOSE" ADDRESS


                                        ,SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
                                        ;LINE CLOCK REGISTER ADDRESSING TEST
                                T00161: SCOPE
                        012737  003516  000004          MOV     #A00110,4       ;SETUP VECTOR IN CASE IT IS NONEXISTANT
                        012737  000340  000006          MOV     #340,6
                        005037  001124                  CLR     SGDDAT
                        012737  003500  001106          MOV     #R00110,SLPADR  ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
                        012737  177446  001120          MOV     #177446,SGDADR  ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 6 CLEAR
                        012737  000340  177776          MOV     #340,@#PS       ;NO INTERRUPTS NOW
                        012706  001000          R00161: MOV     #1000,SP        ;SETUP THE STACK
                        005037  177546                  CLR     LKS
                        012777  000100  175402  100161: MOV     #100,@SGDADR    ;SET THE "CLOSE" ADDRESS TO = 10F
                        032737  000100  177546  A00161: BIT     #100,LKS        ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
                                                        BEQ     B00116
                                                E00161: ERROR 5                 ;IT AFFECTED "LKS"  -- ERROR
                        005037  177546          B00161: CLR     LKS
                        012737  003546  000004          MOV     #T00117  ,4
                        005077  175352                  CLR     @SGDADR         ;CLEAR OUT THE "CLOSE" ADDRESS


                                        ,SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
                                        ;LINE CLOCK REGISTER ADDRESSING TEST
                                T00171: SCOPE
                        012737  003630  000004          MOV     #A00117,4       ;SETUP VECTOR IN CASE IT IS NONEXISTANT
                        012737  000340  000006          MOV     #340,6
                        005037  001124                  CLR     SGDDAT
                        012737  003612  001106          MOV     #R00117,SLPADR  ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
                        012737  177556  001120          MOV     #177556,SGDADR  ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 3 SET
                        012737  000340  177776          MOV     #340,@#PS       ;NO INTERRUPTS NOW
                        012706  001000          R00171: MOV     #1000,SP        ;SETUP THE STACK
                        005037  177546                  CLR     LKS
                        012777  000100  175270  100171: MOV     #100,@SGDADR    ;SET THE "CLOSE" ADDRESS TO = 100
                        032737  000100  177546  A00171: BIT     #100,LKS        ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
                                                        BEQ     B00117
                                                E00171: ERROR 5                 ;IT AFFECTED "LKS"  -- ERROR
                                                B00171: CLR     LKS
                        012737  003660  000004          MOV     #T00120  ,4
                        005077  175240                  CLR     @SGDADR         ;CLEAR OUT THE "CLOSE" ADDRESS


                                        ,SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
```

F2

```
                                        ;LINE CLOCK REGISTER ADDRESSING TEST
001663  012737  000004                  T00201: SCOPE
001664  012737  001742  000004                  MOV     #A00201,4       ;SETUP VECTOR IN CASE IT IS NONEXISTANT
001672  012737  000340  000006                  MOV     #340,6
001700  005037  001124                        CLR     SGDDAT
001704  012737  001724  001106                  MOV     #R00201,SLPADR  ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
001712  012737  177566  001120                  MOV     #177566,SGDADR  ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 4 SET
001720  012737  000340  177776                  MOV     #340,@#PS       ;NO INTERRUPTS NOW
001726  012706  001000                R00201: MOV     #1000,SP        ;SETUP THE STACK
001732  005037  177546                        CLR     LKS
001736  012777  000100  175156        100201: MOV     #100,@SGDADR    ;SET THE "CLOSE" ADDRESS TO = 100
001744  032737  000100  177546        A00201: BIT     #100,LKS        ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
001752  001401                              BEQ     E00201
001754  104005                        E00201: ERROR 5                 ;IT AFFECTED "LKS"  -- ERROR
001756  005037  177546                E00201: CLR     LKS
001762  012737  001772  000004                  MOV     #T00211 ,4
001770  005077  175126                        CLR     @SGDADR         ;CLEAR OUT THE "CLOSE" ADDRESS


                                        ;SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
                                        ;LINE CLOCK REGISTER ADDRESSING TEST
001776  000004                  T00211: SCOPE
002000  012737  002054  000004                  MOV     #A00211,4       ;SETUP VECTOR IN CASE IT IS NONEXISTANT
002006  012737  000340  000006                  MOV     #340,6
002014  005037  001124                        CLR     SGDDAT
002020  012737  002036  001106                  MOV     #R00211,SLPADR  ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
002026  012737  177746  001120                  MOV     #177746,SGDADR  ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 7 SET
002034  012737  000340  177776                  MOV     #340,@#PS       ;NO INTERRUPTS NOW
002042  012706  001000                R00211: MOV     #1000,SP        ;SETUP THE STACK
002046  005037  177546                        CLR     LKS
002052  012777  000100  175144        100211: MOV     #100,@SGDADR    ;SET THE "CLOSE" ADDRESS TO = 100
002060  032737  000100  177546        A00211: BIT     #100,LKS        ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
002066  001401                              BEQ     E00211
002070  104005                        E00211: ERROR 5                 ;IT AFFECTED "LKS"  -- ERROR
002072  005037  177546                E00211: CLR     LKS
002076  012737  002134  000004                  MOV     #T00231 ,4
002104  005077  175114                        CLR     @SGDADR         ;CLEAR OUT THE "CLOSE" ADDRESS


002110  000004                  T00221: SCOPE


                                        ;SBTTL  CLOCK FLAG BIT TEST
                                        ;CLOCK FLAG BIT TEST
002112  000004                  T00231: SCOPE
002114  012737  002572  000004                  MOV     #TRAP8,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
002122  012737  000340  000006                  MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
002130  012737  000200  001124                  MOV     #200,SGDDAT
002136  012737  002150  001106                  MOV     #R00231,SLPADR  ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERR
002144  005037  177546                R00231: CLR     LKS
002150  005003                              CLR     R3              ;INITIALIZE A COUNTER LOCATION
```

```
935  001046  105737  177546          A00231: TSTB    LKS             ;IS THE CLOCK FLAG SET?
936  001052  100404                  BMI     B0023                   ;IF SO, CONTINUE ON WITH THE TEST
937  001054  005203                  INC     R3                      ;IF NOT INCREMENT THE COUNTER LOCATION
938  001056  001373                  BNE     A0023                   ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
939  001060  104001          E00231: ERROR 1                         ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
940  001062  000410                  BR      T0024
941  001064                  B00231:
942  001064  012737  000200  177546  100231: MOV     #200,LKS        ;MOVE A 1 INTO THE CLOCK FLAG BIT
943  001072  023737  177546  001124          CMP     LKS,$GDDAT      ;SHOULD NOT AFFECT THE FLAG BIT
944  001100  001401                  BEQ     T0024
945  001102  104001          E100231:ERROR 1                         ;CLOCK FLAG DID NOT CLEAR
946
947
948
949                              .SBTTL  INTERRUPT TEST
950  001104  000004          T00241: SCOPE
951  001106  012737  005572  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
952  001114  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
953  001122  012737  000200  001124          MOV     #200,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
954  001130  012737  001250  001106          MOV     #R0024,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
955  001136  012737  001304  000100          MOV     #E0024,100
956  001144  005037  177546                  CLR     LKS             ;ALLOW CLOCK INTERRUPTS
957  001150  012737  000340  177776  R00241: MOV     #340,@#PS       ;NO INTERRUPTS NOW
958  001156  012737  000300  177546          MOV     #300,LKS
959  001164  005227  000030          A00241: INC     #30             ;WAIT FOR 20. MS
960  001170  001375                  BNE     A0024                   ;LOOP BACK IF NOT DONE WAITING
961  001172  000005                  RESET                           ;RESET SHOULD CLEAR INTERRUPT ENABLE
962  001174  223737  001124  177546          CMP     $GDDAT,LKS      ;AND LEAVE THE CLOCK FLAG SET
963  001202  001401                  BEQ     T0025                   ;GO ON TO THE NEXT TEST IF IT DID
964  001204  104001          E00241: ERROR 1                         ;RESET SET INTERRUPT ENABLE OR CLEARED CLOCK FLAG
965
966
967
968                              .SBTTL  NO SACK TIMEOUT TEST
969                              ;NO SACK TIMEOUT TEST
970  001206  000004          T00251: SCOPE
971  001210  012737  005572  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
972  001216  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
973  001224  012737  000300  001124          MOV     #300,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
974  001232  012737  001354  001106          MOV     #R0025,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
975  001240  012737  000340  000102          MOV     #340,102        ;NO INTERRUPTS AFTER THE FIRST ONE
976  001246  012737  001420  000100          MOV     #C0025,100
977  001254  005037  177546          R00251: CLR     LKS
978  001260  012737  000340  177776          MOV     #340,PS
979  001266  005003                  CLR     R3                      ;INITIALIZE A COUNTER LOCATION
980  001270  105737  177546          A00251: TSTB    LKS             ;IS THE CLOCK FLAG SET?
981  001274  100404                  BMI     B0025                   ;IF SO, CONTINUE ON WITH THE TEST
982  001276  005203                  INC     R3                      ;IF NOT INCREMENT THE COUNTER LOCATION
983  001300  001373                  BNE     A0025                   ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
984  001302  104001          E00251: ERROR 1                         ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
985  001304  000415                  BR      T0026
986  001306                  B00251:
987  001306  005037  177776          CLR     PS                      ;MAKE ALL INTERRUPTS OK
988  001312  012737  000100  177546          MOV     #100,LKS        ;ENABLE CLOCK INTERRUPTS
```

```
  989  004420  000001                          WAIT                    ;THE ONLY WAY TO LEAVE HERE WITHOUT ERROR IS TO INTERRUP
  990  004422  104006              E100251:    ERROR 6                 ;IF IT ERROR IS HERE ITS BECAUSE OF A "NO-SACK" TIMEOUT
  991  004424  000405                          BR      T0026
  992  004426  022737  000300  177546  C00251:  CMP     #300,LKS
  993  004434  001401                          BEQ     T0026           ;FIND OUT WHAT THE INTERRUPT DID TO THE CLOCK STATUS REG
  994  004436  104001              E200251:    ERROR 1                 ;IT CLEARED THE INTERRUPT ENABLE OR THE FLAG BIT
  995
  996
  997
  998                                          .SBTTL  RESET TEST
  999                                          ;RESET TEST
 1000  004440  000024              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,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
 1004  004464  012737  004514  001106          MOV     #R0020,SLPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
 1005  004472  012737  000340  177776          MOV     #340,@#PS       ;NO INTERRUPTS NOW
 1006  004500  012737  004552  000100          MOV     #E0020,100
 1007  004506  012737  000140  000102          MOV     #140,102        ;SETUP STATUS FOR AFTER THE INTERRUPT
 1008  004514  005037  177546        R00261:   CLR     LKS
 1009  004520  012706  001300                  MOV     #1000,SP        ;SETUP THE STACK
 1010  004524  012737  000100  177546          MOV     #100,LKS        ;SET INTERRUPT ENABLE BIT NOW
 1011  004532  005227  000000        A00261:   INC     #0              ;WAIT FOR CLOCK FLAG TO SET
 1012  004536  001375                          BNE     A0026
 1013  004540  000005                          RESET                   ;RESET SHOULD NOT CLEAR THE FLAG
 1014  004542  023737  177546  001124          CMP     LKS,$GDDAT      ;FIND OUT IF ID BIT 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,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
 1027  004600  012737  004620  001106          MOV     #R0027,SLPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
 1028  004606  005037  000102                  CLR     102
 1029  004612  012737  004676  000100          MOV     #T0030  ,100
 1030  004620  005037  177546        R00271:   CLR     LKS
 1031  004624  012737  000340  177776          MOV     #340,PS
 1032  004632  012706  001000                  MOV     #1000,SP        ;SETUP THE STACK
 1033  004636  005037  001230                  CLR     WORD            ;SETUP A COUNTER LOCATION TO @ 0
 1034  004642  005237  001230        A00271:   INC     WORD
 1035  004646  001375                          BNE     A0027           ;WASTE TIME LOOPING UNTIL THE COUNTER REACHES 0
 1036  004650  012737  000300  177546          MOV     #300,LKS        ;SET INTERRUPT ENABLE AND TRY TO SET THE CLOCK FLAG
 1037  004656  012737  000200  177546          MOV     #200,LKS        ;TRY TO SET IT AGAIN
 1038  004664  023737  177546  001124          CMP     LKS,$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
```

```
1043
1044                                        .SBTTL  CLOCK FLAG AFTER INTERRUPT TEST
1045                                        ;SEE IF AN INTERRUPT CLEARS THE CLOCK FLAG
1046  004676  000004                  T00301: SCOPE
1047  004700  012737  005572  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
1048  004706  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
1049  004714  005037  177546                  CLR     LKS             ;NO CLOCK INTERRUPTS BEFORE WE ARE READY
1050  004720  012737  000100  001124          MOV     #100,SCDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
1051  004726  012737  004734  001106          MOV     #R0030,SLPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
1052  004734  012737  004776  000100  R00301: MOV     #A0030,100      ;SETUP CLOCK INTERRUPT VECTOR
1053  004742  005037  000102                  CLR     102             ;PRIORITY LEVEL WILL ALLOW FURTHER INTERRUPTS
1054  004746  005037  177776                  CLR     PS
1055  004752  012706  001000                  MOV     #1000,SP        ;SETUP THE STACK
1056  004756  005037  177546                  CLR     LKS
1057  004762  105037  001230                  CLRB    WORD            ;CLEAR OUT A COUNTER LOCATION
1058  004766  012737  000100  177546          MOV     #100,LKS        ;ENABLE CLOCK INTERRUPTS NOW
1059  004774  000001                          WAIT                    ;WAIT FOR AN INTERRUPT
1060  004776  012737  005024  000100  A00301: MOV     #E0030,100      ;ERROR IF WE INTERRUPT AGAIN
1061  005004  005037  177776                  CLR     PS              ;LET INTERRUPTS HAPPEN NOW
1062  005010  105237  001230          B00301: INCB    WORD            ;DO A NOTHING LOOP FOR A VERY SHORT PERIOD OF TIME
1063  005014  001375                          BNE     B0030           ;WE SHOULD INCREMENT TO 0 LONG BEFORE AN INTERRUPT COMES
1064  005016  005037  177546                  CLR     LKS
1065  005022  000401                          BR      T0031
1066  005024  104001          E00331: ERROR 1                         ;INTERRUPT DID NOT CLEAR THE CLOCK FLAG
1067
1068
1069
1070                                        .SBTTL  NO INTERRUT AT PRIORITY 7 TEST
1071                                        ;TEST THAT CLOCK WILL NOT INTERRUPT WITH PROCESSR AT PRIORITY 7
1072  005026  000004                  T00311: SCOPE
1073  005030  012737  005572  000004          MOV     #TRAP0,@#4      ;SETJP VECOR IN CASE OF UNFORSEEN PROBLEMS
1074  005036  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
1075  005044  012737  005052  001106          MOV     #R0031,SLPADR   ;SETJP LOOPBACK ADDRESS IN CASE OF AN ERROR
1076  005052  005037  177546          R00311: CLR     LKS
1077  005056  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
1078  005060  105737  177546          A00311: TSTB    LKS             ;IS THE CLOCK FLAG SET?
1079  005064  100404                          BMI     B0031           ;IF SO, CONTINUE ON WITH THE TEST
1080  005066  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
1081  005070  001373                          BNE     A0031           ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
1082  005072  104001          E00311: ERROR 1                         ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
1083  005074  000427                          BR      T0032
1084  005076                  B00311:
1085  005076  012706  001000                  MOV     #1000,SP        ;INITIALIZE THE STACK POINTER
1086  005102  012737  000340  177776          MOV     #340,PS         ;SET PRIORITY 7
1087  005110  012737  005072  000100          MOV     #E0031,100      ;SET UP VECTOR RETURN
1088  005116  012737  000100  177546          MOV     #100,LKS        ;ENABLE INTERRUPT
1089  005124  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
1090  005126  105737  177546          C00311: TSTB    LKS             ;IS THE CLOCK FLAG SET?
1091  005132  100404                          BMI     D0031           ;IF SO, CONTINUE ON WITH THE TEST
1092  005134  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
1093  005136  001373                          BNE     C0031           ;AND GO TEST THE CLOCK FLAG AGAIN, UNLESS...
1094  005140  104001          E10311: ERROR 1                         ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
1095  005142  000404                          BR      T0032
1096  005144                  D00311:
```

J2

```
1097  005144  000240                      NOP
1098  005146  000240                      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 INTERRJPT PUSHES CONDITION CODES ONTO STACK
1106  005154  000004              T0032:  SCOPE
1107  005156  012737  005572  000004      MOV     #TRAP0,@#4      ;SETJP VECOR IN CASE OF JNFORSEEN PROBLEMS
1108  005164  012737  000340  177776      MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
1109  005172  012737  005200  001106      MOV     #R0032,SLPADR   ;SETJP 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  005242  012737  005260  000100      MOV     #C0032,100      ;SET UP VECTOR RETURN
1123  005246  012737  000150  177546      MOV     #150,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,$GOOAT
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      ;SETJP VECOR IN CASE OF JNFORSEEN PROBLEMS
1139  005320  012737  000340  000006      MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
1140  005326  012737  005334  001106      MOV     #R0033,SLPADR   ;SETJP 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
```

K1

K2

```
1151  005364  005037  000776                  CLR     BUF1
1152  005370  005037  000774                  CLR     BUF2
1153  005374  012737  005422  000100          MOV     #C0033,100      ;SET UP VECTOR RETURN
1154  005402  012737  000100  177546          MOV     #100,LKS        ;ENABLE INTERRUPT
1155  005410  012737  000200  177776          MOV     #200,PS         ;SET PRIORITY 4
1156  005416  000277                          ◆SEC!SEV!SEZ!SEN        ;SET ALL CONDITION CODES
1157  005420  000001                          WAIT                    ;WAIT FOR INTERRUPT
1158  005422  022737  005422  000774  C00331  CMP     #C0033,BUF2
1159  005430  001404                          BEQ     T0034
1160  005432  012737  005422  001124  E100331 MOV     #C0033,SCODAT
1161  005440  104010                          ERROR 10                ;ERROR, DID NOT PUSH CORRECT PC ONTO STACK
1162
1163
1164
1165                                  ,SBTTL  END OF PASS INDICATING
1166  005442  000004                T00341  SCOPE
1167  005444  005037  177546                  CLR     LKS             ;TURN THE CLOCK OFF
1168  005450  000005                          RESET                   ;TURN EVERYTHING OFF
1169
1170                                  ;;**************************************************************
1171
1172                                  ,SBTTL  END OF PASS ROUTINE
1173
1174                                  ;◆INCREMENT THE PASS NUMBER (SPASS)
1175                                  ;◆TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
1176                                  ;◆IF THERES A MONITOR GO TO IT
1177                                  ;◆IF THERE ISN'T JUMP TO T0001
1178
1179  005452                        SEOPI
1180  005452  000004                        SCOPE
1181  005454  005037  001132                CLR     STSTNM          ;ZERO THE TEST NUMBER
1182  005460  005037  001214                CLR     STIMES          ;ZERO THE NUMBER OF ITERATIONS
1183  005464  005237  001130                INC     SPASS           ;INCREMENT THE PASS NUMBER
1184  005470  042737  100000  001130        BIC     #100000,SPASS   ;DON'T ALLOW A NEG. NUMBER
1185  005476  005327                        DEC     (PC)+           ;LOOP?
1186  005500  000001                SEOPCT: ,WORD   1
1187  005502  003021                        BGT     SDOAGN          ;YES
1188  005504  012737                        MOV     (PC)+,@(PC)+    ;RESTORE COUNTER
1189  005506  000001                SENDCT: ,WORD   1
1190  005510  005500                        SEOPCT
1191  005512  104400  005552                TYPE    ,SENDMG         ;TYPE "END PASS #"
1192  005516  013746  001100                MOV     SPASS,-(SP)     ;SAVE SPASS FOR TYPEOUT
1193  005522  104410                        TYPOS                   ;GO TYPE-◆DECIMAL ASCII WITH SIGN
1194  005524  104400  005567                TYPE    ,SENULL         ;TYPE A NULL CHARACTER
1195  005530  013700  000042        SGET42: MOV     @#42,R0         ;GET MONITOR ADDRESS
1196  005534  001404                        BEQ     SDOAGN          ;IF NONE
1197  005536  004710                SENDAD: JSR     PC,(R0)         ;GO TO MONITOR
1198  005542  000240                        NOP                     ;SAVE ROOM
1199  005542  000240                        NOP                     ;FOR
1200  005544  000240                        NOP                     ;ACT11
1201  005546  002137  001550        SDOAGN: JMP     @#T0001         ;RETURN
1202  005552  005015  047105  020104  SENDMG: ,ASCIZ  <15><12>/END PASS #/
1203  005560  040520  051523  021440
1204  005566     000
```

```
1205  005567  000377  000377  000  $ENULL1: .BYTE   -1,-1,0                ;NULL CHARACTER STRING
1206  005572  005046           TRAP3I: CLR     -(SP)
1207  005574  004737  006132           JSR     PC,$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    SERFLG            ;HAS AN ERROR OCCURRED?
1257  005754  001421                   BEQ     3$                ;BR IF NO
1258  005756  123737  001115  001103   CMPB    $ERMAX,SERFLG     ;MAX. ERRORS FOR THIS TEST OCCURRED?
```

```
1259  000764  101015                        BNE     3$                      ;BR IF NO
1260  000766  032737  010000  177570        BIT     #SW09,@#SWR             ;LOOP ON ERROR?
1261  000774  001404                        BEQ     4$                      ;BR IF NO
1262  000776  013737  011110  011106  7$1   MOV     $LPERR,$LPADR           ;SET LOOP ADDRESS TO LAST SCOPE
1263  001004  002443                        RR      $OVER
1264  001006  105037  001123          4$1   CLRB    $ERFLG                  ;ZERO THE ERROR FLAG
1265  001012  005037  001214                CLR     $TIMES                  ;CLEAR THE NUMBER OF ITERATIONS TO MAKE
1266  001016  002415                        RR      1$                      ;ESCAPE TO THE NEXT TEST
1267  001022  032737  004000  177570  3$1   BIT     #SW11,@#SWR             ;INHIBIT ITERATIONS?
1270  001026  001011                        BNE     1$                      ;BR IF YES
1271  001030  005737  001120                TST     $PASS                   ;IF FIRST PASS OF PROGRAM
1272  001034  001406                        BEQ     1$                      ;       INHIBIT ITERATIONS
1273  001036  005237  001124                INC     $ICNT                   ;INCREMENT ITERATION COUNT
1274  001042  023737  001214  001124        CMP     $TIMES,$ICNT            ;CHECK THE NUMBER OF ITERATIONS MADE
1275  001050  002021                        BGE     $OVER                   ;BR IF MORE ITERATION REQUIRED
1276  001052  012737  000001  001124  1$1   MOV     #1,$ICNT                ;REINITIALIZE THE ITERATION COUNTER
1277  001060  013737  001130  001214        MOV     $MXCNT,$TIMES           ;SET NUMBER OF ITERATIONS TO DO
1300  001066  105237  001122          $SVLAD: INCB    $TSTNM                  ;COUNT TEST NUMBERS
1301  001072  011637  001106                MOV     (SP),$LPADR             ;SAVE SCOPE LOOP ADDRESS
1302  001076  011637  001110                MOV     (SP),$LPERR             ;SAVE ERROR LOOP ADDRESS
1303  001102  005037  001210                CLR     $ESCAPE                 ;CLEAR THE ESCAPE FROM ERROR ADDRESS
1304  001106  112737  000001  001115        MOVB    #1,$ERMAX               ;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
1305  001114  013737  001122  177570  $OVER: MOV     $TSTNM,@#DISPLAY        ;DISPLAY TEST NUMBER
1306  001122  013716  001100                MOV     @$LPADR,(SP)            ;FUDGE RETURN ADDRESS
1307  001126  000002                        RTI                             ;FIXES PS
1310  001132  000010                $MXCNT: 10                              ;MAX. NUMBER OF ITERATIONS
1311                                        ;;*****************************************************************
1312
1313                                        .SBTTL  TYPE ROUTINE
1314
1315                                        ;+ROUTINE TO TYPE ASCII MESSAGE, MESSAGE MUST TERMINATE WITH A 0 BYTE.
1316                                        ;+THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
1317                                        ;+NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
1320                                        ;+NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
1321                                        ;+NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
1322                                        ;+
1323                                        ;+CALL:
1324                                        ;+1) USING A TRAP INSTRUCTION
1325                                        ;+      TYPE    ,MESADR                 ;MESADR IS FIRST ADDRESS OF AN ASCII STRING
1326                                        ;+OR
1327                                        ;+      TYPE
1330                                        ;+      MESADR
1331                                        ;+
1332                                        ;+2) USING A JSR INSTRUCTION
1333                                        ;+      MOV     PS,-(SP)                ;PUSH PROCESSOR STATUS WORD ON THE STACK
1334                                        ;+      JSR     PC,$TYPE                ;CALL TYPE ROUTINE
1335                                        ;+      MESADDR                         ;FIRST ADRESS OF MESSAGE
1336
1337  001132  105737  001151          $TYPE1: TSTB    $TPFLG                  ;IS THERE A TERMINAL?
1340  001136  100402                        BPL     1$                      ;BR IF YES
1341  001140  000000                        HALT                            ;HALT HERE IF NO TERMINAL
1342  001142  000207                        RR      3$                      ;LEAVE
1343  001144  010046                1$1     MOV     R0,-(SP)                ;SAVE R0
1344  001146  017600  000002                MOV     @2(SP),R0               ;GET ADDRESS OF ASCII STRING
```

```
1313  006152  117646                  251     MOVB    (R0)+,-(SP)             ;PUSH CHARACTER TO BE TYPED ONTO STACK
1314  006154  001375                          BNE     4$                      ;BR IF IT ISN'T THE TERMINATOR
1315  006156  005726                          TST     (SP)+                   ;IF TERMINATOR POP IT OFF THE STACK
1316  006160  012600                          MOV     (SP)+,R0                ;RESTORE R0
1317  006162  062716  000002                351     ADD     #2,(SP)                 ;ADJUST RETURN PC
1318  006166  000002                          RTI                             ;RETURN
1319  006170  004737  006222                451     JSR     PC,7$                   ;GO TYPE THIS CHARACTER
1320  006174  123726  001150                551     CMPB    $FILLC,(SP)+            ;IS IT TIME FOR FILLER CHARS.?
1321  006200  001364                          BNE     2$                      ;IF NO GO GET NEXT CHAR.
1322  006202  013746  001146                MOV     $NULL,-(SP)             ;GET # OF FILLER CHARS, NEEDED
1323                                                                           ;AND THE NULL CHAR.
1324  006206  105366  000001                651     DECB    1(SP)                   ;DOES A NULL NEED TO BE TYPED?
1325  006212  002773                          BLT     5$                      ;BR IF NO--GO POP THE NULL OFF OF STACK
1326  006214  004737  006222                JSR     PC,7$                   ;GO TYPE A NULL
1327  006220  000772                          BR      6$                      ;LOOP
1328  006222  105777  172714                751     TSTB    @$TPS                   ;WAIT UNTIL PRINTER IS READY
1329  006226  100375                          BPL     7$
1330  006230  116677  000002  172706          MOVB    2(SP),@$TPB             ;LOAD CHAR TO BE TYPED INTO DATA REG.
1331  006236  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  006240                        $TYPDS:
1341  006240  010046                          MOV     R0,-(SP)                ;PUSH R0 ON STACK
1342  006242  010146                          MOV     R1,-(SP)                ;PUSH R1 ON STACK
1343  006244  010246                          MOV     R2,-(SP)                ;PUSH R2 ON STACK
1344  006246  010346                          MOV     R3,-(SP)                ;PUSH R3 ON STACK
1345  006250  010546                          MOV     R5,-(SP)                ;PUSH R5 ON STACK
1346  006252  012746  020200                MOV     #20200,-(SP)            ;SET BLANK SWITCH AND SIGN
1347  006256  016605  000020                MOV     20(SP),R5               ;GET THE INPUT NUMBER
1348  006262  100004                          BPL     1$                      ;BR IF INPUT IS POS.
1349  006264  005405                          NEG     R5                      ;MAKE THE BINARY NUMBER POS.
1350  006266  112766  000055  000001        MOVB    #'-,1(SP)               ;MAKE THE ASCII NUMBER NEG.
1351  006274  005000                  1$:     CLR     R0                      ;ZERO THE CONSTANTS INDEX
1352  006276  012703  006454                MOV     #$DBLK,R3               ;SETUP THE OUTPUT POINTER
1353  006302  112723  000040                MOVB    #' ,(R3)+               ;SET THE FIRST CHARACTER TO A BLANK
1354  006306  005002                  2$:     CLR     R2                      ;CLEAR THE BCD NUMBER
1355  006310  016001  006444                MOV     $DTBL(R0),R1            ;GET THE CONSTANT
1356  006314  160105                  3$:     SUB     R1,R5                   ;FORM THIS BCD DIGIT
1357  006316  002402                          BLT     4$                      ;BR IF DONE
1358  006320  005202                          INC     R2                      ;INCREASE THE BCD DIGIT BY 1
1359  006322  000774                          BR      3$
1360  006324  060105                  4$:     ADD     R1,R5                   ;ADD BACK THE CONSTANT
1361  006326  005702                          TST     R2                      ;CHECK IF BCD DIGIT=0
1362  006330  001002                          BNE     5$                      ;FALL THROUGH IF 0
1363  006332  105716                          TSTB    (SP)                    ;STILL DOING LEADING 0'S?
1364  006334  100407                          BMI     7$                      ;BR IF YES
1365  006336  106316                  5$:     ASLB    (SP)                    ;MSD?
1366  006340  103003                          BCC     6$                      ;BR IF NO
```

```
1367  000342  116603  000001  177777          MOVB    1(SP),-1(R3)            ;YES--SET THE SIGN
1368  000346  052702  000060          6$1     BIS     #'0,R2                  ;MAKE THE BCD DIGIT ASCII
1369  000352  052702  000040          7$1     BIS     #' ,R2                  ;MAKE IT A SPACE IF NOT ALREADY A DIGIT
1370  000356  110223                          MOVB    R2,(R3)+                ;PUT THIS CHARACTER IN THE OUTPUT BUFFER
1371  000360  005720                          TST     (R0)+                   ;JUST INCREMENTING
1372  000362  026027  000027  000010          CMP     R0,#10                  ;CHECK THE TABLE INDEX
1373  000370  002746                          BLT     2$                      ;GO DO THE NEXT DIGIT
1374  000372  003002                          BGT     8$                      ;GO TO EXIT
1375  000374  016502                          MOV     R5,R2                   ;GET THE LSD
1376  000376  000764                          BR      6$                      ;GO CHANGE TO ASCII
1377  000400  105726                  8$1     TSTB    (SP)+                   ;WAS THE LSD THE FIRST NON-ZERO?
1378  000402  100003                          BPL     9$                      ;BR IF NO
1379  000404  116603  177777  177776          MOVB    -1(SP),-2(R3)           ;YES--SET THE SIGN FOR TYPING
1380  000412  105013                  9$1     CLRB    (R3)                    ;SET THE TERMINATOR
1381  000414  012605                          MOV     (SP)+,R5                ;POP STACK INTO R5
1382  000416  012603                          MOV     (SP)+,R3                ;POP STACK INTO R3
1383  000420  012602                          MOV     (SP)+,R2                ;POP STACK INTO R2
1384  000422  012601                          MOV     (SP)+,R1                ;POP STACK INTO R1
1385  000424  012600                          MOV     (SP)+,R0                ;POP STACK INTO R0
1386  000426  104403  000454                  TYPE    ,$DBLK                  ;NOW TYPE THE NUMBER
1387  000432  016666  000002  000004          MOV     2(SP),4(SP)             ;ADJUST THE STACK
1388  000440  012616                          MOV     (SP)+,(SP)
1389  000442  000002                          RTI                             ;RETURN TO USER
1390  000444  023420                  $DTBL:  10000.
1391  000446  001750                          1000.
1392  000450  000144                          100.
1393  000452  000012                          10.
1394  000454  000004                  $DBLK:  .BLKW   4
1395                                  ;;*******************************************************************
1396
1397                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
1398
1399                                  ;+STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
1400                                  ;+CALL:
1401                                  ;+      MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
1402                                  ;+      TYPOS                           ;CALL FOR TYPEOUT
1403                                  ;+      .BYTE   N                       ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
1404                                  ;+      .BYTE   M                       ;M=1 OR 0
1405                                  ;+                                      ;1=TYPE LEADING ZEROS
1406                                  ;+                                      ;0=SUPPRESS LEADING ZEROS
1407                                  ;+
1408                                  ;+STYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
1409                                  ;+STYPOS OR STYPOC
1410                                  ;+CALL:
1411                                  ;+      MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
1412                                  ;+      TYPON                           ;CALL FOR TYPEOUT
1413                                  ;+
1414                                  ;+STYPOC---ENTER HERE FOR TYPEO ' OF A 16 BIT NUMBER
1415                                  ;+CALL:
1416                                  ;+      MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
1417                                  ;+      TYPOC                           ;CALL FOR TYPEOUT
1418
1419  000464  017646  177770          STYPOS: MOV     @(SP),-(SP)             ;PICKUP THE MODE
1420  000472  116637  000001  000737          MOVB    1(SP),$BFILL            ;LOAD ZERO FILL SWITCH
```

```
1421  000476  112637  000711            MOVB    (SP)+,$OMODE+1          ;NUMBER OF DIGITS TO TYPE
1422  000502  062716  000002            ADD     #2,(SP)                 ;ADJUST RETURN ADDRESS
1423  000506  000404                    BR      $TYPON
1424  000510  112737  000001  000707  $TYPOC:  MOVB    #1,$OFILL               ;SET THE ZERO FILL SWITCH
1425  000516  112737  000006  000711          MOVB    #6,$OMODE+1             ;SET FOR SIX(6) DIGITS
1426  000524  112737  000005  000706  $TYPDN:  MOVB    #5,$OCNT                ;SET THE ITERATION COUNT
1427  000532  010346                    MOV     R3,-(SP)                ;SAVE R3
1428  000534  010446                    MOV     R4,-(SP)                ;SAVE R4
1429  000536  010546                    MOV     R5,-(SP)                ;SAVE R5
1430  000540  113704  000711            MOVB-   $OMODE+1,R4             ;GET THE NUMBER OF DIGITS TO TYPE
1431  000544  005404                    NEG     R4
1432  000546  062704  000006            ADD     #6,R4                   ;SUBTRACT IT FOR MAX, ALLOWED
1433  000552  110437  000710            MOVB    R4,$OMODE               ;SAVE IT FOR USE
1434  000556  113704  000707            MOVB    $OFILL,R4               ;GET THE ZERO FILL SWITCH
1435  000562  016605  000012            MOV     12(SP),R5               ;PICKUP THE INPUT NUMBER
1436  000566  005003                    CLR     R3                      ;CLEAR THE OUTPUT WORD
1437  000570  006105            1$:     ROL     R5                      ;ROTATE MSB INTO "C"
1438  000572  000404                    BR      3$                      ;GO DO MSB
1439  000574  006105            2$:     ROL     R5                      ;FORM THIS DIGIT
1440  000576  006103                    ROL     R3
1441  000600  006105                    ROL     R5
1442  000602  010503                    MOV     R5,R3
1443  000604  006103            3$:     ROL     R3                      ;GET LSB OF THIS DIGIT
1444  000606  105337  000710            DECB    $OMODE                  ;TYPE THIS DIGIT?
1445  000612  100016                    BPL     7$                      ;BR IF NO
1446  000614  042703  177770            BIC     #177770,R3              ;GET RID OF JUNK
1447  000620  001002                    BNE     4$                      ;TEST FOR 0
1448  000622  005704                    TST     R4                      ;SUPPRESS THIS 0?
1449  000624  001403                    BEQ     5$                      ;BR IF YES
1450  000626  005204            4$:     INC     R4                      ;DON'T SUPPRESS ANYMORE 0'S
1451  000630  052703  000060            BIS     #'0,R3                  ;MAKE THIS DIGIT ASCII
1452  000634  052703  000040    5$:     BIS     #' ,R3                  ;MAKE ASCII IF NOT ALREADY
1453  000640  110337  000704            MOVB    R3,@S                   ;SAVE FOR TYPING
1454  000644  104401                    TYPE    ,@S                     ;GO TYPE THIS DIGIT
1455  000650  105337  000706    7$:     DECB    $OCNT                   ;COUNT BY 1
1456  000654  003347                    BGT     2$                      ;BR IF MORE TO DO
1457  000656  002402                    BLT     6$                      ;BR IF DONE
1458  000660  005204                    INC     R4                      ;INSURE LAST DIGIT ISN'T A BLANK
1459  000662  000744                    BR      2$                      ;GO DO THE LAST DIGIT
1460  000664  012605            6$:     MOV     (SP)+,R5                ;RESTORE R5
1461  000666  012604                    MOV     (SP)+,R4                ;RESTORE R4
1462  000670  012603                    MOV     (SP)+,R3                ;RESTORE R3
1463  000672  016666  000002  000004    MOV     2(SP),4(SP)             ;SET THE STACK FOR RETURNING
1464  000700  012616                    MOV     (SP)+,(SP)
1465  000702  000002                    RTI                             ;RETURN
1466  000704  000     000     000    @S:      .BYTE   0                       ;STORAGE FOR ASCII DIGIT
1467  000705  200     200     200            .BYTE   200                     ;TERMINATOR FOR TYPE ROUTINE
1468  000706  000     000     000    $OCNT:   .BYTE   0                       ;OCTAL DIGIT COUNTER
1469  000707  000     000     000    $OFILL:  .BYTE   0                       ;ZERO FILL SWITCH
1470  000710  000000          000    $OMODE:  0                       ;NUMBER OF DIGITS TO TYPE
1471                                 ;;*****************************************************************
1472
1473                                 .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
1474
```

```
1475                                    ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" (SITEMB) TO DETERMINE WHICH
1476                                    ;*ERROR IS TO BE REPORTED, IT THEN OBTAINS, FROM THE "ERROR TABLE" (SERRTB),
1477                                    ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
1478
1479  001712                            SERRTYP:
1480  001712  104420  001225                  TYPE    ,SCRLF                  ;"CARRIAGE RETURN" & "LINE FEED"
1481  001716  010046                           MOV     R0,-(SP)                ;SAVE R0
1482  001720  005000                           CLR     R0                      ;PICKUP THE ITEM INDEX
1483  001722  153700  001114                 BISB    @#SITEMB,R0
1484  001726  001004                           BNE     1$                      ;IF ITEM NUMBER IS ZERO, JUST
1485                                                                            ;TYPE THE PC OF THE ERROR
1486  001730  013746  001110                 MOV     SERRPC,-(SP)            ;SAVE SERRPC FOR TYPEOUT
1487                                                                            ;ERROR ADDRESS
1488  001734  104402                           TYPOC                           ;GO TYPE--OCTAL ASCII(ALL DIGITS)
1489  001736  000426                           BR      6$                      ;GET OUT
1490  001740  005300                    1$:    DEC     R0                      ;ADJUST THE INDEX SO THAT IT WILL
1491  001742  006300                           ASL     R0                      ;       WORK FOR THE ERROR TABLE
1492  001744  006300                           ASL     R0
1493  001746  006300                           ASL     R0
1494  001750  062700  001232                 ADD     #SERRTB,R0              ;FORM TABLE POINTER
1495  001754  012037  001764                 MOV     (R0)+,2$                ;PICKUP "ERROR MESSAGE" POINTER
1496  001760  001424                           BEQ     3$                      ;SKIP TYPEOUT IF NO POINTER
1497  001762  104403                           TYPE                            ;TYPE THE "ERROR MESSAGE"
1498  001764  000000                    2$:    ,WORD   0                       ;"ERROR MESSAGE" POINTER GOES HERE
1499  001766  104420  001225                 TYPE    ,SCRLF                  ;"CARRIAGE RETURN" & "LINE FEED"
1500  001772  012037  001762                 MOV     (R0)+,4$                ;PICKUP "DATA HEADER" POINTER
1501  001776  001424                           BEQ     5$                      ;SKIP TYPEOUT IF 0
1502  002000  104403                           TYPE                            ;TYPE THE "DATA HEADER"
1503  002002  000000                    4$:    ,WORD   0                       ;"DATA HEADER" POINTER GOES HERE
1504  002004  104420  001225                 TYPE    ,SCRLF                  ;"CARRIAGE RETURN" & "LINE FEED"
1505  002010  011000                    5$:    MOV     (R0),R0                 ;PICKUP "DATA TABLE" POINTER
1506  002012  001004                           BNE     7$                      ;GO TYPE THE DATA
1507  002014  012600                    6$:    MOV     (SP)+,R0                ;RESTORE R0
1508  002016  104420  001225                 TYPE    ,SCRLF                  ;"CARRIAGE RETURN" & "LINE FEED"
1509  002022  000207                           RTS     PC                      ;RETURN
1510  002024                            7$:
1511  002024  013046                           MOV     @(R0)+,-(SP)            ;SAVE @(R0)+ FOR TYPEOUT
1512  002026  104402                           TYPOC                           ;GO TYPE--OCTAL ASCII(ALL DIGITS)
1513  002030  005710                           TST     (R0)                    ;IS THERE ANOTHER NUMBER?
1514  002032  001770                           BEQ     6$                      ;BR IF NO
1515  002034  104403  002042                 TYPE    ,8$                     ;TYPE TWO(2) SPACES
1516  002040  000771                           BR      7$                      ;LOOP
1517  002042  020040  000     8$:    ,ASCIZ  /  /                            ;TWO(2) SPACES
1518  002046                                   .EVEN
1519                                    ;;*****************************************************************
1520
1521                                    .SBTTL  ERROR HANDLER ROUTINE
1522
1523                                    ;*SW15=1        HALT ON ERROR
1524                                    ;*SW13=1        INHIBIT ERROR TYPEOUTS
1525                                    ;*SW12=1        BELL ON ERROR
1526                                    ;*SW09=1        LOOP ON ERROR
1527                                    ;*GO TO SERRTYP ON ERROR
1528
```

```
1529  001046                                SERROR:
1530  001046  010637  001178                      MOV     SP,$REG6        ;GET THE CURRENT STACK POINTER VALUE
1531  001052  162737  000004  001178              SUB     #4,$REG6                ;RESTORE IT TO ITS "PRE ERROR TRAP" VALUE FOR PR
1532  001060  016637  000002  001172              MOV     2(SP),$REG7             ;GET THE PS OFF OF THE STACK
1533  001066  005037  001166                      CLR     $REG5                   ;PREPARE "$REG5" TO HOLD THE TEST #
1534  001072  113737  001132  001166              MOVB    $TSTNM,$REG5            ;TEST # IS HELD IN THE LOW BYTE OF "TSTNM"
1535  001100  010037  001154                      MOV     R0,$REG0                ;MOST OF THE TIME R0 HAS GOOD STUFF IN IT ALSO
1536  001104  105237  001103              7$:     INCB    SERFLG                  ;SET THE ERROR FLAG
1537  001110  001775                      10$:    BEQ     7$                      ;DON'T LET THE FLAG GO TO ZERO
1538  001112  013737  001132  177570              MOV     $TSTNM,@#DISPLAY        ;DISPLAY TEST NUMBER AND ERROR FLAG
1539  001120  032737  002000  177570              BIT     #SW10,@#SWR             ;BELL ON ERROR?
1540  001126  001402                              BEQ     1$                      ;NO = SKIP
1541  001130  104400  001220                      TYPE    ,$BELL                  ;RING BELL
1542  001134  005237  001112              1$:     INC     $ERTTL                  ;COUNT THE NUMBER OF ERRORS
1543  001140  011637  001116                      MOV     (SP),$ERRPC             ;GET ADDRESS OF ERROR INSTRUCTION
1544  001144  162737  000002  001116              SUB     #2,$ERRPC
1545  001152  117737  171746  001114              MOVB    @$ERRPC,$ITEMB          ;STRIP AND SAVE THE ERROR ITEM CODE
1546  001160  032737  020000  177570              BIT     #SW13,@#SWR             ;SKIP TYPEOUT IF SET
1547  001166  001004                              BNE     2$                      ;SKIP TYPEOUTS
1548  001170  004737  000712                      JSR     PC,@#$ERRTYP            ;GO TO USER ERROR ROUTINE
1549  001174  104401  001225                      TYPE    ,$CRLF
1550  001200  005737  177570              2$:     TST     @#SWR                   ;HALT ON ERROR
1551  001204  100001                              BPL     3$                      ;SKIP IF CONTINUE
1552  001206  000000                              HALT                            ;HALT ON ERROR!
1553  001210  032737  001000  177570      3$:     BIT     #SW9,@#SWR              ;LOOP ON ERROR SWITCH SET?
1554  001216  001402                              BEQ     4$                      ;BR IF NO
1555  001220  013716  001114                      MOV     $LPERR,(SP)             ;FUDGE RETURN FOR LOOPING
1556  001224  005737  001210              4$:     TST     $ESCAPE                 ;CHECK FOR AN ESCAPE ADDRESS
1557  001230  001402                              BEQ     5$                      ;BR IF NONE
1558  001232  013716  001210                      MOV     $ESCAPE,(SP)            ;FUDGE RETURN ADDRESS FOR ESCAPE
1559  001236  000002                      5$:     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  001240  010046                      $TRAP:  MOV     R0,-(SP)                ;SAVE R0
1570  001242  016600  000002                      MOV     2(SP),R0                ;GET TRAP ADDRESS
1571  001246  005740                              TST     -(R0)                   ;BACKUP BY 2
1572  001250  111000                              MOVB    (R0),R0                 ;GET RIGHT BYTE OF TRAP
1573  001252  016000  001260                      MOV     $TRPAD(R0),R0           ;INDEX TO TABLE
1574  001256  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  007062                           MOV     POWPUS,-(SP)            ;PUSH POWPUS ON STACK
007326  013746  007064                           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  005015
1639  007516  000012
1640  007520  005015  046412  026004  STMES1  ,ASCIZ  <15><12><12>"MD-11-DDKWA-A LINE FREQUENCY CLOCK TEST"<15><12>
1641  007526  030461  042055  045004
1642  007534  040527  040455  040040
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  040517  040040
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  040501
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  020041  020040  024040
1674  007770  051103  020051  020040
1675  007776  052040  051505  021524
1676  010024  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,SCDDAT
1682  010036  001166  177546  001124
1683  010044  000300                  #
1684  010046  046103  041517  020113  EM21    ,ASCIZ  "CLOCK FAILED TO INTERRUPT"
1685  010054  042524  046111  042105
1686  010062  052040  020117  047111
1687  010070  042524  051122  050125
1688  010076  000124
1689  010103  051050  024503  020040  DH21    ,ASCIZ  "(PC)    (PS)    (SP)    TEST#   (LKS)   "
1690  010106  020040  050050  024523
```

```
1691  010114  022040  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)    TEST0   (LKS)   "
1711  010266  020040  024040  051520
1712  010274  020051  020040  024040
1713  010302  052123  020051  020040
1714  010310  052040  051505  021524
1715  010316  022040  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  044507  042526  020123
1723  010362  047125  050535  040525
1724  010370  022114  020043  043117
1725  010376  050040  046125  042523
1726  010404  022123  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)    TEST0   1ST     2ND"<15><12>"
1733  010452  022040  024040  051520
1734  010460  020051  020040  024040
1735  010466  052123  020051  020040
1736  010474  052040  051505  021524
1737  010502  022040  020440  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  022040  020040  020040
```

```
1745  018562  042520  044522  042117
1746  018570  020040  042520  044522
1747  018576  042117     300
1748          018602                  .EVEN
1749  018602  001116  001172  001170  DT4:   SERRPC,$REG7,$REG6,$REG5,$REG1,$REG2
1750  018610  001166  001156  001160
1751  018616  000000                  0
1752  018620  045514  020123  042522  EM5:   .ASCIZ  "LKS REGISTER RESPONDS TO ANOTHER ADDRESS"
1753  018626  044527  052123  051105
1754  018634  051040  051505  047520
1755  018642  042116  020123  047524
1756  018650  040440  047516  044124
1757  018656  051105  040440  042104
1758  018664  042522  051523     000
1759  018671     050  041520  020051  DH5:   .ASCIZ  "(PC)    (PS)    (SP)    TEST#   ADDRESS"
1760  018676  020040  024040  051520
1761  018704  020051  020040  024040
1762  018712  050123  020051  020040
1763  018720  052040  051505  021524
1764  018726  020040  040440  042104
1765  018734  042522  051523     000
1766          018742                  .EVEN
1767  018742  001116  001172  001170  DT5:   SERRPC,$REG7,$REG6,$REG5,$GOADR
1768  018750  001166  001120
1769  018754  000000                  0
1770  018756  020101  047516  051440  EM6:   .ASCIZ  "A NO SACK TIMEOUT HAS OCCURED"
1771  018764  041501  020113  044524
1772  018772  042515  052517  020124
1773  019000  040516  020123  041517
1774  019006  052503  042522  000104
1775  019014  050050  024503  020040  DH6:   .ASCIZ  "(PC)    (PS)    (SP)    TEST#   (LKS)   "
1776  019022  020040  050050  024523
1777  019030  020040  020040  051450
1778  019036  024520  020040  020040
1779  019044  042524  052123  020043
1780  019052  020040  046050  051513
1781  019060  020051  020040     000
1782          019066                  .EVEN
1783  019066  001116  001172  001170  DT6:   SERRPC,$REG7,$REG6,$REG5,LKS
1784  019074  001166  177546
1785  019100  000000                  0
1786  019102  051127  047117  020107  EM7:   .ASCIZ  "WRONG CONDITION CODES WERE PUT ONTO STACK BY INTERRUPT"
1787  019110  047523  042116  052111
1788  019116  047511  020116  047523
1789  019124  042504  020123  042527
1790  019132  042522  050040  052125
1791  019140  047440  052116  020117
1792  019146  052123  041501  020113
1793  019154  054502  044440  052116
1794  019162  051105  052522  052120
1795  019170     000
1796  019171     050  041520  020051  DH7:   .ASCIZ  "(PC)    (PS)    (SP)    TEST#   CC WAS  CC S/B"
1797  019176  020040  024040  051520
1798  019204  020051  020040  024040
```

```
1799  011212  025123  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:   $ERRPC,$REG7,$REG6,$REG5,BUF1,$GOODAT
1806  011256  001166  000774  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  020110  047111
1814  011332  042524  051122  050125
1815  011340  000124
1816  011342  052050  024503  020040  DH10:  .ASCIZ  "(PC)    (PS)    (SP)    TEST#  @(SP)WAS @(SP)S/B  "
1817  011352  020040  050050  024523
1818  011356  020040  020040  051440
1819  011364  024520  020040  020040
1820  011372  042524  052123  020043
1821  011400  040040  051450  024520
1822  011406  040527  020123  024100
1823  011414  052123  051451  041057
1824  011422  020040     000
1825          011426                  .EVEN
1826  011426  001116  001172  001170  DT10:  $ERRPC,$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  020105  045514  020123
1833  011474  042522  044507  052123
1834  011502  051105  020054  047101
1835  011510  020104  051124  050101
1836  011516  042520  000104
1837  011522  052050  024503  020040  DH11:  .ASCIZ  "(PC)    (PS)    (SP)    TEST#"
1838  011530  020040  050050  024523
1839  011536  020040  020040  051450
1840  011544  024520  020040  020040
1841  011552  042524  052123  020043
1842                                  .EVEN
1843  011560  001116  001172  001170  DT11:  $ERRPC,$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  217412  000000                          ,WORD 0
1854          000001                  ,END
```

```
A0004   001762            537#    540
A0005   002044            555#    558
A0006   002150            579#    582
A0007   002300            607#    610
A0012   002430            639#    642
A0011   002532            667#    670
A0012   002666            702#    705
A0013   002752            729#    732
A0014   003272            802     808#
A0015   003404            823     829#
A0016   003516            841     850#
A0017   003630            862     871#
A0020   003742            883     892#
A0021   004054            904     913#
A0023   004146            935#    938
A0024   004264            959#    960
A0025   004370            980#    983
A0026   004532           1011#   1012
A0027   004642           1034#   1035
A0030   004776           1052    1060#
A0031   005060           1078#   1081
A0032   005206           1112#   1115
A0033   005342           1143#   1146
BIT0  = 000001            293#
BIT10 = 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#
BIT11 = 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  = 002776            312#   1120*   1127    1151*   1805
BUF2  = 002774            311#   1121*   1152*   1158    1826
DBR05   002062            556     561#
DBR06   002166            544     585#
```

```
000[illegible]  002316              638     0138
000[illegible]  002446              648     0458
00011   002552              668     0678
00012   002704              703     0708
00013   002770              738     0758
00014   003004              839     0118
00015   003416              839     0328
00016   003530              851     0938
00017   003642              872     0748
000[illegible]  003754              893     0958
00021   004066              914     0168
00023   004164              936     0418
00025   004476              961     0868
00030   005012             1302#    1303
00031   005076             1279     1304#
00032   005224             1113     1118#
00033   005362             1144     1149#
C0007   002326              616#     618
C0010   002456              648#     658
C0011   002602              679#     682
C0013   003004              739#     742
C0025   004426              976      992#
C0031   005126             1393#    1393
C0032   005266             1122     1127#
C0033   005422             1153     1158#   1163
DH1     007747              413     1071#
DH10    011342              448     1016#
DH11    011522              453     1037#
DH2     012102              418     1089#
DH3     012261              423     1718#
DH4     012445              428     1732#
DH5     013071              433     1759#
DH6     011014              439     1779#
DH7     011171              443     1796#
DISPLA# 177570             2259#    1281#   1538#
DT1     010038              414     1081#
DT10    011426              449     1026#
DT11    011560              454     1043#
DT2     012152              419     1097#
DT3     012332              424     1718#
DT4     010072              429     1749#
DT5     010742              434     1767#
DT6     011066              439     1783#
DT7     011250              444     1805#
DBRE7   002340              631      621#
DBR1[illegible]  002472              633     653#
DBR11   002616              667     685#
DBR13   003022              749     749#
DBR31   005144             1391     1396#
EMTVEC# 000030             3338#     472#    473#
EM1     007066              412     1002#
EM10    011266              447     1038#
EM11    011444              452     1029#
EM2     010046              417     1084#
```

EM3	012166	422	1700#			
EM4	012346	427	1721#			
EM5	012526	432	1752#			
EM6	012706	437	1778#			
EM7	011102	442	1786#			
EOPMLT	000256	329#				
ERR5T	001740	1299#	1214#	1215#	1348#	
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	003002	810#				
E0015	003414	831#				
E0016	003526	852#				
E0017	003640	873#				
E0020	003792	894#				
E0021	004064	915#				
E0023	004160	939#				
E0024	004304	955	964#			
E0025	004402	984#				
E0026	004552	1306	1010#			
E0027	004674	1340#				
E0030	005024	1101	1306#			
E0031	005072	1302#	1087			
E0032	005220	1116#				
E0033	005354	1147#				
E10026	002210	592#				
E10027	002334	619#				
E10010	002464	651#				
E10012	002720	713#				
E10013	003016	743#				
E10023	004202	945#				
E10025	004422	993#				
E10031	005142	1094#				
E10033	005432	1161#				
E1005	002102	565#				
E1011	002612	676	683#			
E20027	002346	623#				
E20010	002476	655#				
E20021	002624	689#				
E20013	003102	763#				
E20025	004436	994#				
E20031	005152	1103#				
E30013	003132	773#				
E40013	003174	786#				
F0013	003026	747#	752			

```
CNS    = 000000 U          323    1585    1586    1587    1588    1589
GOO13   003042             748    7918
HOO13   003066             759#    762
IOTVEC= 000020             381#    470#    471#
IOOE1   001600             494#
IOO14   003264             837#
IOO15   003376             828#
IOO16   003510             849#
IOO17   003622             873#
IOO2C   003734             891#
IOO21   004046             912#
IOO23   004164             942#
IOO26   004548            1013#
JOO13   003104             760    7695
KSTART  001400             324    4005
KOO13   003122             769#    772
LKS    = 177546            309#    494#    588    922    535#    537    553#    559    562#    563    577#    579    586#
                           587#    588    685#    687    614#    621    635#    639    646#    653    665#    667    677#
                           679    700#    702    789#    711    726#    729    737#    739    746#    747    790#    756#
                           759    767#    769    777#    778    781#    886#    888    811#    827#    829    832#    848#
                           852    853#    869#    871    874#    898#    892    895#    911#    913    916#    933#    935
                           942#    943    956#    958#    962    977#    983    988#    992   1008#   1019#   1014   1030#
                          1036#   1037#   1038   1049#   1056#   1058#   1064#   1076#   1078   1088#   1098   1110#   1112
                          1123#   1141#   1143   1154#   1207#   1681   1697   1718   1783
LOO13   003136              778    7798
MOO13   003158              778#    783
NOP    = 000240             314#
NOO13   003164              779    782#
PC     =%000007             237#    335#    462#   1185#   1188#   1197#   1287#   1217#   1319#   1326#   1331#   1589#   1548#
PIRQ   = 177772             223#
PIRQVE= 000240               387#
POWFES  007452              1625   16338
POWFOP  007664              1634   16418
POWFOS  007662              1633   16408
PS     = 177776             228#    221    461    552#    576#    684#    636#    675#    724#    884#    825#    846#    867#
                           888#    989#    957#    978#    987#   1085#   1231#   1854#   1961#   1886#   1124#   1195#
PSW    = 177776            2218
PWRVEC= 000024              342#    476#    477#   1595#   1596#   1688#   1623#   1624#
RESVEC= 000010              297#
RO     =%000000             228#    536#    539#    615#    616#    647#    649#    722#    751#    753#    782#   1179#   1197
                          1311   1312#   1313   1316#   1341   1391#   1395   1371   1372   1385#   1481   1482#   1483#
                          1498#   1491#   1492#   1493#   1494#   1495   1587   1585#   1587#   1511   1513   1535   1569
                          1578#   1571   1572#   1573#   1574#   1577   1622#
RO0P1   001574              492    4938
RO0P2   001634              535    5868
RO0P3   001718              528    5218
RO0P4   001746              533    5348
RO0P5   002036              551    5538
RO0P6   002142              575    5778
RO0P7   002260              603    6038
RO0P8   002418              632    6398
RO011   002524              664    6658
RO012   002660              699    7888
RO013   002724             7228    776
```

```
R0014   000254          880     0890
R0015   000366          823     0260
R0016   000502          844     0470
R2017   000612          865     0680
R0020   000724          886     0890
R0021   001036          907     0910
R0023   001140          932     0930
R0024   001250          954     0970
R0025   001394          974     0770
R0026   001514         1004    10080
R0027   001622         1027    10300
R0030   001734         1051    10520
R0031   002052         1075    10700
R0032   002200         1109    11100
R0033   002334         1140    11410
R1      =%000001          229#     723#     749#      784      786     1342     1395#     1356     1368     1384#     1398     1021#
R1013   002736            725#
R2      =%000002          230#     794#     788#      784      787     1343     1394#     1358#     1361     1368#     1369#     1378     1379#
                         1383#    1399     1620#
R2013   002770            736#
R3      =%000003          231#     554#     557#      578#     581#     686#     689#     638#     641#     666#     669#     678#     681#
                          781#     784#     728#      731#     738#     741#     798#     761#     768#     771#     934#     937#     979#
                          982#    1077#    1088#     1189#    1092#    1111#    1114#    1142#    1145#     1344    1352#    1353#    1367#
                         1378#    1379#    1388#     1382#     1427    1436#    1442#    1443#    1446#    1451#    1452#     1453    1462#
                         1688     1619#
R3013   003052            755#
R4      =%000004          232#     1428     1438#     1431#     1432#     1433     1434#     1448     1458#     1458#     1461#     1081     1618#
R4013   003104            766#
R5      =%000005          233#     1345     1347#     1349#     1356#     1368#     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#      888#      889#
                          913#     1889#    1832#     1855#     1885#     1119#     1197#     1192#     1286#     1218#     1211#     1214     1216#
                         1243#     1248     1248      1249      1277      1278      1282#     1311#     1312      1313#     1315      1316     1317#
                         1322      1322#    1324#     1338      1341#     1342#     1343#     1344#     1345#     1346#     1347      1350#    1363
                         1365#     1367     1377      1379      1381      1382      1383      1384      1385      1387#     1388#     1419#    1428
                         1421      1422#    1427#     1428#     1429#     1435      1468      1461      1462      1463#     1464#     1481#    1486#
                         1537      1511#    1538      1532      1543      1555#     1558#     1569#     1577#     1597#     1598#     1599#    1688#
                         1681#     1682#    1683#     1684#     1685#     1686#     1687      1613#     1617      1618      1619      1628     1621
                         1622
STACK = 001103           217#
START   001416           464#     1224
STKLMT= 177774           222#
STMLS   007522           463      1648#
SWR   = 177570           224#      225     1238#     1252      1254      1268      1267      1539      1546      1558      1553
SWB   = 000001           265#
SWBE  = 000001           255#      265
SWB1  = 000002           254#      264
SWB2  = 000004           253#      263
SWB3  = 000010           252#      262
SWB4  = 000020           251#      261
SWB5  = 000040           250#      260
SWB6  = 000100           249#      259
```

```
SW07  = 000200          248#    256
SW08  = 000400          247#    257    1232    1252
SW09  = 001000          246#    256     389     479    1231    1257    1253    1256    1258    1260    1270    1281    1284
                       1526    1553    1568
SW1   = 000002          264#
SW10  = 002000          245#    393    1525    1539    1567
SW11  = 004000          244#    388     479    1182    1235    1253    1256    1257    1258    1265    1266    1267    1301
                       1284
SW12  = 010000          243#    488    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= 003014          299#
TKVEC = 000060          335#
TPVEC = 000064          336#
TRAPVE= 000034          384#    474#    475#
TRAP2   005572          523     517     531     948     972     997     637     662     697     929     951     971    1001
                       1324    1347    1373    1137    1138    1286#
TRPMES  007575         1238    1289    1649#
TRPM2S  007636         1218    1655#
TRTVEC= 000014          299#
TYPDS = 124412         1193    1589#
TYPE  = 124402         1191    1194    1213    1386    1454    1487    1497    1499    1582    1584    1588    1915    1941
                       1549    1585#   1625
TYPOC = 124402         1212    1488    1512    1586#
TYPOL = 124406         1588#
TYPOS = 124404         1587#
T0001   001552          482     489#   1281    1630
T0002   001610          495     502#
T0003   001656          529     510#
T0004   001722          523     538#
T0005   001776          538     547#
T0006   002102          560     564    5718
T0007   002212          584     589    5968
T0010   002358          612     628     622    6298
T0011   002500          644     652     654    6618    684
T0012   002626          672     688    6958
T0013   002722          707     712    7218
T0014   003218          734     744     764     774     785    7988
T0015   003322          812    8198
T0016   003434          833    8488
T0017   003546          854    8618
T0020   003662          875    8828
T0021   003772          896    9038
T0022   004104         9218
T0023   004106          917    9288
```

```
T0024   004284          947     944     953#
T0025   004306          963     978#
T0026   004442          985     991     993     1383#
T0027   004554          1015    1023#
T0030   004676          1029    1039    1046#
T0031   005026          1065    1272#
T0032   005154          1083    1095    1099    1186#   1131
T0033   005316          1117    1128    1137#
T0034   005442          1148    1159    1166#
WORD    001232          394#    1033*   1034*   1057*   1062*   1001F   1001
SBDADR  001122          358#
SBDDAT  001126          367#
SBELL   001228          398#    1541    1568
SCNTAG  001108          346#    465     472     478     479
SCN1  * 002010          372#    373#    374#    375#    376#    377#    378#    379#    388#
SCN2  * 002020          372#    373#    374#    375#    376#    377#    378#    379#    38F#
SCN3  * 002018          375#    372
SCN4  * 002018          388#    381#    382#    383#    384#    385#    386#    387#    388#
SCRLF   001225          392#    1213    1488    1499    1504    1508    1549    1568
SDBLK   006454          1352    1386    1394#
SDDAGN  005546          1187    1196    1201#
SDTBL   006444          1355    1398#
SENDAD  005536          342     1197#
SENDCT  005506          478     1189#
SENDMG  005552          1191    1202#
SENULL  005567          1194    1209#
SEOP    005452          1179#
SEOPCT  005522          478*    1186#   1195
SERFLG  001103          349#    1234    1256    1258    1264*   1285    1536*   1568
SERMAX  001115          395#    479*    1258    1288*   1285
SERROR  007246          472     1529#
SERRPC  001116          396#    1486    1543*   1544*   1545    1561*   1601    1697    1718    1749    1787    1783    1889
                        1826    1843
SERRTB  001232          411#    1494
SERRTY  006712          1479#   1548
SERTTL  001112          393#    1542*   1563
SESCAP  001216          389#    1279*   1556    1558    1562*
SFILLC  001150          388#    1328    1332
SFILLS  001147          387#    1332
SGDADR  001120          357#    881*    887*    813*    824*    828*    834*    845*    849*    855*    866*    878*    876*
                        887*    891*    897*    988*    912*    918*    1787
SGDDAT  001124          359#    587*    519*    934*    552*    574*    599*    696*    799*    822*    843*    864*    885*
                        986*    931*    943     953*    962     973*    1883*   1814    1826*   1838    1858*   1129*   1168*
                        1681    1885    1826
SGET42  005538          1195#
SHD   * 002003          233
SICNT   001104          398#    1271*   1272    1274*   1284
SILLIP  007434          1595    1627#
SITEMB  001114          394#    1483    1545*   1563
SLF     001226          393#    1508
SLPADR  001106          391#    488*    492*    985*    52P*    933*    551*    575*    683*    632*    684*    699*    725*
                        736*    759*    766*    776*    88P*    923*    844*    865*    886*    987*    932*    954*    974*
                        1834*   1827*   1351*   1875*   1189*   114P*   1202*   1277*   1282    1284
SLPERR  001110          392#    1262    1278*   1284    1555
```

F4

```
$MXCNT  000133          1279    1284#
$NULL   001146           366#   1322    1332
$OCNT   000706          1426*   1455*   1468#
$OMODE  000710          1421*   1429*   1438    1433*   1444*   1478#
$OVER   000114          1239    1255    1263    1273    1281#
$PASS   001100           347#   1183*   1184*   1192    1202    1269    1285
$POWER  007442          1625    1638#
$PWRDN  007272           476    1595#   1623
$PWRUP  007363          1608    1613#
$QUES   001224           391#   1568
$RDCWR= ****** U        1598
$RDDEC= ****** U        1599
$RDLIN= ****** U        1590
$RDOCT= ****** U        1598
$REGAD  001152           370#
$REG0   001154           372#   1539*
$REG1   001156           373#    786*   1749
$REG2   001160           374#    787*   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           470    1236#
$SETUP= 002037           194#    478     472     474     476     478     479     489    1181
$STUP = 177777           193#
$SVLAD  006266          1247    1276#
$SWR  = 167400           165#    288     285     286     287     288     289     210     211     388     389     398     479
                         483    1176    1182    1195    1202    1229    123?    1231    1232    1238    125?    1252    1253
                        1256    1257    1258    1265    1266    1267    1270    1281    1284    1523    1524    1529    1526
                        1539    1546    1553    1553    156?
$TIMES  001214           388#   1182*   1265*   1272    1279*   1284
$TKB    001140           363#
$TKS    001136           362#    467
$TMP    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#    288
$TPB    001144           365#   1338*   1332
$TPFLG  001151           369#   1387    1332
$TPS    001142           364#   1328    1332
$TRAP   002242           474    1569#
$TRP  = 000012          1576#   1586#   1587#   1585#   1589#   159?#
$TRPAD  002262          1573    1584#
$TST'M  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    1585
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     1282    1286
                       1284    1285    1332    1394#   1518#   1567    1617    1628    1659#   1690#   1748#   1766#   1782#
                       1825#   1847#
```

```
ADITAG    525#   394
COMMEN    388#
ENDCOM    388#
ERROR     218#   496   518   524   541   559   565   583   597   611   619   623   643   691   698
           671   683   689   706   713   735   743   763   773   788   818   831   852   873   894
           915   939   945   964   984   998   994  1016  1047  1066  1082  1094  1108  1116  1138
          1147  1161
ESCAPE    388#
MULT      388#
NEWTST    388#
POP       388#  1381  1617
PUSH      388#  1340  1597  1603
SAVE     1519#  1530
SCOPE     219#   489   502   516   530   547   571   596   629   661   695   721   790   819   848
           861   882   903   921   928   950   978  1008  1023  1046  1072  1108  1137  1166  1189
SETRAP    198#   503   517   531   548   572   597   638   662   697   729   751   771  1001  1024
          1047  1073  1107  1138
SETTRA   1576#  1586  1587  1588  1589
SETUP     388#   464
SKIP      388#
SLASH     388#
SPACE     388#
STARS     189#   388#   334   345  1178  1225  1285  1332  1395  1471  1519  1568  1598
TRMTRP   1576#
TYPBIN    388#
TYPDEC    388#  1192
TYPNUM    388#
TYPOCS    388#
TYPOCT    388#  1486  1510
TYPTXT    388#
WAITLK    198#   554   578   686   638   666   678   781   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    388#
$SNELT    388#
$SSET    1576#  1586  1587  1588  1589
$SSKIP    388#
.EQUAT    188#   213
.HEADE    188#   192
.SETUP    188#   198
.SWRHI    188#   208
.SWRLO    188#  2128
.SCATC    188#   313
.SCMTA    189#   334
.SEOP     188#  1172
.SERRO    189#  1519
.SERRT    189#  1471
.SPOWE    189#  1598
.SSCOP    188#  1225
.STRAP    189#  1562
.STYPD    189#  1332
.STYPE    189#  1285
.STYPO    189#  1395
```

```
ADD     1317  1368  1422  1432  1494
ASL     1491  1492  1493
ASLB    1365
BCC     1366
BEQ      482   589   589   785   889   838   851   872   893   914   944   963   993  1019  1039
        1128  1159  1196  1253  1255  1257  1261  1278  1449  1496  1581  1914  1537  1548  1554
        1557
BGE     1273
BGT     1187  1374  1456
BHI     1259
BIC     1184  1446
BIS     1368  1369  1451  1452
BISB    1483
BIT      588   563   588   888   829   858   871   892   913  1252  1268  1287  1539  1548  1553
BLT     1325  1357  1373  1457
BMI      523   538   556   588   688   622   648   654   669   688   783   712   738   748   768
         778   936   981  1879  1891  1113  1144  1239  1304
BNE      468   548   558   564   582   618   618   642   658   678   682   785   732   742   752
         762   772   783   938   968   983  1812  1835  1883  1881  1893  1119  1148  1268  1314
        1321  1362  1447  1484  1586  1547  1616
BPL      748   779  1338  1329  1348  1378  1445  1551
BR       495   562   584   612   628   644   652   672   684   688   787   734   744   764   774
         948   985   991  1885  1883  1895  1899  1117  1131  1148  1241  1247  1258  1283  1288
        1318  1327  1359  1376  1423  1438  1459  1489  1516  1618  1628
CLR      466   494   535   536   553   554   577   578   587   685   686   615   635   638   647
         665   666   678   788   731   789   722   723   726   728   737   738   746   758   753
         754   756   758   767   768   777   781   799   886   811   813   822   827   872   834
         843   848   853   855   864   869   874   876   885   898   895   897   986   911   916
         918   933   934   956   977   979   987  1888  1828  1838  1833  1849  1853  1854  1856
        1861  1864  1876  1877  1889  1113  1111  1128  1121  1141  1142  1151  1152  1167  1181
        1182  1286  1215  1216  1265  1279  1351  1354  1436  1482  1533  1614
CLRB    1257  1264  1382
CMP      467   784   943   962   992  1714  1838  1127  1198  1248  1272  1372
CMPB    1254  1258  1323
DEC     1185  1498
DECB    1324  1444  1455
EMT      218
HALT     328   329  1389  1552  1639  1627
INC      539   557   581   689   616   641   648   669   681   784   731   741   749   751   761
         771   788   782   937   959   982  1811  1834  1887  1892  1114  1145  1183  1271  1358
        1458  1458  1542  1615
INCB    1862  1276  1536
IOT      219
JMP      324  1281  1224
JSR      462  1197  1287  1217  1319  1328  1348
MOV      468   461   465   469   478   471   472   473   474   475   476   477   478   488   498
         491   492   493   583   584   585   587   517   518   519   528   531   532   533   534
         548   549   558   551   552   562   572   573   574   579   576   586   597   598   599
         628   681   602   683   684   614   638   631   632   633   634   636   637   648   662
         663   664   674   675   676   677   696   697   698   699   724   725   736   755   766
         776   786   787   888   881   882   883   884   885   887   812   828   821   823   824
         825   826   828   833   841   842   844   845   846   847   849   854   862   863   865
         866   867   868   878   875   883   884   886   887   888   889   891   896   984   985
         927   928   989   918   912   917   929   938   931   932   942   951   952   953   954
```

```
             955   957   958   971   972   973   974   975   976   978   988  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  1139  1139  1148  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
            1379  1381  1382  1383  1384  1385  1387  1388  1419  1427  1428  1429  1435  1442  1468
            1461  1462  1463  1464  1481  1486  1495  1506  1505  1507  1511  1530  1532  1535  1538
            1543  1555  1558  1569  1578  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  1287  1313  1338  1358  1353  1367  1378  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   718   961  1013  1108  1223
ROL         1238  1437  1439  1440  1441  1443
RTI         1283  1318  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   687   621   639   653   667   679   782   711   729   739   747
            759   769   778   935   983  1078  1095  1112  1143  1256  1307  1328  1363  1377
WAIT        989  1059  1126  1197
.ABS        185
.ASCII      391   392
.ASCIZ      398   393  1232  1517  1638  1635  1642  1649  1655  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  1285  1466  1467  1468  1469
.ENABL      185   187
.END        1854
.ENDC       198   195   208   210   211   212   218   294   309   325   335   345   370   380   388
            389   392   391   395   396   469   478   472   474   476   478   479   488  1171  1174
           1175  1176  1178  1181  1187  1190  1191  1195  1282  1285  1286  1220  1235  1230  1240
           1251  1254  1256  1258  1268  1267  1271  1276  1281  1284  1285  1286  1333  1396  1472
           1498  1519  1528  1528  1536  1543  1549  1598  1562  1561  1578  1573  1589  1586  1587
           1588  1589  1594  1591  1627  1617  1626  1633
.EQUIV      218   219   221   236   237   256   257   258   259   260   261   262   263   264   265
            284   285   286   287   288   289   290   291   292   293
.EVEN      1518  1632  1659  1688  1696  1717  1748  1766  1782  1884  1825  1842
.IF         198   191   208   209   213   211   212   216   266   294   328   334   344   378   388
            388   389   390   394   395   465   469   478   472   474   476   478   479   488  1178
           1174  1175  1176  1177  1178  1180  1186  1189  1191  1195  1281  1282  1225  1234  1237
           1238  1250  1252  1253  1256  1257  1258  1267  1269  1278  1284  1285  1332  1395  1471
           1489  1505  1519  1527  1533  1539  1546  1548  1549  1558  1553  1568  1569  1573  1576
           1586  1587  1588  1589  1598  1683  1617  1625  1626  1638
.IFF         228   218   211   212   216   335   345   378   396   478  1171  1177  1181  1187  1198
           1222  1226  1251  1254  1258  1284  1286  1333  1396  1472  1498  1519  1528  1527  1539
           1568  1561  1578  1591  1626
```

DDKWAA,P11 CROSS REFERENCE TABLE

ERRORS DETECTED: 8

*DDKWAA,DDKWAA/SOL/CRF=DDKWAA,P11
RUN-TIME: 29 16 3 SECONDS
CORE USED: 16K