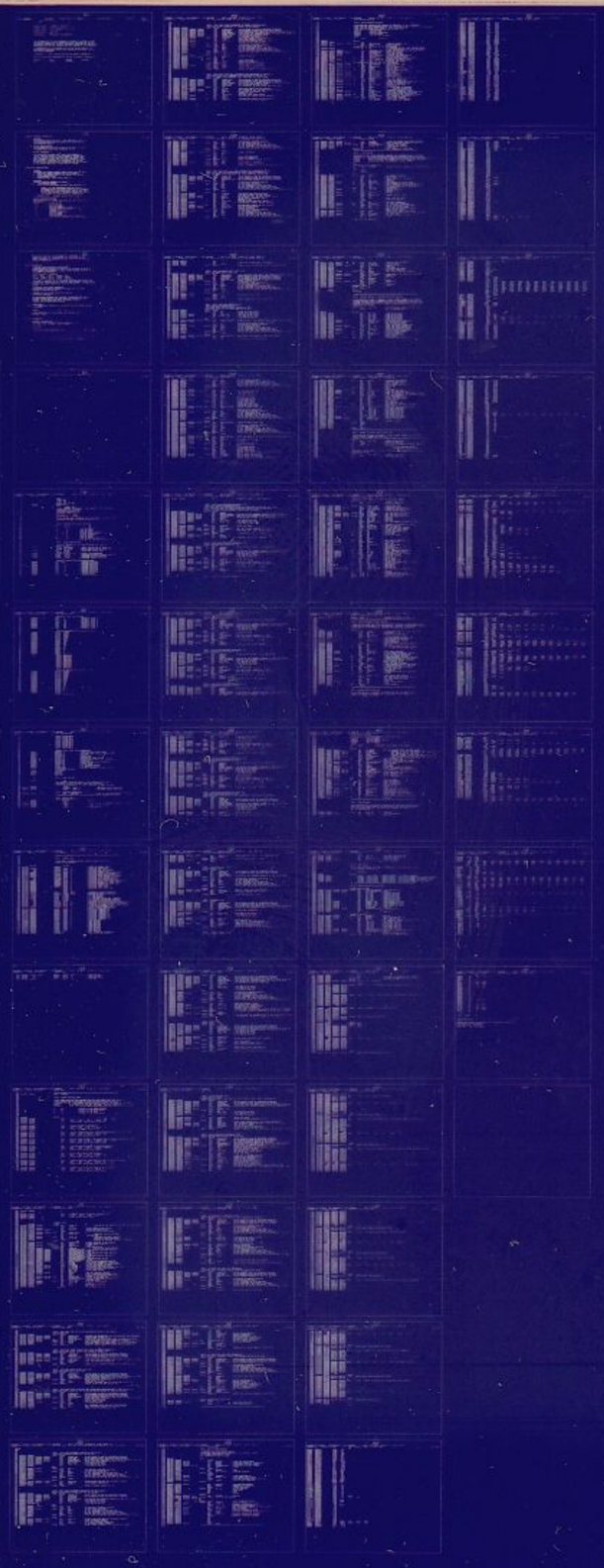
KW11-L

LINE FREQUENCY CLOCK TEST
MD-11-DZKWA-F

EP-DZKWA-F-DL

FICHE 1 OF 1

MAR 1978
digital
MADE IN USA



EOF1DZK[illegible]Q411 00010000 780223 IDENTIFICATI[illegible]P10 411 21HDR1DZKWAFSEQ 00010000 780223 SEQ 0001

PRODUCT CODE: MAINDEC-11-DZKWA-F-D

PRODUCT NAME: LINE FREQUENCY CLOCK TEST

PRODUCT DATE: FEB 1978

MAINTAINER: DIAGNOSTIC ENGINEERING

1.0 GENERAL PROGRAM INFORMATION

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

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

1.3 DEVICE ADDRESSES

THE DIAGNOSTIC ASSUMES THE LINE CLOCK ADDRESS IS 177546. VARIOUS TESTS IN THE DIAGNOSTIC ENSURE THAT THE LINE CLOCK IS NOT AFFECTED WHEN THE PAPER TAPE PUNCH (ADDRESS 177556) AND TELETYPE (ADDRESS 177566) ARE ACCESSED. THEREFORE, AN OCCASIONAL "@" MAY BE OUTPUT AT THE TELETYPE AND A CHARACTER MAY BE PUNCHED AT THE PAPER TAPE PUNCH. THESE ARE NOT ERRORS.

2.0 OPPERATING INSTRUCTIONS

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

2.2 STARTING
PROGRAM STARTING ADDRESS IS 000200
A. LOAD ADDRESS 000200
B. SELECT SWITCH REGISTER OPTIONS (SEE SECTION 2.3)

NOTE: IF THE PROCESSOR DOES NOT HAVE A HARDWARE SWITCH REGISTER, LOCATION # 176 SHOULD BE LOADED WITH THE APPROPRIATE SWITCH SETTINGS AFTER THE DIAGNOSTIC HAS BEEN LOADED INTO MEMORY. THE PROGRAM IS THEN STARTED AT ADDRESS 200.

C. DEPRESS START (PROGRAM SHOULD START RUNNING)

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

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

3.0 ERROR INFORMATION

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

3.2 UNEXPECTED TRAP ERROR REPORTING

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

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

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

3.3 POWER FAIL

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

```
POWER
```

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

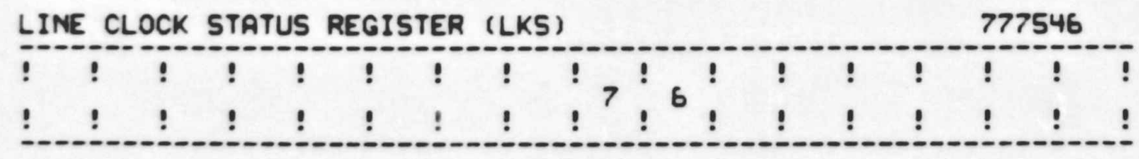
5.0 DEVICE INFORMATION

5.1 GENERAL INFORMATION

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

5.2 REGISTERS

LINE CLOCK STATUS REGISTER (LKS) 777546



BIT6 IF SET MONITOR=1 CAUSES AN INTERRUPT

BIT7 MONITOR BIT. SET BY CLOCK, CLEARED BY USER

7.0 LISTING

```
                        .ABS
                        .ENABL AMA
                        .LIST   ME
                        .NLIST  MC, MD, CND
167400                  $SWR=167400
000000                  $TN=0
                                .ENABL  ABS
                        .TITLE  MAINDEC-11-DZKWA-F  LINE FREQUENCY CLOCK PROGRAM
                        ;*COPYRIGHT (C) 1970,1977
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY J. COMEAU
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-B1),AUG 29,1975.
                        ;*

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

                        .SBTTL  BASIC DEFINITIONS

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

                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
000001                  R1=     %1              ;;GENERAL REGISTER
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
000006                  SP=     %6              ;;STACK POINTER
000007                  PC=     %7              ;;PROGRAM COUNTER
```

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

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

                                ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                  BIT15=  100000
040000                  BIT14=  40000
020000                  BIT13=  20000
010000                  BIT12=  10000
004000                  BIT11=  4000
002000                  BIT10=  2000
001000                  BIT09=  1000
000400                  BIT08=  400
000200                  BIT07=  200
000100                  BIT06=  100
000040                  BIT05=  40
000020                  BIT04=  20
000010                  BIT03=  10
000004                  BIT02=  4
000002                  BIT01=  2
000001                  BIT00=  1
                        .EQUIV  BIT09,BIT9
```

```
                                .EQUIV  BIT08,BIT8
                                .EQUIV  BIT07,BIT7
                                .EQUIV  BIT06,BIT6
                                .EQUIV  BIT05,BIT5
                                .EQUIV  BIT04,BIT4
                                .EQUIV  BIT03,BIT3
                                .EQUIV  BIT02,BIT2
                                .EQUIV  BIT01,BIT1
                                .EQUIV  BIT00,BIT0

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

                                .SBTTL  TRAP CATCHER

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

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

                                .SBTTL  ACT11 HOOKS
                                ;HOOKS REQUIRED BY ACT11
000204                                  $SVPC=.                 ;SAVE PC
000046                                  .=46
000046  006216                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
000052                                  .=52
000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
000204                                  .=$SVPC                 ;; RESTORE PC
```

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

                                .SBTTL  COMMON TAGS

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

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

```
   220  001224  177607  000377          $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
   221  001230     077                  $QUES:  .ASCII  /?/             ;;QUESTION MARK
   222  001231     015                  $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
   223  001232  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
```

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

                                .SBTTL  ERROR POINTER TABLE

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

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

001236                          $ERRTB:
001236  010454                          EM1     ;"                                      LKS     LKS     "
001240  010535                          DH1     ;"PC      PS      SP      TEST#   WAS     S/B     "
001242  010616                          DT1     ;$ERRPC,$REG7,$REG6,$REG5,LKS,$GDDAT
001244  000000                          0

001246  010634                          EM2     ;"CLOCK FAILED TO INTERRUPT"
001250  010666                          DH2     ;"PC      PS      SP      TEST#   LKS     "
001252  010740                          DT2     ;$ERRPC,$REG7,$REG6,$REG5,LKS
001254  000000                          0

001256  010754                          EM3     ;"CLOCK INTERRUPTED WHEN THE PROCESSOR PRIORITY WAS TOO HIGH"
001260  011047                          DH3     ;"PC      PS      SP      TEST#   LKS     "
001262  011120                          DT3     ;$ERRPC,$REG7,$REG6,$REG5,LKS
001264  000000                          0

001266  011134                          EM4     ;"CLOCK GIVES UNEQUAL # OF PULSES OVER TWO EQUAL PERIODS OF TIME"
001270  011233                          DH4     ;"PC      PS      SP      TEST#   1ST     2ND"
001272  011370                          DT4     ;$ERRPC,$REG7,$REG6,$REG5,$REG1,$REG0
001274  000000                          0

001276  011406                          EM5     ;"LKS REGISTER RESPONDS TO ANOTHER ADDRESS"
001300  011457                          DH5     ;"PC      PS      SP      TEST#   ADDRESS"
001302  011530                          DT5     ;$ERRPC,$REG7,$REG6,$REG5,$GDADR
001304  000000                          0

001306  011544                          EM6     ;"A NO SACK TIMEOUT HAS OCCURED"
001310  011602                          DH6     ;"PC      PS      SP      TEST#   LKS     "
001312  011654                          DT6     ;$ERRPC,$REG7,$REG6,$REG5,LKS
001314  000000                          0

001316  011670                          EM7     ;"WRONG CONDITION CODES WERE PUT ONTO STACK BY INTERRUPT"
001320  011757                          DH7     ;"PC      PS      SP      TEST#   CC      CC"
001322  012036                          DT7     ;$ERRPC,$REG7,$REG6,$REG5,BUF1,$GDDAT
001324  000000                          0

001326  012054                          EM10    ;"WRONG PC PUT ONTO THE STACK BY AN INTERRUPT"
001330  012130                          DH10    ;"PC      PS      SP      TEST#   "
001332  012214                          DT10    ;$ERRPC,$REG7,$REG6,$REG5,BUF2,$GDDAT
```

```
                                                        0

                                                        EM11    ;"TRAPPED TRYING TO ACCESS LKS REGISTER"
                                                        DH11    ;"(PC)    (PS)    (SP)    TEST#"
                                                        DT11    ;$ERRPC,$REG7,$REG6,$REG5
                                                0
                                                ;STARTUP CODE
                                                .=1400
001400  012706  001000                  KSTART: MOV     #1000,SP                ;INITIALIZE THE STACK SO WE CAN CALL THE TYPEOUT
001404  005046                                  CLR     -(SP)                   ;;
001406  013746  000034                          MOV     34,-(SP)                ;;SAVE CURRENT TRAP VECTOR
001412  012737  001422  000034                  MOV     #64$,34                 ;;SETUP NEW TRAP VECTOT
001420  104400                                  TRAP                            ;;PUSH OLD PSW AN PCON STACK
001422  016666  000002  000006          64$:    MOV     2(SP),6(SP)             ;;
001430  012716  001436                          MOV     #65$,(SP)                       ;;REPLACE OLD PC WITH NEW
001434  000002                                  RTI                             ;;RESTORE PSW
001436  012637  000034                  65$:    MOV     (SP)+,34                        ;;RESTORE OLD TRAP  VECTOR
001442  004737  006576                          JSR     PC,$TYPE                        ;PRINTOUT STARTUP MESSAGE
001446  010306                                  STMES                           ;ADDRESS OF MESSAGE "MAINDEC-11-DZKWA-F"
001450                                  START:
001450  012706  001100                          MOV     #$CMTAG,R6              ;;FIRST LOCATION TO BE CLEARED
001454  005026                                  CLR     (R6)+                   ;;CLEAR MEMORY LOCATION
001456  022706  001126                          CMP     #$BDDAT,R6              ;;DONE?
001462  001374                                  BNE     .-6                     ;;LOOP BACK IF NO
001464  012706  001000                          MOV     #1000,SP                        ;;SETUP THE STACK POINTER
001470  012737  006324  000020                  MOV     #$SCOPE,@#IOTVEC        ;;IOT VECTOR FOR SCOPE ROUTINE
001476  012737  000340  000022                  MOV     #340,@#IOTVEC+2         ;;LEVEL 7
001504  012737  007620  000030                  MOV     #$ERROR,@#EMTVEC        ;;EMT VECTOR FOR ERROR ROUTINE
001512  012737  000340  000032                  MOV     #340,@#EMTVEC+2         ;;LEVEL 7
001520  012737  010024  000034                  MOV     #$TRAP,@#TRAPVEC        ;;TRAP VECTOR FOR TRAP CALLS
001526  012737  000340  000036                  MOV     #340,@#TRAPVEC+2        ;LEVEL 7
001534  012737  010060  000024                  MOV     #$PWRON,@#PWRVEC        ;;POWER FAILURE VECTOR
001542  012737  000340  000026                  MOV     #340,@#PWRVEC+2         ;;LEVEL 7
001550  013737  006164  006156                  MOV     $ENDCT,$EOPCT           ;;SETUP END-OF-PROGRAM COUNTER
001556  005037  001220                          CLR     $TIMES                  ;;INITIALIZE NUMBER OF ITERATIONS
001562  005037  001222                          CLR     $ESCAPE                 ;;CLEAR THE ESCAPE ON ERROR ADDRESS
001566  112737  000001  001115                  MOVB    #1,$ERMAX               ;;ALLOW ONE ERROR PER TEST
001574  012737  001574  001106                  MOV     #.,$LPADR               ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
001602  012737  001602  001110                  MOV     #.,$LPERR               ;;SETUP THE ERROR LOOP ADDRESS
001610  013746  000004                          MOV     @#4,-(SP)               ;;SAVE ERROR VECTOR
001614  013746  000006                          MOV     @#6,-(SP)
001620  012737  001634  000004                  MOV     #64$,4                  ;;SET UP TIME OUT VECTOR
001626  005777  177304                          TST     @SWR                    ;;TRY TO REFERENCE HARDWARE SWR
001632  000407                                  BR      65$                     ;;BRANCH IF NO TIMEOUT TRAP OCCURS
001634  012737  000176  001136  64$:            MOV     #SWREG,SWR              ;;POINT TO SOFTWARE SWR
001642  012737  000174  001140                  MOV     #DISPREG,DISPLAY        ;;POINT TO SOFTWARE DISPLAY REG
001650  022626                                  CMP     (SP)+,(SP)+             ;;RESTORE STACK
001652  012637  000006                  65$:    MOV     (SP)+,@#6               ;;RESTORE ERROR VECTOR
001656  012637  000004                          MOV     (SP)+,@#4
001662  005737  000042                          TST     42                      ;LOADED BY A MONITOR
001666  001401                                  BEQ     TOOO1                   ;BR IF NO
001670  000005                                  RESET                           ;YES--GENERATE AN INIT
```

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


                                        .SBTTL  TEST THAT START CLEARS LINE CLOCK INTERRUPT ENABLE BIT
                                        ;TEST THAT START CLEARS LINE CLOCK INTERRUPT ENABLE BIT
001732  000004                  T0002:  SCOPE
001734  012737  006252  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
001742  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
001750  012737  001756  001106          MOV     #R0002,$LPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
001756  000005          R0002:  RESET
001760  012737  000200  001124          MOV     #200,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
001766  032737  000100  177546          BIT     #100,LKS        ;TEST THE INTERRUPT ENABLE BIT
001774  001401                          BEQ     T0003
001776  104001                  E0002:  ERROR 1                 ;ERROR, CLOCK INTERRUPT ENABLE NOT CLEARED BY INIT


                                        .SBTTL  TEST THAT START SETS CLOCK FLAG
                                        ;TEST THAT START SETS CLOCK FLAG
002000  000004                  T0003:  SCOPE
002002  012737  006252  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
002010  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
002016  012737  000200  001124          MOV     #200,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
002024  012737  002032  001106          MOV     #R0003,$LPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
002032  000005          R0003:  RESET                           ;SHOULD SET THE  CLOCK FLAG
002034  105737  177546                  TSTB    LKS             ;FIND OUT IF IT DID
002040  100401                          BMI     T0004           ;GO ON TO THE NEXT TEST IF IT SET THE CLOCK FLAG
002042  104001                  E0003:  ERROR 1                 ;ERROR, CLOCK FLAG NOT SET BY INIT


                                        .SBTTL  TEST THAT CLOCK FLAG WILL SET AFTER SUFFICIENT PERIOD OF TIME (20 MS MIN)
                                        ;TEST THAT CLOCK FLAG WILL SET AFTER SUFFICIENT PERIOD OF TIME (20 MS MIN)
002044  000004                  T0004:  SCOPE
002046  012737  006252  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
002054  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
002062  012737  002070  001106          MOV     #R0004,$LPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
002070  012737  000200  001124  R0004:  MOV     #200,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
002076  005037  177546                  CLR     LKS             ;CLEAR THE CLOCK FLAG
002102  005000                          CLR     R0              ;AND A TIMER LOCATION
002104  105737  177546          A0004:  TSTB    LKS             ;IS CLOCK FLAG SET
002110  100403                          BMI     T0005
002112  005200                          INC     R0              ;NO, INCREMENT COUNT 003 WAIT FOR SOMEMORE
002114  001373                          BNE     A0004           ;WAIT SUFFICIENT AMOUNT OF TIME FOR CLOCK
002116  104001                  E0004:  ERROR 1                 ;ERROR, CLOCK FLAG FAILED TO SET
```

```
                                        .SBTTL  TEST THAT INTERRUPT ENABLE BIT MAY BE SET
                                        ;TEST THAT INTERRUPT ENABLE BIT MAY BE SET
002120  000004                  T0005:  SCOPE
002122  012737  006252  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
002130  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
002136  012737  000100  001124          MOV     #100,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
002144  012737  002164  001106          MOV     #R0005,$LPADR   ;SETUP LOOP BACK ADDRESS IN CASE OF ERROR
002152  013746  000340                  MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
002156  012746  002164                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
002162  000002                          RTI                     ;; POP NEW PC AND PS
002164                          64$:
002164  005037  177546          R0005:  CLR     LKS
002170  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
002172  105737  177546          A0005:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
002176  100404                          BMI     B0005           ;IF SO, CONTINUE ON WITH THE TEST
002200  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
002202  001373                          BNE     A0005           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
002204  104001          E0005:  ERROR 1                         ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
002206  000410                  BR      T0006
002210                          B0005:
002210  012737  000100  177546          MOV     #100,LKS        ;CLEAR CLOCK FLAG AND SET INTERRUPT ENABLE
002216  032737  000100  177546          BIT     #100,LKS        ;IS INTERRUPT ENABLE SET?
002224  001001                          BNE     T0006
002226  104001          E1005:  ERROR 1                         ;ERROR INTERRUPT ENABLE NOT SET


                                        .SBTTL  TEST THAT INTERRUPT ENABLE BIT MAY BE CLEARED
                                        ;TEST THAT INTERRUPT ENABLE BIT MAY BE CLEARED
002230  000004                  T0006:  SCOPE
002232  012737  006252  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
002240  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
002246  012737  000000  001124          MOV     #0,$GDDAT
002254  012737  002274  001106          MOV     #R0006,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
002262  013746  000340                  MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
002266  012746  002274                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
002272  000002                          RTI                     ;; POP NEW PC AND PS
002274                          64$:
002274  005037  177546          R0006:  CLR     LKS
002300  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
002302  105737  177546          A0006:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
002306  100404                          BMI     B0006           ;IF SO, CONTINUE ON WITH THE TEST
002310  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
002312  001373                          BNE     A0006           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
002314  104001          E0006:  ERROR 1                         ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
002316  000412                  BR      T0007
002320                          B0006:
002320  012737  000100  177546          MOV     #100,LKS        ;CLEAR CLOCK FLAG AND SET INTERRUPT ENABLE
002326  005037  177546                  CLR     LKS             ;CLEAR INTERRUPT ENABLE
002332  032737  000100  177546          BIT     #100,LKS        ;TEST THE INTERRUPT ENABLE BIT
002340  001401                          BEQ     T0007           ;IS INTERRUPT ENABLE CLEARED
002342  104001          E10006: ERROR 1                         ;ERROR, ERROR INTERRUPT BIT CAN NOT BE CLEARED
```

```
                                        .SBTTL  TEST THAT CLOCK INTERRUPTS TO CORRECT VECTOR ADDRESS
                                        ;TEST THAT CLOCK INTERRUPTS TO CORRECT VECTOR ADDRESS
002344  000004                  T0007:  SCOPE
002346  012737  006252  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
002354  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
002362  012737  000300  001124          MOV     #300,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
002370  012737  002412  001106          MOV     #R0007,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
002376  012737  002474  000100          MOV     #D0007,100      ;SET UP VECTOR RETURN POINTER
002404  012737  000340  000102          MOV     #340,@#102      ;1 INTERRUPT IS ENOUGH
002412  012706  001000          R0007:  MOV     #1000,SP        ;GET STACK READY FOR INTERRUPTS
002416  005046                          CLR     -(SP)           ;DROP CPU PRIORITY LEVEL
002420  012746  002426                  MOV     #1$,-(SP)       ;SET RETURN ADDRESS ON STACK FROM 'RTI'
002424  000002                          RTI                     ;RETURN TO NEXT INSTRUCTION
002426  005037  177546          1$:     CLR     LKS
002432  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
002434  105737  177546          A0007:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
002440  100404                          BMI     B0007           ;IF SO, CONTINUE ON WITH THE TEST
002442  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
002444  001373                          BNE     A0007           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
002446  104001          E0007:  ERROR 1                         ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
002450  000415                  BR      T0010
002452                  B0007:
002452  012737  000100  177546          MOV     #100,LKS        ;ENABLE INTERRUPT
002460  005000                          CLR     R0
002462  005200          C0007:  INC     R0
002464  000240                          NOP                     ;STALL FOR TIME
002466  001375                          BNE     C0007           ;WAIT FOR INTERRUPT
002470  104002          E10007: ERROR 2                         ;ERROR, DIDNT GET INTERRUPT
002472  000404                  BR      T0010
002474  105737  177546          D0007:  TSTB    LKS             ;ENTER HERE IF INTERRUPTED
002500  100401                          BMI     T0010
002502  104001          E20007: ERROR 1                         ;ERROR, INTERRUPT NOT CAUSED BY CLOCK


                                        .SBTTL  TEST THAT CLOCK WILL INTERRUPT WITH PROCESSOR AT PRIORITY 5
                                        ;TEST THAT CLOCK WILL INTERRUPT WITH PROCESSOR AT PRIORITY 5
                                        ;NOTE: IF THERE IS NO HARDWARE 'PSW' THIS TEST WILL BE SKIPPED
002504  000004                  T0010:  SCOPE
002506  012737  006252  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
002514  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
002522  012737  002544  001106          MOV     #R0010,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
002530  012737  002654  000100          MOV     #D0010,100      ;SET UP VECTOR RETURN POINTER
002536  012737  000340  000102          MOV     #340,@#102      ;NO INTERRUPTS ALLOWED AFTER THE FIRST ONE
002544  005037  177546          R0010:  CLR     LKS
002550  013746  000004                  MOV     4,-(SP)         ;SAVE BUS TIMEOUT VECTOR CONTENTS
002554  012737  002572  000004          MOV     #1$,4           ;SET A SERVICE 'PC' IN CASE TIMEOUT OCCURS
002562  012737  000240  177776          MOV     #PRS,PS         ;DO WE HAVE A HARDWARE 'PSW'?
002570  000404                          BR      2$              ;BRANCH IF YES
002572  022626                  1$:     CMP     (SP)+,(SP)+     ;RESTORE STACK FROM TIMEOUT
002574  012637  000004                  MOV     (SP)+,4         ;RESTORE BUS TIMEOUT VECTOR CONTENTS
002600  000431                          BR      T0011
002602  012637  000004          2$:     MOV     (SP)+,4         ;RESTORE BUS TIMEOUT VECTOR CONTENTS
002606  012706  001000                  MOV     #1000,SP        ;INITIALIZE THE STACK POINTER
```

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


                                          .SBTTL  TEST THAT CLOCK WILL NOT INTERRUPT WITH PROCESSOR PRIORITY 6
                                          ;TEST THAT CLOCK WILL NOT INTERRUPT WITH PROCESSOR PRIORITY 6
                                          ;NOTE: IF THERE IS NO HARDWARE 'PSW' THIS TEST WILL BE SKIPPED
002664  000004                            T0011:    SCOPE
002666  012737  006252  000004                      MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
002674  012737  000340  000006                      MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
002702  012737  002710  001106                      MOV     #R0011,$LPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
002710  005037  177546                    R0011:    CLR     LKS
002714  005003                                      CLR     R3              ;INITIALIZE A COUNTER LOCATION
002716  105737  177546                    A0011:    TSTB    LKS             ;IS THE CLOCK FLAG SET?
002722  100404                                      BMI     B0011           ;IF SO, CONTINUE ON WITH THE TEST
002724  005203                                      INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
002726  001373                                      BNE     A0011           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
002730  104001                            E0011:    ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
002732  000443                                      BR      T0012
002734                                    B0011:
002734  012706  001000                              MOV     #1000,SP        ;INITIALIZE THE STACK POINTER
002740  013746  000004                              MOV     4,-(SP)         ;SAVE BUS TIMEOUT VECTOR CONTENTS
002744  012737  002762  000004                      MOV     #1$,4           ;SET SERVICE 'PC' IN CASE TIMEOUT OCCURS
002752  012737  000300  177776                      MOV     #PR6,PS         ;DO WE HAVE A HARDWARE 'PSW'?
002760  000404                                      BR      2$              ;BRANCH IF YES
002762  022626                            1$:       CMP     (SP)+,(SP)+     ;RESTORE STACK FROM BUS TIMEOUT
002764  012637  000004                              MOV     (SP)+,4         ;RESTORE BUS TIMEOUT VECTOR CONTENTS
002770  000424                                      BR      T0012
002772  012637  000004                    2$:       MOV     (SP)+,4         ;RESTORE BUS TIMEOUT VECTOR CONTENTS
002776  012737  003026  000100                      MOV     #E1011,100      ;SET UP VECTOR RETURN
003004  012737  000100  177546                      MOV     #100,LKS        ;ENABLE INTERRUPT
003012  005003                                      CLR     R3              ;INITIALIZE A COUNTER LOCATION
003014  105737  177546                    C0011:    TSTB    LKS             ;IS THE CLOCK FLAG SET?
003020  100404                                      BMI     D0011           ;IF SO, CONTINUE ON WITH THE TEST
003022  005203                                      INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
003024  001373                                      BNE     C0011           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
003026  104001                            E1011:    ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
003030  000715                                      BR      T0011
003032                                    D0011:
```

```
                        000240                          NOP
                        000240                          NOP
                        000401                          BR      T0012
                        104003          E20011: ERROR 3                 ;ERROR, CLOCK INTERRUPTED WITHOUT HAVING PRIORITY


                                        .SBTTL  TEST THAT RESET SETS CLOCK FLAG
                                        ;TEST THAT RESET SETS CLOCK FLAG
                        000004          T0012:  SCOPE
                        012737  000200  001124          MOV     #200,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
                        012737  006252  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
                        012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
                        012737  003074  001106          MOV     #R0012,$LPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
                        005037  177546          R0012:  CLR     LKS             ;CLEAR CLOCK FLAG
                        005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
                        105737  177546          A0012:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
                        100404                          BMI     B0012           ;IF SO, CONTINUE ON WITH THE TEST
                        005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
                        001373                          BNE     A0012           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
                        104001          E0012:  ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
                        000407                          BR      T0013
                                        B0012:
                        005037  177546                  CLR     LKS
                        000005                          RESET                   ;SHOULD SET CLOCK FLAG
                        105737  177546                  TSTB    LKS
                        100401                          BMI     T0013
                        104001          E10012: ERROR 1                 ;ERROR, RESET DIDN'T SET CLOCK FLAG


                                        .SBTTL  TEST LINE CLOCK REPEATABILITY
                                        ;TEST LINE CLOCK REPEATABILITY
                                        ;MAKE SURE THAT OVER TWO EQUAL PERIODS OF TIME
                                        ;THE CLOCK PUTS OUT THE SAME NUMBER OF PULSES
                        000004          T0013:  SCOPE
                        005000          R0013:  CLR     R0              ;CLEAR 1ST TIME COUNT
                        005001                  CLR     R1              ;CLEAR 1ST CLOCK COUNT
                        013746  000340          MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
                        012746  003156          MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
                        000002                  RTI                     ;; POP NEW PC AND PS
                                        64$:
                        012737  003156  001106  R1013:  MOV     #R1013,$LPADR   ;ERROR IN NEXT  FEW INSTRUCTIONS CAUSES A SHORT SCOPE LO
                        005037  177546                  CLR     LKS
                                                                        ;SYNC ON CLOCK FLAG A COUPLE OF TIMES
                        005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
                        105737  177546          A0013:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
                        100404                          BMI     B0013           ;IF SO, CONTINUE ON WITH THE TEST
                        005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
                        001373                          BNE     A0013           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
                        104001          E0013:  ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
                        000524                  BR      T0014
                                        B0013:
                        012737  003210  001106  R2013:  MOV     #R2013,$LPADR   ;MAKE SCOPE LOOP SHORT IN CASE OF AN ERROR
                        005037  177546                  CLR     LKS
                        005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
```

```
                                        C0013:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
                                                BMI     D0013           ;IF SO, CONTINUE ON WITH THE TEST
                                                INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
                                                BNE     C0013           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
                                        E10013: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
                                                BR      T0014
                                        D0013:
                                                CLR     LKS
                                        1$:     BIT     #200,LKS        ;IS THE CLOCK FLAG SET ?
                                                BNE     1$              ; NO, WAIT FOR IT
                                                CLR     LKS
                                        F0013:  TSTB    LKS             ;IS CLOCK FLAG SET
                                                BPL     G0013           ;NO
                                                INC     R1              ;+1 TO CLOCK COUNT
                                                CLR     LKS             ;CLEAR CLOCK IF SET
                                        G0013:  INC     R0              ;+1 TO TIME COUNT
                                                BNE     F0013           ;REPEAT UNTIL R0=0
                                                CLR     R0              ;CLEAR 2ND TIME COUNT
                                                CLR     R2              ;CLEAR 2ND CLOCK COUNT
                                        R3013:  MOV     #R3013,$LPADR   ;INSURE A SHORT SCOPE LOOP
                                                CLR     LKS
                                                                        ;SYNC ON CLOCK FLAG TWICE
                                                CLR     R3              ;INITIALIZE A COUNTER LOCATION
                                        H0013:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
                                                BMI     J0013           ;IF SO, CONTINUE ON WITH THE TEST
                                                INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
                                                BNE     H0013           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
                                        E20013: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
                                                BR      T0014
                                        J0013:
                                        R4013:  MOV     #R4013,$LPADR   ;INSURE A SHORT SCOPE LOOP
                                                CLR     LKS
                                                CLR     R3              ;INITIALIZE A COUNTER LOCATION
                                        K0013:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
                                                BMI     L0013           ;IF SO, CONTINUE ON WITH THE TEST
                                                INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
                                                BNE     K0013           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
                                        E30013: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
                                                BR      T0014
                                        L0013:
                                                MOV     #R0013,$LPADR   ;MUST LOOP BACK TO BEGINING OF TEST IF EEROR COMES NOW
                                                CLR     LKS
                                        1$:     BIT     #200,LKS        ;IS CLOCK FLAG SET ?
                                                BNE     1$              ;NO, WAIT
                                                CLR     LKS
                                        M0013:  TSTB    LKS             ;IS CLOCK FLAG SET
                                                BPL     N0013           ;NO
                                                INC     R2              ;+1 TO CLOCK COUNT
                                                CLR     LKS             ;CLEAR CLOCK IF SET
                                        N0013:  INC     R0              ;+1 TO TIME COUNT
                                                BNE     M0013           ;REPEAT UNTIL R0=0
                                                CMP     R1,R2           ;IS 1ST CLOCK COUNT EQUAL TO 2ND CLOCK COUNT?
                                                BEQ     T0014           ;YES
                                        E40013: MOV     R1,$REG1        ;GET R1 READY FOR PRINTOUT
                                                MOV     R2,$REG2        ;GET R2 READY FOR PRINTOUT
                                                ERROR 4                 ;ERROR, CLOCK FLAG OCCURRED DIFFERENT
```

```
                                        NOP                     ;NUMBER OF TIMES IN EQUAL PERIODS


                                .SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
                                ;LINE CLOCK REGISTER ADDRESSING TEST
                                ;TEST THAT THE "LKS" REGISTER CAN NOT BE ADDRESSED AS ANYTHING BUT 177546
                                ;SET A LOCATION THAT IS CLOSE(DIFFERS BY 1 BIT) TO THE LKS REGISTER
                                ;TO 100. IF THE LKS ALSO CHANGES, THEN SIGNAL AN ERROR
003460  000004                T0014:  SCOPE
003462  005037  001124                CLR     $GDDAT
003466  012737  003530  001106        MOV     #R0014,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
003474  012737  157546  001120        MOV     #157546,$GDADR  ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 13 CLEAR
003502  012737  003546  000004        MOV     #A0014,4        ;SETUP VECTOR IN CASE IT IS NONEXISTANT
003510  012737  000340  000006        MOV     #340,6
003516  013746  000340                MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
003522  012746  003530                MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
003526  000002                        RTI                     ;; POP NEW PC AND PS
003530                          64$:
003530  012706  001000        R0014:  MOV     #1000,SP        ;SETUP THE STACK
003534  005037  177546                CLR     LKS
003540  012777  000100  175352  I0014:  MOV     #100,@$GDADR    ;SET THE "CLOSE" ADDRESS TO = 100
003546  032737  000100  177546  A0014:  BIT     #100,LKS        ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
003554  001401                        BEQ     B0014
003556  104005                E0014:  ERROR 5                 ;IT AFFECTED "LKS"  --  ERROR
003560  005037  177546        B0014:  CLR     LKS
003564  012737  003576  000004        MOV     #T0015  ,4
003572  005077  175322                CLR     @$GDADR         ;CLEAR OUT THE "CLOSE" ADDRESS


                                .SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
                                ;LINE CLOCK REGISTER ADDRESSING TEST
003576  000004                T0015:  SCOPE
003600  012737  003664  000004        MOV     #A0015,4        ;SETUP VECTOR IN CASE IT IS NONEXISTANT
003606  012737  000340  000006        MOV     #340,6
003614  005037  001124                CLR     $GDDAT
003620  012737  003646  001106        MOV     #R0015,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
003626  012737  177146  001120        MOV     #177146,$GDADR  ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 8 CLEAR
003634  013746  000340                MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
003640  012746  003646                MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
003644  000002                        RTI                     ;; POP NEW PC AND PS
003646                          64$:
003646  012706  001000        R0015:  MOV     #1000,SP        ;SETUP THE STACK
003652  005037  177546                CLR     LKS
003656  012777  000100  175234  I0015:  MOV     #100,@$GDADR    ;SET THE "CLOSE" ADDRESS TO = 100
003664  032737  000100  177546  A0015:  BIT     #100,LKS        ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
003672  001401                        BEQ     B0015
003674  104005                E0015:  ERROR 5                 ;IT AFFECTED "LKS"  --  ERROR
003676  005037  177546        B0015:  CLR     LKS
003702  012737  003714  000004        MOV     #T0016  ,4
003710  005077  175204                CLR     @$GDADR         ;CLEAR OUT THE "CLOSE" ADDRESS


                                .SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
```

```
                                        ;LINE CLOCK REGISTER ADDRESSING TEST
003714  000004                  T0016:  SCOPE
003716  012737  004002  000004          MOV     #A0016,4        ;SETUP VECTOR IN CASE IT IS NONEXISTANT
003724  012737  000340  000006          MOV     #340,6
003732  005037  001124                  CLR     $GDDAT
003736  012737  003764  001106          MOV     #R0016,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
003744  012737  177446  001120          MOV     #177446,$GDADR  ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 6 CLEAR
003752  013746  000340                  MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
003756  012746  003764                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
003762  000002                          RTI                     ;; POP NEW PC AND PS
003764                          64$:
003764  012706  001000          R0016:  MOV     #1000,SP        ;SETUP THE STACK
003770  005037  177546                  CLR     LKS
003774  012777  000100  175116  I0016:  MOV     #100,@$GDADR    ;SET THE "CLOSE" ADDRESS TO = 100
004002  032737  000100  177546  A0016:  BIT     #100,LKS        ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
004010  001401                  BEQ     B0016
004012  104005                  E0016:  ERROR 5                 ;IT AFFECTED "LKS"  --  ERROR
004014  005037  177546          B0016:  CLR     LKS
004020  012737  004032  000004          MOV     #T0017  ,4
004026  005077  175066                  CLR     @$GDADR         ;CLEAR OUT THE "CLOSE" ADDRESS


                                        .SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
                                        ;LINE CLOCK REGISTER ADDRESSING TEST
004032  000004                  T0017:  SCOPE
004034  012737  004120  000004          MOV     #A0017,4        ;SETUP VECTOR IN CASE IT IS NONEXISTANT
004042  012737  000340  000006          MOV     #340,6
004050  005037  001124                  CLR     $GDDAT
004054  012737  004102  001106          MOV     #R0017,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
004062  012737  177556  001120          MOV     #177556,$GDADR  ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 3 SET
004070  013746  000340                  MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
004074  012746  004102                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
004100  000002                          RTI                     ;; POP NEW PC AND PS
004102                          64$:
004102  012706  001000          R0017:  MOV     #1000,SP        ;SETUP THE STACK
004106  005037  177546                  CLR     LKS
004112  012777  000100  175000  I0017:  MOV     #100,@$GDADR    ;SET THE "CLOSE" ADDRESS TO = 100
004120  032737  000100  177546  A0017:  BIT     #100,LKS        ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
004126  001401                  BEQ     B0017
004130  104005                  E0017:  ERROR 5                 ;IT AFFECTED "LKS"  --  ERROR
004132  005037  177546          B0017:  CLR     LKS
004136  012737  004150  000004          MOV     #T0020  ,4
004144  005077  174750                  CLR     @$GDADR         ;CLEAR OUT THE "CLOSE" ADDRESS


                                        .SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
                                        ;LINE CLOCK REGISTER ADDRESSING TEST
004150  000004                  T0020:  SCOPE
004152  012737  004236  000004          MOV     #A0020,4        ;SETUP VECTOR IN CASE IT IS NONEXISTANT
004160  012737  000340  000006          MOV     #340,6
004166  005037  001124                  CLR     $GDDAT
004172  012737  004220  001106          MOV     #R0020,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
004200  012737  177566  001120          MOV     #177566,$GDADR  ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 4 SET
004206  013746  000340                  MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
```

```
                                        MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
                                        RTI                     ;; POP NEW PC AND PS
                                64$:
                                R0020:  MOV     #1000,SP        ;SETUP THE STACK
                                        CLR     LKS
004230  012777  000100  174662  I0020:  MOV     #100,@$GDADR    ;SET THE "CLOSE" ADDRESS TO = 100
004236  032737  000100  177546  A0020:  BIT     #100,LKS        ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
004244  001401                          BEQ     B0020
004246  104005                  E0020:  ERROR 5                 ;IT AFFECTED "LKS"  --  ERROR
004250  005037  177546          B0020:  CLR     LKS
004254  012737  004266  000004          MOV     #T0021  ,4
004262  005077  174632                  CLR     @$GDADR         ;CLEAR OUT THE "CLOSE" ADDRESS


                                .SBTTL  LINE CLOCK REGISTER ADDRESSING TEST
                                ;LINE CLOCK REGISTER ADDRESSING TEST
004266  000004                 T0021:  SCOPE
004270  012737  004354  000004          MOV     #A0021,4        ;SETUP VECTOR IN CASE IT IS NONEXISTANT
004276  012737  000340  000006          MOV     #340,6
004304  005037  001124                  CLR     $GDDAT
004310  012737  004336  001106          MOV     #R0021,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
004316  012737  177746  001120          MOV     #177746,$GDADR  ;SAME AS "LKS" ADDRESS EXCEPT WITH BIT 7 SET
004324  013746  000340                  MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
004330  012746  004336                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
004334  000002                          RTI                     ;; POP NEW PC AND PS
004336                          64$:
004336  012706  001000          R0021:  MOV     #1000,SP        ;SETUP THE STACK
004342  005037  177546                  CLR     LKS
004346  012777  000100  174544  I0021:  MOV     #100,@$GDADR    ;SET THE "CLOSE" ADDRESS TO = 100
004354  032737  000100  177546  A0021:  BIT     #100,LKS        ;MAKE SURE THAT "LKS" WAS NOT AFFECTED
004362  001401                          BEQ     B0021
004364  104005                  E0021:  ERROR 5                 ;IT AFFECTED "LKS"  --  ERROR
004366  005037  177546          B0021:  CLR     LKS
004372  012737  004404  000004          MOV     #T0022  ,4
004400  005077  174514                  CLR     @$GDADR         ;CLEAR OUT THE "CLOSE" ADDRESS


                                .SBTTL  LINE CLOCK REGISTER HIGH BYTE TEST
                                ;LINE CLOCK REGISTER HIGH BYTE TEST
                                ;MAKE SURE THE LKS REGISTER LOW BYTE RESPONDS TO THE HIGH BYTES ADDRESS
004404  000004                 T0022:  SCOPE
004406  000137  004520                  JMP     T0023
004412  012737  006252  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
004420  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
004426  012737  000100  001124          MOV     #100,$GDDAT
004434  012737  004462  001106          MOV     #R0022,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
004442  012737  177547  001120          MOV     #177547,$GDADR  ;HIGH BYTE OF THE LKS REGISTER
004450  013746  000340                  MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
004454  012746  004462                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
004460  000002                          RTI                     ;; POP NEW PC AND PS
004462                          64$:
004462  012706  001000          R0022:  MOV     #1000,SP        ;SETUP THE STACK
004466  005037  177546                  CLR     LKS
004472  112777  000100  174420  I0022:  MOVB    #100,@$GDADR    ;SET THE HIGH BYTE ADDRESS TO = 100
```

```
                                        BIT     #100,LKS        ;MAKE SURE THAT "LKS" LOW BYTE WAS AFFECTED
                                        BNE     A0022
                                E0022:  ERROR 5                 ;SHOULD HAVE SET BIT 7.  ERROR
                                A0022:  CLR     LKS
                                        RESET


                                .SBTTL  CLOCK FLAG BIT TEST
                                ;CLOCK FLAG BIT TEST
                                T0023:  SCOPE
                                        MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
                                        MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
                                        MOV     #200,$GDDAT
                                        MOV     #R0023,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
                                R0023:  CLR     LKS
                                        CLR     R3              ;INITIALIZE A COUNTER LOCATION
                                A0023:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
                                        BMI     B0023           ;IF SO, CONTINUE ON WITH THE TEST
                                        INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
                                        BNE     A0023           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
                                E0023:  ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
                                        BR      T0024
                                B0023:
                                I0023:  MOV     #200,LKS        ;MOVE A 1 INTO THE CLOCK FLAG BIT
                                        CMP     LKS,$GDDAT      ;SHOULD NOT AFFECT THE FLAG BIT
                                        BEQ     T0024
                                E10023: ERROR 1                 ;CLOCK FLAG DID NOT CLEAR


                                .SBTTL  INTERRUPT TEST
                                T0024:  SCOPE
                                        MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
                                        MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
                                        MOV     #200,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
                                        MOV     #R0024,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
                                        MOV     #E0024,100
                                        CLR     LKS             ;ALLOW CLOCK INTERRUPTS
                                R0024:
                                        MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
                                        MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
                                        RTI                     ;; POP NEW PC AND PS
                                64$:
                                        MOV     #300,LKS
                                A0024:  INC     #0              ;WAIT FOR 20+ MS
                                        BNE     A0024           ;LOOP BACK IF NOT DONE WAITING
                                        RESET                   ;RESET SHOULD CLEAR INTERRUPT ENABLE
                                        CMP     $GDDAT,LKS      ;AND LEAVE THE CLOCK FLAG SET
                                        BEQ     T0025           ;GO ON TO THE NEXT TEST IF IT DID
                                E0024:  ERROR 1                 ;RESET SET INTERRUPT ENABLE OR CLEARED CLOCK FLAG


                                .SBTTL  NO SACK TIMEOUT TEST
                                ;NO SACK TIMEOUT TEST
```

```
                                        T0025:  SCOPE
000724  000004                                  
000726  012737  006252  000004                  MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
000734  012737  000340  000006                  MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
000742  012737  000300  001124                  MOV     #300,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
000750  012737  004772  001106                  MOV     #R0025,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
000756  012737  000340  000102                  MOV     #340,102        ;NO INTERRUPTS AFTER THE FIRST ONE
000764  012737  005054  000100                  MOV     #C0025,100
000772  005037  177546          R0025:  CLR     LKS
000776  013746  000340                  MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
005002  012746  005010                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
005006  000002                          RTI                     ;; POP NEW PC AND PS
005010                          64$:
005010  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
005012  105737  177546          A0025:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
005016  100404                          BMI     B0025           ;IF SO, CONTINUE ON WITH THE TEST
005020  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
005022  001373                          BNE     A0025           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
005024  104001                  E0025:  ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
005026  000417                          BR      T0026
005030                          B0025:
005030  005046                          CLR     -(SP)           ;DROP CPU PRIORITY LEVEL
005032  012746  005040                  MOV     #1$,-(SP)       ;SET RETURN 'PC' FROM THE RTI
005036  000002                          RTI                     ;RETURN TO NEXT INSTRUCTION
005040  012737  000100  177546  1$:     MOV     #100,LKS        ;ENABLE CLOCK INTERRUPTS
005046  000001                          WAIT                    ;THE ONLY WAY TO LEAVE HERE WITHOUT ERROR IS TO INTERRUP
005050  104006                  E10025: ERROR 6                 ;IF IT ERROR 1S HERE ITS BECAUSE OF A "NO-SACK" TIMEOUT
005052  000405                          BR      T0026
005054  022737  000300  177546  C0025:  CMP     #300,LKS
005062  001401                          BEQ     T0026           ;FIND OUT WHAT THE INTERRUPT DID TO THE CLOCK STATUS REG
005064  104001                  E20025: ERROR 1                 ;IT CLEARED THE INTERRUPT ENABLE OR THE FLAG BIT


                                        .SBTTL  RESET TEST
                                        ;RESET TEST
005066  000004                         T0026:  SCOPE
005070  012737  006252  000004                  MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
005076  012737  000340  000006                  MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
005104  012737  000200  001124                  MOV     #200,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
005112  012737  005146  001106                  MOV     #R0026,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
005120  013746  000340                  MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
005124  012746  005132                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
005130  000002                          RTI                     ;; POP NEW PC AND PS
005132                          64$:
005132  012737  005204  000100                  MOV     #E0026,100
005140  012737  000140  000102                  MOV     #140,102        ;SETUP STATUS FOR AFTER THE INTERRUPT
005146  005037  177546          R0026:  CLR     LKS
005152  012706  001000                  MOV     #1000,SP        ;SETUP THE STACK
005156  012737  000100  177546          MOV     #100,LKS        ;SET INTERRUPT ENABLE BIT NOW
005164  005227  000000          A0026:  INC     #0              ;WAIT FOR CLOCK FLAG TO SET
005170  001375                          BNE     A0026
005172  000005                  I0026:  RESET                   ;RESET SHOULD NOT CLEAR THE FLAG
005174  023737  177546  001124          CMP     LKS,$GDDAT      ;FIND OUT IF ID DIT DID OR NOT
005202  001401                          BEQ     T0027
005204  104001                  E0026:  ERROR 1                 ;RESET DID NOT INITIALIZE THE LKS WORD CORRECTLY
```

```
                                        .SBTTL  CLOCK FLAG BIT TEST
                                        ;CLOCK FLAG BIT TEST
                                        ;MAKE SURE IT DOESNT CLEAR WHEN YOU TRY TO SET IT VIA A 'MOV' INSTRUCTION
005206  000004                  T0027:  SCOPE
005210  012737  006252  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
005216  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
005224  012737  000200  001124          MOV     #200,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
005232  012737  005252  001106          MOV     #R0027,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
005240  005037  000102                  CLR     102
005244  012737  005334  000100          MOV     #T0030  ,100
005252  005037  177546          R0027:  CLR     LKS
005256  013746  000340                  MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
005262  012746  005270                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
005266  000002                          RTI                     ;;  POP NEW PC AND PS
005270                          64$:
005270  012706  001000                  MOV     #1000,SP        ;SETUP THE STACK
005274  005037  001234                  CLR     WORD            ;SETUP A COUNTER LOCATION TO = 0
005300  005237  001234          A0027:  INC     WORD
005304  001375                          BNE     A0027           ;WASTE TIME LOOPING UNTIL THE COUNTER REACHES 0
005306  012737  000300  177546          MOV     #300,LKS        ;SET INTERRUPT ENABLE AND TRY TO SET THE CLOCK FLAG
005314  012737  000200  177546          MOV     #200,LKS        ;TRY TO SET IT AGAIN
005322  023737  177546  001124          CMP     LKS,$GDDAT      ;DID THE CLOCK FLAG STAY SET?
005330  001401                          BEQ     T0030           ;IF NOT GO ON TO THE NEXT TEST
005332  104001                  E0027:  ERROR 1                 ;ERROR - MOVED A '1' INTO THE CLOCK FLAG BIT AND IT STAY


                                        .SBTTL  CLOCK FLAG AFTER INTERRUPT TEST
                                        ;SEE IF AN INTERRUPT CLEARS THE CLOCK FLAG
005334  000004                  T0030:  SCOPE
005336  012737  006252  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
005344  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
005352  005037  177546                  CLR     LKS             ;NO CLOCK INTERRUPTS BEFORE WE ARE READY
005356  012737  000100  001124          MOV     #100,$GDDAT     ;HAVE GOOD DATA INFO READY FOR TYPEOUT IN CASE OF AN ERR
005364  012737  005372  001106          MOV     #R0030,$LPADR   ;INITIALIZE THE LOOPBACK ADDRESS IN CASE OF AN ERROR
005372  012737  005440  000100  R0030:  MOV     #A0030,100      ;SETUP CLOCK INTERRUPT VECTOR
005400  005037  000102                  CLR     102             ;PRIORITY LEVEL WILL ALLOW FURTHER INTERRUPTS
005404  005046                          CLR     -(SP)           ;DROP CPU PRIORITY LEVEL
005406  012746  005414                  MOV     #1$,-(SP)       ;SET RETURN 'PC' FROM THE RTI
005412  000002                          RTI                     ;RETURN TO NEXT INSTRUCTION
005414  012706  001000          1$:     MOV     #1000,SP        ;SETUP THE STACK
005420  005037  177546                  CLR     LKS
005424  105037  001234                  CLRB    WORD            ;CLEAR OUT A COUNTER LOCATION
005430  012737  000100  177546          MOV     #100,LKS        ;ENABLE CLOCK INTERRUPTS NOW
005436  000001                          WAIT                    ;WAIT FOR AN INTERRUPT
005440  012737  005472  000100  A0030:  MOV     #E0030,100      ;ERROR IF WE INTERRUPT AGAIN
005446  005046                          CLR     -(SP)           ;DROP CPU PRIORITY LEVEL
005450  012746  005456                  MOV     #B0030,-(SP)    ;SET RETURN 'PC' FROM THE RTI
005454  000002                          RTI                     ;RETURN TO NEXT INSTRUCTION
005456  105237  001234          B0030:  INCB    WORD            ;DO A NOTHING LOOP FOR A VERY SHORT PERIOD OF TIME
005462  001375                          BNE     B0030           ;WE SHOULD INCREMENT TO 0 LONG BEFORE AN INTERRUPT COMES
005464  005037  177546                  CLR     LKS
005470  000401                          BR      T0031
005472  104001                  E0030:  ERROR 1                 ;INTERRUPT DID NOT CLEAR THE CLOCK FLAG
```

```
                                        .SBTTL  NO INTERRUT AT PRIORITY 7 TEST
                                        ;TEST THAT CLOCK WILL NOT INTERRUPT WITH PROCESSR AT PRIORITY 7
005474  000004                  T0031:  SCOPE
005476  012737  006252  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
005504  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
005512  012737  005520  001106          MOV     #R0031,$LPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
005520  005037  177546          R0031:  CLR     LKS
005524  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
005526  105737  177546          A0031:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
005532  100404                          BMI     B0031           ;IF SO, CONTINUE ON WITH THE TEST
005534  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
005536  001373                          BNE     A0031           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
005540  104001                  E0031:  ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
005542  000431                          BR      T0032
005544                          B0031:
005544  012706  001000                  MOV     #1000,SP        ;INITIALIZE THE STACK POINTER
005550  013746  000340                  MOV     PR7,-(SP)       ;;PUT NEW PS ON STACK
005554  012746  005562                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
005560  000002                          RTI                     ;; POP NEW PC AND PS
005562                          64$:
005562  012737  005540  000100          MOV     #E0031,100      ;SET UP VECTOR RETURN
005570  012737  000100  177546          MOV     #100,LKS        ;ENABLE INTERRUPT
005576  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
005600  105737  177546          C0031:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
005604  100404                          BMI     D0031           ;IF SO, CONTINUE ON WITH THE TEST
005606  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
005610  001373                          BNE     C0031           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
005612  104001                  E10031: ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
005614  000404                          BR      T0032
005616                          D0031:
005616  000240                          NOP
005620  000240                          NOP                     ;GIVE CLOCK EXTRA TIME TO INTERRUPT
005622  000401                          BR      T0032
005624  104003                  E20031: ERROR 3                 ;ERROR, CLOCK INTERRUPTED WITHOUT HAVING PRIORITY


                                        .SBTTL  CC PUSH TEST FOR CLOCK INTERRUPTS
                                        ;TEST THAT CLOCK INTERRUPT PUSHES CONDITION CODES ONTO STACK
005626  000004                  T0032:  SCOPE
005630  012737  006252  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
005636  012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
005644  012737  005652  001106          MOV     #R0032,$LPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
005652  005037  177546          R0032:  CLR     LKS
005656  005003                          CLR     R3              ;INITIALIZE A COUNTER LOCATION
005660  105737  177546          A0032:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
005664  100404                          BMI     B0032           ;IF SO, CONTINUE ON WITH THE TEST
005666  005203                          INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
005670  001373                          BNE     A0032           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
005672  104001                  E0032:  ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
005674  000433                          BR      T0033
005676                          B0032:
005676  012706  001000                  MOV     #1000,SP        ;INITIALIZE THE STACK POINTER
```

```
                        005037  000776          CLR     BUF1
                        005037  000774          CLR     BUF2
                        012737  005742  000100  MOV     #C0032,100      ;SET UP VECTOR RETURN
                        012737  000100  177546  MOV     #100,LKS        ;ENABLE INTERRUPT
                        005046                  CLR     -(SP)           ;DROP CPU PRIORITY LEVEL
                        012746  005736          MOV     #1$,-(SP)       ;SET RETURN 'PC' FROM THE RTI
                        000002                  RTI                     ;RETURN TO NEXT INSTRUCTION
                        000277          1$:     +SEC!SEV!SEZ!SEN        ;SET ALL CONDITION CODES
                        000001                  WAIT                    ;WAIT FOR INTERRUPT
                        022737  000017  000776  C0032:  CMP     #17,BUF1
                        001405                  BEQ     T0033
                        012737  000017  001124          MOV     #17,$GDDAT
                        104007                  ERROR 7                 ;ERROR DID NOT PUSH CORRECT PS ONTO STACK
                        000721                  BR      T0032


                                        .SBTTL  PC PUSH TEST FOR CLOCK INTERRUPTS
                                        ;TEST THAT CLOCK INTERRUPT PUSHES THE PROGRAM COUNTER ONTO STACK
                        000004  T0033:  SCOPE
                        012737  006252  000004          MOV     #TRAP0,@#4      ;SETUP VECOR IN CASE OF UNFORSEEN PROBLEMS
                        012737  000340  000006          MOV     #340,@#6        ;NO INTERRUPTS WHILE PRINTING FATAL MESSAGE
                        012737  006010  001106          MOV     #R0033,$LPADR   ;SETUP LOOPBACK ADDRESS IN CASE OF AN ERROR
                        005037  177546  R0033:  CLR     LKS
                        005003                  CLR     R3              ;INITIALIZE A COUNTER LOCATION
                        105737  177546  A0033:  TSTB    LKS             ;IS THE CLOCK FLAG SET?
                        100404                  BMI     B0033           ;IF SO, CONTINUE ON WITH THE TEST
                        005203                  INC     R3              ;IF NOT INCREMENT THE COUNTER LOCATION
                        001373                  BNE     A0033           ;AND GO TEST THE CLOCK FLAG AGAIN. UNLESS...
                        104001          E0033:  ERROR 1                 ;CLOCK FLAG DID NOT SET AFTER A WAITING PERIOD > 20 MS
                        000432                  BR      T0034
                                        B0033:
                        012706  001000          MOV     #1000,SP        ;INITIALIZE THE STACK POINTER
                        005037  000776          CLR     BUF1
                        005037  000774          CLR     BUF2
                        012737  006100  000100  MOV     #C0033,100      ;SET UP VECTOR RETURN
                        012737  000100  177546  MOV     #100,LKS        ;ENABLE INTERRUPT
                        005046                  CLR     -(SP)           ;DROP CPU PRIORITY LEVEL
                        012746  006074          MOV     #1$,-(SP)       ;SET RETURN 'PC' FROM THE RTI
                        000002                  RTI                     ;RETURN TO NEXT INSTRUCTION
                        000277          1$:     +SEC!SEV!SEZ!SEN        ;SET ALL CONDITION CODES
                        000001                  WAIT                    ;WAIT FOR INTERRUPT
                        022737  006100  000774  C0033:  CMP     #C0033,BUF2
                        001404                  BEQ     T0034
                        012737  006100  001124  E10033: MOV     #C0033,$GDDAT
                        104010                  ERROR 10                ;ERROR, DID NOT PUSH CORRECT PC ONTO STACK


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

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

```
                                        .SBTTL  END OF PASS ROUTINE
                                        ;*INCREMENT THE PASS NUMBER ($PASS)
                                        ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                                        ;*IF THERES A MONITOR GO TO IT
                                        ;*IF THERE ISN'T JUMP TO TOOO1
                                        $EOP:
006130                                          SCOPE
006130  000004                                    CLR     $TSTNM          ;;ZERO THE TEST NUMBER
006132  005037  001102                            CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
006136  005237  001220                            INC     $PASS           ;;INCREMENT THE PASS NUMBER
006142  042737  100000  001100                    BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
006146  005327                                    DEC     (PC)+           ;;LOOP?
006150  000001                            $EOPCT: .WORD   1
006152  003022                                    BGT     $DOAGN          ;;YES
006154  012737                                    MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
006160  000001                            $ENDCT: .WORD   1
006162  006156                                    $EOPCT
006164  104400  006232                            TYPE    ,$ENDMG         ;;TYPE "END PASS #"
006170  013746  001100                            MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
006174  104404                                    TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
006176  104400  006247                            TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
006202                                  $GET42:
006202  013700  000042                            MOV     @#42,R0         ;;GET MONITOR ADDRESS
006206  001405                                    BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
006210  000005                                    RESET                   ;;CLEAR THE WORLD
006212  004710                            $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
006214  000240                                    NOP                     ;;SAVE ROOM
006216  000240                                    NOP                     ;;FOR
006220  000240                                    NOP                     ;;ACT11
006222                                  $DOAGN:
006222  000137  001672                            JMP     @#TOOO1         ;;RETURN
006226  005015  047105  020104  $ENDMG: .ASCIZ   <15><12>/END PASS #/
006234  040520  051523  021440
006242
006243    377     377     000   $ENULL: .BYTE   -1,-1,0                 ;;NULL CHARACTER STRING
006246  005046                    TRAPO:  CLR     -(SP)
006250  004737  006576                    JSR     PC,$TYPE        ;PRINTOUT"TRAPPED TO 4 FROM "
006254  010363                            TRPMES                  ;ADDRESS OF THE MESSAGE
006256  011646                            MOV     (SP),-(SP)      ;GET THE ADDRESS WHERE THE TRAP OCCURED
006260  162716  000002                    SUB     #2,(SP)         ;MAKE IT RIGHT
006264  104401                            TYPOC                   ;TYPE OUT ADDRESS IN OCTAL
006266  104400  001231                    TYPE    ,$CRLF          ;PRINTOUT A CARRIAGE RETURN-LINE FEED
006272  005046                            CLR     -(SP)
006274  004737  006576                    JSR     PC,$TYPE        ;PRINTOUT RESTARTING MESSAGE
006300  010424                            TRPM2S                  ;ADDRESS OF RESTART MESSAGE
006302  000240                            NOP
006304  000240                            NOP
006306  000240                            NOP
006310  000240                            NOP
006312  000005                            RESET
006314  000137  001450                    JMP     START
                                ;******************************************************************
```

```
                                        .SBTTL  SCOPE HANDLER ROUTINE

                                        ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                        ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                                        ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW14=1          LOOP ON TEST
                                        ;*SW11=1          INHIBIT ITERATIONS
                                        ;*SW09=1          LOOP ON ERROR
                                        ;*SW08=1          LOOP ON TEST IN SWR<7:0>
                                        ;*CALL
                                        ;*      SCOPE               ;;SCOPE=IOT

006324                                  $SCOPE:
006324  000240                                  NOP
006326  032777  040000  172602  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
006334  001111                          BNE     $OVER           ;;YES IF SW14=1
                                        ;#####START OF CODE FOR THE XOR TESTER#####
006336  000416                          $XTSTR: BR      6$      ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
006340  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
006344  012737  006364  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
006352  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
006356  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
006362  000463                          BR      $SVLAD          ;;GO TO THE NEXT TEST
006364  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
006366  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
006372  000423                          BR      7$              ;;LOOP ON THE PRESENT TEST
006374                          6$:;#####END OF CODE FOR THE XOR TESTER#####
006374  032777  000400  172534          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
006402  001404                          BEQ     2$              ;;BR IF NO
006404  127737  172526  001102          CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
006412  001464                          BEQ     $OVER           ;;BR IF YES
006414  105737  001103          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
006420  001421                          BEQ     3$              ;;BR IF NO
006422  123737  001115  001103          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
006430  101015                          BHI     3$              ;;BR IF NO
006432  032777  001000  172476          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
006440  001404                          BEQ     4$              ;;BR IF NO
006442  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
006450  000443                          BR      $OVER
006452  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
006456  005037  001220                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
006462  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
006464  032777  004000  172444  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
006472  001011                          BNE     1$              ;;BR IF YES
006474  005737  001100                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
006500  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
006502  005237  001104                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
006506  023737  001220  001104          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
006514  002021                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
006516  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
006524  013737  006574  001220          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
006532  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
006536  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
006542  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
```

```
                                        CLR     $ESCAPE                 ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
                                        MOVB    #1,$ERMAX               ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
                                $OVER:  MOV     $TSTNM,@DISPLAY         ;;DISPLAY TEST NUMBER
                                        MOV     $LPADR,(SP)             ;;FUDGE RETURN ADDRESS
                                        RTI                             ;;FIXES PS
                                $MXCNT: 10                              ;;MAX. NUMBER OF ITERATIONS
                                ;***************************************************************

                                .SBTTL  TYPE ROUTINE

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

                                $TYPE:  TSTB    $TPFLG                  ;;IS THERE A TERMINAL?
                                        BPL     1$                      ;;BR IF YES
                                        HALT                            ;;HALT HERE IF NO TERMINAL
                                        BR      3$                      ;;LEAVE
                                1$:     MOV     R0,-(SP)                ;;SAVE R0
                                        MOV     @2(SP),R0               ;;GET ADDRESS OF ASCIZ STRING
                                2$:     MOVB    (R0)+,-(SP)             ;;PUSH CHARACTER TO BE TYPED ONTO STACK
                                        BNE     4$                      ;;BR IF IT ISN'T THE TERMINATOR
                                        TST     (SP)+                   ;;IF TERMINATOR POP IT OFF THE STACK
                                60$:    MOV     (SP)+,R0                ;;RESTORE R0
                                3$:     ADD     #2,(SP)                 ;;ADJUST RETURN PC
                                        RTI                             ;;RETURN
                                4$:     CMPB    #THT,(SP)               ;;BRANCH IF <HT>
                                        BEQ     8$
                                        CMPB    #TCRLF,(SP)             ;;BRANCH IF NOT <CRLF>
                                        BNE     5$
                                        TST     (SP)+                   ;;POP  <CR><LF> EQUIV
                                        TYPE                            ;;TYPE A CR AND LF
                                        $CRLF
                                        BR      2$                      ;;GET NEXT CHARACTER
                                5$:     JSR     PC,$TYPEC               ;;GO TYPE THIS CHARACTER
                                6$:     CMPB    $FILLC,(SP)+            ;;IS IT TIME FOR FILLER CHARS.?
                                        BNE     2$                      ;;IF NO GO GET NEXT CHAR.
                                        MOV     $NULL,-(SP)             ;;GET # OF FILLER CHARS. NEEDED
                                                                        ;;AND THE NULL CHAR.
                                7$:     DECB    1(SP)                   ;;DOES A NULL NEED TO BE TYPED?
                                        BLT     6$                      ;;BR IF NO--GO POP THE NULL OFF OF STACK
                                        JSR     PC,$TYPEC               ;;GO TYPE A NULL
                                        DECB    $CHARCNT                ;;DO NOT COUNT AS A COUNT
                                        BR      7$                      ;;LOOP

                                ;HORIZONTAL TAB PROCESSOR
```

```
                                        ;
006716  112716  000040          B$:     MOVB    #40,(SP)        ;;REPLACE TAB WITH SPACE
006722  004737  006742          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
006726  132737  000007  007006          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
006734  001372                          BNE     9$              ;;TAB STOP
006736  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
006740  000726                          BR      2$              ;;GET NEXT CHARACTER
006742  105777  172200      $TYPEC:     TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
006746  100375                          BPL     $TYPEC
006750  116677  000002  172172          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
006756  122766  000015  000002          CMPB    #15,2(SP)       ;;BRANCH IF
006764  001003                          BNE     1$              ;;NOT <CR>
006766  105037  007006                  CLRB    $CHARCNT        ;;
006772  000406                          BR      $TYPEX          ;;EXIT
006774  122766  000012  000002  1$:     CMPB    #12,2(SP)       ;;BRANCH IF
007002  002002                          BGE     $TYPEX          ;;<LF>
007004  105227                          INCB    (PC)+           ;;INC SPACE
007006  000000          $CHARCNT:.WORD  0               ;;COUNT
007010  000207          $TYPEX: RTS     PC
                                ;;      EQUATES
                                ;;
000011                          THT=11
000200                          TCRLF=200

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

                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

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

007012                          $TYPDS:
007012  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
007014  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
007016  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
007020  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
007022  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
007024  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
007030  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
007034  100004                          BPL     1$              ;;BR IF INPUT IS POS.
007036  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
007040  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
007046  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
007050  012703  007226                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
007054  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
007060  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
007062  016001  007216                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
007066  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
007070  002402                          BLT     4$              ;;BR IF DONE
007072  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
007074  000774                          BR      3$
```

```
1344  007076  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
1345  007100  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
1346  007102  001002                          BNE     5$              ;;FALL THROUGH IF 0
1347  007104  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
1348  007106  100407                          BMI     7$              ;;BR IF YES
1349  007110  106316                  5$:     ASLB    (SP)            ;;MSD?
1350  007112  103003                          BCC     6$              ;;BR IF NO
1351  007114  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
1352  007122  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
1353  007126  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
1354  007132  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
1355  007134  005720                          TST     (R0)+           ;;JUST INCREMENTING
1356  007136  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
1357  007142  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
1358  007144  003002                          BGT     8$              ;;GO TO EXIT
1359  007146  010502                          MOV     R5,R2           ;;GET THE LSD
1360  007150  000764                          BR      6$              ;;GO CHANGE TO ASCII
1361  007152  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
1362  007154  100003                          BPL     9$              ;;BR IF NO
1363  007156  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
1364  007164  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
1365  007166  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
1366  007170  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
1367  007172  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
1368  007174  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
1369  007176  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
1370  007200  104400  007226                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
1371  007204  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
1372  007212  012616                          MOV     (SP)+,(SP)
1373  007214  000002                          RTI                     ;;RETURN TO USER
1374  007216  023420                  $DTBL:  10000.
1375  007220  001750                          1000.
1376  007222  000144                          100.
1377  007224  000012                          10.
1378  007226  000004                  $DBLK:  .BLKW   4
1379                                  ;*********************************************************************
1380
1381                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
1382
1383                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
1384                                  ;*OCTAL (ASCII) NUMBER AND TYPE IT.
1385                                  ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
1386                                  ;*CALL:
1387                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
1388                                  ;*      TYPOS                   ;;CALL FOR TYPEOUT
1389                                  ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
1390                                  ;*      .BYTE   M               ;;M=1 OR 0
1391                                  ;*                                      ;;1=TYPE LEADING ZEROS
1392                                  ;*                                      ;;0=SUPPRESS LEADING ZEROS
1393                                  ;*
1394                                  ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
1395                                  ;*$TYPOS OR $TYPOC
1396                                  ;*CALL:
1397                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
1398                                  ;*      TYPON                   ;;CALL FOR TYPEOUT
1399                                  ;*
```

```
                                        ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPOC                   ;;CALL FOR TYPEOUT
                                        $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
007236  017646  000000                          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
007242  116637  000001  007461                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
007250  062716  000002                          ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
007254  000406                                  BR      $TYPON
007256  112737  000001  007461  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
007264  112737  000006  007463          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
007272  112737  000005  007460  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
007300  010346                          MOV     R3,-(SP)        ;;SAVE R3
007302  010446                          MOV     R4,-(SP)        ;;SAVE R4
007304  010546                          MOV     R5,-(SP)        ;;SAVE R5
007306  113704  007463                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
007312  005404                          NEG     R4
007314  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
007320  110437  007462                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
007324  113704  007461                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
007330  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
007334  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
007336  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
007340  000404                          BR      3$              ;;GO DO MSB
007342  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
007344  006105                          ROL     R5
007346  006105                          ROL     R5
007350  010503                          MOV     R5,R3
007352  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
007354  105337  007462                  DECB    $OMODE          ;;TYPE THIS DIGIT?
007360  100016                          BPL     7$              ;;BR IF NO
007362  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
007366  001002                          BNE     4$              ;;TEST FOR 0
007370  005704                          TST     R4              ;;SUPPRESS THIS 0?
007372  001403                          BEQ     5$              ;;BR IF YES
007374  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
007376  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
007402  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
007406  110337  007456                  MOVB    R3,8$           ;;SAVE FOR TYPING
007412  104400  007456                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
007416  105337  007460          7$:     DECB    $OCNT           ;;COUNT BY 1
007422  003347                          BGT     2$              ;;BR IF MORE TO DO
007424  002402                          BLT     6$              ;;BR IF DONE
007426  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
007430  000744                          BR      2$              ;;GO DO THE LAST DIGIT
007432  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
007434  012604                          MOV     (SP)+,R4        ;;RESTORE R4
007436  012603                          MOV     (SP)+,R3        ;;RESTORE R3
007440  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
007446  012616                          MOV     (SP)+,(SP)
007450  000002                          RTI                     ;;RETURN
007452      000                 8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
007453      000                         .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
007454      000                 $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
007455      000                 $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
```

```
                                $OMODE: .WORD   0                       ;;NUMBER OF DIGITS TO TYPE
                                ;*******************************************************************

                                .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE

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

007464                          $ERRTYP:
007464  104400  001231                TYPE    ,$CRLF                  ;;"CARRIAGE RETURN" & "LINE FEED"
007470  010046                        MOV     R0,-(SP)                ;;SAVE R0
007472  005000                        CLR     R0                      ;;PICKUP THE ITEM INDEX
007474  153700  001114                BISB    @#$ITEMB,R0
007500  001004                        BNE     1$                      ;;IF ITEM NUMBER IS ZERO, JUST
                                                                        ;;TYPE THE PC OF THE ERROR
007502  013746  001116                MOV     $ERRPC,-(SP)            ;;SAVE $ERRPC FOR TYPEOUT
                                                                        ;;ERROR ADDRESS
007506  104401                        TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
007510  000426                        BR      6$                      ;;GET OUT
007512  005300                1$:     DEC     R0                      ;;ADJUST THE INDEX SO THAT IT WILL
007514  006300                        ASL     R0                      ;;      WORK FOR THE ERROR TABLE
007516  006300                        ASL     R0
007520  006300                        ASL     R0
007522  062700  001236                ADD     #$ERRTB,R0              ;;FORM TABLE POINTER
007526  012037  007536                MOV     (R0)+,2$                ;;PICKUP "ERROR MESSAGE" POINTER
007532  001404                        BEQ     3$                      ;;SKIP TYPEOUT IF NO POINTER
007534  104400                        TYPE                            ;;TYPE THE "ERROR MESSAGE"
007536  000000                2$:     .WORD   0                       ;;"ERROR MESSAGE" POINTER GOES HERE
007540  104400  001231                TYPE    ,$CRLF                  ;;"CARRIAGE RETURN" & "LINE FEED"
007544  012037  007554        3$:     MOV     (R0)+,4$                ;;PICKUP "DATA HEADER" POINTER
007550  001404                        BEQ     5$                      ;;SKIP TYPEOUT IF 0
007552  104400                        TYPE                            ;;TYPE THE "DATA HEADER"
007554  000000                4$:     .WORD   0                       ;;"DATA HEADER" POINTER GOES HERE
007556  104400  001231                TYPE    ,$CRLF                  ;;"CARRIAGE RETURN" & "LINE FEED"
007562  011000                5$:     MOV     (R0),R0                 ;;PICKUP "DATA TABLE" POINTER
007564  001004                        BNE     7$                      ;;GO TYPE THE DATA
007566  012600                6$:     MOV     (SP)+,R0                ;;RESTORE R0
007570  104400  001231                TYPE    ,$CRLF                  ;;"CARRIAGE RETURN" & "LINE FEED"
007574  000207                        RTS     PC                      ;;RETURN
007576                          7$:
007576  013046                        MOV     @(R0)+,-(SP)            ;;SAVE @(R0)+ FOR TYPEOUT
007600  104401                        TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
007602  005710                        TST     (R0)                    ;;IS THERE ANOTHER NUMBER?
007604  001770                        BEQ     6$                      ;;BR IF NO
007606  104400  007614                TYPE    ,8$                     ;;TYPE TWO(2) SPACES
007612  000771                        BR      7$                      ;;LOOP
007614  020040      000       8$:     .ASCIZ  /  /                    ;;TWO(2) SPACES
007620                                .EVEN
                                ;*******************************************************************

                                .SBTTL  ERROR HANDLER ROUTINE

                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                ;*AND GO TO $ERRTYP ON ERROR
```

```
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW15=1        HALT ON ERROR
                                        ;*SW13=1        INHIBIT ERROR TYPEOUTS
                                        ;*SW10=1        BELL ON ERROR
                                        ;*SW09=1        LOOP ON ERROR
                                        ;*CALL
                                        ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

007620                                  $ERROR:
007620  010637  001174                          MOV     SP,$REG6        ;GET THE CURRENT STACK POINTER VALUE
007624  162737  000004  001174                  SUB     #4,$REG6                ;RESTORE IT TO ITS "PRE ERROR TRAP" VALUE FOR PR
007632  016637  000002  001176                  MOV     2(SP),$REG7             ;GET THE PS OFF OF THE STACK
007640  005037  001172                          CLR     $REG5                   ;PREPARE "$REG5" TO HOLD THE TEST #
007644  113737  001102  001172                  MOVB    $TSTNM,$REG5            ;TEST # IS HELD IN THE LOW BYTE OF "TSTNM"
007652  010037  001160                          MOV     R0,$REG0                ;MOST OF THE TIME R0 HAS GOOD STUFF IN IT ALSO
007656  105237  001103                  7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
007662  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
007664  013777  001102  171246          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
007672  032777  002000  171236          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
007700  001402                          BEQ     1$              ;;NO - SKIP
007702  104400  001224                  TYPE    ,$BELL          ;;RING BELL
007706  005237  001112                  1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
007712  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
007716  162737  000002  001116          SUB     #2,$ERRPC
007724  117737  171166  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
007732  032777  020000  171176          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
007740  001004                          BNE     20$             ;;SKIP TYPEOUTS
007742  004737  007464                  JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
007746  104400  001231                  TYPE    ,$CRLF
007752                                  20$:
007752  005777  171160                  2$:     TST     @SWR            ;;HALT ON ERROR
007756  100001                          BPL     31$             ;;SKIP IF CONTINUE
007760  000000                          HALT                    ;;HALT ON ERROR!
007762  022737  006216  000042  31$:     CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
007770  001001                          BNE     3$              ;;BRANCH IF NO
007772  000000                          HALT                    ;;YES
007774  032777  001000  171134  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
010002  001402                          BEQ     4$              ;;BR IF NO
010004  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
010010  005737  001222                  4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
010014  001402                          BEQ     5$              ;;BR IF NONE
010016  013716  001222                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
010022                                  5$:
010022  000002                          RTI                     ;;RETURN
                                        ;******************************************************************

                                        .SBTTL  TRAP DECODER

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

010024  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
010026  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
010032  005740                          TST     -(R0)           ;;BACKUP BY 2
```

```
1568  010034  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
1569  010036  006300                          ASL     R0              ;;POSITION FOR INDEXING
1570  010040  016000  010046                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
1571  010044  000200                          RTS     R0              ;;GO TO ROUTINE
1572
1573
1574                                  .SBTTL  TRAP TABLE
1575
1576                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
1577                                  ;*BY THE "TRAP" INSTRUCTION.
1578
1579                                  ;       ROUTINE
1580                                  ;       -------
1581  010046                          $TRPAD:
1582  010046  006576                          $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
1583  010050  007262                          $TYPOC  ;;CALL=TYPOC    TRAP+1(104401)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
1584  010052  007236                          $TYPOS  ;;CALL=TYPOS    TRAP+2(104402)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
1585  010054  007276                          $TYPON  ;;CALL=TYPON    TRAP+3(104403)  TYPE OCTAL NUMBER (AS PER LAST CALL)
1586  010056  007012                          $TYPDS  ;;CALL=TYPDS    TRAP+4(104404)  TYPE DECIMAL NUMBER (WITH SIGN)
1587                                  ;*****************************************************************
1588
1589                                  .SBTTL  POWER DOWN AND UP ROUTINES
1590
1591                                  ;POWER DOWN ROUTINE
1592  010060  012737  010222  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
1593  010066  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
1594  010074  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
1595  010076  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
1596  010100  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
1597  010102  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
1598  010104  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
1599  010106  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
1600  010110  013746  010450                  MOV     POWPUS,-(SP)    ;;PUSH POWPUS ON STACK
1601  010114  013746  010452                  MOV     POWPOP,-(SP)    ;;PUSH POWPOP ON STACK
1602  010120  013746  010240                  MOV     POWMES,-(SP)    ;;PUSH POWMES ON STACK
1603  010124  013746  001672                  MOV     T0001,-(SP)     ;;PUSH T0001 ON STACK
1604  010130  010637  010226                  MOV     SP,$SAVR6       ;;SAVE SP
1605  010134  012737  010146  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
1606  010142  000000                          HALT
1607  010144  000776                          BR      .-2             ;;HANG UP
1608
1609                                  ;POWER UP ROUTINE
1610  010146  013706  010226          $PWRUP: MOV     $SAVR6,SP       ;;GET SP
1611  010152  005037  010226                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
1612  010156  005237  010226          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
1613  010162  001375                          BNE     1$              ;;OF <POWPUS>,<POWPOP>,<POWMES>,<T0001> WORD
1614  010164  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
1615  010166  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
1616  010170  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
1617  010172  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
1618  010174  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
1619  010176  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
1620  010200  012737  010060  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
1621  010206  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
1622  010214  104400                          TYPE                    ;REPORT THE POWER FAILURE
1623  010216  010230          $PWRMG: .WORD   $POWER                  ;;POWER FAIL MESSAGE POINTER
```

```
                                                        RTI
                                                $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
                                                        BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
                                                $SAVR6: 0                       ;;PUT THE SP HERE
                                                $POWER: .ASCIZ  <15><12>"POWER"
                                                        .EVEN
                                                POWMES: .ASCIZ <15> <12> "RESTARTING AFTER A POWER FIALURE" <15> <12> <12>
                                                STMES:  .ASCIZ  <15><12><12>"MD-11-DZKWA-F LINE FREQUENCY CLOCK TEST"<15><12>
                                                TRPMES: .ASCIZ  "TRAPPED TO LOC 4 FROM LOCATION  "
                                                TRPM2S: .ASCIZ  "RESTARTING PROGRAM"
                                                        .EVEN
                                                POWPUS: WORD
                                                POWPOP: WORD
                                                EM1:    .ASCIZ  "                                       LKS     LKS     "
                                                DH1:    .ASCIZ  "(PC)    (PS)    (SP)    TEST#   WAS     S/B     "
                                                        .EVEN
                                                DT1:    $ERRPC,$REG7,$REG6,$REG5,LKS,$GDDAT
```

```
                                                0
EM2:    .ASCIZ  "CLOCK FAILED TO INTERRUPT"

DH2:    .ASCIZ  "(PC)    (PS)    (SP)    TEST#   (LKS)   "

                                        .EVEN
DT2:    $ERRPC,$REG7,$REG6,$REG5,LKS
                                                0
EM3:    .ASCIZ  "CLOCK INTERRUPTED WHEN THE PROCESSOR PRIORITY WAS TOO HIGH"

DH3:    .ASCIZ  "(PC)    (PS)    (SP)    TEST#   (LKS)   "

                                        .EVEN
DT3:    $ERRPC,$REG7,$REG6,$REG5,LKS
                                                0
EM4:    .ASCIZ  "CLOCK GIVES UNEQUAL # OF PULSES OVER TWO EQUAL PERIODS OF TIME"

DH4:    .ASCIZ  "(PC)    (PS)    (SP)    TEST#   1ST     2ND"<15><12>"
```

```
1736  011276  020040  020040  031040
1737  011304  042116  005015  020040
1738  011312  020040  020040  020040
1739  011320  020040  020040  020040
1740  011326  020040  020040  020040
1741  011334  020040  020040  020040
1742  011342  020040  020040  020040
1743  011350  042520  044522  042117
1744  011356  020040  042520  044522
1745  011364  042117     000
1746          011370                  .EVEN
1747  011370  001116  001176  001174  DT4:    $ERRPC,$REG7,$REG6,$REG5,$REG1,$REG2
1748  011376  001172  001162  001164
1749  011404  000000                  0
1750  011406  045514  020123  042522  EM5:    .ASCIZ  "LKS REGISTER RESPONDS TO ANOTHER ADDRESS"
1751  011414  044507  052123  051105
1752  011422  051040  051505  047520
1753  011430  042116  020123  047524
1754  011436  040440  047516  044124
1755  011444  051105  040440  042104
1756  011452  042522  051523     000
1757  011457     050  041520  020051  DH5:    .ASCIZ  "(PC)    (PS)    (SP)    TEST#   ADDRESS"
1758  011464  020040  024040  051520
1759  011472  020051  020040  024040
1760  011500  050123  020051  020040
1761  011506  052040  051505  021524
1762  011514  020040  040440  042104
1763  011522  042522  051523     000
1764          011530                  .EVEN
1765  011530  001116  001176  001174  DT5:    $ERRPC,$REG7,$REG6,$REG5,$GDADR
1766  011536  001172  001120
1767  011542  000000                  0
1768  011544  020101  047516  051440  EM6:    .ASCIZ  "A NO SACK TIMEOUT HAS OCCURED"
1769  011552  041501  020113  044524
1770  011560  042515  052517  020124
1771  011566  040510  020123  041517
1772  011574  052503  042522  000104
1773  011602  050050  024503  020040  DH6:    .ASCIZ  "(PC)    (PS)    (SP)    TEST#   (LKS)   "
1774  011610  020040  050050  024523
1775  011616  020040  020040  051450
1776  011624  024520  020040  020040
1777  011632  042524  052123  020043
1778  011640  020040  046050  051513
1779  011646  020051  020040     000
1780          011654                  .EVEN
1781  011654  001116  001176  001174  DT6:    $ERRPC,$REG7,$REG6,$REG5,LKS
1782  011662  001172  177546
1783  011666  000000                  0
1784  011670  051127  047117  020107  EM7:    .ASCIZ  "WRONG CONDITION CODES WERE PUT ONTO STACK BY INTERRUPT"
1785  011676  047503  042116  052111
1786  011704  047511  020116  047503
1787  011712  042504  020123  042527
1788  011720  042522  050040  052125
1789  011726  047440  052116  020117
1790  011734  052123  041501  020113
1791  011742  054502  044440  052116
```

```
011750  051105  052522  052120
011756     000
011757     050  041520  020051  DH7:    .ASCIZ  "(PC)    (PS)    (SP)    TEST#   CC WAS  CC S/B"
011764  020040  024040  051520
011772  020051  020040  024040
012000  050123  020051  020040
012006  052040  051505  021524
012014  020040  041440  020103
012022  040527  020123  041440
012030  020103  027523  000102
                                .EVEN
012036  001116  001176  001174  DT7:    $ERRPC,$REG7,$REG6,$REG5,BUF1,$GDDAT
012044  001172  000776  001124
012052  000000                  0
012054  051127  047117  020107  EM10:   .ASCIZ  "WRONG PC PUT ONTO THE STACK BY AN INTERRUPT"
012062  041520  050040  052125
012070  047440  052116  020117
012076  044124  020105  052123
012104  041501  020113  054502
012112  040440  020116  047111
012120  042524  051122  050125
012126  000124
012130  050050  024503  020040  DH10:   .ASCIZ  "(PC)    (PS)    (SP)    TEST#  @(SP)WAS @(SP)S/B  "
012136  020040  050050  024523
012144  020040  020040  051450
012152  024520  020040  020040
012160  042524  052123  020043
012166  040040  051450  024520
012174  040527  020123  024100
012202  050123  051451  041057
012210  020040     000
        012214                  .EVEN
012214  001116  001176  001174  DT10:   $ERRPC,$REG7,$REG6,$REG5,BUF2,$GDDAT
012222  001172  000774  001124
012230  000000                  0
012232  051124  042531  020104  EM11:   .ASCIZ  "TRYED TO ACCESS THE LKS REGISTER, AND TRAPPED"
012240  047524  040440  041503
012246  051505  020123  044124
012254  020105  045514  020123
012262  042522  044507  052123
012270  051105  020054  047101
012276  020104  051124  050101
012304  042520  000104
012310  050050  024503  020040  DH11:   .ASCIZ  "(PC)    (PS)    (SP)    TEST#"
012316  020040  050050  024523
012324  020040  020040  051450
012332  024520  020040  020040
012340  042524  052123  000043
                                .EVEN
012346  001116  001176  001174  DT11:   $ERRPC,$REG7,$REG6,$REG5
012354  001172
012356  000000                  0
        000001                  .END
```

```
A0004    002104            387#     390
A0005    002172            408#     411
A0006    002302            435#     438
A0007    002434            465#     468
A0010    002614            505#     508
A0011    002716            534#     537
A0012    003102            576#     579
A0013    003172            606#     609
A0014    003546            685      694#
A0015    003664            706      718#
A0016    004002            730      742#
A0017    004120            754      766#
A0020    004236            778      790#
A0021    004354            802      814#
A0022    004512            841      843#
A0023    004560            857#     860
A0024    004702            885#     886
A0025    005012            909#     912
A0026    005164            945#     946
A0027    005300            971#     972
A0030    005440            989      999#
A0031    005526           1019#    1022
A0032    005660           1056#    1059
A0033    006016           1089#    1092
BIT0    =  000001            121#
BIT00   =  000001            111#     121
BIT01   =  000002            110#     120
BIT02   =  000004            109#     119
BIT03   =  000010            108#     118
BIT04   =  000020            107#     117
BIT05   =  000040            106#     116
BIT06   =  000100            105#     115
BIT07   =  000200            104#     114
BIT08   =  000400            103#     113     1205
BIT09   =  001000            102#     112     1213     1548
BIT1    =  000002            120#
BIT10   =  002000            101#    1530
BIT11   =  004000            100#    1220
BIT12   =  010000             99#
BIT13   =  020000             98#    1537
BIT14   =  040000             97#    1191
BIT15   =  100000             96#
BIT2    =  000004            119#
BIT3    =  000010            118#
BIT4    =  000020            117#
BIT5    =  000040            116#
BIT6    =  000100            115#
BIT7    =  000200            114#
BIT8    =  000400            113#
BIT9    =  001000            112#
BPTVEC=  000014            128#
BUF1    =  000776            140#    1064*    1073     1097*    1803
BUF2    =  000774            139#    1065*    1098*    1106     1824
B0005    002210            409      414#
B0006    002320            436      441#
B0007    002452            466      471#
```

```
B0010   002632            506     511#
B0011   002734            535     540#
B0012   003120            577     582#
B0013   003210            607     612#
B0014   003560            695     697#
B0015   003676            719     721#
B0016   004014            743     745#
B0017   004132            767     769#
B0020   004250            791     793#
B0021   004366            815     817#
B0023   004576            858     863#
B0025   005030            910     915#
B0030   005456           1001    1003#    1004
B0031   005544           1020    1025#
B0032   005676           1057    1062#
B0033   006034           1090    1095#
C0007   002462            474#    476
C0010   002642            514#    516
C0011   003014            553#    556
C0013   003224            616#    619
C0025   005054            902     923#
C0031   005600           1034#   1037
C0032   005742           1066    1073#
C0033   006100           1099    1106#    1108
DDISP =  177570             43#    191
DH1     010535            243    1669#
DH10    012130            278    1814#
DH11    012310            283    1835#
DH2     010666            248    1687#
DH3     011047            253    1708#
DH4     011233            258    1730#
DH5     011457            263    1757#
DH6     011602            268    1773#
DH7     011757            273    1794#
DISPLA  001140            191#    327*    1234*    1529*
DISPRE  000174            149#    327
DSWR =  177570             42#    190
DT1     010616            244    1679#
DT10    012214            279    1824#
DT11    012346            284    1841#
DT2     010740            249    1695#
DT3     011120            254    1716#
DT4     011370            259    1747#
DT5     011530            264    1765#
DT6     011654            269    1781#
DT7     012036            274    1803#
D0007   002474            457     479#
D0010   002654            492     519#
D0011   003032            554     559#
D0013   003242            617     622#
D0031   005616           1035    1040#
EMTVEC=  000030            131#    309*    310*
EM1     010454            242    1660#
EM10    012054            277    1806#
EM11    012232            282    1827#
EM2     010634            247    1682#
```

```
EM3     010754              252    1698#
EM4     011134              257    1719#
EM5     011406              262    1750#
EM6     011544              267    1768#
EM7     011670              272    1784#
ERRVEC= 000004              124#   1196    1197*   1199*   1202*
E0001   001730              340     346#
E0002   001776              360#
E0003   002042              374#
E0004   002116              391#
E0005   002204              412#
E0006   002314              439#
E0007   002446              469#
E0010   002626              509#
E0011   002730              538#
E0012   003114              580#
E0013   003204              610#
E0014   003556              696#
E0015   003674              720#
E0016   004012              744#
E0017   004130              768#
E0020   004246              792#
E0021   004364              816#
E0022   004510              842#
E0023   004572              861#
E0024   004722              877     890#
E0025   005024              913#
E0026   005204              940     950#
E0027   005332              977#
E0030   005472              999    1007#
E0031   005540             1023#   1031
E0032   005672             1060#
E0033   006030             1093#
E10006  002342              446#
E10007  002470              477#
E10010  002650              517#
E10012  003134              587#
E10013  003236              620#
E10023  004614              867#
E10025  005050              921#
E10031  005612             1038#
E10033  006110             1108#
E1005   002226              418#
E1011   003026              550     557#
E20007  002502              481#
E20010  002662              521#
E20011  003040              563#
E20013  003334              643#
E20025  005064              925#
E20031  005624             1044#
E30013  003366              653#
E40013  003444              669#
F0013   003262              627#    632
GNS   = ###### U            148    1582    1583    1584    1585    1586
G0013   003276              628     631#
H0013   003322              639#    642
```

```
IOTVEC= 000020   129#   307*   308*
IOOOI   001722   344#
IOO14   003540   693#
IOO15   003656   717#
IOO16   003774   741#
IOO17   004112   765#
IOO20   004230   789#
IOO21   004346   813#
IOO22   004472   839#
IOO23   004576   864#
IOO26   005172   947#
JOO13   003340   640    645#
KSTART  001400   153    290#
KOO13   003354   649#   652
LKS    = 177546   137#   344*   358    372    385*   387    406*   408    415*   416    433*   435    442*
                  443*   444    463*   465    472*   479    494*   505    512*   519    532*   534    551*
                  553    574*   576    583*   585    603*   606    614*   616    623*   624    626*   627
                  630*   636*   639    647*   649    657*   658    660*   661    664*   692*   694    697*
                  716*   718    721*   740*   742    745*   764*   766    769*   788*   790    793*   812*
                  814    817*   838*   840    843*   855*   857    864*   865    878*   884*   888    903*
                  909    919*   923    942*   944*   948    964*   973*   974*   975    986*   995*   997*
                 1005*  1017*  1019   1032*  1034   1054*  1056   1067*  1087*  1089   1100*  1115*  1679
                 1695   1716   1781
LOO13   003372   650    655#
MOO13   003420   661#   666
NOP    = 000240   138#
NOO13   003434   662    665#
PIRQ   = 177772    41#
PIRQVE= 000240   135#
POWMES  010240  1602   1631#
POWPOP  010452  1601   1659#
POWPUS  010450  1600   1658#
PRO    = 000000    58#
PR1    = 000040    59#
PR2    = 000100    60#
PR3    = 000140    61#
PR4    = 000200    62#
PR5    = 000240    63#   497
PR6    = 000300    64#   544
PR7    = 000340    65#   402    429    598    687    711    735    759    783    807    833    880    904
                  936   965   1027
PS     = 177776    38#    39    497*   544*
PSW    = 177776    39#
PWRVEC= 000024   130#   313*   314*   1592*  1593*  1605*  1620*  1621*
RESVEC= 000010   125#
ROOOI   001716   342    343#
ROOO2   001756   355    356#
ROOO3   002032   370    371#
ROOO4   002070   383    384#
ROOO5   002164   401    406#
ROOO6   002274   428    433#
ROOO7   002412   456    459#
ROO10   002544   491    494#
ROO11   002710   531    532#
ROO12   003074   573    574#
ROO13   003140   596#   656
```

```
R0014    003530            683     691#
R0015    003646            709     715#
R0016    003764            733     739#
R0017    004102            757     763#
R0020    004220            781     787#
R0021    004336            805     811#
R0022    004462            831     837#
R0023    004552            854     855#
R0024    004662            876     879#
R0025    004772            900     903#
R0026    005146            935     942#
R0027    005252            961     964#
R0030    005372            988     989#
R0031    005520           1016    1017#
R0032    005652           1053    1054#
R0033    006010           1086    1087#
R1013    003156            602#
R2013    003210            613#
R3013    003306            635#
R4013    003340            646#
STACK  = 001100             35#
START    001450            301#    1173
STKLMT= 177774             40#
STMES    010306            300    1638#
SWR      001136            190#     324     326*   1191    1205    1207    1213    1220    1530    1537    1542    1548
SWREG    000176            150#     326
SW0    = 000001             93#
SW00   = 000001             83#      93
SW01   = 000002             82#      92
SW02   = 000004             81#      91
SW03   = 000010             80#      90
SW04   = 000020             79#      89
SW05   = 000040             78#      88
SW06   = 000100             77#      87
SW07   = 000200             76#      86
SW08   = 000400             75#      85
SW09   = 001000             74#      84
SW1    = 000002             92#
SW10   = 002000             73#
SW11   = 004000             72#
SW12   = 010000             71#
SW13   = 020000             70#
SW14   = 040000             69#
SW15   = 100000             68#
SW2    = 000004             91#
SW3    = 000010             90#
SW4    = 000020             89#
SW5    = 000040             88#
SW6    = 000100             87#
SW7    = 000200             86#
SW8    = 000400             85#
SW9    = 001000             84#
TBIYVE= 000014            126#
TCRLF  = 000200           1270    1309#
THT    = 000011           1268    1308#
TKVEC  = 000060            133#
```

```
TPVEC = 000064   134#
TRAPVE= 000034   132#  311*  312*
TRAPO   006252   353   367   381   398   425   453   489   529   571   828   851   873   897
                 932   958   984  1014  1051  1084  1158#
TRPMES  010363  1160  1647#
TRPM2S  010424  1167  1653#
TRTVEC= 000014   127#
TYPDS = 104404  1141  1586#
TYPE  = 104400  1139  1142  1164  1273  1370  1440  1466  1483  1485  1488  1490  1494  1501
                1532  1540  1582#  1622
TYPOC = 104401  1163  1474  1498  1583#
TYPON = 104403  1585#
TYPOS = 104402  1584#
T0001   001672   332   339#  1153  1603
T0002   001732   345   352#
T0003   002000   359   366#
T0004   002044   373   380#
T0005   002120   388   397#
T0006   002230   413   417   424#
T0007   002344   440   445   452#
T0010   002504   470   478   480   488#
T0011   002664   501   510   518   520   528#   558
T0012   003042   539   548   562   569#
T0013   003136   581   586   595#
T0014   003460   611   621   644   654   668   681#
T0015   003576   698   705#
T0016   003714   722   729#
T0017   004032   746   753#
T0020   004150   770   777#
T0021   004266   794   801#
T0022   004404   818   826#
T0023   004520   827   850#
T0024   004616   862   866   872#
T0025   00472?   889   896#
T0026   005026   914   922   924   931#
T0027   005206   949   957#
T0030   005?34   963   976   983#
T0031   005?74  1006  1013#
T0032   00?626  1024  1039  1043  1050#  1077
T0033   0?5764  1061  1074  10?3#
T0034   006120  1094  1107  1114#
WORD    001234   225#   970*   971*   996*  1003*  1658  1659
$BDADR  001122   184#
$BDDAT  001?36   186#   304
$BELL   001224   220#  1532  1556
$CHARC  007006  1284*  1291  1300*  1305#
$CMTAG  001100   172#   302   309   315   316   317
$CM1  = 000010   202#   203#   204#   205#   206#   207#   208#   209#   210#
$CM2  = 000020   202#   203#   204#   205#   206#   207#   208#   209#   210#
$CM3  = 000010   200#   202
$CM4  = 000010   210#   211#   212#   213#   214#   215#   216#   217#   218#
$CRLF   001231   222#  1164  1274  1311  1466  1485  1490  1494  1540  1556
$DBLK   007226  1336  1370  1378#
$DOAGN  006226  1135  1146  1152#
$DTBL   007216  1339  1374#
$ENDAD  006216   160  1148#  1545
```

```
$ENDCT  006164            315  1137#
$ENDMG  006232           1139  1154#
$ENULL  006247           1142  1157#
$EOP    006130           1127#
$EOPCT  006156            315*  1134#  1138
$ERFLG  001103            175#  1180   1209   1211   1217*  1238   1527*  1556
$ERMAX  001115            181#   318*  1211   1233*  1238
$ERROR  007620            309  1520#
$ERRPC  001116            182#  1472   1534*  1535*  1536   1556   1679   1695   1716   1747   1765   1781   1803
                         1824  1841
$ERRTB  001236            241#  1480
$ERRTY  007464           1465#  1539
$ERTTL  001112            179#  1533*  1556
$ESCAP  001222            219#   317*  1232*  1551   1553   1556
$FILLC  001154            198#  1277   1311
$FILLS  001153            197#  1311
$GDADR  001120            183#   684*   693*   699*   710*   717*   723*   734*   741*   747*   758*   765*   771*
                          782*   789*   795*   806*   813*   819*   832*   839*  1765
$GDDAT  001124            185#   357*   369*   384*   400*   427*   455*   570*   682*   708*   732*   756*   780*
                          804*   830*   853*   865    875*   888    899*   934*   948    960*   975    987*  1075*
                         1108*  1679   1803   1824
$GET42  006206           1143#
$HD   = 000000             18
$ICNT   001104            176#  1224*  1225   1227*  1237
$ILLUP  010222           1592  1625#
$ITEMB  001114            180#  1469   1536*  1556
$LF     001232            223#  1311   1556
$LPADR  001106            177#   319*   342*   355*   370*   383*   401*   428*   456*   491*   531*   573*   602*
                          613#   635*   646*   656*   683*   709*   733*   757*   781*   805*   831*   854*   876*
                          900#   935*   961*   988*  1016*  1053*  1086*  1215*  1230*  1235   1237
$LPERR  001110            178#   320*  1215   1231*  1237   1550
$MAIL = ******  U         331   1230   1262   1542
$MXCNT  006574           1228  1237#
$NULL   001152            196#  1279   1311
$OCNT   007460           1412*  1441*  1454#
$OMODE  007462           1407*  1411*  1416   1419*  1430*  1456#
$OVER   006560           1192   1208   1216   1226   1234#
$PASS   001100            173#  1131*  1132*  1140   1154   1222   1238
$POWER  010230           1623   1628#
$PWRDN  010060            313   1592#  1620
$PWRMG  010216           1623#
$PWRUP  010146           1605   1610#
$QUES   001230            221#  1311   1556
$RDCHR= ******  U        1587
$RDDEC= ******  U        1587
$RDLIN= ******  U        1587
$RDOCT= ******  U        1587
$REGAD  001156            200#
$REG0   001160            202#  1526#
$REG1   001162            203#   669*  1747
$REG2   001164            204#   670*  1747
$REG3   001166            205#
$REG4   001170            206#
$REG5   001172            207#  1524*  1525*  1679   1695   1716   1747   1765   1781   1803   1824   1841
$REG6   001174            208#  1521*  1522*  1679   1695   1716   1747   1765   1781   1803   1824   1841
$REG7   001176            209#  1523*  1679   1695   1716   1747   1765   1781   1803   1824   1841
```

```
$R2A   = ****** U          1587
$SAVRE= ****** U          1587
$SAVR6   010226           1604*   1610    1611*   1612*   1627*
$SCOPE   006324            307    1189*
$SETUP= 000037               8*    307     309     311     313     315     316     317     319    1129    1545
$STUP  = 177777              8*
$SVLAD   006532           1200    1229*
$SVPC  = 000204            158*    163
$SWR   = 167400              5*     18      23      24      25      26      27      28      29     218     219     220     316
                           317     319     320    1124    1130    1144    1153    1154    1181    1182    1183    1184    1185
                          1191    1203    1205    1206    1209    1210    1211    1218    1219    1220    1231    1234    1237
                          1512    1513    1514    1515    1516    1530    1537    1542    1548    1556    1624
$SWRMK= 000000              29      30    1185    1186    1207
$TIMES   001220           218*    316*   1130*   1218*   1225    1228*   1237
$TKB     001144           193*
$TKS     001142           192*
$TMP0    001200           210*
$TMP1    001202           211*
$TMP2    001204           212*
$TMP3    001206           213*
$TMP4    001210           214*
$TMP5    001212           215*
$TMP6    001214           216*
$TMP7    001216           217*
$TN    = 000000              6*     18
$TPB     001150           195*   1297*   1311
$TPFLG   001155           199*   1256    1311
$TPS     001146           194*   1295    1311
$TRAP    010024            311   1565*
$TRP   = 000005          1573*   1583*   1584*   1585*   1586*   1587*
$TRPAD   010046          1570    1581*
$TSTNM   001102           174*   1129*   1180    1207    1229*   1234    1238    1525    1529    1556
$TYPBN= ****** U          1587
$TYPDS   007012          1324*   1586
$TYPE    006576            299   1159    1166    1256*   1573    1582
$TYPEC   006742          1276    1283    1290    1295*   1296
$TYPEX   007010          1301    1303    1306*
$TYPOC   007262          1410*   1583
$TYPON   007276          1409    1412*   1585
$TYPOS   007236          1405*   1584
$XTSTR   006336          1194*
$OFILL   007461          1406*   1410*   1420    1455*
$40CAT= ****** U          1191    1539
.      = 012360           144*    148*    158     159*    161*    163*    171*    224     289*    305     319     320    1154
                          1158    1237    1238    1311    1378*   1504*   1556    1607    1626    1657*   1694*   1746*   1764*
                          1780*   1823*
```

```
ADITAG    164#    225
COMMEN      1#    136#
ENDCOM      1#    136#
ERROR      36#    346     360     374     391     412     418     439     446     469     477     481     509     517     521
          538     557     563     580     587     610     620     643     653     671     696     720     744     768     792
          816     842     861     867     890     913     921     925     950     977    1007    1023    1038    1044    1060
         1076    1093    1109
ESCAPE      1#    136#
GETPRI      1#      8#    291
MULT        1#    136#
NEWTST      1#    136#
POP         1#    136#   1365    1614
PUSH        1#    136#   1324    1594    1600
REPORT      1#
SAVE     1505#   1521
SCOPE      37#    339     352     366     380     397     424     452     488     528     569     595     681     705     729
          753     777     801     826     850     872     896     931     957     983    1013    1050    1083    1114    1128
SETPRI      1#      8#    402     429     598     687     711     735     759     783     807     833     879     904     936
          965    1027
SETRAP      8#    353     367     381     398     425     453     489     529     571     828     851     873     897     932
          958     984    1014    1051    1084
SETTRA   1573#   1583    1584    1585    1586
SETUP       1#    136#    301
SKIP        1#    136#
SLASH       1#    136#
SPACE     136#
STARS       1#      8#    136#    154     164     224    1118    1174    1238    1311    1379    1457    1505    1556    1587
SWRSU       1#    321#
TRMTRP   1573#
TYPBIN      1#    136#
TYPDEC      1#    136#   1140
TYPNAM      1#
TYPNUM      1#    136#
TYPOCS      1#    136#
TYPOCT      1#    136#   1472    1496
TYPTXT      1#    136#
WAITLK      8#    407     434     464     504     533     552     575     605     615     638     648     856     908    1018
         1033    1055    1088
$$CMRE    164#    202     203     204     205     206     207     208     209
$$CMTM    164#    210     211     212     213     214     215     216     217
$$ESCA      1#    136#
$$NEWT      1#    136#
$$SET    1573#   1583    1584    1585    1586
$$SKIP      1#    136#
.EQUAT      1#      8#     31
.HEADE      1#      8#
.KT11       1#
.SETUP      1#      8#
.SWRHI      1#      8#     18
.SWRLO      8#     30#
.$ACT1      1#      8#    154
.$APTB      1#
.$APTH      1#
.$APTY      1#
.$ASTA      1#
.$CATC      1#      8#    141
```

```
.$CMTA      1#      8#    164
.$DB2D      1#
.$DB2O      1#
.$DIV       1#
.$EOP       1#      8#   1118
.$ERRO      1#      8#   1505
.$ERRT      1#      8#   1457
.$MULT      1#
.$POWE      1#      8#   1587
.$RAND      1#
.$RDDE      1#
.$RDOC      1#
.$READ      1#
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#      8#   1174
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#      8#   1556
.$TYPB      1#
.$TYPD      1#      8#   1311
.$TYPE      1#      8#   1238
.$TYPO      1#      8#   1379
.$4OCA      1#

. ABS.   012360     000

 ERRORS DETECTED:  0

DZKWAF.BIN,DZKWAF.LST/CRF/SOL/NL:TOC=DZKWAF.SML,DZKWAF.P11
RUN-TIME:  8 10  .8 SECONDS
RUN-TIME RATIO:  88/20=4.3
CORE USED:  28K  (55 PAGES)
```

K04