```
                        .TITLE  MAINDEC-11-DCKBR-B
                        ;COPYRIGHT 1973 DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
                        ;PROGRAM BY BRUCE BURGESS


                        ;OPERATIONAL SWITCH SETTINGS
                        ;
                        ;       SWITCH                  USE
                        ;       ******          ********************
                        ;         15            HALT ON ERROR
                        ;         14            LOOP ON TEST
                        ;         13            INHIBIT ERROR TYPEOUTS
                        ;         11            INHIBIT ITERATIONS
                        ;         10            0 = BELL ON PASS COMPLETE
                        ;                       1 = BELL ON ERROR
                        ;          9            LOOP ON ERROR
                        ;          0            LOOP ON TEST IN SW<7:6>
                                ;SPECIAL USER TYPE SWITCH SW<12>
                                ;
                                ;IF SET INDICATES USER INPUT
                                ;IF CLEAR INDICATES PROGRAM FIND
                                        ;SPECIAL BIT11 DISABLE SWITCH SW<09>
                                        ;
                                        ;IF SET INDICATES DON'T USE IF PRESENT
                                        ;IF CLEAR INDICATES ALLOW USE IF PRESENT

                        ;BASIC DEFINITIONS
                        ;*****************
                        ;INITIAL ADDRESS OF THE STACK POINTER
001100                  STACK=  1100
                        ;*****************
                        .EQUIV  EMT,HLT                         ;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE                       ;BASIC DEFINITION OF SCOPE CALL
177776                  PS=     177776                          ;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
177570                  SWR=    177570                          ;SWITCH REGISTER
177570                  DISPLAY=SWR

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

                        ;SWITCH DEFINITION
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

                        ;MISCELLANEOUS BIT ASSIGNMENT
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
```

```
                                ;VECTOR ADDRESSES
000004                          ERRVEC= 4
000010                          RESVEC= 10
000014                          TTITVEC=14
000014                          TKTVEC= 14
000014                          BPTVEC= 14
000020                          IOTVEC= 20
000024                          PWRVEC= 24
000030                          EMTVEC= 30
000034                          TRAPVEC=34
000001                                  N=1                             ;DEFINE STARTING 'N' FOR, SCOPE
                                                                        ;ROUTINE WHICH PRINTS TEST NO.
                                                                        ;AND APPROPRIATE COMMENT
000000                                  .=0
                                ;TRAP CATCHER IN UNUSED LOCATIONS FROM 0 - 776
                                ;LOCATION 0 WILL CATCH IMPROPERLY LOADED VECTORS
000200                                  .=200
000200  000137  001706                  JMP     @#BEGIN         ;JUMP TO STARTING ADDRESS OF PROGRAM
000046                                  .=46
000046  012042                          $ENDAD
000052                                  .=52
000052  040000                          BIT14
000204                                  .=204
                                ;KT11-D STATUS REGISTER ADDRESSES
000204  177572                  SR0:    177572
000206  177576                  SR2:    177576
                                ;KERNAL PAGE DESCRIPTOR REGISTERS
000210  172300                  KPDR0:  172300
000212  172302                  KPDR1:  172302
000214  172304                  KPDR2:  172304
000216  172306                  KPDR3:  172306
000220  172310                  KPDR4:  172310
000222  172312                  KPDR5:  172312
000224  172314                  KPDR6:  172314
000226  172316                  KPDR7:  172316
                                ;KERNAL PAGE ADDRESS REGISTERS
000230  172340                  KPAR0:  172340
000232  172342                  KPAR1:  172342
000234  172344                  KPAR2:  172344
000236  172346                  KPAR3:  172346
000240  172350                  KPAR4:  172350
000242  172352                  KPAR5:  172352
```



```
000244  172354                  KPAR6:  172354
000246  172356                  KPAR7:  172356
                                ;KT11 VECTOR ADDRESS
000250  000250  000252          $EGVEC: 250,252
```

```
                                ;*****************************************************************
001100                                  .=1100

                                ;ROUTINE TO TYPE ASCII 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.

                                ;CALLS:
                                ;1) USING A TRAP INSTRUCTION
                                ;       TYPE    ,MESADR                 ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                ;OR
                                ;       TYPE
                                ;       MESADR

                                ;2) USING A JSR INSTRUCTION
                                ;       MOV     PS,-(SP)                ;PUSH PROCESSOR STATUS WORD ON THE STACK
                                ;       JSR     PC,$TYPE                ;CALL TYPE ROUTINE
                                ;       MESADDR                         ;FIRST ADRESS OF MESSAGE

001100  177564                 $TPS:   177564                          ;TTY PRINTER STATUS REG. ADDRESS
001102  177566                 $TPB:   177566                          ;TTY PRINTER BUFFER REG. ADDRESS
001104     000                  $NULL:  .BYTE   0                       ;CONTAINS NULL CHARACTER FOR FILLS
001105     002                  $FILLS: .BYTE   2                       ;CONTAINS # OF FILLER CHARACTERS REQUIRED
001106     000                  $TPFLG: .BYTE   0                       ;"TERMINAL AVAILABLE" FLAG (0=YES)
001107     000                          .BYTE   0                       ;RESERVED

001110  105767  177772         $TYPE:  TSTB    $TPFLG                  ;IS THERE A TERMINAL?
001114  001402                         BEQ     6$                      ;BR IF YES
001116  000000                         HALT                            ;HALT HERE IF NO TERMINAL
001120  000407                         BR      7$                      ;LEAVE
001122  010046                 6$:     MOV     R0,-(SP)                ;SAVE R0
001124  017600  000002                 MOV     @2(SP),R0               ;GET ADDRESS OF ASCIZ STRING
001130  112046                 1$:     MOVB    (R0)+,-(SP)             ;PUSH CHARACTER TO BE TYPED ONTO STACK
001132  001005                         BNE     2$                      ;BR IF IT ISN'T THE TERMINATOR
001134  005726                         TST     (SP)+                   ;IF TERMINATOR POP IT OFF THE STACK
001136  012600                         MOV     (SP)+,R0                ;RESTORE R0
001140  062716  000002         7$:     ADD     #2,(SP)                 ;ADJUST RETURN PC
001144  000002                         RTI                             ;RETURN
001146  004767  000026         2$:     JSR     PC,5$                   ;GO TYPE THIS CHARACTER
001152  122726  000012                 CMPB    #12,(SP)+               ;CHECK IF THE CHAR. TYPED WAS A  LINE FEED
001156  001364                         BNE     1$                      ;GO GET NEXT CHAR. IF NOT LINE FEED
001160  016746  177720                 MOV     $NULL,-(SP)             ;GET # OF FILLER CHARS. NEEDED
                                                                        ;AND THE NULL CHAR.
001164  105366  000001         4$:     DECB    1(SP)                   ;DOES A NULL NEED TO BE TYPED?
001170  002770                         BLT     3$                      ;BR IF NO--GO POP THE NULL OFF OF STACK
001172  004767  000002                 JSR     PC,5$                   ;GO TYPE A NULL
001176  000772                         BR      4$                      ;LOOP
001200  105777  177674         5$:     TSTB    @$TPS                   ;WAIT UNTIL PRINTER IS READY
001204  100375                         BPL     5$
001206  116677  000002  177666         MOVB    2(SP),@$TPB             ;LOAD CHAR TO BE TYPED INTO DATA REG.
001214  000207                 3$:     RTS     PC
001216  000062                         .BLKB   50                      ;RESERVE SOME MORE CORE FOR OVERLAY CAPIGILITIES
```



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

                                ;COMMON TAGS

001300                                  .=1300

001300  000000                 $PASS:  .WORD   0                       ;CONTAINS PASS COUNT
001302  000000                 $ISTNM: .WORD   0                       ;CONTAINS THE TEST NUMBER
001304  000000                 $ICNT:  .WORD   0                       ;CONTAINS SUBTEST ITERATION COUNT
001306  000000                 $LPADR: .WORD   0                       ;CONTAINS SCOPE LOOP ADDRESS
001310  000000                 $LPERR: .WORD   0                       ;CONTAINS SCOPE RETURN FOR ERRORS
001312  000000                 $ERTTL: .WORD   0                       ;CONTAINS TOTAL ERRORS DETECTED
001314     000                 $ERFLG: .BYTE   0                       ;CONTAINS ERROR FLAG
001315     000                         .BYTE   0                       ;RESERVED--NOT TO BE USED
001316  000000  000000                 .WORD   0,0                     ;RESERVED--NOT TO BE USED
001322     000                 $ITEMB: .BYTE   0                       ;CONTAINS ITEM CONTROL BYTE
001323     000                         .BYTE   0                       ;RESERVED--NOT TO BE USED
001324  000000                 $MLTAD: .WORD   0                       ;CONTAINS PC OF LAST HLT INSTRUCTION
001326  000000                 $GDADR: .WORD   0                       ;CONTAINS ADDRESS OF 'GOOD' DATA
001330  000000                 $BDADR: .WORD   0                       ;CONTAINS ADDRESS OF 'BAD' DATA
001332  000000                 $GDDAT: .WORD   0                       ;CONTAINS 'GOOD' DATA
001334  000000                 $BDDAT: .WORD   0                       ;CONTAINS 'BAD' DATA
001336  000000                 $REGAD: .WORD   0                       ;CONTAINS THE ADDRESS FROM
                                                                       ;WHICH  (@REG0) WAS OBTAINED
001340  000000                 $REG0:  .WORD   0                       ;CONTAINS ((SREGAD)+0)
001342  000000                 $REG1:  .WORD   0                       ;CONTAINS ((SREGAD)+2)
001344  000000                 $REG2:  .WORD   0                       ;CONTAINS ((SREGAD)+4)
001346  000000                 $REG3:  .WORD   0                       ;CONTAINS ((SREGAD)+6)
001350  000000                 $REG4:  .WORD   0                       ;CONTAINS ((SREGAD)+10)
001352  000000                 $REG5:  .WORD   0                       ;CONTAINS ((SREGAD)+12)
001354  000000                 $REG6:  .WORD   0                       ;CONTAINS ((SREGAD)+14)
001356  000000                 $REG7:  .WORD   0                       ;CONTAINS ((SREGAD)+16)
001360  000000                 $REG10: .WORD   0                       ;CONTAINS ((SREGAD)+20)
001362  000000                 $REG11: .WORD   0                       ;CONTAINS ((SREGAD)+22)
001364  000000                 $REG12: .WORD   0                       ;CONTAINS ((SREGAD)+24)
001366  000000                 $TMP0:  .WORD   0                       ;USER DEFINED
001370  000000                 $TMP1:  .WORD   0                       ;USER DEFINED
001372  000000                 $TMP2:  .WORD   0                       ;USER DEFINED
001374  000000                 $TMP3:  .WORD   0                       ;USER DEFINED
001376  000000                 $TMP4:  .WORD   0                       ;USER DEFINED
001400  000000                 $TMP5:  .WORD   0                       ;USER DEFINED
001402  000000                 $TMP6:  .WORD   0                       ;USER DEFINED
001404  000000                 $TMP7:  .WORD   0                       ;USER DEFINED
001406  000000                 $TMP10: .WORD   0                       ;USER DEFINED
001410  000000                 $TMP11: .WORD   0                       ;USER DEFINED
001412  000000                 $TMP12: .WORD   0                       ;USER DEFINED
                                ;THE FOLLOWING TAG(S) ARE USER DEFINED
001414  000000                 $TMPAD: .WORD   0
001416  000000                 $SETAD: .WORD   0
001420  000000                 $SET0:  .WORD   0                       ;THESE LOCATIONS CONTAIN THE
001422  000000                 $SET1:  .WORD   0                       ;APPROPRIATE OFFSET VALUES
001424  000000                 $SET2:  .WORD   0                       ;FOR THE PARITY CONTROL
001426  000000                 $SET3:  .WORD   0                       ;REGISTERS WHEN MEMORY
001430  000000                 $SET4:  .WORD   0                       ;MANAGEMENT IS ENABLED
```

```
001432  000000          $$ET5:  .WORD   0                       ;DURING PROGRAM EXECUTION
001434  000000          $$ET6:  .WORD   0
001436  000000          $$ET7:  .WORD   0
001440  000000          $$ET10: .WORD   0
001442  000000          $$ET11: .WORD   0
001444  000000          $$ET12: .WORD   0
001446  000000          NIERAD: .WORD   0
001450  000000          NIER0:  .WORD   0                       ;THESE LOCATIONS CONTAIN THE
001452  000000          NIER1:  .WORD   0                       ;APPROPRIATE INTERLEAVE FACTORS
001454  000000          NIER2:  .WORD   0                       ;FOR THE PARITY CONTROL
001456  000000          NIER3:  .WORD   0                       ;REGISTERS (IF ANY)
001460  000000          NIER4:  .WORD   0
001462  000000          NIER5:  .WORD   0
001464  000000          NIER6:  .WORD   0
001466  000000          NIER7:  .WORD   0
001470  000000          NIER10: .WORD   0
001472  000000          NIER11: .WORD   0
001474  000000          NIER12: .WORD   0
001476  000000          NEWSTK: .WORD   0
                        ;END OF USER DEFINED TAG(S)
```



```
                        ;THE FOLLOWING TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                        ;THIS INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                        ;LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                        ;NOTE1: IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($HLTAD).
                        ;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


001500                  $ERRTB:
                        ;NOTE:  ALL NUMBERS ARE TYPED AS 6-DIGIT OCTAL NUMBERS

                        ;ITEM 1
001500  013152                  EM1                     ;TEST DIDN'T ABORT
001502  015003                  DH5                     ;PROGRAM PC
                                                        ;REGISTER UNDER TEST
                                                        ;EXPECTED ABORT PC
001504  015142                  DT5                     ;$HLTAD,PARITY,$GDDAT
001506  000000                  0
                        ;ITEM 2
001510  013776                  EM2                     ;FATAL ERROR TO PROGRAM
001512  014470                  DH1                     ;PROGRAM PC
                                                        ;REGISTER UNDER TEST
001514  015106                  DT1                     ;$HLTAD, PARITY
001516  000000                  0
                        ;ITEM 3
001520  014027                  EM3                     ;ABORTED INCORRECTLY
001522  014604                  DH4                     ;PROGRAM PC
                                                        ;REGISTER UNDER TEST
                                                        ;EXPECTED BITS 5 THRU 11
                                                        ;ACTUAL BITS 5 THRU 11
                                                        ;EXPECTED ABORT PC
                                                        ;ACTUAL ABORT PC
001524  015124                  DT4                     ;$HLTAD,PARITY,$GDADR,$BDADR,$GDDAT,$BDDAT
001526  000000                  0
                        ;ITEM 4
001530  014055                  EM4                     ;NO PARITY MEMORY FOUND BELOW 28K
001532  014540                  DH2                     ;REGISTER UNDER TEST
001534  015114                  DT2                     ;PARITY
001536  000000                  0
                        ;ITEM 5
001540  014120                  EM5                     ;RESET DOESN'T WORK
001542  014470                  DH1                     ;PROGRAM PC
                                                        ;REGISTER UNDER TEST
001544  015106                  DT1                     ;$HLTAD, PARITY
001546  000000                  0
                        ;ITEM 6
001550  014145                  EM6                     ;USER SELECTED REGISTER NOT PRESENT
001552  014566                  DH3                     ;PROGRAM PC
001554  015120                  DT3                     ;$HLTAD
001556  000000                  0
```

```
                                        ;ITEM 7
        001560  014212                          EM7             ;NO PARITY MEMORY FOUND AT ALL
        001562  014540                          DH2             ;REGISTER UNDER TEST
        001564  015114                          DT2             ;PARITY
        001566  000000                          0
                                        ;ITEM 10
        001570  014252                          EM10            ;DIDN'T ABORT OR RECOGNIZE
                                                                ;STACK VIOLATION
        001572  015005                          DH5             ;PROGRAM PC
                                                                ;REGISTER UNDER TEST
                                                                ;EXPECTED ABORT PC
        001574  015142                          DT5             ;SHLTAD, PARITY, SGDDAT
        001576  000000                          0
                                        ;ITEM 11
        001600  014326                          EM11            ;ABORTED BUT STACK VIOLATION
                                                                ;NOT RECOGNIZED
        001602  014470                          DH1             ;PROGRAM PC
                                                                ;REGISTER UNDER TEST
        001604  015106                          DT1             ;SHLTAD, PARITY
        001606  000000                          0
                                        ;ITEM 12
        001610  014403                          EM12            ;STACK VIOLATION PICKED UP BUT
                                                                ;ABORT NOT RECOGNIZED
        001612  014470                          DH1             ;PROGRAM PC
                                                                ;REGISTER UNDER TEST
        001614  015106                          DT1             ;SHLTAD, PARITY
        001616  000000                          0
```



```
;*****************************************************************************
;
;SYSTEM PARITY REGISTER NOTES FOR MF11 AND MS11
;
;*****************************************************************************

;BIT ASSIGNMENTS FOR THE MF11 PARITY REGISTER IS AS FOLLOWS:
;
;BIT15          PARITY ERROR
;BITS 11-5      ERROR ADDRESS           ;HIGH ORDER ADDRESS BITS
                                        ;OF ADDRESS OF MOST RECENT ERROR
                                        ;(BITS 17 THRU 11)
;BIT02          WRITE                   ;NORMAL PARITY (ODD) WHEN CLEAR
                                        ;OTHER PARITY (EVEN) WHEN SET
;BIT00          ERROR ACTION ENABLE     ;NO ACTION WHEN CLEAR
;                                       ;TRAP TO VECTOR 114 WHEN SET
;
;NOTE: THE ABOVE BITS ARE READ/WRITE AND CAN BE CLEARED BY 'INIT' (EXCEPT BITS 11-5)
;
;\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/
;
;BIT ASSIGNMENTS FOR THE MS11 PARITY REGISTER IS AS FOLLOWS:
;
;BIT15          PARITY ERROR
;BIT02          WRITE                   ;NORMAL PARITY (EVEN) WHEN CLEAR
                                        ;OTHER PARITY (ODD) WHEN SET
;BIT00          ERROR ACTION ENABLE     ;NO ACTION WHEN CLEAR
                                        ;TRAP TO VECTOR 114 WHEN SET
;
;NOTE:  THERE ARE NO ERROR ADDRESS BITS IN THE CURRENT MS11 PARITY REGISTER
;       HOWEVER, THERE WILL BE IN A LATER VERSION WHICH WILL BE
;       HANDLED PROPERLY BY THIS PROGRAM

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

;SPECIAL NOTE----THERE ARE 2 GENERAL PURPOSE REGISTERS USED IN THE
;                PROGRAM FOR SPECIFIC CIRCUMSTANCES.  THEY ARE:
;
;                       R1 - WILL ALWAYS CONTAIN THE 1ST ADDRESS OF THE 2
;                            LOCATION MAP USED FOR TESTING.
;                            THE CONTENTS OF R1 IS DETERMINED BY THE 'COMPUT'
;                            ROUTINE SHOWN FURTHER DOWN.
;                            EXAMINATION OF R1 WILL TELL YOU WHERE IN PARITY
;                            MEMORY TESTING IS BEING CONDUCTED.
;
;                        R5 - WILL ALWAYS CONTAIN THE ADDRESS OF THE ROUTINE
;                             FOR SETTING UP THE PARITY VECTOR SERVICE ADDRESS.
;
;*****************************************************************************
;*****************************************************************************
```

```
                                ;**********************************************************
                                ;
                                ;MISCELLANEOUS COMMON PARITY VARIABLES AND FLAGS
                                ;
                                ;**********************************************************
001620  000114                INTVEC: 114                     ;PARITY INTERRUPT VECTOR ADDRESS
001622  000000                PARITY: 0                       ;CONTAINS PARITY REGISTER IN USE
001624  000000                PUPCORZONES: 0                  ;FLAG TO INDICATE TO 'CHECKLOC'
                                ;ROUTINE THAT A PC OR PC FETCH
                                ;OR A ZONE ABORT WAS DONE
                                ;0 = NO;  1 = YES
001626  000000                MEMRESFLAG: 0                   ;INDICATES PARITY TYPE
                                ;0 = CORE ; 1 = MOS
001630  000000                USERTYPE: 0                     ;INDICATES USER SELECTION OF
                                ;PARITY REGISTER
                                ;0 = PROGRAM FIND
                                ;1 = USER SELECTION
001632  000000                BLKCNT: 0                       ;CONTAINS THE NUMBER OF CONSEC-
                                ;UTIVE LOCATIONS TO BE TESTED
                                ;DURING PROGRAM TABULATION TO
                                ;COVER CASES OF MEMORY INTER-
                                ;LEAVING
001634  000000                RESTOREBASE: 0                  ;HOLDS PAGE 1 ADDRESS
                                ;OR CURRENT MEMORY ADDRESS FOR
                                ;RESTORATION DURING
                                ;RUNNING OF PROGRAM.  IT IS USED IF
                                ;WE HAVE CHECKED CONSECUTIVE
                                ;LOCATIONS WITHOUT AN ABORT BEFORE
                                ;GOING TO NEXT OFFSET WHICH WILL
                                ;PUT US IN ANOTHER BANK
001636  000000                LEAFCNT: 0                      ;CONTAINS THE NO. OF ABORTS
                                ;ENCOUNTERED IN DETERMINATION OF
                                ;AN INTERLEAVE FACTOR
001640  000000                MEMAD: 0                        ;CONTAINS A BASE ADDRESS OR A
                                ;CURRENT MEMORY ADDRESS USED IN
001642  000000                CPU40: 0                        ;FLAG TO INDICATE PROCESSOR
                                ;0 = 11/05; 1 = 11/40
                                ;PARITY TABLE CREATION

                                ;THE FOLLOWING TABLE IS USED TO DETERMINE THE
                                ;INTERLEAVE FACTOR FOR THE CONTROL REGISTERS
                                INVERTABLE:
001644
001644  000003  000001        3.,1                            ;3 ABORTS ON 3 CONSECUTIVE LOCS. = 1 WAY LEAVE
001650  000006  000002        6.,2                            ;3 ABORTS ON 6 CONSECUTIVE LOCS. = 2 WAY LEAVE
001654  000011  000003        9.,3                            ;3 ABORTS ON 9 CONSECUTIVE LOCS. = 3 WAY LEAVE
001660  000014  000004        12.,4                           ;3 ABORTS ON 12 CONSECUTIVE LOCS. = 4 WAY LEAVE
001664  000017  000005        15.,5                           ;3 ABORTS ON 15 CONSECUTIVE LOCS. = 5 WAY LEAVE
001670  000022  000006        18.,6                           ;3 ABORTS ON 18 CONSECUTIVE LOCS. = 6 WAY LEAVE
001674  000025  000007        21.,7                           ;3 ABORTS ON 21 CONSECUTIVE LOCS. = 7 WAY LEAVE
001700  000030  000010        24.,8.                          ;3 ABORTS ON 24 CONSECUTIVE LOCS. = 8 WAY LEAVE
001704  000000                0                               ;END OF TABLE TERMINATOR
```



```
001706                          BEGIN:
001706  012706  001100                MOV     #STACK,SP               ;SETUP THE STACK POINTER
001712  012737  012362  000020        MOV     #$SCOPE,@#IOTVEC        ;IOT VECTOR FOR SCOPE ROUTINE
001720  012737  000340  000022        MOV     #340,@#IOTVEC+2         ;LEVEL 7
001726  005067  177353                CLR     SYSTNM                  ;INITIALIZE THE TEST NUMBER
001732  012737  012634  000030        MOV     #EMLT,@#EMTVEC          ;EMT VECTOR FOR HLT(ERROR) ROUTINE
001740  012737  000340  000032        MOV     #340,@#EMTVEC+2         ;LEVEL 7
001746  012737  013556  000034        MOV     #STRAP,@#TRAPVEC        ;TRAP VECTOR FOR TRAP CALLS
001754  012737  000340  000036        MOV     #340,@#TRAPVEC+2        ;LEVEL 7
001762  012737  013606  000024        MOV     #PWRDN,@#PWRVEC         ;POWER FAILURE VECTOR
001770  012737  000340  000026        MOV     #340,@#PWRVEC+2         ;LEVEL 7
001776  005067  177276                CLR     SPASS                   ;CLEAR THE PASS COUNT
002002  005067  177276                CLR     SICNT                   ;INITIALIZE THE ITERATION COUNTER
002006  005067  010276                CLR     STIMES                  ;INITIALIZE NUMBER OF ITERATIONS
002012  105067  177276                CLRB    SERFLG                  ;CLEAR THE ERROR FLAG
002016  005067  177270                CLR     SERTTL                  ;CLEAR THE ERROR COUNT
002022  005067  010746                CLR     SESCAPE                 ;CLEAR THE ESCAPE ON ERROR ADDRESS

002026  005037  001630                CLR     @#USERTYPE              ;SET USER SELECTION INDICATOR
                                        ;TO ZERO INDICATING PROGRAM
                                        ;TABULATION

002032  005037  002304                CLR     @#SKT11                 ;CLEAR KT11 PRESENCE FLAG
002036  005037  001420                CLR     @#$SET0                 ;CLEAR THE OFFSET
002042  005037  001422                CLR     @#$SET1                 ;TABLE LOCATIONS FOR
002046  005037  001424                CLR     @#$SET2                 ;THE KT11 OPTION
002052  005037  001426                CLR     @#$SET3
002056  005037  001430                CLR     @#$SET4
002062  005037  001432                CLR     @#$SET5
002066  005037  001434                CLR     @#$SET6
002072  005037  001436                CLR     @#$SET7
002076  005037  001440                CLR     @#$SET10
002102  005037  001442                CLR     @#$SET11
002106  005037  001444                CLR     @#$SET12
002112  005037  001450                CLR     @#NTER0                 ;CLEAR THE INTERLEAVE TABLE
002116  005037  001452                CLR     @#NTER1                 ;ENTRY LOCATIONS
002122  005037  001454                CLR     @#NTER2
002126  005037  001456                CLR     @#NTER3
002132  005037  001460                CLR     @#NTER4
002136  005037  001462                CLR     @#NTER5
002142  005037  001464                CLR     @#NTER6
002146  005037  001466                CLR     @#NTER7
002152  005037  001470                CLR     @#NTER10
002156  005037  001472                CLR     @#NTER11
002162  005037  001474                CLR     @#NTER12
002166  005037  001636                CLR     @#LEAFCNT               ;CLEAR NO. OF ABORTS PER NO. OF
                                        ;CONSECUTIVE LOCS. TESTED FLAG
002172  005037  001642                CLR     @#CPU40                 ;CLEAR PROCESSOR INDICATOR FLAG
002176  013746  000004                MOV     @#4,-(SP)               ;SAVE CONTENTS OF LOC. 4
002202  013746  000010                MOV     @#10,-(SP)              ;SAVE CONTENTS OF LOC. 10
002206  012737  002226  000010        MOV     #1$,@#RESVEC            ;SET UP FOR 'SPL' TRAP ADDRESS
002214  012737  000340  000012        MOV     #340,@#RESVEC+2         ;SET UP FOR 'SPL' TRAP PS
002222  000237                      1$: SPL     7                       ;ATTEMPT TO SET A PRIORITY LEVEL
002224  000403                      2$: BR      2$                      ;BRANCH INDICATING WE ARE ON AN
```

```
                                                                    ;11/45 PROCESSOR
002226  022626                          1$:     CMP     (SP)+,(SP)+             ;RESET THE STACK FROM TRAP
002230  005237  001142                          INC     @#CPUAD                 ;SET FLAG INDICATING WE ARE ON
                                                                    ;AN 11/48 PROCESSOR
002234  012637  000010                  2$:     MOV     (SP)+,@#10              ;RESTORE CONTENTS OF LOC. 10
002240  005037  000012                          CLR     @#12                    ;RESTORE TRAPCATCHER LOC. 12
002244  012737  002324  000004                  MOV     #KTTIMEOUT,@#ERRVEC     ;SET UP KT TIMEOUT ADDRESS
002252  012737  000340  000006                  MOV     #340,@#ERRVEC+2         ;SET UP KT TIMEOUT PS
002260  005777  175720                          TST     @SR0                    ;KT11 ARE YOU THERE?
002264  005077  175714                          CLR     @SR0                    ;YES - INITIALIZE IT IN CASE
                                                                    ;USER DOESN'T WANT IT
002270  013700  177570                          MOV     @#SWR,R0                ;GET SWR CONTENTS
002274  006300                                  ASL     R0                      ;MOVE BIT06 TO BIT07 POSITION
002276  105700                                  TSTB    R0                      ;KT11 PRESENT (OBVIOUSLY) IF
                                                                    ;WE REACH THIS INSTRUCTION
                                                                    ;DOES USER WANT IT?
002300  100412                                  BMI     GO                      ;BRANCH IF NO
002302  005327                                  DEC     (PC)+                   ;YES - SET KT11 FLAG
002304  000000                          $KT11:  0                               ;CONTAINS A -1 IF KT11 OPTION
                                                                    ;IS PRESENT
002306  004737  013134                          JSR     PC,@#$SIZE              ;SEE HOW MUCH MEMORY IS AVAILABLE
002312  005077  175712                          CLR     @KPAR0                  ;CLEAR PAGE 0 OFFSET REGISTER
002316  005277  175662                          INC     @SR0                    ;TURN ON MEMORY MANAGEMENT
002322  000401                                  BR      GO                      ;SKIP NEXT INSTRUCTION
002324  022626                          KTTIMEOUT:      CMP     (SP)+,(SP)+     ;RESET THE STACK FROM TIMEOUT
                                                                    ;KT11 NOT PRESENT, THEREFORE
                                                                    ;ONLY DO BELOW 28K
002326  012637  000004                  GO:     MOV     (SP)+,@#4               ;RESTORE CONTENTS OF LOC. 4
002332  005037  000006                          CLR     @#6                     ;RESTORE CONTENTS OF LOC. 6
002336  004337  011506                          JSR     R3,@#INITIALIZE         ;SET UP TO BEGIN TESTING
002342  016703  177046                          MOV     $TMPAD,R3               ;SET UP FOR MEMORY TABLE CREATION
002346  016702  176764                          MOV     $REGAD,R2               ;SET UP FOR PARITY TABLE CREATION
002352  016700  177040                          MOV     $SETAD,R0               ;SET UP FOR OFFSET TABLE CREATION
                                                                    ;THIS TABLE ONLY HAS EFFECT IF
                                                                    ;MEMORY MGMT IS TURNED ON
002356  016705  177064                          MOV     NTERAD,R5               ;SET UP FOR INTERLEAVE TABLE
                                                                    ;CREATION (8 - WAY INTERLEAVE
                                                                    ;CAPABILITY EXISTS)
                                        ;*****************************************************************
                                        ;
                                        ;LET'S DETERMINE IF SEVERAL REGISTERS EXIST, FOR EXAMPLE,
                                        ;
                                        ;               172100 GOVERNING CORE MEMORY    0 - 8K
                                        ;       &       172102 GOVERNING MOS MEMORY     8 - 16K
                                        ;       &       172112 GOVERNING CORE MEMORY    40 - 56K
                                        ;
                                        ;IF WE WANT TO PRESELECT ONE OF THEM OR CREATE A TABLE OF ALL THOSE
                                        ;AVAILABLE AND CARRY ON TESTING FROM THE TABLE
                                        ;
                                        ;NOTE:  SEE DOCUMENT CONCERNING TABLE APPEARANCES AS A
                                        ;       FUNCTION OF MEMORY MANAGEMENT (KT11 OPTION) BEING
                                        ;       ENABLED OR DISABLED DURING PROGRAM EXECUTION
                                        ;
                                        ;*****************************************************************
```



```
002362  032737  010000  177570                  BIT     #BIT12,@#SWR            ;DOES THE USER WISH TO SELECT THE
                                                                    ;REGISTER?
002370  001445                                  BEQ     FINDONE                 ;BRANCH IF NO
002372                                  MSGTYP:
002372  104000  002400                          TYPE    ,.+4                    ;TYPE ASCIZ STRING
002376  000433                                  BR      64$                     ;GET OVER THE ASCII
                                                .ASCIZ  <15><12>"TYPE THE REGISTER YOU WANT & HIT CARRIAGE RETURN "
002466                                  64$:
002466  104406  001340                          ACCEPT,$REG0                    ;PICK UP THE DESIRED REGISTER
                                                                    ;FROM THE TELETYPE AND STORE
                                                                    ;IN FIRST TABLE LOCATION
002472  005237  001630                          INC     @#USERTYPE              ;SET FLAG INDICATING USER SELECTION
002476  013746  000004                          MOV     @#4,-(SP)               ;SAVE CONTENTS OF LOC. 4
002502  000404                                  BR      NEXT1                   ;SKIP THE NEXT INSTRUCTIONS
002504  012712  172100                  FINDONE:        MOV     #172100,(R2)    ;MOVE 1ST POSSIBLE PARITY
                                                                    ;REGISTER INTO $REG0
002510  013746  000004                          MOV     @#4,-(SP)               ;PUSH CONTENTS OF LOC.4 ONTO STACK
002514  012704  001644                  NEXT1:  MOV     #INTERTABLE,R4          ;INITIALIZE INTERLEAVE TABLE
                                                                    ;POINTER
002520  005737  002304                          TST     @#$KT11                 ;KT11 ARE YOU THERE?
002524  001402                                  BEQ     5$                      ;BRANCH IF NO
002526  005077  175500                          CLR     @KPAR1                  ;RESET PAGE 1 ADDRESS REGISTER
                                                                    ;BEFORE TESTING NEXT PARITY
                                                                    ;CONTROL REGISTER
002532  012737  003144  000004          5$:     MOV     #NOREG,@#4              ;SET PARITY TIMEOUT VECTOR SERVICE ADDRESS
002540  022712  172136                          CMP     #172136,(R2)            ;IS THE ADDRESS IN BOUNDS?
002544  100002                                  BPL     12$                     ;BRANCH IF YES
002546  000137  003172                          JMP     @#NOMORE                ;OTHERWISE - TERMINATE TABLE!
002552  005772  000000                  12$:    TST     @(R2)                   ;YES - IS THIS REGISTER PRESENT?
002556  004737  003222                          JSR     R7,@#PARTST             ;CHECK OUT FOR FATAL ERRORS
                                        ;
                                        ;
                                        ;WE HAVE CHECKED OUT THE REGISTER AND FOUND IT TO BE WORKING PROPERLY
                                        ;NOW WE WILL FIND ITS ASSOCIATED PARITY MEMORY, IF IT EXISTS!!
                                        ;
                                        ;
002562  012737  003112  000004                  MOV     #PARCORE,@#4            ;SET MEMORY TIMEOUT VECTOR
                                                                    ;SERVICE ADDRESS
002570  012737  013700  001640                  MOV     #13700,@#MEMAD          ;SET UP A STARTING ADDRESS
002576  011437  001632                          MOV     (R4),@#BLKCNT           ;SET A COUNTER FOR CONSECUTIVE
                                                                    ;LOCATION CHECKS TO COVER MEMORY
                                                                    ;INTERLEAVING
002602  005737  002304                          TST     @#$KT11                 ;SHOULD I LOOK ABOVE 28K?
002606  100013                                  BPL     1$                      ;BRANCH IF NO
002610  062737  010000  001640                  ADD     #10000,@#MEMAD          ;STEP UP TO A PAGE 1 BASE ADDRESS
                                                                    ;IF MEMORY MANAGEMENT TURNED ON
002616  013737  001640  001634                  MOV     @#MEMAD,@#RESTOREBASE   ;SAVE PAGE 1 BASE ADDRESS
002624  062710  000140                  2$:     ADD     #140,(R0)               ;SET UP AN OFFSET FOR KPAR1
002630  011077  175376                          MOV     (R0),@KPAR1             ;SET OFFSET IN PAGE 1 REGISTER
002634  000406                                  BR      4$                      ;SKIP NEXT 2 INSTRUCTIONS
002636  062737  004000  001640          1$:     ADD     #4000,@#MEMAD           ;STEP UP TO NEXT BANK
002644  013737  001640  001634                  MOV     @#MEMAD,@#RESTOREBASE   ;SAVE INITIAL MEMORY ADDRESS
002652  005777  176762                  4$:     TST     @MEMAD                  ;IS THIS MEMORY AVAILABLE?
002656  013713  001640                          MOV     @#MEMAD,(R3)            ;YES - STORE THIS MEMORY LOCATION
```

```
                                                        ;NOW LET'S SEE IF IT'S PARITY
                                                        ;MEMORY CORRESPONDING TO THE
                                                        ;PARITY REGISTER WE'VE FOUND
                                                        ;KT11 ARE YOU THERE?
                                                        ;BRANCH IF NO
                                                        ;DECREASE CONSECUTIVE
                                                        ;LOCATION COUNTER
                                                        ;ARE WE DONE CHECKING
                                                        ;CONSECUTIVE LOCATIONS?
                                                        ;BRANCH IF YES
                                                        ;STEP UP 1 LOCATION
                                                        ;GO BACK TO TEST WITH THIS
                                                        ;LOCATION
                                                        ;RESTORE PAGE 1 BASE ADDRESS
                                                        ;BEFORE GOING BACK TO INCREASE
                                                        ;OFFSET
                                                        ;STEP TABLE POINTER UP FOR
                                                        ;NEXT VALUE OF CONSECUTIVE
                                                        ;LOCATION TO BE CHECKED
                                                        ;ARE THERE ANY MORE?
                                                        ;BRANCH IF NO
                                                        ;STORE THIS VALUE OF CONSECUTIVE
                                                        ;LOCATION CHECKS
                                                        ;CLEAR INTERLEAVE VALUE HOLDER
                                                        ;BEFORE RETESTING
                                                        ;GO BACK TO TEST WITH THIS VALUE
                                                        ;OF CONSECUTIVE LOCATIONS
                                                        ;INITIALIZE INTERLEAVE TABLE
                                                        ;POINTER
                                                        ;RESET THE CONSECUTIVE
                                                        ;LOCATION COUNTER
                                                        ;CLEAR INTERLEAVE VALUE HOLDER
                                                        ;BEFORE RETESTING
                                                        ;GO BACK TO INCREASE OFFSET
                                                        ;AND TEST
                                                        ;ARE WE UP TO 28K YET?
                                                        ;BRANCH IF YES
                                                        ;DECREASE CONSECUTIVE
                                                        ;LOCATION COUNTER
                                                        ;ARE WE DONE CHECKING CONSEC-
                                                        ;UTIVE LOCATIONS?
                                                        ;BRANCH IF YES
                                                        ;STEP UP 1 LOCATION
                                                        ;GO BACK TO TEST WITH THIS
                                                        ;LOCATION
                                                        ;RESTORE INITIAL MEMORY
                                                        ;ADDRESS BEFORE GOING BACK TO
                                                        ;STEP UP TO NEXT BANK
                                                        ;STEP TABLE POINTER UP FOR
                                                        ;NEXT VALUE OF CONSECUTIVE
                                                        ;LOCATIONS TO BE CHECKED
                                                        ;ARE THERE ANY MORE?
                                                        ;BRANCH IF NO
                                                        ;STORE THIS VALUE OF
```



```
                                                        ;CONSECUTIVE LOCATION CHECKS
                                                        ;CLEAR INTERLEAVE VALUE HOLDER
                                                        ;BEFORE RETESTING
                                                        ;GO BACK TO STEP UP TO THE
                                                        ;NEXT BANK TO CONDUCT TESTING
                                                        ;INITIALIZE INTERLEAVE
                                                        ;TABLE POINTER
                                                        ;RESET THE CONSECUTIVE
                                                        ;LOCATION COUNTER
                                                        ;CLEAR INTERLEAVE VALUE HOLDER
                                                        ;BEFORE RETESTING
                                                        ;GO BACK TO STEP UP TO NEXT
                                                        ;BANK TO CONDUCT TESTING
                                                        ;STORE THE BAD REGISTER WITH
                                                        ;NO PARITY MEMORY
                                                        ;NO PARITY MEMORY FOUND
                                                        ;BELOW 28K!!!!!!!!
                                                        ;DID USER SELECT REGISTER?
                                                        ;BRANCH IF NO
                                                        ;GO BACK TO RETYPE MESSAGE FOR
                                                        ;USER RESPONSE
                                                        ;PLACE NEXT POSSIBLE REGISTER
                                                        ;INTO SAME TABLE LOCATION
                                                        ;GO BACK TO TEST THIS REGISTER
                                                        ;RESET STACK FROM MEMORY TIMEOUT
                                                        ;STORE THE REGISTER THAT EN-
                                                        ;COUNTERED A POSSIBLE HOLE IN
                                                        ;MEMORY
                                                        ;A POSSIBLE HOLE IN MEMORY EXISTS
                                                        ;WITH NO PARITY BELOW IT!!!!!!!!
                                                        ;DID USER SELECT REGISTER?
                                                        ;BRANCH IF NO
                                                        ;GO BACK TO RETYPE MESSAGE FOR
                                                        ;USER RESPONSE
                                                        ;PLACE NEXT POSSIBLE REGISTER
                                                        ;INTO SAME TABLE LOCATION
                                                        ;GO BACK TO TEST THIS REGISTER
                                                        ;RESET STACK FROM REGISTER TIMEOUT
                                                        ;DID THE USER SELECT THE REGISTER?
                                                        ;BRANCH IF NO
                                                        ;YES - USER SELECTED REGISTER NOT
                                                        ;PRESENT ON SYSTEM
                                                        ;GO BACK TO RETYPE MESSAGE
                                                        ;STEP UP TO NEXT PARITY REGISTER
                                                        ;AT SAME TABLE LOCATION
                                                        ;PREVIOUS PARITY REGISTER NOT
                                                        ;PRESENT - SEE IF THE NEXT ONE IS
                                                        ;RESTORE CONTENTS OF LOC. 4
                                                        ;ALL DONE TABLE CREATION
                                                        ;END IT WITH A '0'
                                                        ;START RUNNING PROGRAM WITH
                                                        ;TABLE CONTENTS
                                        ;*****************************************************
                                        ;
```

```
                                ;THE FOLLOWING ROUTINE WILL CREATE A 2 LOCATION MEMORY MAP AT
                                ;THE HIGH END OF A 1K CORE SECTION.  THIS 2 LOCATION MAP WILL
                                ;INITIALLY BE USED TO DETERMINE WHERE/IF PARITY MEMORY
                                ;PRESIDES AND LATER FOR SUBSEQUENT PROGRAM TESTING OF A REGISTER
                                ;
                                ;*************************************************
003204  162700  000376    COMPUT: SUB     #376,R0                 ;DROP DOWN SO AS NOT TO
                                                                  ;DESTROY .ABS LOADER
003210  010001                    MOV     R0,R1                   ;R1 CONTAINS BEGINNING ADDRESS
                                                                  ;OF MEMORY MAP
003212  020020                    CMP     R0,(R0)+                ;STEP R0 TO NEXT ADDRESS
003214  010011                    MOV     R0,@R1                  ;1ST MEMORY LOCATION
003216  011110                    MOV     @R1,@R0                 ;2ND MEMORY LOCATION
003220  000203                    RTS     R3                      ;RETURN TO TEST A DATI
                                                                  ;WITH CONTENTS OF THESE 2 LOCS.
                                ;
                                ;*************************************************
                                ;
                                ;THIS ROUTINE WILL CHECK IF THE PARITY REGISTER IS STATICALLY IN
                                ;GOOD OPERATION FOR TESTING TO BE CONDUCTED
                                ;
                                ;*************************************************
003222  011267  176374    PARTST: MOV     (R2),PARITY             ;GET PARITY REGISTER TO BE USED
                                ;*************************************************
                                ;TEST 1 SET BITS (USED) OF PARITY REGISTER
                                ;*************************************************
003226  000004            TST1:   SCOPE
003230  052777  000001  176364          BIS     #BIT0,@PARITY
003236  032777  000001  176356          BIT     #BIT0,@PARITY           ;DID IT SET?
003244  001001                        BNE     .+4                     ;YES
003246  104002                        HLT     +2                      ;NO = FATAL ERROR TO PROGRAM!
                                ;*************************************************
                                ;TEST 2 CLEAR BITS (USED) OF PARITY REGISTER
                                ;*************************************************
003250  000004            TST2:   SCOPE
003252  042777  000001  176342          BIC     #BIT0,@PARITY
003260  032777  000001  176334          BIT     #BIT0,@PARITY           ;DID IT CLEAR?
003266  001401                        BEQ     .+4                     ;YES
003270  104002                        HLT     +2                      ;NO = FATAL ERROR TO PROGRAM!
                                ;*************************************************
                                ;TEST 3 SET AND CLEAR BITS (USED) OF PARITY REGISTER
                                ;*************************************************
003272  000004            TST3:   SCOPE
003274  052777  000004  176320          BIS     #BIT2,@PARITY
003302  032777  000004  176312          BIT     #BIT2,@PARITY           ;DID IT SET?
003310  001001                        BNE     .+4                     ;YES
003312  104002                        HLT     +2                      ;NO = FATAL ERROR TO PROGRAM!
003314  042777  000004  176300          BIC     #BIT2,@PARITY
003322  032777  000004  176272          BIT     #BIT2,@PARITY           ;DID IT CLEAR?
003330  001401                        BEQ     .+4                     ;YES
003332  104002                        HLT     +2                      ;NO = FATAL ERROR TO PROGRAM!
                                ;*************************************************
                                ;TEST 4 TEST RESET ON BITS 0, 2 AND 15
                                ;*************************************************
```



```
003334  000004            TST4:   SCOPE
003336  005737  002304          TST     @#SKT11                 ;KT11 ON?
003342  100415                  BMI     WHICH1                  ;BRANCH IF YES AND DON'T DO
                                                                ;THIS TEST BECAUSE THE 'RESET'
                                                                ;WILL CLOBBER SEGMENTATION
003344  052777  100005  176250          BIS     #100005,@PARITY
003352  000005                  RESET                           ;EXPECT BITS 0, 2 AND 15 TO CLEAR
003354  032777  100005  176240          BIT     #100005,@PARITY         ;DID THEY CLEAR?
003362  001404                  BEQ     .+12                    ;YES
003364  042777  100005  176230          BIC     #100005,@PARITY         ;NO - CLEAR OUT REGISTER AS A
                                                                ;PRECAUTION
003372  104005                  HLT     +5                      ;RESET DOESN'T WORK
                                ;*************************************************
                                ;TEST 5 WHICH OPTION IS ABOUT TO BE TESTED
                                ;*************************************************
003374  000004            TST5:   SCOPE
                                ;
003376  052777  007740  176216  WHICH1: BIS     #7740,@PARITY           ;IS AN OLD MS11 OPTION
                                                                ;WITH NO ADDRESS BITS
                                                                ;ABOUT TO BE TESTED?
003404  032777  007740  176210          BIT     #7740,@PARITY           ;ADDRESS BITS ABLE TO BE SET?
003412  001402                  BEQ     1$                      ;BRANCH IF NO INDICATING MS11
003414  000004                  SCOPE
003416  000207                  RTS     R7                      ;RETURN TO NORMAL FLOW
003420  005237  001626    1$:     INC     @#MSREGFLAG             ;SET FLAG INDICATING MS11 OPTION
                                                                ;WITH NO ADDRESS BITS
003424  000004                  SCOPE
003426  000207                  RTS     R7                      ;RETURN TO NORMAL FLOW
                                ;
                                ;*************************************************
                                ;
                                ;THE FOLLOWING ROUTINE WILL TAKE EACH 1K BANK OF MEMORY
                                ;THAT IS AVAILABLE AND PERFORM A DATI IN IT
                                ;TO DETERMINE IF PARITY EXISTS THERE.  THIS ROUTINE IS
                                ;ONLY USED DURING TABLE CREATION
                                ;
                                ;*************************************************
003430  010546    ABORT:  MOV     R5,-(SP)                ;SAVE R5 CONTENTS ON STACK
003432  010046            MOV     R0,-(SP)                ;SAVE R0 CONTENTS ON STACK
003434  011300            MOV     (R3),R0                 ;GET THE MEMORY LOCATION
                                                        ;JUST DETERMINED
003436  004337  003204          JSR     R3,@#COMPUT             ;COMPUTE AN AREA IN THIS BANK
                                                        ;FOR DETERMINING PARITY MEMORY
003442  011267  176156          MOV     (R2),PARITY             ;GET THE PARITY REGISTER JUST
                                                        ;FOUND AND TEST WITH IT
                                ;*************************************************
                                ;               TEST A DATI IN THIS BANK
                                ;*************************************************
                                ;                 11/45 **** ROM STATE 221 ****
                                ;
                                ;       11/40 **** ROM STATE 207 ****
003446  012705  011450          MOV     #VECSET,R5              ;SET UP SERVICE ROUTINE ADDRESS
003452  004015                  JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
003454  003500                  ONETRY                          ;ROUTINE ADDRESS
```

```
                        003156  011100                  MOV     @R1,R0                  ;SET UP FOR A DATO
                        003160  010013                  MOV     R0,@R3                  ;DO THE DATO
                        003162  010020                  MOV     R0,@(R0)+               ;DO A DATI
                        003164  042777  000005  176130  BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
                        003172  012600                  MOV     (SP)+,R0                ;RESTORE R0 CONTENTS
                        003174  012605                  MOV     (SP)+,R5                ;RESTORE R5 CONTENTS
                        003176  042207                  BIC     R7                      ;NOT PARITY MEMORY!
                                                                                        ;RETURN TO TEST AT NEXT
                                                                                        ;INCREMENT
                        003200  042777  000005  176114  ONETRY: BIC     #BIT2!BIT0,@PARITY      ;WE HAVE PARITY MEMORY - PROCEED
                        003206  016600  000000                  MOV     0(SP),R0                ;RESTORE R0 CONTENTS
                        003212  016605  000002                  MOV     6(SP),R5                ;RESTORE R5 CONTENTS
                        003216  005237  001634                  INC     @#LEAFCNT               ;INCREMENT INTERLEAVE COUNTER
                        003222  022737  000003  001636          CMP     #3,@#LEAFCNT            ;3 ABORTS REACHED?
                        003230  001403                  BEQ     1$                      ;BRANCH IF YES
                        003232  062706  000010                  ADD     #10,SP                  ;BYPASS JUNK ON STACK
                        003236  000207                  RTS     R7                      ;RETURN TO TEST AT NEXT INCREMENT
                                                1$:     CMP     (SP)+,(SP)+             ;POP STACK BACK FROM PARITY ABORT
                        003240  012600                  MOV     (SP)+,R0                ;RESTORE R0 CONTENTS
                        003242  012605                  MOV     (SP)+,R5                ;RESTORE R5 CONTENTS
                        003244  005726                  TST     (SP)+                   ;POP STACK ONCE FOR ABORT ROUTINE
                                                                                        ;ENTRY
                        003246  005737  031630                  TST     @#USERTYPE              ;DID USER TYPE IN REGISTER?
                        003252  001404                  BEQ     2$                      ;BRANCH IF NO
                        003254  016415  000002                  MOV     2(R4),(R5)              ;SET INTERLEAVE VALUE INTO
                                                                                        ;TABLE
                        003260  000137  003620                  JMP     @#START                 ;AND LOOK ON THE USER SELECTED
                                                                                        ;REGISTER FOR TESTING
                        003264  005723                  2$:     TST     (R3)+                   ;USER DIDN'T SELECT - SO STEP UP
                                                                                        ;TO NEXT MEMORY TABLE LOCATION
                        003266  005720                  TST     (R0)+                   ;STEP UP TO NEXT OFFSET TABLE
                                                                                        ;LOCATION - THIS TABLE WILL ONLY
                                                                                        ;BE APPLICABLE IF MEMORY MGMT
                                                                                        ;IS TURNED ON
                        003270  012212                  MOV     (R2)+,(R2)              ;SET NEXT POSSIBLE REGISTER INTO
                        003272  062712  000002                  ADD     #2,(R2)                 ;NEXT REGISTER TABLE LOCATION
                        003276  016425  000002                  MOV     2(R4),(R5)+             ;SET INTERLEAVE VALUE INTO
                                                                                        ;TABLE
                        003302  005037  001636                  CLR     @#LEAFCNT               ;RESET NO. OF ABORTS COUNTER
                        003306  005037  001626                  CLR     @#MSREGFLAG             ;CLEAR PARITY TYPE INDICATOR
                        003312  000137  002514                  JMP     @#NEXT1                 ;GO BACK TO CHECK NEXT POSSIBLE
                                                                                        ;PARITY REGISTER
```

```
                        ;*****************************************************************
                        ;*****************************************************************
                        ;*****************************************************************
                        ;*****************************************************************
                        ;
                        ;IF WE HAVE REACHED THIS POINT IN THE PROGRAM THEN,
                        ;
                        ;@REG0(LOCATION 1340) WILL CONTAIN THE FIRST PARITY REGISTER
```



```
                        ;
                        ;               LOCATION 1342 UP TO 1364 WILL CONTAIN ANY MORE
                        ;               REGISTERS FOUND
                        ;@TMP0(LOCATION 1366) WILL CONTAIN AN ADDRESS IN THE BANK
                        ;               THAT HAS PARITY ASSOCIATED WITH THE PARITY
                        ;               REGISTER IN $REG0
                        ;
                        ;               LOCATION 1370 UP TO 1412 WILL CONTAIN THE CORRESPONDING
                        ;               MEMORY PARITY LOCATIONS FOR THE OTHER REGISTERS
                        ;@$ETO(LOCATION 1420) WILL CONTAIN THE OFFSET VALUE TO BE USED
                        ;               WITH THE CORRESPONDING VALUE IN $TMP0
                        ;
                        ;               LOCATION 1422 TO 1444 WILL CONTAIN THE CORRESPONDING
                        ;               OFFSET VALUES FOR THE OTHER REGISTERS
                        ;
                        ;NTER0(LOCATION 1450) WILL CONTAIN THE INTERLEAVE FACTOR TO BE USED
                        ;               WITH THE PARITY REGISTER IN $REG0
                        ;
                        ;               LOCATION 1452 TO 1474 WILL CONTAIN THE CORRESPONDING
                        ;               INTERLEAVE FACTORS FOR THE OTHER REGISTERS
                        ;
                        ;NOTE:  TO UNDERSTAND THE TABLE SETUP SEE THE DOCUMENT
                        ;
                        ;
                        ;*****************************************************************
                        ;*****************************************************************
                        ;*****************************************************************
                        ;*****************************************************************


                        ;*****************************************************************
                        ;
                        ;THE REST OF THE PROGRAM WILL TEST ALL PARITY ABORTS
                        ;USING EITHER THE PARITY REGISTER TYPED BY THE USER OR THOSE
                        ;ENCOUNTERED IN THE TABLE PREVIOUSLY GENERATED.
                        ;
                        ;*****************************************************************
                        003620  005777  175512  START:  TST     @#REGAD                 ;ARE WE AT END OF TABLE?
                        003624  001401                  BEQ     1$                      ;BRANCH IF YES
                        003626  000402                  BR      2$                      ;OTHERWISE SKIP NEXT INSTRUCTION
                        003630  000137  011464  1$:     JMP     @#RESTART               ;SET UP TO START TABLE OVER!!
                        003634  017767  175476  175760  2$:     MOV     @#REGAD,PARITY          ;GET PARITY REGISTER TO BE USED
                        003642  162767  000002  175432          SUB     #2,$TSTNM               ;DECREASE TEST NO. FOR SCOPE
                                                                                        ;LOOPING AN CASE TEST #4 WAS
                                                                                        ;DONE BECAUSE TEST #4 BEING
                                                                                        ;DONE THRUWB $TSTNM COUNT OFF
                                                                                        ;BY 2
                        003650  005737  002304                  TST     @#KT11                  ;KT11 ARE YOU THERE?
                        003654  100006                  BPL     3$                      ;BRANCH IF NO
                        003656  017777  175534  174346          MOV     @#$ETAD,@#PAR1          ;GET THE OFFSET NEEDED
                        003664  062767  000001  175410          ADD     #1,$TSTNM               ;SET TEST NO. TO A CORRECT
                                                                                        ;VALUE; TEST #4 WAS NOT
                                                                                        ;EXECUTED
                        003672  017700  175516  3$:     MOV     @$TMPAD,R0              ;GET PARITY MEMORY ASSOCIATED
```

```
                                                                        ;WITH THE PARITY REGISTER
003676  004337  003200                  JSR     R3,@#COMPUT             ;COMPUTE AN AREA IN THIS BANK
                                                                        ;FOR TESTING
003702  010137  001476                  MOV     R1,@#NEWSTK             ;SET UP A NEW STACK POINTER
                                                                        ;FOR STACK OPERATIONS IN CASE
                                                                        ;NO PARITY MEMORY RESIDES IN
                                                                        ;LOWER 4K
003706  004737  003376                  JSR     PC,@#WHICM1             ;DETERMINE IF WE ARE ABOUT TO
                                                                        ;TEST AN ULD MOS DESIGN
003712  012705  011460                  MOV     #VECSET,R5              ;SET UP SERVICE ROUTINE ADDRESS
                                ;*********************************************************
                                ;TEST 6 TEST (ADDRESS)  SM0,DM3  MOV INSTRUCTION
                                ;*********************************************************
003716  000004                  TST6:   SCOPE
                                ;
                                ;               11/45 #*** ROM STATE 221 ****
                                ;
                                ;       11/40 **** ROM STATE 207 ****
003720  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
003722  003752                          A                               ;ROUTINE ADDRESS
003724  011100                          MOV     @R1,R0                  ;SET UP FOR DATO
003726  012010                          MOV     R0,@R0                  ;DO THE DATO
003730  012737  003740  001332          MOV     #.+10,@#SVDDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
003736  010030                          MOV     R0,@(R0)+               ;DO A DATI
003740  042777  000004  175650          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
003746  104001                          HLT     +1                      ;DIDN'T ABORT
003750  000410                          BR      .+22                    ;GO TO NEXT TEST
003752  042777  000005  175642  A1:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
003760  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
003764  104003                          HLT     +3                      ;ABORTED INCORRECTLY
003766  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                ;*********************************************************
                                ;TEST 7 TEST (ADDRESS)  SM0,DM5  MOV INSTRUCTION
                                ;*********************************************************
003772  000004                  TST7:   SCOPE
                                ;               11/45 #*** ROM STATE 231 ****
                                ;
                                ;       11/40 **** ROM STATE 207 ****
003774  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
003776  004026                          A0                              ;ROUTINE ADDRESS
004000  011100                          MOV     @R1,R0                  ;SET UP FOR DATO
004002  010028                          MOV     R0,(R0)+                ;DO THE DATO
004004  012737  004014  001332          MOV     #.+10,@#SVDDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
004012  010050                          MOV     R0,-(R0)                ;DO A DATI
004014  042777  000004  175600          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
004022  104001                          HLT     +1                      ;DIDN'T ABORT
004024  000410                          BR      .+22                    ;GO TO NEXT TEST
004026  042777  000005  175566  A0:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
004034  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
004040  104003                          HLT     +3                      ;ABORTED INCORRECTLY
004042  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
```



```
                                ;*********************************************************
                                ;TEST 10        TEST (DATA)  SM1,DM6  MOV INSTRUCTION
                                ;*********************************************************
004046  000004                  TST10:  SCOPE
                                ;               11/45 #*** ROM STATE 27 ****
                                ;
                                ;       11/40 **** ROM STATE 206 ****
004050  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
004052  004104                          A1                              ;ROUTINE ADDRESS
004054  011100                          MOV     @R1,R0                  ;SET UP FOR DATO
004056  010010                          MOV     R0,@R0                  ;DO THE DATO
004060  012737  004070  001332          MOV     #.+10,@#SVDDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
004066  011060  177776                  MOV     @R0,-2(R0)              ;DO A DATI
004072  042777  000004  175522          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
004100  104001                          HLT     +1                      ;DIDN'T ABORT
004102  000410                          BR      .+22                    ;GO TO NEXT TEST
004104  042777  000005  175510  A1:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
004112  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
004116  104003                          HLT     +3                      ;ABORTED INCORRECTLY
004120  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                ;*********************************************************
                                ;TEST 11        TEST (ADDRESS)  SM0,DM7  MOV INSTRUCTION
                                ;*********************************************************
004124  000004                  TST11:  SCOPE
                                ;               11/45 #*** ROM STATE 231 ****
                                ;
                                ;       11/40 **** ROM STATE 207 ****
004126  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
004130  004162                          A2                              ;ROUTINE ADDRESS
004132  011100                          MOV     @R1,R0                  ;SET UP FOR A DATO
004134  010020                          MOV     R0,(R0)+                ;DO THE DATO
004136  012737  004150  001332          MOV     #.+12,@#SVDDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
004144  010070  177776                  MOV     R0,@-2(R0)              ;DO A DATI
004150  042777  000004  175444          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
004156  104001                          HLT     +1                      ;DIDN'T ABORT
004160  000410                          BR      .+22                    ;GO TO NEXT TEST
004162  042777  000005  175432  A2:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
004170  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
004174  104003                          HLT     +3                      ;ABORTED INCORRECTLY
004176  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                ;*********************************************************
                                ;TEST 12        TEST (DATA)  SM0,DM2  CMP INSTRUCTION
                                ;*********************************************************
004202  000004                  TST12:  SCOPE
                                ;               11/45 #*** ROM STATE 175 ****
                                ;
                                ;       11/40 **** ROM STATE 267 ****
004204  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
004206  004236                          0                               ;ROUTINE ADDRESS
004210  011100                          MOV     @R1,R0                  ;SET UP FOR DATO
```

```
004212  010010                          MOV     R0,@R0                  ;DO THE DATO
004214  012737  004224  001332          MOV     #.+10,@#$DDAT           ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
004222  020020                          CMP     R0,(R0)+                ;DO A DATIP, DATI
004224  042777  000004  175370          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
004232  104001                          HLT     +1                      ;DIDN'T ABORT
004234  000410                          BR      .+22                    ;GO TO NEXT TEST
004236  042777  000005  175356  B:      BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
004244  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
004250  104003                          HLT     +3                      ;ABORTED INCORRECTLY
004252  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                ;*************************************************************
                                ;TEST 13        TEST (DATA)  SM0,DM4  CMP INSTRUCTION
                                ;*************************************************************
004256  000004                  TST13:  SCOPE
                                ;               11/45 **** ROM STATE 177 ****
                                ;
                                ;       11/40 **** ROM STATE 267 ****
004260  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
004262  004312                          B0                              ;ROUTINE ADDRESS
004264  011100                          MOV     @R1,R0                  ;SET UP FOR DATO
004266  010020                          MOV     R0,(R0)+                ;DO THE DATO
004270  012737  004300  001332          MOV     #.+10,@#$DDAT           ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
004276  020040                          CMP     R0,-(R0)                ;DO A DATIP, DATI
004300  042777  000004  175314          BIC     #BIT2,@PARITY .         ;WRITE NORMAL FOR EMT CALL
004306  104001                          HLT     +1                      ;DIDN'T ABORT
004310  000410                          BR      .+22                    ;GO TO NEXT TEST
004312  042777  000005  175302  B0:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
004320  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
004324  104003                          HLT     +3                      ;ABORTED INCORRECTLY
004326  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                ;*************************************************************
                                ;TEST 14        TEST (DATA)  SM0,DM6  CMP INSTRUCTION
                                ;*************************************************************
004332  000004                  TST14:  SCOPE
                                ;               11/45 **** ROM STATE 177 ****
                                ;
                                ;       11/40 **** ROM STATE 267 ****
004334  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
004336  004370                          B1                              ;ROUTINE ADDRESS
004340  011100                          MOV     @R1,R0                  ;SET UP FOR DATO
004342  010020                          MOV     R0,(R0)+                ;DO THE DATO
004344  012737  004356  001332          MOV     #.+12,@#$DDAT           ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
004352  020060  177776                  CMP     R0,-2(R0)               ;DO A DATI, DATIP
004356  042777  000004  175236          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
004364  104001                          HLT     +1                      ;DIDN'T ABORT
004366  000410                          BR      .+22                    ;GO TO NEXT TEST
004370  042777  000005  175224  B1:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
004376  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
```



```
004402  104003                          HLT     +3                      ;ABORTED INCORRECTLY
004404  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                ;*************************************************************
                                ;TEST 15        TEST (DATA)  SM0,DM1  CMP INSTRUCTION
                                ;*************************************************************
004410  000004                  TST15:  SCOPE
                                ;               11/45 **** ROM STATE 175 ****
                                ;
                                ;       11/40 **** ROM STATE 267 ****
004412  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
004414  004444                          B2                              ;ROUTINE ADDRESS
004416  011100                          MOV     @R1,R0                  ;SET UP FOR DATO
004420  010010                          MOV     R0,@R0                  ;DO THE DATO
004422  012737  004432  001332          MOV     #.+10,@#$DDAT           ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
004430  020010                          CMP     R0,@R0                  ;DO A DATIP, DATI
004432  042777  000004  175162          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
004440  104001                          HLT     +1                      ;DIDN'T ABORT
004442  000410                          BR      .+22                    ;GO TO NEXT TEST
004444  042777  000005  175150  B2:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
004452  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
004456  104003                          HLT     +3                      ;ABORTED INCORRECTLY
004460  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                ;*************************************************************
                                ;TEST 16        TEST (DATA)  SM0,DM3  CMP INSTRUCTION
                                ;*************************************************************
004464  000004                  TST16:  SCOPE
                                ;               11/45 **** ROM STATE 177 ****
                                ;
                                ;       11/40 **** ROM STATE 267 ****
004466  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
004470  004536                          B3                              ;ROUTINE ADDRESS
004472  011100                          MOV     @R1,R0                  ;SET UP FOR DATO
004474  010020                          MOV     R0,(R0)+                ;DO THE DATO
004476  042777  000004  175116          BIC     #BIT2,@PARITY           ;WRITE NORMAL
004504  011110                          MOV     @R1,@R0                 ;WRITE ADDRESS NORMAL (DATI)
004506  052777  000004  175106          BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
004514  012737  004524  001332          MOV     #.+10,@#$DDAT           ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
004522  020030                          CMP     R0,@(R0)+               ;DO A DATI, DATIP
004524  042777  000004  175070          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
004532  104001                          HLT     +1                      ;DIDN'T ABORT
004534  000410                          BR      .+22                    ;GO TO NEXT TEST
004536  042777  000005  175056  B3:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
004544  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
004550  104003                          HLT     +3                      ;ABORTED INCORRECTLY
004552  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                ;*************************************************************
                                ;TEST 17        TEST (ADDRESS)  SM0,DM3  CMP INSTRUCTION
                                ;*************************************************************
004556  000004                  TST17:  SCOPE
                                ;               11/45 **** ROM STATE 221 ****
```

```
                                        ;
                                        ;       11/40 **** ROM STATE 264 ****
004560  004715                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
004562  004612                          B4                              ;ROUTINE ADDRESS
004564  011100                          MOV     @R1,R0                  ;SET UP FOR A DATO
004566  010010                          MOV     R0,@R0                  ;DO THE DATO
004570  012737  004603  001332          MOV     #.+10,@#$$ODAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
004576  020020                          CMP     R0,@(R0)+               ;DO A DATI
004600  042777  000004  175014          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
004606  100001                          MLT     #1                      ;DIDN'T ABORT
004610  000410                          BR      .+22                    ;GO TO NEXT TEST
004612  042777  000005  175002  B4:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
004620  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
004624  100003                          MLT     #3                      ;ABORTED INCORRECTLY
004626  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                ;*****************************************************************
                                ;TEST 20        TEST DATI (DATA)  SM0,DM5  CMP INSTRUCTION
                                ;*****************************************************************
004632  000004                  TST20:  SCOPE
                                ;                       11/05 **** ROM STATE 177 ****
                                ;
                                        ;       11/40 **** ROM STATE 267 ****
004634  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
004636  004706                          B5                              ;ROUTINE ADDRESS
004640  011100                          MOV     @R1,R0                  ;SET UP FOR A DATO
004642  042777  000004  174752          BIC     #BIT2,@PARITY           ;WRITE NORMAL
004650  010060  177776                  MOV     R0,-2(R0)               ;DO THE DATO
004654  052777  000004  174740          BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
004662  011110                          MOV     @R1,@R0                 ;DO THE DATO
004664  012737  004674  001332          MOV     #.+10,@#$$ODAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
004672  020050                          CMP     R0,@-(R0)               ;DO A DATI, DATIP
004674  042777  000004  174726          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
004702  100001                          MLT     #1                      ;DIDN'T ABORT
004704  000410                          BR      .+22                    ;GO TO NEXT TEST
004706  042777  000005  174706  B5:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
004714  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
004720  100003                          MLT     #3                      ;ABORTED INCORRECTLY
004722  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                ;*****************************************************************
                                ;TEST 21        TEST (ADDRESS)  SM0,DM5  CMP INSTRUCTION
                                ;*****************************************************************
004726  000004                  TST21:  SCOPE
                                ;                       11/45 **** ROM STATE 231 ****
                                ;
                                        ;       11/40 **** ROM STATE 264 ****
004730  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
004732  004766                          B6                              ;ROUTINE ADDRESS
004734  011100                          MOV     @R1,R0                  ;SET UP FOR A DATO
004736  010010                          MOV     R0,(R0)                 ;DO THE DATO
004740  062700  000002                  ADD     #2,R0
```



```
004744  012737  004754  001332          MOV     #.+10,@#$$ODAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
004752  020050                          CMP     R0,@-(R0)               ;DO A DATI
004754  042777  000004  174640          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
004762  100001                          MLT     #1                      ;DIDN'T ABORT
004764  000410                          BR      .+22                    ;GO TO NEXT TEST
004766  042777  000005  174626  B6:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
004774  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
005000  100003                          MLT     #3                      ;ABORTED INCORRECTLY
005002  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                ;*****************************************************************
                                ;TEST 22        TEST (DATA)  SM2,DM0  CMP INSTRUCTION
                                ;*****************************************************************
005006  000004                  TST22:  SCOPE
                                ;                       11/45 **** ROM STATE 27 ****
                                ;
                                        ;       11/40 **** ROM STATE 250 ****
005010  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
005012  005042                          C                               ;ROUTINE ADDRESS
005014  011100                          MOV     @R1,R0                  ;SET UP FOR DATO
005016  010010                          MOV     R0,@R0                  ;DO THE DATO
005020  012737  005030  001332          MOV     #.+10,@#$$ODAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
005026  022000                          CMP     (R0)+,R0                ;DO A DATI
005030  042777  000004  174564          BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
005036  100001                          MLT     #1                      ;DIDN'T ABORT
005040  000410                          BR      .+22                    ;GO TO NEXT TEST
005042  042777  000005  174552  C5:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
005050  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
005054  100003                          MLT     #3                      ;ABORTED INCORRECTLY
005056  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                ;*****************************************************************
                                ;TEST 23        TEST (DATA)  SM4,DM0  CMP INSTRUCTION
                                ;*****************************************************************
005062  000004                  TST23:  SCOPE
```

```
1278                                    ;               11/45 **** ROM STATE 27 ****
```



```
1279                                    ;
1280                                    ;       11/40 **** ROM STATE 250 ****
1281  005064  004015                            JSR     R0,(R5)         ;SET UP PARITY VECTOR SERVICE
1282  005066  005110                            CO                      ;ROUTINE ADDRESS
1283  005070  011100                            MOV     @R1,R0          ;SET UP FOR DATO
1284  005072  010020                            MOV     R0,(R0)+        ;DO THE DATO
```

```
005074 012737 005104 001332              MOV     #.+10,@#$WDDAT          ;STORE THE PC THAT SHOULD
                                                                         ;BE PUSHED ON THE STACK
                                                                         ;IF A PARITY ABORT OCCURS
005102 024000                            CMP     -(R0),R0                ;DO A DATI
005104 042777 000004 174510              BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
005112 104001                            HLT     +1                      ;DIDN'T ABORT
005114 000410                            BR      .+22                    ;GO TO NEXT TEST
005116 042777 000005 174476      C0:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
005124 004037 011550                     JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
005130 104003                            HLT     +3                      ;ABORTED INCORRECTLY
005132 012706 001100                     MOV     #STACK,SP               ;RESET THE STACK
                                 ;*****************************************************************
                                 ;TEST 24        TEST (DATA)  SM3,DM0  CMP INSTRUCTION
                                 ;*****************************************************************
005136 000004                    TST24:  SCOPE
                                 ;               11/45 **** ROM STATE 146 ****
                                 ;
                                 ;       11/40 **** ROM STATE 250 ****
005140 004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
005142 005210                            C1                              ;ROUTINE ADDRESS
005144 011100                            MOV     @R1,R0                  ;SET UP FOR DATO
005146 010020                            MOV     R0,(R0)+                ;DO THE DATO (DATA OTHER PARITY)
005150 042777 000004 174444              BIC     #BIT2,@PARITY           ;WRITE NORMAL
005156 011110                            MOV     @R1,@R0                 ;DO A DATO (ADDRESS NORMAL)
005160 052777 000004 174434              BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
005166 012737 005176 001332              MOV     #.+10,@#$WDDAT          ;STORE THE PC THAT SHOULD
                                                                         ;BE PUSHED ON THE STACK
                                                                         ;IF A PARITY ABORT OCCURS
005174 023000                            CMP     @(R0)+,R0               ;DO A DATI
005176 042777 000004 174416              BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
005204 104001                            HLT     +1                      ;DIDN'T ABORT
005206 000410                            BR      .+22                    ;GO TO NEXT TEST
005210 042777 000005 174404      C1:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
005216 004037 011550                     JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
005222 104003                            HLT     +3                      ;ABORTED INCORRECTLY
005224 012706 001100                     MOV     #STACK,SP               ;RESET THE STACK
                                 ;*****************************************************************
                                 ;TEST 25        TEST (DATA)  SM5,DM0  CMP INSTRUCTION
                                 ;*****************************************************************
005230 000004                    TST25:  SCOPE
                                 ;               11/45 **** ROM STATE 146 ****
                                 ;
                                 ;       11/40 **** ROM STATE 250 ****
005232 004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
005234 005304                            C2                              ;ROUTINE ADDRESS
005236 011100                            MOV     @R1,R0                  ;SET UP FOR A DATO
005240 042777 000004 174354              BIC     #BIT2,@PARITY           ;WRITE NORMAL
005246 010060 177776                     MOV     R0,-2(R0)               ;DO A DATO (ADDRESS NORMAL)
005252 052777 000004 174342              BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
005260 011110                            MOV     @R1,@R0                 ;DO A DATO (DATA OTHER PARITY)
005262 012737 005272 001332              MOV     #.+10,@#$WDDAT          ;STORE THE PC THAT SHOULD
                                                                         ;BE PUSHED ON THE STACK
                                                                         ;IF A PARITY ABORT OCCURS
005270 025000                            CMP     @-(R0),R0               ;DO A DATI
```



```
005272 042777 000004 174322              BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
005300 104001                            HLT     +1                      ;DIDN'T ABORT
005302 000410                            BR      .+22                    ;GO TO NEXT TEST
005304 042777 000005 174310      C2:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
005312 004037 011550                     JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
005316 104003                            HLT     +3                      ;ABORTED INCORRECTLY
005320 012706 001100                     MOV     #STACK,SP               ;RESET THE STACK
                                 ;*****************************************************************
                                 ;TEST 26        TEST (DATA)  SM1,DM0  CMP INSTRUCTION
                                 ;*****************************************************************
005324 000004                    TST26:  SCOPE
                                 ;               11/45 **** ROM STATE 27 ****
                                 ;
                                 ;       11/40 **** ROM STATE 250 ****
005326 004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
005330 005360                            C3                              ;ROUTINE ADDRESS
005332 011100                            MOV     @R1,R0                  ;SET UP FOR DATO
005334 010010                            MOV     R0,@R0                  ;DO THE DATO
005336 012737 005346 001332              MOV     #.+10,@#$WDDAT          ;STORE THE PC THAT SHOULD
                                                                         ;BE PUSHED ON THE STACK
                                                                         ;IF A PARITY ABORT OCCURS
005344 021000                            CMP     @R0,R0                  ;DO A DATI
005346 042777 000004 174246              BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
005354 104001                            HLT     +1                      ;DIDN'T ABORT
005356 000410                            BR      .+22                    ;GO TO NEXT TEST
005360 042777 000005 174234      C3:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
005366 004037 011550                     JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
005372 104003                            HLT     +3                      ;ABORTED INCORRECTLY
005374 012706 001100                     MOV     #STACK,SP               ;RESET THE STACK
                                 ;*****************************************************************
                                 ;TEST 27        TEST (DATA)  SM6,DM0  CMP INSTRUCTION
                                 ;*****************************************************************
005400 000004                    TST27:  SCOPE
                                 ;               11/45 **** ROM STATE 142 ****
                                 ;
                                 ;       11/40 **** ROM STATE 250 ****
005402 004015                            JSR R0,(R5)                     ;SET UP PARITY VECTOR SERVICE
005404 005436                            C4                              ;ROUTINE ADDRESS
005406 011100                            MOV     @R1,R0                  ;SET UP FOR A DATO
005410 010020                            MOV     R0,(R0)+                ;DO THE DATO
005412 012737 005424 001332              MOV     #.+12,@#$WDDAT          ;STORE THE PC THAT SHOULD
                                                                         ;BE PUSHED ON THE STACK
                                                                         ;IF A PARITY ABORT OCCURS
005420 026000 177776                     CMP     -2(R0),R0               ;DO A DATI
005424 042777 000004 174170              BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
005432 104001                            HLT     +1                      ;DIDN'T ABORT
005434 000410                            BR      .+22                    ;GO TO NEXT TEST
005436 042777 000005 174156      C4:     BIC     #BIT2!BIT0,@PARITY      ;WRITE NORMAL AND DISABLE
005444 004037 011550                     JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
005450 104003                            HLT     +3                      ;ABORTED INCORRECTLY
005452 012706 001100                     MOV     #STACK,SP               ;RESET THE STACK
                                 ;*****************************************************************
                                 ;TEST 30        TEST (ADDRESS)  SM7,DM0  CMP INSTRUCTION
                                 ;*****************************************************************
```

```
                        TST30:  SCOPE
                                ;
                                ;                       11/45 **** ROM STATE 142 ****
                                ;
                                ;       11/40 **** ROM STATE 245 ****
005454  004004                        JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
                                                                        ;ROUTINE ADDRESS
005456  005462                        MOV     #R1,R0                  ;SET UP FOR A DATO
005460  011100                        MOV     R0,(R0)+                ;DO THE DATO
005462  012737  005502  001332        MOV     #.+12,@#54DDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
005470  027000  177776                CMP     @-2(R0),R0              ;DO A DATI
005474  002777  000004  174112        BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
005502  000001                        HLT     #1                      ;DIDN'T ABORT
005504  104001                        BR      .+22                    ;GO TO NEXT TEST
005506  040410                C3:     BIC     #BIT2!BIT5,@PARITY      ;WRITE NORMAL AND DISABLE
005510  004737  000005  174100        JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
005516  011550                        MLT     +3                      ;ABORTED INCORRECTLY
005520  104003                        MOV     @STACK,SP               ;RESET THE STACK
005522  012706  001100
                        ;*****************************************************************
                        ;TEST 31        TEST (ADDRESS)  SM3,DM3  CMP INSTRUCTION
                        ;*****************************************************************
                        TST31:  SCOPE
                                ;
                                ;                       11/45 **** ROM STATE 27 ****
                                ;
                                ;       11/40 **** ROM STATE 245 ****
005534  004015                        JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
005536  005540                        C6                              ;ROUTINE ADDRESS
005540  011100                        MOV     #R1,R0                  ;SET UP FOR A DATO
005542  010010                        MOV     R0,@R0                  ;DO THE DATO
005544  012737  005556  001332        MOV     #.+10,@#54DDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
005552  023000                        CMP     @(R0)+,R0               ;DO A DATI
005554  002777  000004  174036        BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
005562  000001                        HLT     #1                      ;DIDN'T ABORT
005564  104001                        BR      .+22                    ;GO TO NEXT TEST
005566  040410                C4:     BIC     #BIT2!BIT5,@PARITY      ;WRITE NORMAL AND DISABLE
005570  004737  000005  174024        JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
005576  011550                        MLT     +3                      ;ABORTED INCORRECTLY
005600  104003                        MOV     @STACK,SP               ;RESET THE STACK
005602  012706  001100
                        ;*****************************************************************
                        ;TEST 32        TEST (ADDRESS)  SM3,DM3  CMP INSTRUCTION
                        ;*****************************************************************
005610  000004                TST32:  SCOPE
                                ;
                                ;                       11/45 **** ROM STATE 27 ****
                                ;
                                ;       11/40 **** ROM STATE 245 ****
005612  004015                        JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
005614  005620                        C7                              ;ROUTINE ADDRESS
005616  011100                        MOV     #R1,R0                  ;SET UP FOR A DATO
005620  010020                        MOV     R0,(R0)+                ;DO THE DATO
005622  012737  005632  001332        MOV     #.+10,@#54DDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
```



```
                                                                        ;IF A PARITY ABORT OCCURS
005630  027000  177776                CMP     @-(R0),R0               ;DO A DATI
005634  002777  000004  173762        BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
005642  000001                        HLT     #1                      ;DIDN'T ABORT
005644  104001                        BR      .+22                    ;GO TO NEXT TEST
005646  040410                C7:     BIC     #BIT2!BIT5,@PARITY      ;WRITE NORMAL AND DISABLE
005652  004737  000005  173750        JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
005656  011550                        MLT     +3                      ;ABORTED INCORRECTLY
005660  104003                        MOV     @STACK,SP               ;RESET THE STACK
005662  012706  001100
                        ;*****************************************************************
                        ;TEST 33        TEST (DATA)  SM7,DM3  CMP INSTRUCTION
                        ;*****************************************************************
005664  000004                TST33:  SCOPE
                                ;
                                ;                       11/45 **** ROM STATE 146 ****
                                ;
                                ;       11/40 **** ROM STATE 250 ****
005666  004015                        JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
005670  005702                        C8                              ;ROUTINE ADDRESS
005672  011100                        MOV     #R1,R0                  ;SET UP FOR A DATO
005674  002777  000004  173720        BIC     #BIT2,@PARITY           ;WRITE NORMAL
005702  010060  177776                MOV     R0,-2(R0)               ;DO A DATO (ADDRESS NORMAL)
005706  052777  000004  173706        BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
005714  011110                        MOV     @R1,@R0                 ;DO A DATO (DATA OTHER PARITY)
005716  012737  005730  001332        MOV     #.+12,@#54DDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
005724  027000  177776                CMP     @-2(R0),R0              ;DO A DATI
005730  002777  000004  173664        BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
005736  104001                        HLT     #1                      ;DIDN'T ABORT
005740  040410                        BR      .+22                    ;GO TO NEXT TEST
005742  042777  000005  173652  C8:   BIC     #BIT2!BIT5,@PARITY      ;WRITE NORMAL AND DISABLE
005750  004037  011550                JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
005754  104003                        MLT     +3                      ;ABORTED INCORRECTLY
005756  012706  001100                MOV     @STACK,SP               ;RESET THE STACK
                        ;*****************************************************************
                        ;TEST 34        TEST  DM3  JMP INSTRUCTION
                        ;*****************************************************************
005762  000004                TST34:  SCOPE
                                ;
                                ;                       11/45 **** ROM STATE 221 ****
                                ;
                                ;       11/40 **** ROM STATE 303 ****
005764  004015                        JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
005766  006020                        D                               ;ROUTINE ADDRESS
005770  011100                        MOV     #R1,R0                  ;SET UP FOR A DATO
005772  012710  006006                MOV     #DD,@R0                 ;DO THE DATO
005776  012737  006006  001332        MOV     #.+10,@#54DDAT          ;STORE THE PC THAT SHOULD
                                                                        ;BE PUSHED ON THE STACK
                                                                        ;IF A PARITY ABORT OCCURS
006004  000130                        JMP     @(R0)+                  ;DO A DATI
006006  002777  000004  173606  DD:   BIC     #BIT2,@PARITY           ;WRITE NORMAL FOR EMT CALL
006014  104001                        HLT     #1                      ;DIDN'T ABORT
006016  040410                        BR      .+22                    ;GO TO NEXT TEST
006020  042777  000005  173574  D:    BIC     #BIT2!BIT5,@PARITY      ;WRITE NORMAL AND DISABLE
006026  004037  011550                JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
```

```
                        ;THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS:
                        ;
                        ;1ST PUSH = ADDRESS OF THE TAG 'VECSET' (NORMAL)
                        ;           THIS ADDRESS WOULD BE PLACED IN
                        ;           R5 UPON COMPLETION OF 'RTS R5'
                        ;           INSTRUCTION
                        ;2ND PUSH
                        ;   .
                        ;   .           NO. OF PARAMETERS AS A FUNCTION
                        ;   .           OF MEMORY INTERLEAVING
                        ;   .
                        ;NTH PUSH
                        ;NTH +1 PUSH = MARK INSTRUCTION (OTHER PARITY)
                        ;LAST PUSH = OLD PC FROM THE 'JSR' (NORMAL)
                        ;
                        ;NOTE:  THE TEST SHOULD ABORT ON ATTEMPT TO FETCH THE
                        ;       MARK INSTRUCTION (NTH +1 PUSH)
                        ;
                        ;       WHEN THE PARITY ERROR OCCURS THE STACK POINTER IS
                        ;       POSITIONED AT THE NTH +1 PUSH, THUS GIVING,
                        ;
                        ;NTH +2 PUSH = PS FROM PARITY ERROR
                        ;NTH +3 PUSH = PC FROM PARITY ERROR
                        ;
                        ;
                        ;*********************************************************
                        ;*********************************************************
                        ;*********************************************************
                        ;*********************************************************
1584
1585
1586
1587                                    ;*********************************************
1588                                    ;TEST 37        TEST   MARK "POP"   MARK INSTRUCTION
1589                                    ;*********************************************
1590 006174  000004                    T%T137: SCOPE
1591                                    ;               11/45 ***** ROM STATE 260 ****
1592                                    ;
1593                                    ;       11/40 **** ROM STATE 1 ****
1594 006200  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
1595 006202  006326                            E0                              ;ROUTINE ADDRESS
1596 006204  042777  000004  173410            BIC     #BIT2,@PARITY           ;WRITE NORMAL
1597                                    ;*********************************************
1598                                    ;*********************************************
1599                                    ;*********************************************
1600                                    ;*********************************************
1601                                    ;
1602                                    ;
1603                                    ;NOTE THAT THE NEXT INSTRUCTION WILL MOVE THE STACK TO PARITY AREA
1604                                    ;IN THE EVENT THAT NO PARITY EXISTS WHERE THE STACK IS NORMALLY
1605                                    ;SITTING AT 1100
1606                                    ;
1607                                    ;
1608                                    ;*********************************************
```

```
                                        ;*********************************************************************
                                        ;*********************************************************************
                                        ;*********************************************************************
006212  013706  001476                  MOV     @#NEWSTK,SP             ;SET THE STACK IN PARITY
                                                                        ;MEMORY AREA
006216  017702  173224                  MOV     @INTERAD,R2             ;GET THE INTERLEAVE FACTOR
                                                                        ;FOR THIS CONTROLER
006222  005302                          DEC     R2                      ;CALCULATE NO. OF PARAMETERS
                                                                        ;TO BE PUSHED ON THE STACK
006224  010546                          MOV     R5,-(SP)                ;PUSH R5 CONTENTS ON STACK
006226  005702                  1$:     TST     R2                      ;ANY PARAMETERS TO BE
                                                                        ;PUSHED ON THE STACK?
006230  001404                          BEQ     2$                      ;BRANCH IF NO
006232  005302                          DEC     R2                      ;SUBTRACT 1 FROM PARAMETER COUNT
006234  012746  000001                  MOV     #1,-(SP)                ;PUSH PARAMETER ONTO STACK
006240  000772                          BR      1$                      ;GO BACK TO SEE IF ANY MORE!
006242  012777  000004  173358  2$:     MOV     #BIT2,@PARITY           ;WRITE OTHER PARITY
006250  017702  173172                  MOV     @INTERAD,R2             ;GET THE INTERLEAVE FACTOR
                                                                        ;FOR THIS CONTROLLER
006254  005302                          DEC     R2                      ;CALCULATE NO. OF PARAMETERS
                                                                        ;THAT WERE TO BE PUSHED ON THE
                                                                        ;STACK
006256  062702  006400                  ADD     #6400,R2                ;CALCULATE THE CORRESPONDING
                                                                        ;MARK INSTRUCTION
006262  010246                          MOV     R2,-(SP)                ;PUSH MARK INSTRUCTION ON STACK
006264  012777  000004  173330          MOV     #BIT2,@PARITY           ;WRITE NORMAL
006272  010605                          MOV     SP,R5                   ;PLACE MARK INSTRUCTION ADDRESS
                                                                        ;INTO R5 FOR RTS
006274  010602                          MOV     SP,R2                   ;GET THE PC OF THE MARK
                                                                        ;INSTRUCTION
006276  005737  001042                  TST     @#CPU40                 ;ARE WE ON AN 11/40?
006302  001002                          BNE     3$                      ;BRANCH IF YES
                                                                        ;AND DON'T STEP UP THE PC
                                                                        ;IT WON'T BE UPDATED ON THE
                                                                        ;PARITY ABORT
006304  062702  000002                  ADD     #2,R2                   ;STEP UP THE PC
                                                                        ;IT WILL BE UPDATED ON THE
                                                                        ;PARITY ABORT
006310  010237  001352          3$:     MOV     R2,@#SGDDAT             ;STORE THIS VALUE OF PC
                                                                        ;THAT SHOULD BE PUSHED ON THE
                                                                        ;STACK IF A PARITY ABORT OCCURS
006314  004767  000004                  JSR     PC,MRK0                 ;SUBROUTINE CALL
006320  000001  000407          E1:     HLT     +1                      ;DIDN'T ABORT
006322  000407                          BR      .+20                    ;GO TO RESET THE STACK
006324  000203          MRK0:   RTS     R5                              ;RETURN FROM SUBROUTINE
006326  042777  000001  173266  E2:     BIC     #BIT0,@PARITY           ;DISABLE PARITY
006334  004837  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
006340  000003                          HLT     +3                      ;ABORTED INCORRECTLY
006342  013706  001476                  MOV     @#NEWSTK,SP             ;RESET THE STACK
```



```
        ;*********************************************************************
        ;*********************************************************************
        ;
        ;
        ;THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS
        ;
        ;1ST PUSH = ADDRESS OF THE TAG 'VECSET' (OTHER PARITY)
        ;           THIS ADDRESS WOULD BE PLACED IN
        ;           R5 UPON COMPLETION OF 'RTS R5'
        ;2ND PUSH
        ;   .
        ;   .           NO. OF PARAMETERS AS A FUNCTION
        ;   .           OF MEMORY INTERLEAVING
        ;   .
        ;NTH PUSH
        ;NTH +1 PUSH = MARK INSTRUCTION  (NORMAL)
        ;LAST PUSH = OLD PC FROM THE 'JSR' (NORMAL)
        ;
        ;NOTE:  THE TEST SHOULD FAIL ON ATTEMPT TO RESTORE
        ;       R5 CONTENTS (1ST PUSH)
        ;
        ;       WHEN THE PARITY ERROR OCCURS THE STACK POINTER IS
        ;       PROPERLY UPDATED, THUS GIVING,
        ;
        ;1ST PUSH = PS FROM THE PARITY ERROR
        ;2ND PUSH = PC FROM THE PARITY ERROR
        ;
        ;*********************************************************************
        ;*********************************************************************
        ;*********************************************************************
        ;*********************************************************************
006346  012705  011450                  MOV     #VECSET,R5              ;RESTORE THE PARITY VECTOR
                                                                        ;SERVICE ADDRESS SETUP ROUTINE
                                                                        ;ADDRESS
                                ;*********************************************************************
                                ;TEST 43         TEST   OLD REGISTER CONTENTS   MARK INSTRUCTION
                                ;*********************************************************************
006352  000004                  T43T00: SCOPE
                                ;               11/45 4***  ROM STATE 235 ****
                                ;
                                ;       11/40 **** ROM STATE 356 ****
006354  004815                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
006356  006510                          E2                              ;ROUTINE ADDRESS
006360  017702  173062                  MOV     @INTERAD,R2             ;GET THE INTERLEAVE FACTOR
                                                                        ;FOR THIS CONTROLLER
006364  005302                          DEC     R2                      ;CALCULATE NO. OF PARAMETERS
                                                                        ;TO BE PUSHED ON THE STACK
006366  042777  000004  173226          BIC     #BIT2,@PARITY           ;WRITE NORMAL
006374  005737  001042                  TST     @#CPU40                 ;ARE WE ON AN 11/40?
006400  001407                          BEQ     3$                      ;BRANCH IF NO
```

```
                                                                        ;DATA WILL BE TAKEN FIRST AND
                                                                        ;THEN THE PC UPDATED
001712  012700  001376          MOV     #NEWSTK,R0          ;GET THE INITIAL STACK POINT
001716  162700  000002          SUB     #2,R0               ;DROP DOWN 1 WORD ADDRESS
                                ;BECAUSE THE PC WILL BE UPDATED
                                ;FIRST THEN THE DATA TAKEN
001722  010037  001332          MOV     R0,@#$GDDAT         ;STORE THIS VALUE AS THE PC
                                ;THAT SHOULD BE PUSHED ON THE STACK
001726  000403                  BR      4$                  ;IF A PARITY ABORT OCCURS
                                ;CONTINUE WITH TEST
001730  012737  000502  001332 3$:  MOV     #1$,@#$GDDAT        ;STORE THE PC THAT SHOULD
                                ;BE PUSHED ON THE STACK
                                ;IF A PARITY ABORT OCCURS
001736  052777  000004  173166 4$:  BIS     #BIT2,@PARITY       ;WRITE OTHER PARITY
001744  010546                  MOV     R5,-(SP)            ;STORE OLD R5 CONTENTS ON STACK
001746  042777  000004  173156        BIC     #BIT2,@PARITY       ;WRITE NORMAL
001754  005702                  1$:     TST     R2                  ;ANY PARAMETERS TO BE PUSHED
                                ;ON THE STACK?
001756  001404                  BEQ     2$                  ;BRANCH IF NO
001760  005302                  DEC     R2                  ;SUBTRACT 1 FROM PARAMETER COUNT
001762  012746  000001          MOV     #1,-(SP)            ;PUSH PARAMETER ON STACK
001766  000772                  BR      1$                  ;GO BACK TO SEE IF ANY MORE
001770  017702  172762          2$:     MOV     @INTERAD,R2         ;GET THE INTERLEAVE FACTOR
                                ;FOR THIS CONTROLLER
001774  005302                  DEC     R2                  ;CALCULATE NO. OF PARAMETERS
                                ;THAT WERE TO BE PUSHED ON THE
                                ;STACK
001776  062702  000030          ADD     #6400,R2            ;CALCULATE THE CORRESPONDING
                                ;MARK INSTRUCTION
002002  010246                  MOV     R2,-(SP)            ;PUSH MARK INSTRUCTION ON STACK
002004  010605                  MOV     SP,R5               ;PLACE MARK INSTRUCTION ADDRESS
                                ;INTO R5 FOR RTS
002006  004767  000004          JSR     PC,MRK1             ;SUBROUTINE CALL
002012  104001          E1:     HLT     +1                  ;DIDN'T ABORT
002014  000407                  BR      .+20                ;GO TO RESET THE STACK
002016  000205          MRK1:   RTS     R5                  ;RETURN FROM SUBROUTINE
002020  042777  000001  173104 E4:  BIC     #BIT0,@PARITY       ;DISABLE PARITY
002026  004037  011550          JSR     R0,@#CHECKLOC       ;CHECK FOR GOOD ABORT
002032  104003                  HLT     +3                  ;ABORTED INCORRECTLY
002034  013706  001476          MOV     @#NEWSTK,SP         ;RESET THE STACK
002040  012705  011450          MOV     #VECSET,R5          ;RESTORE THE PARITY VECTOR
                                ;SERVICE ADDRESS SETUP ROUTINE
                                ;ADDRESS
                        ;*********************************************************
                        ;TEST 41        TEST   SOB BRANCH   SOB INSTRUCTION
                        ;*********************************************************
002044  000004          TST41:  SCOPE
                        ;
                        ;               11/45 #### ROM STATE 260 ####
                        ;
                        ;       11/40 #### ROM STATE 1 ####
002046  004015                  JSR     R0,(R5)             ;SET UP PARITY VECTOR SERVICE
002050  006440                  F0                          ;ROUTINE ADDRESS
002052  042777  000004  173058        BIC     #BIT2,@PARITY       ;WRITE NORMAL
002060  012700  000004          MOV     #4,R0               ;MOVE A NUMBER >1 TO R0 SO THAT
```



```
                                                                        ;WHEN DECREMENTED BY "SOB" THE
                                                                        ;RESULT WON'T BE 0
002064  011102                  MOV     R1,R2               ;SET UP FOR A DATO
002066  013703  001476          MOV     @#NEWSTK,R3         ;GET THE INITIAL TEST POINT
002072  005737  001042          TST     @#CPU40             ;ARE WE ON AN 11/40?
002076  001405                  BEQ     1$                  ;BRANCH IF NO
002100  062703  000002          ADD     #2,R3               ;STEP UP 1 WORD
                                ;SINCE THE PC WILL NOT BE
                                ;UPDATED ON THE PARITY ABORT
002104  000402                  BR      2$                  ;GO TO STORE THIS VALUE
002106  062703  000004          1$:     ADD     #4,R3               ;STEP UP 2 WORDS
                                ;SINCE THE PC WILL BE UPDATED
                                ;ON THE PARITY ABORT
002112  010337  001332          2$:     MOV     R3,@#$GDDAT         ;STORE THIS VALUE OF PC
                                ;THAT SHOULD BE PUSHED ON THE
                                ;STACK IF A PARITY ABORT OCCURS
002116  052777  000004  173006        BIS     #BIT2,@PARITY       ;WRITE OTHER PARITY
002124  012722  000240          MOV     #240,(R2)+          ;INSTRUCTION TO BE DONE ON
                                ;COMPLETION OF SOB INSTRUCTION
                                ;THE INSTRUCTION WILL BE IN
                                ;PARITY MEMORY AS DESIGNATED BY
                                ;THE CONTENTS OF R1
002130  042777  000004  172774        BIC     #BIT2,@PARITY       ;WRITE NORMAL
002136  012712  000203          MOV     #203,@R2            ;MOVE AN RTS R3 INTO THE LOCATION
                                ;FOLLOWING THE INSTRUCTION TO BE
                                ;EXECUTED IN PARITY MEMORY IN
                                ;CASE IT DOESN'T ABORT
002142  000403                  BR      .+10                ;GO TO SOB INSTRUCTION
002144  004302          F1:     JSR     R3,-(R2)            ;THE INSTRUCTION TO BE DONE ON
                                ;COMPLETION OF SOB EXECUTION IS
                                ;PLACED HERE
002146  104001                  HLT     +1                  ;DIDN'T ABORT
002150  000407                  BR      .+20                ;GO TO RESET THE STACK
002152  077004                  SOB     R0,F                ;EXECUTE THIS INSTRUCTION
002154  042777  000001  172750 F0:  BIC     #BIT0,@PARITY       ;DISABLE PARITY
002162  004037  011550          JSR     R0,@#CHECKLOC       ;CHECK FOR GOOD ABORT
002166  104003                  HLT     +3                  ;ABORTED INCORRECTLY
002170  013706  001476 . .       MOV     @#NEWSTK,SP         ;RESET THE STACK
                        ;*********************************************************
                        ;TEST 42        TEST  SM3,DM0  MOV INSTRUCTION
                        ;*********************************************************
002174  000004          TST42:  SCOPE
                        ;
                        ;               11/45 #### ROM STATE 20 ####
                        ;
                        ;       11/40 #### ROM STATE 1 ####
002176  004015                  JSR     R0,(R5)             ;SET UP PARITY VECTOR SERVICE
002200  002774                  F1                          ;ROUTINE ADDRESS
002202  011100                  MOV     R1,R0               ;SET UP FOR A DATO
002204  017702  172546          MOV     @INTERAD,R2         ;GET THE INTERLEAVE FACTOR
                                ;FOR THIS CONTROLLER
002210  052777  000004  172710        BIS     #BIT2,@PARITY       ;WRITE NORMAL
002216  005702                  1$:     TST     R2                  ;ANY "MOV R0,R0"S TO BE SHOVED
                                ;INTO PARITY MEMORY AREA?
002220  001404                  BEQ     2$                  ;BRANCH IF NO
```

```
1643  006712  005302                          DEC     R2                      ;SUBTRACT 1 FROM INSTRUCTION
1644                                                                          ;COUNT
1645  006714  012725  010025                  MOV     #10*400,(R5)+           ;MOVE THE INSTRUCTION 'MOV R0,R0'
1646                                                                          ;TO PARITY AREA
1647  006720  005772                          BR      1$                      ;GO BACK TO SEE IF ANY MORE!
1648  006722  042777  000004  172672  2$:    BIC     #BIT2,@#PARITY          ;WRITE OTHER PARITY
1649  006730  012720  010000                  MOV     #10400,(R0)+            ;MOVE THE INSTRUCTION 'MOV R0,R0'
1650                                                                          ;INTO THE NEXT WORD LOCATION AFTER
1651                                                                          ;THE PREVIOUS 'MOV R0,R0' INSTRUCTION
1652  006734  042777  000004  172660          BIC     #BIT2,@#PARITY          ;WRITE NORMAL
1653  006742  012710  000003                  MOV     #3,(R0)                 ;MOVE 'RTS R3' INTO NEXT WORD
1654                                                                          ;LOCATION AFTER THE PREVIOUS
1655                                                                          ;'MOV R0,R0' INSTRUCTION IN CASE
1656                                                                          ;A PARITY ABORT DOESN'T OCCUR
1657  006746  005737  001602                  TST     @#CPU40                 ;ARE WE ON AN 11/40?
1660  006752  001402                          BEQ     3$                      ;BRANCH IF NO
1661                                                                          ;SINCE PC WILL BE UPDATED ON THE
1662                                                                          ;PARITY ABORT
1663  006754  192766  000002                  SUB     #2,R6                   ;DROP BACK 1 WORD ADDRESS
1664                                                                          ;SINCE THE PC WILL NOT BE
1665                                                                          ;UPDATED ON THE PARITY ABORT
1666  006760  010637  001332          3$:     MOV     R6,@#SGDDAT             ;STORE THE PC THAT SHOULD
1667                                                                          ;BE PUSHED ON THE STACK
1668                                                                          ;IF A PARITY ABORT OCCURS
1669  006764  004771  000000                  JSR     R3,0(R1)                ;GO TO PARITY MEMORY AREA
1670  006770  104001                          HLT     +1                      ;DIDN'T ABORT
1671  006772  000406                          BR      .+16                    ;GO TO RESET THE STACK
1672  006774  042777  000001  172620  4$:     BIC     #BIT0,@#PARITY          ;DISABLE PARITY
1673  007002  004737  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
1674  007006  104003                          HLT     +3                      ;ABORTED INCORRECTLY
1675  007010  013706  001476                  MOV     @#NEWSTK,SP             ;RESET THE STACK
1676
1677
1678
```

```
        ;*********************************************************************
        ;*********************************************************************
        ;*********************************************************************
        ;*********************************************************************
        ;
        ;THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS:
        ;
        ;1ST PUSH = OLD R3 FROM THE 'JSR' (OTHER PARITY)
        ;
        ;NOTE:  THE TEST SHOULD FAIL ON ATTEMPT TO RESTORE
        ;       R3 CONTENTS (1ST PUSH)
        ;
        ;
        ;*********************************************************************
        ;*********************************************************************
        ;*********************************************************************
        ;*********************************************************************
```



```
1679                                    ;*********************************************************
1680                                    ;TEST 43        TEST   RTS INSTRUCTION
1681                                    ;*********************************************************
1682  007014  000000                    TST43:  $SCOPE
1683                                    ;                       11/45 **** ROM STATE 224 ****
1684                                    ;
1685                                    ;       11/40 **** ROM STATE 325 ****
1686  007016  004015                            JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
1687  007020  007066                            G0                              ;ROUTINE ADDRESS
1688  007022  042777  000004  172572            BIC     #BIT2,@#PARITY          ;WRITE NORMAL
1689  007030  012700  007056                    MOV     #G,R0                   ;SET UP A SUBROUTINE ADDRESS
1690  007034  012737  007052  001332            MOV     #.+16,@#SGDDAT          ;STORE THE PC THAT SHOULD
1691                                                                            ;BE PUSHED ON THE STACK
1692                                                                            ;IF A PARITY ABORT OCCURS
1693  007042  052777  000004  172552            BIS     #BIT2,@#PARITY          ;WRITE OTHER PARITY
1694  007050  004320                            JSR     R3,(R0)+                ;SUBROUTINE CALL
1695  007052  104001                            HLT     +1                      ;DIDN'T ABORT
1696  007054  000412                            BR      .+26                    ;GO TO RESET THE STACK
1697  007056  042777  000004  172536  G:        BIC     #BIT2,@#PARITY          ;WRITE NORMAL
1698  007064  000203                            RTS     R3                      ;RETURN FROM SUBROUTINE
1699  007066  042777  000001  172526  G0:       BIC     #BIT0,@#PARITY          ;DISABLE PARITY
1920  007074  004037  011550                    JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
1921  007130  104003                            HLT     +3                      ;ABORTED INCORRECTLY
1922  007102  013706  001476                    MOV     @#NEWSTK,SP             ;RESET THE STACK
1923
1924
1925
```

```
        ;*********************************************************************
        ;*********************************************************************
        ;*********************************************************************
        ;*********************************************************************
        ;
        ;
        ;THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS:
        ;
        ;1ST PUSH = OLD PS FOR 'RTI' (OTHER PARITY)
        ;2ND PUSH = OLD PC FOR 'RTI' (OTHER PARITY)
        ;
        ;NOTE:  THE TEST SHOULD FAIL ON ATTEMPT TO 'POP'
        ;       THE OLD PC (2ND PUSH)
        ;
        ;       WHEN THE PARITY ERROR OCCURS THE STACK POINTER
        ;       IS POSITIONED AT THE 1ST PUSH, THUS GIVING,
        ;       THE 'NEW' PS FOR THE RTI INSTRUCTION WHICH
        ;       WILL OVERLAY THE OLD PC SET UP FOR THE 'RTI'
        ;       INSTRUCTION (2ND PUSH) THUS GIVING,
        ;
        ;1ST PUSH = OLD PS FOR 'RTI' (OTHER PARITY)
        ;2ND PUSH = NEW PS FROM THE 'RTI'
        ;
        ;3RD PUSH = NEW PC FROM THE 'RTI'
        ;
        ;*********************************************************************
        ;*********************************************************************
```

```
                        ;*********************************************************************
                        ;*********************************************************************
                        ;*********************************************************************
                        ;*********************************************************************
                        ;
                        ;
                        ;*********************************************************************
                        ;TEST 44        TEST   1ST POP   RTI INSTRUCTION
                        ;*********************************************************************
007106  002204          TST44:  SCOPE
                        ;                       11/45 **** ROM STATE 212 ****
                        ;
                        ;       11/40 **** ROM STATE 320 ****
007110  004015                  JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
007112  007150                  H2                              ;ROUTINE ADDRESS
007114  012746  000340          MOV     #340,-(SP)              ;SET PS FOR 'RTI'
007120  012746  007134          MOV     #H1,-(SP)               ;SET RETURN FROM 'RTI'
007124  012737  007134  001332  MOV     #H1,@#$GDVAT            ;STORE THE PC THAT SHOULD
                                                                ;BE PUSHED ON THE STACK
                                                                ;IF A PARITY ABORT OCCURS
007132  000002                  BR      H                       ;GO DO 'RTI'
007134  000001          H1:     HLT     #1                      ;DIDN'T ABORT
007136  000412                  BR      .+26                    ;GO TO RESET THE STACK
007140  042777  000004  172456  H:      BIC     #BIT2,@#PAKITY  ;WRITE NORMAL
007146  000002                  RTI                             ;RETURN FROM INTERRUPT
007150  042777  000001  172446  H0:     BIC     #BIT0,@#PARITY  ;DISABLE PARITY
007156  004737  011550          JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
007162  000003                  HLT     #3                      ;ABORTED INCORRECTLY
007164  013706  001476          MOV     @#NEWSTK,SP             ;RESET THE STACK
                        ;
                        ;
                        ;*********************************************************************
                        ;*********************************************************************
                        ;*********************************************************************
                        ;*********************************************************************
                        ;
                        ;
                        ;THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS:
                        ;
                        ;1ST PUSH = OLD PS FOR 'RTI' (OTHER PARITY)
                        ;2ND PUSH = OLD PC FOR 'RTI' (OTHER PARITY)
                        ;
                        ;NOTE:  THE TEST SHOULD FAIL ON ATTEMPT TO 'POP'
                        ;       THE OLD PS (1ST PUSH)
                        ;
                        ;       THE 2ND PUSH IS REWRITTEN NORMAL BEFORE
                        ;       DOING THE 'RTI'
                        ;
                        ;       WHEN THE PARITY ERROR OCCURS THE STACK
                        ;       POINTER IS PROPERLY UPDATED AND THE NEW
                        ;       PS AND PC FROM THE RTI INSTRUCTION IS
                        ;       PUSHED ONTO THE STACK, THUS GIVING,
                        ;
                        ;1ST PUSH = NEW PS FROM 'RTI' (NORMAL)
                        ;2ND PUSH = NEW PC FROM 'RTI' (NORMAL)
```



```
                        ;
                        ;
                        ;*********************************************************************
                        ;*********************************************************************
                        ;*********************************************************************
                        ;*********************************************************************
                        ;TEST 45        TEST   2ND POP   RTI INSTRUCTION
                        ;*********************************************************************
007170  000004          TST45:  SCOPE
                        ;                       11/45 **** ROM STATE 214 ****
                        ;
                        ;       11/40 **** ROM STATE 322 ****
007172  004015                  JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
007174  007242                  H2                              ;ROUTINE ADDRESS
007176  012746  000340          MOV     #340,-(SP)              ;SET PS FOR 'RTI'
007202  012746  007216          MOV     #H3,-(SP)               ;SET RETURN FROM 'RTI'
007206  012737  007216  001332  MOV     #H3,@#$GDVAT            ;STORE THE PC THAT SHOULD
                                                                ;BE PUSHED ON THE STACK
007214  000402                  BR      H4                      ;IF A PARITY ABORT OCCURS
007216  104001          H3:     HLT     #1                      ;GO DO 'RTI'
007220  000416                  BR      .+36                    ;DIDN'T ABORT
007222  042777  000005  172372  H4:     BIC     #BIT2!BIT0,@#PARITY ;GO TO RESET THE STACK
007230  012646                  MOV     (SP)+,-(SP)             ;WRITE NORMAL AND DISABLE PARITY
                                                                ;WRITE 1ST "POP" NORMAL SO NO
007232  042777  000001  172362  BIS     #BIT0,@#PAKITY          ;ERROR WILL OCCUR
007240  000002                  RTI                             ;ENABLE PARITY
007242  042777  000001  172352  H2:     BIC     #BIT0,@#PARITY  ;RETURN FROM INTERRUPT
007250  004037  011550          JSR     R0,@#CHECKLOC           ;DISABLE PARITY
007254  104003                  HLT     #3                      ;CHECK FOR GOOD ABORT
007256  012706  001100          MOV     #$TACK,SP               ;ABORTED INCORRECTLY
                                                                ;INITIALIZE THE STACK BACK
                                                                ;TO 1100 FOR WE DON'T NEED IT
                                                                ;TO BE IN PARITY MEMORY AREA
                                                                ;ANYMORE
                        ;*********************************************************************
                        ;TEST 46        TEST (DATA)   SM6,DM0   ADD INSTRUCTION
                        ;*********************************************************************
007262  000004          TST46:  SCOPE
                        ;                       11/45 **** ROM STATE 142 ****
                        ;
                        ;       11/40 **** ROM STATE 250 ****
007264  004015                  JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
007266  007320                  L                               ;ROUTINE ADDRESS
007270  011100                  MOV     @R1,R0                  ;SET UP FOR A DATO
007272  010020                  MOV     R0,(R0)+                ;DO THE DATO
007274  012737  007306  001332  MOV     #.+12,@#$GDDAT          ;STORE THE PC THAT SHOULD
                                                                ;BE PUSHED ON THE STACK
                                                                ;IF A PARITY ABORT OCCURS
007302  066000  177776          ADD     -2(R0),R0               ;DO A DATI
007306  042777  000004  172306  BIC     #BIT2,@#PAKITY          ;WRITE NORMAL FOR EMT CALL
007314  104001                  HLT     #1                      ;DIDN'T ABORT
```

```
	007316	000410			BR	.+22		;GO TO NEXT TEST
	007320	042777	000005	172270	L:	BIC	#BIT2!BIT0,@PARITY	;WRITE NORMAL AND DISABLE
	007326	004037	011550			JSR	R0,@#CHECKLOC	;CHECK FOR GOOD ABORT
	007332	104003				HLT	+3		;ABORTED INCORRECTLY
	007334	012706	001100			MOV	#STACK,SP	;RESET THE STACK
						;*************************************************************
						;TEST 47		TEST DM1 TSTB INSTRUCTION
						;*************************************************************
	007340	000004			TST47:	SCOPE
						;		11/45 **** ROM STATE 175 ****
						;
						;	11/40 **** ROM STATE 267 ****
	007342	004015				JSR	R0,(R5)		;SET UP PARITY VECTOR SERVICE
	007344	007374				M			;ROUTINE ADDRESS
	007346	011100				MOV	@R1,R0		;SET UP FOR A DATO
	007350	010010				MOV	R0,(R0)		;DO THE DATO
	007352	012737	007362	001332		MOV	#.+10,@#$DDAT	;STORE THE PC THAT SHOULD
									;BE PUSHED ON THE STACK
									;IF A PARITY ABORT OCCURS
	007360	105710				TSTB	(R0)		;DO A DATI, DATIP
	007362	042777	000004	172232		BIC	#BIT2,@PARITY	;WRITE NORMAL FOR EMT CALL
	007370	104001				HLT	+1		;DIDN'T ABORT
	007372	000410				BR	.+22		;GO TO NEXT TEST
	007374	042777	000005	172220	M:	BIC	#BIT2!BIT0,@PARITY	;WRITE NORMAL AND DISABLE
	007402	004037	011550			JSR	R0,@#CHECKLOC	;CHECK FOR GOOD ABORT
	007406	104003				HLT	+3		;ABORTED INCORRECTLY
	007410	012706	001100			MOV	#STACK,SP	;RESET THE STACK
						;*************************************************************
						;TEST 50		TEST DM3 CLRB INSTRUCTION
						;*************************************************************
	007414	000004			TST50:	SCOPE
						;		11/45 **** ROM STATE 221 ****
						;
						;	11/40 **** ROM STATE 264 ****
	007416	004015				JSR	R0,(R5)		;SET UP PARITY VECTOR SERVICE
	007420	007450				NN			;ROUTINE ADDRESS
	007422	011100				MOV	@R1,R0		;SET UP FOR A DATO
	007424	010010				MOV	R0,(R0)		;DO THE DATO
	007426	012737	007436	001332		MOV	#.+10,@#$DDAT	;STORE THE PC THAT SHOULD
									;BE PUSHED ON THE STACK
									;IF A PARITY ABORT OCCURS
	007434	105030				CLRB	@(R0)+		;DO A DATO
	007436	042777	000004	172156		BIC	#BIT2,@PARITY	;WRITE NORMAL FOR EMT CALL
	007444	104001				HLT	+1		;DIDN'T ABORT
	007446	000410				BR	.+22		;GO TO NEXT TEST
	007450	042777	000005	172144	NN:	BIC	#BIT2!BIT0,@PARITY	;WRITE NORMAL AND DISABLE
	007456	004037	011550			JSR	R0,@#CHECKLOC	;CHECK FOR GOOD ABORT
	007462	104003				HLT	+3		;ABORTED INCORRECTLY
	007464	012706	001100			MOV	#STACK,SP	;RESET THE STACK
						;*************************************************************
						;TEST 51		TEST SM1 SUBTRACT INSTRUCTION
						;*************************************************************
	007470	000004			TST51:	SCOPE
						;		11/45 **** ROM STATE 27 ****
```



```
						;
						;	11/40 **** ROM STATE 255 ****
	007472	004015				JSR	R0,(R5)		;SET UP PARITY VECTOR SERVICE
	007474	007526				P			;ROUTINE ADDRESS
	007476	011100				MOV	@R1,R0		;SET UP FOR A DATO
	007500	010010				MOV	R0,(R0)		;DO THE DATO
	007502	012737	007512	001332		MOV	#.+10,@#$DDAT	;STORE THE PC THAT SHOULD
									;BE PUSHED ON THE STACK
									;IF A PARITY ABORT OCCURS
	007510	161027	177777			SUB	(R0),#177777	;DO A DATI
	007514	042777	000004	172100		BIC	#BIT2,@PARITY	;WRITE NORMAL FOR EMT CALL
	007522	104001				HLT	+1		;DIDN'T ABORT
	007524	000410				BR	.+22		;GO TO NEXT TEST
	007526	042777	000005	172066	P:	BIC	#BIT2!BIT0,@PARITY	;WRITE NORMAL AND DISABLE
	007534	004037	011550			JSR	R0,@#CHECKLOC	;CHECK FOR GOOD ABORT
	007540	104003				HLT	+3		;ABORTED INCORRECTLY
	007542	012706	001100			MOV	#STACK,SP	;RESET THE STACK
						;*************************************************************
						;TEST 52		TEST (DATA) SM5 BISB INSTRUCTION
						;*************************************************************
	007546	000004			TST52:	SCOPE
						;		11/45 **** ROM STATE 177 ****
						;
						;	11/40 **** ROM STATE 245 ****
	007550	004015				JSR	R0,(R5)		;SET UP PARITY VECTOR SERVICE
	007552	007604				R			;ROUTINE ADDRESS
	007554	011100				MOV	@R1,R0		;SET UP FOR A DATO
	007556	010010				MOV	R0,(R0)		;DO THE DATO
	007560	012737	007570	001332		MOV	#.+10,@#$DDAT	;STORE THE PC THAT SHOULD
									;BE PUSHED ON THE STACK
									;IF A PARITY ABORT OCCURS
	007566	155060	000002			BISB	@-(R0),2(R0)	;DO A DATI
	007572	042777	000004	172022		BIC	#BIT2,@PARITY	;WRITE NORMAL FOR EMT CALL
	007600	104001				HLT	+1		;DIDN'T ABORT
	007602	000410				BR	.+22		;GO TO NEXT TEST
	007604	042777	000005	172010	R:	BIC	#BIT2!BIT0,@PARITY	;WRITE NORMAL AND DISABLE
	007612	004037	011550			JSR	R0,@#CHECKLOC	;CHECK FOR GOOD ABORT
	007616	104003				HLT	+3		;ABORTED INCORRECTLY
	007620	012706	001100			MOV	#STACK,SP	;RESET THE STACK
						;*************************************************************
						;TEST 53		NEXT 4 TESTS ARE NON-INTERLEAVE 4K DEPENDENT
						;*************************************************************
	007624	000004			TST53:	SCOPE
	007626	022701	017360			CMP	#17360,R1	;HAVE WE PARITY IN THE LOWER 4K?
									;AND IS THE SELECTED REGISTER
									;GOVERNING THE 4K AREA?
									;THIS COMPARE IS IF THE KT11
									;ISN'T ENABLED DURING PROGRAM
									;EXECUTION
	007632	101005				BHI	2$		;YES - PROCEED TO NEXT TESTS
									;WHICH DEPEND ON PARITY IN LOWER 4K
	007634	022777	000140	171554		CMP	#140,@SSEIAD	;THE ABOVE COMPARE DIDN'T
									;CHECK - DO THIS COMPARE TO
									;SEE IF IT WAS BECAUSE THE
```

```
                                                                ;TABLE WAS CREATED WITH MEMORY
                                                                ;MANAGEMENT TURNED ON
007640  001402                          BEQ     2$              ;PROCEED TO NEXT TESTS IF THIS
                                                                ;COMPARE CHECKS
007644  000404                          BR      4$              ;THE REGISTER UNDER TEST IS NOT
                                                                ;CONTROLLING THE LOWER 4K!!!
                                                                ;GO TO REEVALUATE $TSTNM AND
                                                                ;JUMP OVER THE (4) 4K DEPENDENT
                                                                ;TESTS
007646  022777  000001  171572  2$:     CMP     #1,@NTERAD      ;IS THIS CONTROLLER
                                                                ;INTERLEAVED??
007654  001405                          BEQ     3$              ;BRANCH IF NO
007656  062767  000004  171016  4$:     ADD     #4,$TSTNM       ;SET $TSTNM TO PROPER VALUE
                                                                ;SINCE THE NEXT 4 TESTS WILL
                                                                ;BE SKIPPED
007664  000137  010426                  JMP     @#RED+26        ;JUMP TO 1ST INDEX WORD TEST
007670  005237  001020          3$:     INC     @#PSPCORZONES   ;SET FLAG INDICATING TO 'CHECKLOC'
                                                                ;ROUTINE THAT PS AND PC FETCH
                                                                ;AND RED & YELLOW ZONE AREAS
                                                                ;ARE GOING TO BE TESTED

        ;**********************************************************************
        ;**********************************************************************
        ;**********************************************************************
        ;**********************************************************************
        ;
        ;
        ;THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS:
        ;
        ;1ST PUSH - OLD PS FROM ERROR TRAP (NORMAL)
        ;2ND PUSH - OLD PC FROM ERROR TRAP (NORMAL)
        ;
        ;NOTE:  THE TEST SHOULD FAIL ON ATTEMPT TO FETCH THE NEW PS
        ;
        ;       WHEN THE PARITY ERROR OCCURS THE STACK POINTER IS
        ;       ALTERED FROM THE ORIGINAL ERROR TRAP, THUS GIVING,
        ;
        ;1ST PUSH - OLD PS FROM ERROR TRAP (NORMAL)
        ;2ND PUSH - NEW PS FROM THE PARITY ERROR
        ;
        ;3RD PUSH - NEW PC FROM THE PARITY ERROR
        ;
        ;**********************************************************************
        ;**********************************************************************
        ;**********************************************************************
        ;**********************************************************************

                                        ;**********************************************************************
                                        ;TEST 54        TEST NEW 'PS' FETCH
                                        ;**********************************************************************
007674  000004                          TST54:  SCOPE
```



```
                                ;               11/45 **** ROM STATE 357 ****
                                ;
                                ;       11/40 **** ROM STATE 115 ****
007676  004015                          JSR     R0,(R5)         ;SET UP PARITY VECTOR SERVICE
007700  007754                          T0                      ;ROUTINE ADDRESS
007702  042777  000004  171712          BIC     #BIT2,@PARITY   ;WRITE NORMAL
007710  012737  007750  000010          MOV     #T,@#RESVEC     ;RESERVED INSTRUCTION TIMEOUT
                                                                ;VECTOR ADDRESS
007716  012737  007750  001332          MOV     #1,@#$GODAT     ;STORE THE PC THAT SHOULD
                                                                ;BE PUSHED ON THE STACK
                                                                ;IF A PARITY ABORT OCCURS
007724  052777  000004  171670          BIS     #BIT2,@PARITY   ;WRITE OTHER PARITY
007732  012737  000340  000012          MOV     #340,@#RESVEC+2 ;NEW 'PS' FOR ERROR TRAP
007740  042777  000004  171654          BIC     #BIT2,@PARITY   ;WRITE NORMAL
007746  007000                          7000                    ;NON-RECOGNIZABLE OP-CODE
                                                                ;SHOULD ATTEMPT TO TRAP TO 'T:'
007750  104001                  T:      HLT     +1              ;DIDN'T ABORT
007752  000406                          BR      .+16            ;GO TO RESET THE STACK
007754  042777  000001  171640  T0:     BIC     #BIT0,@PARITY   ;DISABLE PARITY
007762  004037  011550                  JSR     R0,@#CHECKLOC   ;CHECK FOR GOOD ABORT
007766  104003                          HLT     +3              ;ABORTED INCORRECTLY
007770  012706  001100                  MOV     #STACK,SP       ;RESET THE STACK

        ;**********************************************************************
        ;**********************************************************************
        ;**********************************************************************
        ;**********************************************************************
        ;
        ;
        ;THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS:
        ;
        ;1ST PUSH - OLD PS FROM ERROR TRAP (NORMAL)
        ;2ND PUSH - OLD PC FROM ERROR TRAP (NORMAL)
        ;
        ;NOTE:  THE TEST SHOULD FAIL ON ATTEMPT TO FETCH THE NEW PC
        ;
        ;       WHEN THE PARITY ERROR OCCURS THE STACK POINTER IS
        ;       NOT ALTERED FROM THE ORIGINAL ERROR TRAP AND THE
        ;       PC FOR THE PARITY ERROR IS THE OLD PS, THUS GIVING,
        ;
        ;1ST PUSH - OLD PS FROM ERROR TRAP (NORMAL)
        ;2ND PUSH - OLD PS FROM ERROR TRAP (NORMAL)
        ;
        ;IN OTHER WORDS THE ORIGINAL ERROR TRAP AND VECTOR IS LOST!!!
        ;
        ;
        ;**********************************************************************
        ;**********************************************************************
        ;**********************************************************************
        ;**********************************************************************
```

```
                                          ;*************************************************************
                                          ;TEST 55          TEST NEW 'PC' FETCH
                                          ;*************************************************************
007774  000004                            TST55:  SCOPE
                                          ;                 11/45 **** ROM STATE 355 ****
                                          ;
                                          ;         11/40 **** ROM STATE 333 ****
007776  004015                                    JSR     R0,(R5)              ;SET UP PARITY VECTOR SERVICE
010000  010056                                    V0                           ;ROUTINE ADDRESS
010002  012737  010032  000010                    MOV     #V,@#RESVEC          ;RESERVED INSTRUCTION TIMEOUT
                                                                               ;VECTOR ADDRESS
010010  012737  000340  000012                    MOV     #340,@#RESVEC+2      ;NEW 'PS' FOR ERROR TRAP
010016  005737  001642                            TST     @#CPU40              ;ARE WE ON AN 11/40?
010022  001404                                    BEQ     1$                   ;BRANCH IF NO
010024  012737  010052  001332                    MOV     #V,@#SGDDAT          ;STORE THIS VALUE OF THE PC THAT
                                                                               ;SHOULD BE PUSHED ON THE STACK
                                                                               ;IF A PARITY ABORT OCCURS
                                                                               ;SINCE THE PC IS UPDATED FIRST
                                                                               ;THEN THE VECTOR DATA TAKEN
010032  000403                                    BR      2$                   ;CONTINUE WITH TEST
010034  012737  000010  001332  1$:               MOV     #10,@#SGDDAT         ;STORE THE PC THAT SHOULD
                                                                               ;BE PUSHED ON THE STACK
                                                                               ;IF A PARITY ABORT OCCURS
010042  042777  000004  171552  2$:               BIC     #BIT2,@PARITY        ;WRITE NORMAL
010050  007000                                    7000                         ;NON-RECOGNIZABLE OP-CODE
                                                                               ;SHOULD ATTEMPT TO TRAP TO 'V1'
010052  104001                            V:      HLT     +1                   ;DIDN'T ABORT
010054  000406                                    BR      .+16                 ;GO TO RESET THE STACK
010056  042777  000001  171536  V0:               BIC     #BIT0,@PARITY        ;DISABLE PARITY
010064  004037  011550                            JSR     R0,@#CHECKLOC        ;CHECK FOR GOOD ABORT
010070  104003                                    HLT     +3                   ;ABORTED INCORRECTLY
010072  012706  001100                            MOV     #STACK,SP            ;RESET THE STACK

          ;*************************************************************
          ;*************************************************************
          ;*************************************************************
          ;*************************************************************
          ;
          ;
          ;THE FOLLOWING TEST WILL/SHOULD CAUSE A PARITY ABORT IN THE
          ;'YELLOW' ZONE.  THE 'YELLOW' ZONE IS AN AREA UP TO A 16 WORD
          ;LOCATION LIMIT BEYOND THE STACK OVERFLOW BOUNDARY OF 400
          ;(OCTAL).  I.E. LOCATIONS 376 - 340 COMPRISE THE YELLOW ZONE.
          ;
          ;SINCE PARITY ERRORS HAVE HIGHEST PRIORITY WE WILL BE LOOKING
          ;FOR THE PARITY ABORT TO OCCUR BEFORE THE STACK VIOLATION
          ;TRAP TO LOCATION 4.
          ;
          ;
          ;THE CONTENTS OF THE STACK AFTER EXECUTION OF THE NEXT TEST
          ;SHOULD BE AS FOLLOWS:
          ;
```



```
          ;1ST PUSH - PS FROM THE PARITY ERROR
          ;2ND PUSH - PC FROM THE PARITY ERROR
          ;3RD PUSH - PS FROM THE STACK VIOLATION
          ;4TH PUSH - PC FROM THE STACK VIOLATION
          ;
          ;NOTE:  THE ABOVE CONTENTS WILL EXIST ON THE STACK IF BOTH
          ;       THE PARITY ERROR AND THE STACK VIOLATION WERE
          ;       RECOGNIZED AND THE PARITY ERROR TOOK PRECEDENCE!
          ;
          ;
          ;
          ;*************************************************************
          ;*************************************************************
          ;*************************************************************
          ;*************************************************************

                                          ;*************************************************************
                                          ;TEST 56          TEST ABORT IN 'YELLOW' ZONE
                                          ;*************************************************************
010076  000004                            TST56:  SCOPE
                                          ;                 11/45 **** ROM STATE 177 ****
                                          ;
                                          ;         11/40 **** ROM STATE 267 ****
010100  004015                                    JSR     R0,(R5)              ;SET UP PARITY VECTOR SERVICE
010102  010162                                    V1                           ;ROUTINE ADDRESS
010104  005037  000340                            CLR     @#340                ;CLEAR BOTTOM LIMIT LOCATION
                                                                               ;OF 'YELLOW' ZONE USING 'OTHER'
                                                                               ;PARITY
010110  042777  000004  171504                    BIC     #BIT2,@PARITY        ;WRITE NORMAL
010116  012706  000376                            MOV     #376,SP              ;SET STACK IN 'YELLOW' AREA
010122  013727  000004                            MOV     @#4,(PC)+            ;SAVE CONTENTS OF LOC. 4 IN
                                                                               ;NEXT LOCATION
010126  000000                            SAVLOC4:        .WORD   0            ;ORIGINAL CONTENTS OF LOC. 4
                                                                               ;GO HERE
010130  012737  010166  000004                    MOV     #V2,@#ERRVEC         ;SET UP A TIMEOUT VECTOR SERVICE
                                                                               ;ROUTINE ADDRESS FOR STACK VIO-
                                                                               ;LATION
010136  012737  000340  000006                    MOV     #340,@#ERRVEC+2      ;NEW PS ON TIMEOUT TRAP
010144  012737  010156  001332                    MOV     #.+12,@#SGDDAT       ;STORE THE PC THAT SHOULD
                                                                               ;BE PUSHED ON THE STACK
                                                                               ;IF A PARITY ABORT OCCURS
010152  005766  177742                            TST     -36(SP)              ;REFERENCE BOTTOM LIMIT OF
                                                                               ;'YELLOW' ZONE USING REGISTER 6
                                                                               ;THIS INSTRUCTION SHOULD CAUSE
                                                                               ;AN ABORT 1ST, THEN A STACK
                                                                               ;VIOLATION
010156  104010                                    HLT     +10                  ;DIDN'T ABORT OR RECOGNIZE THE
                                                                               ;STACK VIOLATION
010160  000421                                    BR      YELLOW               ;GO TO RESET STACK AND RESTORE
                                                                               ;TIMEOUT VECTORS
010162  104011                            V1:     HLT     +11                  ;ABORTED BUT STACK VIOLATION
                                                                               ;NOT RECOGNIZED
010164  000417                                    BR      YELLOW               ;GO TO RESET STACK AND RESTORE
                                                                               ;TIMEOUT VECTORS
```

```
010166  042737  000001  171006  V4:   BIC     #BIT0,@PARITY           ;DISABLE PARITY
010174  005737  000372                TST     @#372                   ;STACK VIOLATION PICKED UP -
                                                                      ;WAS THE PARITY ABORT?
010200  001002                        BNE     1$                      ;BRANCH IF YES
010202  104012                        HLT     +12                     ;STACK VIOLATION PICKED UP
                                                                      ;BUT ABORT NOT RECOGNIZED
010204  000407                        BR      YELLOW                  ;GO TO RESET STACK AND RESTORE
                                                                      ;TIMEOUT VECTORS
010206  012706  001100          1$:   MOV     #STACK,SP               ;RESET STACK BACK TO NORMAL
010212  013746  000372                MOV     @#372,-(SP)             ;PUSH ABORT PC ONTO NORMAL
                                                                      ;STACK OUT OF VIOLATION AREAS
                                                                      ;FOR CHECKING PURPOSES
010216  004037  011550                JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
010222  104003                        HLT     +3                      ;ABORTED INCORRECTLY
010224  012706  001100          YELLOW: MOV   #STACK,SP               ;RESET STACK BACK TO NORMAL
010230  013737  010176  000004        MOV     @#SAVLOC4,@#4           ;RESTORE CONTENTS OF LOC. 4
010236  005037  000006                CLR     @#6                     ;RESTORE CONTENTS OF LOC. 6
010242  005037  000372                CLR     @#372                   ;RESET TRAPCATCHER LOCATION
                                                                      ;FOR NEXT TEST
010246  005037  000340                CLR     @#340                   ;CLEAR BOTTOM LIMIT LOCATION
                                                                      ;OF "YELLOW" ZONE USING NORMAL
                                                                      ;PARITY

        ;*****************************************************************
        ;*****************************************************************
        ;*****************************************************************
        ;*****************************************************************
        ;
        ;
        ;THE FOLLOWING TEST WILL/SHOULD CAUSE A PARITY ABORT IN THE
        ;"RED" ZONE.  THE "RED" ZONE IS THE AREA BEYOND THE "YELLOW"
        ;ZONE DESCRIBED IN THE ABOVE TEST.  I.E. LOCATIONS 336 ON
        ;DOWN COMPRISE THE "RED" ZONE
        ;
        ;SINCE PARITY ERRORS HAVE HIGHEST PRIORITY WE WILL BE LOOKING
        ;FOR THE PARITY ABORT TO OCCUR BEFORE THE STACK VIOLATION
        ;TRAP TO LOCATION 4.
        ;
        ;
        ;THE CONTENTS OF THE STACK AFTER EXECUTION OF THE NEXT TEST
        ;SHOULD BE AS FOLLOWS:
        ;
        ;LOC. 0 - PC FROM STACK VIOLATION
        ;LOC. 2 - PS FROM STACK VIOLATION
        ;
        ;NOTE:  THE PS AND PC FROM THE STACK VIOLATION ARE IN LOCS. 0 &2
        ;       BECAUSE THE STACK POINTER (R6) IS REPOSITIONED TO LOC. 4
        ;       (BY HARDWARE)!!
        ;
```



```
        ;LOC. 374 - PS FROM PARITY ABORT
        ;LOC. 372 - PC FROM PARITY ABORT
        ;
        ;NOTE:  THE ABOVE CONTENTS WILL EXIST IN THE 2 DIFFERENT STACK
        ;       AREAS IF BOTH THE PARITY ERROR AND STACK VIOLATION WERE
        ;       RECOGNIZED.  THE TEST BELOW WILL DIFFERENTIATE BETWEEN
        ;       WHICH OCCURRED FIRST.
        ;
        ;
        ;*****************************************************************
        ;*****************************************************************
        ;*****************************************************************
        ;*****************************************************************

                                      ;*************************************************************
                                      ;TEST 57         TEST ABORT IN "RED" ZONE
                                      ;*************************************************************
010252  000004                        TST57:  SCOPE
                                      ;               11/45 **** ROM STATE 177 ****
                                      ;
                                      ;       11/40 **** ROM STATE 207 ****
010254  004015                                JSR     R0,(R5)         ;SET UP A PARITY VECTOR SERVICE
010256  010336                                V3                      ;ROUTINE ADDRESS
010260  005037  000336                        CLR     @#336           ;CLEAR 1ST LOCATION OF "RED"
                                                                      ;ZONE USING "OTHER" PARITY
010264  042777  000004  171330                BIC     #BIT2,@PARITY   ;WRITE NORMAL
010272  012706  000376                        MOV     #376,SP         ;SET STACK IN "YELLOW"AREA
010276  013727  000004                        MOV     @#4,(PC)+       ;SAVE CONTENTS OF LOC. 4
                                                                      ;IN NEXT LOCATION
010302  000000                  HOLDLOC4:     .WORD   0               ;ORIGINAL CONTENTS OF LOC. 4
                                                                      ;GO HERE
010304  012737  010342  000004                MOV     #V4,@#ERRVEC    ;SET UP A TIMEOUT VECTOR SERVICE
                                                                      ;ROUTINE ADDRESS FOR STACK
                                                                      ;VIOLATION
010312  012737  000340  000006                MOV     #340,@#ERRVEC+2 ;NEW PS ON TIMEOUT TRAP
010320  012737  010332  001332                MOV     #.+12,@#SUDDAT  ;STORE THE PC THAT SHOULD
                                                                      ;BE PUSHED ON THE STACK
                                                                      ;IF A PARITY ABORT OCCURS
010326  005766  177740                        TST     -40(SP)         ;REFERENCE 1ST LOCATION OF "RED"
                                                                      ;ZONE USING REGISTER 6
                                                                      ;THIS INSTRUCTION SHOULD CAUSE
                                                                      ;AN ABORT 1ST, THEN A STACK
                                                                      ;VIOLATION
010332  104010                                HLT     +10             ;DIDN'T ABORT OR RECOGNIZE THE
                                                                      ;STACK VIOLATION
010334  000421                                BR      RED             ;GO TO RESET STACK AND RESTORE
                                                                      ;TIMEOUT VECTORS
010336  104011                  V3:           HLT     +11             ;ABORTED BUT STACK VIOLATION
                                                                      ;NOT RECOGNIZED
010340  000417                                BR      RED             ;GO TO RESET STACK AND RESTORE
                                                                      ;TIMEOUT VECTORS
010342  042777  000001  171252  V4:           BIC     #BIT0,@PARITY   ;DISABLE PARITY
010350  005737  000372                        TST     @#372           ;STACK VIOLATION PICKED UP -
```

```
                                                                ;WAS THE PARITY ABORT?
        010354  001002                    BNE     1$            ;BRANCH IF YES
        010356  104012                    HLT     +12           ;STACK VIOLATION PICKED UP BUT
                                                                ;ABORT NOT RECOGNIZED
        010360  000407                    BR      RED           ;GO TO RESET STACK AND RESTORE
                                                                ;TIMEOUT VECTORS
        010362  012706  001100    1$:     MOV     #STACK,SP     ;RESET STACK BACK TO NORMAL
        010366  013746  000372            MOV     @#372,-(SP)   ;PUSH ABORT PC ONTO KERNAL
                                                                ;STACK OUT OF VIOLATION AREAS
                                                                ;FOR CHECKING PURPOSES
        010372  004037  011550            JSR     R0,@#CHECKLOC ;CHECK FOR GOOD ABORT
        010376  104003                    HLT     +3            ;ABORTED INCORRECTLY
        010400  012706  001100    RED:    MOV     #STACK,SP     ;RESET STACK BACK TO NORMAL
        010404  013737  010332  000004    MOV     @#HOLDLOC4,@#4 ;RESTORE CONTENTS OF LOC. 4
        010412  005037  000006            CLR     @#6           ;RESTORE CONTENTS OF LOC. 6
        010416  005037  000372            CLR     @#372         ;RESET TRAPCATCHER LOCATION
                                                                ;FOR NEXT TEST
        010422  005037  000336            CLR     @#336         ;CLEAR 1ST LOCATION OF "RED"
                                                                ;ZONE USING NORMAL PARITY
                          ;*********************************************************************
                          ;TEST 60        TEST (INDEX WORD)  SM0,DM7  MOV INSTRUCTION
                          ;*********************************************************************
        010426  000004    TST60:  SCOPE
                          ;               11/45 **** ROM STATE 7 ****
                          ;
                          ;               11/40 **** ROM STATE 206 ****
        010430  005037  001624            CLR     @#PSPCORZONES ;CLEAR PS, PC OR ZONES
                                                                ;ABORT FLAG
        010434  004015                    JSR     R0,(R5)       ;SET UP PARITY VECTOR SERVICE
        010436  010514                    W                     ;ROUTINE ADDRESS
        010440  042777  000004  171154    BIC     #BIT2,@PARITY ;WRITE NORMAL
        010446  010100                    MOV     R1,R0         ;SET UP FOR A DATO
        010450  012720  010070            MOV     #10070,(R0)+  ;MOVE THE INSTRUCTION
                                                                ;"MOV R0,@-2(R0)" TO PARITY
                                                                ;MEMORY AREA
        010454  052777  000004  171140    BIS     #BIT2,@PARITY ;WRITE OTHER PARITY
        010462  012720  177776            MOV     #-2,(R0)+     ;WRITE THE INDEX WORD IN NEXT
                                                                ;PARITY MEMORY AREA LOCATION
        010466  042777  000004  171126    BIC     #BIT2,@PARITY ;WRITE NORMAL
        010474  010037  001332            MOV     R0,@#SGDDAT   ;STORE THE PC THAT SHOULD
                                                                ;BE PUSHED ON THE STACK
                                                                ;IF A PARITY ABORT OCCURS
        010500  012710  000203            MOV     #203,(R0)     ;MOVE "RTS R3" INTO NEXT
                                                                ;PARITY MEMORY AREA LOCATION
                                                                ;IN CASE WE DON'T ABORT
        010504  004360  177774            JSR     R3,-4(R0)     ;GO TO PARITY MEMORY AREA
        010510  104001                    HLT     +1            ;DIDN'T ABORT
        010512  000410                    BR      .+22          ;GO TO NEXT TEST
        010514  042777  000001  171100  W:  BIC   #BIT0,@PARITY ;DISABLE PARITY
        010522  004037  011550            JSR     R0,@#CHECKLOC ;CHECK FOR GOOD ABORT
        010526  104003                    HLT     +3            ;ABORTED INCORRECTLY
        010530  012706  001100            MOV     #STACK,SP     ;RESET THE STACK
                          ;*********************************************************************
                          ;TEST 61        TEST (INDEX WORD)  SM5,DM6  CMP INSTRUCTION
```



```
                          ;*********************************************************************
        010534  000004    TST61:  SCOPE
                          ;               11/45 **** ROM STATE 26 ****
                          ;
                          ;               11/40 **** ROM STATE 241 ****
        010536  004015                    JSR     R0,(R5)       ;SET UP PARITY VECTOR SERVICE
        010540  010616                    XX                    ;ROUTINE ADDRESS
        010542  042777  000004  171052    BIC     #BIT2,@PARITY ;WRITE NORMAL
        010550  010100                    MOV     R1,R0         ;SET UP FOR A DATO
        010552  012720  026000            MOV     #26000,(R0)+  ;MOVE THE INSTRUCTION
                                                                ;"CMP -2(R0),R0" TO PARITY
                                                                ;MEMORY AREA
        010556  052777  000004  171036    BIS     #BIT2,@PARITY ;WRITE OTHER PARITY
        010564  012720  177776            MOV     #-2,(R0)+     ;WRITE THE INDEX WORD IN NEXT
                                                                ;PARITY MEMORY AREA LOCATION
        010570  042777  000004  171024    BIC     #BIT2,@PARITY ;WRITE NORMAL
        010576  010037  001332            MOV     R0,@#SGDDAT   ;STORE THE PC THAT SHOULD
                                                                ;BE PUSHED ON THE STACK
                                                                ;IF A PARITY ABORT OCCURS
        010602  012710  000203            MOV     #203,(R0)     ;MOVE "RTS R3" INTO NEXT
                                                                ;PARITY MEMORY AREA LOCATION
                                                                ;IN CASE WE DON'T ABORT
        010606  004360  177774            JSR     R3,-4(R0)     ;GO TO PARITY MEMORY AREA
        010612  104001                    HLT     +1            ;DIDN'T ABORT
        010614  000410                    BR      .+22          ;GO TO NEXT TEST
        010616  042777  000001  170776  XX:  BIC  #BIT0,@PARITY ;DISABLE PARITY
        010624  004037  011550            JSR     R0,@#CHECKLOC ;CHECK FOR GOOD ABORT
        010630  104003                    HLT     +3            ;ABORTED INCORRECTLY
        010632  012706  001100            MOV     #STACK,SP     ;RESET THE STACK
                          ;*********************************************************************
                          ;TEST 62        TEST (INDEX WORD)  SM0,DM6  MOV INSTRUCTION
                          ;*********************************************************************
        010636  000004    TST62:  SCOPE
                          ;               11/45 **** ROM STATE 6 ****
                          ;
                          ;               11/40 **** ROM STATE 206 ****
        010640  004015                    JSR     R0,(R5)       ;SET UP PARITY VECTOR SERVICE
        010642  010720                    Y                     ;ROUTINE ADDRESS
        010644  042777  000004  170750    BIC     #BIT2,@PARITY ;WRITE NORMAL
        010652  010100                    MOV     R1,R0         ;SET UP FOR A DATO
        010654  012720  010060            MOV     #10060,(R0)+  ;MOVE THE INSTRUCTION
                                                                ;"MOV R0,-2(R0)" TO PARITY
                                                                ;MEMORY AREA
        010660  052777  000004  170734    BIS     #BIT2,@PARITY ;WRITE OTHER PARITY
        010666  012720  177776            MOV     #-2,(R0)+     ;WRITE THE INDEX WORD IN NEXT
                                                                ;PARITY MEMORY AREA LOCATION
        010672  042777  000004  170722    BIC     #BIT2,@PARITY ;WRITE NORMAL
        010700  010037  001332            MOV     R0,@#SGDDAT   ;STORE THE PC THAT SHOULD
                                                                ;BE PUSHED ON THE STACK
                                                                ;IF A PARITY ABORT OCCURS
        010704  012710  000203            MOV     #203,(R0)     ;MOVE "RTS R3" INTO NEXT
                                                                ;PARITY MEMORY AREA LOCATION
                                                                ;IN CASE WE DON'T ABORT
        010710  004360  177774            JSR     R3,-4(R0)     ;GO TO PARITY MEMORY AREA
```

```
       010716  104001                                  HLT     +1                      ;DIDN'T ABORT
       010720  000410                                  BR      .+22                    ;GO TO NEXT TEST
       010722  042777  000001  170574  Y:              BIC     #BIT0,@PARITY           ;DISABLE PARITY
       010730  004037  011550                          JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
       010734  104003                                  HLT     +3                      ;ABORTED INCORRECTLY
       010736  012706  001100                          MOV     #STACK,SP               ;RESET THE STACK
                                               ;*****************************************************************
                                               ;TEST 63         TEST    BPL INSTRUCTION (-BCOK)
                                               ;*****************************************************************
       010740  000004                          TST63:  SCOPE
                                               ;               11/45 **** ROM STATE 321 ****
                                               ;
                                               ;               11/40 **** ROM STATE 1 ****
       010742  004015                                  JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
       010744  011056                                  Z0                              ;ROUTINE ADDRESS
       010746  042777  000004  170666                  BIC     #BIT2,@PARITY           ;WRITE NORMAL
       010754  010102                                  MOV     R1,R2                   ;SET UP FOR A DATO
       010756  162702  000004                          SUB     #4,R2                   ;CALCULATE THE START
                                                                                       ;ADDRESS FOR THIS TEST
       010762  012722  012700                          MOV     #12700,(R2)+            ;MOVE THE INSTRUCTION
                                                                                       ;'MOV #-1,R0' TO PARITY
                                                                                       ;MEMORY AREA
       010766  012722  177777                          MOV     #-1,(R2)+               ;MOVE A -1 INTO NEXT
                                                                                       ;PARITY MEMORY AREA LOCATION
       010772  012722  100001                          MOV     #100001,(R2)+           ;MOVE THE INSTRUCTION
                                                                                       ;'BPL .+4' INTO NEXT PARITY
                                                                                       ;MEMORY AREA LOCATION
       010776  052777  000004  170516                  BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
       011004  005737  001642                          TST     @#CPU40                 ;ARE WE ON AN 11/40?
       011010  001405                                  BEQ     2$                      ;BRANCH IF NO
                                                                                       ;SINCE THE PC WILL BE UPDATED
                                                                                       ;ON THE PARITY ABORT
       011012  010237  001332                          MOV     R2,@#$GDDAT             ;STORE THAS PC THAT SHOULD
                                                                                       ;BE PUSHEW ON THE STACK
                                                                                       ;IF A PARITY ABORT OCCURS
                                                                                       ;SINCE THE PC IS NOT UPDATED
                                                                                       ;ON THE PARITY ABORT
       011016  012722  000240                          MOV     #240,(R2)+              ;MOVE A 'NOP' INTO NEXT
                                                                                       ;PARITY MEMORY AREA LOCATION
       011022  000404                                  BR      1$                      ;CONTINUE WITH TEST
       011024  012722  000240          2$:             MOV     #240,(R2)+              ;MOVE A 'NOP' INTO NEXT
                                                                                       ;PARITY MEMORY AREA LOCATION
       011030  010237  001332                          MOV     R2,@#$GDDAT             ;STORE THE PC THAT SHOULD
                                                                                       ;BE PUSHEW ON THE STACK
                                                                                       ;IF A PARITY ABORT OCCURS
       011034  042777  000004  170560  1$:             BIC     #BIT2,@PARITY           ;WRITE NORMAL
       011042  012712  000203                          MOV     #203,(R2)               ;MOVE 'RTS R3' INTO NEXT
                                                                                       ;PARITY MEMORY AREA LOCATION
                                                                                       ;IN CASE WE DON'T ABORT
       011046  004362  177710                          JSR     R3,-10(R2)              ;GO TO PARITY MEMORY AREA
       011052  104001                                  HLT     +1                      ;DIDN'T ABORT
       011054  000410                                  BR      .+22                    ;GO TO NEXT TEST
       011056  042777  000001  170536  Z0:             BIC     #BIT0,@PARITY           ;DISABLE PARITY
       011064  004037  011550                          JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
```



```
       011070  104003                                  HLT     +3                      ;ABORTED INCORRECTLY
       011072  012706  001100                          MOV     #STACK,SP               ;RESET THE STACK
                                               ;*****************************************************************
                                               ;TEST 64         TEST    BNE INSTRUCTION   (-BCOK)
                                               ;*****************************************************************
       011076  000004                          TST64:  SCOPE
                                               ;               11/45 **** ROM STATE 322 ****
                                               ;
                                               ;               11/40 **** ROM STATE 1 ****
       011100  004015                                  JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
       011102  011214                                  Z1                              ;ROUTINE ADDRESS
       011104  042777  000004  170510                  BIC     #BIT2,@PARITY           ;WRITE NORMAL
       011112  010102                                  MOV     R1,R2                   ;SET UP FOR A DATO
       011114  162702  000004                          SUB     #4,R2                   ;CALCULATE THE START ADDRESS
                                                                                       ;FOR THIS TEST
       011120  012722  012700                          MOV     #12700,(R2)+            ;MOVE THE INSTRUCTION
                                                                                       ;'MOV #0,R0' INTO PARITY
                                                                                       ;MEMORY AREA
       011124  012722  000000                          MOV     #0,(R2)+                ;MOVE A 0 INTO NEXT PARITY
                                                                                       ;MEMORY AREA LOCATION
       011130  012722  001001                          MOV     #1001,(R2)+             ;MOVE THE INSTRUCTION
                                                                                       ;'BNE .+4' INTO NEXT
                                                                                       ;PARITY MEMORY AREA LOCATION
       011134  052777  000004  170460                  BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
       011142  005737  001642                          TST     @#CPU40                 ;ARE WE ON AN 11/40?
       011146  001405                                  BEQ     2$                      ;BRANCH IF NO
                                                                                       ;SINCE THE PC WILL BE UPDATED
                                                                                       ;ON THE PARITY ABORT
       011150  010237  001332                          MOV     R2,@#$GDDAT             ;STORE THIS PC THAT SHOULD
                                                                                       ;BE PUSHEW ON THE STACK
                                                                                       ;IF A PARITY ABORT OCCURS
                                                                                       ;SINCE THE PC WILL NOT BE
                                                                                       ;UPDATED ON THE PARITY ABORT
       011154  012722  000240                          MOV     #240,(R2)+              ;MOVE A 'NOP' INTO NEXT
                                                                                       ;PARITY MEMORY AREA LOCATION
       011160  000404                                  BR      1$                      ;CONTINUE WITH TEST
       011162  012722  000240          2$:             MOV     #240,(R2)+              ;MOVE A 'NOP' INTO NEXT
                                                                                       ;PARITY MEMORY AREA LOCATION
       011166  010237  001332                          MOV     R2,@#$GDDAT             ;STORE THE PC THAT SHOULD
                                                                                       ;BE PUSHEW ON THE STACK
                                                                                       ;IF A PARITY ABORT OCCURS
       011172  042777  000004  170422  1$:             BIC     #BIT2,@PARITY           ;WRITE NORMAL
       011200  012712  000203                          MOV     #203,(R2)               ;MOVE 'RTS R3' INTO NEXT
                                                                                       ;PARITY MEMORY AREA LOCATION
                                                                                       ;IN CASE WE DON'T ABORT
       011204  004362  177770                          JSR     R3,-10(R2)              ;GO TO PARITY MEMORY AREA
       011210  104001                                  HLT     +1                      ;DIDN'T ABORT
       011212  000410                                  BR      .+22                    ;GO TO NEXT TEST
       011214  042777  000001  170400  Z1:             BIC     #BIT0,@PARITY           ;DISABLE PARITY
       011222  004037  011550                          JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
       011226  104003                                  HLT     +3                      ;ABORTED INCORRECTLY
       011230  012706  001100                          MOV     #STACK,SP               ;RESET THE STACK
                                               ;*****************************************************************
                                               ;TEST 65         TEST    BEQ INSTRUCTION   (-BCOK)
```

```
                                       ;*****************************************************************
       011234  000004                  TST65:  SCOPE
                                       ;                       11/45 **** ROM STATE 324 ****
                                       ;
                                       ;       11/40 **** ROM STATE 1 ****
       011236  004015                          JSR     R0,(R5)                 ;SET UP PARITY VECTOR SERVICE
       011240  011552                          2$                              ;ROUTINE ADDRESS
       011242  042777  000004  170352          BIC     #BIT2,@PARITY           ;WRITE NORMAL
       011250  010102                          MOV     R1,R2                   ;SET UP FOR A DATO
       011252  162702  000004                  SUB     #4,R2                   ;CALCULATE THE START ADDRESS
                                                                               ;FOR THIS TEST
       011256  012722  012700                  MOV     #12700,(R2)+            ;MOVE THE INSTRUCTION
                                                                               ;'MOV #-1,R0' TO PARITY MEMORY
                                                                               ;AREA
       011262  012722  177777                  MOV     #-1,(R2)+               ;MOVE A -1 INTO NEXT PARITY
                                                                               ;MEMORY AREA LOCATION
       011266  012722  001401                  MOV     #1401,(R2)+             ;MOVE THE INSTRUCTION
                                                                               ;'BEQ .+4' INTO NEXT PARITY
                                                                               ;MEMORY AREA LOCATION
       011272  052777  000004  170322          BIS     #BIT2,@PARITY           ;WRITE OTHER PARITY
       011300  005737  001402                  TST     @#CPU40                 ;ARE WE ON AN 11/40?
       011304  001403                          BEQ     2$                      ;BRANCH IF NO
                                                                               ;SINCE THE PC WILL BE UPDATED
                                                                               ;ON THE PARITY ABORT
       011306  010237  001332                  MOV     R2,@#$GDDAT             ;STORE THIS PC THAT SHOULD
                                                                               ;BE PUSHED ON THE STACK
                                                                               ;IF A PARITY ABORT OCCURS
                                                                               ;SINCE THE PC WILL NOT BE
                                                                               ;UPDATED ON THE PARITY ABORT
       011312  012722  000240                  MOV     #240,(R2)+              ;MOVE A 'NOP' INTO NEXT
                                                                               ;PARITY MEMORY AREA LOCATION
       011316  000404                          BR      1$                      ;CONTINUE WITH TEST
       011320  012722  000240          2$:     MOV     #240,(R2)+              ;MOVE A 'NOP' INTO NEXT
                                                                               ;PARITY MEMORY AREA LOCATION
       011324  010237  001332                  MOV     R2,@#$GDDAT             ;STORE THE PC THAT SHOULD
                                                                               ;BE PUSHED ON THE STACK
                                                                               ;IF A PARITY ABORT OCCURS
       011330  042777  000004  170264  1$:     BIC     #BIT2,@PARITY           ;WRITE NORMAL
       011336  012712  000203                  MOV     #203,(R2)               ;MOVE 'RTS R3' INTO NEXT
                                                                               ;PARITY MEMORY AREA LOCATION
                                                                               ;IN CASE WE DON'T ABORT
       011342  004362  177770                  JSR     R3,-10(R2)              ;GO TO PARITY MEMORY AREA
       011346  104001                          HLT     +1                      ;DIDN'T ABORT
       011350  000410                          BR      .+22                    ;GO TO NEXT TEST
       011352  042777  000001  170242  3$:     BIC     #BIT0,@PARITY           ;DISABLE PARITY
       011360  004037  011550                  JSR     R0,@#CHECKLOC           ;CHECK FOR GOOD ABORT
       011364  104003                          HLT     +3                      ;ABORTED INCORRECTLY
       011366  012706  001100                  MOV     #STACK,SP               ;RESET THE STACK
                                       ;*****************************************************************
                                       ;TEST 66         END OF PROGRAM
                                       ;*****************************************************************
       011372  000004                  TST66:  SCOPE
       011374  042777  100005  170220          BIC     #BIT15!BIT2!BIT0,@PARITY        ;DISABLE ALL PARITY
                                                                               ;AND CLEAR ERROR BIT!
```



```
       011402  005737  001630                  TST     @#USERTYPE              ;DID THE USER SELECT THE REGISTERS?
       011406  001014                          BNE     1$                      ;BRANCH IF SO AND DON'T STEP
                                                                               ;UP THE TABLE
       011410  062737  000002  001336          ADD     #2,@#$REGAD             ;STEP UP TO NEXT REGISTER
       011416  062737  000002  001414          ADD     #2,@#$THPAD             ;STEP UP TO CORRESPONDING
                                                                               ;PARITY MEMORY
       011424  062737  000002  001416          ADD     #2,@#$SETAD             ;STEP UP TO NEXT OFFSET - THIS
                                                                               ;IS ONLY APPLICABLE IF MEMORY
                                                                               ;MGMT IS TURNED ON
       011432  062737  000002  001446          ADD     #2,@CNTERAD             ;STEP UP TO NEXT INTERLEAVE
                                                                               ;VALUE
       011440  004337  011474          1$:     JSR     R3,@#FLAGSCLR           ;CLEAR PERTINENT FLAGS
       011444  000167  000320                  JMP     $EOP                    ;GO TO RING-A-DING
                                                                               ;BEFORE REITERATING THE PROGRAM




                                       ;*****************************************************************
                                       ;
                                       ;ROUTINE FOR SETTING UP THE PARITY VECTOR SERVICE ADDRESS
                                       ;
                                       ;*****************************************************************
       011450  012077  170144          VECSET: MOV     (R0)+,@INTVEC           ;WRITE ADDRESS INTO LOCATION 114
       011454  052777  000005  170140          BIS     #BIT2!BIT0,@PARITY      ;WRITE OTHER PARITY AND DISABLE
       011462  000200                          RTS     R0                      ;RETURN TO TESTING
                                       ;*****************************************************************
                                       ;
                                       ;ROUTINE TO RESET AND GO BACK TO TABLE BEGINNING
                                       ;
                                       ;*****************************************************************
       011464  004337  011506          RESTART:        JSR     R3,@#INITIALIZE
       011470  000167  000274                  JMP     $EOP                    ;GO TO RING-A-DING
                                       ;*****************************************************************
                                       ;
                                       ;ROUTINE TO CLEAR PERTINENT FLAGS BEFORE PASSING THRU THE PROGRAM
                                       ;WITH ANOTHER TABLE ENTRY
                                       ;
                                       ;*****************************************************************
       011474  005037  001624          FLAGSCLR:       CLR     @#PSPCORZONES           ;CLEAR PS, PC AND ZONES ABORT
                                                                               ;FLAG
       011500  005037  001626                  CLR     @#MSREGFLAG             ;CLEAR MS11 REGISTER PRESENCE
                                                                               ;FLAG
       011504  000203                          RTS     R3                      ;RETURN
                                       ;*****************************************************************
                                       ;
                                       ;ROUTINE TO COMPLETELY REINITIALIZE BEFORE RESTARTING PROGRAM OVER
                                       ;AT THE BEGINNING OF THE TABLE
                                       ;
                                       ;*****************************************************************
       011506  005037  001624          INITIALIZE:     CLR     @#PSPCORZONES           ;CLEAR PS, PC AND ZONES
                                                                               ;ABORT FLAG
       011512  005037  001626                  CLR     @#MSREGFLAG             ;CLEAR MS11 REGISTER PRESENCE
                                                                               ;FLAG
       011516  012737  001340  001336          MOV     #SREG0,@#REGAD          ;MOVE FIRST REGISTER CONTAINER
```

```
                                                        ;INTO PREGAD WHICH IS USED TO
                                                        ;POINT TO THE PARITY REGISTER
                                                        ;TABLE
011524  012737  001366  001414      MOV   #STMP0,@#STMPAD  ;MOVE FIRST MEMORY CONTAINER
                                                        ;INTO STMPAD WHICH IS USED TO
                                                        ;POINT TO THE MEMORY LOCATION
                                                        ;TABLE
011532  012737  001420  001416      MOV   #SSETO,@#SSETAD  ;MOVE FIRST OFFSET CONTAINER
                                                        ;INTO SSETAD WHICH IS USED TO
                                                        ;POINT TO THE OFFSET LOCATION
                                                        ;TABLE IF MEMORY MGMT IS USED
011540  012737  001452  001406      MOV   #NTER0,@#NTERAD  ;MOVE 1ST INTERLEAVE CONTAINER
                                                        ;INTO 'INTERAD' WHICH IS USED
                                                        ;TO POINT TO THE INTERLEAVE
                                                        ;LOCATION TABLE
011546  000203                      RTS   R3               ;RETURN


                        ;*******************************************************************
                        ;
                        ;SUBROUTINE TO CHECK THAT IF A TEST HAS ABORTED IT DID SHOULD
                        ;ABORT IN THE PROPER PLACE.  IT IS QUITE CONCEIVABLE THAT A TEST
                        ;THAT SHOULD HAVE ABORTED IN THE TWO LOCATION MAP AREA ABORTED
                        ;ON THE FETCH OF THE INSTRUCTION THAT WAS TO CAUSE THE ABORT.
                        ;THIS SUBROUTINE WILL FLAG SUCH OCCURRENCES.  WITHOUT THIS CHECK
                        ;THE PROGRAM WOULD APPEAR TO HAVE RUN PROPERLY.
                        ;
                        ;BOTH THE CORRECT HIGH ORDER ERROR ADDRESS BITS AND PROPER PC
                        ;PUSH ON THE STACK ARE LOOKED FOR AFTER A PARITY ABORT OCCURS.
                        ;
                        ;*******************************************************************
011550  010246                CHECKLOC:  MOV   R2,-(SP)    ;SAVE R2 CONTENTS ON STACK
011552  010346                      MOV   R3,-(SP)         ;SAVE R3 CONTENTS ON STACK
011554  010446                      MOV   R4,-(SP)         ;SAVE R4 CONTENTS ON STACK
011556  005002                      CLR   R2               ;CLEAR ERROR ADDRESS BIT COMPARE
011560  005004                      CLR   R4               ;REGISTERS IN CASE WE HAVE AN OLD
                                                        ;MOS DESIGN THAT DOESN'T HAVE
                                                        ;ADDRESS BITS
011562  005737  001626              TST   @#MSREGFLAG      ;IS AN MF11 OR MS11 PARITY
                                                        ;OPTION BEING TESTED?
011566  001027                      BNE   7$               ;BRANCH IF MS11 AND DON'T DO
                                                        ;ADDRESS BITS CHECKING
011570  005737  001624              TST   @#PSPCORZONES    ;ARE WE DOING A PS OR PC FETCH
                                                        ;FOR ZONE ABORT TEST?
011574  001402                      BEQ   5$               ;BRANCH IF NO
011576  005002                      CLR   R2               ;SET THE PARITY REGISTER HIGH
                                                        ;ORDER ADDRESS BITS VALUE
                                                        ;(BITS 5 THRU 11) TO ZERO FOR
                                                        ;THE PS OR PC FETCH ABORT TESTS
011600  000414                      BR    2$               ;GO CHECK IT AGAINST ACTUAL
                                                        ;PARITY REGISTER VALUE
011602  005737  002304        5$:   TST   @#KT11           ;MEMORY MANAGEMENT ON?
011606  100003                      BPL   6$               ;BRANCH IF NO
011610  017702  167602              MOV   @SSETAD,R2       ;PICK UP THE OFFSET VALUE - IT
```



```
                                                        ;SHOULD BE THE VALUE THAT WILL
                                                        ;APPEAR IN THE PARITY REGISTER
                                                        ;ERROR ADDRESS BITS
011614  000406                      BR    2$               ;GO TO CHECK!
011616  012703  017400        6$:   MOV   #17400,R3        ;GET A FIRST POSSIBLE ABORT
                                                        ;LOCATION AREA
011622  012702  000140              MOV   #140,R2          ;GET THE FIRST POSSIBLE PARITY
                                                        ;REGISTER HIGH ORDER ADDRESS
                                                        ;BITS VALUE (BITS 5 THRU 11)
011626  020103                1$:   CMP   R1,R3            ;IS THIS THE ABORT AREA BEING
                                                        ;USED?
011630  101027                      BHI   3$               ;NO - SEE IF IT'S THE NEXT ONE
011632  017704  167764        2$:   MOV   @PARITY,R4       ;PROCEED TO SEE IF
                                                        ;IT WAS A PROPER ABORT BY LOOKING
                                                        ;AT THE HIGH ORDER ADDRESS BITS
                                                        ;OF THE PARITY REGISTER
                                                        ;(BITS 5 THRU 11)
011636  042704  170037              BIC   #170037,R4       ;CLEAR ALL BITS EXCEPT 5 THRU 11
011642  020402                      CMP   R4,R2            ;ARE THE ERROR ADDRESS BITS
                                                        ;WHAT THEY SHOULD BE?
011644  001026                      BNE   4$               ;BRANCH IF NO
011646  010237  001326        7$:   MOV   R2,@#SGDADR      ;STORE WHAT THE ADDRESS BITS
                                                        ;SHOULD HAVE BEEN
011652  010437  001330              MOV   R4,@#SBDADR      ;STORE WHAT THE ADDRESS BITS
                                                        ;WERE
011656  012604                      MOV   (SP)+,R4         ;RESTORE R4 CONTENTS
011660  012603                      MOV   (SP)+,R3         ;RESTORE R3 CONTENTS
011662  012602                      MOV   (SP)+,R2         ;RESTORE R2 CONTENTS

                        ;IF WE HAVE REACHED THIS PATH 1 OF 2 CONDITIONS EXIST --
                        ;WE HAVE AN OLD MOS DESIGN WITH NO ADDRESS BITS, OR WE HAVE
                        ;CORE OR THE NEW MOS DESIGN WITH ADDRESS BITS OK!!

011664  026637  000002  001332      CMP   2(SP),@#SGDDAT   ;WAS THE CORRECT PC PUSHED
                                                        ;ON THE STACK?
011672  001404                      BEQ   9$               ;BRANCH IF YES
011674  016637  000002  001334      MOV   2(SP),@#SBDDAT   ;SAVE INCORRECT PC FOR PRINTOUT
011702  000200                      RTS   R0               ;GO BACK TO INDICATE BAD ABORT
011704  005720                9$:   TST   (R0)+            ;STEP UP RETURN ADDRESS
                                                        ;TO BYPASS THE ERROR HLT
011706  000200                      RTS   R0               ;RETURN TO CONTINUE TESTING
011710  062703  004000        3$:   ADD   #4000,R3         ;STEP UP TO THE NEXT POSSIBLE
                                                        ;ABORT LOCATION AREA
011714  062702  000040              ADD   #40,R2           ;CHANGE THE VALUE OF THE HIGH
                                                        ;ORDER ADDRESS BITS VALUE TO BE
                                                        ;CHECKED
011720  000742                      BR    1$               ;GO BACK TO CHECK THIS ONE
011722  010237  001326        4$:   MOV   R2,@#SGDADR      ;STORE WHAT THE ADDRESS BITS
                                                        ;SHOULD HAVE BEEN
011726  010437  001330              MOV   R4,@#SBDADR      ;STORE WHAT THE
                                                        ;ADDRESS BITS WERE
011732  012604                      MOV   (SP)+,R4         ;RESTORE R4 CONTENTS
011734  012603                      MOV   (SP)+,R3         ;RESTORE R3 CONTENTS
011736  012602                      MOV   (SP)+,R2         ;RESTORE R2 CONTENTS
```

```
                                                ;IF WE HAVE REACHED THIS PATH WE HAVE CORE OR THE NEW MOS DESIGN
                                                ;WITH BAD ADDRESS BITS WHICH SHOULD INDICATE AN IMPROPER ABORT!!

011740  026627  000002  001334                  CMP     2(SP),@#$BDDAT          ;WAS THE CORRECT PC PUSHED
                                                                                ;ON THE STACK?
011746  001404                                  BEQ     10$                     ;BRANCH IF YES
011750  016637  000002  001334                  MOV     2(SP),@#$BDDAT          ;SAVE INCORRECT PC FOR PRINTOUT
011756  000200                                  RTS     R0                      ;GO BACK TO INDICATE BAD ABORT
011760  012737  167346  001334  10$:    MOV     #GOODAT,@#$BDDAT        ;RECORD THE FACT THAT THE
                                                                                ;CORRECT PC WAS PUSHED ON STACK
011766  000200                                  RTS     R0                      ;GO BACK TO INDICATE BAD ABORT
```



```
                                ;*******************************************************************
                                ;END OF PASS ROUTINE
                                ;INCREMENT THE PASS NUMBER
                                ;IF SW10=0 DING THE TTY BELL ON END OF PROGRAM
                                ;IF THERE IS A MONITOR GO TO IT.
                                ;IF NONE JUMP TO START
011770  000004                  $EOP:   SCOPE
011772  005067  167304                  CLR     $TSTNM          ;ZERO THE TEST NUMBER
011776  005067  000304                  CLR     $TIMES          ;ZERO THE NUMBER OF ITERATIONS
012002  005267  167272                  INC     $PASS           ;INCREMENT THE PASS NUMBER
012006  032737  002000  177570          BIT     #SW10,@#SWR     ;RING THE BELL?
012014  001002                          BNE     4$              ;NO
012016  104400  012056                  TYPE    ,$BELL          ;RING A DING
012022  013700  000042          4$:     MOV     @#42,R0         ;GET MONITOR ADDRESS
012026  001411                          BEQ     $DOAGN          ;IF NONE
012030  022760  177777  000002          CMP     #-1,2(R0)
012036  001001                          BNE     $ENDAD
012040  000005                          RESET
012042  004710                  $ENDAD: JSR     PC,(R0)         ;GO TO MONITOR
012044  000240                          NOP                     ;SAVE ROOM
012046  000240                          NOP                     ;FOR
012050  000240                          NOP                     ;ACT11
012052  000137  003620          $DOAGN: JMP     @#START         ;RETURN
012056  177607  020377          $BELL:  .ASCIZ  <207><377><377>
```

```
                                          ;*********************************************************************
                                          ;THIS ROUTINE IS THE SCOPE HANDLER
                                          ;SW14=1 LOOP ON TEST
                                          ;SW11=1 INHIBIT ITERATIONS
                                          ;SW09=1 LOOP ON ERROR
                                          ;SW07=1 LOOP ON TEST IN SWR<7:0>
                                          ;THE TEST NUMBER ($TSTNM) IS UPDATED AND DISPLAYED
012062                                    $SCOPE:
012062  006137  177570                            ROL     @#SWR                   ;LOOP ON PRESENT TEST?
012066  100502                                    BMI     $OVER                   ;YES IF SW14=1
                                                  ;#####START OF CODE FOR THE XOR TESTER#####
012070  000416                            $XTSTR: BR      6$                      ;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                                  ;THIS INSTRUCTION TO A "NOP" (NOP=240)
012072  013746  000004                            MOV     @#ERRVEC,-(SP)          ;SAVE THE CONTENTS OF THE ERROR VECTOR
012076  012737  012116  000004                    MOV     #5$,@#ERRVEC            ;SET FOR TIMEOUT
012104  005737  177060                            TST     @#177060                ;TIME OUT ON XOR?
012110  012637  000004                            MOV     (SP)+,@#ERRVEC          ;RESTORE THE ERROR VECTOR
012114  000457                                    BR      $SVLAD                  ;GO TO THE NEXT TEST
012116  022626                            5$:     CMP     (SP)+,(SP)+             ;CLEAR THE STACK AFTER A TIME OUT
012120  012637  000004                            MOV     (SP)+,@#ERRVEC          ;RESTORE THE ERROR VECTOR
012124  000417                                    BR      7$                      ;LOOP ON THE PRESENT TEST
012126                                    6$:     ;#####END OF CODE FOR THE XOR TESTER#####
012126  032737  000400  177570                    BIT     #SW08,@#SWR             ;LOOP ON SPEC. TEST?
012134  001404                                    BEQ     2$                      ;BR IF NO
012136  123767  177570  167136                    CMPB    @#SWR,$TSTNM            ;ON THE RIGHT TEST?     SWR<7:0>
012144  001453                                    BEQ     $OVER                   ;BR IF YES
012146  105767  167142                    2$:     TSTB    $ERFLG                  ;HAS AN ERROR OCCURRED?
012152  001415                                    BEQ     3$                      ;BR IF NO
012154  032737  001000  177570                    BIT     #SW09,@#SWR             ;LOOP ON ERROR?
012162  001404                                    BEQ     4$                      ;BR IF NO
012164  016767  167120  167114            7$:     MOV     $LPERR,$LPADR           ;SET LOOP ADDRESS TO LAST SCOPE
012172  000440                                    BR      $OVER
012174  105067  167114                    4$:     CLRB    $ERFLG                  ;ZERO THE ERROR FLAG
012200  005067  000104                            CLR     $TIMES                  ;CLEAR THE NUMBER OF ITERATIONS TO MAKE
012204  000415                                    BR      1$                      ;ESCAPE TO THE NEXT TEST
012206  032737  004000  177570            3$:     BIT     #SW11,@#SWR             ;INHIBIT ITERATIONS?
012214  001011                                    BNE     1$                      ;BR IF YES
012216  005767  167056                            TST     $PASS                   ;IF FIRST PASS OF PROGRAM
012222  001406                                    BEQ     1$                      ;       INHIBIT ITERATIONS
012224  005267  167054                            INC     $ICNT                   ;INCREMENT ITERATION COUNT
012230  026767  000054  167046                    CMP     $TIMES,$ICNT            ;CHECK THE NUMBER OF ITERATIONS MADE
012236  002016                                    BGE     $OVER                   ;BR IF MORE ITERATION REQUIRED
012240  012767  000001  167036            1$:     MOV     #1,$ICNT                ;REINITIALIZE THE ITERATION COUNTER
012246  016767  000040  000034                    MOV     $MXCNT,$TIMES           ;SET NUMBER OF ITERATIONS TO DO
012254  005267  167022                    $SVLAD: INC     $TSTNM                  ;COUNT TEST NUMBERS
012260  011667  167022                            MOV     (SP),$LPADR             ;SAVE SCOPE LOOP ADDRESS
012264  011667  167020                            MOV     (SP),$LPERR             ;SAVE ERROR LOOP ADDRESS
012270  005067  000000                            CLR     $ESCAPE                 ;CLEAR THE ESCAPE FROM ERROR ADDRESS
012274  016737  167002  177570            $OVER:  MOV     $TSTNM,@#DISPLAY        ;DISPLAY TEST NUMBER
012302  016716  167000                            MOV     $LPADR,(SP)             ;FUDGE RETURN ADDRESS
012306  000002                                    RTI                             ;FIXES PS
012310  000000                            $TIMES: 0                               ;NUMBER OF ITERATIONS TO PERFORM
012312  000000                            $MXCNT: 0                               ;MAX. NUMBER OF ITERATIONS
```



```
                                          ;*********************************************************************
                                          ;ROUTINE TO ACCEPT AN OCTAL NUMBER FROM THE TTY
                                          ;CALL:
                                          ;               ACCEPT  ,ADDR           ;PUT OCTAL NUMBER IN ADDR
                                          ;
012314                                    $ACCEPT:
012314  010046                                    MOV     R0,-(SP)                        ;PUSH R0 ON STACK
012316  010146                                    MOV     R1,-(SP)                        ;PUSH R1 ON STACK
012320  010246                                    MOV     R2,-(SP)                        ;PUSH R2 ON STACK
012322  010346                                    MOV     R3,-(SP)                        ;PUSH R3 ON STACK
012324  016600  000010                            MOV     10(SP),R0                       ;GET ADDRESS POINTER OF WHERE TO PUT NUMBER
012330  005001                            1$:     CLR     R1              ;CLEAR PARTIAL NUMBER
012332  012702  000006                            MOV     #6,R2           ;MAX. # OF DIGITS ALLOWED
012336  104402                            2$:     GETCHR                  ;GET ONE CHARACTER
012340  112603                                    MOVB    (SP)+,R3        ;AND PUT IT IN R3
012342  110367  000102                            MOVB    R3,6$
012346  104402  012450                            TYPE    ,6$             ;ECHO THE CHARACTER
012352  022703  000015                            CMP     #15,R3          ;WAS THIS CHARACTER A "CR"?
012356  001402                                    BEQ     5$              ;BR IF YES
012360  022703  000060                            CMP     #'0,R3          ;INSURE THE CHARACTER IS
012364  003014                                    BGT     4$              ;A DIGIT BETWEEN 0 AND 7.
012366  022703  000067                            CMP     #'7,R3
012372  002411                                    BLT     4$
012374  005302                                    DEC     R2              ;CHECK NUMBER OF CHARACTERS
012376  002407                                    BLT     4$              ;BR IF TO MANY
012400  006301                                    ASL     R1              ;POSITION PARTIAL NUMBER
012402  006301                                    ASL     R1              ;FOR THIS DIGIT
012404  006301                                    ASL     R1
012406  042703  177770                            BIC     #^C<7>,R3       ;GET RID OF THE ASCII JUNK
012412  050301                                    BIS     R3,R1           ;COMBINE THIS DIGIT WITH PARTIAL
012414  000750                                    BR      2$              ;GO GET ANOTHER DIGIT
012416  104400  012452                    4$:     TYPE    ,$QUES          ;TYPE "?"
012422  000742                                    BR      1$              ;GO START OVER
012424  104400  012454                    5$:     TYPE    ,$LF            ;FOLLOW "CR" WITH A "LF"
012430  010130                                    MOV     R1,@(R0)+       ;PASS THE NUMBER TO THE USER
012432  010066  000010                            MOV     R0,10(SP)       ;SET FOR RETURN
012436  012603                                    MOV     (SP)+,R3                        ;POP STACK INTO R3
012440  012602                                    MOV     (SP)+,R2                        ;POP STACK INTO R2
012442  012601                                    MOV     (SP)+,R1                        ;POP STACK INTO R1
012444  012600                                    MOV     (SP)+,R0                        ;POP STACK INTO R0
012446  000002                                    RTI
012450     000     000                    6$:     .BYTE   0,0             ;STORAGE FOR ASCII CHAR. AND TERMINATOR
012452     077                            $QUES:  .ASCII  "?"                             ;QUESTION MARK
012453     015                            $CRLF:  .ASCII  <15>                            ;CARRAIGE RETURN
012454  000012                            $LF:    .ASCIZ  <12>                            ;LINEFEED
                                                                                          ;SELECTED REGISTER FROM THE TTY
                                          ;*********************************************************************
                                          ;THIS ROUTINE INPUTS A SINGLE CHARACTER FROM THE TTY
                                          ;CALL:
                                          ;
                                          ;               GETCHR                  ;INPUT A SINGLE CHARACTER FROM THE TTY
                                          ;               RETURN HERE             ;CHARACTER IS ON THE STACK
                                          ;
012456  011646                            $READ:  MOV     (SP),-(SP)      ;PUSH DOWN THE PC
```

```
012460  016666  000004  000002          MOV     4(SP),2(SP)     ;SAVE THE PS
012466  105737  177560          1$:     TSTB    @TKS            ;WAIT FOR
012472  100375                          BPL     1$              ;A CHARACTER
012474  116766  000122  000004          MOVB    TKB,4(SP)       ;READ THE TTY
012502  042766  177600  000004          BIC     #^C<177>,4(SP)  ;GET RID OF JUNK IF ANY
012510  000002                          RTI                     ;GO BACK TO USER
                                ;*******************************************************************
                                ;THIS ROUTINE INPUTS A STRING FROM THE TTY
                                ;CALL:
                                ;       GETSTR                  ;INPUT A STRING FROM THE TTY
                                ;       RETURN HERE             ;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                ;                               ;TERMINATOR WILL BE A BINARY 0
                                ;
012512  010346                  $READS: MOV     R3,-(SP)        ;SAVE R3
012514  012703  012624          1$:     MOV     #$TTYIN,R3      ;GET ADDRESS
012520  022703  012634          2$:     CMP     #$TTYIN+8.,R3   ;BUFFER FULL?
012524  101405                          BLOS    4$              ;BR IF YES
012526  104402                          GETCHR                  ;GO READ ONE CHARACTER FROM THE TTY
012530  112613                          MOVB    (SP)+,(R3)      ;GET CHARACTER
012532  122713  000177                  CMPB    #177,(R3)       ;IS IT A RUBOUT
012536  001003                          BNE     3$              ;SKIP IF NOT
012540  104400  012602          4$:     TYPE    ,$QUES          ;TYPE A '?'
012544  000763                          BR      1$              ;ZAP THE BUFFER AND LOOP
012546  111367  000044          3$:     MOVB    (R3),8$         ;ECHO THE CHARACTER
012552  104400  012616                  TYPE    ,8$
012556  122723  000015                  CMPB    #15,(R3)+       ;CHECK FOR RETURN
012562  001356                          BNE     2$              ;LOOP IF NOT RETURN
012564  105063  177777                  CLRB    -1(R3)          ;ZAP RETURN (THE 15)
012570  104400  012454                  TYPE    ,$LF            ;TYPE A LINE FEED
012574  012603                          MOV     (SP)+,R3        ;RESTORE R3
012576  011646                          MOV     (SP),-(SP)      ;ADJUST THE STACK AND PUT ADDRESS OF THE
012600  016666  000004  000002          MOV     4(SP),2(SP)     ;       FIRST ASCII CHARACTER ON IT
012606  012766  012624  000004          MOV     #$TTYIN,4(SP)
012614  000002                          RTI                     ;RETURN
012616     000                  8$:     .BYTE   0               ;STORAGE FOR ASCII CHAR. TO TYPE
012617     000                          .BYTE   0               ;TERMINATOR
012620  177560                  TKS:    177560                  ;TTY KBD STATUS
012622  177562                  TKB:    177562                  ;TTY KBD BUFFER
012624  000010                  $TTYIN: .BLKB   8.              ;RESERVE 8 BYTES FOR TTY INPUT
                                                                ;FROM THE TTY
```



```
                                ;*******************************************************************
                                ;THIS ROUTINE IS THE HLT HANDLER
                                ;SW09=1 LOOP ON ERROR
                                ;SW10=1 BELL ON ERROR
                                ;SW13=1 INHIBIT ERROR TYPEOUTS
                                ;SW15=1 HALT ON ERROR
                                ;GO TO TYPERR ON ERROR
        012634                  $HLT:
012634  042777  000001  164760          BIC     #BIT0,@PARITY
012642  112767  000001  151040          MOVB    #1,$ERFLG               ;SET THE ERROR FLAG
012650  032737  002000  177570          BIT     #SW10,@#SWR             ;BELL ON ERROR?
012656  001402                          BEQ     1$                      ;NO - SKIP
012660  104400  012056                  TYPE    ,$BELL                  ;RING BELL
012664  005267  166422          1$:     INC     $ERTTL                  ;COUNT THE NUMBER OF ERRORS
012670  001775                          BEQ     1$                      ;INSURE $ERTTL NOT=0
012672  011667  166426                  MOV     (SP),$HLTAD             ;GET ADDRESS OF HLT INSTRUCTION
012676  162767  000002  166420          SUB     #2,$HLTAD
012704  117767  166414  166410          MOVB    @$HLTAD,$ITEMB          ;STRIP AND SAVE THE ERROR ITEM CODE
012712  032737  020000  177570          BIT     #SW13,@#SWR             ;SKIP TYPEOUT IF SET
012720  001004                          BNE     2$                      ;SKIP TYPEOUTS
012722  004737  012776                  JSR     PC,@#TYPERR             ;GO TO USER ERROR ROUTINE
012726  104400  012453                  TYPE    ,$CRLF
012732  005737  177570          2$:     TST     @#SWR                   ;HALT ON ERROR
012736  100001                          BPL     3$                      ;SKIP IF CONTINUE
012740  000000                          HALT                            ;HALT ON ERROR!
012742  032737  001000  177570  3$:     BIT     #SW09,@#SWR             ;LOOP ON ERROR SWITCH SET?
012750  001403                          BEQ     4$                      ;BR IF NO
012752  016716  166332                  MOV     $LPERR,(SP)             ;FUDGE RETURN FOR LOOPING
012756  000002                          RTI
012760  005767  000010          4$:     TST     $ESCAPE                 ;CHECK FOR AN ESCAPE ADDRESS
012764  001402                          BEQ     5$                      ;BR IF NONE--TAKE NORMAL RETURN
012766  016716  000002                  MOV     $ESCAPE,(SP)            ;FUDGE RETURN ADDRESS FOR ESCAPE
012772  000002                  5$:     RTI
012774  000000                  $ESCAPE:        0                       ;ESCAPE ON ERROR ADDRESS
                                                                        ;MESSAGE TYPEOUT ROUTINE
                                                                        ;DEFINED BY 'TYPERR' AND
                                                                        ;1ST INSTRUCTION OF ERROR 'HLT'
                                                                        ;ROUTINE
                                ;*******************************************************************
                                ;
                                ;THIS ROUTINE WILL TYPE OUT THE ERROR MESSAGES
                                ;
                                ;*******************************************************************
012776  104400  012453          TYPERR: TYPE    ,$CRLF                  ;TYPE A CARRIAGE RETURN
                                                                        ;AND LINE FEED
013002  010046                          MOV     R0,-(SP)                ;SAVE R0 CONTENTS
013004  005000                          CLR     R0                      ;CLEAR R0
013006  116700  166310                  MOVB    $ITEMB,R0               ;PICK UP THE ITEM INDEX
013012  005300                          DEC     R0                      ;ADJUST THE INDEX
013014  006300                          ASL     R0                      ;SO IT WILL WORK FOR
013016  006300                          ASL     R0                      ;FOR THE ERROR TABLE
013020  006300                          ASL     R0
013022  062700  001500                  ADD     #$ERRTB,R0              ;FORM THE TABLE POINTER
```

```
3161  013026  012702  000002          MOV     (R0)+,R2                ;PICK UP 'ERROR MESSAGES' POINTER
3162  013032  104412                  TYPE                            ;TYPE 'ERROR MESSAGE'
3163  013034  000000          1$:     0                               ;'ERROR MESSAGE' POINTER GOES HERE
3164  013036  104412  012403          TYPE    ,$CRLF                  ;TYPE A CARRIAGE RETURN AND
3165                                                                  ;LINE FEED
3166  013042  012067  000664          MOV     (R7)+,20                ;PICK UP 'DATA HEADER' POINTER
3167  013046  001402                  BEQ     3$                      ;IF '0' DON'T TYPE
3168  013050  104412                  TYPE                            ;TYPE 'DATA HEADER'
3169  013052  000000          2$:     0                               ;'DATA HEADER' POINTER GOES HERE
3170  013054  104412  012403          TYPE    ,$CRLF                  ;TYPE A CARRIAGE RETURN AND
3171                                                                  ;LINE FEED
3172  013060  012000                  MOV     (R0)+,R0                ;PICK UP 'DATA POINTER'
3173  013062  001004                  BNE     9$                      ;IF THERE IS DATA TO TYPE GO DO IT
3174  013064  012603          4$:     MOV     (SP)+,R3                ;RESTORE R0
3175  013066  104410  012403          TYPE    ,$CRLF                  ;TYPE A CARRIAGE RETURN AND
3176                                                                  ;AND LINE FEED
3177  013072  000207                  RTS     PC                      ;RETURN TO TESTING
3178  013074                  5$:
3179  013074  013046                  MOV     @(R6)+,-(SP)            ;SAVE @(R6)+ FOR TYPEOUT
3180                                                                  ;TYPE DATA
3181  013076  004437  002046          JSR     R4,$B2OCT               ;GO TYPE-"OCTAL ASCII
3182  013102      006                 .BYTE   6                       ;TYPE 6 DIGITS
3183  013103      001                 .BYTE   1                       ;TYPE LEADING ZEROS
3184  013104  005715                  TST     (R5)                    ;HAVE WE REACHED THE "0" TERMINATOR
3185  013106  001406                  BEQ     8$                      ;YES - CLEAN UP FOR RETURN
3186  013110  104402  013116          TYPE    ,6$                     ;TYPE 5 SPACES
3187  013114  000767                  BR      5$                      ;LOOP TILL "0" TERMINATOR REACHED
3188  013116  040040  020040  000006  6$:     .ASCIZ  /     /
3189                                  .EVEN
```



```
3190                                  ;*****************************************************************
3191                                  ;
3192                                  ;THE FOLLOWING ROUTINE WILL SIZE MEMORY
3193                                  ;
3194                                  ;$LSTBLK WILL CONTAIN THE LAST BANK AS AN SAF (SEGMENT ADDRESS FIELD)
3195                                  ;
3196                                  ;THE SEGMENT ADDRESS FIELD CONTAINS THE 4 MOST SIGNIFICANT BITS OF
3197                                  ;THE LAST ADDRESS OF THE LAST BANK FOUND
3198                                  ;
3199                                  ;*****************************************************************
3200  013120  010046          $SIZE:  MOV     R5,-(SP)                ;SAVE R5 CONTENTS
3201  013122  010665                  MOV     SP,R5                   ;SAVE THE STACK POINTER
3202  013124  012703  077406          MOV     #77406,R3
3203  013130  010377  104052          MOV     R3,@#KPDR0              ;SET THE FOLLOWING PAGE
3204  013134  010377  165206          MOV     R3,@#KPDR1              ;DESCRIPTOR REGISTERS TO
3205  013140  010377  165204          MOV     R3,@#KPDR2              ;READ/WRITE AND TRANSFER OF
3206  013144  010377  165202          MOV     R3,@#KPDR3              ;4096 (10) WORDS PER SEGMENT
3207  013150  010377  165200          MOV     R3,@#KPDR4
3208  013154  010377  165216          MOV     R3,@#KPDR5
3209  013160  010377  165034          MOV     R3,@#KPDR6
3210  013164  010377  165032          MOV     R3,@#KPDR7
3211  013170  005077  165230          CLR     @#KPAR0                 ;SET THE FOLLOWING PAGE
3212  013174  012777  000200  165024  MOV     #200,@#KPAR1            ;ADDRESS REGISTERS TO THEIR
3213  013202  012777  000400  165020  MOV     #400,@#KPAR2            ;RESPECTIVE OFFSET VALUES
3214  013210  012777  000600  165014  MOV     #600,@#KPAR3            ;FOR RELOCATION PURPOSES
3215  013216  012777  001000  165010  MOV     #1000,@#KPAR4
3216  013224  012777  001200  165004  MOV     #1200,@#KPAR5
3217  013232  012777  001400  165000  MOV     #1400,@#KPAR6
3218  013240  012777  007600  164774  MOV     #7600,@#KPAR7           ;THIS ONE'S THE I/O RECORD
3219                                                                  ;PAGE CONTAINING CONTROL STATUS
3220                                                                  ;REGISTERS, ETC.
3221  013252  016704  164766          MOV     KPAR6,R4                ;GET ADDRESS OF PAGE 6 REGISTER
3222  013256  005014                  CLR     (R4)                    ;CLEAR THE REGISTER
3223  013260  005777  164720          INC     @$R0                    ;TURN ON MEMORY MANAGEMENT
3224  013264  010746                  MOV     PC,-(SP)                ;MAKE KT11 TIMEOUT SERVICE
3225  013266  062716  000026          ADD     @$KTOUT-.,(SP)          ;ROUTINE ADDRESS POSITION
3226                                                                  ;INDEPENDENT
3227  013272  012637  000004          MOV     (SP)+,@#ERRVEC          ;SET FOR TIMEOUT
3228  013276  005737  143776  1$:     TST     @#143776                ;TRAP ON NON-EXISTENT MEMORY
3229  013302  062714  000040          ADD     #40,(R4)                ;MAKE A 1K STEP
3230  013306  047714  164734          CMP     @KPAR7,(R4)             ;LAST ONE?
3231  013312  003371                  BGT     1$                      ;NO - TRY IT!
3232  013314  011400                  MOV     (R4),R0                 ;GET LAST BANK +1
3233  013316  162700  000040          SUB     #40,R0                  ;DROP BACK
3234  013322  012737  000706  000004  MOV     #6,@#ERRVEC             ;SET FOR ERRORS
3235  013330  010506                  MOV     R5,SP                   ;RESTORE THE STACK POINTER
3236  013332  010067  000310          MOV     R0,$LSTBLK              ;STORE THE SAF
3237  013336  005077  164602          CLR     @$R0                    ;TURN MEMORY MGMT OFF
3238  013342  012605          1$:     MOV     (SP)+,R5                ;RESTORE R5
3239  013344  000207                  RTS     PC                      ;RETURN TO NORMAL FLOW
3240  013346  000000          $LSTBLK:        .WORD   0               ;CONTAINS THE SAF
3241                                  ;*****************************************************************
3242                                  ;BINARY TO OCTAL (ASCII) AND TYPE
3243                                  ;B2OCT--ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
```

```
                        ;CALL;
                        ;       MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
                        ;       JSR     R5,$OUTOCT              ;CALL FOR TYPEOUT
                        ;       .BYTE   N                       ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                        ;       .BYTE   M                       ;M=1 OR 0
                        ;                                               ;1=TYPE LEADING ZEROS
                        ;                                               ;0=SUPPRESS LEADING ZEROS
                        ;
                        ;
                        ;$OUTO1----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST OUTPUT OR $OUTO16
                        ;CALL;
                        ;       MOV     NUM,-(SP)
                        ;       JSR     R0,$OUTO1
                        ;
                        ;$OUTO16--ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                        ;CALL;
                        ;       MOV     NUM,-(SP)
                        ;       JSR     R0,$OUTO1
                        ;
013350  112067  000201  $OUTOCT: MOVB    (R0)+,$CNVCDE+1         ;PICKUP THE NUMBER OF DIGITS TO TYPE
013354  112067  000175          MOVB    (R0)+,$OFILL            ;GET THE ZERO FILL SWITCH
013360  000430                  BR      $OUTO1
013362  112767  000001  000163 $OUTO16: MOVB    #1,$OFILL       ;SET THE ZERO FILL OPTION
013370  112767  000006  000150          MOVB    #6,$CNVCDE+1    ;SET FOR SIX(6) DIGITS
013376  112767  000006  000146 $OUTO1: MOVB    #6,$OCNT        ;SET THE ITERATION COUNT
013404  010346                  MOV     R3,-(SP)                ;SAVE R3
013406  010446                  MOV     R4,-(SP)                ;SAVE R4
013410  010546                  MOV     R5,-(SP)                ;SAVE R5
013412  116700  000137          MOVB    $CNVCDE+1,R4            ;GET THE NUMBER OF DIGITS TO TYPE
013416  005400                  NEG     R4
013420  062704  000006          ADD     #6,R4                   ;SUBTRACT IT FOR MAX. ALLOWED
013424  110467  000126          MOVB    R4,$CNVCDE              ;SAVE IT FOR USE
013430  116700  000117          MOVB    $OFILL,R4               ;GET THE ZERO FILL SWITCH
013434  016605  000010          MOV     10(SP),R5               ;PICKUP THE INPUT NUMBER
013440  005003                  CLR     R3                      ;CLEAR THE OUTPUT WORD
013442  006105          1$:     ROL     R5                      ;ROTATE MSB INTO "C"
013444  000403                  BR      3$                      ;GO DO MSB
013446  006105          2$:     ROL     R5                      ;FORM THIS DIGIT
013450  006105                  ROL     R5
013452  006103                  ROL     R3
013454  010503                  MOV     R5,R3
013456  006103          3$:     ROL     R3                      ;GET LSB OF THIS DIGIT
013460  105367  000070          DECB    $CNVCDE                 ;TYPE THIS DIGIT?
013464  100216                  BPL     7$                      ;BR IF NO
013466  042703  177770          BIC     #177770,R3              ;GET RID OF JUNK
013472  001002                  BNE     4$                      ;TEST FOR 0
013474  005704                  TST     R4                      ;SUPPRESS THIS 0?
013476  001403                  BEQ     5$                      ;BR IF YES
013500  005200          4$:     INC     R0                      ;DON'T SUPPRESS ANYMORE 0'S
013502  052703  000060          BIS     #'0,R3                  ;MAKE THIS DIGIT ASCII
013506  052703  000040  5$:     BIS     #' ,R3                  ;MAKE ASCII IF NOT ALREADY
013512  110367  000232          MOVB    R3,$S                   ;SAVE FOR TYPING
013516  104400  013550          TYPE    ,$S                     ;GO TYPE THIS DIGIT
013522  105367  000024  7$:     DECB    $OCNT                   ;COUNT BY 1
013526  003347                  BGT     2$                      ;BR IF MORE TO DO
```



```
013530  002402                  BLT     6$                      ;BR IF DONE
013532  005304                  INC     R4                      ;INSURE LAST DIGIT ISN'T A BLANK
013534  000740                  BR      2$                      ;GO DO THE LAST DIGIT
013536  012605          6$:     MOV     (SP)+,R5                ;RESTORE R5
013540  012604                  MOV     (SP)+,R4                ;RESTORE R4
013542  012603                  MOV     (SP)+,R3                ;RESTORE R3
013544  012616                  MOV     (SP)+,(SP)              ;SET THE STACK FOR RETURNING
013546  000200                  RTS     R0                      ;RETURN
013550      000         $S:     .BYTE   0                       ;STORAGE FOR ASCII DIGIT
013551      000                 .BYTE   0                       ;TERMINATOR FOR TYPE ROUTINE
013552      000         $OCNT:  .BYTE   0                       ;OCTAL DIGIT COUNTER
013553      000         $OFILL: .BYTE   0                       ;ZERO FILL SWITCH
013554  000000          $OMODE: 0                               ;NUMBER OF DIGITS TO TYPE
```

```
                                    ;*****************************************************************
                                    ;TRAP HANDLER
013556  010046                      $TRAP:  MOV     R0,-(SP)        ;SAVE R0
013560  016000  000002                      MOV     2(SP),R0        ;GET TRAP ADDRESS
013564  005740                              TST     -(R0)           ;BACKUP BY 2
013566  111000                              MOVB    (R0),R0         ;GET RIGHT BYTE OF TRAP
013570  016000  013576                      MOV     $TRPAD(R0),R0   ;INDEX TO TABLE
013574  000200                              RTS     R0              ;GO TO ROUTINE

                                    ;TRAP TABLE
                                    ;       ROUTINE
                                    ;       -------
013576  001110                      $TRPAD: $TYPE   ;CALL=TYPE      TRAP+0(104400)  TTY TYPEOUT ROUTINE
013600  012456                              $READC  ;CALL=GETCHR    TRAP+2(104402)  TTY TYPEIN CHARACTER ROUTINE
013602  012512                              $READS  ;CALL=GETSTR    TRAP+4(104404)  TTY TYPEIN STRING ROUTINE
013604  012314                              $ACCEPT ;CALL=ACCEPT    TRAP+6(104406)  READ AN OCTAL NUMBER FROM TTY
```



```
                                    ;*****************************************************************
                                    ;POWER DOWN ROUTINE
013606  012737  013734  000024      $PWRDN: MOV     #$ILLUP,@#PWRVEC        ;SET FOR FAST UP
013614  012737  000340  000026              MOV     #340,@#PWRVEC+2         ;PRIO:7
013622  010046                              MOV     R0,-(SP)                ;PUSH R0 ON STACK
013624  010146                              MOV     R1,-(SP)                ;PUSH R1 ON STACK
013626  010246                              MOV     R2,-(SP)                ;PUSH R2 ON STACK
013630  010346                              MOV     R3,-(SP)                ;PUSH R3 ON STACK
013632  010446                              MOV     R4,-(SP)                ;PUSH R4 ON STACK
013634  010546                              MOV     R5,-(SP)                ;PUSH R5 ON STACK
013636  010667  000076                      MOV     SP,$SAVR6               ;SAVE SP
013642  012737  013654  000024              MOV     #$PWRUP,@#PWRVEC        ;SET UP VECTOR
013650  000000                              HALT
013652  000776                              BR      .-2                     ;HANG UP
                                    ;*****************************************************************
                                    ;POWER UP ROUTINE
013654  016706  000060              $PWRUP: MOV     $SAVR6,SP               ;GET SP
013660  005067  000054                      CLR     $SAVR6                  ;WAIT LOOP FOR THE TTY
013664  005267  000050              1$:     INC     $SAVR6                  ;WAIT FOR THE INC
013670  001375                              BNE     1$                      ;OF WORD
013672  012605                              MOV     (SP)+,R5                ;POP STACK INTO R5
013674  012604                              MOV     (SP)+,R4                ;POP STACK INTO R4
013676  012603                              MOV     (SP)+,R3                ;POP STACK INTO R3
013700  012602                              MOV     (SP)+,R2                ;POP STACK INTO R2
013702  012601                              MOV     (SP)+,R1                ;POP STACK INTO R1
013704  012600                              MOV     (SP)+,R0                ;POP STACK INTO R0
013706  012737  013606  000024              MOV     #$PWRDN,@#PWRVEC        ;SET UP THE POWER DOWN VECTOR
013714  012737  000340  000026              MOV     #340,@#PWRVEC+2         ;PRIO:7
013722  104400  013742                      TYPE    ,$POWER                 ;POWER FAIL MESSAGE
013726  012716  001706                      MOV     #BEGIN,(SP)             ;RESTART AT BEGIN
013732  000002                              RTI
013734  000000                      $ILLUP: HALT                            ;THE POWER UP SEQUENCE WAS STARTED
013736  000776                              BR      .-2                     ; BEFORE THE POWER DOWN WAS COMPLETE
013740  000000                      $SAVR6: 0                               ;PUT THE SP HERE
013742  005015  047520  042527      $POWER: .ASCIZ  <15><12>"POWER"
013750  000122
                                            .EVEN
```

```
                                        ;*********************************************************************
                                        ;
                                        ;ERROR AND MESSAGE TABLE COMPONENTS
                                        ;
                                        ;*********************************************************************

013752  042524  052123  042040  EM1:    .ASCIZ  /TEST DIDN'T ABORT  /
013760  042111  047516  020124
013766  041101  051117  020124
013774  020040
013776  040506  040524  020114  EM2:    .ASCIZ  /FATAL ERROR TO PROGRAM  /
014004  051105  047522  020122
014012  047524  050040  047522
014020  051107  046501  020040
014026     000
014027     101  047502  052122  EM3:    .ASCIZ  /ABORTED INCORRECTLY  /
014034  042105  044440  041516
014042  051117  042522  052103
014050  054514  020040     000
014055     116  020117  040520  EM4:    .ASCIZ  /NO PARITY MEMORY FOUND BELOW 28K  /
014062  044522  054524  046440
014070  046505  051117  020131
014076  047506  047125  020104
014104  042502  047514  020127
014112  034062  020113  000040
014120  042522  042523  020124  EM5:    .ASCIZ  /RESET DOESN'T WORK  /
014126  047504  051505  023516
014134  020124  047527  045522
014142  020040     000
014145     125  042523  020122  EM6:    .ASCIZ  /USER SELECTED REGISTER NOT PRESENT  /
014152  042523  042514  052103
014160  042105  051040  043505
014166  051511  042524  020122
014174  047516  020124  051120
014202  051505  047105  020124
014210  020040
014212  047516  050040  051101  EM7:    .ASCIZ  /NO PARITY MEMORY FOUND AT ALL  /
014220  052111  020131  042515
014226  047515  054522  043040
014234  052517  042116  040440
014242  020124  046101  020114
014250  020040
014252  044504  047104  052047  EM10:   .ASCIZ  /DIDN'T ABORT OR RECOGNIZE STACK VIOLATION  /
014260  040440  047502  052122
014266  047440  020122  042522
014274  047503  047107  055111
014302  020105  052123  041501
014310  020113  044526  046117
014316  052101  047511  020116
014324  020040
014326  041101  051117  042524  EM11:   .ASCIZ  /ABORTED BUT STACK VIOLATION NOT RECOGNIZED  /
014334  020104  052502  020124
014342  052123  041501  020113
014350  044526  046117  052101
```



```
014356  047511  020116  047516
014364  020124  042522  047503
014372  047107  055111  042105
014400  020040     000
014403     123  040524  045503  EM12:   .ASCIZ  /STACK VIOLATION PICKED UP BUT ABORT NOT RECOGNIZED  /
014410  053040  047511  040514
014416  044524  047117  050040
014424  041511  042513  020104
014432  050125  041040  052125
014440  040040  047502  052122
014446  047040  052117  051040
014454  041505  043517  044516
014462  042532  020104  000040
014470  051120  043517  040522  DM1:    .ASCII  /PROGRAM   REGISTER/<15><12>
014476  020115  020040  042522
014504  044507  052123  051105
014512  005015
014514  020040  041520  020040          .ASCIZ  /  PC     UNDER TEST/
014522  020040  052440  042116
014530  051105  052040  051505
014536  000124
014540  051040  043505  051511  DM2:    .ASCII  / REGISTER/<15><12>
014546  042524  006522     012
014553     125  042116  051105          .ASCIZ  /UNDER TEST/
014560  052040  051505  000124
014566  051120  043517  040522  DM3:    .ASCII  /PROGRAM/<15><12>
014574  006515     012
014577     040  050040  000103          .ASCIZ  /  PC/
014604  051120  043517  040522  DM4:    .ASCII  /PROGRAM   REGISTER    EXPECTED     ACTUAL     EXPECTED   ACTUAL/<15><12>
014612  020115  020040  042522
014620  044507  052123  051105
014626  020040  020040  054105
014634  042520  052103  042105
014640  020040  020040  040440
014650  052103  040525  020114
014656  020040  042440  050130
014664  041505  042524  020104
014672  020040  041501  052524
014700  046101  005015
014704  020040  041520  020040          .ASCIZ  /  PC     UNDER TEST  ADDR, BITS  ADDR, BITS  ABORT PC   ABORT PC/
014712  020040  052440  042116
014720  051105  052040  051505
014726  020124  040440  042104
014734  027122  041040  052111
014742  020123  040440  042104
014750  027122  041040  052111
014756  020123  040440  047502
014764  052122  050040  020103
014772  020040  041101  051117
015000  020124  041520     000
015005     120  047522  051107  DM5:    .ASCII  /PROGRAM   REGISTER    EXPECTED/<15><12>
015012  046501  020040  051040
015020  043505  051511  042524
015026  020122  020040  042440
```

```
  3455  015030  052130  041505  042524
  3456  015042  000504     012
  3457  015044     040  052040  020103          .ASCIZ  /  PC       UNDER TEST   ABORT PC/
  3458  015052  [illegible]  [illegible]  052441
  3459  015060  042116  051105  052040
  3460  015066  051525  [illegible]  020040
  3461  015074  041101  051117  020124
  3462  015102  041520     000
  3463          015106                          .EVEN
  3464  015106  001324  001622  000000  DT1:   .WORD   SHLTAD,PARITY,0
  3465  015114  001622  000000          DT2:   .WORD   PARITY,0
  3466  015120  001324  000000          DT3:   .WORD   SHLTAD,0
  3467  015124  001324  001622  001326  DT4:   .WORD   SHLTAD,PARITY,SGDADR,SGDADR,SGDDAT,SGDDAT,0
  3468  015132  001330  001332  001334
  3469  015140  000000
  3470  015142  001324  001622  001332  DT5:   .WORD   SHLTAD,PARITY,SGDDAT,0
  3471  015150  000000
  3472
  3473          000001                  .END
```



```
A         003752         ABORT     003430         ACCEPT= 104406         A0        004026
A1        004100         A2        004102         B         004236         BEGIN     001706
BIT0    = 000001         BIT00   = 000001         BIT01   = 000002         BIT02   = 000004
BIT03   = 000010         BIT04   = 000020         BIT05   = 000040         BIT06   = 000100
BIT07   = 000200         BIT08   = 000400         BIT09   = 001000         BIT1    = 000002
BIT10   = 002000         BIT11   = 004000         BIT12   = 010000         BIT13   = 020000
BIT14   = 040000         BIT15   = 100000         BIT2    = 000004         BIT3    = 000010
BIT4    = 000020         BIT5    = 000040         BIT6    = 000100         BIT7    = 000200
BIT8    = 000400         BIT9    = 001000         BLKCNT    001032         BPTVEC= 000014
B0        004312         B1        004370         B2        004444         B3        004536
B4        004612         B5        004706         B6        004766         C         005042
CHECKL    011550         COMPUT    003200         CPU40     001642         C0        005116
C1        005210         C2        005304         C3        005360         C4        005436
C5        005514         C6        005570         C7        005644         C8        005702
D         006220         DD        006006         DD0       006064         DD1       006144
DH1       014470         DH2       014520         DH3       014566         DH4       014604
DH5       015005         DISPLA= 177570         DT1       015106         DT2       015114
DT3       015120         DT4       015124         DT5       015142         D0        006076
D1        006156         E         006320         EMTVEC= 000030         EM1       013752
EM10      014252         EM11      014326         EM12      014403         EM2       013776
EM3       014027         EM4       014055         EM5       014120         EM6       014145
EM7       014212         ERRVEC= 000004         E0        006326         E1        006502
E2        006510         F         006634         FINDON    002504         FLAGSC    011474
F0        006604         F1        006774         G         007056         GETCHR= 104402
GETSTR= 104404         GO        002326         G0        007066         H         007140
HOLDLO    010302         H0        007150         H1        007134         H2        007242
H3        007216         H4        007222         INITIA    011506         INTERT    001644
INTVEC    001620         IOTVEC= 000020         KPAR0     000230         KPAR1     000232
KPAR2     000234         KPAR3     000236         KPAR4     000240         KPAR5     000242
KPAR6     000244         KPAR7     000246         KPDR0     000210         KPDR1     000212
KPDR2     000214         KPDR3     000216         KPDR4     000220         KPDR5     000222
KPDR6     000224         KPDR7     000226         KTTIME    002324         L         007320
LEAFCN    001636         M         007374         MEMAD     001640         MRK0      006324
MRK1      006506         MSGTYP    002372         MSREGF    001626         N       = 000067
NEWSTK    031476         NEXT1     002514         NN        007450         NOMORE    003172
NOREG     003144         NTERAD    001446         NTER0     001450         NTER1     001452
NTER10    001470         NTER11    001472         NTER12    001474         NTER2     001454
NTER3     001456         NTER4     001460         NTER5     001462         NTER6     001464
NTER7     001466         ONETRY    003500         P         007526         PARCOR    003112
PARITY    001622         PARTST    003222         PC      =%000007         PS      = 177776
PSPCOR    001624         PSW     = 177776         PWRVEC= 000024         R         007604
RED       010400         RESTAR    011464         RESTOR    001634         RESVEC= 000010
R0      =%000000         R1      =%000001         R2      =%000002         R3      =%000003
R4      =%000004         R5      =%000005         R6      =%000006         R7      =%000007
SAVLOC    010126         SEGVEC    000250         SP      =%000006         SR0       000204
SR2       000206         STACK   = 001100         START     003620         SWR     = 177570
SW0     = 000001         SW00    = 000001         SW01    = 000002         SW02    = 000004
SW03    = 000010         SW04    = 000020         SW05    = 000040         SW06    = 000100
SW07    = 000200         SW08    = 000400         SW09    = 001000         SW1     = 000002
SW10    = 002000         SW11    = 004000         SW12    = 010000         SW13    = 020000
SW14    = 040000         SW15    = 100000         SW2     = 000004         SW3     = 000010
SW4     = 000020         SW5     = 000040         SW6     = 000100         SW7     = 000200
SW8     = 000400         SW9     = 001000         T         007750         TBITVE= 000014
TKB       012622         TKS       012620         TRAPVE= 000034         TRTVEC= 000014
```

MAINDEC-11-[illegible] MACY11 27(655) 22-MAY-73 10:33 PAGE 75
DCKBRE.P11 SYMBOL TABLE

TST1	[illegible]	TST10	[illegible]	TST11	004120	TST12	004222
TST13	[illegible]	TST14	[illegible]	TST15	004110	TST16	004450
TST17	[illegible]	TST2	[illegible]	TST20	004652	TST21	006726
TST22	[illegible]	TST23	[illegible]	TST24	005136	TST25	005250
TST26	[illegible]	TST27	[illegible]	TST3	003272	TST30	005456
TST31	[illegible]	TST32	[illegible]	TST33	[illegible]	TST34	005762
TST35	[illegible]	TST36	[illegible]	TST37	006176	TST4	003334
TST40	006350	TST41	[illegible]	TST42	[illegible]	TST43	007014
TST44	007104	TST45	007170	TST46	[illegible]	TST47	007340
TST5	003374	TST50	007414	TST51	007470	TST52	007536
TST53	007604	TST54	007670	TST55	007774	TST56	010076
TST57	010252	TST6	003716	TST60	010426	TST61	010530
TST62	010636	TST63	010732	TST64	011076	TST65	011234
TST66	011372	TST7	003772	TYPE =	104400	TYPERR	012776
T0	007750	USCRTY	001632	V	010452	VECSET	011450
V0	010256	V1	010162	V2	010166	V3	010336
V4	010342	W	010516	WHICH1	005376	XX	010616
Y	010720	YELLOW	010724	Z0	011054	Z1	011214
Z2	011352	$ACCEP	012314	$ADACR	001330	$BDDAT	001334
$BELL	012056	$B2OCT	013326	$B2O1	013376	$B2O16	013362
$CR =	[illegible]	$CR2 =	[illegible]	$CR3 =	000013	$CR4 =	020015
$CRLF	012025	$DB2D4	013602	$DB2D0	012646	$DTP	011772
$ERFLG	001314	$ERRTB	001510	$ERTTL	001512	$ESCAP	012774
$FILLS	001330	$GDADR	001310	$GDDAT	001332	$HLT	012634
$HLTAD	001374	$ICNT	001334	$ILLUP	013734	$ITEMB	001322
$INTOUT	013310	$KT11	002304	$LF	012450	$LPADR	001306
$LPERR	001312	$LSTBL	012746	$MXCNT	013312	$NULL	001104
$OCNT	013552	$OMODE	013354	$OVER	012276	$PASS	001300
$POWER	013742	$PWRDN	013606	$PWRUP	013654	$QUES	012452
$RDSZ =	000010	$READC	013456	$RECADS	012512	$RECAD	001336
$REG0	001340	$REG1	001342	$REG10	001360	$REG11	001362
$REG12	001364	$REG2	001344	$REG3	001346	$REG4	001350
$REG5	001352	$REG6	001354	$REG7	001356	$SAVR6	013740
$SCOPE	012662	$SETAD	001416	$SETUP =	000017	$SET0	001420
$SET1	001422	$SET10	001440	$SET11	001442	$SET12	001444
$SET2	001424	$SET3	001426	$SET4	001430	$SET5	001432
$SET6	001434	$SET7	001436	$SIZE	013124	$SS =	000001
$STUP =	177777	$SVLAD	012654	$SWR =	167400	$TIMES	012310
$TMPAD	001414	$TMP0	001366	$TMP1	001370	$TMP10	001406
$TMP11	001410	$TMP12	001412	$TMP2	001372	$TMP3	001374
$TMP4	001376	$TMP5	001400	$TMP6	001402	$TMP7	001404
$TN =	000021	$TPB	001102	$TPFLG	001106	$TPS	001100
$TRAP	013556	$TRP =	000010	$TRPAD	013576	$TSTNM	001302
$TTYIN	012624	$TYPE	001110	$XTSTR	012070	$0FILL	013553
. =	015152						

ERRORS DETECTED: 0

*DCKBRE,DCKBRE/SOL=DCKBRE/DS:CRF2
RUN-TIME: 37 30 0 SECONDS
CORE USED: 10K