```
   SSSSSSSSSSS  YYY         YYY    SSSSSSSSSSS  LLL              OOOOOOOOO       AAAAAAAAA
   SSSSSSSSSSS  YYY         YYY    SSSSSSSSSSS  LLL              OOOOOOOOO       AAAAAAAAA
   SSSSSSSSSSS  YYY         YYY    SSSSSSSSSSS  LLL              OOOOOOOOO       AAAAAAAAA
SSS             YYY         YYY SSS             LLL           OOO         OOO   AAA         AAA
SSS             YYY         YYY SSS             LLL           OOO         OOO   AAA         AAA
SSS             YYY         YYY SSS             LLL           OOO         OOO   AAA         AAA
SSS                YYY   YYY    SSS             LLL           OOO         OOO   AAA         AAA
SSS                YYY   YYY    SSS             LLL           OOO         OOO   AAA         AAA
SSS                YYY   YYY    SSS             LLL           OOO         OOO   AAA         AAA
   SSSSSSSSS          YYY          SSSSSSSSS     LLL           OOO         OOO   AAA         AAA
   SSSSSSSSS          YYY          SSSSSSSSS     LLL           OOO         OOO   AAA         AAA
   SSSSSSSSS          YYY          SSSSSSSSS     LLL           OOO         OOO   AAA         AAA
            SSS       YYY                   SSS  LLL           OOO         OOO   AAAAAAAAAAAAAAA
            SSS       YYY                   SSS  LLL           OOO         OOO   AAAAAAAAAAAAAAA
            SSS       YYY                   SSS  LLL           OOO         OOO   AAAAAAAAAAAAAAA
            SSS       YYY                   SSS  LLL           OOO         OOO   AAA         AAA
            SSS       YYY                   SSS  LLL           OOO         OOO   AAA         AAA
            SSS       YYY                   SSS  LLL           OOO         OOO   AAA         AAA
SSSSSSSSSSS           YYY SSSSSSSSSSS     LLLLLLLLLLLLLLL        OOOOOOOOO     AAA         AAA
SSSSSSSSSSS           YYY SSSSSSSSSSS     LLLLLLLLLLLLLLL        OOOOOOOOO     AAA         AAA
SSSSSSSSSSS           YYY SSSSSSSSSSS     LLLLLLLLLLLLLLL        OOOOOOOOO     AAA         AAA
```

```
  AAAAAA    DDDDDDDD    PPPPPPPP      SSSSSSSS  UU      UU  BBBBBBBB    77777777    5555555555    000000
  AAAAAA    DDDDDDDD    PPPPPPPP      SSSSSSSS  UU      UU  BBBBBBBB    77777777    5555555555    000000
AA      AA  DD      DD  PP      PP  SS            UU      UU  BB      BB          77  55            00    00
AA      AA  DD      DD  PP      PP  SS            UU      UU  BB      BB          77  55            00    00
AA      AA  DD      DD  PP      PP  SS            UU      UU  BB      BB          77  555555        00    0000
AA      AA  DD      DD  PP      PP  SS            UU      UU  BB      BB          77  555555        00    0000
AA      AA  DD      DD  PPPPPPPP      SSSSSS      UU      UU  BBBBBBBB        77      55    55    00  00  00
AA      AA  DD      DD  PPPPPPPP      SSSSSS      UU      UU  BBBBBBBB        77      55    55    00  00  00
AAAAAAAAAA  DD      DD  PP                  SS    UU      UU  BB      BB    77          55    55  0000    00
AAAAAAAAAA  DD      DD  PP                  SS    UU      UU  BB      BB    77          55    55  0000    00
AA      AA  DD      DD  PP                  SS    UU      UU  BB      BB  77    55    55  55    55  00      00    ....
AA      AA  DD      DD  PP                  SS    UU      UU  BB      BB  77    55    55  55    55  00      00    ....
AA      AA  DDDDDDDD    PP            SSSSSSSS    UUUUUUUUUU  BBBBBBBB  77      555555      000000        ....
AA      AA  DDDDDDDD    PP            SSSSSSSS    UUUUUUUUUU  BBBBBBBB  77      555555      000000        ....
```

```
LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

Table of contents

```
0000     1          .NOSHOW CONDITIONALS
0000     5
0000     7          .TITLE  ADPSUB750 - ADAPTER SUBROUTINES FOR VAX 11/750
0000     9
0000    13
0000    17
0000    21
0000    22          .IDENT  'V04-000'
0000    23
0000    24 ;*****************************************************************************
0000    25 ;*                                                                           *
0000    26 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    27 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000    28 ;*  ALL RIGHTS RESERVED.                                                     *
0000    29 ;*                                                                           *
0000    30 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    31 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    32 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    33 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    34 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    35 ;*  TRANSFERRED.                                                             *
0000    36 ;*                                                                           *
0000    37 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    38 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    39 ;*  CORPORATION.                                                             *
0000    40 ;*                                                                           *
0000    41 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    42 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    43 ;*                                                                           *
0000    44 ;*                                                                           *
0000    45 ;*****************************************************************************
0000    46 ;
0000    47 ; Facility: System bootstrapping and initialization
0000    48 ;
0000    49 ; Abstract: This module contains initialization routines that are loaded
0000    50 ;           during system initialization (rather than linked into the system).
0000    51 ;
0000    52 ; Environment: Mode = KERNEL, Executing on INTERRUPT stack, IPL=31
0000    53 ;
0000    54 ; Author:  Kerbey T. Altmann            Creation date: 30-Oct-1982
0000    55 ;
0000    56 ; Modification history:
0000    57 ;
0000    58 ;       V03-007 TCM0002         Trudy C. Matthews       04-Jun-1984
0000    59 ;               Include more 780-specific code for the 11/790 version of
0000    60 ;               this routine.
0000    61 ;
0000    62 ;       V03-006 KPL0001         Peter Lieberwirth       12-Apr-1984
0000    63 ;               Init ADP$L_SHB properly again; V03-004 ASSUMEd this field
0000    64 ;               was at a certain constant offset, and a change to the ADP
0000    65 ;               moved it.  Note - this is a 780 change only.
0000    66 ;
0000    67 ;       V03-005 KDM0081         Kathleen D. Morse       13-Sep-1983
0000    68 ;               Create version for Micro-VAX I.
0000    69 ;
0000    70 ;       V03-004 ROW0196         Ralph O. Weber          27-JUL-1983
0000    71 ;               Correct INI$MPMADP so the ADP$L_SHB is correctly initialized
```

```
0000     72 ;                  to zero.
0000     73 ;
0000     74 ;        V03-003 MSH0001         Maryann Hinden          06-Dec-1982
0000     75 ;                Add initialization for DW750.
0000     76 ;
0000     77 ;        V03-002 ROW0142         Ralph O. Weber          23-NOV-1982
0000     78 ;                Correct JMP in multiport memory interrupt dispatching code
0000     79 ;                prototype, MPMINTD, to a JSB.  MA$INT expects to receive
0000     80 ;                control via a JSB.
0000     81 ;
0000     82 ;        V03-001 TCM0001         Trudy C. Matthews       8-Nov-1982
0000     83 ;                Initialize field ADP$L_AVECTOR in INI$MPMADP.
0000     84 ;
0000     85 ;--
```

```
                     0000     90
                     0000     94
00000000             0000     96          C780_LIKE = 0
                     0000     98
                     0000    102
                     0000    106 ;
                     0000    107 ; MACRO LIBRARY CALLS
                     0000    108 ;
                     0000    109          $ADPDEF                         ; Define ADP offsets.
                     0000    110          $CRBDEF                         ; Define CRB offsets.
                     0000    111          $DCDEF                          ; Define AT codes.
                     0000    112          $DDBDEF                         ; Define DDB offsets.
                     0000    113          $DDTDEF                         ; Define DDT offsets.
                     0000    114          $DYNDEF                         ; Define data structure type codes.
                     0000    115          $IDBDEF                         ; Define interrupt dispatcher offsets.
                     0000    116          $MBADEF                         ; Define MASSBUS registers.
                     0000    117          $MCHKDEF                        ; Define machine check masks.
                     0000    118          $MPMDEF                         ; Define multi-port memory.
                     0000    119          $NDTDEF                         ; Define nexus device types.
                     0000    120          $PRDEF                          ; Define IPR numbers.
                     0000    121          $PTEDEF                         ; Define Page Table Entry bits.
                     0000    122          $RPBDEF                         ; Define Restart Parameter Block fields.
                     0000    123          $SSDEF                          ; Define system service codes.
                     0000    124          $UBADEF                         ; Define UBA register offsets.
                     0000    125          $UBIDEF                         ; Define UNIBUS interconnect
                     0000    126                                          ;  register offsets.
                     0000    127          $UCBDEF                         ; Define unit control block.
                     0000    128          $VADEF                          ; Define virtual address fields.
                     0000    129          $VECDEF                         ; Define vec offsets.
                     0000    130
                     0000    141
                     0000    143          $UASDEF                         ; DEFINE DW750 IPEC REGISTERS
                     0000    145
00000000             0000    146          .PSECT  SYSLOA,LONG
```

```
                                B 15
0000   148          .SBTTL  CI$INT - CI INTERRUPT HANDLER
0000   149 ;+
0000   150 ; CI$INT - CI INTERRUPT HANDLER
0000   151 ;
0000   152 ;       THIS MODULE IS A DUMMY CI32 INTERRUPT HANDLER WHICH IS USED
0000   153 ;       UNTIL THE REAL CI DRIVER (PADRIVER) IS LOADED. IT ALSO CONTAINS
0000   154 ;       A DUMMY CI32 CONTROLLER INITIALIZATION  ENTRY POINT.
0000   155 ;
0000   156 ; INPUTS:
0000   157 ;
0000   158 ;       THE STACK ON ENTRY IS AS FOLLOWS:
0000   159 ;
0000   160 ;                 0(SP)           ADDRESS OF IDB ADDRESS
0000   161 ;       4(SP) - 16(SP)            SAVED R2 - R5
0000   162 ;                20(SP)           INTERRUPT PC
0000   163 ;                24(SP)           INTERRUPT PSL
0000   164 ;
0000   165 ; OUTPUTS:
0000   166 ;
0000   167 ;       NONE
0000   168 ;
0000   169 ; SIDE EFFECTS:
0000   170 ;
0000   171 ;       INTERRUPTS ARE DISABLED ON THE CI32
0000   172 ;-
0000   173
0000   176
0000   177 ;
0000   178 ; $PAREGDEF -- Define offsets to CI registers and fields in the registers.
0000   179 ;
0000   180
0000   181         $DEFINI PAREG
0000   182
0000   183         $DEF    PA_CNF  .BLKL   1               ; Configuration register
0004   184
0004   185         _VIELD  PA_CNF,0,<-                     ; Define config register fields:
0004   186         <ADPTYP,8,M>,-                          ;  Adapter type code
0004   187         <PFD,,M>,-                              ;  Powerfail disable
0004   188         <TDEAD,,M>,-                            ;  Transmit dead
0004   189         <TFAIL,,M>,-                            ;  Transmit fail
0004   190         <,5>,-                                  ;  5 unused bits
0004   191         <CRD,,M>,-                              ;  CRD on port init'd read
0004   192         <RDS,,M>,-                              ;  RDS on port init'd read
0004   193         <CXTER,,M>,-                            ;  SBI error confirm
0004   194         <RDTO,,M>,-                             ;  Port init'd read timeout on SBI
0004   195         <CSTMO,,M>,-                            ;  Port init'd command xmit timeout
0004   196         <,1>,-                                  ;  1 unused bit
0004   197         <PUP,,M>,-                              ;  Adapter power up
0004   198         <PDN,,M>,-                              ;  Adaptor power down
0004   199         >                                       ;
0004   200
0004   201         $DEF    PA_PMC  .BLKL   1               ; Port maint control/status register
0008   202
0008   203         _VIELD  PA_PMC,0,<-                     ; Define register fields:
0008   204         <MIN,,M>,-                              ;  Maint initialized
0008   205         <MTD,,M>,-                              ;  Maint timer disable
0008   206         <MIE,,M>,-                              ;  Maint interrupt enable
```

```
                                0008   207          <MIF,,M>,-                              ;  Maint intterupt flag
                                0008   208          >                                       ;
                                0008   209
                                0008   210          $DEFEND PAREG
                                0000   211
                                0000   212 CI$INT::
                     53   9E    D0   0000   213          MOVL    @(SP)+,R3                  ; GET ADDRESS OF IDB
                     54   63    D0   0003   214          MOVL    IDB$L_CSR(R3),R4           ; GET ADDRESS OF FIRST CSR
                64   00400000  8F    D0   0006   215          MOVL    #PA_CNF_M_PUP,PA_CNF(R4)   ; CLEAR POWER UP
                64   00800000  8F    D0   000D   216          MOVL    #PA_CNF_M_PDN,PA_CNF(R4)   ; CLEAR POWER DOWN
                   04   A4      01    D0   0014   217          MOVL    #PA_PMC_M_MIN,PA_PMC(R4)   ; SET MAINTENCE INITIALIZE
                     52   8E    7D   0018   218          MOVQ    (SP)+,R2                   ; RESTORE REGISTERS
                     54   8E    7D   001B   219          MOVQ    (SP)+,R4
                                02   001E   220          REI
                                     001F   221
                                     001F   224
                                     001F   225 CI$INITIAL::                                ; CONTROLLER INITIALIZATION
                                     001F   226 CI$SHUTDOWN::                               ; CONTROLLER SHUTDOWN
                                     001F   227
                                     001F   230
                   04   A4      01    D0   001F   231          MOVL    #PA_PMC_M_MIN,PA_PMC(R4)   ; SET MAINTENCE INITIALIZE
                                     0023   234
                                05   0023   235          RSB
```

```
                        0024   237          .SBTTL  DR$INT - DR INTERRUPT HANDLER
                        0024   238 ;+
                        0024   239 ; DR$INT - DR INTERRUPT HANDLER
                        0024   240 ;
                        0024   241 ;       THIS MODULE IS A DUMMY DR32 INTERRUPT HANDLER WHICH IS USED
                        0024   242 ;       UNTIL THE REAL DR DRIVER (XFDRIVER) IS LOADED. IT ALSO CONTAINS
                        0024   243 ;       A DUMMY DR32 CONTROLLER INITIALIZATION  ENTRY POINT.
                        0024   244 ;
                        0024   245 ; INPUTS:
                        0024   246 ;
                        0024   247 ;       THE STACK ON ENTRY IS AS FOLLOWS:
                        0024   248 ;
                        0024   249 ;                 0(SP)           ADDRESS OF IDB ADDRESS
                        0024   250 ;       4(SP) - 16(SP)            SAVED R2 - R5
                        0024   251 ;                20(SP)           INTERRUPT PC
                        0024   252 ;                24(SP)           INTERRUPT PSL
                        0024   253 ;
                        0024   254 ; OUTPUTS:
                        0024   255 ;
                        0024   256 ;       NONE
                        0024   257 ;
                        0024   258 ; SIDE EFFECTS:
                        0024   259 ;
                        0024   260 ;       INTERRUPTS ARE DISABLED ON THE DR32
                        0024   261 ;-
                        0024   262
                        0024   265
                        0024   266 ;
                        0024   267 ; DR32 DCR REGISTER DEFINITIONS
                        0024   268 ;-
                        0024   269
                        0024   270          $DEFINI DR
                        0000   271 $DEF     DR_DCR          .BLKL   1       ; DR32 CONTROL REGISTER
                        0004   272          _VIELD  DR_DCR,0,<-
                        0004   273                  <ADPTYP,8>,-            ; ADAPTER TYPE
                        0004   274                  <ID2ERR,,M>,-           ; ID2 ERROR
                        0004   275                  <ID2TOS,2>,-            ; ID2 TIME-OUT STATUS
                        0004   276                  <,1>,-                  ; RESERVED
                        0004   277                  <ID1ERR,,M>,-           ; ID1 ERROR
                        0004   278                  <ID1TOS,2>,-            ; ID1 TIME-OUT STATUS
                        0004   279                  <RDS,,M>,-              ; READ DATA SUBSTITUTE
                        0004   280                  <CRD,,M>,-              ; CORRECTED READ DATA
                        0004   281                  <DCRHLT,,M>,-           ; DCR HALT
                        0004   282                  <DCRABT,,M>,-           ; DCR ABORT INTERRUPT
                        0004   283                  <PKTINT,,M>,-           ; PACKET INTERRUPT
                        0004   284                  <INTENB,,M>,-           ; INTERRUPT ENABLE
                        0004   285                  <,1>,-                  ; RESERVED
                        0004   286                  <PWR_UP,,M>,-           ; ADAPTER POWER UP
                        0004   287                  <PWR_DN,,M>,-           ; ADAPTER POWER DOWN
                        0004   288                  <EXTABT,,M>,-           ; EXTERNAL ABORT
                        0004   289                  <,1>,-                  ; RESERVED
                        0004   290                  <IMPDEP,6>,-            ; IMPLEMENTATION DEPENDENT BITS
                        0004   291          >
                        0004   292
                        0004   293 ; DCR CONTROL FIELD A CODES (USED WHEN WRITING TO DCR)
                        0004   294
00000100                0004   295          DCR_K_CLRPWRUP=^X100
```

```
                     00000200  0004    296          DCR_K_CLRPWRDN=^X200                    ; CLEAR POWER DOWN
                     00000300  0004    297          DCR_K_CLREXTABT=^X300                   ; CLEAR EXTERNAL ABORT
                     00000400  0004    298          DCR_K_CLRABTINT=^X400                   ; CLEAR ABORT INTERRUPT
                     00000500  0004    299          DCR_K_CLRINTENB=^X500                   ; CLEAR INTERRUPT ENABLE
                     00000600  0004    300          DCR_K_SETINTENB=^X600                   ; SET INTERRUPT ENABLE
                     00000700  0004    301          DCR_K_CLRHLT=^X700                      ; CLEAR HALT
                               0004    302
                               0004    303  ; DCR CONTROL FIELD B CODES (USED WHEN WRITING TO DCR)
                               0004    304
                     00001000  0004    305          DCR_K_CLRCRD=^X1000                     ; CLEAR CRD
                     00002000  0004    306          DCR_K_SETEXTABT=^X2000                  ; SET EXTERNAL ABORT
                     00003000  0004    307          DCR_K_CLRPKTINT=^X3000                  ; CLEAR PACKET INTERRUPT
                     00004000  0004    308          DCR_K_RESET=^X4000                      ; RESET
                     00005000  0004    309          DCR_K_SETOSQTST=^X5000                  ; SET OSEQ TEST
                     00006000  0004    310          DCR_K_CLROSQTST=^X6000                  ; CLEAR OSEQ TEST
                               0004    311          $DEFEND DR
                               0024    312
                               0024    313  DR$INT::
                 53    9E    D0  0024    314          MOVL    @(SP)+,R3                       ; GET ADDRESS OF IDB
                 54    63    D0  0027    315          MOVL    IDB$L_CSR(R3),R4                ; GET ADDRESS OF FIRST CSR
          64  00000100 8F    D0  002A    316          MOVL    #DCR_K_CLRPWRUP,DR_DCR(R4)      ; CLEAR POWER UP
          64  00000200 8F    D0  0031    317          MOVL    #DCR_K_CLRPWRDN,DR_DCR(R4)      ; CLEAR POWER DOWN
                 52    8E    7D  0038    318          MOVQ    (SP)+,R2                        ; RESTORE REGISTERS
                 54    8E    7D  003B    319          MOVQ    (SP)+,R4
                           02    003E    320          REI
                               003F    321
                               003F    324
                               003F    325  DR$INITIAL::                                    ; CONTROLLER INITIALIZATION
                               003F    326  DR$SHUTDOWN::                                   ; CONTROLLER SHUTDOWN
                               003F    327
                               003F    330
          64    4000 8F    3C  003F    331          MOVZWL  #DCR_K_RESET,(R4)               ; RESET DR (R4 POINTS TO CSR)
                               0044    334
                           05  0044    335          RSB
```

```
                        0045   337          .SBTTL  UBA$INITIAL - CPU-DEPENDENT UNIBUS ADAPTER INITIALIZATION
                        0045   338 ;+
                        0045   339 ; UBA$INITIAL - UNIBUS ADAPTER INITIALIZATION
                        0045   340 ;
                        0045   341 ; THIS ROUTINE IS CALLED VIA A JSB INSTRUCTION AT SYSTEM STARTUP AND AFTER
                        0045   342 ; A POWER RECOVERY RESTART TO ALLOW INITIALIZATION OF UNIBUS ADAPTERS.
                        0045   343 ;       (POWERFAIL AND INITADP)
                        0045   344 ;
                        0045   345 ; INPUTS:
                        0045   346 ;
                        0045   347 ;       R2 = ADDRESS OF ADAPTER CONTROL BLOCK  (11/780 AND 11/750)
                        0045   348 ;       R4 = ADDRESS OF UNIBUS ADAPTER CONFIGURATION STATUS REGISTER (11/780)
                        0045   349 ;
                        0045   350 ;       ALL INTERRUPTS ARE LOCKED OUT.
                        0045   351 ;
                        0045   352 ; OUTPUTS:
                        0045   353 ;
                        0045   354 ;       THE UNIBUS ADAPTER IS INITIALIZED AND INTERRUPTS ARE ENABLED.
                        0045   355 ;-
                        0045   356
                        0045   357 UBA$INITIAL::                               ;UNIBUS ADAPTER INITIALIZATION
                        0045   358
                        0045   373
                        0045   375
          50   0C A2    3C   0045   376          MOVZWL  ADP$W_TR(R2),R0         ;GET TR NUMBER
               50  09    B1   0049   377          CMPW    #9,R0                   ;IS THIS FOR ADAPTER AT TR#9?
                   06    12   004C   378          BNEQ    10$                     ;IF NOT, DON'T BOTHER
       50 B2   1000 8F    A8   004E   379          BISW    #UAS$M_IP_CR1_PIE,-
                        0054   380                  @ADP$L_UBASCB+12(R2)    ;SET POWERFAIL INT ENABLE IN IPEC REG
                        0054   381
                        0054   383
                        0054   384 10$:                                        ;NO SPECIAL INIT FOR 11/730 OR uVAX I
                05      0054   385          RSB                             ;
                        0055   386 ;
                        0055   387 ; IGNORE UNEXPECTED UNIBUS INTERRUPTS
                        0055   388 ;
                        0055   389
                        0055   390          .ALIGN  LONG
                        0058   391
                        0058   392 UBA$INTO::                                 ; PASSIVE RELEASES THROUGH VECTOR 0
                        0058   393
00000000'9F     D6      0058   394          INCL    @#IO$GL_UBA_INTO        ; COUNT THEM
          00    11      005E   395          BRB     UBA_UNEXINT             ; JOIN COMMON CODE, VECTORS ARE ALLIGNED
                        0060   396
                        0060   397          .ALIGN  LONG
                        0060   398
                        0060   399 ;
                        0060   400 ; NOTE: UBA$UNEXINT is the label in the EXEC that is a JMP @#UBA_UNEXINT.
                        0060   401 ;       This seeming duplicity is necesary since there is code that must
                        0060   402 ;       refer to the EXEC address from within the SYSLOA image.
                        0060   403 ;
                        0060   404 UBA_UNEXINT::                               ; UNEXPECTED INTERRUPT CODE
                        0060   405
                02      0060   414          REI                             ; IGNORE INTERRUPT
```

```
                     0061   418          .SBTTL  MASSBUS ADAPTER INTERRUPT DISPATCHER
                     0061   419 ;+
                     0061   420 ; MBA$INT - MASSBUS ADAPTER INTERRUPT DISPATCHER
                     0061   421 ;
                     0061   422 ; THIS ROUTINE IS ENTERED VIA A JSB INSTRUCTION WHEN AN INTERRUPT OCCURS
                     0061   423 ; ON A MASSBUS ADAPTER. THE STATE OF THE STACK ON ENTRY IS:
                     0061   424 ;
                     0061   425 ;       00(SP) = ADDRESS OF IDB ADDRESS.
                     0061   426 ;       04(SP) = SAVED R2.
                     0061   427 ;       08(SP) = SAVED R3.
                     0061   428 ;       12(SP) = SAVED R4.
                     0061   429 ;       16(SP) = SAVED R5.
                     0061   430 ;       20(SP) = INTERRUPT PC.
                     0061   431 ;       24(SP) = INTERRUPT PSL.
                     0061   432 ;
                     0061   433 ; INTERRUPT DISPATCHING OCCURS AS FOLLOWS:
                     0061   434 ;
                     0061   435 ;       IF THE INTERRUPTING ADAPTER IS CURRENTLY OWNED AND THE OWNER UNIT
                     0061   436 ;       IS EXPECTING AN INTERRUPT, THEN THAT UNIT IS DISPATCHED FIRST. ALL
                     0061   437 ;       OTHER UNITS ARE DISPATCHED BY READING THE ATTENTION SUMMARY REG-
                     0061   438 ;       ISTER AND SCANNING FOR UNITS THAT HAVE ATTENTION SET. AS EACH UNIT
                     0061   439 ;       IS FOUND, ITS ATTENTION SUMMARY BIT IS CLEARED AND THEN A TEST IS
                     0061   440 ;       MADE TO DETERMINE IF AN INTERRUPT IS EXPECTED ON THE UNIT. IF YES,
                     0061   441 ;       THEN THE DRIVER IS CALLED AT ITS INTERRUPT RETURN ADDRESS. ELSE
                     0061   442 ;       THE DRIVER IS CALLED AT ITS UNSOLICITED INTERRUPT ADDRESS. AS EACH
                     0061   443 ;       CALL TO THE DRIVER RETURNS, THE ATTENTION SUMMARY REGISTER IS RE-
                     0061   444 ;       READ AND AN ATTEMPT IS MADE TO FIND ANOTHER UNIT TO DISPATCH. WHEN
                     0061   445 ;       NO UNITS REQUESTING ATTENTION REMAIN, THE INTERRUPT IS DISMISSED.
                     0061   446 ;-
                     0061   447
                     0061   448          .ALIGN  LONG
                     0064   449
                     0064   450 MBA$INT::                                    ;MASSBUS ADAPTER INTERRUPT DISPATCHER
          53    00 BE    D0  0064   451          MOVL    @(SP),R3            ;GET ADDRESS OF IDB
             54    63    D0  0068   452          MOVL    IDB$L_CSR(R3),R4    ;GET ADDRESS OF CONFIGURATION STATUS REGISTE
                               006B   453
                               006B   461
      00800000 8F    D3      006B   462          BITL    #MBA$M_SR_CBHUNG,-
                  08 A4               0071   463                           MBA$L_SR(R4)        ;CHECK FOR MBA HUNG
                     61    12  0073   464          BNEQ    50$                 ;BRANCH IF HUNG
                               0075   465
                               0075   467
          55    04 A3    D0  0075   468          MOVL    IDB$L_OWNER(R3),R5  ;GET OWNER UNIT UCB ADDRESS
                     0A    13  0079   469          BEQL    10$                 ;IF EQL NO OWNER
       52    0090 C5    9A  007B   470          MOVZBL  UCB$B_SLAVE(R5),R2  ;GET OWNER SLAVE CONTROLLER NUMBER
     21 64 A5    01    E0  0080   471          BBS     #UCB$V_INT,UCB$W_STS(R5),20$ ;IF SET, INTERRUPT EXPECTED
          53    00 BE    D0  0085   472 10$:     MOVL    @(SP),R3            ;RETRIEVE ADDRESS OF IDB
             54    63    D0  0089   473          MOVL    IDB$L_CSR(R3),R4    ;RETRIEVE MBA CONFIGURATION REGISTER ADDRESS
          08 A4    00    D2  008C   474          MCOML   #0,MBA$L_SR(R4)     ;CLEAR ALL MBA STATUS BITS
       52    0410 C4    D0  0090   475          MOVL    MBA$L_AS(R4),R2     ;READ ATTENTION SUMMARY REGISTER
52     52    08    00    EA  0095   476          FFS     #0,#8,R2,R2         ;FIND FIRST UNIT REQUESTING ATTENTION
                     0A    12  009A   477          BNEQ    20$                 ;IF NEQ UNIT FOUND
                  5E    04    C0  009C   478          ADDL    #4,SP               ;REMOVE IDB ADDRESS FROM STACK
                  52    8E    7D  009F   479          MOVQ    (SP)+,R2            ;RESTORE REGISTERS
                  54    8E    7D  00A2   480          MOVQ    (SP)+,R4            ;
                        02        00A5   481          REI                         ;
                                  00A6   482
```

```
                55   18 A342   D0   00A6   483 20$:    MOVL    IDB$L_UCBLST(R3)[R2],R5 ;GET ADDRESS OF UCB OR INTERRUPT DISPATCHER
                       22 55    E8   00AB   484          BLBS    R5,40$                  ;IF LBS INTERRUPT DISPATCHER FOR MULTI-
                                 00AE   485                                              ; DEVICE CONTROLLER
          0410 C4    01   52    78   00AE   486          ASHL    R2,#1,MBA$L_AS(R4)      ;CLEAR ATTENTION SUMMARY BIT
                          55    D5   00B4   487          TSTL    R5                      ;SEE IF UCB DEFINED
                          CD    13   00B6   488          BEQL    10$                     ;IF EQL NONE DEFINED
              09 64 A5    01    E5   00B8   489          BBCC    #UCB$V_INT,UCB$W_STS(R5),30$ ;IF CLR, INTERRUPT NOT EXPECTED
                53   10 A5    7D   00BD   490          MOVQ    UCB$L_FR3(R5),R3        ;RESTORE DRIVER CONTEXT
                        0C B5    16   00C1   491          JSB     @UCB$L_FPC(R5)          ;CALL DRIVER AT INTERRUPT RETURN ADDRESS
                          BF    11   00C4   492          BRB     10$                     ;
                                 00C6   493
                53   0088 C5    D0   00C6   494 30$:    MOVL    UCB$L_DDT(R5),R3        ;GET ADDRESS OF DDT
                        04 B3    16   00CB   495          JSB     @DDT$L_UNSOLINT(R3)     ;CALL UNSOLICITED INTERRUPT ROUTINE
                          B5    11   00CE   496          BRB     10$                     ;
                                 00D0   497
                          7E    DC   00D0   498 40$:    MOVPSL  -(SP)                   ;READ CURRENT PSL
                          75    16   00D2   499          JSB     -(R5)                   ;CALL SLAVE CONTROLLER INTERRUPT DISPATCHER
                          AF    11   00D4   500          BRB     10$                     ;
                                 00D6   501
                                 00D6   513
                                 00D6   514 ;
                                 00D6   515 ; IN CASE OF CBHUNG SAVE MBA INFORMATIION FOR BUGCHECK LOG AND
                                 00D6   516 ; BUGCHECK.  CBHUNG IS IMPLEMENTED ONLY ON THE VAX 11/750 CPU.
                                 00D6   517 ; IT MEANS THAT AN ACCESS TO A REGISTER OF AN EXISTENT CONTROLLER
                                 00D6   518 ; FAILED TO COMPLETE IN 1.5 USEC.
                                 00D6   519 ;
                                 00D6   520
             55    04 A3    D0   00D6   521 50$:    MOVL    IDB$L_OWNER(R3),R5      ;SAVE OWNER UCB IF ANY
             50    08 A4    D0   00DA   522          MOVL    MBA$L_SR(R4),R0         ;SAVE MBA STATUS REGISTER,
                51    64    D0   00DE   523          MOVL    MBA$L_CSR(R4),R1        ; CONFIGURATION REGISTER,
             52    14 A4    D0   00E1   524          MOVL    MBA$L_DR(R4),R2         ; DIAGNOSTIC REGISTER,
             53    14 A3    D0   00E5   525          MOVL    IDB$L_ADP(R3),R3        ;GET ADP ADDRESS
             53    0C A3    3C   00E9   526          MOVZWL  ADP$W_TR(R3),R3         ;SAVE NEXUS NUMBER TO IDENTIFY
                                 00ED   527                                          ; OFFENDING MBA
                                 00ED   528          BUG_CHECK MBACBHUNG,FATAL       ;FATAL ERROR
                                 00F1   529
                                 00F1   533
```

```
                     00F1   535          .SBTTL  MASSBUS ADAPTER INITIALIZATION
                     00F1   536 ;+
                     00F1   537 ; MBA$INITIAL - MASSBUS ADAPTER INITIALIZATION
                     00F1   538 ;
                     00F1   539 ; THIS ROUTINE IS CALLED VIA A JSB INSTRUCTION AT SYSTEM STARTUP AND AFTER
                     00F1   540 ; A POWER RECOVERY RESTART TO ALLOW INITIALIZATION OF MASSBUS ADAPTERS.
                     00F1   541 ;
                     00F1   542 ; INPUTS:
                     00F1   543 ;
                     00F1   544 ;       R4 = CSR ADDRESS OF MASSBUS ADAPTER.
                     00F1   545 ;       R5 = ADDRESS OF ADAPTER IDB.
                     00F1   546 ;
                     00F1   547 ;       ALL INTERRUPTS ARE LOCKED OUT.
                     00F1   548 ;
                     00F1   549 ; OUTPUTS:
                     00F1   550 ;
                     00F1   551 ;       THE MASSBUS ADAPTER IS INITIALIZED AND INTERRUPTS ARE ENABLED.
                     00F1   552 ;-
                     00F1   553
                     00F1   554 MBA$INITIAL::                           ;MASSBUS ADAPTER INITIALIZATION
                     00F1   555
          01   D0    00F1   558          MOVL    #MBA$M_CR_INIT,-
     04   A4         00F3   559           MBA$L_CR(R4)                  ;INITIALIZE MASSBUS ADAPTER
          04   D0    00F5   560          MOVL    #MBA$M_CR_IE,-
     04   A4         00F7   561           MBA$L_CR(R4)                  ;ENABLE INTERRUPTS
                     00F9   564
                05   00F9   565          RSB
```

```
                     00FA   567          .SBTTL  INI$MPMADP - BUILD ADP AND INITIALIZE MULTI-PORT MEMORY
                     00FA   568 ;+
                     00FA   569 ; INI$MPMADP IS CALLED AFTER  MAPPING THE REGISTERS FOR A MULTI-PORT
                     00FA   570 ; MEMORY ADAPTER.  AN ADAPTER CONTROL BLOCK IS ALLOCATED AND FILLED.
                     00FA   571 ; THE HARDWARE ADAPTER IS THEN INITIALIZED BY CALLING MPM$INITIAL.
                     00FA   572 ;
                     00FA   573 ; NOTE: THIS ROUTINE HAS BEEN LOCATED HERE IN SYSLOAXXX.EXE INSTEAD OF
                     00FA   574 ;       INILOA.EXE BECAUSE IT CAN BE CALLED WHILE THE SYSTEM IS RUNNING
                     00FA   575 ;       LONG AFTER INILOA.EXE HAS BEEN DELETED!!!
                     00FA   576 ;
                     00FA   577 ; INPUT:
                     00FA   578 ;       R4 - nexus identification number of this nexus
                     00FA   579 ;
                     00FA   580 ; OUTPUTS:
                     00FA   581 ;       ALL REGISTERS PRESERVED
                     00FA   582 ;-
                     00FA   583
           00000010  00FA   584 NUMMPMVEC = 16                          ; NUMBER OF INTER-PORT INTERRUPT VECTORS
                     00FA   585
                     00FA   586 INI$MPMADP::                           ; INITIALIZE MPM DATA STRUCTURES
                     00FA   587
                  05 00FA   589          RSB                          ; DUMMY ENTRY FOR SYSGEN
                     00FB   590
```

```
                        .SBTTL  MA$INITIAL - INITIALIZE MULTI-PORT MEMORY ADAPTER
                ;++
                ;
                ; MPM$INITIAL - INITIALIZE MULTI-PORT MEMORY ADAPTER
                ;
                ; THIS ROUTINE IS CALLED AT SYSTEM INTIALIZATION AND AFTER A POWER
                ; RECOVERY RESTART TO INITIALIZE THE PORT ADAPTER BY CLEARING ANY
                ; ERRORS AND ENABLING ALL INTERRUPTS.
                ;
                ; INPUTS:
                ;
                ;       R4 = ADDR OF ADAPTER CSR.
                ;
                ;       IPL = 31
                ;
                ; OUPUTS:
                ;
                ;       ANY ERRORS IN PORT ARE CLEARED AND ALL INTERRUPTS ARE ENABLED.
                ;--
                
00FB    681     MA$INITIAL::                            ; INTIALIZE PORT
                
05      00FB    684             RSB
        00FC    685
```

```
                        .SBTTL  INTER-PROCESSOR REQUEST HANDLER
                ;++
                ;
                ; FUNCTIONAL DESCRIPTION:
                ;
                ;       THIS ROUTINE IS CALLED BY A DRIVER OR AN EXEC FUNCTION TO
                ;       EITHER SEND A REQUEST TO OR JUST INTERRUPT ANOTHER PROCESSOR
                ;       THAT IS CONNECTED TO A PORT OF THE MULTIPORT MEMORY.
                ;
                ; INPUTS:
                ;
                ;       R4 = ADAPTER CONTROL BLOCK ADDRESS.
                ;       R5 = IF LSS 0 - ADDRESS OF A FORK BLOCK TO USE IF REQUEST
                ;                       BLOCK IS NOT AVAILABLE.
                ;            IF GEQ 0 - PORT NUMBER OF PROCESSOR TO JUST INTERRUPT.
                ;
                ; OUTPUTS:
                ;
                ;       WHEN THIS ROUTINE IS CALLED WITH A FORK BLOCK ADDRESS, IT WILL
                ;       ATTEMPT TO ALLOCATE A REQUEST BLOCK.  IF THE REQUEST FAILS,
                ;       THE CONTEXT OF THE CALLER WILL BE SAVED IN THE FORK BLOCK, THE
                ;       FORK BLOCK BLOCK WILL BE INSERTED IN THE REQUEST BLOCK WAIT
                ;       QUEUE AND A RETURN TO THE CALLER'S CALLER IS EXECUTED.
                ;
                ;       IF A REQUEST BLOCK IS ALLOCATED SUCCESSFULLY, CONTROL WILL
                ;       RETURN TO THE CALLER VIA A CO-ROUTINE CALL SO THE CALLER CAN
                ;       FILL-IN THE REQUEST BLOCK.
                ;
                ;       THE CALLER WILL THEN PERFORM ANOTHER CO-ROUTINE CALL TO RETURN
                ;       TO THIS ROUTINE SO THE BLOCK CAN BE INSERTED IN THE DESIRED
                ;       PROCESSOR'S INTER-PROCESSOR REQUEST QUEUE.  IF IT IS THE
                ;       FIRST REQUEST IN THE QUEUE AN INTER-PORT INTERRUPT WILL
                ;       ALSO BE REQUESTED TO WAKE-UP THE DISPATCHER ON THE PORT.
                ;
                ;       IF THIS ROUTINE IS CALLED WITH A PORT NUMBER INSTEAD OF A
                ;       FORK BLOCK ADDRESS, IT WILL JUST REQUEST AN INTERRUPT FOR
                ;       THE PROCESSOR ON THE SPECIFIED PORT.  IT IS THEN UP TO THE
                ;       INTERRUPTED PROCESSOR TO DETERMINE WHAT THE INTERRUPT WAS
                ;       FOR.
                ;
                ;       R0 = SUCCESS OR FAILURE OF OPERATION.  THIS SHOULD BE CHECKED
                ;            BY THE CALLER BOTH TIMES THIS ROUTINE RETURNS.
                ;
                ;       R3,R4,R5 ARE PRESERVED.
                ;
                ;--
                ;
MA$REQUEST::                                    ; REQUEST HANDLER
                ;
05              RSB
                ;
```

```
                        .SBTTL  REPORT RESOURCE AVAILABILITY TO INTERESTED PORTS
                ;++
                ;
                ; FUNCTIONAL DESCRIPTION:
                ;
                ;       THIS ROUTINE IS CALLED TO REPORT TO ANY PROCESSORS THAT A RESOURCE
                ;       HAS BEEN MADE AVAILABLE.
                ;
                ; INPUTS:
                ;
                ;       R0 = RESOURCE NUMBER OF RESOURCE MADE AVAILABLE.
                ;       R1 = SHARED MEMORY CONTROL BLOCK (SHB) ADDRESS.
                ;
                ; OUTPUTS:
                ;
                ;       ANY PROCESSORS WAITING FOR THE SPECIFIED RESOURCE ARE INTERRUPTED
                ;       TO NOTIFY THEM THE RESOURCE IS AVAILABLE.
                ;
                ;       R0,R1,R2,R3 ARE NOT PRESERVED.
                ;--

                MA$RAVAIL::

                        RSB

                        .END
```

```
ADP$L_UBASCB                    = 00000044
ADP$W_TR                        = 0000000C
BUG$_MBACBHUNG                    ********   X   02
C780_LIKE                       = 00000000
CI$INITIAL                        0000001F RG    02
CI$INT                            00000000 RG    02
CI$SHUTDOWN                       0000001F RG    02
CPU_TYPE                        = 00000002
DCR_K_CLRPWRDN                  = 00000200
DCR_K_CLRPWRUP                  = 00000100
DCR_K_RESET                     = 00004000
DDT$L_UNSOLINT                  = 00000004
DR$INITIAL                        0000003F RG    02
DR$INT                            00000024 RG    02
DR$SHUTDOWN                       0000003F RG    02
DR_DCR                            00000000
IDB$L_ADP                       = 00000014
IDB$L_CSR                       = 00000000
IDB$L_OWNER                     = 00000004
IDB$L_UCBLST                    = 00000018
INI$MPMADP                        000000FA RG    02
IO$GL_UBA_INTO                    ********   X   02
MA$INITIAL                        000000FB RG    02
MA$RAVAIL                         000000FD RG    02
MA$REQUEST                        000000FC RG    02
MBA$INITIAL                       000000F1 RG    02
MBA$INT                           00000064 RG    02
MBA$L_AS                        = 00000410
MBA$L_CR                        = 00000004
MBA$L_CSR                       = 00000000
MBA$L_DR                        = 00000014
MBA$L_SR                        = 00000008
MBA$M_CR_IE                     = 00000004
MBA$M_CR_INIT                   = 00000001
MBA$M_SR_CBHUNG                 = 00800000
NUMMPAVEC                       = 00000010
PA_CNF                            00000000
PA_CNF_M_PDN                    = 00800000
PA_CNF_M_PUP                    = 00400000
PA_PMC                            00000004
PA_PMC_M_MIN                    = 00000001
PR$_SID_TYP730                  = 00000003
PR$_SID_TYP750                  = 00000002
PR$_SID_TYP780                  = 00000001
PR$_SID_TYP790                  = 00000004
PR$_SID_TYPUV1                  = 00000007
SIZ...                          = 00000006
UAS$M_IP_CR1_PIE                = 00001000
UBA$INITIAL                       00000045 RG    02
UBA$INTO                          00000058 RG    02
UBA_UNEXINT                       00000060 RG    02
UCB$B_SLAVE                     = 00000090
UCB$L_DDT                       = 00000088
UCB$L_FPC                       = 0000000C
UCB$L_FR3                       = 00000010
UCB$V_INT                       = 00000001
UCB$W_STS                       = 00000064
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	00000008 (8.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SYSLOA	000000FE (254.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.05	00:00:01.19
Command processing	106	00:00:00.45	00:00:03.30
Pass 1	501	00:00:12.72	00:00:48.72
Symbol table sort	6	00:00:02.03	00:00:09.13
Pass 2	115	00:00:02.78	00:00:13.78
Symbol table output	7	00:00:00.08	00:00:00.69
Psect synopsis output	2	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	769	00:00:18.13	00:01:16.83

The working set limit was 1800 pages.
120628 bytes (236 pages) of virtual memory were used to buffer the intermediate code.
There were 110 pages of symbol table space allocated to hold 1977 non-local and 6 local symbols.
1179 source lines were read in Pass 1, producing 13 object records in Pass 2.
32 pages of virtual memory were used to define 31 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	19
_255DUA28:[SYSLIB]STARLET.MLB;2	7
TOTALS (all libraries)	26

2031 GETS were required to define 26 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:ADPSUB750/OBJ=OBJ$:ADPSUB750 MSRC$:CPUSW750/UPDATE=(ENH$:CPUSW750)+MSRC$:ADPSUB/UPDATE=(ENH$:ADPSUB)+EXECML$/LIB
```

ADPSUB780
LIS

ACKMSG
LIS

MCF790
SDL

MCDEF
MDL

ADPERR750
LIS

ADPSUB730
LIS

CSPDEF
SDL

CLUMBX
SDL

ADPERR780
LIS

ADPSUB750
LIS

CLUSTRMAC
MAR

CLUSTER
SDL