L 5

```
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS  LLL                 OOOOOOOOO        AAAAAAAAA
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS  LLL                 OOOOOOOOO        AAAAAAAAA
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS  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
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS     LLLLLLLLLLLLLLL     OOOOOOOOO     AAA         AAA
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS     LLLLLLLLLLLLLLL     OOOOOOOOO     AAA         AAA
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS     LLLLLLLLLLLLLLL     OOOOOOOOO     AAA         AAA
```

```
_$2

Sym
---
SS1
SS1

SS1
SS1
SS1
SS1
SS1
SS1
SYS
SYS
SYS
TRY
UNL
WRI
```

```
**FILE**ID**ADPSUB790                                         I  2




  AAAAAA    DDDDDDDD    PPPPPPPP      SSSSSSSS  UU      UU  BBBBBBBB    77777777      999999      000000
  AAAAAA    DDDDDDDD    PPPPPPPP      SSSSSSSS  UU      UU  BBBBBBBB    77777777      999999      000000
AA      AA  DD      DD  PP      PP  SS          UU      UU  BB      BB          77  99      99  00      00
AA      AA  DD      DD  PP      PP  SS          UU      UU  BB      BB          77  99      99  00      00
AA      AA  DD      DD  PP      PP  SS          UU      UU  BB      BB          77  99      99  00    0000
AA      AA  DD      DD  PP      PP  SS          UU      UU  BB      BB          77  99      99  00    0000
AA      AA  DD      DD  PPPPPPPP      SSSSSS    UU      UU  BBBBBBBB          77      99999999  00  00  00
AA      AA  DD      DD  PPPPPPPP      SSSSSS    UU      UU  BBBBBBBB          77      99999999  00  00  00
AAAAAAAAAA  DD      DD  PP                  SS  UU      UU  BB      BB      77              99  0000    00
AAAAAAAAAA  DD      DD  PP                  SS  UU      UU  BB      BB      77              99  0000    00
AA      AA  DD      DD  PP                  SS  UU      UU  BB      BB    77                99  00      00      ....
AA      AA  DD      DD  PP                  SS  UU      UU  BB      BB    77                99  00      00      ....
AA      AA  DDDDDDDD    PP          SSSSSSSS    UUUUUUUUUU  BBBBBBBB    77            999999      000000        ....
AA      AA  DDDDDDDD    PP          SSSSSSSS    UUUUUUUUUU  BBBBBBBB    77            999999      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
```

ADP
V04

```
0000            .NOSHOW CONDITIONALS
0000
0000
0000
0000            .TITLE  ADPSUB790 - ADAPTER SUBROUTINES FOR VAX 11/790
0000
0000
0000            .IDENT  'V04-000'
0000
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000    ; Facility: System bootstrapping and initialization
0000    ;
0000    ; Abstract: This module contains initialization routines that are loaded
0000    ;            during system initialization (rather than linked into the system).
0000    ;
0000    ; Environment: Mode = KERNEL, Executing on INTERRUPT stack, IPL=31
0000    ;
0000    ; Author:  Kerbey T. Altmann            Creation date: 30-Oct-1982
0000    ;
0000    ; Modification history:
0000    ;
0000    ;       V03-007 TCM0002         Trudy C. Matthews       04-Jun-1984
0000    ;               Include more 780-specific code for the 11/790 version of
0000    ;               this routine.
0000    ;
0000    ;       V03-006 KPL0001         Peter Lieberwirth       12-Apr-1984
0000    ;               Init ADP$L_SHB properly again; V03-004 ASSUMEd this field
0000    ;               was at a certain constant offset, and a change to the ADP
0000    ;               moved it.  Note - this is a 780 change only.
0000    ;
0000    ;       V03-005 KDM0081         Kathleen D. Morse       13-Sep-1983
0000    ;               Create version for Micro-VAX I.
0000    ;
0000    ;       V03-004 ROW0196         Ralph O. Weber          27-JUL-1983
0000    ;               Correct INI$MPMADP so the ADP$L_SHB is correctly initialized
```

```
0000    ;                to zero.
0000    ;
0000    ;        V03-003 MSH0001         Maryann Hinden          06-Dec-1982
0000    ;                Add initialization for DW750.
0000    ;
0000    ;        V03-002 ROW0142         Ralph O. Weber          23-NOV-1982
0000    ;                Correct JMP in multiport memory interrupt dispatching code
0000    ;                prototype, MPMINTD, to a JSB.  MA$INT expects to receive
0000    ;                control via a JSB.
0000    ;
0000    ;        V03-001 TCM0001         Trudy C. Matthews       8-Nov-1982
0000    ;                initialize field ADP$L_AVECTOR in INI$MPMADP.
0000    ;
0000    ;--
```

```
                0000
       00000001 0000          C780_LIKE = 1
                0000
                0000
                0000
                0000    ;
                0000    ; MACRO LIBRARY CALLS
                0000    ;
                0000          $ADPDEF                          ; Define ADP offsets.
                0000          $CRBDEF                          ; Define CRB offsets.
                0000          $DCDEF                           ; Define AT codes.
                0000          $DDBDEF                          ; Define DDB offsets.
                0000          $DDTDEF                          ; Define DDT offsets.
                0000          $DYNDEF                          ; Define data structure type codes.
                0000          $IDBDEF                          ; Define interrupt dispatcher offsets.
                0000          $MBADEF                          ; Define MASSBUS registers.
                0000          $MCHKDEF                         ; Define machine check masks.
                0000          $MPMDEF                          ; Define multi-port memory.
                0000          $NDTDEF                          ; Define nexus device types.
                0000          $PRDEF                           ; Define IPR numbers.
                0000          $PTEDEF                          ; Define Page Table Entry bits.
                0000          $RPBDEF                          ; Define Restart Parameter Block fields.
                0000          $SSDEF                           ; Define system service codes.
                0000          $UBADEF                          ; Define UBA register offsets.
                0000          $UBIDEF                          ; Define UNIBUS interconnect
                0000                                           ;  register offsets.
                0000          $UCBDEF                          ; Define unit control block.
                0000          $VADEF                           ; Define virtual address fields.
                0000          $VECDEF                          ; Define vec offsets.
                0000
                0000          $CEBDEF                          ; COMMON EVENT BLOCK
                0000          $FKBDEF                          ; FORK BLOCK
                0000          $IPLDEF                          ; INTERRUPT PRIORITY LEVELS
                0000          $PRIDEF                          ; PRIORITY INCREMENT DEFINITIONS
                0000          $PRQDEF                          ; INTER-PROCESSOR REQUEST
                0000          $RSNDEF                          ; RESOURCE NUMBER DEFINITIONS
                0000          $SHBDEF                          ; SHARED MEMORY CONTROL BLOCK
                0000          $SHDDEF                          ; SHARED MEMORY DATAPAGE
                0000
                0000
       00000000               .PSECT  SYSLOA,LONG
```

```
0000            .SBTTL  CI$INT - CI INTERRUPT HANDLER
0000    ;+
0000    ; CI$INT - CI INTERRUPT HANDLER
0000    ;
0000    ;       THIS MODULE IS A DUMMY CI32 INTERRUPT HANDLER WHICH IS USED
0000    ;       UNTIL THE REAL CI DRIVER (PADRIVER) IS LOADED. IT ALSO CONTAINS
0000    ;       A DUMMY CI32 CONTROLLER INITIALIZATION  ENTRY POINT.
0000    ;
0000    ; INPUTS:
0000    ;
0000    ;       THE STACK ON ENTRY IS AS FOLLOWS:
0000    ;
0000    ;               0(SP)                   ADDRESS OF IDB ADDRESS
0000    ;       4(SP) - 16(SP)                  SAVED R2 - R5
0000    ;               20(SP)                  INTERRUPT PC
0000    ;               24(SP)                  INTERRUPT PSL
0000    ;
0000    ; OUTPUTS:
0000    ;
0000    ;       NONE
0000    ;
0000    ; SIDE EFFECTS:
0000    ;
0000    ;       INTERRUPTS ARE DISABLED ON THE CI32
0000    ;-
0000
0000
0000    ;
0000    ; $PAREGDEF -- Define offsets to CI registers and fields in the registers.
0000    ;
0000
0000            $DEFINI PAREG
0000
0000            $DEF    PA_CNF  .BLKL   1                       ; Configuration register
0004
0004            _VIELD  PA_CNF,0,<-                             ; Define config register fields:
0004            <ADPTYP,8,M>,-                                  ;  Adapter type code
0004            <PFD,,M>,-                                      ;  Powerfail disable
0004            <TDEAD,,M>,-                                    ;  Transmit dead
0004            <TFAIL,,M>,-                                    ;  Transmit fail
0004            <,5>,-                                          ;  5 unused bits
0004            <CRD,,M>,-                                      ;  CRD on port init'd read
0004            <RDS,,M>,-                                      ;  RDS on port init'd read
0004            <CXTER,,M>,-                                    ;  SBI error confirm
0004            <RDTO,,M>,-                                     ;  Port init'd read timeout on SBI
0004            <CSTMO,,M>,-                                    ;  Port init'd command xmit timeout
0004            <,1>,-                                          ;  1 unused bit
0004            <PUP,,M>,-                                      ;  Adapter power up
0004            <PDN,,M>,-                                      ;  Adaptor power down
0004            >                                               ;
0004
0004            $DEF    PA_PMC  .BLKL   1                       ; Port maint control/status register
0008
0008            _VIELD  PA_PMC,0,<-                             ; Define register fields:
0008            <MIN,,M>,-                                      ;  Maint initialized
0008            <MTD,,M>,-                                      ;  Maint timer disable
0008            <MIE,,M>,-                                      ;  Maint interrupt enable
```

```
ADP
Sym

ADP
C78
CI$
CI$
CI$
CPU
DCR
DCR
DCR
DDT
DR$
DR$
DR$
DR_
EXE
IDB
IDB
IDB
IDB
INI
IO$
MA$
MA$
MA$
MBA
MBA
MBA
MBA
MBA
MBA
MBA
MBA
MBA
NUM
PA_
PA_
PA_
PA_
PA_
PR$
PR$
PR$
PR$
PR$
SIZ
UBA
UBA
UB4
UBA
UBA
UBA
UBA
UBA
UBA
UBA
UBA
UBA_
```

```
                                0008             <MIF,,M>,-                                 ;  Maint intterupt flag
                                0008             >                                          ;
                                0008
                                0008             $DEFEND PAREG
                                0000
                                0000     CI$INT::
           53    9E    D0       0000             MOVL    @(SP)+,R3                          ; GET ADDRESS OF IDB
           54    63    D0       0003             MOVL    IDB$L_CSR(R3),R4                   ; GET ADDRESS OF FIRST CSR
  64   00400000  8F    D0       0006             MOVL    #PA_CNF_M_PUP,PA_CNF(R4)           ; CLEAR POWER UP
  64   008C0000  8F    D0       000D             MOVL    #PA_CNF_M_PDN,PA_CNF(R4)           ; CLEAR POWER DOWN
        04 A4    01    D0       0014             MOVL    #PA_PMC_M_MIN,PA_PMC(R4)           ; SET MAINTENCE INITIALIZE
           52    8E    7D       0018             MOVQ    (SP)+,R2                           ; RESTORE REGISTERS
           54    8E    7D       001B             MOVQ    (SP)+,R4
                       02       001E             REI
                                001F
                                001F
                                001F     CI$INITIAL::                                       ; CONTROLLER INITIALIZATION
                                001F     CI$SHUTDOWN::                                      ; CONTROLLER SHUTDOWN
                                001F
                                001F
        04 A4    01    D0       001F             MOVL    #PA_PMC_M_MIN,PA_PMC(R4)           ; SET MAINTENCE INITIALIZE
                                0023
                       05       0023             RSB
```

```
ADP
Pse

PSE
---
.
$AB
SYS

Pha
---
Ini
Com
Pas
Sym
Pas
Sym
Pse
Cro
Ass

The
131
The
117
38

Mac
---
_$2
_$2
_$2
TOT

221

The

MAC
```

```
          0024          .SBTTL  DR$INT - DR INTERRUPT HANDLER
          0024  ;+
          0024  ; DR$INT - DR INTERRUPT HANDLER
          0024  ;
          0024  ;       THIS MODULE IS A DUMMY DR32 INTERRUPT HANDLER WHICH IS USED
          0024  ;       UNTIL THE REAL DR DRIVER (XFDRIVER) IS LOADED. IT ALSO CONTAINS
          0024  ;       A DUMMY DR32 CONTROLLER INITIALIZATION  ENTRY POINT.
          0024  ;
          0024  ; INPUTS:
          0024  ;
          0024  ;       THE STACK ON ENTRY IS AS FOLLOWS:
          0024  ;
          0024  ;               0(SP)           ADDRESS OF IDB ADDRESS
          0024  ;       4(SP) - 16(SP)          SAVED R2 - R5
          0024  ;              20(SP)           INTERRUPT PC
          0024  ;              24(SP)           INTERRUPT PSL
          0024  ;
          0024  ; OUTPUTS:
          0024  ;
          0024  ;       NONE
          0024  ;
          0024  ; SIDE EFFECTS:
          0024  ;
          0024  ;       INTERRUPTS ARE DISABLED ON THE DR32
          0024  ;-
          0024
          0024
          0024  ;
          0024  ; DR32 DCR REGISTER DEFINITIONS
          0024  ;-
          0024
          0024          $DEFINI DR
          0000  $DEF    DR_DCR          .BLKL   1       ; DR32 CONTROL REGISTER
          0004          _VIELD  DR_DCR,0,<-
          0004                  <ADPTYP,8>,-            ; ADAPTER TYPE
          0004                  <ID2ERR,,M>,-           ; ID2 ERROR
          0004                  <ID2TOS,2>,-            ; ID2 TIME-OUT STATUS
          0004                  <,1>,-                  ; RESERVED
          0004                  <ID1ERR,,M>,-           ; ID1 ERROR
          0004                  <ID1TOS,2>,-            ; ID1 TIME-OUT STATUS
          0004                  <RDS,,M>,-              ; READ DATA SUBSTITUTE
          0004                  <CRD,,M>,-              ; CORRECTED READ DATA
          0004                  <DCRHLT,,M>,-           ; DCR HALT
          0004                  <DCRABT,,M>,-           ; DCR ABORT INTERRUPT
          0004                  <PKTINT,,M>,-           ; PACKET INTERRUPT
          0004                  <INTENB,,M>,-           ; INTERRUPT ENABLE
          0004                  <,1>,-                  ; RESERVED
          0004                  <PWR_UP,,M>,-           ; ADAPTER POWER UP
          0004                  <PWR_DN,,M>,-           ; ADAPTER POWER DOWN
          0004                  <EXTABT,,M>,-           ; EXTERNAL ABORT
          0004                  <,1>,-                  ; RESERVED
          0004                  <IMPDEP,6>,-            ; IMPLEMENTATION DEPENDENT BITS
          0004          >
          0004
          0004  ; DCR CONTROL FIELD A CODES (USED WHEN WRITING TO DCR)
          0004
00000100  0004          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             .SBTTL  UBA$INITIAL - CPU-DEPENDENT UNIBUS ADAPTER INITIALIZATION
                           0045      ;+
                           0045      ; UBA$INITIAL - UNIBUS ADAPTER INITIALIZATION
                           0045      ;
                           0045      ; THIS ROUTINE IS CALLED VIA A JSB INSTRUCTION AT SYSTEM STARTUP AND AFTER
                           0045      ; A POWER RECOVERY RESTART TO ALLOW INITIALIZATION OF UNIBUS ADAPTERS.
                           0045      ;        (POWERFAIL AND INITADP)
                           0045      ;
                           0045      ;
                           0045      ; INPUTS:
                           0045      ;
                           0045      ;        R2 = ADDRESS OF ADAPTER CONTROL BLOCK  (11/780 AND 11/750)
                           0045      ;        R4 = ADDRESS OF UNIBUS ADAPTER CONFIGURATION STATUS REGISTER (11/780)
                           0045      ;
                           0045      ;        ALL INTERRUPTS ARE LOCKED OUT.
                           0045      ;
                           0045      ; OUTPUTS:
                           0045      ;
                           0045      ;        THE UNIBUS ADAPTER IS INITIALIZED AND INTERRUPTS ARE ENABLED.
                           0045      ;-
                           0045
                           0045     UBA$INITIAL::                                        ;UNIBUS ADAPTER INITIALIZATION
                           0045
                           0045                                                          ;
          64    00    D2   0045             MCOML   #0,UBA$L_CSR(R4)                     ;CLEAR ALL ADAPTER CONFIGURATION ERRORS
       08 A4    00    D2   0048             MCOML   #0,UBA$L_SR(R4)                      ;CLEAR ALL ADAPTER STATUS BITS
     50    0256   C2  3C   004C             MOVZWL  ADP$W_UMR_DIS(R2),R0                 ;PICK UP THE NUMBER OF UMR'S TO DISABLE
     50      50   16  78   0051             ASHL    #UBA$V_CR_MRDSB-4,R0,R0              ;DIVIDE BY 16 THEN SHIFT INTO POSITION
                      C9   0055             BISL3   #UBA$M_CR_SUEFIE!-                   ;ENABLE INTERRUPTS
                           0056                      UBA$M_CR_BRIE!-
                           0056                      UBA$M_CR_CNFIE!-
                           0056                      UBA$M_CR_USEFIE!-
                           0056                      UBA$M_CR_IFSIE,-
04 A4    50   0000007C 8F  0056                     R0,UBA$L_CR(R4)
                           005E
                           005E
                           005E
                           005E     10$:                                                 ;NO SPECIAL INIT FOR 11/730 OR uVAX I
                      05   005E             RSB                                          ;
                           005F      ;
                           005F      ; IGNORE UNEXPECTED UNIBUS INTERRUPTS
                           005F      ;
                           005F
                           005F             .ALIGN  LONG
                           0060
                           0060     UBA$INT0::                                           ; PASSIVE RELEASES THROUGH VECTOR 0
                           0060
        00000000'9F   D6   0060             INCL    @#IO$GL_UBA_INT0                     ; COUNT THEM
                 00   11   0066             BRB     UBA_UNEXINT                          ; JOIN COMMON CODE, VECTORS ARE ALLIGNED
                           0068
                           0068             .ALIGN  LONG
                           0068
                           0068      ;
                           0068      ; NOTE: UBA$UNEXINT is the label in the EXEC that is a JMP @#UBA_UNEXINT.
                           0068      ;        This seeming duplicity is necesary since there is code that must
                           0068      ;        refer to the EXEC address from within the SYSLOA image.
                           0068      ;
                           0068     UBA_UNEXINT::                                        ; UNEXPECTED INTERRUPT CODE
                 0068   405
                 0068   407                                               ; FOR 780-LIKE PROCESSORS, RESTORE
                 0068   408                                               ; SAVED REGISTERS
      3F    BA   0068   409            POPR     #^M<R0,R1,R2,R3,R4,R5>    ;
                 006A   412                                               ; FOR 11/750, NO REGISTERS SAVED
            02   006A   414            REI                                ; IGNORE INTERRUPT
```

```
                           006B              .SBTTL  MASSBUS ADAPTER INTERRUPT DISPATCHER
                           006B      ;+
                           006B      ; MBA$INT - MASSBUS ADAPTER INTERRUPT DISPATCHER
                           006B      ;
                           006B      ; THIS ROUTINE IS ENTERED VIA A JSB INSTRUCTION WHEN AN INTERRUPT OCCURS
                           006B      ; ON A MASSBUS ADAPTER. THE STATE OF THE STACK ON ENTRY IS:
                           006B      ;
                           006B      ;       00(SP) = ADDRESS OF IDB ADDRESS.
                           006B      ;       04(SP) = SAVED R2.
                           006B      ;       08(SP) = SAVED R3.
                           006B      ;       12(SP) = SAVED R4.
                           006B      ;       16(SP) = SAVED R5.
                           006B      ;       20(SP) = INTERRUPT PC.
                           006B      ;       24(SP) = INTERRUPT PSL.
                           006B      ;
                           006B      ; INTERRUPT DISPATCHING OCCURS AS FOLLOWS:
                           006B      ;
                           006B      ;       IF THE INTERRUPTING ADAPTER IS CURRENTLY OWNED AND THE OWNER UNIT
                           006B      ;       IS EXPECTING AN INTERRUPT, THEN THAT UNIT IS DISPATCHED FIRST. ALL
                           006B      ;       OTHER UNITS ARE DISPATCHED BY READING THE ATTENTION SUMMARY REG-
                           006B      ;       ISTER AND SCANNING FOR UNITS THAT HAVE ATTENTION SET. AS EACH UNIT
                           006B      ;       IS FOUND, ITS ATTENTION SUMMARY BIT IS CLEARED AND THEN A TEST IS
                           006B      ;       MADE TO DETERMINE IF AN INTERRUPT IS EXPECTED ON THE UNIT. IF YES,
                           006B      ;       THEN THE DRIVER IS CALLED AT ITS INTERRUPT RETURN ADDRESS. ELSE
                           006B      ;       THE DRIVER IS CALLED AT ITS UNSOLICITED INTERRUPT ADDRESS. AS EACH
                           006B      ;       CALL TO THE DRIVER RETURNS, THE ATTENTION SUMMARY REGISTER IS RE-
                           006B      ;       READ AND AN ATTEMPT IS MADE TO FIND ANOTHER UNIT TO DISPATCH. WHEN
                           006B      ;       NO UNITS REQUESTING ATTENTION REMAIN, THE INTERRUPT IS DISMISSED.
                           006B      ;-
                           006B
                           006B              .ALIGN  LONG
                           006C
                           006C      MBA$INT::                                         ;MASSBUS ADAPTER INTERRUPT DISPATCHER
        53    00 BE    D0  006C              MOVL    @(SP),R3                          ;GET ADDRESS OF IDB
           54    63    D0  0070              MOVL    IDB$L_CSR(R3),R4                  ;GET ADDRESS OF CONFIGURATION STATUS REGISTE
                           0073
                           0073
  00800000 8F          D3  0073              BITL    #MBA$M_CSR_PD,-
           64              0079                       MBA$L_CSR(R4)                    ;CHECK FOR MBA POWER DOWN
           61          12  007A              BNEQ    45$                               ;BRANCH IF POWERFAIL
                           007C
                           007C
        55    04 A3    D0  007C              MOVL    IDB$L_OWNER(R3),R5                ;GET OWNER UNIT UCB ADDRESS
                 0A    13  0080              BEQL    10$                               ;IF EQL NO OWNER
      52    0090 C5    9A  0082              MOVZBL  UCB$B_SLAVE(R5),R2                ;GET OWNER SLAVE CONTROLLER NUMBER
     21 64 A5    01    E0  0087              BBS     #UCB$V_INT,UCB$W_STS(R5),20$ ;IF SET, INTERRUPT EXPECTED
        53    00 BE    D0  008C      10$:    MOVL    @(SP),R3                          ;RETRIEVE ADDRESS OF IDB
           54    63    D0  0090              MOVL    IDB$L_CSR(R3),R4                  ;RETRIEVE MBA CONFIGURATION REGISTER ADDRESS
        08 A4    00    D2  0093              MCOML   #0,MBA$L_SR(R4)                   ;CLEAR ALL MBA STATUS BITS
      52    0410 C4    D0  0097              MOVL    MBA$L_AS(R4),R2                   ;READ ATTENTION SUMMARY REGISTER
  52  52    08   00    EA  009C              FFS     #0,#8,R2,R2                       ;FIND FIRST UNIT REQUESTING ATTENTION
                 0A    12  00A1              BNEQ    20$                               ;IF NEQ UNIT FOUND
           5E    04    C0  00A3              ADDL    #4,SP                             ;REMOVE IDB ADDRESS FROM STACK
           52    8E    7D  00A6              MOVQ    (SP)+,R2                          ;RESTORE REGISTERS
           54    8E    7D  00A9              MOVQ    (SP)+,R4                          ;
                       02  00AC              REI                                       ;
                           00AD
```

```
          55     18 A342   D0  00AD    483 20$:      MOVL    IDB$L_UCBLST(R3)[R2],R5  ;GET ADDRESS OF UCB OR INTERRUPT DISPATCHER
                    22 55  E8  00B2    484           BLBS    R5,40$                    ;IF LBS INTERRUPT DISPATCHER FOR MULTI-
                               00B5    485                                             ; DEVICE CONTROLLER
      0410 C4    01     52 78  00B5    486           ASHL    R2,#1,MBA$L_AS(R4)        ;CLEAR ATTENTION SUMMARY BIT
                        55 D5  00BB    487           TSTL    R5                        ;SEE IF UCB DEFINED
                        CD 13  00BD    488           BEQL    10$                       ;IF EQL NONE DEFINED
         09 64 A5       01 E5  00BF    489           BBCC    #UCB$V_INT,UCB$W_STS(R5),30$ ;IF CLR, INTERRUPT NOT EXPECTED
            53     10 A5   7D  00C4    490           MOVQ    UCB$L_FR3(R5),R3          ;RESTORE DRIVER CONTEXT
                   0C B5   16  00C8    491           JSB     @UCB$L_FPC(R5)            ;CALL DRIVER AT INTERRUPT RETURN ADDRESS
                      BF   11  00CB    492           BRB     10$                       ;
                               00CD    493
          53     0088 C5   D0  00CD    494 30$:      MOVL    UCB$L_DDT(R5),R3          ;GET ADDRESS OF DDT
                    04 B3  16  00D2    495           JSB     @DDT$L_UNSOLINT(R3)       ;CALL UNSOLICITED INTERRUPT ROUTINE
                       B5  11  00D5    496           BRB     10$                       ;
                               00D7    497
                       7E  DC  00D7    498 40$:      MOVPSL  -(SP)                     ;READ CURRENT PSL
                       75  16  00D9    499           JSB     -(R5)                     ;CALL SLAVE CONTROLLER INTERRUPT DISPATCHER
                       AF  11  00DB    500           BRB     10$                       ;
                               00DD    501
                               00DD    503
                               00DD    504 ;
                               00DD    505 ; IN CASE OF ADAPTER POWER DOWN BIT ASSERTED, RETRIEVE ADP ADDRESS AND JUMP
                               00DD    506 ; TO ADAPTER ERROR ROUTINE IN SYSLOA780.
                               00DD    507 ;
                               00DD    508
          54     14 A3     D0  00DD    509 45$:      MOVL    IDB$L_ADP(R3),R4          ;GET ADP ADDRESS
                   FF1C'   31  00E1    510           BRW     EXE$RH780_INT             ;JUMP TO ERROR ROUTINE
                               00E4    511
                               00E4    533
```

```
                  00E4          .SBTTL  MASSBUS ADAPTER INITIALIZATION
                  00E4  ;+
                  00E4  ; MBA$INITIAL - MASSBUS ADAPTER INITIALIZATION
                  00E4  ;
                  00E4  ; THIS ROUTINE IS CALLED VIA A JSB INSTRUCTION AT SYSTEM STARTUP AND AFTER
                  00E4  ; A POWER RECOVERY RESTART TO ALLOW INITIALIZATION OF MASSBUS ADAPTERS.
                  00E4  ;
                  00E4  ; INPUTS:
                  00E4  ;
                  00E4  ;       R4 = CSR ADDRESS OF MASSBUS ADAPTER.
                  00E4  ;       R5 = ADDRESS OF ADAPTER IDB.
                  00E4  ;
                  00E4  ;       ALL INTERRUPTS ARE LOCKED OUT.
                  00E4  ;
                  00E4  ; OUTPUTS:
                  00E4  ;
                  00E4  ;       THE MASSBUS ADAPTER IS INITIALIZED AND INTERRUPTS ARE ENABLED.
                  00E4  ;-
                  00E4
                  00E4  MBA$INITIAL::                               ;MASSBUS ADAPTER INITIALIZATION
                  00E4
         01   D0  00E4          MOVL    #MBA$M_CR_INIT,-
      04 A4       00E6                    MBA$L_CR(R4)              ;INITIALIZE MASSBUS ADAPTER
         04   D0  00E8          MOVL    #MBA$M_CR_IE,-
      04 A4       00EA                    MBA$L_CR(R4)              ;ENABLE INTERRUPTS
                  00EC
              05  00EC          RSB
```

```
                    00ED               .SBTTL  INI$MPMADP - BUILD ADP AND INITIALIZE MULTI-PORT MEMORY
                    00ED          ;+
                    00ED          ; INI$MPMADP IS CALLED AFTER  MAPPING THE REGISTERS FOR A MULTI-PORT
                    00ED          ; MEMORY ADAPTER.  AN ADAPTER CONTROL BLOCK IS ALLOCATED AND FILLED.
                    00ED          ; THE HARDWARE ADAPTER IS THEN INITIALIZED BY CALLING MPM$INITIAL.
                    00ED          ;
                    00ED          ; NOTE: THIS ROUTINE HAS BEEN LOCATED HERE IN SYSLOAXXX.EXE INSTEAD OF
                    00ED          ;       INILOA.EXE BECAUSE IT CAN BE CALLED WHILE THE SYSTEM IS RUNNING
                    00ED          ;       LONG AFTER INILOA.EXE HAS BEEN DELETED!!!
                    00ED          ;
                    00ED          ; INPUT:
                    00ED          ;       R4 - nexus identification number of this nexus
                    00ED          ;
                    00ED          ; OUTPUTS:
                    00ED          ;       ALL REGISTERS PRESERVED
                    00ED          ;-
                    00ED
          00000010  00ED          NUMMPMVEC = 16                          ; NUMBER OF INTER-PORT INTERRUPT VECTORS
                    00ED
                    00ED          INI$MPMADP::                            ; INITIALIZE MPM DATA STRUCTURES
                    00ED
                05  00ED                  RSB                             ; DUMMY ENTRY FOR SYSGEN
                    00EE
```

```
             00EE   661           .SBTTL  MA$INITIAL - INITIALIZE MULTI-PORT MEMORY ADAPTER
             00EE   662  ;++
             00EE   663  ;
             00EE   664  ; MPM$INITIAL - INITIALIZE MULTI-PORT MEMORY ADAPTER
             00EE   665  ;
             00EE   666  ; THIS ROUTINE IS CALLED AT SYSTEM INTIALIZATION AND AFTER A POWER
             00EE   667  ; RECOVERY RESTART TO INITIALIZE THE PORT ADAPTER BY CLEARING ANY
             00EE   668  ; ERRORS AND ENABLING ALL INTERRUPTS.
             00EE   669  ;
             00EE   670  ; INPUTS:
             00EE   671  ;
             00EE   672  ;         R4 = ADDR OF ADAPTER CSR.
             00EE   673  ;
             00EE   674  ;         IPL = 31
             00EE   675  ;
             00EE   676  ; OUPUTS:
             00EE   677  ;
             00EE   678  ;         ANY ERRORS IN PORT ARE CLEARED AND ALL INTERRUPTS ARE ENABLED.
             00EE   679  ;--
             00EE   680
             00EE   681  MA$INITIAL::                               ; INTIALIZE PORT
             00EE   682
         05  00EE   684           RSB
             00EF   685
```

```
        00EF            .SBTTL  INTER-PROCESSOR REQUEST HANDLER
        00EF    ;++
        00EF    ;
        00EF    ; FUNCTIONAL DESCRIPTION:
        00EF    ;
        00EF    ;       THIS ROUTINE IS CALLED BY A DRIVER OR AN EXEC FUNCTION TO
        00EF    ;       EITHER SEND A REQUEST TO OR JUST INTERRUPT ANOTHER PROCESSOR
        00EF    ;       THAT IS CONNECTED TO A PORT OF THE MULTIPORT MEMORY.
        00EF    ;
        00EF    ; INPUTS:
        00EF    ;
        00EF    ;       R4 = ADAPTER CONTROL BLOCK ADDRESS.
        00EF    ;       R5 = IF LSS 0 - ADDRESS OF A FORK BLOCK TO USE IF REQUEST
        00EF    ;                       BLOCK IS NOT AVAILABLE.
        00EF    ;            IF GEQ 0 - PORT NUMBER OF PROCESSOR TO JUST INTERRUPT.
        00EF    ;
        00EF    ; OUTPUTS:
        00EF    ;
        00EF    ;       WHEN THIS ROUTINE IS CALLED WITH A FORK BLOCK ADDRESS, IT WILL
        00EF    ;       ATTEMPT TO ALLOCATE A REQUEST BLOCK.  IF THE REQUEST FAILS,
        00EF    ;       THE CONTEXT OF THE CALLER WILL BE SAVED IN THE FORK BLOCK, THE
        00EF    ;       FORK BLOCK BLOCK WILL BE INSERTED IN THE REQUEST BLOCK WAIT
        00EF    ;       QUEUE AND A RETURN TO THE CALLER'S CALLER IS EXECUTED.
        00EF    ;
        00EF    ;       IF A REQUEST BLOCK IS ALLOCATED SUCCESSFULLY, CONTROL WILL
        00EF    ;       RETURN TO THE CALLER VIA A CO-ROUTINE CALL SO THE CALLER CAN
        00EF    ;       FILL-IN THE REQUEST BLOCK.
        00EF    ;
        00EF    ;       THE CALLER WILL THEN PERFORM ANOTHER CO-ROUTINE CALL TO RETURN
        00EF    ;       TO THIS ROUTINE SO THE BLOCK CAN BE INSERTED IN THE DESIRED
        00EF    ;       PROCESSOR'S INTER-PROCESSOR REQUEST QUEUE.  IF IT IS THE
        00EF    ;       FIRST REQUEST IN THE QUEUE AN INTER-PORT INTERRUPT WILL
        00EF    ;       ALSO BE REQUESTED TO WAKE-UP THE DISPATCHER ON THE PORT.
        00EF    ;
        00EF    ;
        00EF    ;       IF THIS ROUTINE IS CALLED WITH A PORT NUMBER INSTEAD OF A
        00EF    ;       FORK BLOCK ADDRESS, IT WILL JUST REQUEST AN INTERRUPT FOR
        00EF    ;       THE PROCESSOR ON THE SPECIFIED PORT.  IT IS THEN UP TO THE
        00EF    ;       INTERRUPTED PROCESSOR TO DETERMINE WHAT THE INTERRUPT WAS
        00EF    ;       FOR.
        00EF    ;
        00EF    ;       R0 = SUCCESS OR FAILURE OF OPERATION.  THIS SHOULD BE CHECKED
        00EF    ;            BY THE CALLER BOTH TIMES THIS ROUTINE RETURNS.
        00EF    ;
        00EF    ;       R3,R4,R5 ARE PRESERVED.
        00EF    ;
        00EF    ;--
        00EF
        00EF    MA$REQUEST::                                    ; REQUEST HANDLER
        00EF
    05  00EF            RSB
        00F0
        00F0   847           .SBTTL  REPORT RESOURCE AVAILABILITY TO INTERESTED PORTS
        00F0   848 ;++
        00F0   849 ;
        00F0   850 ; FUNCTIONAL DESCRIPTION:
        00F0   851 ;
        00F0   852 ;         THIS ROUTINE IS CALLED TO REPORT TO ANY PROCESSORS THAT A RESOURCE
        00F0   853 ;         HAS BEEN MADE AVAILABLE.
        00F0   854 ;
        00F0   855 ; INPUTS:
        00F0   856 ;
        00F0   857 ;         R0 = RESOURCE NUMBER OF RESOURCE MADE AVAILABLE.
        00F0   858 ;         R1 = SHARED MEMORY CONTROL BLOCK (SHB) ADDRESS.
        00F0   859 ;
        00F0   860 ; OUTPUTS:
        00F0   861 ;
        00F0   862 ;         ANY PROCESSORS WAITING FOR THE SPECIFIED RESOURCE ARE INTERRUPTED
        00F0   863 ;         TO NOTIFY THEM THE RESOURCE IS AVAILABLE.
        00F0   864 ;
        00F0   865 ;         R0,R1,R2,R3 ARE NOT PRESERVED.
        00F0   866 ;--
        00F0   867
        00F0   868 MA$RAVAIL::
        00F0   869
     05 00F0   871           RSB
        00F1   872
        00F1  1175           .END
```

```
ADP$W_UMR_DIS                   = 00000256              UCB$B_SLAVE                     = 00000090
C780_CIKE                       = 00000001              UCB$L_DDT                       = 00000088
CI$INITIAL                        0000001F RG     02    UCB$L_FPC                       = 0000000C
CI$INT                            00000000 RG     02    UCB$L_FR3                       = 00000010
CI$SHUTDOWN                       0000001F RG     02    UCB$V_INT                       = 00000001
CPU_TYPE                        = 00000004              UCB$W_STS                       = 00000064
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
EXE$RH780_INT                     ********    X   02
IDB$L_ADP                       = 00000014
IDB$L_CSR                       = 00000000
IDB$L_OWNER                     = 00000004
IDB$L_UCBLST                    = 00000018
INI$MPMADP                        000000ED RG     02
IO$GL_UBA_INT0                    ********    X   02
MA$INITIAL                        000000EE RG     02
MA$RAVAIL                         000000F0 RG     02
MA$REQUEST                        000000EF RG     02
MBA$INITIAL                       000000E4 RG     02
MBA$INT                           0000006C RG     02
MBA$L_AS                        = 00000410
MBA$L_CR                        = 00000004
MBA$L_CSR                       = 00000000
MBA$L_SR                        = 00000008
MBA$M_CR_IE                     = 00000004
MBA$M_CR_INIT                   = 00000001
MBA$M_CSR_PD                    = 00800000
NUMMPMVEC                       = 00000010
PA_CNF                            00000000
PA_CNF_M_PDN                    = 00800000
PA_CNF_M_PUP                    = 00400000
PA_PMC                            00000004
PA_PMC_M_MIN                    = 00000001
PR$_SID_TYP730                  = 00000003
PR$_SID_TYP750                  = 00000002
PR$_SID_TYP780                  = 00000001
PR$_SID_TYP790                  = 00000004
PR$_SID_TYPUV1                  = 00000007
SIZ...                          = 00000006
UBA$INITIAL                       00000045 RG     02
UBA$INT0                          00000060 RG     02
UBA$L_CR                        = 00000004
UBA$L_CSR                       = 00000000
UBA$L_SR                        = 00000008
UBA$M_CR_BRIE                   = 00000020
UBA$M_CR_CNFIE                  = 00000004
UBA$M_CR_IFSIE                  = 00000040
UBA$M_CR_SUEFIE                 = 00000008
UBA$M_CR_USEFIE                 = 00000010
UBA$V_CR_MRDSB                  = 0000001A
UBA_UNEXINT                       00000068 RG     02
```

ADP
V04

```
                                               +------------------+
                                               ! 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                            000000F1  (   241.)  02 (   2.)  NOPIC     USR    CON    REL    LCL NOSHR   EXE   RD     WRT NOVEC LONG

                                            +--------------------------+
                                            ! Performance indicators !
                                            +--------------------------+

Phase                         Page faults    CPU Time       Elapsed Time
-----                         -----------    --------       ------------
Initialization                         29    00:00:00.05    00:00:02.13
Command processing                    129    00:00:00.57    00:00:03.97
Pass 1                                541    00:00:13.90    00:00:53.79
Symbol table sort                       0    00:00:02.21    00:00:07.70
Pass 2                                113    00:00:02.88    00:00:10.56
Symbol table output                     8    00:00:00.08    00:00:00.29
Psect synopsis output                   2    00:00:00.01    00:00:00.53
Cross-reference output                  0    00:00:00.00    00:00:00.00
Assembler run totals                  824    00:00:19.70    00:01:18.97

The working set limit was 1950 pages.
131956 bytes (258 pages) of virtual memory were used to buffer the intermediate code.
There were 110 pages of symbol table space allocated to hold 2138 non-local and 6 local symbols.
1179 source lines were read in Pass 1, producing 13 object records in Pass 2.
38 pages of virtual memory were used to define 37 macros.

                                           +----------------------------+
                                           ! Macro library statistics !
                                           +----------------------------+

Macro library name                                 Macros defined
------------------                                 --------------
_$255$DUA28:[SYSLOA.OBJ]790DEF.MLB;1                          0
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                               25
_$255$DUA28:[SYSLIB]STARLET.MLB;2                             7
TOTALS (all libraries)                                       32

2215 GETS were required to define 32 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:ADPSUB790/OBJ=OBJ$:ADPSUB790 MSRC$:CPUSW790/UPDATE=(ENH$:CPUSW790)+MSRC$:ADPSUB/UPDATE=(ENH$:ADPSUB)+EXECML$/LIB+LIB$
```

0392 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY