```
   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
```

FILEID**ADPSUBUV1

```
  AAAAAA    DDDDDDDD    PPPPPPPP      SSSSSSSS  UU      UU  BBBBBBBB    UU      UU  VV      VV      11
  AAAAAA    DDDDDDDD    PPPPPPPP      SSSSSSSS  UU      UU  BBBBBBBB    UU      UU  VV      VV      11
AA      AA  DD      DD  PP      PP  SS            UU      UU  BB      BB  UU      UU  VV      VV    1111
AA      AA  DD      DD  PP      PP  SS            UU      UU  BB      BB  UU      UU  VV      VV    1111
AA      AA  DD      DD  PP      PP  SS            UU      UU  BB      BB  UU      UU  VV      VV      11
AA      AA  DD      DD  PPPPPPPP      SSSSSS      UU      UU  BBBBBBBB    UU      UU  VV      VV      11
AA      AA  DD      DD  PPPPPPPP      SSSSSS      UU      UU  BBBBBBBB    UU      UU  VV      VV      11
AAAAAAAAAA  DD      DD  PP                  SS    UU      UU  BB      BB  UU      UU  VV      VV      11
AAAAAAAAAA  DD      DD  PP                  SS    UU      UU  BB      BB  UU      UU  VV      VV      11
AA      AA  DD      DD  PP                  SS    UU      UU  BB      BB  UU      UU    VV  VV        11
AA      AA  DD      DD  PP                  SS    UU      UU  BB      BB  UU      UU    VV  VV        11
AA      AA  DDDDDDDD    PP            SSSSSSSS    UUUUUUUUUU  BBBBBBBB    UUUUUUUUUU      VV      111111
AA      AA  DDDDDDDD    PP            SSSSSSSS    UUUUUUUUUU  BBBBBBBB    UUUUUUUUUU      VV      111111

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

```
                                E 4
0000     1          .NOSHOW CONDITIONALS
0000     5
0000     9
0000    13
0000    17
0000    19          .TITLE  ADPSUBUV1 - ADAPTER SUBROUTINES FOR MICRO-VAX I
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
                     0000    98
                     0000   102
00000000             0000   104          C780_LIKE = 0
                     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   145
                00000000    146          .PSECT  SYSLOA,LONG
```

```
                        .SBTTL  CI$INT - CI INTERRUPT HANDLER
                ;+
                ; CI$INT - CI INTERRUPT HANDLER
                ;
                ;       THIS MODULE IS A DUMMY CI32 INTERRUPT HANDLER WHICH IS USED
                ;       UNTIL THE REAL CI DRIVER (PADRIVER) IS LOADED. IT ALSO CONTAINS
                ;       A DUMMY CI32 CONTROLLER INITIALIZATION  ENTRY POINT.
                ;
                ; INPUTS:
                ;
                ;       THE STACK ON ENTRY IS AS FOLLOWS:
                ;
                ;               0(SP)           ADDRESS OF IDB ADDRESS
                ;       4(SP) - 16(SP)          SAVED R2 - R5
                ;               20(SP)          INTERRUPT PC
                ;               24(SP)          INTERRUPT PSL
                ;
                ; OUTPUTS:
                ;
                ;       NONE
                ;
                ; SIDE EFFECTS:
                ;
                ;       INTERRUPTS ARE DISABLED ON THE CI32
                ;-

                CI$INITIAL::                                    ; CONTROLLER INITIALIZATION
                CI$SHUTDOWN::                                   ; CONTROLLER SHUTDOWN

05              RSB
```

```
                                    I  4
     0001   237          .SBTTL  DR$INT - DR INTERRUPT HANDLER
     0001   238 ;+
     0001   239 ; DR$INT - DR INTERRUPT HANDLER
     0001   240 ;
     0001   241 ;       THIS MODULE IS A DUMMY DR32 INTERRUPT HANDLER WHICH IS USED
     0001   242 ;       UNTIL THE REAL DR DRIVER (XFDRIVER) IS LOADED. IT ALSO CONTAINS
     0001   243 ;       A DUMMY DR32 CONTROLLER INITIALIZATION  ENTRY POINT.
     0001   244 ;
     0001   245 ; INPUTS:
     0001   246 ;
     0001   247 ;       THE STACK ON ENTRY IS AS FOLLOWS:
     0001   248 ;
     0001   249 ;               0(SP)           ADDRESS OF IDB ADDRESS
     0001   250 ;          4(SP) - 16(SP)       SAVED R2 - R5
     0001   251 ;              20(SP)           INTERRUPT PC
     0001   252 ;              24(SP)           INTERRUPT PSL
     0001   253 ;
     0001   254 ; OUTPUTS:
     0001   255 ;
     0001   256 ;       NONE
     0001   257 ;
     0001   258 ; SIDE EFFECTS:
     0001   259 ;
     0001   260 ;       INTERRUPTS ARE DISABLED ON THE DR32
     0001   261 ;-
     0001   262
     0001   324
     0001   325 DR$INITIAL::                                    ; CONTROLLER INITIALIZATION
     0001   326 DR$SHUTDOWN::                                   ; CONTROLLER SHUTDOWN
     0001   327
     0001   334
05   0001   335          RSB
```

```
                                                                .SBTTL  UBA$INITIAL - CPU-DEPENDENT UNIBUS ADAPTER INITIALIZATION
                                                        ;+
                                                        ; UBA$INITIAL - UNIBUS ADAPTER INITIALIZATION
                                                        ;
                                                        ; THIS ROUTINE IS CALLED VIA A JSB INSTRUCTION AT SYSTEM STARTUP AND AFTER
                                                        ; A POWER RECOVERY RESTART TO ALLOW INITIALIZATION OF UNIBUS ADAPTERS.
                                                        ;       (POWERFAIL AND INITADP)
                                                        ;
                                                        ; INPUTS:
                                                        ;
                                                        ;       R2 = ADDRESS OF ADAPTER CONTROL BLOCK  (11/780 AND 11/750)
                                                        ;       R4 = ADDRESS OF UNIBUS ADAPTER CONFIGURATION STATUS REGISTER (11/780)
                                                        ;
                                                        ;       ALL INTERRUPTS ARE LOCKED OUT.
                                                        ;
                                                        ; OUTPUTS:
                                                        ;
                                                        ;       THE UNIBUS ADAPTER IS INITIALIZED AND INTERRUPTS ARE ENABLED.
                                                        ;-

                                                        UBA$INITIAL::                           ;UNIBUS ADAPTER INITIALIZATION



                                                        10$:                                    ;NO SPECIAL INIT FOR 11/730 OR uVAX I
                                                05              RSB                             ;
                                                        ;
                                                        ; IGNORE UNEXPECTED UNIBUS INTERRUPTS
                                                        ;

                                                                .ALIGN  LONG

                                                        UBA$INTO::                              ; PASSIVE RELEASES THROUGH VECTOR 0

                                00000000'9F     D6              INCL    @#IO$GL_UBA_INTO        ; COUNT THEM
                                             00 11              BRB     UBA_UNEXINT             ; JOIN COMMON CODE, VECTORS ARE ALLIGNED

                                                                .ALIGN  LONG

                                                        ;
                                                        ; NOTE: UBA$UNEXINT is the label in the EXEC that is a JMP @#UBA_UNEXINT.
                                                        ;       This seeming duplicity is necesary since there is code that must
                                                        ;       refer to the EXEC address from within the SYSLOA image.
                                                        ;
                                                        UBA_UNEXINT::                           ; UNEXPECTED INTERRUPT CODE

                                                                                                ; FOR 11/750, NO REGISTERS SAVED
                                                02              REI                             ; IGNORE INTERRUPT
```

```
                000D   535          .SBTTL  MASSBUS ADAPTER INITIALIZATION
                000D   536 ;+
                000D   537 ; MBA$INITIAL - MASSBUS ADAPTER INITIALIZATION
                000D   538 ;
                000D   539 ; THIS ROUTINE IS CALLED VIA A JSB INSTRUCTION AT SYSTEM STARTUP AND AFTER
                000D   540 ; A POWER RECOVERY RESTART TO ALLOW INITIALIZATION OF MASSBUS ADAPTERS.
                000D   541 ;
                000D   542 ; INPUTS:
                000D   543 ;
                000D   544 ;       R4 = CSR ADDRESS OF MASSBUS ADAPTER.
                000D   545 ;       R5 = ADDRESS OF ADAPTER IDB.
                000D   546 ;
                000D   547 ;       ALL INTERRUPTS ARE LOCKED OUT.
                000D   548 ;
                000D   549 ; OUTPUTS:
                000D   550 ;
                000D   551 ;       THE MASSBUS ADAPTER IS INITIALIZED AND INTERRUPTS ARE ENABLED.
                000D   552 ;-
                000D   553
                000D   554 MBA$INITIAL::                           ;MASSBUS ADAPTER INITIALIZATION
                000D   555
                000D   564
     05         000D   565          RSB
```

```
                     000E   567          .SBTTL  INI$MPMADP - BUILD ADP AND INITIALIZE MULTI-PORT MEMORY
                     000E   568 ;+
                     000E   569 ; INI$MPMADP IS CALLED AFTER  MAPPING THE REGISTERS FOR A MULTI-PORT
                     000E   570 ; MEMORY ADAPTER.  AN ADAPTER CONTROL BLOCK IS ALLOCATED AND FILLED.
                     000E   571 ; THE HARDWARE ADAPTER IS THEN INITIALIZED BY CALLING MPM$INITIAL.
                     000E   572 ;
                     000E   573 ; NOTE: THIS ROUTINE HAS BEEN LOCATED HERE IN SYSLOAXXX.EXE INSTEAD OF
                     000E   574 ;       INILOA.EXE BECAUSE IT CAN BE CALLED WHILE THE SYSTEM IS RUNNING
                     000E   575 ;       LONG AFTER INILOA.EXE HAS BEEN DELETED!!!
                     000E   576 ;
                     000E   577 ; INPUT:
                     000E   578 ;       R4 - nexus identification number of this nexus
                     000E   579 ;
                     000E   580 ; OUTPUTS:
                     000E   581 ;       ALL REGISTERS PRESERVED
                     000E   582 ;-
                     000E   583
00000010             000E   584 NUMMPMVEC = 16                          ; NUMBER OF INTER-PORT INTERRUPT VECTORS
                     000E   585
                     000E   586 INI$MPMADP::                            ; INITIALIZE MPM DATA STRUCTURES
                     000E   587
                  05 000E   589          RSB                           ; DUMMY ENTRY FOR SYSGEN
                     000F   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.
                ;--
                MA$INITIAL::                            ; INTIALIZE PORT
                ;
05              RSB
                ;
```

```
                                N 4
0010   730          .SBTTL  INTER-PROCESSOR REQUEST HANDLER
0010   731 ;++
0010   732 ;
0010   733 ; FUNCTIONAL DESCRIPTION:
0010   734 ;
0010   735 ;       THIS ROUTINE IS CALLED BY A DRIVER OR AN EXEC FUNCTION TO
0010   736 ;       EITHER SEND A REQUEST TO OR JUST INTERRUPT ANOTHER PROCESSOR
0010   737 ;       THAT IS CONNECTED TO A PORT OF THE MULTIPORT MEMORY.
0010   738 ;
0010   739 ; INPUTS:
0010   740 ;
0010   741 ;       R4 = ADAPTER CONTROL BLOCK ADDRESS.
0010   742 ;       R5 = IF LSS 0 - ADDRESS OF A FORK BLOCK TO USE IF REQUEST
0010   743 ;                       BLOCK IS NOT AVAILABLE.
0010   744 ;            IF GEQ 0 - PORT NUMBER OF PROCESSOR TO JUST INTERRUPT.
0010   745 ;
0010   746 ; OUTPUTS:
0010   747 ;
0010   748 ;       WHEN THIS ROUTINE IS CALLED WITH A FORK BLOCK ADDRESS, IT WILL
0010   749 ;       ATTEMPT TO ALLOCATE A REQUEST BLOCK.  IF THE REQUEST FAILS,
0010   750 ;       THE CONTEXT OF THE CALLER WILL BE SAVED IN THE FORK BLOCK, THE
0010   751 ;       FORK BLOCK BLOCK WILL BE INSERTED IN THE REQUEST BLOCK WAIT
0010   752 ;       QUEUE AND A RETURN TO THE CALLER'S CALLER IS EXECUTED.
0010   753 ;
0010   754 ;       IF A REQUEST BLOCK IS ALLOCATED SUCCESSFULLY, CONTROL WILL
0010   755 ;       RETURN TO THE CALLER VIA A CO-ROUTINE CALL SO THE CALLER CAN
0010   756 ;       FILL-IN THE REQUEST BLOCK.
0010   757 ;
0010   758 ;       THE CALLER WILL THEN PERFORM ANOTHER CO-ROUTINE CALL TO RETURN
0010   759 ;       TO THIS ROUTINE SO THE BLOCK CAN BE INSERTED IN THE DESIRED
0010   760 ;       PROCESSOR'S INTER-PROCESSOR REQUEST QUEUE.  IF IT IS THE
0010   761 ;       FIRST REQUEST IN THE QUEUE AN INTER-PORT INTERRUPT WILL
0010   762 ;       ALSO BE REQUESTED TO WAKE-UP THE DISPATCHER ON THE PORT.
0010   763 ;
0010   764 ;
0010   765 ;       IF THIS ROUTINE IS CALLED WITH A PORT NUMBER INSTEAD OF A
0010   766 ;       FORK BLOCK ADDRESS, IT WILL JUST REQUEST AN INTERRUPT FOR
0010   767 ;       THE PROCESSOR ON THE SPECIFIED PORT.  IT IS THEN UP TO THE
0010   768 ;       INTERRUPTED PROCESSOR TO DETERMINE WHAT THE INTERRUPT WAS
0010   769 ;       FOR.
0010   770 ;
0010   771 ;       R0 = SUCCESS OR FAILURE OF OPERATION.  THIS SHOULD BE CHECKED
0010   772 ;            BY THE CALLER BOTH TIMES THIS ROUTINE RETURNS.
0010   773 ;
0010   774 ;       R3,R4,R5 ARE PRESERVED.
0010   775 ;
0010   776 ;--
0010   777
0010   778 MA$REQUEST::                             ; REQUEST HANDLER
0010   779
05  0010   781          RSB
0011   782
```

```
                0011   847          .SBTTL  REPORT RESOURCE AVAILABILITY TO INTERESTED PORTS
                0011   848 ;++
                0011   849 ;
                0011   850 ; FUNCTIONAL DESCRIPTION:
                0011   851 ;
                0011   852 ;       THIS ROUTINE IS CALLED TO REPORT TO ANY PROCESSORS THAT A RESOURCE
                0011   853 ;       HAS BEEN MADE AVAILABLE.
                0011   854 ;
                0011   855 ; INPUTS:
                0011   856 ;
                0011   857 ;       R0 = RESOURCE NUMBER OF RESOURCE MADE AVAILABLE.
                0011   858 ;       R1 = SHARED MEMORY CONTROL BLOCK (SHB) ADDRESS.
                0011   859 ;
                0011   860 ; OUTPUTS:
                0011   861 ;
                0011   862 ;       ANY PROCESSORS WAITING FOR THE SPECIFIED RESOURCE ARE INTERRUPTED
                0011   863 ;       TO NOTIFY THEM THE RESOURCE IS AVAILABLE.
                0011   864 ;
                0011   865 ;       R0,R1,R2,R3 ARE NOT PRESERVED.
                0011   866 ;--
                0011   867
                0011   868 MA$RAVAIL::
                0011   869
     05         0011   871          RSB
                0012   872
                0012  1175          .END
```

```
C780_LIKE                       = 00000000
CI$INITIAL                        00000000 RG    02
CI$SHUTDOWN                       00000000 RG    02
CPU_TYPE                        = 00000007
DR$INITIAL                        00000001 RG    02
DR$SHUTDOWN                       00000001 RG    02
INI$MPMADP                        0000000E RG    02
IO$GL_UBA_INTO                    ******** X     02
MA$INITIAL                        0000000F RG    02
MA$RAVAIL                         00000011 RG    02
MA$REQUEST                        00000010 RG    02
MBA$INITIAL                       0000000D RG    02
NUMMPMVEC                       = 00000010
PR$_SID_TYP730                  = 00000003
PR$_SID_TYP750                  = 00000002
PR$_SID_TYP780                  = 00000001
PR$_SID_TYP790                  = 00000004
PR$_SID_TYPUV1                  = 00000007
UBA$INITIAL                       00000002 RG    02
UBA$INTO                          00000004 RG    02
UBA_UNEXINT                       0000000C RG    02
```

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	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SYSLOA	00000012 (18.)	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	00:00:02.65
Command processing	106	00:00:00.50	00:00:04.75
Pass 1	466	00:00:11.66	00:00:43.72
Symbol table sort	9	00:00:01.91	00:00:07.45
Pass 2	78	00:00:02.53	00:00:10.41
Symbol table output	4	00:00:00.05	00:00:00.04
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	697	00:00:16.72	00:01:09.05

```
The working set limit was 1500 pages.
110652 bytes (217 pages) of virtual memory were used to buffer the intermediate code.
There were 100 pages of symbol table space allocated to hold 1886 non-local and 1 local symbols.
1179 source lines were read in Pass 1, producing 13 object records in Pass 2.
27 pages of virtual memory were used to define 26 macros.
```

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

Macro library name                                  Macros defined
------------------                                  --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                              17
_$255$DUA28:[SYSLIB]STARLET.MLB;2                            6
TOTALS (all libraries)                                      23

2008 GETS were required to define 23 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:ADPSUBUV1/OBJ=OBJ$:ADPSUBUV1 MSRC$:CPUSWUV1/UPDATE=(ENH$:CPUSWUV1)+MSRC$:ADPSUB/UPDATE=(ENH$:ADPSUB)+EXECML$/LIB
```

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CNXOPT
LIS

ADPSUBUV1
LIS

CNXMAN
LIS

CJFCLUSTR
LIS

CLUSTRLOA
LIS

CONMAN
LIS

ADPSUB790
LIS

CLUMESSAG
LIS