


```

AAAAAA DDDDDDD PP PPPPPP SSSSSSS UU UU BBBB BBBB UU UU VV VV 11
AAAAAA DDDDDDD PP PPPPPP SSSSSSS UU UU BBBB BBBB UU UU VV VV 11
AA AA DD DD PP PP SS SS UU UU BB BB UU UU VV VV 1111
AA AA UD DD PP PP SS SS UU UU BB BB UU UU VV VV 1111
AA AA DD DD PP PP SS SS UU UU BB BB UU UU VV VV 11
AA AA DD DD PP PP SS SS UU UU BB BB UU UU VV VV 11
AA AA DD DD PPPPPPP UU UU BBBB BBBB UU UU VV VV 11
AA AA DD DD PPPPPPP UU UU BBBB BBBB UU UU VV VV 11
AAAAAAAAA DD DD PP SS SS UU UU BB BB UU UU VV VV 11
AAAAAAAAA DD DD PP SS SS UU UU BB BB UU UU VV VV 11
AA AA DD DD PP SS SS UU UU BB BB UU UU VV VV 11
AA AA DD DD FP SS SS UU UU BB BB UU UU VV VV 11
AA AA DDDDDDD PP SSSSSSS UUUUUUUUUU BBBB BBBB UUUUUUUUUU VV VV 111111
AA AA DDDDDDD PP SSSSSSS UUUUUUUUUU BBBB BBBB UUUUUUUUUU VV VV 111111

```

```

LL IIIIII SSSSSSS
LL IIIIII SSSSSSS
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
LLLLLLLLLL IIIIII SSSSSSS
LLLLLLLLLL IIIIII SSSSSSS

```

ADP
Syn
C71
C19
C19
CPU
DR1
DR1
IN1
IO1
MA1
MA1
MA1
MB1
NUP
PR1
PR1
PR1
PR1
UB1
UB1
PSE

SAS
SYS
Pha

In
Com
Pas
Syn
Pas
Syn
Pse
Cre
As
The
110
The
11
27

(3)	148	CISINT - CI INTERRUPT HANDLER
(4)	237	DRSINT - DR INTERRUPT HANDLER
(5)	337	UBASINITIAL - CPU-DEPENDENT UNIBUS ADAPTER INITIALIZATION
(5)	535	MASSBUS ADAPTER INITIALIZATION
(6)	567	INISMPMADP - BUILD ADP AND INITIALIZE MULTI-PORT MEMORY
(6)	661	MA\$INITIAL - INITIALIZE MULTI-PORT MEMORY ADAPTER
(6)	730	INTER-PROCESSOR REQUEST HANDLER
(6)	847	REPORT RESOURCE AVAILABILITY TO INTERESTED PORTS

ADI
VA

Ma
--
-S
-S
TO
20
Th
MA

```
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$S_SHB properly again; V03-004 ASSUMEd this field
0000 64 : was at a certain constant offset, and a change to the ADP
0000 65 : moved it. Note - this is a 780 change only.
0000 66
0000 67 : V03-005 KDM0081 Kathleen D. Morse 13-Sep-1983
0000 68 : Create version for Micro-VAX I.
0000 69
0000 70 : V03-004 ROW0196 Ralph O. Weber 27-JUL-1983
0000 71 : Correct INISMPMADP so the ADP$S_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. MASINT 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 ADPSL_AVECIOR in INISMPMADP.
0000 84 :
0000 85 :--
```

```
00000000 0000 90
00000000 0000 94
00000000 0000 98
00000000 0000 102
00000000 0000 104      C780_LIKE = 0
00000000 0000 106      :
00000000 0000 107      : MACRO LIBRARY CALLS
00000000 0000 108      :
00000000 0000 109      $ADPDEF      : Define ADP offsets.
00000000 0000 110      $CRBDEF      : Define CRB offsets.
00000000 0000 111      $DCDEF      : Define AT codes.
00000000 0000 112      $DDBDEF      : Define DDB offsets.
00000000 0000 113      $DDTDEF      : Define DDT offsets.
00000000 0000 114      $DYNDEF      : Define data structure type codes.
00000000 0000 115      $IDBDEF      : Define interrupt dispatcher offsets.
00000000 0000 116      $MBADEF      : Define MASSBUS registers.
00000000 0000 117      $MCHKDEF     : Define machine check masks.
00000000 0000 118      $MPMDEF      : Define multi-port memory.
00000000 0000 119      $NDTDEF      : Define nexus device types.
00000000 0000 120      $PRDEF      : Define IPR numbers.
00000000 0000 121      $PTEDEF      : Define Page Table Entry bits.
00000000 0000 122      $RPBDEF      : Define Restart Parameter Block fields.
00000000 0000 123      $SSDEF      : Define system service codes.
00000000 0000 124      $UBADEF      : Define UBA register offsets.
00000000 0000 125      $SUBIDEF     : Define UNIBUS interconnect
00000000 0000 126      : register offsets.
00000000 0000 127      $UCBDEF      : Define unit control block.
00000000 0000 128      $VADEF      : Define virtual address fields.
00000000 0000 129      $VECDEF      : Define vec offsets.
00000000 0000 130
00000000 0000 141
00000000 0000 145
00000000 0000 146      .PSECT SYSLOA, LONG
```

```
0000 148      .SBTTL  CISINT - CI INTERRUPT HANDLER
0000 149      :+
0000 150      : CISINT - 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 224      :
0000 225      CIS$INITIAL::           ; CONTROLLER INITIALIZATION
0000 226      CIS$SHUTDOWN::       ; CONTROLLER SHUTDOWN
0000 227      :
0000 234      :
05 0000 235      RSB
```

```
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 324
0001 325 DR$INITIAL::                ; CONTROLLER INITIALIZATION
0001 326 DR$SHUTDOWN::            ; CONTROLLER SHUTDOWN
0001 327
0001 334
05 0001 335 RSB
```



```

0002 337      .SBTTL  UBAS$INITIAL - CPU-DEPENDENT UNIBUS ADAPTER INITIALIZATION
0002 338      :+
0002 339      : UBAS$INITIAL - UNIBUS ADAPTER INITIALIZATION
0002 340      :
0002 341      : THIS ROUTINE IS CALLED VIA A JSB INSTRUCTION AT SYSTEM STARTUP AND AFTER
0002 342      : A POWER RECOVERY RESTART TO ALLOW INITIALIZATION OF UNIBUS ADAPTERS.
0002 343      : (POWERFAIL AND INITADP)
0002 344      :
0002 345      : INPUTS:
0002 346      :
0002 347      : R2 = ADDRESS OF ADAPTER CONTROL BLOCK (11/780 AND 11/750)
0002 348      : R4 = ADDRESS OF UNIBUS ADAPTER CONFIGURATION STATUS REGISTER (11/780)
0002 349      :
0002 350      : ALL INTERRUPTS ARE LOCKED OUT.
0002 351      :
0002 352      : OUTPUTS:
0002 353      :
0002 354      : THE UNIBUS ADAPTER IS INITIALIZED AND INTERRUPTS ARE ENABLED.
0002 355      :-
0002 356
0002 357 UBAS$INITIAL::          ;UNIBUS ADAPTER INITIALIZATION
0002 358
0002 373
0002 383
05 0002 384 10$:              ;NO SPECIAL INIT FOR 11/730 OR UVAX I
0002 385          RSB          ;
0002 386          :
0002 387          : IGNORE UNEXPECTED UNIBUS INTERRUPTS
0002 388          :
0002 389          :
0002 390          .ALIGN LONG
0002 391
0002 392 UBAS$INTO::          ; PASSIVE RELEASES THROUGH VECTOR 0
0002 393
0002 394          INCL  @#IOSGL UBA_INT0      ; COUNT THEM
0002 395          BRB   UBA_UNEXINT        ; JOIN COMMON CODE, VECTORS ARE ALLIGNED
0002 396
0002 397          .ALIGN LONG
0002 398
0002 399          :
0002 400          : NOTE: UBAS$UNEXINT is the label in the EXEC that is a JMP @#UBA UNEXINT.
0002 401          : This seeming duplicity is necessary since there is code that must
0002 402          : refer to the EXEC address from within the SYSLOA image.
0002 403          :
0002 404          UBA_UNEXINT::          ; UNEXPECTED INTERRUPT CODE
0002 405
0002 412          : FOR 11/750, NO REGISTERS SAVED
02 0002 414          REI          ; IGNORE INTERRUPT
0002 533

```

```
000D 535      .SBTTL MASSBUS ADAPTER INITIALIZATION
000D 536      :+
000D 537      : MBASINITIAL - 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      : MBASINITIAL::                                ;MASSBUS ADAPTER INITIALIZATION
000D 555      :
000D 564      :
05 000D 565      RSB
```

```
000E 567 .SBTTL INISMPMADP - BUILD ADP AND INITIALIZE MULTI-PORT MEMORY
000E 568 :+
000E 569 : INISMPMADP 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 MPMSINITIAL.
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 INISMPMADP:: ; INITIALIZE MPM DATA STRUCTURES
000E 587
05 000E 588 RSB ; DUMMY ENTRY FOR SYSGEN
000F 589
000E 590
```

CJF
Sym
CDF
CDF
CJF
CJF
CJF
CJF
CJF
CLS
CLS
CLS
CLS
CLS
CLS
CLL
CLL
CNY
CNY
CNY
CSE
EXE
IPL
IPL
SSS

PSE

\$AE
SSS

Pha

Ini
Com
Pas
Syn
Pas
Syn
Pse
Crc
Ass

The
595
The
194
15

```
000F 661 .SBTTL MASINITIAL - INITIALIZE MULTI-PORT MEMORY ADAPTER
000F 662 :++
000F 663 :
000F 664 : MPMSINITIAL - INITIALIZE MULTI-PORT MEMORY ADAPTER
000F 665 :
000F 666 : THIS ROUTINE IS CALLED AT SYSTEM INTIALIZATION AND AFTER A POWER
000F 667 : RECOVERY RESTART TO INITIALIZE THE PORT ADAPTER BY CLEARING ANY
000F 668 : ERRORS AND ENABLING ALL INTERRUPTS.
000F 669 :
000F 670 : INPUTS:
000F 671 :
000F 672 : R4 = ADDR OF ADAPTER CSR.
000F 673 :
000F 674 : IPL = 31
000F 675 :
000F 676 : OUPUTS:
000F 677 :
000F 678 : ANY ERRORS IN PORT ARE CLEARED AND ALL INTERRUPTS ARE ENABLED.
000F 679 :--
000F 680 :
000F 681 MASINITIAL:: : INTIALIZE PORT
000F 682 :
05 000F 684 RSB
0010 685
```

```
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 MULTIPOINT 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 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 MASREQUEST:: ; 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 MASRAVAIL::
0011 869 :
05 0011 871 RSB
0012 872 :
0012 1175 .END
```

ADPSUBUV1
Symbol table

- ADAPTER SUBROUTINES FOR MICRO-VAX I

16-SEP-1984 01:06:00 VAX/VMS Macro V04-00
5-SEP-1984 04:06:45 [SYSLOA.SRC]ADPSUB.MAR;1

Page 12
(6)

CL
VO

```

C780 LIKE = 00000000
CISINITIAL 00000000 RG 02
CISSHUTDOWN 00000000 RG 02
CPU TYPE = 00000007
DR$INITIAL 00000001 RG 02
DR$SHUTDOWN 00000001 RG 02
INISMPMADP 0000000E RG 02
IOSGL UBA INTO ***** X 02
MASINITIAL 0000000F RG 02
MASRAVAIL 00000011 RG 02
MASREQUEST 00000010 RG 02
MBASINITIAL 0000000D RG 02
NUMPMVEC = 00000010
PRS_SID_TYP730 = 00000003
PRS_SID_TYP750 = 00000002
PRS_SID_TYP780 = 00000001
PRS_SID_TYP790 = 00000004
PRS_SID_TYPUV1 = 00000007
UBASINITIAL 00000002 RG 02
UBASINTO 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
\$ABSS	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)+EXECMLS/LIB

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

This image displays a grid of numerous small document thumbnails. The thumbnails are arranged in a regular pattern, with some larger and more legible than others. Several thumbnails are clearly labeled with titles and 'LIS' (likely Logical Interchange Specification):

- ADPSUBV1 LIS
- CJFLUSTR LIS
- CLUSTRLOA LIS
- CONMAN LIS
- ADPSUB790 LIS
- CLIMESSAG LIS
- CNXOPT LIS
- CNXMAN LIS

The thumbnails themselves contain various types of content, including text, diagrams, and tables, representing different technical documents or manuals.