L 12

```
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
SSS              YYY         YYY  SSS
SSS              YYY         YYY  SSS
SSS              YYY         YYY  SSS
SSS                 YYY   YYY     SSS
SSS                 YYY   YYY     SSS
SSS                 YYY   YYY     SSS
   SSSSSSSSS           YYY           SSSSSSSSS
   SSSSSSSSS           YYY           SSSSSSSSS
   SSSSSSSSS           YYY           SSSSSSSSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
```

```
_$
Ps
--
YZ
Z$
Z$
Z$
Z$
Z$
Z$
Z$
Z$
Z$
Z$
Z$
Z$
```

```
LL            000000      AAAAAA    DDDDDDDD    MM      MM  RRRRRRRR    EEEEEEEEEE    GGGGGGGG
LL            000000      AAAAAA    DDDDDDDD    MM      MM  RRRRRRRR    EEEEEEEEEE    GGGGGGGG
LL          OO      OO  AA      AA  DD      DD  MMMM  MMMM  RR      RR  EE          GG
LL          OO      OO  AA      AA  DD      DD  MMMM  MMMM  RR      RR  EE          GG
LL          OO      OO  AA      AA  DD      DD  MM  MM  MM  RR      RR  EE          GG
LL          OO      OO  AA      AA  DD      DD  MM  MM  MM  RR      RR  EE          GG
LL          OO      OO  AA      AA  DD      DD  MM      MM  RRRRRRRR    EEEEEEEE    GG
LL          OO      OO  AA      AA  DD      DD  MM      MM  RRRRRRRR    EEEEEEEE    GG
LL          OO      OO  AAAAAAAAAA  DD      DD  MM      MM  RR  RR      EE          GG  GGGGGG
LL          OO      OO  AAAAAAAAAA  DD      DD  MM      MM  RR  RR      EE          GG  GGGGGG
LL          OO      OO  AA      AA  DD      DD  MM      MM  RR    RR    EE          GG      GG      ....
LL          OO      OO  AA      AA  DD      DD  MM      MM  RR    RR    EE          GG      GG      ....
LLLLLLLLLL    000000    AA      AA  DDDDDDDD    MM      MM  RR      RR  EEEEEEEEEE    GGGGGG        ....
LLLLLLLLLL    000000    AA      AA  DDDDDDDD    MM      MM  RR      RR  EEEEEEEEEE    GGGGGG        ....

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

**

Table of contents

```
0000            .TITLE  LOADMREG - LOAD MBA AND UBA MAP REGISTERS
0000            .IDENT  'V04-000'
0000
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    ; D. N. CUTLER 1-NOV-77
0000    ;
0000    ; LOAD MBA MAP REGISTERS
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V03-003 RLRLBCNT        Robert L. Rappaport     6-Jul-1984
0000    ;               In IOC$LUBAUDAMAP, pickup longword BCNT from IRP rather
0000    ;               than word BCNT.
0000    ;
0000    ;       V03-002 RLRPDTADP       Robert L. Rappaport     9-Apr-1984
0000    ;               Modify IOC$LUBAUDAMAP so as to pickup the pointer to the
0000    ;               ADP from PDT$L_ADP(R4).
0000    ;
0000    ;       V03-001 KDM0002         Kathleen D. Morse       28-Jun-1982
0000    ;               Add $VADEF.
0000    ;
```

```
0000    ;
0000    ; MACRO LIBRARY CALLS
0000    ;
0000
0000            $ADPDEF                         ;DEFINE ADP OFFSETS
0000            $CDRPDEF                        ;DEFINE CDRP OFFSETS
0000            $CRBDEF                         ;DEFINE CRB OFFSETS
0000            $MBADEF                         ;DEFINE MBA REGISTER OFFSET DEFINITIONS
0000            $PDTDEF                         ;DEFINE PDT OFFSETS
0000            $PTEDEF                         ;DEFINE PAGE TABLE ENTRY FIELDS
0000            $UBADEF                         ;DEFINE UCB OFFSETS
0000            $UBMDDEF                        ;DEFINE UBMD OFFSETS
0000            $UCBDEF                         ;DEFINE UCB OFFSETS
0000            $VADEF                          ;DEFINE VIRTUAL ADDRESS FIELDS
0000            $VECDEF                         ;DEFINE CRB TRANSFER VECTOR OFFSETS
```

```
                                            0000           .SBTTL  LOAD MASSBUS ADAPTER MAP REGISTERS
                                            0000   ;+
                                            0000   ; IOC$LOADMBAMAP - LOAD MASSBUS ADAPTER MAP REGISTERS
                                            0000   ;
                                            0000   ; THIS ROUTINE IS CALLED TO LOAD THE MASSBUS ADAPTER MAP REGISTERS, THE
                                            0000   ; BYTE COUNT REGISTER, AND THE VIRTUAL ADDRESS REGISTER.
                                            0000   ;
                                            0000   ; INPUTS:
                                            0000   ;
                                            0000   ;       R4 = ADDRESS OF MBA CONFIGURATION STATUS REGISTER.
                                            0000   ;       R5 = UCB ADDRESS OF UNIT TRANSFER IS TO OCCUR ON.
                                            0000   ;
                                            0000   ; OUTPUTS:
                                            0000   ;
                                            0000   ;       THE TRANSFER BYTE COUNT, STARTING PAGE OFFSET, AND ADDRESS OF THE
                                            0000   ;       PAGE TABLE ENTIRES THAT DESCRIBE THE TRANSFER ARE RETRIEVED FROM
                                            0000   ;       THE SPECIFED UCB AND USED TO LOAD THE MBA BYTE COUNT, VIRTUAL ADDRESS,
                                            0000   ;       AND MAP REGISTERS. ONE ADDITIONAL MAP REGISTER IS LOADED AS INVALID
                                            0000   ;       TO STOP THE TRANSFER IF A HARDWARE FAILURE SHOULD OCCUR.
                                            0000   ;
                                            0000   ;       R3 IS PRESERVED ACROSS CALL.
                                            0000   ;-
                                            0000
                                        00000000           .PSECT  WIONONPAGED
                                            0000   IOC$LOADMBAMAP::                        ;LOAD MASSBUS ADAPTER MAP REGISTERS
                                   53   DD  0000           PUSHL   R3                      ;SAVE REGISTERS
                          52    7E A5   3C  0002           MOVZWL  UCB$W_BCNT(R5),R2       ;GET TRANSFER BYTE COUNT
                          10 A4    52   CE  0006           MNEGL   R2,MBA$L_BCR(R4)        ;LOAD BYTE COUNT REGISTER
                          51    7C A5   3C  000A           MOVZWL  UCB$W_BOFF(R5),R1       ;GET BYTE OFFSET IN PAGE
                          0C A4    51   D0  000E           MOVL    R1,MBA$L_VAR(R4)        ;LOAD STARTING VIRTUAL ADDRESS
                          0C A4    51   D0  0012           MOVL    R1,MBA$L_VAR(R4)        ;*****TEMP UNTIL MBA ECO ******
                     52     01FF C241   9E  0016           MOVAB   ^X1FF(R2)[R1],R2        ;CALCULATE HIGHEST RELATIVE BYTE AND ROUND
                     52   52    F7 8F   78  001C           ASHL    #-9,R2,R2               ;CALCULATE NUMBER OF MAP REGISTERS TO LOAD
                        51    0800 C4   DE  0021           MOVAL   MBA$L_MAP(R4),R1        ;GET ADDRESS OF MBA MAP REGISTERS
                          50    78 A5   D0  0026           MOVL    UCB$L_SVAPTE(R5),R0     ;GET ADDRESS OF PAGE TABLE
                             81    80   D0  002A   10$:    MOVL    (R0)+,(R1)+             ;LOAD MAP REGISTER
                                   09   18  002D           BGEQ    30$                     ;IF GEQ PTE INVALID
                                F8 52   F5  002F   20$:    SOBGTR  R2,10$                  ;ANY MORE TO LOAD?
                                   61   D4  0032           CLRL    (R1)                    ;LOAD INVALID MAP ENTRY
                             53    8E   D0  0034           MOVL    (SP)+,R3                ;RESTORE REGISTER
                                        05  0037           RSB                             ;
                          53    FC A0   D0  0038   30$:    MOVL    -4(R0),R3               ;GET THE PTE (NOT FROM MAP REGISTER!)
                                 0095   30  003C           BSBW    IOC$PTETOPFN            ;GET PFN FROM INVALID PTE
            FC A1   53    80000000 8F   C9  003F           BISL3   #^X80000000,R3,-4(R1)   ;AND LOAD THE MAP REGISTER
                                   E5   11  0048           BRB     20$                     ;
```

```
                      004A          .SBTTL  LOAD UNIBUS ADAPTER MAP REGISTERS
                      004A  ;+
                      004A  ; IOC$LOADUBAMAP  - LOAD UNIBUS ADAPTER MAP REGISTERS
                      004A  ; IOC$LOADUBAMAPA - LOAD UNIBUS ADAPTER MAP REGISTERS ALTERNATE ENTRY FOR
                      004A  ;       BYTE ALIGNED UNIBJS DMA DEVICES WHICH NEVER WISH TO SET THE BYTE
                      004A  ;       OFFSET BIT IN MAP REGISTERS.  IN ALL OTHER RESPECTS THESE TWO
                      004A  ;       ENTRYPOINTS PRODUCE IDENTICAL RESULTS.
                      004A  ;
                      004A  ; THIS ROUTINE IS CALLED TO LOAD THE UNIBUS ADAPTER MAP REGISTERS.
                      004A  ;
                      004A  ; INPUTS:
                      004A  ;
                      004A  ;       R5 = UCB ADDRESS OF UNIT TRANSFER IS TO OCCUR ON.
                      004A  ;
                      004A  ;       IT IS ASSUMED THAT THE DATAPATH AND MAP REGISTERS HAVE BEEN PREVIOUSLY
                      004A  ;       ASSIGNED.
                      004A  ;
                      004A  ; OUTPUTS:
                      004A  ;
                      004A  ;       EACH MAP REGISTER IS LOADED WITH THE APPROPRIATE PAGE FRAME NUMBER
                      004A  ;       MERGED WITH THE DATAPATH DESIGNATOR AND BYTE OFFSET BIT. ONE ADDITIONAL
                      004A  ;       MAP REGISTER IS LOADED AS INVALID TO STOP THE TRANSFER IF A HARDWARE
                      004A  ;       FAILURE SHOULD OCCUR.
                      004A  ;
                      004A  ;       R3 IS PRESERVED ACROSS CALL.
                      004A  ;-
                      004A
                      004A          .ENABL  LSB
                      004A  IOC$LOADUBAMAPA::               ;LOAD UNIBUS ADAPTER MAP REGISTERS - ALTERNATE
                      004A                                  ; HERE WE DUPLICATE THE CODE IN THE OTHER ENTRY
                      004A                                  ;  EXCEPT THAT WE DO NOT CHECK WHETHER THE BYTE
                      004A                                  ;   OFFSET IS ODD.  INSTEAD WE BRANCH DIRECTLY
                      004A                                  ;    PAST THE SETTING OF THE BYTE OFFSET BIT.
        7E   53   7D  004A          MOVQ    R3,-(SP)                ;SAVE REGISTERS
     51   7C A5   3C  004D          MOVZWL  UCB$W_BOFF(R5),R1       ;GET BYTE OFFSET IN PAGE
     52   7E A5   3C  0051          MOVZWL  UCB$W_BCNT(R5),R2       ;GET TRANSFER BYTE COUNT
     53   24 A5   D0  0055          MOVL    UCB$L_CRB(R5),R3        ;GET ADDRESS OF CRB
             00   EF  0059          EXTZV   #VEC$V_DATAPATH,-       ;GET DATAPATH
             05       005B                  #VEC$S_DATAPATH,-       ; NUMBER
     54   37 A3       005C                  CRB$L_INTD+VEC$B_DATAPATH(R3),R4
             1B   11  005F          BRB     10$                     ; BRANCH AROUND TO JOIN COMMON CODE
                      0061  IOC$LOADUBAMAP::                        ;LOAD UNIBUS ADAPTER MAP REGISTERS
        7E   53   7D  0061          MOVQ    R3,-(SP)                ;SAVE REGISTERS
     51   7C A5   3C  0064          MOVZWL  UCB$W_BOFF(R5),R1       ;GET BYTE OFFSET IN PAGE
     52   7E A5   3C  0068          MOVZWL  UCB$W_BCNT(R5),R2       ;GET TRANSFER BYTE COUNT
     53   24 A5   D0  006C          MOVL    UCB$L_CRB(R5),R3        ;GET ADDRESS OF CRB
             00   EF  0070          EXTZV   #VEC$V_DATAPATH,-       ;GET DATAPATH
             05       0072                  #VEC$S_DATAPATH,-       ; NUMBER
     54   37 A3       0073                  CRB$L_INTD+VEC$B_DATAPATH(R3),R4
          03 51   E9  0076          BLBC    R1,10$                  ;IF LOC WORD ALIGNED TRANSFER
        54   10   88  0079          BISB    #^X10,R4                ;SET BYTE OFFSET BIT
   54   0400 8F   A8  007C  10$:    BISW    #^X400,R4               ;MERGE VALID WITH BYTE OFFSET AND DATAPATH
             05   E1  0081          BBC     #VEC$V_LWAE,-           ;BRANCH IF LONGWORD ACCESS NOT ENABLED
       03 37 A3       0083                  CRB$L_INTD+VEC$B_DATAPATH(R3),15$
        54   20   88  0086          BISB    #^X20,R4                ;ELSE SET LWAE FOR MAP REG
52    01FF C241   9E  0089  15$:    MOVAB   ^X1FF(R2)[R1],R2        ;CALCULATE HIGHEST RELATIVE BYTE AND ROUND
52   52   F7 8F   78  008F          ASHL    #-9,R2,R2               ;CALCULATE NUMBER OF MAP REGISTERS TO LOAD
```

```
          36 A3    52   91  0094   165           CMPB    R2,CRB$L_INTD+VEC$B_NUMREG(R3) ;ENOUGH MAP REGISTERS ASSIGNED?
                   2C   1E  0098   166           BGEQU   40$                      ;IF GEQU NO
          51    38 B3   D0  009A   167           MOVL    @CRB$L_INTD+VEC$L_ADP(R3),R1 ;GET ADDRESS OF CONFIGURATION REGISTER
                   00   EF  009E   168           EXTZV   #VEC$V_MAPREG,-          ;GET STARTING REGISTER
                   0F       00A0   169                   #VEC$S_MAPREG,-
          50    34 A3       00A1   170                   CRB$L_INTD+VEC$W_MAPREG(R3),R0
      51    0800 C140   DE  00A4   171           MOVAL   UBA$L_MAP(R1)[R0],R1     ;GET ADDRESS OF FIRST MAP REGISTER TO LOAD
          50    78 A5   D0  00AA   172           MOVL    UCB$L_SVAPTE(R5),R0      ;GET ADDRESS OF PAGE TABLE
             53    80   D0  00AE   173 20$:      MOVL    (R0)+,R3                 ;GET NEXT PAGE TABLE ENTRY
                   02   19  00B1   174           BLSS    30$                      ;IF LSS VALID PAGE TABLE ENTRY
                   1F   10  00B3   175           BSBB    IOC$PTETOPFN             ;GET PFN FROM INVALID PTE
  53    0B    15   54   F0  00B5   176 30$:      INSV    R4,#21,#11,R3            ;INSERT VALID, BYTE OFFSET, AND DATAPATH
             81    53   D0  00BA   177           MOVL    R3,(R1)+                 ;LOAD UBA MAP REGISTER
                EE 52   F5  00BD   178           SOBGTR  R2,20$                   ;ANY MORE TO LOAD?
                   61   D4  00C0   179           CLRL    (R1)                     ;LOAD INVALID MAP ENTRY
             53    8E   7D  00C2   180           MOVQ    (SP)+,R3                 ;RESTORE REGISTERS
                        05  00C5   181           RSB                              ;
                            00C6   182 40$:      BUG_CHECK UBMAPEXCED,FATAL       ;UNIBUS MAP REGISTER ALLOCATION EXCEEDED
                            00CA   183           .DSABL  LSB
```

```
                                  00CA          .SBTTL  GET PFN FROM INVALID PTE
                                  00CA     ;+
                                  00CA     ; IOC$PTETOPFN - GET PFN FROM INVALID PTE
                                  00CA     ;
                                  00CA     ; THIS ROUTINE IS CALLED TO RETURN THE PAGE FRAME NUMBER FROM A
                                  00CA     ; PAGE TABLE ENTRY WHICH HAS ALREADY BEEN DETERMINED TO BE NOT VALID.
                                  00CA     ;
                                  00CA     ; INPUTS:
                                  00CA     ;
                                  00CA     ;       R3 = PAGE TABLE ENTRY
                                  00CA     ;
                                  00CA     ; OUTPUTS:
                                  00CA     ;
                                  00CA     ;       R3 = PAGE FRAME NUMBER AND MAY INCLUDE THE FOLLOWING FIELDS
                                  00CA     ;               VALID BIT, MODIFY BIT, PROTECTION FIELD, OWNER FIELD
                                  00CA     ;
                                  00CA     ;       ALL OTHER REGISTERS PRESERVED
                                  00CA     ;-
                                  00CA
                                  00CA          .ENABL  LSB
                                  00CA     GLOBAL:
  53   00000000'FF43        D0    00CA          MOVL    @MMG$GL_GPTBASE[R3],R3  ;GLOBAL PAGE TABLE ENTRY
                   0F       19    00D2          BLSS    10$                     ;BRANCH IF VALID
                                  00D4     IOC$PTETOPFN::
   53   FB800000 8F         CA    00D4          BICL    #^C<PTE$M_TYP1 ! PTE$M_TYP0 !- ;PTE TYPE BITS
                                  00DB                  PTE$M_GPTX>,R3          ;AND GPTX/PFN
         05 53     1A       E0    00DB          BBS     #PTE$V_TYP1,R3,20$      ;BRANCH IF BAD PTE FOR I/O
         E7 53     16       E4    00DF          BBSC    #PTE$V_TYP0,R3,GLOBAL   ;BRANCH IF GLOBAL PAGE
                            05    00E3     10$: RSB
                                  00E4     20$: BUG_CHECK INVPTEFMT,FATAL       ;INVALID PAGE TABLE ENTRY FORMAT
                                  00E8          .DSABL  LSB
                                  00E8
```

```
                              00E8               .SBTTL  Load UBA map registers
                              00E8      ;++
                              00E8      ; IOC$LOADUBAMAPN - Load UBA map registers
                              00E8      ;
                              00E8      ;  Functional description:
                              00E8      ;
                              00E8      ;        This routine is called to load the UNIBUS adapter map registers.
                              00E8      ;        It differs from IOC$LOADUBAMAPN in that it does not obtain its
                              00E8      ;        variant inputs from the UCB and CRB (which are normally synchronized
                              00E8      ;        at fork IPL).  Also, the byte offset and longword aligned
                              00F8      ;        capabilities are not supported.
                              00E8      ;
                              00E8      ; Inputs:
                              00E8      ;
                              00E8      ;        R1 = buffer address
                              00E8      ;        R2 = number of map registers allocated
                              00E8      ;             (last one should be extra one for wild transfer stopper)
                              00E8      ;        R3 = starting map register allocated
                              00E8      ;        R4 = datapath number
                              00E8      ;        R5 = UCB address
                              00E8      ;
                              00E8      ; Outputs:
                              00E8      ;
                              00E8      ;        R0-R4 destroyed.  R5 preserved.
                              00E8      ;--
                              00E8      IOC$LOADUBAMAPN::                                 ;
                   54    DD   00E8               PUSHL   R4                               ; Save datapath number
   51   51   15    09    EF   00EA               EXTZV   S^#VA$V_VPN,S^#VA$S_VPN,R1,R1 ; Get buffer virtual page number
  54   00000000'GF       D0   00EF               MOVL    G^MMG$GL_SPTBASE,R4              ; Get system page table a jres
        50     6441      DE   00F6               MOVAL   (R4)[R1],R0                      ; Get first PTE address
         54   24 A5      D0   00FA               MOVL    UCB$L_CRB(R5),R4                 ; Get CRB address
         54   38 B4      D0   00FE               MOVL    @CRB$L_INTD+VEC$L_ADP(R4),R4 ; Get first UBA register address
     51   0800 C443      DE   0102               MOVAL   UBA$L_MAP(R4)[R3],R1             ; Get address of first map register
      00000400 8F        C9   0108               BISL3   #1@<UBA$V_MAP_VALID-UBA$V_MAP_DPD>,- ; Get datapath number
              54 8E           010E                       (SP)+,R4                         ; and set map register valid bit
                 52      D7   0110               DECL    R2                               ; Subtract last register from count
              53 80      D0   0112      20$:     MOVL    (R0)+,R3                         ; Get next page table entry
                 02      19   0115               BLSS    30$                              ; Br if valid page
                 BB      10   0117               BSBB    IOC$PTETOPFN                     ; Get PFN from invalid (global) page
              15 54      F0   0119      30$:     INSV    R4,#UBA$V_MAP_DPD,-              ; Insert datapath and valid bit,
              53 0B           011C                       #32-UBA$V_MAP_DPD,R3             ; clearing other PTE flags
              81 53      D0   011E               MOVL    R3,(R1)+                         ; Load the map register
              EE 52      F5   0121               SOBGTR  R2,20$                           ; Loop through all registers
                 61      D4   0124               CLRL    (R1)                             ; Invalidate last one to stop wild xfer
                 05           0126               RSB                                      ;
```

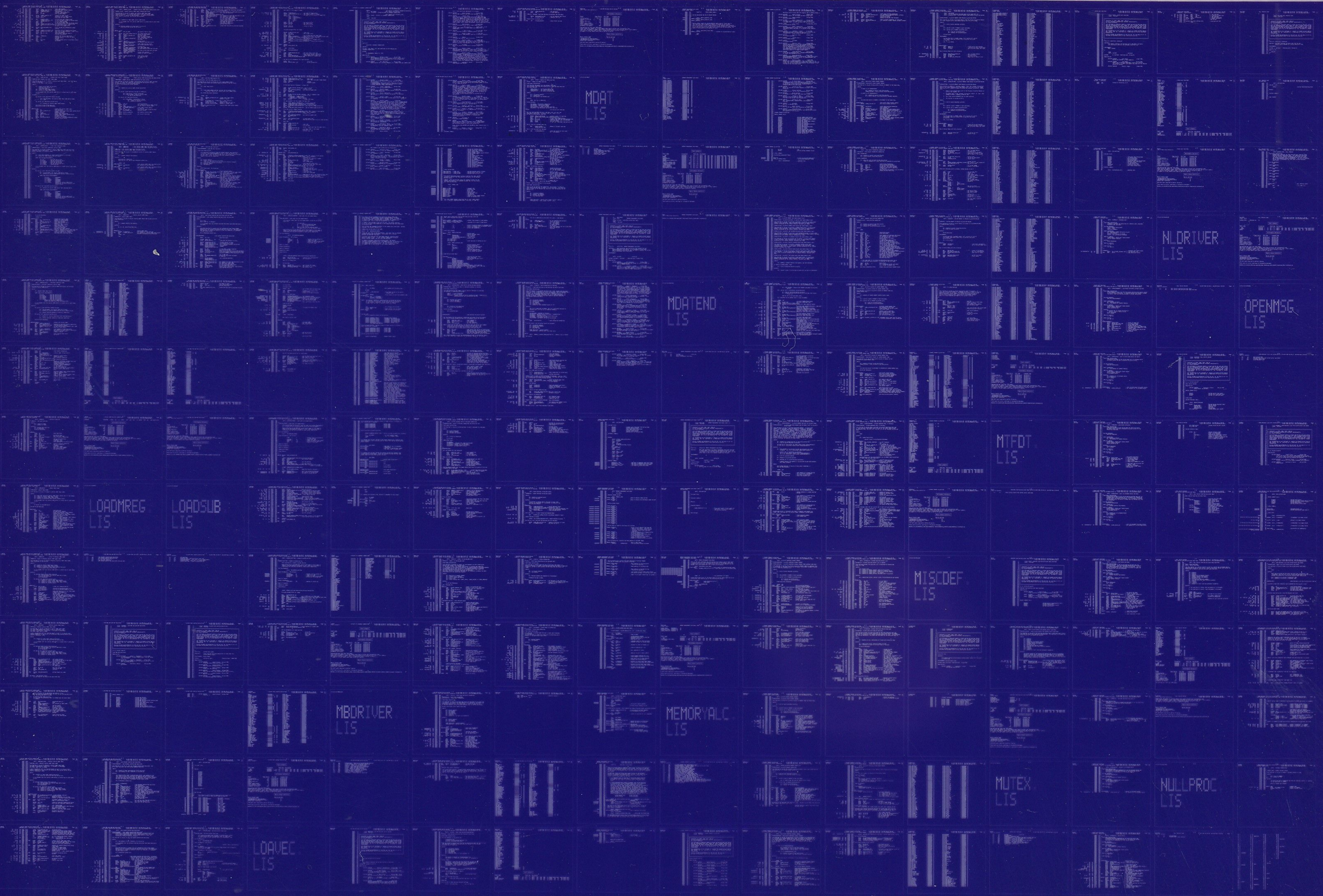
```
                              0127           .SBTTL  LOAD UNIBUS ADAPTER MAP REGISTERS FOR UDA PCRT
                              0127   ;+
                              0127   ; IOC$LUBAUDAMAP - LOAD UNIBUS ADAPTER MAP REGISTERS FOR UDA PORT
                              0127   ;
                              0127   ; INPUTS:
                              0127   ;
                              0127   ;       R4 => PDT.
                              0127   ;       R5 => CDRP OF I/O REQUEST.
                              0127   ;
                              0127   ;       IT IS ASSUMED THAT THE DATAPATH AND MAP REGISTERS HAVE BEEN PREVIOUSLY
                              0127   ;       ASSIGNED.
                              0127   ;
                              0127   ; OUTPUTS:
                              0127   ;
                              0127   ;       EACH MAP REGISTER IS LOADED WITH THE APPROPRIATE PAGE FRAME NUMBER
                              0127   ;       MERGED WITH THE DATAPATH DESIGNATOR AND BYTE OFFSET BIT. ONE ADDITIONAL
                              0127   ;       MAP REGISTER IS LOADED AS INVALID TO STOP THE TRANSFER IF A HARDWARE
                              0127   ;       FAILURE SHOULD OCCUR.
                              0127   ;
                              0127   ;       R3 IS PRESERVED ACROSS CALL.
                              0127   ;-
                              0127
                              0127   IOC$LUBAUDAMAP::                        ;LOAD UNIBUS ADAPTER MAP REGISTERS
                              0127
              7E    53    7D  0127           MOVQ    R3,-(SP)                ;SAVE REGISTERS
                              012A
                              012A           ASSUME  ADP$L_CSR       EQ      0
              00E0 D4     DD  012A           PUSHL   @PDT$[_ADP(R4)          ; Push UBA CSR address on stack.
                              012E
           51    D0 A5    3C  012E           MOVZWL  CDRP$W_BOFF(R5),R1      ; R1=BYTE OFFSET IN PAGE
           52    D2 A5    D0  0132           MOVL    CDRP$L_BCNT(R5),R2      ; R2=TRANSFER BYTE COUNT
           50    3C A5    9E  0136           MOVAB   CDRP$L_UBARSRCE(R5),R0  ; R0 => MAPPING RESOURCE DESCRIPTOR
                    00    EF  013A           EXTZV   #VEC$V_DATAPATH,-       ; GET DATAPATH
                    05        013C                   #VEC$S_DATAPATH,-       ; NUMBER
           54    03 A0        013D                   UBMD$B_DATAPATH(R0),R4
                              0140
                 03 51    E9  0140           BLBC    R1,10$                  ;IF LBC WORD ALIGNED TRANSFER
              54    10    88  0143           BISB    #^X10,R4                ;SET BYTE OFFSET BIT
         54   0400 8F     A8  0146   10$:    BISW    #^X400,R4               ;MERGE VALID WITH BYTE OFFSET AND DATAPATH
       52   01FF C241     9E  014B           MOVAB   ^X1FF(R2)[R1],R2        ;CALCULATE HIGHEST RELATIVE BYTE AND ROUND
      52    52    F7 8F   78  0151           ASHL    #-9,R2,R2               ;CALCULATE NUMBER OF MAP REGISTERS TO LOAD
                              0156
                 51 8ED0      0156           POPL    R1                      ; R1 => UBA CSR.
           02 A0    52    91  0159           CMPB    R2,UBMD$B_NUMREG(R0)    ;ENOUGH MAP REGISTERS ASSIGNED?
                    28    1E  015D           BGEQU   40$                     ;IF GEQU NO
                              015F
                    00    EF  015F           EXTZV   #VEC$V_MAPREG,-         ; GET STARTING REGISTER
                    0F        0161                   #VEC$S_MAPREG,-
              50    60        0162                   UBMD$W_MAPREG(R0),R0
                              0164
         51   0800 C140   DE  0164           MOVAL   UBA$L_MAP(R1)[R0],R1    ;GET ADDRESS OF FIRST MAP REGISTER TO LOAD
           50    CC A5    D0  016A           MOVL    CDRP$[_SVAPTE(R5),R0    ;GET ADDRESS OF PAGE TABLE
              53    80    D0  016E   20$:    MOVL    (R0)+,R3                ;GET NEXT PAGE TABLE ENTRY
                    03    19  0171           BLSS    30$                     ;IF LSS VALID PAGE TABLE ENTRY
                  FF5E    30  0173           BSBW    IOC$PTETOPFN            ;GET PFN FROM INVALID PTE
   53    0B    15   54    F0  0176   30$:    INSV    R4,#21,#11,R3           ;INSERT VALID, BYTE OFFSET, AND DATAPATH
              81    53    D0  017B           MOVL    R3,(R1)+                ;LOAD UBA MAP REGISTER
       ED 52   F5  017E    321          SOBGTR  R2,20$                  ;ANY MORE TO LOAD?
          61   D4  0181    322          CLRL    (R1)                    ;LOAD INVALID MAP ENTRY
    53    8E   7D  0183    323          MOVQ    (SP)+,R3                ;RESTORE REGISTERS
               05  0186    324          RSB                             ;
                   0187    325 40$:     BUG_CHECK UBMAPEXCED,FATAL      ;UNIBUS MAP REGISTER ALLOCATION EXCEEDED
                   018B    326          .END
```

```
ADP$L_CSR                       = 00000000
BUG$_INVPTEFMT                    ********      X    02
BUG$_UBMAPEXCED                   ********      X    02
CDRP$L_BCNT                     = FFFFFFD2
CDRP$L_SVAPTE                   = FFFFFFCC
CDRP$L_UBARSRCE                 = 0000003C
CDRP$W_BOFF                     = FFFFFFD0
CRB$L_INTD                      = 00000024
GLOBAL                            000000CA R         02
IOC$LOADMBAMAP                    00000000 RG        02
IOC$LOADUBAMAP                    00000061 RG        02
IOC$LOADUBAMAPA                   0000004A RG        02
IOC$LOADUBAMAPN                   000000E8 RG        02
IOC$LUBAUDAMAP                    00000127 RG        02
IOC$PTETOPFN                      000000D4 RG        02
MBA$L_BCR                       = 00000010
MBA$L_MAP                       = 00000800
MBA$L_VAR                       = 0000000C
MMG$GL_GPTBASE                    ********      X    02
MMG$GL_SPTBASE                    ********      X    02
PDT$L_ADP                       = 000000E0
PTE$M_GPTX                      = 003FFFFF
PTE$M_TYP0                      = 00400000
PTE$M_TYP1                      = 04000000
PTE$V_TYP0                      = 00000016
PTE$V_TYP1                      = 0000001A
UBA$L_MAP                       = 00000800
UBA$V_MAP_DPD                   = 00000015
UBA$V_MAP_VALID                 = 0000001F
UBMD$B_DATAPATH                 = 00000003
UBMD$B_NUMREG                   = 00000002
UBMD$W_MAPREG                   = 00000000
UCB$L_CRB                       = 00000024
UCB$L_SVAPTE                    = 00000078
UCB$W_BCNT                      = 0000007E
UCB$W_BOFF                      = 0000007C
VA$S_VPN                        = 00000015
VA$V_VPN                        = 00000009
VEC$B_DATAPATH                  = 00000013
VEC$B_NUMREG                    = 00000012
VEC$L_ADP                       = 00000014
VEC$S_DATAPATH                  = 00000005
VEC$S_MAPREG                    = 0000000F
VEC$V_DATAPATH                  = 00000000
VEC$V_LWAE                      = 00000005
VEC$V_MAPREG                    = 00000000
VEC$W_MAPREG                    = 00000010

                                                +----------------+
                                                ! 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
WIONONPAGED                       0000018B  (  395.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
```

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

Phase                        Page faults    CPU Time       Elapsed Time
-----                        -----------    --------       ------------
Initialization                        30    00:00:00.09    00:00:01.41
Command processing                   107    00:00:00.52    00:00:03.18
Pass 1                               287    00:00:08.90    00:00:23.96
Symbol table sort                      0    00:00:01.41    00:00:03.55
Pass 2                                75    00:00:01.70    00:00:03.86
Symbol table output                    6    00:00:00.07    00:00:00.07
Psect synopsis output                  2    00:00:00.03    00:00:00.03
Cross-reference output                 0    00:00:00.00    00:00:00.00
Assembler run totals                 509    00:00:12.73    00:00:36.07

The working set limit was 1200 pages.
49798 bytes (98 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 913 non-local and 16 local symbols.
326 source lines were read in Pass 1, producing 14 object records in Pass 2.
20 pages of virtual memory were used to define 19 macros.

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

Macro library name                                Macros defined
------------------                                --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                                12
_$255$DUA28:[SYSLIB]STARLET.MLB;2                              4
TOTALS (all libraries)                                        16

1012 GETS were required to define 16 macros.

There were no errors, warnings or information messages.

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

0377 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



MDAT
LIS

NLDRIVER
LIS

MDATEND
LIS

OPENMSG
LIS

MTFDT
LIS

LOADMREG
LIS

LOADSUB
LIS

MISCDEF
LIS

MBDRIVER
LIS

MEMORYALC
LIS

MUTEX
LIS

NULLPROC
LIS

LOAVEC
LIS