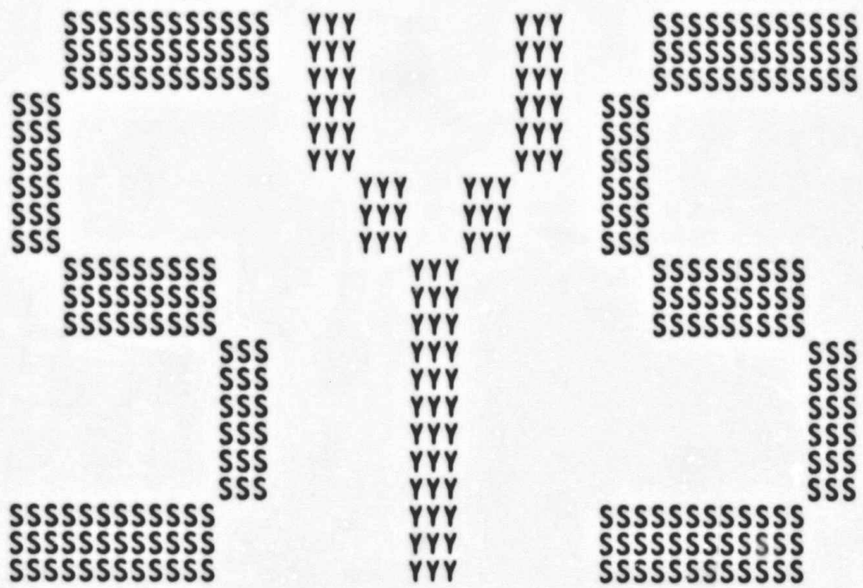
SSSSSSSSSSS YYY YYY SSSSSSSSSSS
SSSSSSSSSSS YYY YYY SSSSSSSSSSS
SSSSSSSSSSS YYY YYY SSSSSSSSSSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSS YYY YYY SSS
SSSSSSSSS YYY SSSSSSSSS
SSSSSSSSS YYY SSSSSSSSS
SSSSSSSSS YYY SSSSSSSSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSS YYY SSS
SSSSSSSSSSS YYY SSSSSSSSSSS
SSSSSSSSSSS YYY SSSSSSSSSSS
SSSSSSSSSSS YYY SSSSSSSSSSS

```
RRRRRRRR    EEEEEEEEEE  LL            OOOOOO      CCCCCCCC  DDDDDDDD    RRRRRRRR    VV      VV
RRRRRRRR    EEEEEEEEEE  LL            OOOOOO      CCCCCCCC  DDDDDDDD    RRRRRRRR    VV      VV
RR      RR  EE          LL          OO      OO  CC          DD      DD  RR      RR  VV      VV
RR      RR  EE          LL          OO      OO  CC          DD      DD  RR      RR  VV      VV
RR      RR  EE          LL          OO      OO  CC          DD      DD  RR      RR  VV      VV
RR      RR  EE          LL          OO      OO  CC          DD      DD  RR      RR  VV      VV
RRRRRRRR    EEEEEEE     LL          OO      OO  CC          DD      DD  RRRRRRRR    VV      VV
RRRRRRRR    EEEEEEE     LL          OO      OO  CC          DD      DD  RRRRRRRR    VV      VV
RR  RR      EE          LL          OO      OO  CC          DD      DD  RR  RR      VV      VV
RR  RR      EE          LL          OO      OO  CC          DD      DD  RR  RR      VV      VV
RR    RR    EE          LL          OO      OO  CC          DD      DD  RR    RR      VV  VV      ....
RR    RR    EE          LL          OO      OO  CC          DD      DD  RR    RR      VV  VV      ....
RR      RR  EEEEEEEEEE  LLLLLLLLLL    OOOOOO      CCCCCCCC  DDDDDDDD    RR      RR      VV      ....
RR      RR  EEEEEEEEEE  LLLLLLLLLL    OOOOOO      CCCCCCCC  DDDDDDDD    RR      RR      VV      ....

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

Table of contents

```
0000     1          .TITLE  RELOCDRV - RELOCATE DRIVER DATABASE
0000     2          .IDENT  'V04-000'
0000     3
0000     4  ;
0000     5  ;*****************************************************************************
0000     6  ;*                                                                           *
0000     7  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9  ;*  ALL RIGHTS RESERVED.                                                     *
0000    10  ;*                                                                           *
0000    11  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    12  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    13  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    14  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    15  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    16  ;*  TRANSFERRED.                                                             *
0000    17  ;*                                                                           *
0000    18  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    19  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    20  ;*  CORPORATION.                                                             *
0000    21  ;*                                                                           *
0000    22  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    23  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24  ;*                                                                           *
0000    25  ;*                                                                           *
0000    26  ;*****************************************************************************
0000    27  ;
0000    28  ; D. N. CUTLER 16-JUN-78
0000    29  ;
0000    30  ; RELOCATE DRIVER DATABASE
0000    31  ;
0000    32  ; MODIFICATION HISTORY:
0000    33  ;
0000    34  ;       V03-003 LMP0221         L. Mark Pilant,         26-Mar-1984  16:02
0000    35  ;               Add an ORB to the list of known structures that may be
0000    36  ;               initialized.
0000    37  ;
0000    38  ;       V03-002 ROW0289         Ralph O. Weber          26-JAN-1984
0000    39  ;               Add DDT$L_MNTV_SQD, DDT$L_MNTV_FOR, and DDT$L_MNTV_SSSC to the
0000    40  ;               list of DDT cells to be relocated.  Update size of DDT in
0000    41  ;               ASSUME statement.
0000    42  ;
0000    43  ;       V03-001 ROW0125         Ralph O. Weber          19-SEP-1982
0000    44  ;               Add DDT$L_CLONEDUCB to list of DDT cells to be relocated.
0000    45  ;               Update size of DDT in ASSUME statement.
0000    46  ;
0000    47  ;       V02-003 KTA0058         Kerbey T. Altmann       30-Dec-1981
0000    48  ;               Change name to IOC$RELOC_DDT so symbol len <= 15.
0000    49  ;
0000    50  ;       V02-002 JLV0047         Jake VanNoy             13-Jul-1981
0000    51  ;               Added IOC$RELOC_DDT_FDT, called from SYSGEN and the
0000    52  ;               INIT module in SYS to relocate DDT's and FDT's to
0000    53  ;               absolute system addreses.
0000    54  ;
0000    55  ; MACRO LIBRARY CALLS
0000    56  ;
0000    57
```

```
                     0000     58          $DDTDEF                         ;DEFINE DDT OFFSETS
                     0000     59          $CRBDEF                         ;DEFINE CRB OFFSETS
                     0000     60          $DDBDEF                         ;DEFINE DDB OFFSETS
                     0000     61          $DPTDEF                         ;DEFINE DPT OFFSETS
                     0000     62          $DYNDEF                         ;DEFINE DATA STRUCTURE TYPES
                     0000     63          $IDBDEF                         ;DEFINE IDB OFFSETS
                     0000     64          $UCBDEF                         ;DEFINE UCB OFFSETS
                     0000     65          $VECDEF                         ;DEFINE VEC OFFSETS
                     0000     66
                     0000     67  ;
                     0000     68  ; LOCAL DATA
                     0000     69  ;
                     0000     70  ; STRUCTURE TYPE BYTE ARRAY
                     0000     71  ;
                     0000     72
                00000000     73          .PSECT  Z$INIT_DRIVER
      10 09 06 05    0000     74  STYPE:  .BYTE   DYN$C_CRB,DYN$C_DDB,DYN$C_IDB,DYN$C_UCB ;
               49    0004     75          .BYTE   DYN$C_ORB                               ;
                     0005     76
                     0005     77  ASSUME  DDT$C_LENGTH EQ 56      ; So new entries in DDT will be added here
                     0005     78
                     0005     79  RELOC_DDT_TABLE: ; All these offsets in the DDT must be relocated
                     0005     80
                  00 0005     81          .BYTE   DDT$L_START
                  04 0006     82          .BYTE   DDT$L_UNSOLINT
                  08 0007     83          .BYTE   DDT$L_FDT
                  0C 0008     84          .BYTE   DDT$L_CANCEL
                  10 0009     85          .BYTE   DDT$L_REGDUMP
                  18 000A     86          .BYTE   DDT$L_UNITINIT
                  1C 000B     87          .BYTE   DDT$L_ALTSTART
                  20 000C     88          .BYTE   DDT$L_MNTVER
                  24 000D     89          .BYTE   DDT$L_CLONEDUCB
                  2C 000E     90          .BYTE   DDT$L_MNTV_SSSC
                  30 000F     91          .BYTE   DDT$L_MNTV_FOR
                  34 0010     92          .BYTE   DDT$L_MNTV_SQD
                  FF 0011     93          .BYTE   -1                      ; End of table
                     0012     94
```

```
                     0012     96          .SBTTL  INITIALIZE DRIVER DATA BASE
                     0012     97 ;+
                     0012     98 ; IOC$INITDRV - INITIALIZE DRIVER DATA BASE
                     0012     99 ;
                     0012    100 ; THIS ROUTINE IS CALLED AFTER HAVING LOADED ALL OR PART OF A DEVICE DATA BASE.
                     0012    101 ; IT MUST BE CALLED FOR EACH UNIT THAT IS DEFINED. ITS FUNCTION IS TO USE THE
                     0012    102 ; INITIALIZATION TABLE IN THE DRIVER PROLOGUE TO INITIALIZE STATIC INFORMATION
                     0012    103 ; IN THE DRIVER DATA BASE.
                     0012    104 ;
                     0012    105 ; INPUTS:
                     0012    106 ;
                     0012    107 ;       R4 = ADDRESS OF DRIVER PROLOGUE TABLE.
                     0012    108 ;       R5 = ADDRESS OF DEVICE UCB.
                     0012    109 ;
                     0012    110 ; OUTPUTS:
                     0012    111 ;
                     0012    112 ;       THE DRIVER PROLOGUE TABLE IS INTERPRETED AND THE RESPECTIVE DATA BASE
                     0012    113 ;       IS INITIALIZED.
                     0012    114 ;
                     0012    115 ;       R0 LOW BIT CLEAR INDICATES A BAD DATA STRUCTURE.
                     0012    116 ;
                     0012    117 ;       R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.
                     0012    118 ;-
                     0012    119
                     0012    120          .ENABL  LSB
                     0012    121 IOC$INITDRV::                               ;INITIALIZE DRIVER DATA BASE
 53   10 A4   3C     0012    122          MOVZWL  DPT$W_INITTAB(R4),R3       ;GET OFFSET TO INITIALIZATION TABLE
       04   11     0016    123          BRB     10$                        ;BR TO COMMON CODE
```

```
                     0018   125          .SBTTL  REINITIALIZE DRIVER DATA BASE
                     0018   126 ;+
                     0018   127 ; IOC$REINITDRV - REINITIALIZE DRIVER DATA BASE
                     0018   128 ;
                     0018   129 ; THIS ROUTINE IS CALLED AFTER HAVING LOADED A DRIVER WHOSE DEVICE DATA BASE IS
                     0018   130 ; ALREADY RESIDENT IN MEMORY. IT MUST BE CALLED FOR EACH UNIT THAT IS DEFINED.
                     0018   131 ; ITS FUNCTION IS TO INITIALIZE ONLY THOSE FIELDS IN THE DATA BASE THAT ARE NOT
                     0018   132 ; DYNAMIC.
                     0018   133 ;
                     0018   134 ; INPUTS:
                     0018   135 ;
                     0018   136 ;       R4 = ADDRESS OF DRIVER PROLOGUE TABLE.
                     0018   137 ;       R5 = ADDRESS OF DEVICE UCB.
                     0018   138 ;
                     0018   139 ; OUTPUTS:
                     0018   140 ;
                     0018   141 ;       THE DRIVER PROLOGUE TABLE IS INTERPRETED AND THE RESPECTIVE DATA BASE
                     0018   142 ;       IS REINITIALIZED.
                     0018   143 ;
                     0018   144 ;       R0 LOW BIT CLEAR INDICATES A BAD DATA STRUCTURE.
                     0018   145 ;
                     0018   146 ;       R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.
                     0018   147 ;-
                     0018   148
                     0018   149 IOC$REINITDRV::                              ;REINITIALIZE DRIVER DATA BASE
 53    12 A4    3C   0018   150          MOVZWL  DPT$W_REINITTAB(R4),R3      ;GET OFFSET TO REINITIALIZATION TABLE
    53    54    C0   001C   151 10$:     ADDL    R4,R3                       ;CALCULATE ADDRESS OF INITIALIZE TABLE
          5F    10   001F   152 20$:     BSBB    LOCATE                      ;LOCATE STRUCTURE TYPE
    51    83    9A   0021   153          MOVZBL  (R3)+,R1                    ;GET OFFSET IN STRUCTURE
    52    51    C0   0024   154          ADDL    R1,R2                       ;CALCULATE BASE ADDRESS OF FIELD
    51    83    9A   0027   155          MOVZBL  (R3)+,R1                    ;GET OPERATION CODE
          02    10   002A   156          BSBB    EXECUTE                     ;EXECUTE OPERATION CODE
          F1    11   002C   157          BRB     20$                         ;
                     002E   158          .DSABL  LSB
```

```
                     002E    160          .SBTTL  LOCAL SUBROUTINE TO EXECUTE OPERATION CODE
                     002E    161 ;
                     002E    162 ; EXECUTE - EXECUTE OPERATION CODE
                     002E    163 ;
                     002E    164 ; THIS ROUTINE IS CALLED TO EXECUTE A SINGLE RELOCATION OPERATION CODE.
                     002E    165 ;
                     002E    166 ; INPUTS:
                     002E    167 ;
                     002E    168 ;       R1 = OPERATION CODE.
                     002E    169 ;       R2 = STRUCTURE FIELD BASE ADDRESS.
                     002E    170 ;       R3 = ADDRESS OF NEXT RELOCATION BYTE.
                     002E    171 ;       R4 = BASE ADDRESS OF DRIVER PROLOGUE TABLE.
                     002E    172 ;       R5 = ADDRESS OF DEVICE UCB.
                     002E    173 ;
                     002E    174 ; OUTPUTS:
                     002E    175 ;
                     002E    176 ;       IF AN INVALID OPERATION CODE IS SPECIFIED, A FAILURE IS RETURNED TO THE
                     002E    177 ;       CALLERS CALLER. ELSE THE OPERAND VALUE IS STORE IN THE SPECIFIED FIELD
                     002E    178 ;       IN THE SPECIFIED STRUCTURE.
                     002E    179 ;
                     002E    180
                     002E    181 EXECUTE:
                     002E    182         CASE    R1,<-                           ;EXECUTE RELOCATION OPERATION
                     002E    183                 10$,-                           ;DISPATCH TO FETCH ROUTINE
                     002E    184                 30$,-                           ;BYTE IMMEDIATE
                     002E    185                 50$,-                           ;WORD IMMEDIATE
                     002E    186                 70$,-                           ;WORD DISPLACED
                     002E    187                 90$,-                           ;LONGWORD IMMEDIATE
                     002E    188                 >                               ;FIELD IMMEDIATE
                     003C    189
                     003C    190 ;
                     003C    191 ; INDIRECT OPERAND
                     003C    192 ;
                     003C    193
 50    93    D0      003C    194         MOVL    @(R3)+,R0                       ;GET DATA LONGWORD
                     003F    195         CASE    R1,<-                           ;DISPATCH TO STORE ROUTINE
                     003F    196                 20$,-                           ;STORE BYTE
                     003F    197                 40$,-                           ;STORE WORD
                     003F    198                 60$,-                           ;STORE WORD DISPLACED
                     003F    199                 80$,-                           ;STORE LONGWORD
                     003F    200                 100$,-                          ;STORE FILED
                     003F    201                 >,LIMIT=#^X80                   ;
      8E    D5      004F    202         TSTL    (SP)+                           ;REMOVE RETURN FROM STACK
      50    D4      0051    203         CLRL    R0                              ;SET FAILURE INDICATION
            05      0053    204         RSB                                     ;
                     0054    205
                     0054    206 ;
                     0054    207 ; BYTE IMMEDIATE
                     0054    208 ;
                     0054    209
 50    83    9A      0054    210 10$:    MOVZBL  (R3)+,R0                        ;GET DATA BYTE
 62    50    90      0057    211 20$:    MOVB    R0,(R2)                         ;STORE BYTE
            05      005A    212         RSB                                     ;
                     005B    213
                     005B    214 ;
                     005B    215 ; WORD IMMEDIATE
                     005B    216 ;
```

```
                        005B   217
                50 83   3C   005B   218 30$:    MOVZWL  (R3)+,R0                ;GET DATA WORD
                62 50   B0   005E   219 40$:    MOVW    R0,(R2)                 ;STORE WORD
                        05   0061   220         RSB                             ;
                             0062   221
                             0062   222 ;
                             0062   223 ; WORD DISPLACED
                             0062   224 ;
                             0062   225
                50 83   32   0062   226 50$:    CVTWL   (R3)+,R0                ;GET WORD DISPLACEMENT
                50 54   C0   0065   227 60$:    ADDL    R4,R0                   ;CALCULATE ACTUAL ADDRESS
                        03   11   0068   228         BRB     80$                     ;
                             006A   229
                             006A   230 ;
                             006A   231 ; LONGWORD IMMEDIATE
                             006A   232 ;
                             006A   233
                50 83   D0   006A   234 70$:    MOVL    (R3)+,R0                ;GET DATA LONGWORD
                62 50   D0   006D   235 80$:    MOVL    R0,(R2)                 ;STORE DATA LONGWORD
                        05   0070   236         RSB                             ;
                             0071   237
                             0071   238 ;
                             0071   239 ; FIELD STRUCTURE
                             0071   240 ;
                             0071   241
                50 83   D0   0071   242 90$:    MOVL    (R3)+,R0                ;GET DATA LONGWORD
                7E 83   9A   0074   243 100$:   MOVZBL  (R3)+,-(SP)             ;GET STARTING BIT NUMBER
                51 83   9A   0077   244         MOVZBL  (R3)+,R1                ;GET SIZE OF FIELD
          62 51 8E 50   F0   007A   245         INSV    R0,(SP)+,R1,(R2)        ;STORE FIELD
                        05   007F   246         RSB                             ;
```

```
                     0080   248          .SBTTL  LOCAL SUBROUTINE TO LOCATE DATA STRUCTURE TYPE
                     0080   249 ;
                     0080   250 ; LOCATE - LOCATE DATA STRUCTURE TYPE
                     0080   251 ;
                     0080   252 ; INPUTS:
                     0080   253 ;
                     0080   254 ;       R5 = ADDRESS OF DEVICE UCB.
                     0080   255 ;
                     0080   256 ; OUTPUTS:
                     0080   257 ;
                     0080   258 ;       IF THE STRUCTURE TYPE CODE IS ZERO, THEN SUCCESS IS RETURNED TO THE CALLERS
                     0080   259 ;       CALLER. IF THE STRUCTURE TYPE CODE IS INVALID, THEN AN ERROR IS RETURNED
                     0080   260 ;       TO THE CALLERS CALLER. ELSE THE APPROPRIATE STRUCTURE BASE ADDRESS IS
                     0080   261 ;       RETURNED IN R2.
                     0080   262 ;
                     0080   263
                     0080   264 LOCATE:                                      ;LOCATE DATA STRUCTURE TYPE
          50  01 88 0080   265          BISB    #1,R0                       ;ASSUME END OF RELOCATION TABLE
              63 95 0083   266          TSTB    (R3)                        ;END OF TABLE?
              14 13 0085   267          BEQL    10$                         ;IF EQL YES
    FF73 CF   05 83 3A 0087   268          LOCC    (R3)+,#5,STYPE              ;LOCATE STRUCTURE CODE
                     008D   269          CASE    R0,<-                       ;DISPATCH TO STRUCTURE ROUTINE
                     008D   270                  60$,-                       ;ORB
                     008D   271                  20$,-                       ;UCB
                     008D   272                  30$,-                       ;IDB
                     008D   273                  40$,-                       ;DDB
                     008D   274                  50$,-                       ;CRB
                     008D   275                  >,LIMIT=#1                  ;
          8E  D5 009B   276 10$:    TSTL    (SP)+                       ;REMOVE RETURN FROM STACK
              05 009D   277          RSB                                 ;
                     009E   278
                     009E   279 ;
                     009E   280 ; UCB STRUCTURE
                     009E   281 ;
                     009E   282
      52  55  D0 009E   283 20$:    MOVL    R5,R2                       ;SET UCB ADDRESS
              05 00A1   284          RSB                                 ;
                     00A2   285
                     00A2   286 ;
                     00A2   287 ; IDB STRUCTURE
                     00A2   288 ;
                     00A2   289
  52  24 A5  D0 00A2   290 30$:    MOVL    UCB$L_CRB(R5),R2            ;GET ADDRESS OF CRB
  52  2C A2  D0 00A6   291          MOVL    CRB$L_INTD+VEC$L_IDB(R2),R2 ;GET ADDRESS OF IDB
              05 00AA   292          RSB                                 ;
                     00AB   293
                     00AB   294 ;
                     00AB   295 ; DDB STRUCTURE
                     00AB   296 ;
                     00AB   297
  52  28 A5  D0 00AB   298 40$:    MOVL    UCB$L_DDB(R5),R2            ;GET ADDRESS OF DDB
              05 00AF   299          RSB                                 ;
                     00B0   300
                     00B0   301 ;
                     00B0   302 ; CRB STRUCTURE
                     00B0   303 ;
                     00B0   304
```

```
                52   24 A5   D0   00B0   305 50$:    MOVL    UCB$L_CRB(R5),R2        ;GET ADDRESS OF CRB
                             05   00B4   306         RSB                             ;
                                  00B5   307
                                  00B5   308 ;
                                  00B5   309 ; ORB STRUCTURE
                                  00B5   310 ;
                                  00B5   311
                52   1C A5   D0   00B5   312 60$:    MOVL    UCB$L_ORB(R5),R2;       ;GET ADDRESS OF ORB
                             05   00B9   313         RSB                             ;
```

```
                     00BA   315 .SBTTL  IOC$RELOC_DDT - relocate DDT and FDT to system addresses
                     00BA   316
                     00BA   317 ;+
                     00BA   318 ;
                     00BA   319 ; FUNCTIONAL DESCRIPTION:
                     00BA   320 ;
                     00BA   321 ;       This routine will relocate any driver-relative addresses to be
                     00BA   322 ;       absolute system addresses.
                     00BA   323 ;
                     00BA   324 ; CALLING SEQUENCE:
                     00BA   325 ;
                     00BA   326 ;       JSB     IOC$RELOC_DDT
                     00BA   327 ;
                     00BA   328 ; INPUTS:
                     00BA   329 ;
                     00BA   330 ;       R11 - Address of DDB
                     00BA   331 ;
                     00BA   332 ; OUTPUTS:
                     00BA   333 ;
                     00BA   334 ;       R4,R5,R6 are destroyed, DDT and FDT are relocated.
                     00BA   335 ;
                     00BA   336 ; COMPLETION CODES:
                     00BA   337 ;
                     00BA   338 ;       None.
                     00BA   339 ;
                     00BA   340 ;-
                     00BA   341
                     00BA   342 IOC$RELOC_DDT::
                     00BA   343
                     00BA   344 ;
                     00BA   345 ; Relocate the driver dispatch table (DDT)
                     00BA   346 ;
                     00BA   347
 55   FF47 CF   9E   00BA   348         MOVAB   RELOC_DDT_TABLE,R5      ; Address of offsets table
   54   0C AB   D0   00BF   349         MOVL    DDB$L_DDT(R11),R4       ; Addess of DDT
                     00C3   350
      56   85   98   00C3   351 10$:    CVTBL   (R5)+,R6                ; Get next offset
           0C   19   00C6   352         BLSS    20$                     ; Branch if end of table
      56   54   C0   00C8   353         ADDL2   R4,R6                   ; Add start address of driver to offset
 F4   66   1F   E0   00CB   354         BBS     #31,(R6),10$            ; Branch if already system address
      66   54   C0   00CF   355         ADDL2   R4,(R6)                 ; Add in driver base
           EF   11   00D2   356         BRB     10$                     ; Loop
                     00D4   357
                     00D4   358 ;
                     00D4   359 ; Relocate the function dispatch table (FDT)
                     00D4   360 ;
                     00D4   361
 55   28 A4   3C   00D4   362 20$:    MOVZWL  DDT$W_FDTSIZE(R4),R5    ; Size of FDT
 54   08 A4   D0   00D8   363         MOVL    DDT$L_FDT(R4),R4        ; Address of FDT
      55   54   C0   00DC   364         ADDL2   R4,R5                   ; Add in base of FDT to get end
      54   04   C0   00DF   365         ADDL2   #4,R4                   ; Initial offset past legal and
                     00E2   366                                         ;  buffered I/O masks
                     00E2   367
      54   0C   C0   00E2   368 30$:    ADDL2   #12,R4                  ; Get next dispatch block (12 bytes)
      54   55   D1   00E5   369         CMPL    R5,R4                   ; Past end ?
           0B   15   00E8   370         BLEQ    40$                     ; branch if yes
F3 08 A4   1F   E0   00EA   371         BBS     #31,8(R4),30$           ; Branch if already system address
```

```
08 A4   54   C0  00EF   372          ADDL2   R4,8(R4)              ; Add in current base
        ED   11  00F3   373          BRB     30$
                 00F5   374
             05  00F5   375 40$:     RSB
                 00F6   376
                 00F6   377          .END
```

```
CRB$L_INTD                      = 00000024
DDB$L_DDT                       = 0000000C
DDT$C_LENGTH                    = 00000038
DDT$L_ALTSTART                  = 0000001C
DDT$L_CANCEL                    = 0000000C
DDT$L_CLONEDUCB                 = 00000024
DDT$L_FDT                       = 00000008
DDT$L_MNTVER                    = 00000020
DDT$L_MNTV_FOR                  = 00000030
DDT$L_MNTV_SQD                  = 00000034
DDT$L_MNTV_SSSC                 = 0000002C
DDT$L_REGDUMP                   = 00000010
DDT$L_START                     = 00000000
DDT$L_UNITINIT                  = 00000018
DDT$L_UNSOLINT                  = 00000004
DDT$W_FDTSIZE                   = 00000028
DPT$W_INITTAB                   = 00000010
DPT$W_REINITTAB                 = 00000012
DYN$C_CRB                       = 00000005
DYN$C_DDB                       = 00000006
DYN$C_IDB                       = 00000009
DYN$C_ORB                       = 00000049
DYN$C_UCB                       = 00000010
EXECUTE                           0000002E R      02
IOC$INITDRV                       00000012 RG     02
IOC$REINITDRV                     00000018 RG     02
IOC$RELOC_DDT                     000000BA RG     02
LOCATE                            00000080 R      02
RELOC_DDT_TABLE                   00000005 R      02
STYPE                             00000000 R      02
UCB$L_CRB                       = 00000024
UCB$L_DDB                       = 00000028
UCB$L_ORB                       = 0000001C
VEC$L_IDB                       = 00000008
```

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
Z$INIT_DRIVER	000000F6 (246.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.10	00:00:02.52
Command processing	124	00:00:00.55	00:00:03.04
Pass 1	250	00:00:06.53	00:00:22.52
Symbol table sort	0	00:00:00.93	00:00:03.57
Pass 2	78	00:00:01.42	00:00:05.05
Symbol table output	5	00:00:00.06	00:00:00.39

```
Psect synopsis output                1     00:00:00.02    00:00:00.02
Cross-reference output               0     00:00:00.00    00:00:00.00
Assembler run totals               495     00:00:09.61    00:00:37.33

The working set limit was 1350 pages.
36935 bytes (73 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 644 non-local and 28 local symbols.
377 source lines were read in Pass 1, producing 13 object records in Pass 2.
17 pages of virtual memory were used to define 16 macros.
```

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

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	9
_255DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	13

```
738 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

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

0379 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

RELOCDRV
LIS

RMSRESET
LIS

PROCSTRT
LIS

PHDUTL
LIS

PMSDAT
LIS

POSTEF
LIS

POWERFAIL
LIS

PRDEF
LIS

PTEDUMP
LIS

PDAT
LIS

RMSVECTOR
LIS