```
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS  LLL                 OOOOOOOOO        AAAAAAAAA
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS  LLL                 OOOOOOOOO        AAAAAAAAA
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS  LLL                 OOOOOOOOO        AAAAAAAAA
SSS              YYY         YYY  SSS              LLL              OOO         OOO  AAA         AAA
SSS              YYY         YYY  SSS              LLL              OOO         OOO  AAA         AAA
SSS              YYY         YYY  SSS              LLL              OOO         OOO  AAA         AAA
SSS                 YYY   YYY     SSS              LLL              OOO         OOO  AAA         AAA
SSS                 YYY   YYY     SSS              LLL              OOO         OOO  AAA         AAA
SSS                 YYY   YYY     SSS              LLL              OOO         OOO  AAA         AAA
   SSSSSSSSS           YYY           SSSSSSSSS     LLL              OOO         OOO  AAA         AAA
   SSSSSSSSS           YYY           SSSSSSSSS     LLL              OOO         OOO  AAA         AAA
   SSSSSSSSS           YYY           SSSSSSSSS     LLL              OOO         OOO  AAA         AAA
            SSS        YYY                    SSS  LLL              OOO         OOO  AAAAAAAAAAAAAAA
            SSS        YYY                    SSS  LLL              OOO         OOO  AAAAAAAAAAAAAAA
            SSS        YYY                    SSS  LLL              OOO         OOO  AAAAAAAAAAAAAAA
            SSS        YYY                    SSS  LLL              OOO         OOO  AAA         AAA
            SSS        YYY                    SSS  LLL              OOO         OOO  AAA         AAA
            SSS        YYY                    SSS  LLL              OOO         OOO  AAA         AAA
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS     LLLLLLLLLLLLLLL     OOOOOOOOO     AAA         AAA
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS     LLLLLLLLLLLLLLL     OOOOOOOOO     AAA         AAA
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS     LLLLLLLLLLLLLLL     OOOOOOOOO     AAA         AAA
```

```
EEEEEEEEEE  RRRRRRRR    RRRRRRRR      SSSSSSSS  UU      UU  BBBBBBBB    77777777    5555555555    000000
EEEEEEEEEE  RRRRRRRR    RRRRRRRR      SSSSSSSS  UU      UU  BBBBBBBB    77777777    5555555555    000000
EE          RR      RR  RR      RR  SS          UU      UU  BB      BB          77  55            00      00
EE          RR      RR  RR      RR  SS          UU      UU  BB      BB          77  55            00      00
EE          RR      RR  RR      RR  SS          UU      UU  BB      BB          77  555555        00    0000
EE          RR      RR  RR      RR  SS          UU      UU  BB      BB          77  555555        00    0000
EEEEEEEE    RRRRRRRR    RRRRRRRR      SSSSSS    UU      UU  BBBBBBBB        77          55    00  00  00
EEEEEEEE    RRRRRRRR    RRRRRRRR      SSSSSS    UU      UU  BBBBBBBB        77          55    00  00  00
EE          RR  RR      RR  RR              SS  UU      UU  BB      BB    77              55  0000    00
EE          RR  RR      RR  RR              SS  UU      UU  BB      BB    77              55  0000    00
EE          RR    RR    RR    RR            SS  UU      UU  BB      BB  77          55      55  00      00      ....
EE          RR    RR    RR    RR            SS  UU      UU  BB      BB  77          55      55  00      00      ....
EEEEEEEEEE  RR      RR  RR      RR  SSSSSSSS    UUUUUUUUUU  BBBBBBBB  77            555555      000000        ....
EEEEEEEEEE  RR      RR  RR      RR  SSSSSSSS    UUUUUUUUUU  BBBBBBBB  77            555555      000000        ....

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

```
0000     1          .NOSHOW CONDITIONALS
0000     5
0000     7          .TITLE  ERRSUB750 - ERROR SUBROUTINES FOR VAX 11/750
0000     9
0000    13
0000    17
0000    21
0000    22          .IDENT  'V04-002'
0000    23
0000    24 ;
0000    25 ;*****************************************************************************
0000    26 ;*                                                                           *
0000    27 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    28 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000    29 ;*  ALL RIGHTS RESERVED.                                                     *
0000    30 ;*                                                                           *
0000    31 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    32 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    33 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    34 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    35 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    36 ;*  TRANSFERRED.                                                             *
0000    37 ;*                                                                           *
0000    38 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    39 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    40 ;*  CORPORATION.                                                             *
0000    41 ;*                                                                           *
0000    42 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    43 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    44 ;*                                                                           *
0000    45 ;*                                                                           *
0000    46 ;*****************************************************************************
0000    47 ;
0000    48 ;++
0000    49 ;
0000    50 ; FACILITY:
0000    51 ;
0000    52 ;       EXECUTIVE,  LOADABLE SUBROUTINES USED BY POWERFAIL AND BUGCHECK.
0000    53 ;
0000    54 ; ABSTRACT:
0000    55 ;
0000    56 ; LOADABLE SUBROUTINES USED BY POWERFAIL AND BUGCHECK.
0000    57 ;
0000    58 ; AUTHOR:
0000    59 ;
0000    60 ;       N. KRONENBERG, JULY 2, 1979.
0000    61 ;
0000    62 ; MODIFIED BY:
0000    63 ;
0000    64 ;       V04-003 WMC00001        Wayne Cardoza           13-Sep-1984
0000    65 ;               CRD reporting must not be turned off for VENUS.
0000    66 ;
0000    67 ;       V04-002 CWH4002         CW Hobbs                08-Sep-1984
0000    68 ;               Correct typo in TCM0010, use "-" instead of "="
0000    69 ;
0000    70 ;       V04-001 TCM0010         Trudy C. Matthews       07-Sep-1984
0000    71 ;               For the venus processor: move turning on cache from routine
```

```
                                ;
                                ;               EXE$INIPROCREG to a new routine: INI$CACHE.  Correct the
                                ;               order in which registers are saved on the stack in EXE$REGSAVE.
                                ;
                                ;       V03-022 TCM0009         Trudy C. Matthews       30-Jul-1984
                                ;               When turning off CRD interrupts in EXE$INIPROCREG for VENUS,
                                ;               read the processor register and write it back to preserve
                                ;               the state of other bits in the register.
                                ;
                                ;       V03-021 TCM0008         Trudy C. Matthews       23-Jul-1984
                                ;               Remove venus code that queries the console for how to set up
                                ;               cache and FBOX state.  Instead always turn the cache and
                                ;               FBOX on (and let the normal error handling code turn it off
                                ;               if its bad).
                                ;
                                ;       V03-020 DWT0214         David W. Thiel          02-May-1984
                                ;               Revise MicroVAX I TODR register simulation.
                                ;
                                ;       V03-019 KDM0096         Kathleen D. Morse       27-Mar-1984
                                ;               Add missing indirection in MicroVAX I memory CSR
                                ;               CRD enabling.
                                ;
                                ;       V03-018 KPL0101         Peter Lieberwirth       4-Mar-1984
                                ;               Add extra vectors now defined in SYSLOAVEC.  These vectors
                                ;               are insurance for v4.x
                                ;
                                ;       V03-017 KPL0100         Peter Lieberwirth       12-Feb-1984
                                ;               Change RPB$B_BOOTNDT to RPB$W_BOOTNDT, since BI devices
                                ;               will have 16-bit device types.
                                ;
                                ;       V03-016 KDM0092         Kathleen D. Morse       23-Jan-1984
                                ;               Correct the number of cpu-specific IPRs logged for the
                                ;               11/730 and MicroVAX I cpus.
                                ;
                                ;       V03-015 CWH8001         CW Hobbs                5-Dec-1983
                                ;               Add entry points for EXE$READP_TODR and EXE$WRITEP_TODR
                                ;               to access physical TODR register for Nautilus CPU.  For
                                ;               other processors, these amount to duplicate labels on
                                ;               EXE$READ_TODR and EXE$WRITE_TODR.
                                ;
                                ;       V03-014 KTA3088         Kerbey T. Altmann       17-Oct-1983
                                ;               Fix bug in 730 conditional for EXE$INIBOOTADP.
                                ;
                                ;       V03-013 KDM0081         Kathleen D. Morse       13-Sep-1983
                                ;               Create Micro-VAX I version.
                                ;
                                ;       V03-012 KDM0055         Kathleen D. Morse       12-Jul-1983
                                ;               Move IPR PME into the cpu-dependent register save and
                                ;               restore routines.
                                ;
                                ;       V03-011 KDM0049         Kathleen D. Morse       07-Jul-1983
                                ;               Add the following processor registers to the cpu-specific
                                ;               dump IPRs routine:  ICR, TODR, ACCS.  Add usage of
                                ;               register:  EXE$READ_TODR and EXE$WRITE_TODR.
                                ;
                                ;       V03-010 KDM0048         Kathleen D. Morse       07-Jul-1983
                                ;               Add loadable routines for referencing the time-of-day
                                ;               clock:  EXE$READ_TODR, EXE$WRITE_TODR.
```

```
0000   129 ;
0000   130 ;       V03-009 TCM0007         Trudy C. Matthews       02-Jun-1983
0000   131 ;               Fix routine SYSL$CLRSBIA so that it calculates the address
0000   132 ;               of SBI adapter register space correctly.
0000   133 ;
0000   134 ;       V03-008 TCM0006         Trudy C. Matthews       9-Feb-1983
0000   135 ;               Store enable/disable state of 11/790 cache and FBOX in
0000   136 ;               EXE$GB_CPUDATA cell during system initialization.
0000   137 ;
0000   138 ;       V03-007 TCM0005         Trudy C. Matthews       11-Jan-1983
0000   139 ;               Add routine SYSL$CLRSBIA.  Add SBIA register initialization
0000   140 ;               to EXE$INIPROCREG.  Add 11/790 machine check handler to
0000   141 ;               EXE$TEST_CSR.  Change 11/780 machine check handler to
0000   142 ;               write PR$_SBIFS back to itself to clear error bits.
0000   143 ;               Add labels for two "extra" routines, that can be patched
0000   144 ;               if extra vectors from SYS to SYSLOA are needed in between
0000   145 ;               major releases.  Make EXE$DUMPCPUREG log the SBI registers
0000   146 ;               from the SBI the 11/790 system disk is on.
0000   147 ;
0000   148 ;       V03-006 TCM0004         Trudy C. Matthews       3-Jan-1983
0000   149 ;               Add more 11/790-specific code.
0000   150 ;
0000   151 ;       V03-005 TCM0003         Trudy C. Matthews       17-Dec-1982
0000   152 ;               Add conditional assembly switch to the invocations
0000   153 ;               of 11/790-specific definition macros.
0000   154 ;
0000   155 ;       V03-004 TCM0002         Trudy C. Matthews       15-Dec-1982
0000   156 ;               Added 11/790-specific code to EXE$INIPROCREG.
0000   157 ;
0000   158 ;       V03-003 TCM0001         Trudy C. Matthews       13-Dec-1982
0000   159 ;               Added 11/790-specific code to power down/power up
0000   160 ;               routines.
0000   161 ;
0000   162 ;       V03-002 KTA3018         Kerbey T. Altmann       30-Oct-1982
0000   163 ;               Remove CI and UBA routines to another module.
0000   164 ;
0000   165 ;--
```

```
                     0000   167
                     0000   168 ;
                     0000   169 ; MACRO L'BRARY CALLS:
                     0000   170 ;
                     0000   171
                     0000   172          $ADPDEF                         ;DEFINE ADAPTER OFFSETS
                     0000   173          $BQODEF                         ;DEFINE BOOT QIO OFFSETS
                     0000   174          $BTDDEF                         ;DEFINE BOOT DEVICE TYPES
                     0000   175          $EMBCRDEF                       ;DEFINE ERROR MSG BUFFER OFFSETS
                     0000   176          $IDBDEF                         ;DEFINE INTERRUPT DISPACH OFFSETS
                     0000   177          $IPLDEF                         ;DEFINE INTERRUPT PRIORITY LEVELS
                     0000   178          $MBADEF                         ;DEFINE MASSBUS ADAPTER OFFSETS
                     0000   179          $NDTDEF                         ;DEFINE NEXUS DEVICE TYPES
                     0000   180          $PRDEF                          ;DEFINE INTERNAL PROCESSOR REGISTERS
                     0000   181          $RPBDEF                         ;DEFINE RESTART PARAM BLOCK OFFSETS
                     0000   182          $SSDEF                          ;DEFINE SYSTEM STATUS CODES
                     0000   183          $UBADEF                         ;DEFINE UNIBUS ADAPTER OFFSETS
                     0000   195
                     0000   199
                     0000   201          $PR750DEF                       ;DEFINE 11/750 INTERNAL PROCESSOR REGS
                     0000   203
                     0000   207
                     0000   211 ;
                     0000   212 ; EQUATED SYMBOLS:
                     0000   213 ;
                     0000   218
           00000000  0000   220          C780_LIKE = 0
           00000001  0000   221          C750_LIKE = 1
                     0000   223
                     0000   228
                     0000   233
                     0000   238
                     0000   239 ;
                     0000   240 ; Define labels for two "extra" routines.  This reserves some vectors from
                     0000   241 ; SYS.EXE into SYSLOAxxx.EXE that can be patched if another routine must
                     0000   242 ; be added in between major releases.
                     0000   243 ;
                     0000   244 EXE$EXTRA1::                            ; aligned
                     0000   245 EXE$EXTRA2::                            ;  aligned
                     0000   246 EXE$EXTRA3::                            ;   aligned
                     0000   247 EXE$EXTRA4::                            ;    aligned
                     0000   248 EXE$EXTRA5::                            ;     aligned
                     0000   249 EXE$EXTRA6::                            ; packed
                     0000   250 EXE$EXTRA7::                            ;  packed
                     0000   251 EXE$EXTRA8::                            ;   packed
                     0000   252 EXE$EXTRA9::                            ;    packed
                     0000   253 EXE$EXTRA10::                           ;     packed (think this is enough?)
                     0000   254
                  00 0000   255          HALT                            ; Error if these labels are used.
```

```
                                0001   257          .SBTTL  EXE$INIBOOTADP - INITIALIZE THE BOOT DEVICE ADAPTER
                                0001   258 ;+
                                0001   259 ; EXE$INIBOOTADP - GET THE SYSTEM BOOT DEVICE ADAPTER AND INIT IT.
                                0001   260 ;       THIS ROUTINE IS CALLED FROM BUGCHECK BEFORE THE BOOTDRIVER IS CALLED.
                                0001   261 ;
                                0001   262 ; INPUTS:
                                0001   263 ;
                                0001   264 ;       R6 = RPB ADDRESS
                                0001   265 ;
                                0001   266 ; OUTPUTS:
                                0001   267 ;
                                0001   268 ;       R0-R2 DESTROYED
                                0001   269 ;       OTHER REGISTERS PRESERVED
                                0001   270 ;-
                                0001   271
                           00000000    272          .PSECT  SYSLOA,LONG
                                0000   273          .ENABLE LSB
                                0000   274
                                0000   275 EXE$INIBOOTADP::                          ;SUBROUTINE ENTRY
                                0000   277
                      66 A6     91 0000   278          CMPB    RPB$B_DEVTYP(R6),-      ;IS BOOT DEVICE THE CONSOLE
                      40 8F        0003   279                  #BTD$K_CONSOLE          ;BLOCK STORAGE DEVICE?
                         58     13 0005   280          BEQL    40$                     ;YES, RETURN
                   50    60 A6     D0 0007   281          MOVL    RPB$L_ADPVIR(R6),R0     ;GET ADDR OF ADAPTER REG SPACE
                                000B   282
                                000B   284
               52  00A1 C6     03     AB 000B   285          BICW3   #3,RPB$W_BOOTNDT(R6),R2 ;GET GENERIC ADAPTER TYPE
                         38    52     B1 0011   286          CMPW    R2,#NDT$_CI             ;CI ADAPTER?
                               21     13 0014   287          BEQL    20$                     ;YES, RETURN
                         20    52     B1 0016   288          CMPW    R2,#NDT$_MB             ;MASS BUS ADAPTER?
                               1D     12 0019   289          BNEQ    INI_UBADP               ;BRANCH IF NOT
                               02     D0 001B   290          MOVL    #MBA$M_CR_ABORT,-       ;ABORT ACTIVE TRANSFER
                            04 A0        001D   291                  MBA$L_CR(R0)            ;
                                001F   292
                                001F   296
                      51    1B     DB 001F   298          MFPR    #PR750$_TODR,R1         ;GET CURRENT TIME (10 MS UNITS)
                                0022   300
                   51    64 A1     9E 0022   304          MOVAB   100(R1),R1              ;ALLOW ONE SECOND
                         08 A0     D5 0026   305 10$:     TSTL    MBA$L_SR(R0)            ;WAIT UNTIL TRANSFER
                               08     18 0029   306          BGEQ    15$                     ; IS COMPLETE
                                002B   307
                                002B   311
                      52    1B     DB 002B   313          MFPR    #PR750$_TODR,R2         ;GET CURRENT TIME
                                002E   315
                                002E   319
                      52    51     D1 002E   320          CMPL    R1,R2                   ;CHECK FOR INTERVAL EXPIRED
                               F3     1A 0031   321          BGTRU   10$                     ;NOT YET, WAIT SOME MORE
                               01     D0 0033   322 15$:     MOVL    #MBA$M_CR_INIT,-        ;NOW INIT MBA
                            04 A0        0035   323                  MBA$L_CR(R0)            ;
                               05     05 0037   324 20$:     RSB                             ;DONE
                                0038   325
                                0038   327
                                0038   331
                                0038   332 INI_UBADP:                                 ;INIT UBA
                                0038   333
                                0038   341
                                0038   343
```

```
                37   00   DA  0038   345          MTPR    #0,#PR750$_UBRESET      ;INIT UBI AND UNIBUS
                                0038   347
                                0038   351
                                0038   356
                                0038   358
                                0038   360 ;
                                0038   361 ; CHECK THE VMB VERSION NUMBER.  IF IT EXISTS AND IF IT IS 7 OR GREATER, THEN
                                0038   362 ; SEE IF ANY UNIBUS MAP REGISTERS TO DISABLE.
                                0038   363 ;
                                0038   364
              52    34   A6   D0  003B   365          MOVL    RPB$L_IOVEC(R6),R2      ;PICK UP THE IOVECTOR FROM RPB
              51    10   A2   B2  003F   366          MCOMW   BQO$W_VERSION(R2),R1    ;GET VMB VERSION NUMBER 1'S COMPLEMENTED
              12 A2      51   B1  0043   367          CMPW    R1,BQO$W_VERSION+2(R2)  ;CHECK AGAINST CHECK WORD IN VMB
                         16   12  0047   368          BNEQ    40$                     ;IF NOT, ASSUME NO VERSION NUMBER
              07    10   A2   B1  0049   369          CMPW    BQO$W_VERSION(R2),#7    ;VERSION 7 OR GREATER OF VMB?
                         10   1F  004D   370          BLSSU   40$                     ;NO, DON'T BOTH WITH UMR'S
              52    24   A2   D0  004F   371          MOVL    BQO$L_UMR_DIS(R2),R2    ;GRAB THE NUMBER OF UMR'S TO DISABLE
                         0A   13  0053   372          BEQL    40$                     ;NONE, LEAVE
                                0055   373
                                0055   377
                                0055   378 ;
                                0055   379 ; THIS CODE IS EXECUTED FOR ALL PROCESSORS.  ITS DISABLES ANY UNIBUS MAP
                                0055   380 ; REGISTERS ASSOCIATED WITH UNIBUS MEMORY TO PREVENT CONTENTION BETWEEN
                                0055   381 ; SBI AND UNIBUS ADDRESSES.
                                0055   382 ;
                                0055   383
            51    0800   C0   DE  0055   384          MOVAL   UBA$L_MAP(R0),R1        ;ADDRESS OF FIRST REGISTER
                         81   D4  005A   385 30$:      CLRL    (R1)+                   ;DISABLE IT
                           FB 52   F5  005C   386          SOBGTR  R2,30$              ;LOOP UNTIL ALL DONE
                                05   005F   388 40$:      RSB                         ;DONE WITH UBA INIT
                                0060   389          .DISABLE LSB
```

```
                     0060   391          .SBTTL  EXE$SHUTDWNADP - SHUTDOWN ANY ADAPTERS DURING BUGCHECK
                     0060   392          .SBTTL  EXE$STARTUPADP - STARTUP ANY ADAPTERS
                     0060   393 ;+
                     0060   394 ; EXE$SHUTDWNADP - SHUTDOWN ANY ADAPTERS DURING BUGCHECK
                     0060   395 ;       THIS ROUTINE IS CALLED FROM BUGCHECK BEFORE THE DUMP IS TAKEN TO
                     0060   396 ;       ENSURE THAT ALL ADAPTERS THAT NEED TO BE QUIESENT ARE.
                     0060   397 ;
                     0060   398 ; INPUTS:
                     0060   399 ;
                     0060   400 ;       IPL = 31
                     0060   401 ;
                     0060   402 ; OUTPUTS:
                     0060   403 ;
                     0060   404 ;       OTHER REGISTERS PRESERVED
                     0060   405 ;-
                     0060   406          .ENABLE LSB
                     0060   407
                     0060   408 EXE$STARTUPADP::
          17   BB    0060   409          PUSHR   #^M<R0,R1,R2,R4>        ; Save a register
    51  A7'AF   DE    0062   410          MOVAL   B^ADP_TBL_UP,R1         ; Address of startup table
          06   11    0066   411          BRB     5$                      ; Join common code
                     0068   412
                     0068   413 EXE$SHUTDWNADP::
          17   BB    0068   414          PUSHR   #^M<R0,R1,R2,R4>        ; Save a register
    51  BF'AF   DE    006A   415          MOVAL   B^ADP_TBL_DWN,R1        ; Address of shutdown table
FFFFFFFC'9F   DE    006E   416 5$:       MOVAL   @#<IO[$GL_ADPLIST- -
          52            0074   417                    ADP$L_LINK>,R2      ; Get pointer to head of adapter list
    52  04 A2   D0    0075   418 10$:     MOVL    ADP$L_LINK(R2),R2       ; Flink onward
          11   13    0079   419          BEQL    20$                     ; Branch if at end of list
       54  62   D0    007B   420          MOVL    ADP$L_CSR(R2),R4        ; Get address of CSR
 50  0E A2   3C    007E   421          MOVZWL  ADP$W_ADPTYPE(R2),R0    ; Get adapter type code
  50  6140   DE    0082   422          MOVAL   (R1)[R0],R0             ; Get table entry of adap shutdown
    00 B040   16    0086   423          JSB     @(R0)[R0]               ; Call adapter shutdown
          E9   11    008A   424          BRB     10$                     ; Next adapter
                     008C   425
          17   BA    008C   426 20$:     POPR    #^M<R0,R1,R2,R4>
          05   008E   427 30$:     RSB
                     008F   428
                     008F   429 ;
                     008F   430 ; Table of addresses of adapter shutdown routines ordered
                     008F   431 ; by adapter type in ADP$W_ADPTYPE.
                     008F   432 ;
                     008F   433
                     008F   434 ADP_TBL_DWN:                            ; Address table start
        FFFFFFFF     008F   435          .LONG   30$-.                   ; 0-MBA
        FFFFFFFB     0093   439          .LONG   30$-.                   ; 1-UBA
        FFFFFFF7     0097   441          .LONG   30$-.                   ; 2-DR32
        FFFFFFF3     009B   442          .LONG   30$-.                   ; 3-MA780
        FFFFFF61'    009F   443          .LONG   CI$SHUTDOWN-.           ; 4-CI
        FFFFFFEB     00A3   444          .LONG   30$-.                   ; Rsvrd for future expansion
                     00A7   445
                     00A7   446 ;
                     00A7   447 ; Table of addresses of adapter startup routines ordered
                     00A7   448 ; by adapter type in ADP$W_ADPTYPE.
                     00A7   449 ;
                     00A7   450
                     00A7   451 ADP_TBL_UP:                             ; Address table start
```

```
FFFFFF59' 00A7   452          .LONG   MBA$INITIAL-.           ; 0-MBA
FFFFFF55' 00AB   453          .LONG   UBA$INITIAL-.           ; 1-UBA
FFFFFFDF  00AF   454          .LONG   30$-.                   ; 2-DR32
FFFFFF4D' 00B3   455          .LONG   MA$INITIAL-.            ; 3-MA780
FFFFFFD7  00B7   456          .LONG   30$-.                   ; 4-CI
FFFFFFD3  00BB   457          .LONG   30$-.                   ; Rsvrd for future expansion
          00BF   458
          00BF   459          .DISABLE LSB
```

```
                     00BF    461          .SBTTL  EXE$DUMPCPUREG - DUMP CPU-SPECIFIC IPR'S
                     00BF    462 ;+
                     00BF    463 ; DUMP CPU-SPECIFIC IPR'S INTO ERROR MESSAGE BUFFER.
                     00BF    464 ;
                     00BF    465 ; TWENTY-FOUR LONGWORDS ARE RESERVED IN THE EMB FOR CPU-SPECIFIC
                     00BF    466 ; IPR'S.  THE FORMATS FOR VARIOUS CPU'S ARE:
                     00BF    467 ;
                     00BF    468 ; 11/780:       11/750:         11/730:         11/790:                 uVAX I:
                     00BF    469 ;
                     00BF    470 ; ICR           ICR             ICR             ICR                     UNUSED(0)
                     00BF    471 ; TODR          TODR            TODR            TODR                    APPROX TODR
                     00BF    472 ; ACCS          ACCS            ACCS            ACCS                    UNUSED(0)
                     00BF    473 ; SBIFS         TBDR            21 UNUSED(0)    SBISTS  (1st SBI)       21 UNUSED(0)
                     00BF    474 ; SBISC         CADR                            SILOCMP      ''
                     00BF    475 ; SBIMT         MCESR                           MAINT        ''
                     00BF    476 ; SBIER         CAER                            SBIERR       ''
                     00BF    477 ; SBIS          CMIERR                          TMOADDRS     ''
                     00BF    478 ; 16 SBI SILO   16 UNUSED(0)                    16 SBI SILO  ''
                     00BF    479 ;
                     00BF    480 ; INPUTS:
                     00BF    481 ;
                     00BF    482 ;       R0 - ADDR IN EMB OF START OF CPU-SPECIFIC REGISTERS=
                     00BF    483 ;            OFFSET EMB$L_CR_CPUREG
                     00BF    484 ;
                     00BF    485 ; OUTPUTS:
                     00BF    486 ;
                     00BF    487 ;       R0,R1 DESTROYED
                     00BF    488 ;       ALL OTHER REGISTERS PRESERVED
                     00BF    489 ;-
                     00BF    490
                     00BF    491          .ENABL  LSB
                     00BF    492
                     00BF    493 EXE$DUMPCPUREG::                        ;SUBROUTINE ENTRY
                     00BF    494
                     00BF    495
                     00BF    509
                     00BF    510
80   1A   DB   00BF    512          MFPR    #PR750$_ICR,(R0)+              ;LOG INTERVAL COUNT REG,
80   1B   DB   00C2    513          MFPR    #PR750$_TODR,(R0)+             ; TIME-OF-DAY REG,
80   28   DB   00C5    514          MFPR    #PR750$_ACCS,(R0)+             ; ACCELERATOR CONTROL REG,
80   24   DB   00C8    515          MFPR    #PR750$_TBDR,(R0)+                     ; TB DISABLE REG,
80   25   DB   00CB    516          MFPR    #PR750$_CADR,(R0)+             ; CACHE DISABLE REG,
80   26   DB   00CE    517          MFPR    #PR750$_MCESR,(R0)+            ; MCHECK ERROR SUMMARY REG
80   27   DB   00D1    518          MFPR    #PR750$_CAER,(R0)+             ; CACHE ERROR REG
80   17   DB   00D4    519          MFPR    #PR750$_CMIERR,(R0)+           ; CMI ERROR SUMMARY REGISTER
51   10   D0   00D7    520          MOVL    #<EMB$L_CR_CODE-<EMB$L_CR_CMIERR+4>>/4,R1 ;GET # LONGWDS OF
     80   D4   00DA    521 10$:     CLRL    (R0)+                          ; CPU-SPECIFIC REG LEFT IN EMB
  FB 51   F5   00DC    522          SOBGTR  R1,10$                         ; AND ZERO THEM
                     00DF    524
                     00DF    525
                     00DF    536
                     00DF    537
                     00DF    558
                     00DF    559
                     00DF    572 90$:
               05    00DF    573          RSB
                     00E0    574          .DISABLE LSB
```

```
        00E0   575
```

```
                     00E0   577          .SBTTL  EXE$READ_TODR (P) - READ TIME-OF-DAY CLOCK
                     00E0   578 ;+
                     00E0   579 ; READS THE TIME-OF-DAY CLOCK, SINCE IT MAY BE ACCESSED IN
                     00E0   580 ; DIFFERENT WAYS:  AS AN INTERNAL PROCESSOR REGISTER, AS PART
                     00E0   581 ; OF THE CONSOLE, OR BY READING AN ADDRESS IN I/O SPACE.  IT
                     00E0   582 ; MAY ALSO BE IN DIFFERENT FORMATS AND HAVE TO BE CONVERTED.
                     00E0   583 ;
                     00E0   584 ; INPUTS:
                     00E0   585 ;
                     00E0   586 ;       NONE.
                     00E0   587 ;
                     00E0   588 ; OUTPUTS:
                     00E0   589 ;
                     00E0   590 ;       R0 - TODR VALUE
                     00E0   591 ;       ALL OTHER REGISTERS PRESERVED
                     00E0   592 ;-
                     00E0   593
                     00E0   594 EXE$READP_TODR::                             ; SUBROUTINE ENTRY
                     00E0   595
                     00E0   596         ; NAUTILUS PROCESSOR NEEDS TO USE A SEPARATE ROUTINE TO ACCESS PHYSICAL TODR
                     00E0   597         ; REGISTER IN THE CONSOLE PROCESSOR FOR TWO REASONS.  FIRST, THE PHYSICAL
                     00E0   598         ; TODR HAS ONE SECOND RESOLUTION INSTEAD OF 10 MSEC RESOLUTION.  SECOND, A
                     00E0   599         ; REFERENCE TO THE PHYSICAL TODR IS A VERY SLOW, NON-INTERRUPTIBLE ACTION.
                     00E0   600         ; NON-PHYSICAL NAUTILUS TODR REFERENCES WILL USE THE EXE$READ_TODR ENTRY
                     00E0   601         ; WHICH WILL FABRICATE THE TIME FROM THE QUADWORD SYSTEM TIME.
                     00E0   602
                     00E0   603                                             ; NOT NAUTILUS - FALL THROUGH TO READ_TODR
                     00E0   604
                     00E0   605 EXE$READ_TODR::                              ; SUBROUTINE ENTRY
                     00E0   606
                     00E0   607
                     00E0   611
                     00E0   612
50   1B   DB        00E0   614          MFPR    #PR750$_TODR,R0             ; TODR IS A PROCESSOR REGISTER.
                     00E3   616
                     00E3   617
                     00E3   621
                     00E3   622
                     00E3   626
                     00E3   662
                05   00E3   663          RSB
                     00E4   664
```

```
                 00E4          .SBTTL  EXE$WRITE_TODR (P) - WRITES TIME-OF-DAY CLOCK
                 00E4      ;+
                 00E4      ; WRITES THE TIME-OF-DAY CLOCK, SINCE IT MAY BE ACCESSED IN
                 00E4      ; DIFFERENT WAYS:  AS AN INTERNAL PROCESSOR REGISTER, AS PART
                 00E4      ; OF THE CONSOLE, OR BY READING AN ADDRESS IN I/O SPACE.  IT
                 00E4      ; MAY ALSO BE IN DIFFERENT FORMATS AND HAVE TO BE CONVERTED.
                 00E4      ;
                 00E4      ; INPUTS:
                 00E4      ;
                 00E4      ;       R0 - CONTAINS VALUE TO BE WRITTEN INTO TODR
                 00E4      ;
                 00E4      ; OUTPUTS:
                 00E4      ;
                 00E4      ;       NEW TIME VALUE WRITTEN INTO TODR.
                 00E4      ;       ALL REGISTERS PRESERVED.
                 00E4      ;-
                 00E4
                 00E4  EXE$WRITEP_TODR::                          ; SUBROUTINE ENTRY
                 00E4
                 00E4          ; NAUTILUS PROCESSOR NEEDS TO USE A SEPARATE ROUTINE TO ACCESS PHYSICAL TODR
                 00E4          ; REGISTER IN THE CONSOLE PROCESSOR FOR TWO REASONS.  FIRST, THE PHYSICAL
                 00E4          ; TODR HAS ONE SECOND RESOLUTION INSTEAD OF 10 MSEC RESOLUTION.  SECOND, A
                 00E4          ; REFERENCE TO THE PHYSICAL TODR IS A VERY SLOW, NON-INTERRUPTIBLE ACTION.
                 00E4          ; NON-PHYSICAL NAUTILUS TODR REFERENCES WILL USE THE EXE$WRITE_TODR ENTRY
                 00E4          ; WHICH WILL FABRICATE A NEW QUADWORD SYSTEM TIME.
                 00E4
                 00E4                                             ; NOT NAUTILUS - FALL THROUGH TO WRITE_TODR
                 00E4
                 00E4  EXE$WRITE_TODR::                           ; SUBROUTINE ENTRY
                 00E4
                 00E4
                 00E4
                 00E4
 1B    50    DA  00E4          MTPR    R0,#PR750$_TODR            ; TODR IS A PROCESSOR REGISTER.
                 00E7
                 00E7
                 00E7
                 00E7
                 00E7
                 00E7
                 00E7
             05  00E7          RSB
```

```
                                00E8   724          .SBTTL  EXE$REGSAVE - SAVE CPU-SPECIFIC IPR'S
                                00E8   725 ;+
                                00E8   726 ; EXE$REGSAVE - CALLED BY POWERFAIL TO SAVE CPU-SPECIFIC IPR'S ON
                                00E8   727 ;               THE STACK
                                00E8   728 ;
                                00E8   729 ; INPUTS: NONE
                                00E8   730 ;
                                00E8   731 ; OUTPUTS:
                                00E8   732 ;
                                00E8   733 ;               R0 DESTROYED
                                00E8   734 ;               OTHER GENERAL REGISTERS PRESERVED
                                00E8   735 ;               IPR'S SAVED ON THE STACK AS FOLLOWS:
                                00E8   736 ;
                                00E8   737 ;        11/780:     11/750:        11/730:        11/790:   uVAX I:
                                00E8   738 ;
                                00E8   739 ; 0(SP)   PME          PME            PME            ACCS      (none)
                                00E8   740 ; 4(SP)   SBIMT        TBDR                          CSWP
                                00E8   741 ; 8(SP)                CADR                          PME
                                00E8   742 ;
                                00E8   743 ;-
                                00E8   744
                                00E8   745          .ENABL  LSB
                                00E8   746
                                00E8   747 EXE$REGSAVE::                                ;SUBROUTINE ENTRY
          01    BA              00E8   749          POPR    #^M<R0>                     ;CLEAR RETURN FROM STACK
                                00EA   750
                                00EA   751
                                00EA   756
                                00EA   757
     7E   3D    DB              00EA   759          MFPR    #PR750$_PME,-(SP)           ;SAVE PERFORMANCE MONITOR ENABLE
     7E   25    DB              00ED   760          MFPR    #PR750$_CADR,-(SP)          ;SAVE CACHE DISABLE REG,
     7E   24    DB              00F0   761          MFPR    #PR750$_TBDR,-(SP)          ; AND TB DISABLE REG
                                00F3   763
                                00F3   764
                                00F3   768
                                00F3   769
                                00F3   776
          60    17              00F3   777          JMP     (R0)                        ;DONE, RETURN
                                00F5   779
                                00F5   783
                                00F5   784          .DSABL  LSB
```

```
                     00F5   786          .SBTTL  EXE$REGRESTOR - RESTORE CPU-SPECIFIC IPR'S
                     00F5   787 ;+
                     00F5   788 ; EXE$REGRESTOR - CALLED BY POWERFAIL RECOVERY TO RESTORE CPU-SPECIFIC
                     00F5   789 ;                 IPR'S FROM THE STACK.
                     00F5   790 ;
                     00F5   791 ; INPUTS:
                     00F5   792 ;
                     00F5   793 ;                 R6 - TOP OF STACK
                     00F5   794 ;                 STACK SET UP AS DEFINED IN OUTPUTS OF EXE$REGSAVE.
                     00F5   795 ;
                     00F5   796 ; OUTPUTS:
                     00F5   797 ;
                     00F5   798 ;                 R0 DESTROYED
                     00F5   799 ;                 OTHER GENERAL REGISTERS PRESERVED
                     00F5   800 ;                 CPU-SPECIFIC IPR'S RESTORED FROM STACK
                     00F5   801 ;                 R6 - ADDRESS OF 1ST CPU-INDEPENDENT SAVED IPR
                     00F5   802 ;
                     00F5   803 ;-
                     00F5   804
                     00F5   805          .ENABL  LSB
                     00F5   806
                     00F5   807 EXE$REGRESTOR::                          ;SUBROUTINE ENTRY
          01   BA   00F5   809          POPR    #^M<R0>                    ;CLEAR RETURN FROM STACK
                     00F7   810
                     00F7   811
                     00F7   816
                     00F7   817
     24   86   DA   00F7   819          MTPR    (R6)+,#PR750$_TBDR         ;RESTORE TB DISABLE REG,
     25   86   DA   00FA   820          MTPR    (R6)+,#PR750$_CADR         ; AND CACHE DISABLE REG
     3D   86   DA   00FD   821          MTPR    (R6)+,#PR750$_PME          ;RESTORE PERFORMANCE MONITOR ENABLE
                     0100   823
                     0100   824
                     0100   828
                     0100   829
                     0100   837
     60   17        0100   838          JMP     (R0)                       ;DONE, RETURN
                     0102   843
                     0102   844          .DSABL  LSB
```

```
                                 0102    846          .SBTTL  EXE$INIPROCREG - CPU-DEPENDENT INITIALIZATION OF IPR'S
                                 0102    847 ;+
                                 0102    848 ; EXE$INIPROCREG - PERFORM INITIALIZATION OF INTERVAL TIMER AND
                                 0102    849 ;       CPU-DEPENDENT REGISTERS. CALLED FROM INIT AND POWERFAIL.
                                 0102    850 ;
                                 0102    851 ; INPUTS:
                                 0102    852 ;
                                 0102    853 ;       NONE
                                 0102    854 ;
                                 0102    855 ; OUTPUTS:
                                 0102    856 ;
                                 0102    857 ;       NONE
                                 0102    858 ;-
                                 0102    859
                                 0102    860 EXE$INIPROCREG::                        ; INIT PROCESSOR REGISTERS
                                 0102    861
                                 0102    875
                      00'  E1    0102    876          BBC     S^#EXE$V_CRDENABL,-    ; BRANCH IF FLAG CLEAR
          12 00000000'9F         0104    877                  @#EXE$GL_FLAGS,20$     ;  (IGNORE ERRORS)
     50   00000000'9F   D0    010A    878          MOVL    @#MMG$GL_SBICONF,R0    ; GET ADDR OF MEMORY CONTROLLER
          50    60      D0    0111    879          MOVL    (R0),R0                ;  CONFIG REGISTER (1ST SLOT)
  04 A0    10000000 8F   C8    0114    880          BISL    #<1@28>,4(R0)          ; SET CRD REPORT BIT
                                 011C    881                                     ;
                      00'  E0    011C    925 20$:     BBS     S^#EXE$V_NOCLOCK,-
       0E 00000000'9F           011E    926                  @#EXE$GL_FLAGS,30$     ; BRANCH IF NOT USING CLOCK
                                 0124    927
                                 0124    931
                                 0124    935
     19   FFFFD8F0 8F   DA    0124    937          MTPR    #-<10*1000>,S^#PR750$_NICR ; LOAD NEXT INTERVAL REGISTER
                                 012B    939
                                 012B    943
     18   800000D1 8F   DA    012B    944          MTPR    #^X800000D1,S^#PR$_ICCS ; CLEAR ERROR AND START CLOCK
                        05    0132    945 30$:     RSB                             ; AND RETURN
                                 0133    946
                                 0133    962
```

```
                     0133   985          .SBTTL  SYSL$CLRSBIA
                     0133   986 ;++
                     0133   987 ; SYSL$CLRSBIA  - ON 11/790, CLEAR SBIA ERROR REGISTERS
                     0133   988 ;                - ON 11/780, 11/750, 11/730, AND MICRO-VAX I, THIS IS A NOP
                     0133   989 ;
                     0133   990 ; THIS ROUTINE IS CALLED TO CLEAR OUT SBIA ERROR BITS AFTER A MACHINE CHECK
                     0133   991 ; OCCURS (WHEN MACHINE CHECK IS HANDLED LOCALLY).
                     0133   992 ;
                     0133   993 ; THIS ROUTINE SHOULD BE CALLED AT IPL 31.
                     0133   994 ;
                     0133   995 ; INPUTS:
                     0133   996 ;       ABUS_TYPE       - AN ARRAY TYPE CODES; IDENTIFIES EACH ADAPTER ON THE
                     0133   997 ;                         ABUS.
                     0133   998 ;       ABUS_VA         - AN ARRAY OF ADAPTER SPACE VA'S FOR EACH ADAPTER
                     0133   999 ;                         ON THE ABUS.
                     0133  1000 ;
                     0133  1001 ; OUTPUTS:
                     0133  1002 ;       SBI ERROR BITS ARE CLEARED FOR EACH SBIA ON THE ABUS.
                     0133  1003 ;       ALL REGISTERS PRESERVED.
                     0133  1004 ;--
                     0133  1005 SYSL$CLRSBIA::
 05                  0133  1023          RSB                             ; AND RETURN
```

```
                                0134  1025          .SBTTL  EXE$TEST_CSR
                                0134  1026  ;+
                                0134  1031  ; EXE$TEST_CSR - TEST A UNIBUS CONTROLLER CSR FOR EXISTENCE
                                0134  1033  ;
                                0134  1034  ; THIS TEST IS CPU-DEPENDENT.  THE FOLLOWING CPU'S ARE SUPPORTED:
                                0134  1035  ;
                                0134  1036  ;       11/780  -TEST CSR AND CHECK RESULT IN THE UBA STATUS REGISTER.
                                0134  1037  ;       11/750  -NON-EXISTENT CSR IS REPORTED VIA MACHINE CHECK AS A
                                0134  1038  ;                NON-EXISTENT MEMORY REFERENCE.  CONNECT A TEMPORARY
                                0134  1039  ;                MACHINE CHECK HANDLER, TEST THE CSR, AND RESTORE THE
                                0134  1040  ;                ORIGINAL MACHINE CHECK HANDLER.
                                0134  1041  ;       11/730  -ACTION IS THE SAME AS FOR THE 11/750.
                                0134  1042  ;       11/790  -ACTION IS THE SAME AS FOR THE 11/780.
                                0134  1043  ;       MICRO-VAX I -ACTION IS SAME AS FOR THE 11/750.
                                0134  1044  ;
                                0134  1045  ; THIS SUBROUTINE SHOULD BE CALLED VIA BRANCH OR JUMP TO SUBROUTINE AT IPL 31.
                                0134  1046  ;
                                0134  1047  ; INPUTS:
                                0134  1048  ;
                                0134  1049  ;       R0 = CSR ADDRESS
                                0134  1050  ;       R6 = ADAPTER CONFIGURATION REGISTER ADDRESS
                                0134  1051  ;
                                0134  1052  ; OUTPUTS:
                                0134  1053  ;
                                0134  1054  ;       R0 LOW BIT SET/CLEAR FOR EXISTENT/NONEX CSR
                                0134  1055  ;       OTHER REGISTERS PRESERVED.
                                0134  1056  ;-
                                0134  1057
                                0134  1058          .ENABL  LSB
                                0134  1059
                                0134  1060  EXE$TEST_CSR::                          ;SUBROUTINE ENTRY
                                0134  1061
                      06   BB   0134  1062          PUSHR   #^M<R1,R2>              ;SAVE REGISTERS
                                0136  1063
                                0136  1122
                  00000024      0136  1123          MCK_BER = ^X24                  ;OFFSET INTO 750 MACHINE CHECK FRAME
                                0136  1124                                          ; FOR BUS ERROR REGISTER
                  00000003      0136  1125          NEX     = 3                     ; BIT POSITION FOR NON-EXISTENT MEMORY
                                0136  1130
                                0136  1141
   51   00000000'GF        D0   0136  1142  10$:    MOVL    G^EXE$GL_SCB,R1         ;GET SCB ADDRESS
                  04 A1    DD   013D  1143          PUSHL   4(R1)                   ;SAVE CURRENT MCHECK HANDLER ADDR
               52    5E    D0   0140  1144          MOVL    SP,R2                   ;MARK CURRENT STACK POSITION
       04 A1    50'AF      DE   0143  1145          MOVAL   B^MCHK_HANDLER,4(R1)    ;CONNECT TEMP MCHECK HANDLER
                     60    B5   0148  1146          TSTW    (R0)                    ;ATTEMPT TO READ CSR
               50    01    9A   014A  1147  OK:     MOVZBL  #SS$_NORMAL,R0          ;IF NO MCHECK, SET STATUS TO
                                014D  1148                                          ; SUCCESS
                     19    11   014D  1149          BRB     TEST_DONE               ;JOIN COMMON EXIT
                                014F  1150
                                014F  1151  ;
                                014F  1152  ; TEMPORARY CSR TEST MACHINE CHECK HANDLER
                                014F  1153  ;
                                014F  1154
                                014F  1155          .ALIGN  LONG                    ;REQ'D MACHINE CHECK ALIGNMENT
                                0150  1156  MCHK_HANDLER:                           ;
                                0150  1157
          26    0F         DA   0150  1159          MTPR    #^XF,#PR750$_MCESR      ;CLEAR NON-EX MEMORY CONDITION
```

```
                        0153  1161
                        0153  1165
                        0153  1169
                        0153  1170
    50    08    D0      0153  1172          MOVL    #<1@NEX>,R0             ;SETUP
    0C    6E    D1      0156  1173          CMPL    (SP),#^X0C              ;IS THIS A 730 FRAME?
          04    13      0159  1174          BEQL    50$                     ;YES, THEN DON'T CHECK FURTHER
50    24  AE    D0      015B  1175          MOVL    MCK_BER(SP),R0          ;SAVE BUS ERROR REGISTER
    5E    52    D0      015F  1176 50$:     MOVL    R2,SP                   ;CLEAR MCHECK INFO FROM STACK
E4  50    03    E1      0162  1177          BBC     #NEX,R0,OK              ;MEMORY EXISTS, PARITY FAILURE
                        0166  1179
                        0166  1186
                        0166  1188 NONEX_DEV:                              ;
          50    D4      0166  1189          CLRL    R0                      ;SET STATUS TO FAILURE
                        0168  1190 TEST_DONE:
       04 A1  8ED0      0168  1191          POPL    4(R1)                   ;RESTORE SYSTEM MCHECK HANDLER
                        016C  1192 TEST_DONE_2:
          06    BA      016C  1193          POPR    #^M<R1,R2>              ;RESTORE REGISTERS
                05      016E  1194          RSB                             ;RETURN RESULT TO CALLER
                        016F  1195          .DISABLE LSB
```

```
                                 016F  1197          .SBTTL  ADPLINK - LINK ADAPTER CONTROL BLOCK INTO ADP LIST
                                 016F  1198 ;+
                                 016F  1199 ; ADPLINK LINKS THE ADAPTER CONTROL BLOCK TO THE END OF THE ADP LIST
                                 016F  1200 ;
                                 016F  1201 ; INPUT:
                                 016F  1202 ;       R2 - ADDRESS OF NEW ADP
                                 016F  1203 ; OUTPUTS:
                                 016F  1204 ;       ADP IS LINK TO THE END OF THE ADPLIST LOCATED BY IOC$GL_ADPLIST.
                                 016F  1205 ;       R0,R1 destroyed.
                                 016F  1206 ;-
                                 016F  1207
                                 016F  1208 ADPLINK::                                            ;
      50   FFFFFFFC'9F   9E      016F  1209          MOVAB   @#<IOC$GL_ADPLIST-ADP$L_LINK>,R0
                                 0176  1210                                                      ; START OF LIST
       51    04 A0    D0      0176  1211 10$:     MOVL    ADP$L_LINK(R0),R1                   ; FLINK TO FIRST ENTRY
                 05    13      017A  1212          BEQL    20$                                 ; AT END
            50   51    D0      017C  1213          MOVL    R1,R0                               ; TRY AGAIN
                 F5    11      017F  1214          BRB     10$                                 ;
       04 A0     52    D0      0181  1215 20$:     MOVL    R2,ADP$L_LINK(R0)                   ; CHAIN NEW ADP TO END OF LIST
                       05      0185  1216          RSB                                         ; AND RETURN
                               0186  1217
                               0186  1218          .END
```

```
ADP$L_CSR                       = 00000000
ADP$L_LINK                      = 00000004
ADP$W_ADPTYPE                   = 0000000E
ADPLINK                           0000016F RG     03
ADP_TBL_DWN                       0000008F R      03
ADP_TBL_UP                        000000A7 R      03
BQO$L_UMR_DIS                   = 00000024
BQO$W_VERSION                   = 00000010
BTD$K_CONSOLE                   = 00000040
C750_LIKE                       = 00000001
C780_LIKE                       = 00000000
CI$SHUTDOWN                       ********    X   03
CPU_TYPE                        = 00000002
EMB$L_CR_CMIERR                 = 00000080
EMB$L_CR_CODE                   = 000000F4
EXE$DUMPCPUREG                    000000BF RG     03
EXE$EXTRA1                        00000000 RG     01
EXE$EXTRA10                       00000000 RG     01
EXE$EXTRA2                        00000000 RG     01
EXE$EXTRA3                        00000000 RG     01
EXE$EXTRA4                        00000000 RG     01
EXE$EXTRA5                        00000000 RG     01
EXE$EXTRA6                        00000000 RG     01
EXE$EXTRA7                        00000000 RG     01
EXE$EXTRA8                        00000000 RG     01
EXE$EXTRA9                        00000000 RG     01
EXE$GL_FLAGS                      ********    X   03
EXE$GL_SCB                        ********    X   03
EXE$INITBOOTADP                   00000000 RG     03
EXE$INIPROCREG                    00000102 RG     03
EXE$READP_TODR                    000000E0 RG     03
EXE$READ_TODR                     000000E0 RG     03
EXE$REGRESTOR                     000000F5 RG     03
EXE$REGSAVE                       000000E8 RG     03
EXE$SHUTDWNADP                    00000068 RG     03
EXE$STARTUPADP                    00000060 RG     03
EXE$TEST_CSR                      00000134 RG     03
EXE$V_CRDENABL                    ********    X   03
EXE$V_NOCLOCK                     ********    X   03
EXE$WRITEP_TODR                   000000E4 RG     03
EXE$WRITE_TODR                    000000E4 RG     03
INI_UBADP                         00000038 R      03
IOC$GL_ADPLIST                    ********    X   03
MA$INITIAL                        ********    X   03
MBA$INITIAL                       ********    X   03
MBA$L_CR                        = 00000004
MBA$L_SR                        = 00000008
MBA$M_CR_ABORT                  = 00000002
MBA$M_CR_INIT                   = 00000001
MCHK_HANDLER                      00000150 R      03
MCK_BER                         = 00000024
MMG$GL_SBICONF                    *********   X   03
NDT$_CT                         = 00000038
NDT$_MB                         = 00000020
NEX                             = 00000003
NONEX_DEV                         00000166 R      03
OK                                0000014A R      03

PR$_ICCS                        = 00000018
PR$_SID_TYP730                  = 00000003
PR$_SID_TYP750                  = 00000002
PR$_SID_TYP780                  = 00000001
PR$_SID_TYP790                  = 00000004
PR$_SID_TYPUV1                  = 00000007
PR750$_ACCS                     = 00000028
PR750$_CADR                     = 00000025
PR750$_CAER                     = 00000027
PR750$_CMIERR                   = 00000017
PR750$_ICR                      = 0000001A
PR750$_MCESR                    = 00000026
PR750$_NICR                     = 00000019
PR750$_PME                      = 0000003D
PR750$_TBDR                     = 00000024
PR750$_TODR                     = 0000001B
PR750$_UBRESET                  = 00000037
RPB$B_DEVTYP                    = 00000066
RPB$L_ADPVIR                    = 00000060
RPB$L_IOVEC                     = 00000034
RPB$W_BOOTNDT                   = 000000A1
SS$_NORMAL                      = 00000001
SYS[$CLRSBIA                      00000133 RG     03
TEST_DONE                         00000168 R      03
TEST_DONE_2                       0000016C R      03
UBA$INITIAL                       ********    X   03
UBA$L_MAP                       = 00000800
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
. BLANK .	00000001 (1.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
ABS	00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SYSLOA	00000186 (390.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.02	00:00:03.66
Command processing	119	00:00:00.48	00:00:03.92
Pass 1	343	00:00:07.43	00:00:31.73
Symbol table sort	0	00:00:01.09	00:00:03.55
Pass 2	133	00:00:02.02	00:00:09.53
Symbol table output	11	00:00:00.06	00:00:00.06
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	639	00:00:11.13	00:00:52.47

The working set limit was 1500 pages.
70766 bytes (139 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 1061 non-local and 17 local symbols.
1222 source lines were read in Pass 1, producing 16 object records in Pass 2.
20 pages of virtual memory were used to define 19 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	10
_255DUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	16

1124 GETS were required to define 16 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:ERRSUB750/OBJ=OBJ$:ERRSUB750 MSRC$:CPUSW750/UPDATE=(ENH$:CPUSW750)+MSRC$:ERRSUB/UPDATE=(ENH$:ERRSUB)+EXECML$/LIB
```

ERRSUBUV1
LIS

ERRSUB790
LIS

ERRSUB780
LIS

ERRSUB750
LIS

ERRSUB730
LIS

FALMAN
LIS