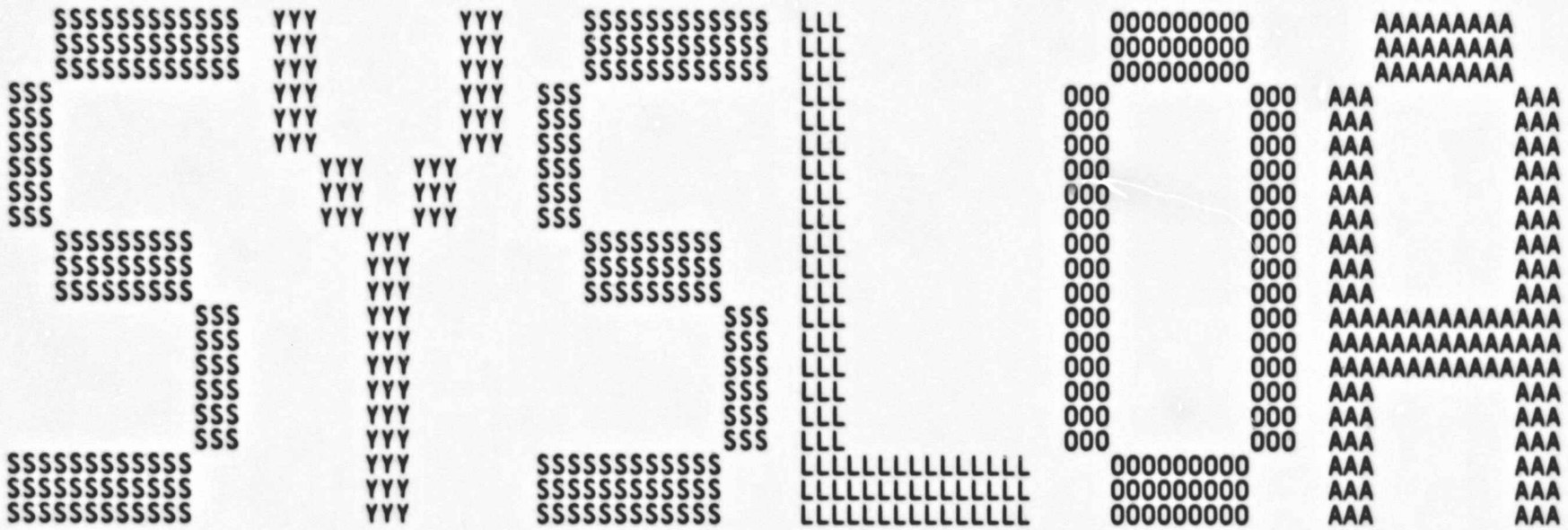
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL OOOOOOOOO AAAAAAAAA
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL OOOOOOOOO AAAAAAAAA
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL OOOOOOOOO AAAAAAAAA
SSS YYY YYY SSS LLL OOO OOO AAA AAA
SSS YYY YYY SSS LLL OOO OOO AAA AAA
SSS YYY YYY SSS LLL OOO OOO AAA AAA
SSS YYY YYY SSS LLL OOO OOO AAA AAA
SSS YYY YYY SSS LLL OOO OOO AAA AAA
SSS YYY YYY SSS LLL OOO OOO AAA AAA
SSSSSSSSS YYY SSSSSSSSS LLL OOO OOO AAA AAA
SSSSSSSSS YYY SSSSSSSSS LLL OOO OOO AAA AAA
SSSSSSSSS YYY SSSSSSSSS LLL OOO OOO AAA AAA
SSS YYY SSS LLL OOO OOO AAAAAAAAAAAAAAA
SSS YYY SSS LLL OOO OOO AAAAAAAAAAAAAAA
SSS YYY SSS LLL OOO OOO AAAAAAAAAAAAAAA
SSS YYY SSS LLL OOO OOO AAA AAA
SSS YYY SSS LLL OOO OOO AAA AAA
SSSSSSSSSSS YYY SSSSSSSSSSS LLLLLLLLLLLLLLL OOO OOO AAA AAA
SSSSSSSSSSS YYY SSSSSSSSSSS LLLLLLLLLLLLLLL OOOOOOOOO AAA AAA
SSSSSSSSSSS YYY SSSSSSSSSSS LLLLLLLLLLLLLLL OOOOOOOOO AAA AAA

```
EEEEEEEEEE  RRRRRRRR    RRRRRRRR      SSSSSSSS  UU      UU  BBBBBBBB    77777777      888888      000000
EEEEEEEEEE  RRRRRRRR    RRRRRRRR      SSSSSSSS  UU      UU  BBBBBBBB    77777777      888888      000000
EE          RR      RR  RR      RR  SS            UU      UU  BB      BB          77  88      88  00      00
EE          RR      RR  RR      RR  SS            UU      UU  BB      BB          77  88      88  00      00
EE          RR      RR  RR      RR  SS            UU      UU  BB      BB          77  88      88  00    0000
EE          RR      RR  RR      RR  SS            UU      UU  BB      BB          77  88      88  00    0000
EEEEEEEE    RRRRRRRR    RRRRRRRR      SSSSSS    UU      UU  BBBBBBBB        77      888888      00  00  00
EEEEEEEE    RRRRRRRR    RRRRRRRR      SSSSSS    UU      UU  BBBBBBBB        77      888888      00  00  00
EE          RR  RR      RR  RR              SS  UU      UU  BB      BB    77        88      88  0000    00
EE          RR  RR      RR  RR              SS  UU      UU  BB      BB    77        88      88  0000    00
EE          RR    RR    RR    RR            SS  UU      UU  BB      BB  77          88      88  00      00
EE          RR    RR    RR    RR            SS  UU      UU  BB      BB  77          88      88  00      00
EEEEEEEEEE  RR      RR  RR      RR  SSSSSSSS    UUUUUUUUUU  BBBBBBBB  77            888888      000000
EEEEEEEEEE  RR      RR  RR      RR  SSSSSSSS    UUUUUUUUUU  BBBBBBBB  77            888888      000000

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
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

```
                                I 9
0000     1          .NOSHOW CONDITIONALS
0000     3          .TITLE  ERRSUB780 - ERROR SUBROUTINES FOR VAX 11/780
0000     5
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
```

```
0000     72 ;               EXE$INIPROCREG to a new routine: INI$CACHE.  Correct the
0000     73 ;               order in which registers are saved on the stack in EXE$REGSAVE.
0000     74 ;
0000     75 ;       V03-022 TCM0009         Trudy C. Matthews       30-Jul-1984
0000     76 ;               When turning off CRD interrupts in EXE$INIPROCREG for VENUS,
0000     77 ;               read the processor register and write it back to preserve
0000     78 ;               the state of other bits in the register.
0000     79 ;
0000     80 ;       V03-021 TCM0008         Trudy C. Matthews       23-Jul-1984
0000     81 ;               Remove venus code that queries the console for how to set up
0000     82 ;               cache and FBOX state.  Instead always turn the cache and
0000     83 ;               FBOX on (and let the normal error handling code turn it off
0000     84 ;               if its bad).
0000     85 ;
0000     86 ;       V03-020 DWT0214         David W. Thiel          02-May-1984
0000     87 ;               Revise MicroVAX I TODR register simulation.
0000     88 ;
0000     89 ;       V03-019 KDM0096         Kathleen D. Morse       27-Mar-1984
0000     90 ;               Add missing indirection in MicroVAX I memory CSR
0000     91 ;               CRD enabling.
0000     92 ;
0000     93 ;       V03-018 KPL0101         Peter Lieberwirth        4-Mar-1984
0000     94 ;               Add extra vectors now defined in SYSLOAVEC.  These vectors
0000     95 ;               are insurance for v4.x
0000     96 ;
0000     97 ;       V03-017 KPL0100         Peter Lieberwirth       12-Feb-1984
0000     98 ;               Change RPB$B_BOOTNDT to RPB$W_BOOTNDT, since BI devices
0000     99 ;               will have 16-bit device types.
0000    100 ;
0000    101 ;       V03-016 KDM0092         Kathleen D. Morse       23-Jan-1984
0000    102 ;               Correct the number of cpu-specific IPRs logged for the
0000    103 ;               11/730 and MicroVAX I cpus.
0000    104 ;
0000    105 ;       V03-015 CWH8001         CW Hobbs                5-Dec-1983
0000    106 ;               Add entry points for EXE$READP_TODR and EXE$WRITEP_TODR
0000    107 ;               to access physical TODR register for Nautilus CPU.  For
0000    108 ;               other processors, these amount to duplicate labels on
0000    109 ;               EXE$READ_TODR and EXE$WRITE_TODR.
0000    110 ;
0000    111 ;       V03-014 KTA3088         Kerbey T. Altmann       17-Oct-1983
0000    112 ;               Fix bug in 730 conditional for EXE$INIBOOTADP.
0000    113 ;
0000    114 ;       V03-013 KDM0081         Kathleen D. Morse       13-Sep-1983
0000    115 ;               Create Micro-VAX I version.
0000    116 ;
0000    117 ;       V03-012 KDM0055         Kathleen D. Morse       12-Jul-1983
0000    118 ;               Move IPR PME into the cpu-dependent register save and
0000    119 ;               restore routines.
0000    120 ;
0000    121 ;       V03-011 KDM0049         Kathleen D. Morse       07-Jul-1983
0000    122 ;               Add the following processor registers to the cpu-specific
0000    123 ;               dump IPRs routine:  ICR, TODR, ACCS.  Add usage of
0000    124 ;               register:  EXE$READ_TODR and EXE$WRITE_TODR.
0000    125 ;
0000    126 ;       V03-010 KDM0048         Kathleen D. Morse       07-Jul-1983
0000    127 ;               Add loadable routines for referencing the time-of-day
0000    128 ;               clock:  EXE$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 LIBRARY 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   197          $PR780DEF                       ;DEFINE 11/780 INTERNAL PROCESSOR REGS
                     0000   199
                     0000   203
                     0000   207
                     0000   211 ;
                     0000   212 ; EQUATED SYMBOLS:
                     0000   213 ;
00000001             0000   215          C780_LIKE = 1
00000000             0000   216          C750_LIKE = 0
                     0000   218
                     0000   223
                     0000   228
                     0000   233
                     0000   238
                     0000   239 ;
                     0000   240 ; Define labels for two "extra" routines.  This reserves some vectors from
                     0000   241 ; SYS.EXE into SYSLOAxxx.EXE that can be patched if another routine must
                     0000   242 ; be added in between major releases.
                     0000   243 ;
                     0000   244 EXE$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$K_CONSOLE          ;BLOCK STORAGE DEVICE?
                         67     13 0005   280          BEQL    40$                     ;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$L_CR(R0)            ;
                                001F   292
                         51    1B DB 001F   294          MFPR    #PR780$_TODR,R1         ;GET CURRENT TIME (10 MS UNITS)
                                0022   296
                                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
                         52    1B DB 002B   309          MFPR    #PR780$_TODR,R2         ;GET CURRENT TIME
                                002E   311
                                002E   315
                                002E   319
                         52    51 D1 002E   320          CMPL    R1,R2                   ;CHECK FOR INTERVAL EXPIRED
                               F3 1A 0031   321          BGTRU   10$                     ;NOT YET, WAIT SOME MORE
                               01 D0 0033   322 15$:     MOVL    #MBA$M_CR_INIT,-        ;NOW INIT MBA
                            04 A0    0035   323                  MBA$L_CR(R0)            ;
                               05 0037   324 20$:     RSB                             ;DONE
                                0038   325
                                0038   327
                                0038   331
                                0038   332 INI_UBADP:                                 ;INIT UBA
                                0038   333
                               01 D0 0038   335          MOVL    #UBA$M_CR_INIT,-
                            04 A0    003A   336                  UBA$L_CR(R0)            ;INIT UBA
```

```
                00010000 8F    D3  003C   337 25$:    BITL    #UBA$M_CSR_UBIC,-
                      60       0042   338                UBA$L_CSR(R0)           ;WAIT FOR UBA INIT
                      F7       13  0043   339        BEQL    25$                     ; TO COMPLETE
                                   0045   341
                                   0045   358
                                   0045   360 ;
                                   0045   361 ; CHECK THE VMB VERSION NUMBER.  IF IT EXISTS AND IF IT IS 7 OR GREATER, THEN
                                   0045   362 ; SEE IF ANY UNIBUS MAP REGISTERS TO DISABLE.
                                   0045   363 ;
                                   0045   364
                      52    34 A6  D0  0045   365        MOVL    RPB$L_IOVEC(R6),R2      ;PICK UP THE IOVECTOR FROM RPB
                      51    10 A2  B2  0049   366        MCOMW   BQO$W_VERSION(R2),R1    ;GET VMB VERSION NUMBER 1'S COMPLEMENTED
                      12 A2    51  B1  004D   367        CMPW    R1,BQO$W_VERSION+2(R2)  ;CHECK AGAINST CHECK WORD IN VMB
                               1B  12  0051   368        BNEQ    40$                     ;IF NOT, ASSUME NO VERSION NUMBER
                      07    10 A2  B1  0053   369        CMPW    BQO$W_VERSION(R2),#7    ;VERSION 7 OR GREATER OF VMB?
                               15  1F  0057   370        BLSSU   40$                     ;NO, DON'T BOTH WITH UMR'S
                      52    24 A2  D0  0059   371        MOVL    BQO$L_UMR_DIS(R2),R2    ;GRAB THE NUMBER OF UMR'S TO DISABLE
                               0F  13  005D   372        BEQL    40$                     ;NONE, LEAVE
                                   005F   373
                04 A0 52    16     78  005F   375        ASHL    #22,R2,UBA$L_CR(R0)     ;SET THE UMR DISABLE BITS
                                   0064   377
                                   0064   378 ;
                                   0064   379 ; THIS CODE IS EXECUTED FOR ALL PROCESSORS.  ITS DISABLES ANY UNIBUS MAP
                                   0064   380 ; REGISTERS ASSOCIATED WITH UNIBUS MEMORY TO PREVENT CONTENTION BETWEEN
                                   0064   381 ; SBI AND UNIBUS ADDRESSES.
                                   0064   382 ;
                                   0064   383
                   51   0800 C0    DE  0064   384        MOVAL   UBA$L_MAP(R0),R1        ;ADDRESS OF FIRST REGISTER
                               81  D4  0069   385 30$:    CLRL    (R1)+                   ;DISABLE IT
                                 FB 52 F5  006B   386        SOBGTR  R2,30$                  ;LOOP UNTIL ALL DONE
                                       05  006E   388 40$:    RSB                     ;DONE WITH UBA INIT
                                           006F   389        .DISABLE LSB
```

```
                       006F        .SBTTL  EXE$SHUTDWNADP - SHUTDOWN ANY ADAPTERS DURING BUGCHECK
                       006F        .SBTTL  EXE$STARTUPADP - STARTUP ANY ADAPTERS
                       006F  ;+
                       006F  ; EXE$SHUTDWNADP - SHUTDOWN ANY ADAPTERS DURING BUGCHECK
                       006F  ;       THIS ROUTINE IS CALLED FROM BUGCHECK BEFORE THE DUMP IS TAKEN TO
                       006F  ;       ENSURE THAT ALL ADAPTERS THAT NEED TO BE QUIESENT ARE.
                       006F  ;
                       006F  ; INPUTS:
                       006F  ;
                       006F  ;       IPL = 31
                       006F  ;
                       006F  ; OUTPUTS:
                       006F  ;
                       006F  ;       OTHER REGISTERS PRESERVED
                       006F  ;-
                       006F          .ENABLE LSB
                       006F
                       006F  EXE$STARTUPADP::
            17    BB   006F          PUSHR   #^M<R0,R1,R2,R4>          ; Save a register
     51  B6'AF    DE   0071          MOVAL   B^ADP_TBL_UP,R1           ; Address of startup table
            06    11   0075          BRB     5$                        ; Join common code
                       0077
                       0077  EXE$SHUTDWNADP::
            17    BB   0077          PUSHR   #^M<R0,R1,R2,R4>          ; Save a register
     51  9E'AF    DE   0079          MOVAL   B^ADP_TBL_DWN,R1          ; Address of shutdown table
   FFFFFFFC'9F    DE   007D  5$:     MOVAL   @#<IOC$GL_ADPLIST- -
            52         0083                  ADP$L_LINK>,R2          ; Get pointer to head of adapter list
     52  04 A2    D0   0084  10$:    MOVL    ADP$L_LINK(R2),R2         ; Flink onward
            11    13   0088          BEQL    20$                       ; Branch if at end of list
        54  62    D0   008A          MOVL    ADP$L_CSR(R2),R4          ; Get address of CSR
     50  0E A2    3C   008D          MOVZWL  ADP$W_ADPTYPE(R2),R0      ; Get adapter type code
      50  6140    DE   0091          MOVAL   (R1)[R0],R0               ; Get table entry of adap shutdown
      00  B040    16   0095          JSB     @(R0)[R0]                 ; Call adapter shutdown
            E9    11   0099          BRB     10$                       ; Next adapter
                       009B
            17    BA   009B  20$:    POPR    #^M<R0,R1,R2,R4>
            05         009D  30$:    RSB
                       009E
                       009E  ;
                       009E  ; Table of addresses of adapter shutdown routines ordered
                       009E  ; by adapter type in ADP$W_ADPTYPE.
                       009E  ;
                       009E
                       009E  ADP_TBL_DWN:                              ; Address table start
      FFFFFFFF         009E          .LONG   30$-.                     ; 0-MBA
      FFFFFFFB         00A2          .LONG   30$-.                     ; 1-UBA
      FFFFFFF7         00A6          .LONG   30$-.                     ; 2-DR32
      FFFFFFF3         00AA          .LONG   30$-.                     ; 3-MA780
      FFFFFF52'        00AE          .LONG   CI$SHUTDOWN-.             ; 4-CI
      FFFFFFEB         00B2          .LONG   30$-.                     ; Rsvrd for future expansion
                       00B6
                       00B6  ;
                       00B6  ; Table of addresses of adapter startup routines ordered
                       00B6  ; by adapter type in ADP$W_ADPTYPE.
                       00B6  ;
                       00B6
                       00B6  ADP_TBL_UP:                               ; Address table start
```

```
                                                C 10
FFFFFF4A' 00B6   452          .LONG   MBA$INITIAL-.           ; 0-MBA
FFFFFF46' 00BA   453          .LONG   UBA$INITIAL-.           ; 1-UBA
FFFFFFDF  00BE   454          .LONG   30$-.                   ; 2-DR32
FFFFFF3E' 00C2   455          .LONG   MA$INITIAL-.            ; 3-MA780
FFFFFFD7  00C6   456          .LONG   30$-.                   ; 4-CI
FFFFFFD3  00CA   457          .LONG   30$-.                   ; Rsvrd for future expansion
          00CE   458
          00CE   459          .DISABLE LSB
```

```
                     00CE   461          .SBTTL  EXE$DUMPCPUREG - DUMP CPU-SPECIFIC IPR'S
                     00CE   462 ;+
                     00CE   463 ; DUMP CPU-SPECIFIC IPR'S INTO ERROR MESSAGE BUFFER.
                     00CE   464 ;
                     00CE   465 ; TWENTY-FOUR LONGWORDS ARE RESERVED IN THE EMB FOR CPU-SPECIFIC
                     00CE   466 ; IPR'S.  THE FORMATS FOR VARIOUS CPU'S ARE:
                     00CE   467 ;
                     00CE   468 ; 11/780:       11/750:         11/730:         11/790:                 uVAX I:
                     00CE   469 ;
                     00CE   470 ; ICR           ICR             ICR             ICR                     UNUSED(0)
                     00CE   471 ; TODR          TODR            TODR            TODR                    APPROX TODR
                     00CE   472 ; ACCS          ACCS            ACCS            ACCS                    UNUSED(0)
                     00CE   473 ; SBIFS         TBDR            21 UNUSED(0)    SBISTS  (1st SBI)       21 UNUSED(0)
                     00CE   474 ; SBISC         CADR                            SILOCMP      ''
                     00CE   475 ; SBIMT         MCESR                           MAINT        ''
                     00CE   476 ; SBIER         CAER                            SBIERR       ''
                     00CE   477 ; SBIS          CMIERR                          TMOADDRS     ''
                     00CE   478 ; 16 SBI SILO   16 UNUSED(0)                    16 SBI SILO  ''
                     00CE   479 ;
                     00CE   480 ; INPUTS:
                     00CE   481 ;
                     00CE   482 ;       R0 - ADDR IN EMB OF START OF CPU-SPECIFIC REGISTERS=
                     00CE   483 ;            OFFSET EMB$L_CR_CPUREG
                     00CE   484 ;
                     00CE   485 ; OUTPUTS:
                     00CE   486 ;
                     00CE   487 ;       R0,R1 DESTROYED
                     00CE   488 ;       ALL OTHER REGISTERS PRESERVED
                     00CE   489 ;-
                     00CE   490
                     00CE   491          .ENABL  LSB
                     00CE   492
                     00CE   493 EXE$DUMPCPUREG::                         ;SUBROUTINE ENTRY
                     00CE   494
                     00CE   495
80   1A   DB   D0   00CE   497          MFPR    #PR780$_ICR,(R0)+       ;LOG INTERVAL COUNT REG,
80   1B   DB   D0   00D1   498          MFPR    #PR780$_TODR,(R0)+      ; TIME-OF-DAY REG,
80   28   DB   D0   00D4   499          MFPR    #PR780$_ACCS,(R0)+      ; ACCELERATOR CONTROL REG,
80   30   DB   D0   00D7   500          MFPR    #PR780$_SBIFS,(R0)+     ; SBI FAULT REG,
80   32   DB   D0   00DA   501          MFPR    #PR780$_SBISC,(R0)+     ; SBI COMPARATOR REG
80   33   DB   D0   00DD   502          MFPR    #PR780$_SBIMT,(R0)+     ; SBI MAINT REG,
80   34   DB   D0   00E0   503          MFPR    #PR780$_SBIER,(R0)+     ; SBI ERROR REG,
80   31   DB   D0   00E3   504          MFPR    #PR780$_SBIS,(R0)+      ; SBI TIMEOUT REG.
51   10   D0        00E6   505          MOVL    #16,R1                  ;GET # SILO ENTRIES TO DUMP
80   31   DB   D0   00E9   506 10$:     MFPR    #PR780$_SBIS,(R0)+      ;DUMP SILO TO EMB
   FA 51   F5        00EC   507          SOBGTR  R1,10$                  ;
                     00EF   509
                     00EF   510
                     00EF   524
                     00EF   525
                     00EF   536
                     00EF   537
                     00EF   558
                     00EF   559
                     00EF   572 90$:
                05   00EF   573          RSB
                     00F0   574          .DISABLE LSB
```

```
                00F0    575
```

```
                     00F0   577          .SBTTL  EXE$READ_TODR (P) - READ TIME-OF-DAY CLOCK
                     00F0   578 ;+
                     00F0   579 ; READS THE TIME-OF-DAY CLOCK, SINCE IT MAY BE ACCESSED IN
                     00F0   580 ; DIFFERENT WAYS:  AS AN INTERNAL PROCESSOR REGISTER, AS PART
                     00F0   581 ; OF THE CONSOLE, OR BY READING AN ADDRESS IN I/O SPACE.  IT
                     00F0   582 ; MAY ALSO BE IN DIFFERENT FORMATS AND HAVE TO BE CONVERTED.
                     00F0   583 ;
                     00F0   584 ; INPUTS:
                     00F0   585 ;
                     00F0   586 ;       NONE.
                     00F0   587 ;
                     00F0   588 ; OUTPUTS:
                     00F0   589 ;
                     00F0   590 ;       R0 - TODR VALUE
                     00F0   591 ;       ALL OTHER REGISTERS PRESERVED
                     00F0   592 ;-
                     00F0   593
                     00F0   594 EXE$READP_TODR::                             ; SUBROUTINE ENTRY
                     00F0   595
                     00F0   596         ; NAUTILUS PROCESSOR NEEDS TO USE A SEPARATE ROUTINE TO ACCESS PHYSICAL TODR
                     00F0   597         ; REGISTER IN THE CONSOLE PROCESSOR FOR TWO REASONS.  FIRST, THE PHYSICAL
                     00F0   598         ; TODR HAS ONE SECOND RESOLUTION INSTEAD OF 10 MSEC RESOLUTION.  SECOND, A
                     00F0   599         ; REFERENCE TO THE PHYSICAL TODR IS A VERY SLOW, NON-INTERRUPTIBLE ACTION.
                     00F0   600         ; NON-PHYSICAL NAUTILUS TODR REFERENCES WILL USE THE EXE$READ_TODR ENTRY
                     00F0   601         ; WHICH WILL FABRICATE THE TIME FROM THE QUADWORD SYSTEM TIME.
                     00F0   602
                     00F0   603                                             ; NOT NAUTILUS - FALL THROUGH TO READ_TODR
                     00F0   604
                     00F0   605 EXE$READ_TODR::                              ; SUBROUTINE ENTRY
                     00F0   606
                     00F0   607
 50    1B    DB      00F0   609         MFPR    #PR780$_TODR,R0             ; TODR IS A PROCESSOR REGISTER.
                     00F3   611
                     00F3   612
                     00F3   616
                     00F3   617
                     00F3   621
                     00F3   622
                     00F3   626
                     00F3   662
           05        00F3   663         RSB
                     00F4   664
```

```
                        00F4   666          .SBTTL  EXE$WRITE_TODR (P) - WRITES TIME-OF-DAY CLOCK
                        00F4   667 ;+
                        00F4   668 ; WRITES THE TIME-OF-DAY CLOCK, SINCE IT MAY BE ACCESSED IN
                        00F4   669 ; DIFFERENT WAYS:  AS AN INTERNAL PROCESSOR REGISTER, AS PART
                        00F4   670 ; OF THE CONSOLE, OR BY READING AN ADDRESS IN I/O SPACE.  IT
                        00F4   671 ; MAY ALSO BE IN DIFFERENT FORMATS AND HAVE TO BE CONVERTED.
                        00F4   672 ;
                        00F4   673 ; INPUTS:
                        00F4   674 ;
                        00F4   675 ;       R0 - CONTAINS VALUE TO BE WRITTEN INTO TODR
                        00F4   676 ;
                        00F4   677 ; OUTPUTS:
                        00F4   678 ;
                        00F4   679 ;       NEW TIME VALUE WRITTEN INTO TODR.
                        00F4   680 ;       ALL REGISTERS PRESERVED.
                        00F4   681 ;-
                        00F4   682
                        00F4   683 EXE$WRITEP_TODR::                            ; SUBROUTINE ENTRY
                        00F4   684
                        00F4   685          ; NAUTILUS PROCESSOR NEEDS TO USE A SEPARATE ROUTINE TO ACCESS PHYSICAL TODR
                        00F4   686          ; REGISTER IN THE CONSOLE PROCESSOR FOR TWO REASONS.  FIRST, THE PHYSICAL
                        00F4   687          ; TODR HAS ONE SECOND RESOLUTION INSTEAD OF 10 MSEC RESOLUTION.  SECOND, A
                        00F4   688          ; REFERENCE TO THE PHYSICAL TODR IS A VERY SLOW, NON-INTERRUPTIBLE ACTION.
                        00F4   689          ; NON-PHYSICAL NAUTILUS TODR REFERENCES WILL USE THE EXE$WRITE_TODR ENTRY
                        00F4   690          ; WHICH WILL FABRICATE A NEW QUADWORD SYSTEM TIME.
                        00F4   691
                        00F4   692                                              ; NOT NAUTILUS - FALL THROUGH TO WRITE_TODR
                        00F4   693
                        00F4   694 EXE$WRITE_TODR::                             ; SUBROUTINE ENTRY
                        00F4   695
                        00F4   696
1B   50   DA            00F4   698          MTPR    R0,#PR780$_TODR             ; TODR IS A PROCESSOR REGISTER.
                        00F7   700
                        00F7   701
                        00F7   705
                        00F7   706
                        00F7   710
                        00F7   711
                        00F7   715
                        00F7   716
                        00F7   721
              05        00F7   722          RSB
```

```
                                                H 10
                     00F8   724          .SBTTL  EXE$REGSAVE - SAVE CPU-SPECIFIC IPR'S
                     00F8   725 ;+
                     00F8   726 ; EXE$REGSAVE - CALLED BY POWERFAIL TO SAVE CPU-SPECIFIC IPR'S ON
                     00F8   727 ;               THE STACK
                     00F8   728 ;
                     00F8   729 ; INPUTS: NONE
                     00F8   730 ;
                     00F8   731 ; OUTPUTS:
                     00F8   732 ;
                     00F8   733 ;               R0 DESTROYED
                     00F8   734 ;               OTHER GENERAL REGISTERS PRESERVED
                     00F8   735 ;               IPR'S SAVED ON THE STACK AS FOLLOWS:
                     00F8   736 ;
                     00F8   737 ;         11/780:      11/750:          11/730:          11/790:    uVAX I:
                     00F8   738 ;
                     00F8   739 ; 0(SP)    PME           PME              PME              ACCS       (none)
                     00F8   740 ; 4(SP)    SBIMT         TBDR                              CSWP
                     00F8   741 ; 8(SP)                  CADR                              PME
                     00F8   742 ;
                     00F8   743 ;-
                     00F8   744
                     00F8   745          .ENABL  LSB
                     00F8   746
                     00F8   747 EXE$REGSAVE::                                ;SUBROUTINE ENTRY
          01   BA    00F8   749          POPR    #^M<R0>                     ;CLEAR RETURN FROM STACK
                     00FA   750
                     00FA   751
       7E    3D DB   00FA   753          MFPR    #PR780$_PME,-(SP)           ;SAVE PERFORMANCE MONITOR ENABLE
       7E    33 DB   00FD   754          MFPR    #PR780$_SBIMT,-(SP)         ;SAVE SBI MAINT REG
                     0100   756
                     0100   757
                     0100   763
                     0100   764
                     0100   768
                     0100   769
                     0100   776
       60    17      0100   777          JMP     (R0)                        ;DONE, RETURN
                     0102   779
                     0102   783
                     0102   784          .DSABL  LSB
```

```
                    0102   786          .SBTTL  EXE$REGRESTOR - RESTORE CPU-SPECIFIC IPR'S
                    0102   787  ;+
                    0102   788  ; EXE$REGRESTOR - CALLED BY POWERFAIL RECOVERY TO RESTORE CPU-SPECIFIC
                    0102   789  ;               IPR'S FROM THE STACK.
                    0102   790  ;
                    0102   791  ; INPUTS:
                    0102   792  ;
                    0102   793  ;                       R6 - TOP OF STACK
                    0102   794  ;                       STACK SET UP AS DEFINED IN OUTPUTS OF EXE$REGSAVE.
                    0102   795  ;
                    0102   796  ; OUTPUTS:
                    0102   797  ;
                    0102   798  ;                       R0 DESTROYED
                    0102   799  ;                       OTHER GENERAL REGISTERS PRESERVED
                    0102   800  ;                       CPU-SPECIFIC IPR'S RESTORED FROM STACK
                    0102   801  ;                       R6 - ADDRESS OF 1ST CPU-INDEPENDENT SAVED IPR
                    0102   802  ;
                    0102   803  ;-
                    0102   804
                    0102   805          .ENABL  LSB
                    0102   806
                    0102   807 EXE$REGRESTOR::                                 ;SUBROUTINE ENTRY
         01    BA   0102   809          POPR    #^M<R0>                         ;CLEAR RETURN FROM STACK
                    0104   810
                    0104   811
    33   86    DA   0104   813          MTPR    (R6)+,#PR780$_SBIMT             ;RESTORE SBI MAINT REGISTER
    3D   86    DA   0107   814          MTPR    (R6)+,#PR780$_PME               ;RESTORE PERFORMANCE MONITOR ENABLE
                    010A   816
                    010A   817
                    010A   823
                    010A   824
                    010A   828
                    010A   829
                    010A   837
         60    17   010A   838          JMP     (R0)                            ;DONE, RETURN
                    010C   843
                    010C   844          .DSABL  LSB
```

```
                                010C   846          .SBTTL  EXE$INIPROCREG - CPU-DEPENDENT INITIALIZATION OF IPR'S
                                010C   847 ;+
                                010C   848 ; EXE$INIPROCREG - PERFORM INITIALIZATION OF INTERVAL TIMER AND
                                010C   849 ;       CPU-DEPENDENT REGISTERS. CALLED FROM INIT AND POWERFAIL.
                                010C   850 ;
                                010C   851 ; INPUTS:
                                010C   852 ;
                                010C   853 ;       NONE
                                010C   854 ;
                                010C   855 ; OUTPUTS:
                                010C   856 ;
                                010C   857 ;       NONE
                                010C   858 ;-
                                010C   859
                                010C   860 EXE$INIPROCREG::                             ; INIT PROCESSOR REGISTERS
                                010C   861
                                010C   864                                             ; FOR 11/780, CONTINUE
                                010C   865                                             ;
                     00'  E1    010C   866          BBC     S^#EXE$V_CRDENABL,-
  07 00000000'9F                010E   867                  @#EXE$GL_FLAGS,10$         ; IF CLR, IGNORE CRD ERRORS
34   0000C000 8F    DA  0114   868          MTPR    #<3@14>,S^#PR780$_SBIER ; SET CRD INTERRUPT ENABLE
                     00'  E1    011B   869 10$:     BBC     S^#EXE$V_SBIERR,-
  07 00000000'9F                011D   870                  @#EXE$GL_FLAGS,20$         ; IF CLR, IGNORE SBI ERRORS
30   00040000 8F    DA  0123   871          MTPR    #<1@18>,S^#PR780$_SBIFS ; SET SBI FAULT ENABLE
                                012A   872
                     00'  E0    012A   925 20$:     BBS     S^#EXE$V_NOCLOCK,-
  0E 00000000'9F                012C   926                  @#EXE$GL_FLAGS,30$         ; BRANCH IF NOT USING CLOCK
                                0132   927
                                0132   931
19   FFFFD8F0 8F    DA  0132   933          MTPR    #-<10*1000>,S^#PR780$_NICR ; LOAD NEXT INTERVAL REGISTER
                                0139   935
                                0139   939
                                0139   943
18   800000D1 8F    DA  0139   944          MTPR    #^X800000D1,S^#PR$_ICCS ; CLEAR ERROR AND START CLOCK
                     05  0140   945 30$:     RSB                                        ; AND RETURN
                                0141   946
                                0141   962
```

```
                0141    985          .SBTTL  SYSL$CLRSBIA
                0141    986 ;++
                0141    987 ; SYSL$CLRSBIA  - ON 11/790, CLEAR SBIA ERROR REGISTERS
                0141    988 ;               - ON 11/780, 11/750, 11/730, AND MICRO-VAX I, THIS IS A NOP
                0141    989 ;
                0141    990 ; THIS ROUTINE IS CALLED TO CLEAR OUT SBIA ERROR BITS AFTER A MACHINE CHECK
                0141    991 ; OCCURS (WHEN MACHINE CHECK IS HANDLED LOCALLY).
                0141    992 ;
                0141    993 ; THIS ROUTINE SHOULD BE CALLED AT IPL 31.
                0141    994 ;
                0141    995 ; INPUTS:
                0141    996 ;       ABUS_TYPE       - AN ARRAY TYPE CODES; IDENTIFIES EACH ADAPTER ON THE
                0141    997 ;                         ABUS.
                0141    998 ;       ABUS_VA         - AN ARRAY OF ADAPTER SPACE VA'S FOR EACH ADAPTER
                0141    999 ;                         ON THE ABUS.
                0141   1000 ;
                0141   1001 ; OUTPUTS:
                0141   1002 ;       SBI ERROR BITS ARE CLEARED FOR EACH SBIA ON THE ABUS.
                0141   1003 ;       ALL REGISTERS PRESERVED.
                0141   1004 ;--
                0141   1005 SYSL$CLRSBIA::
05              0141   1023          RSB                                     ; AND RETURN
```

```
                                0142  1025          .SBTTL  EXE$TEST_CSR
                                0142  1026 ;+
                                0142  1031 ; EXE$TEST_CSR - TEST A UNIBUS CONTROLLER CSR FOR EXISTENCE
                                0142  1033 ;
                                0142  1034 ; THIS TEST IS CPU-DEPENDENT.  THE FOLLOWING CPU'S ARE SUPPORTED:
                                0142  1035 ;
                                0142  1036 ;       11/780  -TEST CSR AND CHECK RESULT IN THE UBA STATUS REGISTER.
                                0142  1037 ;       11/750  -NON-EXISTENT CSR IS REPORTED VIA MACHINE CHECK AS A
                                0142  1038 ;                NON-EXISTENT MEMORY REFERENCE.  CONNECT A TEMPORARY
                                0142  1039 ;                MACHINE CHECK HANDLER, TEST THE CSR, AND RESTORE THE
                                0142  1040 ;                ORIGINAL MACHINE CHECK HANDLER.
                                0142  1041 ;       11/730  -ACTION IS THE SAME AS FOR THE 11/750.
                                0142  1042 ;       11/790  -ACTION IS THE SAME AS FOR THE 11/780.
                                0142  1043 ;       MICRO-VAX I -ACTION IS SAME AS FOR THE 11/750.
                                0142  1044 ;
                                0142  1045 ; THIS SUBROUTINE SHOULD BE CALLED VIA BRANCH OR JUMP TO SUBROUTINE AT IPL 31.
                                0142  1046 ;
                                0142  1047 ; INPUTS:
                                0142  1048 ;
                                0142  1049 ;       R0 = CSR ADDRESS
                                0142  1050 ;       R6 = ADAPTER CONFIGURATION REGISTER ADDRESS
                                0142  1051 ;
                                0142  1052 ; OUTPUTS:
                                0142  1053 ;
                                0142  1054 ;       R0 LOW BIT SET/CLEAR FOR EXISTENT/NONEX CSR
                                0142  1055 ;       OTHER REGISTERS PRESERVED.
                                0142  1056 ;-
                                0142  1057
                                0142  1058          .ENABL  LSB
                                0142  1059
                                0142  1060 EXE$TEST_CSR::                          ;SUBROUTINE ENTRY
                                0142  1061
                        06   BB 0142  1062          PUSHR   #^M<R1,R2>              ;SAVE REGISTERS
                                0144  1063
                                0144  1065 ;
                                0144  1066 ; This next line of code is present so that this routine continues to function
                                0144  1067 ; correctly when the UNIBUS adapter is powered down.  Moving 0 into the UBA
                                0144  1068 ; Status Register has no effect when addressing the actual adapter register,
                                0144  1069 ; and clears out any garbage bits in memory when UNIBUS space is re-mapped to
                                0144  1070 ; the "black hole" page.
                                0144  1071 ;
                    08 A6   00  D0 0144  1072          MOVL    #0,UBA$L_SR(R6)         ;WHEN UBA IS REMAPPED
                51  00000000'GF  D0 0148  1073          MOVL    G^EXE$GL_SCB,R1         ;GET SCB ADDRESS
                          04 A1 DD 014F  1074          PUSHL   4(R1)                   ;SAVE CURRENT MCHECK HANDLER ADDR
                       52    5E D0 0152  1075          MOVL    SP,R2                   ;MARK CURRENT STACK POSITION
                04 A1    68'AF DE 0155  1077          MOVAL   B^MCHK_780,4(R1)        ;CONNECT TEMP 11/780 MCHECK HANDLER
                          60    B5 015A  1082          TSTW    (R0)                    ;ATTEMPT TO READ CSR
                08 A6    08 A6 D0 015C  1083          MOVL    UBA$L_SR(R6),UBA$L_SR(R6) ;CLEAR AND CHECK FOR ERROR
                          1B    12 0161  1084          BNEQ    NONEX_DEV               ;BRANCH IF ERROR
                       50    01 9A 0163  1085 OK:      MOVZBL  #SS$_NORMAL,R0          ;SET STATUS TO SUCCESS
                          18    11 0166  1086          BRB     TEST_DONE               ;JOIN COMMON EXIT
                                0168  1087
                                0168  1089 ;
                                0168  1090 ; TEMPORARY CSR TEST MACHINE CHECK HANDLER FOR THE 11/780:
                                0168  1091 ;
                                0168  1092
                                0168  1093          .ALIGN  LONG
```

```
                        0168  1094 MCHK_780:
    7E    30    DB      0168  1095          MFPR    S^#PR780$_SBIFS,-(SP)   ;GET SBI FAULT STATUS REGISTER
00  6E    19    E5      016B  1096          BBCC    #25,(SP),TO$            ;CLEAR ERROR 1ST PASS BIT
    30    8E    DA      016F  1097 10$:     MTPR    (SP)+,S^#PR780$_SBIFS   ;WRITE BACK TO CLEAR THE FAULT
50      04  AE    D0      0172  1098          MOVL    4(SP),R0                ;PICK UP SUMMARY PARAMETER
    5E    52    D0      0176  1099          MOVL    R2,SP                   ;CLEAR MCHECK FRAME OFF STACK
    05    50    D1      0179  1100          CMPL    R0,#5                   ;IS IT READ DATA SUBSTITUTE?
          E5    13      017C  1101          BEQL    OK                      ;YES, THEN IT IS READ W/BAD PARITY
                        017E  1103
                        017E  1118
                        017E  1188 NONEX_DEV:                               ;
          50    D4      017E  1189          CLRL    R0                      ;SET STATUS TO FAILURE
                        0180  1190 TEST_DONE:
       04 A1  8ED0    0180  1191          POPL    4(R1)                   ;RESTORE SYSTEM MCHECK HANDLER
                        0184  1192 TEST_DONE_2:
          06    BA      0184  1193          POPR    #^M<R1,R2>              ;RESTORE REGISTERS
                05      0186  1194          RSB                             ;RETURN RESULT TO CALLER
                        0187  1195          .DISABLE LSB
```

```
                              0187          .SBTTL  ADPLINK - LINK ADAPTER CONTROL BLOCK INTO ADP LIST
                              0187     ;+
                              0187     ; ADPLINK LINKS THE ADAPTER CONTROL BLOCK TO THE END OF THE ADP LIST
                              0187     ;
                              0187     ; INPUT:
                              0187     ;        R2 - ADDRESS OF NEW ADP
                              0187     ; OUTPUTS:
                              0187     ;        ADP IS LINK TO THE END OF THE ADPLIST LOCATED BY IOC$GL_ADPLIST.
                              0187     ;        R0,R1 destroyed.
                              0187     ;-
                              0187
                              0187     ADPLINK::
        50   FFFFFFFC'9F  9E  0187              MOVAB   @#<IOC$GL_ADPLIST-ADP$L_LINK>,R0
                              018E                                                  ; START OF LIST
          51     04 A0    D0  018E     10$:     MOVL    ADP$L_LINK(R0),R1            ; FLINK TO FIRST ENTRY
                     05   13  0192              BEQL    20$                          ; AT END
             50      51   D0  0194              MOVL    R1,R0                        ; TRY AGAIN
                     F5   11  0197              BRB     10$                          ;
          04 A0      52   D0  0199     20$:     MOVL    R2,ADP$L_LINK(R0)            ; CHAIN NEW ADP TO END OF LIST
                          05  019D              RSB                                  ; AND RETURN
                              019E
                              019E              .END
```

```
ADP$L_CSR                       = 00000000
ADP$L_LINK                      = 00000004
ADP$W_ADPTYPE                   = 0000000E
ADPLINK                           00000187  RG    03
ADP_TBL_DWN                       0000009E  R     03
ADP_TBL_UP                        000000B6  R     03
BQO$L_UMR_DIS                   = 00000024
BQO$W_VERSION                   = 00000010
BTD$K_CONSOLE                   = 00000040
C750_LIKE                       = 00000000
C780_LIKE                       = 00000001
CI$SHUTDOWN                       ********    X   03
CPU_TYPE                        = 00000001
EXE$DUMPCPUREG                    000000CE  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$INIBOOTADP                    00000000  RG    03
EXE$INIPROCREG                    0000010C  RG    03
EXE$READP_TODR                    000000F0  RG    03
EXE$READ_TODR                     000000F0  RG    03
EXE$REGRESTOR                     00000102  RG    03
EXE$REGSAVE                       000000F8  RG    03
EXE$SHUTDWNADP                    00000077  RG    03
EXE$STARTUPADP                    0000006F  RG    03
EXE$TEST_CSR                      00000142  RG    03
EXE$V_CRDENABL                    ********    X   03
EXE$V_NOCLOCK                     ********    X   03
EXE$V_SBIERR                      ********    X   03
EXE$WRITEP_TODR                   000000F4  RG    03
EXE$WRITE_TODR                    000000F4  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$M_CR_INIT                   = 00000001
MCHK_780                          00000168  R     03
NDT$_CI                         = 00000038
NDT$_MB                         = 00000020
NONEX_DEV                         0000017E  R     03
OK                                00000163  R     03
PR$_ICCS                        = 00000018
PR$_SID_TYP730                  = 00000003
PR$_SID_TYP750                  = 00000002
PR$_SID_TYP780                  = 00000001
PR$_SID_TYP790                  = 00000004
PR$_SID_TYPUV1                  = 00000007
PR780$_ACCS                     = 00000028
PR780$_ICR                      = 0000001A
PR780$_NICR                     = 00000019
PR780$_PME                      = 0000003D
PR780$_SBIER                    = 00000034
PR780$_SBIFS                    = 00000030
PR780$_SBIMT                    = 00000033
PR780$_SBIS                     = 00000031
PR780$_SBISC                    = 00000032
PR780$_TODR                     = 0000001B
RPB$B_DEVTYP                    = 00000066
RPB$L_ADPVIR                    = 00000060
RPB$L_IOVEC                     = 00000034
RPB$W_BOOTNDT                   = 000000A1
SS$_NORMAL                      = 00000001
SYS[$CLRSBIA                      00000141  RG    03
TEST_DONE                         00000180  R     03
TEST_DONE_2                       00000184  R     03
UBA$INITIAL                       ********    X   03
UBA$L_CR                        = 00000004
UBA$L_CSR                       = 00000000
UBA$L_MAP                       = 00000800
UBA$L_SR                        = 00000008
UBA$M_CR_INIT                   = 00000001
UBA$M_CSR_UBIC                  = 00010000
```

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	0000019E (414.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	36	00:00:00.04	00:00:00.83
Command processing	131	00:00:00.43	00:00:04.04
Pass 1	343	00:00:07.55	00:00:24.25
Symbol table sort	0	00:00:01.06	00:00:02.56
Pass 2	133	00:00:01.99	00:00:06.21
Symbol table output	11	00:00:00.06	00:00:00.47
Psect synopsis output	3	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	659	00:00:11.16	00:00:38.38

The working set limit was 1650 pages.
70789 bytes (139 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 1058 non-local and 18 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

1123 GETS were required to define 16 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:ERRSUB780/OBJ=OBJ$:ERRSUB780 MSRC$:CPUSW780/UPDATE=(ENH$:CPUSW780)+MSRC$:ERRSUB/UPDATE=(ENH$:ERRSUB)+EXECML$/LIB

ERRSUBUV1
LIS

ERRSUB790
LIS

ERRSUB780
LIS

ERRSUB750
LIS

FALMAN
LIS

ERRSUB730
LIS