```
SSSSSSSSSSS   YYY         YYY    SSSSSSSSSSSS   LLL                    000000000         AAAAAAAAA
SSSSSSSSSSS   YYY         YYY    SSSSSSSSSSSS   LLL                   000000000         AAAAAAAAAA
SSSSSSSSSSS   YYY         YYY    SSSSSSSSSSSS   LLL                  000000000         AAAAAAAAAA
SSS            YYY         YYY    SSS            LLL              000        000      AAA        AAA
SSS             YYY       YYY     SSS            LLL              000        000      AAA        AAA
SSS              YYY     YYY      SSS            LLL              000        000      AAA        AAA
SSS               YYY   YYY       SSS            LLL              000        000      AAA        AAA
SSS                YYY YYY        SSS            LLL              000        000      AAA        AAA
     SSSSSSSS         YYY              SSSSSSSS  LLL              000        000      AAA        AAA
     SSSSSSSS         YYY              SSSSSSSS  LLL              000        000      AAA        AAA
     SSSSSSSS         YYY              SSSSSSSS  LLL              000        000      AAA        AAA
            SSS       YYY                    SSS LLL              000        000      AAAAAAAAAAAAAAAA
            SSS       YYY                    SSS LLL              000        000      AAAAAAAAAAAAAAAA
            SSS       YYY                    SSS LLL              000        000      AAA        AAA
            SSS       YYY                    SSS LLL              000        000      AAA        AAA
            SSS       YYY                    SSS LLL              000        000      AAA        AAA
SSSSSSSSSSS   YYY         SSSSSSSSSSSS  LLLLLLLLLLLLLLL    000000000         AAA        AAA
SSSSSSSSSSS   YYY         SSSSSSSSSSSS  LLLLLLLLLLLLLLL    000000000         AAA        AAA
SSSSSSSSSSS   YYY         SSSSSSSSSSSS  LLLLLLLLLLLLLLL    000000000         AAA        AAA
```

ERRSUB730
LIS

```
0000      1              .NOSHOW CONDITIONALS
0000      5
0000      9
0000     11              .TITLE  ERRSUB730 - ERROR SUBROUTINES FOR VAX 11/730
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      199
                 0000      203
                 0000      205          $PR730DEF                               ;DEFINE 11/730 INTERNAL PROCESSOR REGS
                 0000      207
                 0000      211  ;
                 0000      212  ; EQUATED SYMBOLS:
                 0000      213  ;
                 0000      218
                 0000      223
        00000000 0000      225          C780_LIKE = 0
        00000001 0000      226          C750_LIKE = 1
                 0000      228
                 0000      233
                 0000      238
                 0000      239
                 0000      240  ; Define labels for two "extra" routines.  This reserves some vectors from
                 0000      241  ; SYS.EXE into SYSLOAxxx.EXE that can be patched if another routine must
                 0000      242  ; be added in between major releases.
                 0000      243  ;
                 0000      244  EXE$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$INIBOOTADP::                          ;SUBROUTINE ENTRY
                        0000   276
          66 A6   91    0000   278              CMPB    RPB$B_DEVTYP(R6),-    ;IS BOOT DEVICE THE CONSOLE
          40 8F         0003   279                      #BTD$K_CONSOLE       ;BLOCK STORAGE DEVICE?
             2B   13    0005   280              BEQL    40$                  ;YES, RETURN
       50 60 A6   D0    0007   281              MOVL    RPB$L_ADPVIR(R6),R0  ;GET ADDR OF ADAPTER REG SPACE
                        000B   282
                        000B   327
                        000B   331
                        000B   332  INI_UBADP:                                ;INIT UBA
                        000B   333
                        000B   341
                        000B   343
                        000B   347
          37 00   DA    000B   349              MTPR    #0,#PR730$_UBRESET   ;INIT UBI AND UNIBUS
                        000E   351
                        000E   356
                        000E   358  ;
                        000E   360  ;
                        000E   361  ; CHECK THE VMB VERSION NUMBER.  IF IT EXISTS AND IF IT IS 7 OR GREATER, THEN
                        000E   362  ; SEE IF ANY UNIBUS MAP REGISTERS TO DISABLE.
                        000E   363  ;
                        000E   364
       52 34 A6   D0    000E   365              MOVL    RPB$L_IOVEC(R6),R2       ;PICK UP THE IOVECTOR FROM RPB
       51 10 A2   B2    0012   366              MCOMW   BQO$W_VERSION(R2),R1    ;GET VMB VERSION NUMBER 1'S COMPLEMENTED
       12 A2 51   B1    0016   367              CMPW    R1,BQO$W_VERSION+2(R2)  ;CHECK AGAINST CHECK WORD IN VMB
             16   12    001A   368              BNEQ    40$                     ;IF NOT, ASSUME NO VERSION NUMBER
       07 10 A2   B1    001C   369              CMPW    BQO$W_VERSION(R2),#7    ;VERSION 7 OR GREATER OF VMB?
             10   1F    0020   370              BLSSU   40$                     ;NO, DON'T BOTH WITH UMR'S
       52 24 A2   D0    0022   371              MOVL    BQO$L_UMR_DIS(R2),R2    ;GRAB THE NUMBER OF UMR'S TO DISABLE
             0A   13    0026   372              BEQL    40$                     ;NONE, LEAVE
                        0028   373
                        0028   377
                        0028   378
                        0028   379  ; THIS CODE IS EXECUTED FOR ALL PROCESSORS.  ITS DISABLES ANY UNIBUS MAP
                        0028   380  ; REGISTERS ASSOCIATED WITH UNIBUS MEMORY TO PREVENT CONTENTION BETWEEN
                        0028   381  ; SBI AND UNIBUS ADDRESSES.
                        0028   382  ;
                        0028   383
```

```
        51   0800 C0   DE   0028   384  30$:   MOVAL   UBA$L_MAP(R0),R1        ;ADDRESS OF FIRST REGISTER
                  81   D4   002D   385         CLRL    (R1)+                  ;DISABLE IT
             FB 52   F5   002F   386         SOBGTR  R2,30$                 ;LOOP UNTIL ALL DONE
                  05        0032   388  40$:   RSB                            ;DONE WITH UBA INIT
                            0033   389         .DISABLE LSB
```

ERRSUB730
V04-002

I 6

- ERROR SUBROUTINES FOR VAX 11/730   16-SEP-1984 00:54:20 VAX/VMS Macro V04-00   Page   7
EXE$SHUTDWNADP - SHUTDOWN ANY ADAPTERS D 13-SEP-1984 15:49:22 [SYSLOA.SRC]ERRSUB.MAR;5   (5)

```
                          0033    391              .SBTTL  EXE$SHUTDWNADP - SHUTDOWN ANY ADAPTERS DURING BUGCHECK
                          0033    392              .SBTTL  EXE$STARTUPADP - STARTUP ANY ADAPTERS
                          0033    393      ;+
                          0033    394      ;  EXE$SHUTDWNADP - SHUTDOWN ANY ADAPTERS DURING BUGCHECK
                          0033    395      ;       THIS ROUTINE IS CALLED FROM BUGCHECK BEFORE THE DUMP IS TAKEN TO
                          0033    396      ;       ENSURE THAT ALL ADAPTERS THAT NEED TO BE QUIESENT ARE.
                          0033    397      ;
                          0033    398      ; INPUTS:
                          0033    399      ;
                          0033    400      ;       IPL = 31
                          0033    401      ;
                          0033    402      ; OUTPUTS:
                          0033    403      ;
                          0033    404      ;       OTHER REGISTERS PRESERVED
                          0033    405      ;-
                          0033    406              .ENABLE LSB
                          0033    407
                          0033    408      EXE$STARTUPADP::
            17     BB     0033    409              PUSHR   #^M<R0,R1,R2,R4>          ; Save a register
      51   7A'AF   DE     0035    410              MOVAL   B^ADP_TBL_UP,R1          ; Address of startup table
            06     11     0039    411              BRB     5$                       ; Join common code
                          003B    412
                          003B    413      EXE$SHUTDWNADP::
            17     BB     003B    414              PUSHR   #^M<R0,R1,R2,R4>          ; Save a register
      51   62'AF   DE     003D    415              MOVAL   B^ADP_TBL_DWN,R1         ; Address of shutdown table
  FFFFFFFC'9F       DE     0041    416      5$:     MOVAL   @#<IOC$GL_ADPLIST- -
                  52             0047    417                      ADP$L_LINK>,R2   ; Get pointer to head of adapter list
      52   04 A2   D0     0048    418      10$:    MOVL    ADP$L_LINK(R2),R2        ; Flink onward
            11     13     004C    419              BEQL    20$                      ; Branch if at end of list
      54   62     D0     004E    420              MOVL    ADP$L_CSR(R2),R4         ; Get address of CSR
  50   0E A2   3C     0051    421              MOVZWL  ADP$W_ADPTYPE(R2),R0     ; Get adapter type code
  50   6140   DE     0055    422              MOVAL   (R1)[R0],R0              ; Get table entry of adap shutdown
      00 B040   16     0059    423              JSB     @(R0)[R0]                ; Call adapter shutdown
            E9     11     005D    424              BRB     10$                      ; Next adapter
                          005F    425
            17     BA     005F    426      20$:    POPR    #^M<R0,R1,R2,R4>
            05           0061    427      30$:    RSB
                          0062    428
                          0062    429      ;
                          0062    430      ; Table of addresses of adapter shutdown routines ordered
                          0062    431      ; by adapter type in ADP$W_ADPTYPE.
                          0062    432      ;
                          0062    433
                          0062    434      ADP_TBL_DWN:                             ; Address table start
  FFFFFFFF   0062    435              .LONG   30$-.                    ; 0-MBA
  FFFFFFFB   0066    439              .LONG   30$-.                    ; 1-UBA
  FFFFFFF7   006A    441              .LONG   30$-.                    ; 2-DR32
  FFFFFFF3   006E    442              .LONG   30$-.                    ; 3-MA780
  FFFFFF8E'  0072    443              .LONG   CI$SHUTDOWN-.            ; 4-CI
  FFFFFFEB   0076    444              .LONG   30$-.                    ; Rsvrd for future expansion
                          007A    445
                          007A    446      ;
                          007A    447      ; Table of addresses of adapter startup routines ordered
                          007A    448      ; by adapter type in ADP$W_ADPTYPE.
                          007A    449      ;
                          007A    450
                          007A    451      ADP_TBL_UP:                              ; Address table start
```

```
FFFFFF86'  007A   452        .LONG    MBA$INITIAL-.       ; 0-MBA
FFFFFF82'  007E   453        .LONG    UBA$INITIAL-.       ; 1-UBA
FFFFFFDF   0082   454        .LONG    30$-.               ; 2-DR32
FFFFFF7A'  0086   455        .LONG    MA$INITIAL-.        ; 3-MA780
FFFFFFD7   008A   456        .LONG    30$-.               ; 4-CI
FFFFFFD3   008E   457        .LONG    30$-.               ; Rsvrd for future expansion
           0092   458
           0092   459        .DISABLE LSB
```

ERRSUB730
V04-002

K 6
- ERROR SUBROUTINES FOR VAX 11/730   16-SEP-1984 00:54:20   VAX/VMS Macro V04-00   Page   9
EXE$DUMPCPUREG - DUMP CPU-SPECIFIC IPR'S 13-SEP-1984 15:49:22 [SYSLOA.SRC]ERRSUB.MAR;5   (6)

```
              0092   461              .SBTTL  EXE$DUMPCPUREG - DUMP CPU-SPECIFIC IPR'S
              0092   462      ;+
              0092   463      ; DUMP CPU-SPECIFIC IPR'S INTO ERROR MESSAGE BUFFER.
              0092   464      ;
              0092   465      ; TWENTY-FOUR LONGWORDS ARE RESERVED IN THE EMB FOR CPU-SPECIFIC
              0092   466      ; IPR'S.  THE FORMATS FOR VARIOUS CPU'S ARE:
              0092   467      ;
              0092   468      ; 11/780:          11/750:          11/730:          11/790:          uVAX I:
              0092   469      ;
              0092   470      ; ICR              ICR              ICR              ICR              UNUSED(0)
              0092   471      ; TODR             TODR             TODR             TODR             APPROX TODR
              0092   472      ; ACCS             ACCS             ACCS             ACCS             UNUSED(0)
              0092   473      ; SBIFS            TBDR             21 UNUSED(0)     SBISTS  (1st SBI) 21 UNUSED(0)
              0092   474      ; SBISC            CADR                              SILOCMP      ''
              0092   475      ; SBIMT            MCESR                             MAINT        ''
              0092   476      ; SBIER            CAER                              SBIERR       ''
              0092   477      ; SBIS             CMIERR                            TMOADDRS     ''
              0092   478      ; 16 SBI SILO      16 UNUSED(0)                      16 SBI SILO  ''
              0092   479      ;
              0092   480      ; INPUTS:
              0092   481      ;
              0092   482      ;     R0 - ADDR IN EMB OF START OF CPU-SPECIFIC REGISTERS=
              0092   483      ;          OFFSET EMB$L_CR_CPUREG
              0092   484      ;
              0092   485      ; OUTPUTS:
              0092   486      ;
              0092   487      ;     R0,R1 DESTROYED
              0092   488      ;     ALL OTHER REGISTERS PRESERVED
              0092   489      ;-
              0092   490
              0092   491              .ENABL  LSB
              0092   492
              0092   493      EXE$DUMPCPUREG::                              ;SUBROUTINE ENTRY
              0092   494
              0092   495
              0092   509
              0092   510
              0092   524
              0092   525
80  1A  DB    0092   527              MFPR    #PR730$_ICR,(R0)+            ;LOG INTERVAL COUNT REG,
80  1B  DB    0095   528              MFPR    #PR730$_TODR,(R0)+           ;    TIME-OF-DAY REG,
80  28  DB    0098   529              MFPR    #PR730$_ACCS,(R0)+           ;    ACCELERATOR CONTROL REG.
51  15  D0    009B   530              MOVL    #<<EMB$L_CR_CODE - EMB$L_CR_CPUREG>/4>-3, R1 ;-3 FOR ICR,
              009E   531                                                   ;  TODR, AND ACCS ALREADY LOGGED.
              009E   532                                                   ;THERE ARE NO OTHER CPU-SPECIFIC
    80  D4    009E   533      10$:     CLRL    (R0)+                       ;  REGISTERS TO LOG, SO ZERO THE
FB  51  F5    00A0   534               SOBGTR  R1, 10$                     ;  SPACE IN THE ERROR MSG BUFFER
              00A3   536
              00A3   537
              00A3   558
              00A3   559
              00A3   572      90$:
        05    00A3   573               RSB
              00A4   574              .DISABLE LSB
              00A4   575
```

```
                    00A4   577                 .SBTTL  EXE$READ_TODR (P) - READ TIME-OF-DAY CLOCK
                    00A4   578 ;+
                    00A4   579 ; READS THE TIME-OF-DAY CLOCK, SINCE IT MAY BE ACCESSED IN
                    00A4   580 ; DIFFERENT WAYS:  AS AN INTERNAL PROCESSOR REGISTER, AS PART
                    00A4   581 ; OF THE CONSOLE, OR BY READING AN ADDRESS IN I/O SPACE.  IT
                    00A4   582 ; MAY ALSO BE IN DIFFERENT FORMATS AND HAVE TO BE CONVERTED.
                    00A4   583 ;
                    00A4   584 ; INPUTS:
                    00A4   585 ;
                    00A4   586 ;     NONE.
                    00A4   587 ;
                    00A4   588 ; OUTPUTS:
                    00A4   589 ;
                    00A4   590 ;     R0 - TODR VALUE
                    00A4   591 ;     ALL OTHER REGISTERS PRESERVED
                    00A4   592 ;-
                    00A4   593
                    00A4   594 EXE$READP_TODR::                                 ; SUBROUTINE ENTRY
                    00A4   595
                    00A4   596         ; NAUTILUS PROCESSOR NEEDS TO USE A SEPARATE ROUTINE TO ACCESS PHYSICAL TODR
                    00A4   597         ; REGISTER IN THE CONSOLE PROCESSOR FOR TWO REASONS.  FIRST, THE PHYSICAL
                    00A4   598         ; TODR HAS ONE SECOND RESOLUTION INSTEAD OF 10 MSEC RESOLUTION.  SECOND, A
                    00A4   599         ; REFERENCE TO THE PHYSICAL TODR IS A VERY SLOW, NON-INTERRUPTIBLE ACTION.
                    00A4   600         ; NON-PHYSICAL NAUTILUS TODR REFERENCES WILL USE THE EXE$READ_TODR ENTRY
                    00A4   601         ; WHICH WILL FABRICATE THE TIME FROM THE QUADWORD SYSTEM TIME.
                    00A4   602
                    00A4   603                                                  ; NOT NAUTILUS - FALL THROUGH TO READ_TODR
                    00A4   604
                    00A4   605 EXE$READ_TODR::                                  ; SUBROUTINE ENTRY
                    00A4   606
                    00A4   607
                    00A4   611
                    00A4   612
                    00A4   616
                    00A4   617
         50  1B  DB 00A4   619         MFPR    #PR730$_TODR,R0          ; TODR IS A PROCESSOR REGISTER.
                    00A7   621
                    00A7   622
                    00A7   626
                    00A7   662
             05     00A7   663         RSB
                    00A8   664
```

```
              00A8    666              .SBTTL  EXE$WRITE_TODR (P) - WRITES TIME-OF-DAY CLOCK
              00A8    667      ;+
              00A8    668      ; WRITES THE TIME-OF-DAY CLOCK, SINCE IT MAY BE ACCESSED IN
              00A8    669      ; DIFFERENT WAYS:  AS AN INTERNAL PROCESSOR REGISTER, AS PART
              00A8    670      ; OF THE CONSOLE, OR BY READING AN ADDRESS IN I/O SPACE.  IT
              00A8    671      ; MAY ALSO BE IN DIFFERENT FORMATS AND HAVE TO BE CONVERTED.
              00A8    672      ;
              00A8    673      ; INPUTS:
              00A8    674      ;
              00A8    675      ;       R0 - CONTAINS VALUE TO BE WRITTEN INTO TODR
              00A8    676      ;
              00A8    677      ; OUTPUTS:
              00A8    678      ;
              00A8    679      ;       NEW TIME VALUE WRITTEN INTO TODR.
              00A8    680      ;       ALL REGISTERS PRESERVED.
              00A8    681      ;-
              00A8    682
              00A8    683      EXE$WRITEP_TODR::                          ; SUBROUTINE ENTRY
              00A8    684
              00A8    685              ; NAUTILUS PROCESSOR NEEDS TO USE A SEPARATE ROUTINE TO ACCESS PHYSICAL TODR
              00A8    686              ; REGISTER IN THE CONSOLE PROCESSOR FOR TWO REASONS.  FIRST, THE PHYSICAL
              00A8    687              ; TODR HAS ONE SECOND RESOLUTION INSTEAD OF 10 MSEC RESOLUTION.  SECOND, A
              00A8    688              ; REFERENCE TO THE PHYSICAL TODR IS A VERY SLOW, NON-INTERRUPTIBLE ACTION.
              00A8    689              ; NON-PHYSICAL NAUTILUS TODR REFERENCES WILL USE THE EXE$WRITE_TODR ENTRY
              00A8    690              ; WHICH WILL FABRICATE A NEW QUADWORD SYSTEM TIME.
              00A8    691
              00A8    692                                                 ; NOT NAUTILUS - FALL THROUGH TO WRITE_TODR
              00A8    693
              00A8    694      EXE$WRITE_TODR::                           ; SUBROUTINE ENTRY
              00A8    695
              00A8    696
              00A8    700
              00A8    701
              00A8    705
              00A8    706
1B   50   DA  00A8    708              MTPR    R0,#PR730$_TODR            ; TODR IS A PROCESSOR REGISTER.
              00AB    710
              00AB    711
              00AB    715
              00AB    716
              00AB    721
         05   00AB    722              RSB
```

N 6

ERRSUB730   - ERROR SUBROUTINES FOR VAX 11/730   16-SEP-1984 00:54:20 VAX/VMS Macro V04-00   Page 12
V04-002   EXE$REGSAVE - SAVE CPU-SPECIFIC IPR'S   13-SEP-1984 15:49:22 [SYSLOA.SRC]ERRSUB.MAR;5   (9)

```
                      00AC   724              .SBTTL   EXE$REGSAVE - SAVE CPU-SPECIFIC IPR'S
                      00AC   725      ;+
                      00AC   726      ; EXE$REGSAVE - CALLED BY POWERFAIL TO SAVE CPU-SPECIFIC IPR'S ON
                      00AC   727      ;               THE STACK
                      00AC   728      ;
                      00AC   729      ; INPUTS: NONE
                      00AC   730      ;
                      00AC   731      ; OUTPUTS:
                      00AC   732      ;
                      00AC   733      ;               R0 DESTROYED
                      00AC   734      ;               OTHER GENERAL REGISTERS PRESERVED
                      00AC   735      ;               IPR'S SAVED ON THE STACK AS FOLLOWS:
                      00AC   736      ;
                      00AC   737      ;               11/780:         11/750:         11/730:         11/790:    uVAX I:
                      00AC   738      ;
                      00AC   739      ; 0(SP)    PME             PME             PME             ACCS       (none)
                      00AC   740      ; 4(SP)    SBIMT           TBDR                            CSWP
                      00AC   741      ; 8(SP)                    CADR                            PME
                      00AC   742      ;
                      00AC   743      ;-
                      00AC   744
                      00AC   745              .ENABL   LSB
                      00AC   746
                      00AC   747  EXE$REGSAVE::                                    ;SUBROUTINE ENTRY
              01  BA  00AC   749              POPR     #^M<R0>                     ;CLEAR RETURN FROM STACK
                      00AE   750
                      00AE   751
                      00AE   756
                      00AE   757
                      00AE   763
                      00AE   764
         7E  3D  DB   00AE   766              MFPR     #PR730$_PME,-(SP)           ;SAVE PERFORMANCE MONITOR ENABLE
                      00B1   768
                      00B1   769
                      00B1   776
              60  17  00B1   777              JMP      (R0)                        ;DONE, RETURN
                      00B3   779
                      00B3   783
                      00B3   784              .DSABL   LSB
```

```
                        00B3    786                .SBTTL  EXE$REGRESTOR - RESTORE CPU-SPECIFIC IPR'S
                        00B3    787        ;+
                        00B3    788        ; EXE$REGRESTOR - CALLED BY POWERFAIL RECOVERY TO RESTORE CPU-SPECIFIC
                        00B3    789        ;                 IPR'S FROM THE STACK.
                        00B3    790        ;
                        00B3    791        ; INPUTS:
                        00B3    792        ;
                        00B3    793        ;            R6 - TOP OF STACK
                        00B3    794        ;            STACK SET UP AS DEFINED IN OUTPUTS OF EXE$REGSAVE.
                        00B3    795        ;
                        00B3    796        ; OUTPUTS:
                        00B3    797        ;
                        00B3    798        ;            R0 DESTROYED
                        00B3    799        ;            OTHER GENERAL REGISTERS PRESERVED
                        00B3    800        ;            CPU-SPECIFIC IPR'S RESTORED FROM STACK
                        00B3    801        ;            R6 - ADDRESS OF 1ST CPU-INDEPENDENT SAVED IPR
                        00B3    802        ;
                        00B3    803        ;-
                        00B3    804
                        00B3    805                .ENABL  LSB
                        00B3    806
                        00B3    807 EXE$REGRESTOR::                           ;SUBROUTINE ENTRY
             01   BA    00B3    809                POPR    #^M<R0>            ;CLEAR RETURN FROM STACK
                        00B5    810
                        00B5    811
                        00B5    816
                        00B5    823
                        00B5    824
       3D    86   DA    00B5    826                MTPR    (R6)+,#PR730$_PME  ;RESTORE PERFORMANCE MONITOR ENABLE
                        00B8    828
                        00B8    829
                        00B8    837
             60   17    00B8    838                JMP     (R0)               ;DONE, RETURN
                        00BA    843
                        00BA    844                .DSABL  LSB
```

ERRSUB730
V04-002
       C 7
- ERROR SUBROUTINES FOR VAX 11/730   16-SEP-1984 00:54:20 VAX/VMS Macro V04-00   Page 14
EXE$INIPROCREG - CPU-DEPENDENT INITIALIZ 13-SEP-1984 15:49:22 [SYSLOA.SRC]ERRSUB.MAR;5   (11)

```
                                  00BA     846                 .SBTTL  EXE$INIPROCREG - CPU-DEPENDENT INITIALIZATION OF IPR'S
                                  00BA     847      ;+
                                  00BA     848      ; EXE$INIPROCREG - PERFORM INITIALIZATION OF INTERVAL TIMER AND
                                  00BA     849      ;    CPU-DEPENDENT REGISTERS. CALLED FROM INIT AND POWERFAIL.
                                  00BA     850      ;
                                  00BA     851      ; INPUTS:
                                  00BA     852      ;
                                  00BA     853      ;        NONE
                                  00BA     854      ;
                                  00BA     855      ; OUTPUTS:
                                  00BA     856      ;
                                  00BA     857      ;        NONE
                                  00BA     858      ;-
                                  00BA     859
                                  00BA     860      EXE$INIPROCREG::                            ; INIT PROCESSOR REGISTERS
                                  00BA     861
                                  00BA     875
                    00'    E1     00BA     876                 BBC     S^#EXE$V_CRDENABL,-         ; BRANCH IF FLAG CLEAR
         12  00000000'9F          00BC     877                         @#EXE$GL_FLAGS,20$         ;  (IGNORE ERRORS)
         50  00000000'9F   D0     00C2     878                 MOVL    @#MMG$GL_SBICONF,R0        ; GET ADDR OF MEMORY CONTROLLER
                 50    60  D0     00C9     879                 MOVL    (R0),R0                   ;  CONFIG REGISTER (1ST SLOT)
    04 A0  10000000 8F    C8     00CC     880                 BISL    #<1@28>,4(R0)             ; SET CRD REPORT BIT
                                  00D4     881                                                  ;
                    00'    E0     00D4     925 20$:            BBS     S^#EXE$V_NOCLOCK,-
         0E  00000000'9F          00D6     926                         @#EXE$GL_FLAGS,30$        ; BRANCH IF NOT USING CLOCK
                                  00DC     927
                                  00DC     931
                                  00DC     935
                                  00DC     939
         19  FFFFD8F0 8F   DA     00DC     941                 MTPR    #-<10*1000>,S^#PR730$_NICR ; LOAD NEXT INTERVAL REGISTER
                                  00E3     943
         18  800000D1 8F   DA     00E3     944                 MTPR    #^X800000D1,S^#PR$_ICCS   ; CLEAR ERROR AND START CLOCK
                          05     00EA     945 30$:            RSB                               ; AND RETURN
                                  00EB     946
                                  00EB     962
```

D 7

ERRSUB730   - ERROR SUBROUTINES FOR VAX 11/730   16-SEP-1984 00:54:20   VAX/VMS Macro V04-00   Page 15
V04-002   SYSL$CLRSBIA   13-SEP-1984 15:49:22   [SYSLOA.SRC]ERRSUB.MAR;5   (13)

```
              00EB    985                .SBTTL  SYSL$CLRSBIA
              00EB    986    ;++
              00EB    987    ; SYSL$CLRSBIA  - ON 11/790, CLEAR SBIA ERROR REGISTERS
              00EB    988    ;               - ON 11/780, 11/750, 11/730, AND MICRO-VAX I, THIS IS A NOP
              00EB    989    ;
              00EB    990    ; THIS ROUTINE IS CALLED TO CLEAR OUT SBIA ERROR BITS AFTER A MACHINE CHECK
              00EB    991    ; OCCURS (WHEN MACHINE CHECK IS HANDLED LOCALLY).
              00EB    992    ;
              00EB    993    ; THIS ROUTINE SHOULD BE CALLED AT IPL 31.
              00EB    994    ;
              00EB    995    ; INPUTS:
              00EB    996    ;         ABUS_TYPE        - AN ARRAY TYPE CODES; IDENTIFIES EACH ADAPTER ON THE
              00EB    997    ;                            ABUS.
              00EB    998    ;         ABUS_VA          - AN ARRAY OF ADAPTER SPACE VA'S FOR EACH ADAPTER
              00EB    999    ;                            ON THE ABUS.
              00EB   1000    ;
              00EB   1001    ; OUTPUTS:
              00EB   1002    ;         SBI ERROR BITS ARE CLEARED FOR EACH SBIA ON THE ABUS.
              00EB   1003    ;         ALL REGISTERS PRESERVED.
              00EB   1004    ;++
              00EB   1005    SYSL$CLRSBIA::
       05     00EB   1023                RSB                                      ; AND RETURN
```

```
                              00EC  1025                    .SBTTL  EXE$TEST_CSR
                              00EC  1026          ;+
                              00EC  1031          ; EXE$TEST_CSR - TEST A UNIBUS CONTROLLER CSR FOR EXISTENCE
                              00EC  1033          ;
                              00EC  1034          ; THIS TEST IS CPU-DEPENDENT.  THE FOLLOWING CPU'S ARE SUPPORTED:
                              00EC  1035          ;
                              00EC  1036          ;        11/780  -TEST CSR AND CHECK RESULT IN THE UBA STATUS REGISTER.
                              00EC  1037          ;        11/750  -NON-EXISTENT CSR IS REPORTED VIA MACHINE CHECK AS A
                              00EC  1038          ;                 NON-EXISTENT MEMORY REFERENCE.  CONNECT A TEMPORARY
                              00EC  1039          ;                 MACHINE CHECK HANDLER, TEST THE CSR, AND RESTORE THE
                              00EC  1040          ;                 ORIGINAL MACHINE CHECK HANDLER.
                              00EC  1041          ;        11/730  -ACTION IS THE SAME AS FOR THE 11/750.
                              00EC  1042          ;        11/790  -ACTION IS THE SAME AS FOR THE 11/780.
                              00EC  1043          ;        MICRO-VAX I -ACTION IS SAME AS FOR THE 11/750.
                              00EC  1044          ;
                              00EC  1045          ; THIS SUBROUTINE SHOULD BE CALLED VIA BRANCH OR JUMP TO SUBROUTINE AT IPL 31.
                              00EC  1046          ;
                              00EC  1047          ; INPUTS:
                              00EC  1048          ;
                              00EC  1049          ;        R0 = CSR ADDRESS
                              00EC  1050          ;        R6 = ADAPTER CONFIGURATION REGISTER ADDRESS
                              00EC  1051          ;
                              00EC  1052          ; OUTPUTS:
                              00EC  1053          ;
                              00EC  1054          ;        R0 LOW BIT SET/CLEAR FOR EXISTENT/NONEX CSR
                              00EC  1055          ;        OTHER REGISTERS PRESERVED.
                              00EC  1056          ;-
                              00EC  1057
                              00EC  1058                    .ENABL  LSB
                              00EC  1059
                              00EC  1060          EXE$TEST_CSR::                         ;SUBROUTINE ENTRY
                              00EC  1061
                  06    BB    00EC  1062                    PUSHR   #^M<R1,R2>           ;SAVE REGISTERS
                              00EE  1063
                              00EE  1122
            00000024          00EE  1123                    MCK_BER = ^X24               ;OFFSET INTO 750 MACHINE CHECK FRAME
                              00EE  1124                                                 ; FOR BUS ERROR REGISTER
            00000003          00EE  1125                    NEX     = 3                  ; BIT POSITION FOR NON-EXISTENT MEMORY
                              00EE  1130
                              00EE  1132          ;
                              00EE  1133          ; Test for non-UNIBUS I/O space addres first (IDC specific code)
                              00EE  1134          ;
51  56  00000800 8F    C1    00EE  1135                    ADDL3   #512*4,R6,R1         ;GET LOWEST LEGAL ADDRESS
            51    50    D1    00F6  1136                    CMPL    R0,R1                ;IS CSR GREATER ?
            05    1A          00F9  1137                    BGTRU   10$                  ;IF YES, DO CHECK
      50    01    9A          00FB  1138                    MOVZBL  #SS$_NORMAL,R0       ;NO CHECK TO DO, EXIT
            38    11          00FE  1139                    BRB     TEST_DONE_2          ;
                              0100  1141
51  00000000'GF       D0     0100  1142  10$:              MOVL    G^EXE$GL_SCB,R1      ;GET SCB ADDRESS
      04 A1          DD       0107  1143                    PUSHL   4(R1)                ;SAVE CURRENT MCHECK HANDLER ADDR
      52    5E       D0       010A  1144                    MOVL    SP,R2                ;MARK CURRENT STACK POSITION
04 A1  1C'AF   DE             010D  1145                    MOVAL   B^MCHK_HANDLER,4(R1) ;CONNECT TEMP MCHECK HANDLER
            60    B5          0112  1146                    TSTW    (R0)                 ;ATTEMPT TO READ CSR
      50    01    9A          0114  1147  OK:               MOVZBL  #SS$_NORMAL,R0       ;IF NO MCHECK, SET STATUS TO
                              0117  1148                                                ; SUCCESS
            1B    11          0117  1149                    BRB     TEST_DONE            ;JOIN COMMON EXIT
                              0119  1150
```

```
                          0119  1151 ;
                          0119  1152 ; TEMPORARY CSR TEST MACHINE CHECK HANDLER
                          0119  1153 ;
                          0119  1154
                          0119  1155         .ALIGN  LONG                    ;REQ'D MACHINE CHECK ALIGNMENT
                          011C  1156 MCHK_HANDLER:                           ;
                          011C  1157
                          011C  1161
            26   OF   DA  011C  1163         MTPR    #^XF,#PR730$_MCESR      ;CLEAR NON-EX MEMORY CONDITION
                          011F  1165
                          011F  1169
                          011F  1170
            50   08   D0  011F  1172         MOVL    #<1@NEX>,R0             ;SETUP
            0C   6E   D1  0122  1173         CMPL    (SP),#^X0C              ;IS THIS A 730 FRAME?
                 04   13  0125  1174         BEQL    50$                     ;YES, THEN DON'T CHECK FURTHER
       50   24   AE   D0  0127  1175         MOVL    MCK_BER(SP),R0          ;SAVE BUS ERROR REGISTER
            5E   52   D0  012B  1176 50$:     MOVL    R2,SP                  ;CLEAR MCHECK INFO FROM STACK
  E2   50   03   E1       012E  1177         BBC     #NEX,R0,OK             ;MEMORY EXISTS, PARITY FAILURE
                          0132  1179
                          0132  1186
                          0132  1188 NONEX_DEV:                              ;
                 50   D4  0132  1189         CLRL    R0                      ;SET STATUS TO FAILURE
                          0134  1190 TEST_DONE:
            04   A1 8ED0  0134  1191         POPL    4(R1)                   ;RESTORE SYSTEM MCHECK HANDLER
                          0138  1192 TEST_DONE_2:
                 06   BA  0138  1193         POPR    #^M<R1,R2>              ;RESTORE REGISTERS
                      05  013A  1194         RSB                             ;RETURN RESULT TO CALLER
                          013B  1195         .DISABLE LSB
```

```
                              013B  1197                    .SBTTL  ADPLINK - LINK ADAPTER CONTROL BLOCK INTO ADP LIST
                              013B  1198          ;+
                              013B  1199          ; ADPLINK LINKS THE ADAPTER CONTROL BLOCK TO THE END OF THE ADP LIST
                              013B  1200          ;
                              013B  1201          ; INPUT:
                              013B  1202          ;     R2 - ADDRESS OF NEW ADP
                              013B  1203          ; OUTPUTS:
                              013B  1204          ;     ADP IS LINK TO THE END OF THE ADPLIST LOCATED BY IOC$GL_ADPLIST.
                              013B  1205          ;     R0,R1 destroyed.
                              013B  1206          ;-
                              013B  1207
                              013B  1208  ADPLINK::                                  ;
    50   FFFFFFFC'9F   9E     013B  1209                    MOVAB   @#<IOC$GL_ADPLIST-ADP$L_LINK>,R0
                              0142  1210                                             ; START OF LIST
    51      04 A0      D0     0142  1211  10$:             MOVL    ADP$L_LINK(R0),R1 ; FLINK TO FIRST ENTRY
               05      13     0146  1212                    BEQL    20$              ; AT END
         50    51      D0     0148  1213                    MOVL    R1,R0            ; TRY AGAIN
               F5      11     014B  1214                    BRB     10$
      04 A0    52      D0     014D  1215  20$:             MOVL    R2,ADP$L_LINK(R0) ; CHAIN NEW ADP TO END OF LIST
               05             0151  1216                    RSB                      ; AND RETURN
                              0152  1217
                              0152  1218                    .END
```

```
ADP$L_CSR                        = 00000000              PR730$_ACCS                     = 00000028
ADP$L_LINK                       = 00000004              PR730$_ICR                      = 0000001A
ADP$W_ADPTYPE                    = 0000000E              PR730$_MCESR                    = 00000026
ADPLINK                            0000013B RG    03     PR730$_NICR                     = 00000019
ADP_TBL_DWN                        00000062 R     03     PR730$_PME                      = 0000003D
ADP_TBL_UP                         0000007A R     03     PR730$_TODR                     = 0000001B
BQO$L_UMR_DIS                    = 00000024              PR730$_UBRESET                  = 00000037
BQO$W_VERSION                    = 00000010              RPB$B_DEVTYP                    = 00000066
BID$K_CONSOLE                    = 00000040              RPB$L_ADPVIR                    = 00000060
C750_LIKE                        = 00000001              RPB$L_IOVEC                     = 00000034
C780_LIKE                        = 00000000              SS$_NORMAL                      = 00000001
CI$SHUTDOWN                        ******** X     03     SYS[$CLRSBIA                       000000EB RG    03
CPU_TYPE                         = 00000003              TEST_DONE                         00000134 R     03
EMB$L_CR_CODE                    = 000000F4              TEST_DONE_2                       00000138 R     03
EMB$L_CR_CPUREG                  = 00000094              UBA$INITIAL                        ******** X     03
EXE$DUMPCPUREG                     00000092 RG    03     UBA$L_MAP                       = 00000800
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                     000000BA RG    03
EXE$READP_TODR                     000000A4 RG    03
EXE$READ_TODR                      000000A4 RG    03
EXE$REGRESTOR                      000000B3 RG    03
EXE$REGSAVE                        000000AC RG    03
EXE$SHUTDWNADP                     0000003B RG    03
EXE$STARTUPADP                     00000033 RG    03
EXE$TEST_CSR                       000000EC RG    03
EXE$V_CRDENABL                     ******** X     03
EXE$V_NOCLOCK                      ******** X     03
EXE$WRITEP_TODR                    000000A8 RG    03
EXE$WRITE_TODR                     000000A8 RG    03
INI_UBADP                          0000000B R     03
IOC$GL_ADPLIST                     ******** X     03
MA$INITIAL                         ******** X     03
MBA$INITIAL                        ******** X     03
MCHK_HANDLER                       0000011C R     03
MCK_BER                          = 00000024
MMG$GL_SBICONF                     ******** X     03
NEX                              = 00000003
NONEX_DEV                          00000132 R     03
OK                                 00000114 R     03
PR$_ICCS                         = 00000018
PR$_SID_TYP730                   = 00000003
PR$_SID_TYP750                   = 00000002
PR$_SID_TYP780                   = 00000001
PR$_SID_TYP790                   = 00000004
PR$_SID_TYPUV1                   = 00000007
```

```
                                          +------------------+
                                          ! 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                        00000152  (    338.)    03 (   3.)  NOPIC   USR  CON  REL  LCL NOSHR  EXE   RD    WRT  NOVEC LONG

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

Phase                  Page faults   CPU Time        Elapsed Time
-----                  -----------   --------        ------------
Initialization              29       00:00:00.05     00:00:01.01
Command processing         107       00:00:00.48     00:00:05.05
Pass 1                     343       00:00:07.38     00:00:29.02
Symbol table sort            0       00:00:01.06     00:00:04.97
Pass 2                     131       00:00:01.98     00:00:10.41
Symbol table output          9       00:00:00.05     00:00:00.05
Psect synopsis output        2       00:00:00.02     00:00:00.01
Cross-reference output       0       00:00:00.00     00:00:00.00
Assembler run totals       623       00:00:11.02     00:00:50.53
```

The working set limit was 1500 pages.
70472 bytes (138 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 14 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
------------------                        --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                  10
_$255$DUA28:[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$:ERRSUB730/OBJ=OBJ$:ERRSUB730 MSRC$:CPUSW730/UPDATE=(ENH$:CPUSW730)+MSRC$:ERRSUB/UPDATE=(ENH$:ERRSUB)+EXECML$/LIB

ERRSUBUV1
LIS

ERRSUB790
LIS

ERRSUB780
LIS

ERRSUB750
LIS

FALMAN
LIS

ERRSUB730
LIS