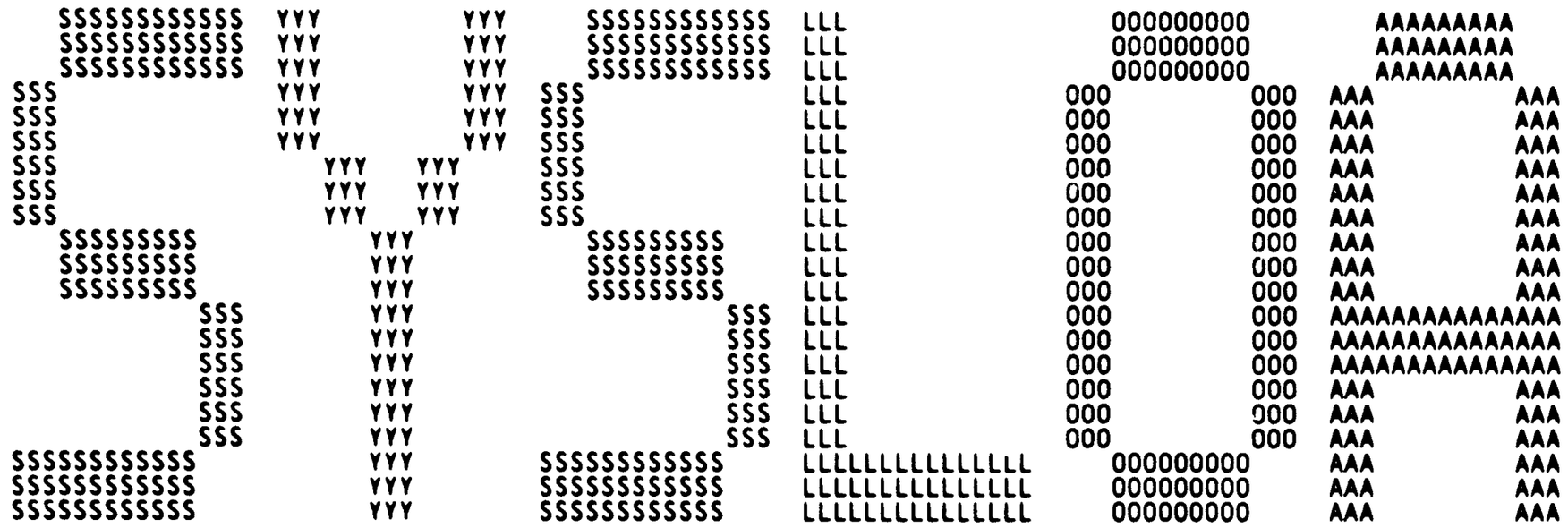
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL 000000000 AAAAAAAAA
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL 000000000 AAAAAAAAA
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL 000000000 AAAAAAAAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSSSSSSSS YYY SSSSSSSSS LLL 000 000 AAA AAA
SSSSSSSSS YYY SSSSSSSSS LLL 000 000 AAA AAA
SSSSSSSSS YYY SSSSSSSSS LLL 000 000 AAA AAA
SSS YYY SSS LLL 000 000 AAAAAAAAAAAAAAA
SSS YYY SSS LLL 000 000 AAAAAAAAAAAAAAA
SSS YYY SSS LLL 000 000 AAAAAAAAAAAAAAA
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 SSSSSSSSSSS LLLLLLLLLLLLLLL 000000000 AAA AAA
SSSSSSSSSSS YYY SSSSSSSSSSS LLLLLLLLLLLLLLL 000000000 AAA AAA
SSSSSSSSSSS YYY SSSSSSSSSSS LLLLLLLLLLLLLLL 000000000 AAA AAA


AAAAAA DDDDDDDD PPPPPPPP EEEEEEEEEE RRRRRRRR RRRRRRRR 77777777 888888 000000
AAAAAA DDDDDDDD PPPPPPPP EEEEEEEEEE RRRRRRRR RRRRRRRR 77777777 888888 000000
AA AA DD DD PP PP EE RR RR RR RR 77 88 88 00 00
AA AA DD DD PP PP EE RR RR RR RR 77 88 88 00 00
AA AA DD DD PP PP EE RR RR RR RR 77 88 88 00 0000
AA AA DD DD PP PP EE RR RR RR RR 77 88 88 00 0000
AA AA DD DD PPPPPPPP EEEEEEEE RRRRRRRR RRRRRRRR 77 888888 00 00 00
AA AA DD DD PPPPPPPP EEEEEEEE RRRRRRRR RRRRRRRR 77 888888 00 00 00
AAAAAAAAAA DD DD PP EE RR RR RR RR 77 88 88 0000 00
AAAAAAAAAA DD DD PP EE RR RR RR RR 77 88 88 0000 00
AA AA DD DD PP EE RR RR RR RR 77 88 88 00 00
AA AA DD DD PP EE RR RR RR RR 77 88 88 00 00
AA AA DDDDDDDD PP EEEEEEEEEE RR RR RR RR 77 888888 000000
AA AA DDDDDDDD PP EEEEEEEEEE RR RR RR RR 77 888888 000000
LL IIIIII SSSSSSSS
LL IIIIII SSSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

Table of contents

```
0000     1          .TITLE  ADPERR780 - I/O ADAPTER ERROR HANDLERS FOR 11/780
0000     2          .IDENT  'V04-000'
0000     3 ;
0000     4 ;*****************************************************************************
0000     5 ;*                                                                           *
0000     6 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     7 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     8 ;*  ALL RIGHTS RESERVED.                                                     *
0000     9 ;*                                                                           *
0000    10 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    11 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    12 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    13 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    14 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    15 ;*  TRANSFERRED.                                                             *
0000    16 ;*                                                                           *
0000    17 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    18 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    19 ;*  CORPORATION.                                                             *
0000    20 ;*                                                                           *
0000    21 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    22 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    23 ;*                                                                           *
0000    24 ;*                                                                           *
0000    25 ;*****************************************************************************
0000    26
0000    27 ;++
0000    28 ; FACILITY:     EXECUTIVE, ERROR HANDLING
0000    29 ;
0000    30 ; ABSTRACT:     HANDLE ERROR INTERRUPTS GENERATED BY I/O ADAPTERS (UBA AND MBA)
0000    31 ;               ON 11/780.
0000    32 ;
0000    33 ; ENVIRONMENT:  RUNS ON INTERRUPT STACK AT ADAPTER IPL.
0000    34
0000    35 ;--
0000    36          .SBTTL  HISTORY                    ; DETAILED
0000    37 ;
0000    38 ; AUTHOR: CHARLES A. SAMUELSON, CREATION DATE: 12-AUG-1980
0000    39 ;
0000    40 ; MODIFIED BY:
0000    41 ;
0000    42 ;       V03-005 KPL0100         Peter Lieberwirth       10-Feb-1984
0000    43 ;               Change to use CONFREGL.
0000    44 ;
0000    45 ;       V03-004 KDM0065         Kathleen D. Morse       3-Aug-1983
0000    46 ;               Replace PR$_TODR with cpu-specific symbol, PR780$_TODR.
0000    47 ;
0000    48 ;       V03-003 ROW0161         Ralph O. Weber          29-JAN-1983
0000    49 ;               Make MASSBUS Adapter power failure recovery work by:
0000    50 ;                1. Changing stack expansion in EXE$RH780_INT from <7*3> to
0000    51 ;                   <7*4>.
0000    52 ;                2. Changing MBA not yet powered up path in ADAPTER_ISR to not
0000    53 ;                   pop 2 non-existant longwords from the stack.
0000    54 ;                3. Correct R4 argument in MBA$INITIAL call to be adapter CSR
0000    55 ;                   address.
0000    56 ;                4. Correct SCB rebuild after power restoration so that the
0000    57 ;                   single interrupt service routine address in the ADP is
```

```
0000     58 ;                  placed in all four SCB locations associated with the
0000     59 ;                  MASSBUS.
0000     60 ;
0000     61 ;       V03-002 KTA3018        Kerbey T. Altmann       30-Oct-1982
0000     62 ;               Remove INI$MPMADP to another module.
0000     63 ;
0000     64 ;---
```

```
                     0000     66 ;
                     0000     67 ; LOCALLY DEFINED SYMBOLS
                     0000     68 ;
00000003             0000     69 UBALOGMAX       = 3                     ;MAXIMUM NUMBER OF SOFT UBA ADAPTER
                     0000     70                                         ;ERRORS TO LOG IN 15 MINUTES
00000003             0000     71 UBAMAXRETRY     = 3                     ;MAXIMUM TIMES TO TRY AND RE-INITIALIZE
                     0000     72                                         ;THE UBA AFTER POWER FAIL
00000006             0000     73 UBAMAXTIM       = 6                     ;NUMBER OF 10MS CLOCK TICKS TO WAIT
                     0000     74                                         ;FOR UBA STATUS BITS TO ASSERT
                     0000     75
                     0000     76 ;       INCLUDED SYMBOL DEFINITIONS
                     0000     77
                     0000     78         $ADPDEF                         ;DEFINE ADAPTER CONTROL BLOCK SYMBOLS
                     0000     79         $DYNDEF                         ;DEFINE DYNAMIC BLOCK TYPES
                     0000     80         $EMBDEF <MC,SB,SE>              ;DEFINE EMB OFFSETS
                     0000     81         $MCHKDEF                        ;DEFINE RECOVERY BLOCK MASK BITS
                     0000     82         $IDBDEF                         ; DEFINE INTERRUPT DISPATCH BLOCK SYMBOLS
                     0000     83         $MBADEF                         ; DEFINE MASSBUS ADAPTER SYMBOLS
                     0000     84         $MPMDEF                         ;DEFINE MULTI-PORT MEMORY
                     0000     85         $PRDEF                          ;DEFINE PROCESSOR REGISTER NUMBERS
                     0000     86         $PR780DEF                       ;DEFINE 11/780-SPECIFIC IPR NUMBERS
                     0000     87         $UBADEF                         ;DEFINE UNIBUS ADAPTOR SYMBOLS
                     0000     88         $VADEF                          ;DEF IN PFN PITS
                     0000     89
                     0000     90
                     0000     91 ;       LOCAL DATA STORAGE
                     0000     92
                     0000     93 ;       LOCAL MACROS
                     0000     94
                     0000     95         .MACRO  VECGEN                  ; Macro to generate ISR entry point
                     0000     96 $$$TEMP=.
                     0000     97         PUSHL   I^#0                    ; Replaced by ADP address for this NEXUS
                     0000     98         BRB     VECEND                  ; Join common code
                     0000     99 $$$TEMP=.-$$$TEMP
                     0000    100         ASSUME  $$$TEMP EQ 8            ; Each entry must be 8 bytes long
                     0000    101         .ENDM
                     0000    102
                00000000     103         .PSECT  WIONONPAGED,QUAD,RD,WRT
```

```
                        0000   105          .SBTTL  EXE$UBAERR_INT - UBA ERROR INTERRUPT HANDLER
                        0000   106
                        0000   107 ;++
                        0000   108 ; EXE$UBAERR_INT - UBA ADAPTOR ERROR INTERRUPT HANDLER
                        0000   109 ;
                        0000   110 ; FUNCTIONAL DESCRIPTION:
                        0000   111 ;
                        0000   112 ;       PROCESS ALL INTERRUPTS FROM DW780, 11/780 UBA
                        0000   113 ;       LOG AND TRY TO RECOVER.
                        0000   114 ;
                        0000   115 ; INPUTS:
                        0000   116 ;
                        0000   117 ;       R4 = ADDRESS OF UBA ADP
                        0000   118 ;       R5 = UNIBUS VECTOR DISPATCH ADDRESS
                        0000   119 ;       R2-R5 SAVED ON STACK+PC,PSL FROM INTERRUPT
                        0000   120 ;
                        0000   121 ; OUTPUTS:
                        0000   122 ;
                        0000   123 ;--
                        0000   124
                        0000   125
                        0000   126          .ENABL  LSB
                        0000   127
                        0000   128 EXE$UBAERR_INT::
                        0000   129
      7E    50    7D    0000   130          MOVQ    R0,-(SP)                ; SAVE R0,R1 ON STACK
      7E    54    7D    0003   131          MOVQ    R4,-(SP)                ; SAVE ADP ADDRESS AND VECTOR ADDRESS
                        0006   132          DSBINT                          ; LOCK OUT THE WORLD
                        000C   133 RESTART_UBA:                             ; ENTER HERE ON POWER UP
   7E   24 AE    7D    000C   134          MOVQ    <9*4>(SP),-(SP)         ; MAKE SECOND COPY OF PC,PSL FOR LOGGER
      53    5E    D0    0010   135          MOVL    SP,R3                   ; Mark stack
      5E    1C    C2    0013   136          SUBL    #<7*4>,SP               ; Make room for registers on stack
         017A    30    0016   137          BSBW    READ_REGISTERS          ; Read Adapter Registers
       03 50    E8    0019   138          BLBS    R0,30$                  ; UBA THERE, KEEP GOING
        007E    31    001C   139          BRW     NO_UBA                  ; CANNOT ACCESS UBA, ERROR
                        001F   140
                        001F   141 ; AT THIS POINT IN TIME THE STACK LOOKS AS FOLLOWS
                        001F   142 ;
                        001F   143 ;       PSL
                        001F   144 ;       PC
                        001F   145 ;       R5
                        001F   146 ;       R4
                        001F   147 ;       R3
                        001F   148 ;       R2
                        001F   149 ;       R1
                        001F   150 ;       R0
                        001F   151 ;       UNIBUS VECTOR DISPATCH ADDRESS
                        001F   152 ;       ADP ADDRESS FOR THIS UBA
                        001F   153 ;       IPL
                        001F   154 ;       PSL (COPY)
                        001F   155 ;       PC  (COPY)
                        001F   156 ;       TR # OF THIS ADAPTER
                        001F   157 ;       UBA$L_FUBAR
                        001F   158 ;       UBA$L_FMER
                        001F   159 ;       UBA$L_DCR
                        001F   160 ;       UBA$L_SR
                        001F   161 ;       UBA$L_CR
```

```
                                001F   162 ; SP -->UBA$L_CSR
                                001F   163
                                001F   164 ; DISPATCH ON ERROR TYPE BITS IN UBA REGISTERS
                                001F   165 ;       THE ORDER OF LOOKING AT THE BITS AND THE ACTION TAKEN IS SHOWN
                                001F   166 ;       ALL ERRORS ARE LOGGED
                                001F   167 ;               1) ADAPTER POWER DOWN - UNMAP UBA AND WAIT FOR POWER UP
                                001F   168 ;               2) ADAPTER POWER UP - REMAP THE UNIBUS AND CONTINUE
                                001F   169 ;               3) UNIBUS INIT ASSERTED - UNMAP THE UBA AND WAIT FOR POWER UP
                                001F   170 ;               4) UNIBUS POWER DOWN - UNMAP UBA AND WAIT FOR UNIBUS POWER UP
                                001F   171 ;               5) UNIBUS POWER UP - REMAP UBA, INITIALIZE DPIVERS AND CONTINUE
                                001F   172 ;               6) UBSTO - RESET UBA, INITIALIZE DRIVERS AND CONTINUE
                                001F   173 ;               7) UBSSYNTO - CONTINUE - ONLY LOG 3 EVERY 15 MINUTES
                                001F   174 ;               8) SBI ERRORS - CONTINUE - ONLY LOG 3 EVERY 15 MINUTES
                                001F   175
                                001F   176 30$:
    6E   00860000 8F    D3      001F   177         BITL    #<UBA$M_CSR_PD!UBA$M_CSR_UBIIP!UBA$M_CSR_UBPDN>,(SP)
                         5D    12 0026   178         BNEQ    UBA_POWER_DOWN
                                0028   179
          03 6E    10    E1      0028   180         BBC     #UBA$V_CSR_UBIC,(SP),210$       ; BRANCH IF NO POWER UP
                         0081   31 002C   181 200$:   BRW     UBA_POWER_UP                    ; POWER IS UP
          F9 6E    16    E0      002F   182 210$:   BBS     #UBA$V_CSR_PU,(SP),200$         ; BRANCH IF ADAPTER POWER UP
                                0033   183
       03 08 AE    01    E1      0033   184         BBC     #UBA$V_SR_UBSTO,<2*4>(SP),220$  ; BRANCH IF TIME-OUT
                         00A3   31 0038   185         BRW     UBA_STO
                                003B   186
       0A 08 AE    00    E0      003B   187 220$:   BBS     #UBA$V_SR_SSYNC,<2*4>(SP),UBA_SSYNC ; BRANCH IF SLAVE SYNCH
                                0040   188
 08 AE    000006F8 8F    D3      0040   189         BITL    #<UBA$M_SR_CXTER!-
                                0048   190                 UBA$M_SR_CXTMO!-
                                0048   191                 UBA$M_SR_DPPE!-
                                0048   192                 UBA$M_SR_IVMR!-
                                0048   193                 UBA$M_SR_MRPE!-
                                0048   194                 UBA$M_SR_RDS!-
                                0048   195                 UBA$M_SR_RDTO>,<2*4>(SP) ; ALL THE OTHER RANDOM ERRORS
             15    13          0048   196         BEQL    UBA_NOLOG_RET                   ; BRANCH IF NONE OF THEM
                                004A   197
                                004A   198 UBA_SSYNC:                                    ; DO NOTHING HERE AND HOPE FOR THE BEST **
                                004A   199 UBA_ADAPTER_ERR:                              ; JUST FOR THE RECORD **
                                004A   200
    50    0C A4    3C          004A   201         MOVZWL  ADP$W_TR(R4),R0                 ; GET TR NUMBER
0000'CF40    03    91          004E   202         CMPB    #UBALOGMAX,W^EXE$AB_MEMERR[R0] ; Log 3 of these error in 15 minutes
             09    1B          0054   203         BLEQU   UBA_NOLOG_RET
      0000'CF40    96          0056   204         INCB    W^EXE$AB_MEMERR[R0]             ; Count another one.
                                005B   205
                                005B   206 UBA_RETURN:
                                005B   207
          0154    30          005B   208         BSBW    LOG_ADAPTER                     ; Log UBA error
                  09          005E   209         .BYTE   EMB$K_UBA                       ; UBA errorlog type
                                005F   210
                                005F   211 UBA_NOLOG_RET:
                                005F   212
       53    5E    D0          005F   213         MOVL    SP,R3                           ; GET POINTER TO ADAPTER REGISTERS
                                0062   214
                                0062   215         $PRTCTINI B^40$,#MCHK$M_NEXM
                                006E   216
       65    63    D0          006E   217         MOVL    (R3),UBA$L_CSR(R5)              ; CLEAR UBA ERRORS
 08 A5    08 A3    D0          0071   218         MOVL    <2*4>(R3),UBA$L_SR(R5)          ; DITO
```

```
                        0076    219
                        0076    220         $PRTCTEND 40$
      23 50     E9      0077    221         BLBC    R0,NO_UBA               ; CANNOT ACCESS ADAPTER
                        007A    222
   5E   24     C0      007A    223         ADDL    #<9*4>,SP               ; GET RID OF  STUFF ON STACK
                        007D    224         ENBINT                          ; BACK TO DEVICE IPL
   54   8E     7D      0080    225         MOVQ    (SP)+,R4                ; GET VECTOR DISPATCH ADDRESS IN R5
        95     17      0083    226         JMP     @(R5)+                  ; OFF TO ISR
                        0085    227
```

```
                        0085   229          .SBTTL  ADAPTER ERROR HANDLERS
                        0085   230
                        0085   231 ADP_POWER_DOWN:                          ; UNIBUS ADAPTER POWER LOSS
                        0085   232 UBA_POWER_DOWN:                          ; UNIBUS (NOT ADAPTER) HAS LOST POWER
                        0085   233
                        0085   234          $PRTCTINI B^55$,#MCHK$M_NEXM
                        0091   235
04 A5    04    D0       0091   236          MOVL    #UBA$M_CR_CNFIE,UBA$L_CR(R5) ; ALLOW ONLY CONFIGURATION INTERRUPTS
   65    00    D2       0095   237          MCOML   #0,UBA$L_CSR(R5)          ; CLEAR ALL UBA ERROR BITS
08 A5    00    D2       0098   238          MCOML   #0,UBA$L_SR(R5)           ; DITO
                        009C   239
                        009C   240          $PRTCTEND 55$
                        009D   241
                        009D   242 NO_UBA:
        0127    30      009D   243          BSBW    UNMAP_UNIBUS              ; REMAP UBA ADDRESS SPACE TO RABIT HOLE PAGE
                        00A0   244
                        00A0   245 VEC_RETURN:                               ; RETURN, NOT GOING THRU VECTOR
                        00A0   246
        010F    30      00A0   247          BSBW    LOG_ADAPTER               ; Log UBA error
                09      00A3   248          .BYTE   EMB$K_UBA                 ; Log UBA error
                        00A4   249 VEC_NOLOG_RET:
     5E   24    C0      00A4   250          ADDL    #<9*4>,SP                 ; CLEAR STACK OF LOG
                        00A7   251 UBA_PWR_RET:
                        00A7   252          ENBINT                            ; RE-ENABLE INTERRRUPTS
     5E   08    C0      00AA   253          ADDL    #<2*4>,SP                 ; CLEAR STACK OF VECTOR ADDRESS, ETC.
                        00AD   254 UBA_VEC_DSPAT:
          3F    BA      00AD   255          POPR    #^M<R0,R1,R2,R3,R4,R5>    ; RESTORE GENERAL REGISTERS
                02      00AF   256          REI                               ; AT LAST!
                        00B0   257
                        00B0   258 UBA_POWER_UP:                             ; UNIBUS HAS REGAINED IS STRENGTH
                        00B0   259 RESTORE_UBA:
                        00B0   260
   3FFE 8F      BB      00B0   261          PUSHR   #^M<R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,AP,FP> ; Save regs destroyed
                        00B4   262          $PRTCTINI B^100$,#MCHK$M_NEXM
 5C   0C A4    3C      00C0   263          MOVZWL  ADP$W_TR(R4),AP           ; Initialize for this adapter only
           30   BB      00C4   264          PUSHR   #^M<R4,R5>                ; Need ADP/CSR after INIT_DEVICE
00000000'GF     16      00C6   265          JSB     G^EXE$INIT_DEVICE         ; Initialize devices on this TR level
           14   BA      00CC   266          POPR    #^M<R2,R4>                ; Restore registers for next routine
00000000'GF     16      00CE   267          JSB     G^UBA$INITIAL             ; Re-enable interrupts/init registers
                        00D4   268          $PRTCTEND 100$
   3FFE 8F      BA      00D5   269          POPR    #^M<R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,AP,FP>
     C4 50      E8      00D9   270          BLBS    R0,VEC_RETURN             ; We made it
        BF      11      00DC   271          BRB     NO_UBA                    ; No Unibus Yet
                        00DE   272
                        00DE   273 UBA_STO:                                  ; UBA SACK TIMEOUT, BAD ERROR
                        00DE   274
                        00DE   275          $PRTCTINI B^120$,#MCHK$M_NEXM
                        00EA   276
04 A5    01    D0       00EA   277          MOVL    #UBA$M_CR_INIT,UBA$L_CR(R5) ; HARD INIT THE UBA
                        00EE   278
                        00EE   279 ; WAIT FOR 60 MS OR INIT COMPLET, WHICH EVER COMES FIRST
                        00EE   280
   50   64    D0      00EE   281          MOVL    ADP$L_CSR(R4),R0          ; ADDRESS OF UBA CSR
        0323    30      00F1   282          BSBW    WAIT_TEST                 ; TEST BIT AND/OR TIME OUT
                10      00F4   283          .BYTE   UBA$V_CSR_UBIC            ; BIT POSITION TO TEST
   52   50    D0      00F5   284          MOVL    R0,R2                     ; PUT RETURN STATUS IN R2
                        00F8   285
```

```
                        00F8    286          $PRTCTEND 120$
                        00F9    287
03 52   E8      00F9    288          BLBS    R2,130$                 ; R2 IS SET, NO TIMEOUT
 FF86   31      00FC    289 115$:    BRW     ADP_POWER_DOWN          ; ERROR,WHAT ELSE CAN WE DO?
FA 50   E9      00FF    290 130$:    BLBC    R0,115$                 ; NO UBA, TAKE IT AWAY FROM SYSTEM
   AC   11      0102    291          BRB     RESTORE_UBA             ; RESTORE THE UNIBUS
                0104    292
                0104    293          .DSABL  LSB
```

```
                     0104   295          .SBTTL  EXE$RH780_INT - MBA ERROR INTERRUPT HANDLER
                     0104   296
                     0104   297 ;++
                     0104   298 ; EXE$RH780_INT - MBA ADAPTER ERROR INTERRUPT HANDLER
                     0104   299 ;
                     0104   300 ; FUNCTIONAL DESCRIPTION:
                     0104   301 ;
                     0104   302 ;       Process Adapter power down interrupt from MBA.  Log error.
                     0104   303 ;       Remap MBA adapter address space to black hole page.  Reset
                     0104   304 ;       SCB vectors for this NEXUS to point to power up ISR in this
                     0104   305 ;       module.  REI and wait for power up interrupt.
                     0104   306 ;
                     0104   307 ; INPUTS:
                     0104   308 ;
                     0104   309 ;       R3 - Address of IDB
                     0104   310 ;       R4 - Address of MBA ADP
                     0104   311 ;       00(SP) - Address of IDB address
                     0104   312 ;       04(SP) to 24(SP) - Saved R2 to R5 plus PC,PSL from interrupt
                     0104   313 ;
                     0104   314 ; OUTPUTS:
                     0104   315 ;
                     0104   316 ;--
                     0104   317
                     0104   318 EXE$RH780_INT::
                     0104   319
          6E   51    D0  0104   320          MOVL    R1,(SP)                 ; Save R1, replace IDB address
               50    DD  0107   321          PUSHL   R0                      ; Save R0
                     0109   322          DSBINT                          ; Raise to IPL 31
                     010F   323 RESTART_MBA:
       7E   1C AE    7D  010F   324          MOVQ    <7*4>(SP),-(SP)         ; Make second copy of PC,PSL for logger
          53   5E    D0  0113   325          MOVL    SP,R3                   ; Mark Stack
          5E   1C    C2  0116   326          SUBL    #<7*4>,SP               ; Make room for registers on stack
             0077    30  0119   327          BSBW    READ_REGISTERS          ; Read Adapter Registers
            18 50    E9  011C   328          BLBC    R0,NO_MBA               ; Cannot access MBA registers
                     011F   329
                     011F   330 ; The stack now looks as follows:
                     011F   331 ;
                     011F   332 ;       PSL
                     011F   333 ;       PC
                     011F   334 ;       R5
                     011F   335 ;       R4
                     011F   336 ;       R3
                     011F   337 ;       R2
                     011F   338 ;       R1
                     011F   339 ;       R0
                     011F   340 ;       IPL
                     011F   341 ;       PSL (copy)
                     011F   342 ;       PC  (copy)
                     011F   343 ;       TR # of this adapter
                     011F   344 ;       MBA$L_DR
                     011F   345 ;       MBA$L_BCR
                     011F   346 ;       MBA$L_VAR
                     011F   347 ;       MBA$L_SR
                     011F   348 ;       MBA$L_CR
                     011F   349 ; SP -->MBA$L_CSR
                     011F   350
                     011F   351 ; Log MBA status
```

```
                                011F   352
                         0090   30  011F   353          BSBW    LOG_ADAPTER             ; Call logger routine
                                0C  0122   354          .BYTE   EMB$K_MBA               ; Error log type code
                                    0123   355
                                    0123   356 ; Dispatch on either Power up or Power down
                                    0123   357 ;       If power down, remap MBA address to black hole page and change
                                    0123   358 ;       adapter SCB vectors to point into power up ISR in this module.
                                    0123   359 ;       If power up, map MBA addresses back to normal system virtual and
                                    0123   360 ;       point SCB vectors into normal system ISR.
                                    0123   361
                6E   00800000 8F    D3  0123   362          BITL    #MBA$M_CSR_PD,(SP)      ; Check for power down first
                           0B       12  012A   363          BNEQ    MBA_POWER_DOWN
                6E   00400000 8F    D3  012C   364          BITL    #MBA$M_CSR_PU,(SP)      ; Better be power up
                           22       12  0133   365          BNEQ    MBA_POWER_UP
                           53       11  0135   366          BRB     MBA_RETURN              ; No status, just REI
                                    0137   367
                                    0137   368 MBA_POWER_DOWN:                           ; Handle MBA power down interrupt
                                    0137   369 NO_MBA:
                                    0137   370          $PRTCTINI B^100$,#MCHK$M_NEXM
                           54       DD  0143   371          PUSHL   R4                      ; Save ADP address.
                     54    64       D0  0145   372          MOVL    ADP$L_CSR(R4),R4        ; Get MBA CSR address.
                00000000'GF    16       0148   373          JSB     G^MBA$INITIAL           ; Start up MBA.
                           54 8ED0       014E   374          POPL    R4                      ; Restore ADP address.
                                    0151   375          $PRTCTEND 100$
                         007B   30  0152   376          BSBW    UNMAP_MASSBUS           ; Remap to black hole page
                           33   11  0155   377          BRB     MBA_RETURN
                                    0157   378
                                    0157   379 MBA_POWER_UP:                             ; Handle MBA power up interrupt
                         010A   30  0157   380          BSBW    MAP_MASSBUS             ; Restore normal system mapping
                      3FFE 8F    BB  015A   381          PUSHR   #^M<R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,AP,FP> ; Save regs destroyed
                                    015E   382          $PRTCTINI B^100$,#MCHK$M_NEXM
                           54       DD  016A   383          PUSHL   R4                      ; Save ADP address.
                     54    64       D0  016C   384          MOVL    ADP$L_CSR(R4),R4        ; Get MBA CSR address.
                00000000'GF    16       016F   385          JSB     G^MBA$INITIAL           ; Start up MBA.
                           54 8ED0       0175   386          POPL    R4                      ; Restore ADP address.
                 5C    0C  A4    3C  0178   387          MOVZWL  ADP$W_TR(R4),AP         ; Initialize for this adapter only
                00000000'GF    16       017C   388          JSB     G^EXE$INIT_DEVICE       ; Initialize devices on this TR level
                                    0182   389          $PRTCTEND 100$
                      3FFE 8F    BA  0183   390          POPR    #^M<R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,AP,FP>
                        AD 50    E9  0187   391          BLBC    R0,MBA_POWER_DOWN       ; No MBA yet
                                    018A   392 MBA_RETURN:                              ; Common interrupt return
                           5E    24  018A   393          ADDL    #<9*4>,SP               ; Clear logging stuff off stack
                                    018D   394          ENBINT                          ; Re-enable interrupts
                           3F    BA  0190   395          POPR    #^M<R0,R1,R2,R3,R4,R5>  ; Restore GPR's
                                 02  0192   396          REI                             ; Return
```

```
                     0193   398          .SBTTL  READ_REGISTERS - READ ADAPTER REGISTERS
                     0193   399 ;++
                     0193   400 ; READ_REGISTERS - READ ADAPTER REGISTERS INTO BUFFER
                     0193   401 ;
                     0193   402 ;
                     0193   403 ; FUNCTIONAL DESCRIPTION:
                     0193   404 ;
                     0193   405 ;       Called on adapter error entry to save TR number of adapter and
                     0193   406 ;       six adapter registers in buffer.
                     0193   407 ;
                     0193   408 ; INPUTS:
                     0193   409 ;
                     0193   410 ;       R3 - Address of buffer to read registers into
                     0193   411 ;       R4 - ADP address for adapter
                     0193   412 ;
                     0193   413 ; OUTPUTS:
                     0193   414 ;
                     0193   415 ;       Adapter's TR # saved in buffer
                     0193   416 ;       Six adapter registers pushed onto buffer in reverse order
                     0193   417 ;       R1,R3 destroyed
                     0193   418 ;       R5 <- Adpater CSR address
                     0193   419 ;       R0 - LBC if adapter could not be referenced
                     0193   420 ;--
                     0193   421
                     0193   422 READ_REGISTERS:
                     0193   423
    55   64   D0     0193   424          MOVL    ADP$L_CSR(R4),R5        ; Fetch adapter CSR address
73    0C A4   3C     0196   425          MOVZWL  ADP$W_TR(R4),-(R3)      ; Save TR of this adapter for log
    51   05   D0     019A   426          MOVL    #5,R1                   ; Counter for 6 registers
                     019D   427
                     019D   428          $PRTCTINI B^20$,#MCHK$M_NEXM    ; Protect from machine check
                     01A9   429
73     6541   D0     01A9   430 10$:     MOVL    (R5)[R1],-(R3)          ; Place registers in buffer
    F9 51   F4     01AD   431          SOBGEQ  R1,10$                  ; Loop thru all 6
                     01B0   432
                     01B0   433          $PRTCTEND 20$                   ; End of protected code
                05   01B1   434          RSB
```

```
                     01B2   436          .SBTTL  LOG_ADAPTER - ERROR LOG ADAPTER ERROR
                     01B2   437 ;++
                     01B2   438 ; LOG_ADAPTER - ERROR LOGGING FOR UBA AND MBA ADAPTER ERROR INTERRUPTS
                     01B2   439 ;
                     01B2   440 ; FUNCTIONAL DESCRIPTION:
                     01B2   441 ;
                     01B2   442 ;       INTERFACE TO ERROR LOGGING ROUTINE IN MACHINE CHECK
                     01B2   443 ;
                     01B2   444 ; INPUTS:
                     01B2   445 ;
                     01B2   446 ;       (SP) - Points to error log type code
                     01B2   447 ;       (SP)+1 - Return address
                     01B2   448 ;       9 longwords are logged, defined as follows:
                     01B2   449 ;               6 Adapter registers
                     01B2   450 ;               IR # of this adapter
                     01B2   451 ;               PC
                     01B2   452 ;               PSL
                     01B2   453 ;
                     01B2   454 ; OUTPUTS:
                     01B2   455 ;
                     01B2   456 ;       Entry made in error log
                     01B2   457 ;       R0-R3 modified
                     01B2   458 ;--
                     01B2   459
                     01B2   460 LOG_ADAPTER:
                     01B2   461
          30   BB    01B2   462          PUSHR   #^M<R4,R5>              ; Save R4, R5
    55  0C AE   DE    01B4   463          MOVAL   <3*4>(SP),R5            ; Address of error log frame
  54    24   D0    01B8   464          MOVL    #<9*4>,R4               ; Length of error log frame
53   08 BE   9A    01BB   465          MOVZBL  @<2*4>(SP),R3           ; Error log type
     FE3E'  30    01BF   466          BSBW    MCHK$GL_LOG             ; Call error loggin routine in MCHK780
          30   BA    01C2   467          POPR    #^M<R4,R5>              ; Restore R4, R5
          6E   D6    01C4   468          INCL    (SP)                    ; Bump Stack past type
               05    01C6   469          RSB
```

```
                     01C7   471          .SBTTL  UNMAP ADAPTER ADDRESSES, REPLACE SCB VECTOR
                     01C7   472 ;++
                     01C7   473 ; UNMAP_UNIBUS - Remap UBA address space to black hole page
                     01C7   474 ; UNMAP_MASSBUS - Remap MBA address space to black hole page
                     01C7   475 ;
                     01C7   476 ; FUNCTIONAL DESCRIPTION:
                     01C7   477 ;
                     01C7   478 ;       For each supported adapter, modify SPTE's that map the adapter
                     01C7   479 ;       to point PFN to black hole page.  Replace the ISR address of
                     01C7   480 ;       the four adapter SCB vectors and point to entry point in this
                     01C7   481 ;       module.  This allows this routine to handle the power up
                     01C7   482 ;       interrupts.  Remap to the black hole page allows all code which
                     01C7   483 ;       touches the adapter I/O address space to continue without
                     01C7   484 ;       machine checks.  Note that any such code (drivers, map register
                     01C7   485 ;       initialization, etc.) will of course not do what it thought it
                     01C7   486 ;       was doing.  The result should be device timeout for any I/O
                     01C7   487 ;       request.
                     01C7   488 ;
                     01C7   489 ; INPUTS:
                     01C7   490 ;
                     01C7   491 ;       R4 - ADP address of the adapter
                     01C7   492 ;       IPL 31
                     01C7   493 ;
                     01C7   494 ; OUTPUTS:
                     01C7   495 ;
                     01C7   496 ;       Adpater address space (and Unibus I/O space for UBA) mapped to
                     01C7   497 ;       black hole page.
                     01C7   498 ;       Adapter SCB vectors replace to point to this module
                     01C7   499 ;       All general registers preserved.
                     01C7   500 ;
                     01C7   501 ;
                     01C7   502 ;--
                     01C7   503
                     01C7   504 UNMAP_UNIBUS:
004F 8F    BB        01C7   505          PUSHR   #^M<R0,R1,R2,R3,R6>     ; Save GPR's used
50   18    D0        01CB   506          MOVL    #24,R0                  ; Number of SPTE's to modify
     07    11        01CE   507          BRB     UNMAP
                     01D0   508
                     01D0   509 UNMAP_MASSBUS:
004F 8F    BB        01D0   510          PUSHR   #^M<R0,R1,R2,R3,R6>     ; Save GPR's used
50   08    D0        01D4   511          MOVL    #8,R0                   ; Number of SPTE's to modify
                     01D7   512
                     01D7   513 ;++
                     01D7   514 ; UNMAP - Common routine to unmap adapter address space and replace adpater
                     01D7   515 ;         SCB vector ISR addresses
                     01D7   516 ;
                     01D7   517 ; INPUTS:
                     01D7   518 ;
                     01D7   519 ;       R0 - Number of pages to remap to black hole page for this adapter
                     01D7   520 ;       R4 - Adapter ADP address
                     01D7   521 ;
                     01D7   522 ; There are interrupt service routine entry points in module ADAPTERR for
                     01D7   523 ; each of the 16 possible NEXUS on the system.  When a NEXUS is unmapped,
                     01D7   524 ; its four SCB vectors are redirected to the interrupt service routine
                     01D7   525 ; in this module.  Thus, this module receives all interrupts from an
                     01D7   526 ; unmapped adapter.
                     01D7   527 ;
```

```
                                01D7   528 ; Each interrupt service routine entry point is of the form:
                                01D7   529 ;
                                01D7   530 ;       PUSHL   I^#0
                                01D7   531 ;       BRB     VECEND
                                01D7   532 ;
                                01D7   533 ; When an adapter is unmapped, its ADP address is stored in the immediate
                                01D7   534 ; constant that is pushed on the stack.   This location is cleared when
                                01D7   535 ; the adapter is mapped and is used as a flag to prevent double mapping or
                                01D7   536 ; unmapping.
                                01D7   537 ;
                                01D7   538 ; The interrupt service routines form an array with each entry being
                                01D7   539 ; exactly 8 bytes long (a quadword).  Thus the vectors are all long word
                                01D7   540 ; alligned and some assumptions can be made for indexing into the array.
                                01D7   541 ; The index is based on the TR number of the NEXUS.  Index into the
                                01D7   542 ; array must be PIC.  With this introduction, understanding the following
                                01D7   543 ; code is left as an exercise for the student.
                                01D7   544 ;--
                                01D7   545
                                01D7   546 UNMAP:
          53    0C A4    3C   01D7   547         MOVZWL  ADP$W_TR(R4),R3         ; Get adapter TR number
       52   53    02    C5   01DB   548         MULL3   #2,R3,R2                ; Quadword index (used with indexed addressi
       56   02C6'CF     DE   01DF   549         MOVAL   W^ADP_BASE,R6           ; Get address of base of ISR table
               6642     D5   01E4   550         TSTL    (R6)[R2]                ; Test saved ADP address cell
                 43     12   01E7   551         BNEQ    100$                    ; If not zero, already unmapped
          6642   54     D0   01E9   552         MOVL    R4,(R6)[R2]             ; Save ADP address for power up
    51   00000000'GF     D0   01ED   553         MOVL    G^MMG$A_SYSPARAM+<EXE$GL_SCB-EXE$A_SYSPARAM>,R1 ; SCB base address
      51   0100 C143     DE   01F4   554         MOVAL   ^X100(R1)[R3],R1        ; Base of NEXUS vectors for this TR
                 56     D7   01FA   555         DECL    R6                      ; Calculate ISR base +1 (vector) address
          61   6642     DE   01FC   556         MOVAL   (R6)[R2],(R1)           ; Change ISR address for 4 vectors
        40 A1    61     D0   0200   557         MOVL    (R1),64(R1)
      0080 C1    61     D0   0204   558         MOVL    (R1),128(R1)
      00C0 C1    61     D0   0209   559         MOVL    (R1),192(R1)
               52   64     D0   020E   560         MOVL    ADP$L_CSR(R4),R2        ; Virtural address of adapter space
        00000000'GF     16   0211   561         JSB     G^MMG$SVAPTECHK         ; Address of SPTE that maps adapter
    51  00000000'GF     D0   0217   562         MOVL    G^EXE$GL_BLAKHOLE,R1    ; PFN of black hole page
63  15    00   51     F0   021E   563 10$:      INSV    R1,#0,#VA$S_VPN,(R3)    ; Insert new PFN in SPTE
               53   04   C0   0223   564         ADDL    #4,R3                   ; Point to next SPTE
                 F5 50    F5   0226   565         SOBGTR  R0,10$                  ; Do for all
                             0229   566         INVALID                         ; Invalidate TB
               004F 8F    BA   022C   567 100$:    POPR    #^M<R0,R1,R2,R3,R6>     ; Restore GPR's
                          05   0230   568         RSB
```

```
                     0231   570          .SBTTL  MAP ADAPTER ADDRESS, RESTORE SCB VECTORS
                     0231   571 ;++
                     0231   572 ; MAP_UNIBUS - Map UBA adapter addresses, restore UBA SCB vectors
                     0231   573 ; MAP_MASSBUS - Map MBA adapter addresses, restore MBA SCB vectors
                     0231   574 ;
                     0231   575 ; FUNCTIONAL DESCRIPTION:
                     0231   576 ;
                     0231   577 ;       Restore SCB vector address from ADP for the adapter.  Restore
                     0231   578 ;       SPTE contents that map adpater and I/O address space.  These
                     0231   579 ;       routines essentially undo are restore everything done by
                     0231   580 ;       UNMAP_ADAPTER.
                     0231   581 ;
                     0231   582 ; INPUTS:
                     0231   583 ;
                     0231   584 ;       R4 - Adapter ADP address
                     0231   585 ;       IPL 31
                     0231   586 ;
                     0231   587 ; OUTPUTS:
                     0231   588 ;
                     0231   589 ;       PFN field of SPTE's that map adapter and adapter I/O space are
                     0231   590 ;       restored to point to the adapter.
                     0231   591 ;       SCB vectors are pointed to the origonal system adapter ISR.
                     0231   592 ;--
                     0231   593
                     0231   594 MAP_UNIBUS:
            0F   BB  0231   595          PUSHR   #^M<R0,R1,R2,R3>        ; Save GPR's
            5D   10  0233   596          BSBB    MAP                     ; Common re-map routine
     50  44   A4  DE  0235   597          MOVAL   ADP$L_UBASCB(R4),R0     ; Address of table containing ISR address
        61   80  D0  0239   598          MOVL    (R0)+,(R1)              ; Restore 4 SCB vectors
     40 A1   80  D0  023C   599          MOVL    (R0)+,64(R1)
   0080 C1   80  D0  0240   600          MOVL    (R0)+,128(R1)
   00C0 C1   60  D0  0245   601          MOVL    (R0),192(R1)
     52  54   A4  D0  024A   602          MOVL    ADP$L_UBASPTE(R4),R2    ; Saved SPTE value that maps adapter
        50   18  D0  024E   603          MOVL    #24,R0                  ; UBA uses 24 SPTE's
        83   52  D0  0251   604 10$:      MOVL    R2,(R3)+                ; Replace each SPTE
             52  D6  0254   605          INCL    R2                      ; Point to next page
        50   11  D1  0256   606          CMPL    #17,R0                  ; Switch from adapter to I/O space
             04  12  0259   607          BNEQ    20$                     ; For Unibus I/O page
     52  58   A4  D0  025B   608          MOVL    ADP$L_UBASPTE+4(R4),R2  ; SPTE value that maps I/O space
        EF   50  F5  025F   609 20$:     SOBGTR  R0,10$                  ; All SPTE's
             28  11  0262   610          BRB     MAP_RET                 ; Join common code
                     0264   611
                     0264   612 MAP_MASSBUS:
            0F   BB  0264   613          PUSHR   #^M<R0,R1,R2,R3>        ; Save GPR's
            2A   10  0266   614          BSBB    MAP                     ; Common re-map routine
     50  14   A4  D0  0268   615          MOVL    ADP$L_MBASCB(R4),R0     ; Get ISR routine address
        61   50  D0  026C   616          MOVL    R0,(R1)                 ; Restore 4 SCB vectors
     40 A1   50  D0  026F   617          MOVL    R0,64(R1)
   0080 C1   50  D0  0273   618          MOVL    R0,128(R1)
   00C0 C1   50  D0  0278   619          MOVL    R0,192(R1)
     52  18   A4  D0  027D   620          MOVL    ADP$L_MBASPTE(R4),R2    ; Saved SPTE value that maps adapter
        50   08  D0  0281   621          MOVL    #8,R0                   ; MBA uses 8 SPTE's
        83   52  D0  0284   622 10$:      MOVL    R2,(R3)+                ; Replace each SPTE
             52  D6  0287   623          INCL    R2                      ; Point to next page
        F8   50  F5  0289   624          SOBGTR  R0,10$                  ; All SPTE's
                     028C   625 MAP_RET:
                     028C   626          INVALID                         ; Invalidate TB
```

```
                     028F   627 MAP_END:
                0F   BA  028F   628          POPR    #^M<R0,R1,R2,R3>        ; Restore GPR's
                     05  0291   629          RSB
                         0292   630
                         0292   631 ;++
                         0292   632 ; MAP - Common adapter re-map routines
                         0292   633 ;
                         0292   634 ; INPUTS:
                         0292   635 ;
                         0292   636 ;       R4 - address of adapter ADP
                         0292   637 ;
                         0292   638 ; OUTPUTS:
                         0292   639 ;
                         0292   640 ;       R1 - Address of first SCB vector containing ISR address for this
                         0292   641 ;            adapter
                         0292   642 ;       R2 - Address of adapter CSR
                         0292   643 ;       R3 - Address of the SPTE that maps the adapter CSR
                         0292   644 ;
                         0292   645 ;       If Adapter already mapped, stack is cleared and return to caller's
                         0292   646 ;       caller is made.
                         0292   647 ;
                         0292   648 ; Refer to comments at start of paragraph "UNMAP".
                         0292   649 ;--
                         0292   650
                         0292   651 MAP:
          53   0C A4    3C  0292   652          MOVZWL  ADP$W_TR(R4),R3         ; Adapter NEXUS TR number
        52   53   02    C5  0296   653          MULL3   #2,R3,R2                ; Make quadword index
        51   02C6'CF    DE  029A   654          MOVAL   W^ADP_BASE,R1           ; Address of vector table
             6142       D5  029F   655          TSTL    (R1)[R2]                ; Adapter already mapped?
               1A       13  02A2   656          BEQL    100$                    ; Yes, exit
             6142       D4  02A4   657          CLRL    (R1)[R2]                ; Flag adapter already mapped
    51   00000000'GF    D0  02A7   658          MOVL    G^EXE$GL_SCB,R1         ; SCB base address
      51    0100 C143   DE  02AE   659          MOVAL   ^X100(R1)[R3],R1        ; Base of NEXUS vectors for this TR
            52   64     D0  02B4   660          MOVL    ADP$L_CSR(R4),R2        ; Address of Adapter CSR
         00000000'GF    16  02B7   661          JSB     G^MMG$SVAPTECHK         ; Address of SPTE that maps adapter CSR
                        05  02BD   662          RSB
                       8E   D5  02BE   663 100$:   TSTL    (SP)+               ; Clear callers return address from stack
                       CD   11  02C0   664          BRB     MAP_END            ; Restore GPR's and return to caller's calle
```

```
                        02C2   666          .SBTTL  ADAPTER_ISR - Adapter power-up interrupt service
                        02C2   667 ;++
                        02C2   668 ; FUNCTIONAL DESCRIPTION:
                        02C2   669 ;
                        02C2   670 ;       ISR Vector intry points for adapter interrupt service on adapter
                        02C2   671 ;       power up after remapping adapter I/O space to black hole page and
                        02C2   672 ;       replacing adapter SCB vector pointers.  The adapter type is determined.
                        02C2   673 ;       Adapter specific power up code is dispatched to.
                        02C2   674 ;
                        02C2   675 ; INPUTS:
                        02C2   676 ;
                        02C2   677 ;       NONE - directly vectored to from SCB
                        02C2   678 ;
                        02C2   679 ; OUTPUS:
                        02C2   680 ;
                        02C2   681 ;       R0-R5 saved on stack
                        02C2   682 ;       R4 <- ADP address
                        02C2   683 ;       IPL Raised to 31
                        02C2   684 ;--
                        02C2   685
                        02C2   686          .ALIGN  LONG
                000002C6 02C4   687 ADP_BASE=.+2                        ; Base of ADP address save table
                000002C4 02C4   688 ISR_BASE=.                          ; Base of ISR entry point table
                        02C4   689          .REPT   16                  ; Repete once for each SCB NEXUS vector
                        02C4   690          VECGEN                      ; Generate a vector entry point
                        02C4   691          .ENDR
                        0344   692 VECEND:
                     55 DD      0344   693          PUSHL   R5                  ; Save R5
                55   04 AE    DO 0346   694          MOVL    4(SP),R5            ; Replace R5 with ADP address
                   6E   8E    DO 034A   695          MOVL    (SP)+,(SP)          ; Move stack up one longword
                     1F BB    034D   696          PUSHR   #^M<R0,R1,R2,R3,R4> ; Save rest of GPR's
                   54   55    DO 034F   697          MOVL    R5,R4               ; R4 <- ADP address
                     4B 13    0352   698          BEQL    120$                ; If no ADP address, we're in trouble
                        0354   699
                        0354   700 ; Now retrieve adapter TR # and adapter type.  Dispatch on adapter type to
                        0354   701 ; proper adapter handling code.
                        0354   702
                    50   0C A4 3C 0354   703          MOVZWL  ADP$W_TR(R4),R0     ; Fetch adapter TR number
             53 00000000'GF    DO 0358   704          MOVL    G^EXE$GL_CONFREGL,R3 ; Get address of CONFREG array
                   53   6340  DO 035F   705          MOVL    (R3)[R0],R3         ; Get adapter type code.
                        3A 13    0363   706          BEQL    120$                ; If type is 0, its bad news
                     20   53 91 0365   707          CMPB    R3,#^X20            ; Is adapter an MBA?
                        03 12    0368   708          BNEQ    100$                ; No, test for UBA
                      0062 31    036A   709          BRW     500$                ; Yes, go to MBA interrupt handler
                     28   53 91 036D   710 100$:     CMPB    R3,#^X28            ; Is adapter an UBA?
                        2D 19    0370   711          BLSS    120$                ; No, inconsistent
                     2B   53 91 0372   712          CMPB    R3,#^X2B
                        28 14    0375   713          BGTR    120$
                        0377   714
                        0377   715 ; This is a UBA interrupt
                        0377   716
                55 FD32 CF    DE 0377   717          MOVAL   W^UBA_VEC_DSPAT,R5  ; IN CASE WE TAKE UBA_RETURN
                     7E   54 7D 037C   718          MOVQ    R4,-(SP)            ; MAKE STACK COMMON
                        037F   719          DSBINT                      ; TO IPL 31
                        0385   720
                      FEA9 30  0385   721          BSBW    MAP_UNIBUS          ; REMAP THE UNIBUS ADDRESS SPACE
                        0388   722
```

```
                                                C 13
                     0388    723          $PRTCTINI B^200$,#MCHK$M_NEXM
                     0394    724
          50   64    D0  0394    725          MOVL    ADP$L_CSR(R4),R0        ; GET ADAPTER CSR ADDRESS
                     0397    726
                     0397    727 ; NOTE: THIS IS THE FIRST REFERENCE TO THE UNIBUS ADAPTER ADDRESS SPACE
                     0397    728 ; ON THE WAY UP.
                     0397    729
          51   60    D0  0397    730          MOVL    (R0),R1                 ; READ UBA CSR
          51   53    91  039A    731          CMPB    R3,R1                   ; IS THIS THE SAME ADAPTER TYPE?
               04    13  039D    732          BEQL    150$                    ; YES, CONTINUE
                     039F    733
                     039F    734 120$:    BUG_CHECK UNEXUBAINT,FATAL          ; CRASH, CANNOT SWITCH ADAPTERS
                     03A3    735
                     03A3    736 150$:
               52    D4  03A3    737          CLRL    R2                      ; ASSUME NO POWER YET
     51  00410000 8F    D3  03A5    738          BITL    #<UBA$M_CSR_UBIC!UBA$M_CSR_PU>,R1
                     03AC    739                                             ; IS THIS A POWER UP INTERRUPT?
               0E    13  03AC    740          BEQL    165$                    ; NO, IGNORE
               03    BB  03AE    741          PUSHR   #^M<R0,R1>              ; SAVE REGISTERS
            0064     30  03B0    742          BSBW    WAIT_TEST               ; WAIT FOR INIT COMPLETE, ADDRESS IN R0
                     10  03B3    743          .BYTE   UBA$V_CSR_UBIC          ; BIT TO WAIT ON
          52   50    D0  03B4    744          MOVL    R0,R2                   ; COPY RETURN CODE
               03    BA  03B7    745          POPR    #^M<R0,R1>              ; RESTORE REGISTERS
            03 52    E8  03B9    746          BLBS    R2,170$                 ; BRANCH IF OK
                     03BC    747 165$:
          60   51    D0  03BC    748          MOVL    R1,(R0)                 ; CLEAR UBA CSR ERROR BITS
                     03BF    749 170$:
                     03BF    750          $PRTCTEND 200$                      ; END PROTECTED CODE
                     03C0    751
          06 50    E9  03C0    752          BLBC    R0,300$                 ; UBA CANNOT BE ACCESSED
          03 52    E9  03C3    753          BLBC    R2,300$                 ; UBA INIT COMPLETE NOT ASSERTED
           FC43    31  03C6    754          BRW     RESTART_UBA             ; UBA IS UP AND KICKING
                     03C9    755 300$:
           FDFB    30  03C9    756          BSBW    UNMAP_UNIBUS            ; WELL, WE'RE NOT THERE YET
           FCD8    31  03CC    757          BRW     UBA_PWR_RET             ; REI AND WAIT FOR ANOTHER POWER UP
                     03CF    758
                     03CF    759 ; This is a MBA interrupt
                     03CF    760
                     03CF    761 500$:
                     03CF    762          DSBINT                          ; To IPL 31
           FE8C    30  03D5    763          BSBW    MAP_MASSBUS             ; Remap the Massbus address space
                     03D8    764
                     03D8    765          $PRTCTINI B^600$,#MCHK$M_NEXM
                     03E4    766
          50   64    D0  03E4    767          MOVL    ADP$L_CSR(R4),R0        ; Get adapter CSR address
                     03E7    768
                     03E7    769 ; This is the first reference to the Massbus Adapter address space
                     03E7    770
          51   60    D0  03E7    771          MOVL    MBA$L_CSR(R0),R1        ; Read MBA CSR
          51   53    91  03EA    772          CMPB    R3,R1                   ; Is this he same adapter type?
               B0    12  03ED    773          BNEQ    120$                    ; No, time to reboot
          52   01    9A  03EF    774          MOVZBL  #1,R2                   ; Assume success
60  00400000 8F    D3  03F2    775          BITL    #MBA$M_CSR_PU,MBA$L_CSR(R0) ; Power up?
               09    12  03F9    776          BNEQ    560$                    ; Yes
60  00800000 8F    D0  03FB    777          MOVL    #MBA$M_CSR_PD,MBA$L_CSR(R0) ; No, clear other MBA CSR error bits
               52    D4  0402    778          CLRL    R2                      ; Flag no power up yet
                     0404    779 560$:
```

```
                     0404    780          $PRTCTEND 600$
                     0405    781
06 50   E9           0405    782          BLBC    R0,700$                 ; MBA cannot be accessed
03 52   E9           0408    783          BLBC    R2,700$                 ; MBA power up not asserted
 FD01   31           040B    784          BRW     RESTART_MBA             ; MBA is up and kicking
                     040E    785 700$:
 FDBF   30           040E    786          BSBW    UNMAP_MASSBUS           ; Not there yet
                     0411    787          ENBINT                          ; Enable interrupts
 FC96   31           0414    788          BRW     UBA_VEC_DSPAT           ; REI and wait for another interrupt
                     0417    789
```

```
                     0417   791          .SBTTL  WAIT_TEST - WAIT FOR BIT OR TIMEOUT
                     0417   792 ;++
                     0417   793 ; WAIT_TEST - TEST FOR BIT SET OR WAIT FOR 60 MS, WHICHEVER COMES FIRST
                     0417   794 ;
                     0417   795 ; FUNCTIONAL DESCRIPTION:
                     0417   796 ;
                     0417   797 ;       WAIT FOR SPECIFIED BIT TO BE SET IN SPECIFIED REGISTER
                     0417   798 ;       WAIT FOR MAXIMUM SIX 10MS TODR CLOCK TICKS (50-60 MS)
                     0417   799 ;       RETURN SUCCESS IF BIT SET, FAILURE IF BIT DID NOT SET IN 60 MS
                     0417   800 ;
                     0417   801 ; INPUTS:
                     0417   802 ;
                     0417   803 ;       R0 - ADDRESS OF LOCATION THAT CONTAINS BIT TO TEST
                     0417   804 ;       (SP) - BIT NUMBER TO TEST
                     0417   805 ;
                     0417   806 ; OUTPUTS:
                     0417   807 ;
                     0417   808 ;       R0 - 1 --> BIT SET IN TIME
                     0417   809 ;       R0 - 0 --> BIT DID NOT SET WITHIN 60 MS
                     0417   810 ;       R1,R2,R3 DESTROYED, OTHER REGISTERS PRESERVED
                     0417   811 ;--
                     0417   812
                     0417   813 WAIT_TEST:
52   00 BE   9A      0417   814          MOVZBL  @(SP),R2                ; GET BIT NUMBER TO TEST
        6E   D6      041B   815          INCL    (SP)                    ; BUMP RETURN PAST PARAMATER
   51   1B   DB      041D   816          MFPR    #PR780$_TODR,R1         ; GET CURRENT TIME IN 10 MS TICKS
   53   60   D0      0420   817 10$:     MOVL    (R0),R3                 ; READ REGISTER
0E 53   52   E0      0423   818          BBS     R2,R3,20$               ; BRANCH IF BIT SET
   53   1B   DB      0427   819          MFPR    #PR780$_TODR,R3         ; GET NEW TIME
   53   51   C2      042A   820          SUBL    R1,R3                   ; GET ELAPSED TIME
   53   06   D1      042D   821          CMPL    #UBAMAXTIM,R3           ; HAVE 6 TICKS GONE BY?
        EE   1A      0430   822          BGTRU   10$                     ; NO, KEEP LOOKING
        50   D4      0432   823          CLRL    R0                      ; YES, RETURN .FALSE.
             05      0434   824          RSB
50   01   D0      0435   825 20$:     MOVL    #1,R0                           ; BIT BECAME SET, RETURN .TRUE.
             05      0438   826          RSB
                     0439   827
                     0439   828          .END
```

```
$$$TEMP                 = 00000008
ADP$L_CSR               = 00000000
ADP$L_MBASCB            = 00000014
ADP$L_MBASPTE           = 00000018
ADP$L_UBASCB            = 00000044
ADP$L_UBASPTE           = 00000054
ADP$W_TR                = 0000000C
ADP_BASE                = 000002C6 R      02
ADP_POWER_DOWN            00000085 R      02
BUG$_UNEXUBAINT           ******** X      02
EMB$K_MBA               = 0000000C
EMB$K_UBA               = 00000009
EXE$AB_MEMERR             ******** X      02
EXE$A_SYSPARAM            ******** X      02
EXE$GL_BLAKHOLE           ******** X      02
EXE$GL_CONFREGL           ******** X      02
EXE$GL_SCB                ******** X      02
EXE$INIT_DEVICE           ******** X      02
EXE$MCHK_PRTCT            ******** X      02
EXF$RH780_INT             00000104 RG     02
EXE$UBAERR_INT            00000000 RG     02
ISR_BASE                = 000002C4 R      02
LOG_ADAPTER               000001B2 R      02
MAP                       00000292 R      02
MAP_END                   0000028F R      02
MAP_MASSBUS               00000264 R      02
MAP_RET                   0000028C R      02
MAP_UNIBUS                00000231 R      02
MBA$INITIAL               ******** X      02
MBA$L_CSR               = 00000000
MBA$M_CSR_PD            = 00800000
MBA$M_CSR_PU            = 00400000
MBA_POWER_DOWN            00000137 R      02
MBA_POWER_UP              00000157 R      02
MBA_RETURN                0000018A R      02
MCHK$GL_LOG               ******** X      02
MCHK$M_NEXM             = 00000004
MMG$A_SYSPARAM            ******** X      02
MMG$$UAPTECHK             ******** X      02
NO_MBA                    00000137 R      02
NO_UBA                    0000009D R      02
PR$_IPL                 = 00000012
PR$_TBIA                = 00000039
PR780$_TODR             = 0000001B
READ_REGISTERS            00000193 R      02
RESTART_MBA               0000010F R      02
RESTART_UBA               0000000C R      02
RESTORE_UBA               000000B0 R      02
UBA$INITIAL               ******** X      02
UBA$L_CR                = 00000004
UBA$L_CSR               = 00000000
UBA$L_SR                = 00000008
UBA$M_CR_CNFIE          = 00000004
UBA$M_CR_INIT           = 00000001
UBA$M_CSR_PD            = 00800000
UBA$M_CSR_PU            = 00400000
UBA$M_CSR_UBIC          = 00010000
UBA$M_CSR_UBIIP         = 00040000
UBA$M_CSR_UBPDN         = 00020000
UBA$M_SR_CXTER          = 00000080
UBA$M_SR_CXTMO          = 00000040
UBA$M_SR_DPPE           = 00000020
UBA$M_SR_IVMR           = 00000010
UBA$M_SR_MRPE           = 00000008
UBA$M_SR_RDS            = 00000200
UBA$M_SR_RDTO           = 00000400
UBA$V_CSR_PU            = 00000016
UBA$V_CSR_UBIC          = 00000010
UBA$V_SR_SSYNC          = 00000000
UBA$V_SR_UBSTO          = 00000001
UBALOGMAX               = 00000003
UBAMAXRETRY             = 00000003
UBAMAXTIM               = 00000006
UBA_ADAPTER_ERR           0000004A R      02
UBA_NOLOG_RET             0000005F R      02
UBA_POWER_DOWN            00000085 R      02
UBA_POWER_UP              000000B0 R      02
UBA_PWR_RET               000000A7 R      02
UBA_RETURN                0000005B R      02
UBA_SSYNC                 0000004A R      02
UBA_STO                   000000DE R      02
UBA_VEC_DSPAT             000000AD R      02
UNMAP                     000001D7 R      02
UNMAP_MASSBUS             000001D0 R      02
UNMAP_UNIBUS              000001C7 R      02
VA$S_VPN                = 00000015
VECEND                    00000344 R      02
VEC_NOLOG_RET             000000A4 R      02
VEC_RETURN                000000A0 R      02
WAIT_TEST                 00000417 R      02
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
WIONONPAGED	00000439 (1081.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC QUAD

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.05	00:00:01.38
Command processing	129	00:00:00.34	00:00:01.88
Pass 1	293	00:00:06.54	00:00:29.46
Symbol table sort	0	00:00:00.81	00:00:02.92
Pass 2	154	00:00:01.82	00:00:08.65
Symbol table output	11	00:00:00.06	00:00:00.07
Psect synopsis output	2	00:00:00.01	00:00:00.37
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	626	00:00:09.65	00:00:44.90

The working set limit was 1650 pages.
56353 bytes (111 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 819 non-local and 33 local symbols.
828 source lines were read in Pass 1, producing 16 object records in Pass 2.
31 pages of virtual memory were used to define 30 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	20
_255DUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	26

939 GETS were required to define 26 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:ADPERR780/OBJ=OBJ$:ADPERR780 MSRC$:ADPERR780/UPDATE=(ENH$:ADPERR780)+EXECML$/LIB
```

ADPSUB780
LIS

ACKMSG
LIS

MCF790
SDL

MCDEF
MDL

ADPERR750
LIS

ADPSUB730
LIS

CSPDEF
SDL

CLUMBX
SDL

ADPERR780
LIS

ADPSUB750
LIS

CLUSTRMAC
MAR

CLUSTER
SDL