F 5

```
BBBBBBBBBBBB        000000000        000000000     TTTTTTTTTTTTTTT     SSSSSSSSSSSS
BBBBBBBBBBBB        000000000        000000000     TTTTTTTTTTTTTTT     SSSSSSSSSSSS
BBBBBBBBBBBB        000000000        000000000     TTTTTTTTTTTTTTT     SSSSSSSSSSSS
BBB         BBB  000         000  000         000        TTT        SSS
BBB         BBB  000         000  000         000        TTT        SSS
BBB         BBB  000         000  000         000        TTT        SSS
BBB         BBB  000         000  000         000        TTT        SSS
BBB         BBB  000         000  000         000        TTT        SSS
BBB         BBB  000         000  000         000        TTT        SSS
BBBBBBBBBBBB     000         000  000         000        TTT           SSSSSSSSS
BBBBBBBBBBBB     000         000  000         000        TTT           SSSSSSSSS
BBBBBBBBBBBB     000         000  000         000        TTT           SSSSSSSSS
BBB         BBB  000         000  000         000        TTT                    SSS
BBB         BBB  000         000  000         000        TTT                    SSS
BBB         BBB  000         000  000         000        TTT                    SSS
BBB         BBB  000         000  000         000        TTT                    SSS
BBB         BBB  000         000  000         000        TTT                    SSS
BBB         BBB  000         000  000         000        TTT                    SSS
BBBBBBBBBBBB        000000000        000000000           TTT        SSSSSSSSSSSS
BBBBBBBBBBBB        000000000        000000000           TTT        SSSSSSSSSSSS
BBBBBBBBBBBB        000000000        000000000           TTT        SSSSSSSSSSSS
```

```
                                                                        L  2
**FILE**ID**PABTDRIVR








PPPPPPPP      AAAAAA    BBBBBBBB    TTTTTTTTTT  DDDDDDDD    RRRRRRRR      IIIIII    VV      VV  RRRRRRRR
PPPPPPPP      AAAAAA    BBBBBBBB    TTTTTTTTTT  DDDDDDDD    RRRRRRRR      IIIIII    VV      VV  RRRRRRRR
PP      PP  AA      AA  BB      BB      TT      DD      DD  RR      RR      II      VV      VV  RR      RR
PP      PP  AA      AA  BB      BB      TT      DD      DD  RR      RR      II      VV      VV  RR      RR
PP      PP  AA      AA  BB      BB      TT      DD      DD  RR      RR      II      VV      VV  RR      RR
PP      PP  AA      AA  BB      BB      TT      DD      DD  RR      RR      II      VV      VV  RR      RR
PPPPPPPP    AA      AA  BBBBBBBB        TT      DD      DD  RRRRRRRR        II      VV      VV  RRRRRRRR
PPPPPPPP    AA      AA  BBBBBBBB        TT      DD      DD  RRRRRRRR        II      VV      VV  RRRRRRRR
PP          AAAAAAAAAA  BB      BB      TT      DD      DD  RR  RR          II      VV      VV  RR  RR
PP          AAAAAAAAAA  BB      BB      TT      DD      DD  RR  RR          II      VV      VV  RR  RR
PP          AA      AA  BB      BB      TT      DD      DD  RR    RR        II        VV  VV    RR    RR        ....
PP          AA      AA  BB      BB      TT      DD      DD  RR    RR        II        VV  VV    RR    RR        ....
PP          AA      AA  BBBBBBBB        TT      DDDDDDDD    RR      RR    IIIIII        VV      RR      RR      ....
PP          AA      AA  BBBBBBBB        TT      DDDDDDDD    RR      RR    IIIIII        VV      RR      RR      ....


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            .TITLE  PABTDRIVR - CI PORT BOOT DRIVER
0000            .IDENT  'V04-001'
0000
0000    ;
0000    ;*****************************************************************************
0000    ;*                                                                           *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000    ;*  ALL RIGHTS RESERVED.                                                     *
0000    ;*                                                                           *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    ;*  TRANSFERRED.                                                             *
0000    ;*                                                                           *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    ;*  CORPORATION.                                                             *
0000    ;*                                                                           *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    ;*                                                                           *
0000    ;*                                                                           *
0000    ;*****************************************************************************
0000    ;
0000
0000    ;++
0000    ; FACILITY:     BOOTS
0000    ;
0000    ; ABSTRACT:
0000    ;       This module contains the bootstrap device driver for the
0000    ;       CI port and disks accessed over it.
0000    ;
0000    ; ENVIRONMENT:  IPL 31, kernel mode, code must be PIC
0000    ;
0000    ; AUTHOR:  Kerbey T. Altmann,   CREATION DATE:  20-Nov-1981
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V04-001 WMC0001         Wayne Cardoza           05-Sep-1984
0000    ;               Built in page table must allow for bad pages.
0000    ;
0000    ;       V03-013 KTA3200         Kerbey T. Altmann       26-Jun-1984
0000    ;               Fix some bugs found during new processor testing.
0000    ;
0000    ;       V03-012 KTA3111         Kerbey T. Altmann       12-Mar-1984
0000    ;               Add support for booting off dual-pathed disks.
0000    ;               MAINT-INIT the port after doing dump.
0000    ;
0000    ;       V03-011 KTA3082         Kerbey T. Altmann       03-Oct-1983
0000    ;               Redo startup for better VC OPEN conditions.
0000    ;
0000    ;       V03-010 ROW0230         Ralph O. Weber          28-SEP-1983
0000    ;               Change a B^PQB_PTR offset to a W^PQB_PTR offset near label
0000    ;               350$ to correct a link truncation error.
```

```
          0000  ;
          0000  ;          V03-009 KTA3075          Kerbey T. Altmann          29-Aug-1983
          0000  ;                  Make status check look at major code only - this
          0000  ;                  will allow booting off disk already ONLINE.
          0000  ;
          0000  ;          V03-008 KDM0059          Kathleen D. Morse          13-Jul-1983
          0000  ;                  Replace use of IPR TODR with new TIMEDWAIT macro.
          0000  ;
          0000  ;          V03-006 KTA3067          Kerbey T. Altmann          02-Jul-1983
          0000  ;                  Enhance UNIT_DISC.
          0000  ;
          0000  ;          V03-005 KTA3057          Kerbey T. Altmann          17-Jun-1983
          0000  ;                  Redo for new SCS definitions.  Add SET CNTRL CHAR
          0000  ;                  command, support for boot device name, UNIT_DISC.
          0000  ;
          0000  ;          V03-004 KTA3034          Kerbey T. Altmann          02-Feb-1983
          0000  ;                  Recover the boot node name.
          0000  ;
          0000  ;          V03-003 KTA3011          Kerbey T. Altmann          23-Sep-1982
          0000  ;                  Fix misc bugs, add CREDIT_RSP, loop forever if no path.
          0000  ;
          0000  ;
          0000  ;--
          0000             .SBTTL  DECLARATIONS
          0000  ;
          0000  ; INCLUDE FILES:
          0000  ;
          0000
          0000             $BQODEF                          ; Boot qio offsets
          0000             $BTDDEF                          ; Boot device types
          0000             $CIBDDEF                         ; CI BDT entry def
          0000             $CIBHANDEF                       ; CI Buffer handle
          0000             $IODEF                           ; I/O function codes
          0000             $MSCPDEF                         ; MSCP definitions
          0000             $NDTDEF                          ; Adapter codes
          0000             $PAREGDEF                        ; CI port registers
          0000             $PPDDEF                          ; PPD layer definitions
          0000             $PRDEF                           ; Processor registers
          0000             $PTEDEF                          ; Page table entries
          0000             $RPBDEF                          ; RPB offsets
          0000             $SCSDEF                          ; SCS layer definitions
          0000             $SSDEF                           ; Status codes
          0000             $VADEF                           ; Virtual addresses
          0000             $VMBARGDEF                       ; VMB argument list offsets
          0000
          0000  ;
          0000  ; EQUATED SYMBOLS:
          0000  ;
          0000
00061A80  0000             TIME    =       400*1000         ; Number of 10 micro-sec intervals
          0000                                              ;  in a 4 second time-wait loop
00000060  0000             DG_SIZ  =       96
00000060  0000             MS_SIZ  =       96
          0000
          0000             $DEFINI PQB
          0000
          0000  $DEF       PQB_Q_CMDQ0     .BLKQ
          0008  $DEF      PQB_Q_CMDQ1      .BLKQ
          0010  $DEF      PQB_Q_CMDQ2      .BLKQ
          0018  $DEF      PQB_Q_CMDQ3      .BLKQ
          0020  $DEF      PQB_Q_RESPQ      .BLKQ
          0028  $DEF      PQB_L_DFRQ_HDR   .BLKL
          002C  $DEF      PQB_L_MFRQ_HDR   .BLKL
          0030  $DEF      PQB_L_DQE_LEN    .BLKL
          0034  $DEF      PQB_L_MQE_LEN    .BLKL
          0038  $DEF      PQB_L_VPQB_BASE  .BLKL
          003C  $DEF      PQB_L_BDT_BASE   .BLKL
          0040  $DEF      PQB_L_BDT_LEN    .BLKL
          0044  $DEF      PQB_L_SPT_BASE   .BLKL
          0048  $DEF      PQB_L_SPT_LEN    .BLKL
          004C  $DEF      PQB_L_GPT_BASE   .BLKL
          0050  $DEF      PQB_L_GPT_LEN    .BLKL
          0054  $DEF      TAB_L_HOLE       .BLKL
          0058  $DEF      TAB_Q_DFRQ       .BLKQ
          0060  $DEF      TAB_Q_MFRQ       .BLKQ
          0068  $DEF      TAB_L_STATE      .BLKL
          006C  $DEF      TAB_L_TIMER      .BLKL
          0070  $DEF      TAB_L_RSTAID     .BLKL
          0074  $DEF      TAB_L_PAGETBL    .BLKL
          0078  $DEF      TAB_B_BDT        .BLKB    16
          0088  $DEF      TAB_L_RCONID     .BLKL
          008C  $DEF      TAB_L_LCONID     .BLKL
          0090  $DEF      TAB_PKT0         .BLKB    MS_SIZ
          00F0  $DEF      TAB_PKT1         .BLKB    MS_SIZ
          0150  $DEF      TAB_PKT2         .BLKB    MS_SIZ
          01B0  $DEF      TAB_PKT3         .BLKB    MS_SIZ
          0210  $DEF      TAB_PKT4         .BLKB    MS_SIZ
          0270  $DEF      TAB_PKT5         .BLKB    MS_SIZ
          02D0  $DEF      TAB_PKT6         .BLKB    MS_SIZ
          0330  $DEF      TAB_PKT7         .BLKB    MS_SIZ
          0390  $DEF      TAB_PKT8         .BLKB    MS_SIZ
          03F0  $DEF      TAB_PKT9         .BLKB    MS_SIZ
          0450
00000450  0450  TAB_LEN = .-PQB_Q_CMDQ0
          0450
          0450            $DEFEND
          0000
          0000  ;
          0000  ; LOCAL MACROS
          0000  ;
          0000            .MACRO  $QRETRY OPCODE,OPER1,OPER2,ERROR,?LOOP,?OK
          0000
          0000              CLRL    R0
          0000  LOOP:       OPCODE OPER1,OPER2
          0000              BCC     OK
          0000              AOBLSS #63,R0,LOOP
          0000              BRW     ERROR
          0000  OK:
          0000            .ENDM   $QRETRY
          0000
          0000  ;
          0000  ; OWN STORAGE:
          0000  ;
          0000
```

```
                                      0000   172 ;
                                      0000   173 ; Boot driver table entry
                                      0000   174 ;
                                      0000   175
                                      0000   176         $BOOT_DRIVER    DEVTYPE = BTD$K_HSCCI,- ; Device type (MSCP/CI)
                                      0000   177                         SIZE = PA_DRVSIZ,-      ; Driver size
                                      0000   178                         ADDR = START_DRV,-      ; Driver starting address
                                      0000   179                         ENTRY = PA_DRIVER,-     ; Driver entry point
                                      0000   180                         UNIT_INIT = PA_INIT,-   ; Driver unit init entry
                                      0000   181                         UNIT_DISC = PA_DISC,-   ; Driver unit disconnect entry
                                      0000   182                         DRIVRNAME = DSKDRVNAME,-; Driver disk name
                                      0000   183                         AUXDRNAME = PRTDRVNAME,-; Driver port name
                                      0000   184                         DEVNAME = DEVNAME       ; Boot device name
                                      0000   185
                                      0000   186 START_DRV:
                                      0000   187 DSKDRVNAME:
58 45 2E 52 45 56 49 52 44 55 44 00'  0000   188         .ASCIC  /DUDRIVER.EXE/          ; Disk class driver filename
                                  45  000C
                                  0C  0000
                                      000D   189 PRTDRVNAME:
58 45 2E 52 45 56 49 52 44 41 50 00'  000D   190         .ASCIC  /PADRIVER.EXE/          ; Port driver filename
                                  45  0019
                                  0C  000D
                               55 44  001A   191 DEVNAME:.ASCII  /DU/                    ; Boot device name
                                      001C   192 TEMPL_MSG:
                                0042  001C   193         .WORD   SCS$C_CON_REQL          ; SCS$W_LENGTH
                                0004  001E   194         .WORD   PPD$C_SCS_MSG           ; PPD$W_MTYPE
                                0000  0020   195         .WORD   SCS$C_CON_REQ           ; SCS$W_MTYPE
                                0003  0022   196         .WORD   3                       ; Credit
                            00000000  0024   197         .LONG   0
                            00010001  0028   198         .LONG   ^X10001                 ; Conid
                            00000001  002C   199         .LONG   1                       ; Min send/Status
 20 20 20 4B 53 49 44 24 50 43 53 4D  0030   200         .ASCII  /MSCP$DISK       /
                         20 20 20 20  003C
 5F 4C 43 5F 4B 53 49 44 24 53 4D 56  0040   201         .ASCII  /VMS$DISK_CL_DRVR/
                         52 56 52 44  004C
                            00000034  0050   202 TEMPL_MSG_LEN=.-TEMPL_MSG
                            0050         .SBTTL  ACTION ROUTINES
                            0050 ;
                            0050 ; Routines
                            0050 ;
                            0050
                            0050 ROUT_TABLE:                               ; Start of the routines
                            0050 COPY_SYSID:
         0A 5A   1F    E1   0050         BBC     #VA$V_SYSTEM,R10,10$      ; If we are in physical mode...
            14 A2      7D   0054         MOVQ    PPD$B_SYSTEMID(R2),-      ;  (i.e. called from bootdriver)
            24 AC           0057          VMB$B_SYSTEMID(AP)               ;  then return the remote system id
            40 A2      7D   0059         MOVQ    PPD$Q_NODENAME(R2),-      ;    and the node name.
            34 AC           005C          VMB$Q_NODENAME(AP)
                       05   005E 10$:    RSB                               ; Leave
                            005F
                            005F         .ENABLE LSB
                            005F ALLOC_DG:
                            005F         $QRETRY REMQHI,TAB_Q_DFRQ(R7),R2,400$ ; Yank an entry from free queue
               05      1C   006E         BVC     500$                      ; None, trouble!
               8E      D5   0070 400$:   TSTL    (SP)+                     ; Get rid of first level return
             04F5      31   0072         BRW     ERROR                     ; Return an error status
                            0075
               05           0075 500$:   RSB                               ; Success
                            0076
                            0076 DISCARD:
         54   08 A2    9A   0076         MOVZBL  PPD$W_SIZE(R2),R4         ; Get software flag
        51   58 A744   7E   007A         MOVAQ   TAB_Q_DFRQ(R7)[R4],R1     ; Get que head
                            007F         $QRETRY INSQTI,(R2),(R1),400$     ; Put it back
      51   0928 C344   DE   008D         MOVAL   PA_DFQ(R3)[R4],R1         ; Get correct register
          61    01     D0   0093         MOVL    #PA_DFQ_M_DFQC,(R1)       ; Tell the port its there
                       05   0096         RSB
                            0097
                            0097 SEND_ACK:
     00020004 8F       D0   0097         MOVL    #<PPD$C_ACK@16+PPD$C_ACK_LEN>,-
            10 A2           009D          PPD$W_LENGTH(R2)
               3C      11   009F         BRB     20$                       ; Send it out
                            00A1
                            00A1 SEND_START:
         50   10 A2    9E   00A1         MOVAB   PPD$W_LENGTH(R2),R0       ; Set pointer
           80   3E     D0   00A5         MOVL    #<PPD$C_START@16+PPD$C_START_LEN>,(R0)+
               12      11   00A8         BRB     10$
                            00AA
                            00AA OPEN_VC:
               34      11   00AA         BRB     OPEN_VC_CONT              ; Spacing
                            00AC
                            00AC SPIN:
               45      11   00AC         BRB     SPIN_CONT
                            00AE
                            00AE         ASSUME  <.-ROUT_TABLE> LT 128
                            00AE SEND_STACK:
            40 A2      7C   00AE         CLRQ    PPD$Q_NODENAME(R2)        ; Clear out the node name field
         50   10 A2    9E   00B1         MOVAB   PPD$W_LENGTH(R2),R0       ; Set pointer
  80   0001003E 8F     D0   00B5         MOVL    #<PPD$C_STACK@16+PPD$C_STACK_LEN>,(R0)+
           80   3E     DB   00BC 10$:    MFPR    #PR$_SID,(R0)+            ; Set fake system id
                80     D4   00BF         CLRL    (R0)+                     ; Clear hi order
  80   00600060 8F     D0   00C1         MOVL    #<MS_SIZ@16!DG_SIZ>,(R0)+
  80   42534D56 8F     D0   00C8         MOVL    #^A/VMSB/,(R0)+           ; Opsys = VMS/VMB
  80   544F4F42 8F     D0   00CF         MOVL    #^A/BOOT/,(R0)+           ; Version
```

```
                 80    7C  00D6   261            CLRQ     (R0)+                   ; Boot time
                 80    D4  00D8   262            CLRL     (R0)+
            60   3E    DB  00DA   263            MFPR     #PR$_SID,(R0)           ; Processor id
               0365    31  00DD   264  20$:      BRW      SEND_DG
                           00E0   265
                           00E0   266  OPEN_VC_CONT:
                 19    B0  00E0   267            MOVW     #PPD$C_SETCKT,-
              0E A2        00E2   268                      PPD$B_OPC(R2)          ; Set to SETCKT
                       3C  00E4   269            MOVZWL   #<PPD$M_CST!PPD$M_NR-
                           00E5   270                      !PPD$M_NS>,-
      10 A2    E000 8F     00E5   271                       PPD$W_MASK(R2)        ; Open the virtual circuit
               8000 8F 3C  00EA   272            MOVZWL   #PPD$M_CST,-
              14 A2        00EE   273                      PPD$W_M_VAL(R2)        ; Clear all but the circuit status
               0356    31  00F0   274            BRW      SEND
                           00F3   275
                           00F3   276  SPIN_CONT:
                           00F3   277            TIMEDWAIT TIME=#33*1000,-        ; 1/3 second wait
                           00F3   278                     INS1=<MOVL TAB_L_RSTAID(R7),TAB_L_HOLE(R7)>,-
                           00F3   279                     DONELBL=5$               ; Waste time
                       05  0121   280            RSB
                           0122   281
                           0122   282            .DISABLE LSB
```

```
                    0122              .SBTTL  STATE/ACTION TABLES
                    0122    ;
                    0122    ; STATE TABLE
                    0122    ;
          00000000  0122              CLOSED  = 0
          00000001  0122              ST_SENT = 1
          00000002  0122              ST_RECV = 2
          00000003  0122              OPEN    = 3
                    0122
          000000C3  0122              TIMEOUT = 3
                    0122
                    0122              .MACRO  ACTION,ROUT
                    0122              .BYTE   ROUT-ROUT_TABLE
                    0122              .ENDM
                    0122
                    0122              .MACRO  SET_ST,STATE,FINISH
                    0122              .IF B FINISH
                    0122              .BYTE   128!STATE
                    0122              .IFF
                    0122              .BYTE   128!64!STATE
                    0122              .ENDC
                    0122              .ENDM
                    0122
                    0122    ;
                    0122    ; Action table
                    0122    ;
                    0122    ACTION_TABLE:
                    0122    X:
                    0122    ACT1:     ACTION  COPY_SYSID             ; Receipt of START from other side
                    0123              ACTION  OPEN_VC
                    0124              ACTION  SPIN
                    0125    ACT1A:    ACTION  ALLOC_DG
                    0126              ACTION  SEND_STACK
                    0127              SET_ST  ST_RECV
                    0128
                    0128    ACT2:     ACTION  COPY_SYSID             ; Receipt of STACK from other side
                    0129              ACTION  OPEN_VC
                    012A              ACTION  SPIN
                    012B              ACTION  ALLOC_DG
                    012C              ACTION  SEND_ACK
                    012D              SET_ST  OPEN,FINISH
                    012E
                    012E    ACT3A:    ACTION  ALLOC_DG               ; Initial or timedout action -
                    012F    ACT3:     ACTION  SEND_START             ;  send the 1st START
                    0130              SET_ST  ST_SENT
                    0131
                    0131    ACT4:     SET_ST  OPEN,FINISH
                    0132
                    0132    ACT5:     SET_ST  OPEN
                    0133
                    0133    ACT6:     ACTION  SEND_ACK
                    0134              SET_ST  OPEN
                    0135
                    0135    ACT7:     ACTION  DISCARD
                    0136              SET_ST  OPEN
                    0137
                    0137    ACT8:     ACTION  DISCARD
```

```
             0138         SET_ST  CLOSED
             0139
             0139 ACT9:   ACTION  COPY_SYSID
             013A         ACTION  SEND_STACK
             013B         SET_ST  ST_RECV
             013C ;
             013C ; State table
             013C ;
             013C STATE_TABLE:
             013C ;               CLOSED  ST_SENT ST_RECV OPEN
             013C ;               ------  ------- ------- ----
15 00 00 17  013C         .BYTE   ACT9-X, ACT1-X, ACT1-X, ACT8-X  ; START received
11 06 06 15  0140         .BYTE   ACT8-X, ACT2-X, ACT2-X, ACT6-X  ; STACK received
13 0F 15 15  0144         .BYTE   ACT8-X, ACT8-X, ACT4-X, ACT7-X  ; ACK received
10 03 0C 0D  0148         .BYTE   ACT3-X, ACT3A-X,ACT1A-X,ACT5-X  ; Timeout
             014C ;
             014C ; Dual path data
             014C ;
             014C REM_NODE:
       00 00 014C         .BYTE   0,0                     ; Dual path nodes
             014E REM_NODE_INDEX:
          01 014E         .BYTE   1                       ; Starting index to select REM_NODE
          C0 014F         .BYTE   0                       ; Extra
```

```
                           0150              .SBTTL  CI port bootstrap device initialization
                           0150
                           0150      ;++
                           0150      ;
                           0150      ; Inputs:
                           0150      ;
                           0150      ;        R9 -->  RPB
                           0150      ;        AP -->  VMB argument list
                           0150      ;
                           0150      ; Outputs:
                           0150      ;
                           0150      ;        R0 - status code
                           0150      ;
                           0150      ;--
                           0150              .ENABLE LSB
                           0150
                           0150      PA_INIT:
                     05FC  0150              .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R10>
                           0152
      000005AA'EF   5C  D0 0152              MOVL    AP,SAVE_AP              ; Save the VMB arg list
            F1 AF   01  90 0159              MOVB    #1,REM_NODE_INDEX       ; Initialize node index
               24 A9    90 015D              MOVB    RPB$L_BOOTR2(R9),-      ; Pick up path1 node
               EA AF       0160                       REM_NODE
               25 A9    90 0162              MOVB    RPB$L_BOOTR2+1(R9),-    ; Pick up path2 node
               E6 AF       0165                       REM_NODE+1
                  05    12 0167              BNEQ    5$                      ; Something there
               24 A9    90 0169              MOVB    RPB$L_BOOTR2(R9),-      ; Duplicate path1 node
               DF AF       016C                       REM_NODE+1
                  03    DD 016E      5$:     PUSHL   #3                      ; Try this three times
                  07    10 0170      10$:    BSBB    RE_INIT                 ; Do the complete initialization
               03 50    E8 0172              BLBS    R0,20$                  ; Return with success
               F8 6E    F5 0175              SOBGTR  (SP),10$                ; Failed, try again
                        04 0178      20$:    RET
                           0179      ;
                           0179      ; Iritialize the tables needed
                           0179      ;
          57   086C'CF  9E 0179      RE_INIT:MOVAB   W^TABLE,R7              ; Cover the area
          57   01FF C7  9E 017E              MOVAB   511(R7),R7              ; Set to round up
          57   01FF 8F  AA 0183              BICW    #511,R7                 ;  to next page
 67  0450 8F   00  6E 00  2C 0188              MOVC5   #0,(SP),#0,#TAB_LEN,(R7); Zero it all out
          50     38     DB 0190              MFPR    #PR$_MAPEN,R0           ; Get the mapping status
                           0193              ASSUME  RPB$L_ADPVIR EQ RPB$L_ADPPHY+4
          53   5C A940  D0 0193              MOVL    RPB$L_ADPPHY(R9)[R0],R3 ; Get correct pointer to port registers
               2A 50    E8 0198              BLBS    R0,50$                  ; If virtual, skip some set up
                           019B      ;
                           019B      ; Mapping is physical
                           019B      ;
             58  57     D0 019B              MOVL    R7,R8                   ; PQB PA=VA
          5A   01   1F  9C 019E              ROTL    #VA$V_SYSTEM,#1,R10     ; Set the system VA bit
               1C AC    C1 01A2              ADDL3   VMB$Q_UCODE(AP),-       ; Start the page table in pre-allocated
            56  20 AC      01A5                       VMB$Q_UCODE+4(AP),R6   ;  memory just beyond the ucode
             55  56     D0 01A8              MOVL    R6,R5                   ; Save the PA of page table
            54  4C A9   D0 01AB              MOVL    RPB$L_PFNCNT(R9),R4     ; Get the number of page table entries
          54   0104 C9  C0 01AF              ADDL    RPB$L_BADPGS(R9),R4     ; Add in the bad pages
             51  54     D0 01B4              MOVL    R4,R1                   ; Save
          52   09   1C  9C 01B7              ROTL    #28,S^#<<PTE$C_KW!PTE$M_VALID>@-28>&^XF,R2 ; Set up fake a SPT
             86  52     D0 01BB      40$:    MOVL    R2,(R6)+                ; Put it in
```

```
                     52   D6  01BE    422          INCL    R2                        ; Step to next page
                  F8 54   F5  01C0    423          SOBGTR  R4,40$                    ; Loop until done
                     1A   11  01C3    424          BRB     60$                       ; Join common
                              01C5    425 ;
                              01C5    426 ; Mapping is virtual
                              01C5    427 ;
      58   57   15   09   EF  01C5    428 50$:     EXTZV   #VA$V_VPN,#VA$S_VPN,R7,R8 ; Get virtual page number
           58   50 B948   DE  01CA    429          MOVAL   @RPB$[_SVASPT(R9)[R8],R8  ; Find page table entry
           58   68   09   78  01CF    430          ASHL    #9,(R8),R8                ; Turn into physical address
                     5A   D4  01D3    431          CLRL    R10                       ; Initialize
           51   00B8 C9   D0  01D5    432          MOVL    RPB$L_SLR(R9),R1          ; Size of page table
           55   00AC C9   D0  01DA    433          MOVL    RPB$L_SBR(R9),R5          ; Address
                              01DF    434 ;
                              01DF    435 ; Set up the PQB
                              01DF    436 ;
             56    58 A7  7E  01DF    437 60$:     MOVAQ   TAB_Q_DFRQ(R7),R6         ; Point to start
             52    28 A7  DE  01E3    438          MOVAL   PQB_L_DFRQ_HDR(R7),R2     ; Ditto
                62   86   7E  01E7    439          MOVAQ   (R6)+,(R2)                ; Datagram free que header
                82   5A   C8  01EA    440          BISL    R10,(R2)+                 ; Set sys VA bit
                              01ED    441          ASSUME  TAB_Q_MFRQ EQ TAB_Q_DFRQ+8
                              01ED    442          ASSUME  PQB_L_MFRQ_HDR EQ PQB_L_DFRQ_HDR+4
                62   86   7E  01ED    443          MOVAQ   (R6)+,(R2)                ; Message free que header
                82   5A   C8  01F0    444          BISL    R10,(R2)+                 ; Set sys VA bit
                              01F3    445          ASSUME  PQB_L_DQE_LEN EQ PQB_L_MFRQ_HDR+4
           82   0060 8F   3C  01F3    446          MOVZWL  #DG_SIZ,(R2)+             ; Datagram size
                              01F8    447          ASSUME  PQB_L_MQE_LEN EQ PQB_L_DQE_LEN+4
           82   0060 8F   3C  01F8    448          MOVZWL  #MS_SIZ,(R2)+             ; Message size
                86   05   D0  01FD    449          MOVL    #5,(R6)+                  ; Set STATE to initial
      86   00061A80 8F    D0  0200    450          MOVL    #TIME,(R6)+               ; Set timer
                              0207    451 ;        MOVL    RPB$L_BOOTR2(R9),(R6)+    ; Transfer remote port
           50   FF43 CF   90  0207    452          MOVB    REM_NODE_INDEX,R0         ; Pick up remote port/node index
      FF3C CF   50   01   8D  020C    453          XORB3   #1,R0,REM_NODE_INDEX      ; Set other path for next time
         86   FF35 CF40   9A  0212    454          MOVZBL  REM_NODE[R0],(R6)+        ; Transfer remote port
           86   55   5A   C9  0218    455          BISL3   R10,R5,(R6)+              ; Store the psuedo page table
                              021C    456          ASSUME  PQB_L_VPQB_BASE EQ PQB_L_MQE_LEN+4
           82   57   5A   C9  021C    457          BISL3   R10,R7,(R2)+              ; Set virtual address of self
                              0220    458          ASSUME  PQB_L_BDT_BASE EQ PQB_L_VPQB_BASE+4
           82   56   5A   C9  0220    459          BISL3   R10,R6,(R2)+              ; Address of BDT
           66   01   10   9C  0224    460          ROTL    #16,#1,(R6)               ; Set the valid bit
                              0228    461          ASSUME  PQB_L_BDT_LEN EQ PQB_L_BDT_BASE+4
                     82   D6  0228    462          INCL    (R2)+                     ; Num of entries in BDT
                              022A    463          ASSUME  PQB_L_SPT_BASE EQ PQB_L_BDT_LEN+4
                82   55   D0  022A    464          MOVL    R5,(R2)+                  ; Set phys addr of page table
                              022D    465          ASSUME  PQB_L_SPT_LEN EQ PQB_L_SPT_BASE+4
                82   51   D0  022D    466          MOVL    R1,(R2)+                  ; And length
                              0230    467
                              0230    468 ;
                              0230    469 ; Now go thru the complicated startup sequence.
                              0230    470 ;
                              0230    471
                63   63   D0  0230    472          MOVL    PA_CNF(R3),PA_CNF(R3)     ; Clear any SBI errors
             04 A3   01   D0  0233    473          MOVL    #PA_PMC_M_MIN,PA_PMC(R3); Do maint init
             54    34 A9  D0  0237    474          MOVL    RPB$L_IOVEC(R9),R4        ; Pick up address of IOVECTOR
             54    28 A4  D0  023B    475          MOVL    BQO$L_UCODE(R4),R4        ; Get address of ucode
                     52   D4  023F    476          CLRL    R2                        ; Set control store address
             14 A3   52   D0  0241    477 70$:     MOVL    R2,PA_MADR(R3)            ; Give CS to CI
             18 A3   84   D0  0245    478          MOVL    (R4)+,PA_MDATR(R3)        ; Write 4 bytes of ucode
      14 A3   52   00001000 8F   C9   0249          BISL3   #^X1000,R2,PA_MADR(R3)  ; Set CS addr of h.o. word
                    18 A3   84   3C   0252          MOVZWL  (R4)+,PA_MDATR(R3)      ; Write 2 bytes h.o.
           E3 52   00000C00 8F   F2   0256          AOBLSS  #PA_C_WCSSIZ,R2,70$     ; Loop until loaded
           04 A3   00000040 8F   C8   025E          BISL    #PA_PMC_M_PSA,PA_PMC(R3); Set program start addr
           14 A3   00000400 8F   D0   0266          MOVL    #PA_C_UCODEST,PA_MADR(R3); Set it
                  0924 C3   01   D0   026E          MOVL    #PA_PIC_M_PIC,PA_PIC(R3); Start port
                                      0273  ;
                                      0273  ; Wait a while for port to do its thing.
                                      0273  ;
                                      0273          TIMEDWAIT TIME=TAB_L_TIMER(R7),- ; Time to wait
                                      0273                  INS1=<BITL      #PA_PMC_M_MIF,PA_PMC(R3)>,- ; Check status reg
                                      0273                  INS2=<BNEQ      80$>,-  ; Br if all done
                                      0273                  DONELBL=80$             ; Label if all done
                         03 50   E8   029F          BLBS    R0,100$                 ; Br if wait time not exceeded
                          02C5   31   02A2  90$:    BRW     ERROR                   ; *** Br on yes, ERROR ***
                                      02A5
                  0900 C3   08   D1   02A5  100$:   CMPL    #PA_PS_M_PIC,PA_PS(R3)  ; Check we are done
                            F6   12   02AA          BNEQ    90$                     ; *** ERROR ***
                  0904 C3   58   D0   02AC          MOVL    R8,PA_PQBBR(R3)         ; Set the physical addr of PQB
                    04 A3   02   C8   02B1          BISL    #PA_PMC_M_MTD,PA_PMC(R3); Disable MST
                  091C C3   01   D0   02B5          MOVL    #PA_PEC_M_PEC,PA_PEC(R3); Enable the port
                  0918 C3   01   D0   02BA          MOVL    #PA_PSR_M_PSC,PA_PSR(R3); Release the status register
                                      02BF  ;
                                      02BF  ; Initialization complete. Shutdown all circuits except that of our
                                      02BF  ; remote port. Now send out a REQID to remote port.
                                      02BF  ;
                  52   0090 C7   DE   02BF          MOVAL   TAB_PKTO(R7),R2         ; Set to cover DG
                       55   0F   D0   02C4          MOVL    #15,R5                  ; Set for max port
                    70 A7   55   91   02C7  110$:   CMPB    R5,TAB_L_RSTAID(R7)     ; Our remote port?
                            1A   13   02CB          BEQL    120$                    ; Yes, skip this
                    0C A2   55   B0   02CD          MOVW    R5,PPD$B_PORT(R2)       ; Set port number
                                 B0   02D1          MOVW    #<PPD$M_RSP@8-
                                      02D2                    !PPD$C_SETCKT>,-
               0E A2   0119 8F        02D2                     PPD$B_OPC(R2)        ; Set to get it back
                       1000 8F   3C   02D7          MOVZWL  #PPD$M_DQI,-
                         10 A2        02DB                   PPD$W_MASK(R2)         ; Inhibit receipt of datagram
                         10 A2   D0   02DD          MOVL    PPD$W_MASK(R2),-
                         14 A2        02E0                   PPD$Q_M_VAL(R2)        ; Copy mask to value
                          0169   30   02E2          BSBW    SENDX                   ; Set the circuit
                            26   11   02E5          BRB     LOOP                    ;  and wait for response
                                      02E7
                                      02E7  SETCKT:
                         DD 55   F4   02E7  120$:   SOBGEQ  R5,110$                 ; Loop thru them all
                                      02EA  ;
                                      02EA  ; Now give 5 of the 6 datagrams to the port to put on the free queue.
                                      02EA  ; By giving it 6 we can be assured that 3 should always be on the actual
                                      02EA  ; queue and not hidden away in internal caches.
                                      02EA  ;
                       55   09   D0   02EA          MOVL    #9,R5                   ; Keep one in reserve
                       04   55   91   02ED  130$:   CMPB    R5,#4                   ; First four are messages
                            04   15   02F0          BLEQ    132$
                    08 A2   01   D0   02F2          MOVL    #1,PPD$W_SIZE(R2)       ; Show as message
                          FD7D   30   02F6  132$:   BSBW    DISCARD                 ; Put it on free queue
                    52   60 A2   9E   02F9          MOVAB   DG_SIZ(R2),R2           ; Step to next one
                         ED 55   F5   02FD          SOBGTR  R5,130$
                                      0300  ;
                                      0300  ; Now send out a REQID to the remote node to get the handshake started.
```

```
                              0300  ; This will also confirm that the other side is there
                              0300  ;
                05    B0      0300              MOVW    #PPD$C_REQID,-
             0E A2            0302                       PPD$B_OPC(R2)            ; Request ID from target
             10 A2    7C      0304              CLRQ    PPD$Q_XCT_ID(R2)
              013F    30      0307              BSBW    SEND                     ; Send it out and wait
             68 A7    D4      030A              CLRL    TAB_L_STATE(R7)          ; Set state to closed
                              030D  ;
                              030D  ;*************************************************************************
                              030D  ;
                              030D  ;                   Wait loop and pseudo-interrupt handler
                              030D  ;
                              030D  ;*************************************************************************
                              030D  ;
             24 A7    D5      030D  LOOP:       TSTL    PQB_Q_RESPQ+4(R7)        ; Anything already there?
                59    12      0310              BNEQ    170$                     ; Yes, take it off
                              0312              TIMEDWAIT TIME=TAB_L_TIMER(R7),-; Time to wait
                              0312                      INS1=<BITL      #PA_PMC_M_MIF,PA_PMC(R3)>,- ; Check status reg
                              0312                      INS2=<BNEQ      135$>,- ; Br if all done
                              0312                      DONELBL=135$            ; Label if all done
        0918 C3   01    D0      033E              MOVL    #PA_PSR_M_PSC,PA_PSR(R3); Release status register to port
             03 50    E8      0343              BLBS    R0,T40$                  ; Br if wait time not exceeded
              00AA    31      0346              BRW     TIMOUT                   ; Br on yes
                              0349
          38    63    D1      0349  140$:       CMPL    PA_CNF(R3),#NDT$_CI      ; Any config reg bits set except type?
                0C    13      034C              BEQL    150$                     ; No, okay
                      D0      034E              MOVL    #<PA_CNF_M_CRD-
                              034F                       !PA_CNF_M_MXTFLT-
                              034F                       !PA_CNF_M_PARFLT-
                              034F                       !PA_CNF_M_URDFLT-
                              034F                       !PA_CNF_M_WSQFLT-
                              034F                       !PA_CNF_M_XMTFLT>,-
    63    EC010000 8F         034F                       PA_CNF(R3)               ; Clear 'don't care' bits
          38    63    D1      0355              CMPL    PA_CNF(R3),#NDT$_CI      ; Try again
                0E    12      0358              BNEQ    160$                     ; *** ERROR ***
        51   0900 C3    D0      035A  150$:       MOVL    PA_PS(R3),R1
    51    FFFFFFF6 8F   D3      035F              BITL    #^C<PA_PS_M_RQA!PA_PS_M_PIC>,R1 ; Any bits but RQA or PIC?
                03    13      0366              BEQL    170$                     ; No, okay
              01FF    31      0368  160$:       BRW     ERROR                    ; *** ERROR ***
                              036B  ;
                              036B  ; Remove the entry from the response queue.
                              036B  ;
                              036B  170$:       $QRETRY REMQHI,PQB_Q_RESPQ(R7),-
                              036B                       R2,ERROR                 ; Get next response, addr in R2
                03    1C      037A              BVC     175$                     ; Br if one
              FF8E    31      037C              BRW     LOOP                     ; Br if none
                              037F
                05    EF      037F  175$:       EXTZV   #PPD$V_STSTYP,-
                03            0381                      #PPD$S_STSTYP,-
        51    0D A2           0382                       PPD$B_STATUS(R2),R1      ; Check the status
                36    13      0385              BEQL    180$                     ; Okay
          05    51    91      0387              CMPB    R1,#PPD$C_TYPNP          ; Bad, No path?
                DC    12      038A              BNEQ    160$                     ; No, something else
                              038C              TIMEDWAIT TIME=#100*1000,-       ; One second wait
                              038C                      INS1=<MOVL TAB_L_RSTAID(R7),TAB_L_HOLE(R7)>,-
                              038C                      DONELBL=500$            ; Waste time
              FDBC    31      03BA              BRW     RE_INIT                  ; Retry the whole thing
```

```
                      03BD  ;
                      03BD  ; Dispatch on opcode type.  The skip chain used here is very ugly but it
                      03BD  ; is shorter than any other way to pick 5 items out of a list of 40.
                      03BD  ;
          0E A2   91  03BD  180$:   CMPB    PPD$B_OPC(R2),-
             19       03C0                  #PPD$C_SETCKT           ; Is it a sent SETCKT?
             03   12  03C1          BNEQ    190$                    ; No, skip on
           FF21   31  03C3          BRW     SETCKT
                      03C6
          0E A2   91  03C6  190$:   CMPB    PPD$B_OPC(R2),-
             18       03C9                  #PPD$C_INVTC            ; Is it a sent INVTC?
             03   12  03CA          BNEQ    200$                    ; No, skip on
           022A   31  03CC          BRW     INVTC
                      03CF
          0C A2   91  03CF  200$:   CMPB    PPD$B_PORT(R2),-        ; Is this from our friend?
          70 A7       03D2                   TAB_L_RSTAID(R7)
             06   13  03D4          BEQL    220$                    ; Yes accept it
                      03D6
           FC9D   30  03D6  210$:   BSBW    DISCARD                 ; No, get rid of it
           FF31   31  03D9          BRW     LOOP
                      03DC
          0E A2   91  03DC  220$:   CMPB    PPD$B_OPC(R2),-
             02       03DF                  #PPD$C_SNDMSG           ; Is it a sent message?
             03   12  03E0          BNEQ    230$
           018B   31  03E2          BRW     MSG_SNT                 ; Yes, go deal with it
                      03E5
          0E A2   91  03E5  230$:   CMPB    PPD$B_OPC(R2),-
             21       03E8                  #PPD$C_DGREC            ; Is it a received datagram?
             16   13  03E9          BEQL    250$                    ; Yes
             E9   1F  03EB          BLSSU   210$                    ; Not anywhere close
          0E A2   91  03ED          CMPB    PPD$B_OPC(R2),-
             2B       03F0                  #PPD$C_IDREC            ; Is it a received ID
             05   12  03F1          BNEQ    240$                    ; Yes, fake a time out to start up
      50     03   D0  03F3  TIMOUT: MOVL    #TIMEOUT,R0             ; Timed out, set code
             12   11  03F6          BRB     260$                    ; Go figure out what to do
                      03F8
          0E A2   91  03F8  240$:   CMPB    PPD$B_OPC(R2),-
             22       03FB                  #PPD$C_MSGREC           ; Is it a received msg?
             D8   1A  03FC          BGTRU   210$                    ; No, ignore it
           007E   31  03FE          BRW     SCS_MSG                 ; Yes, go process it
                      0401
   50     12 A2   9A  0401  250$:   MOVZBL  PPD$W_MTYPE(R2),R0      ; Pick up type
      02     50   91  0405          CMPB    R0,#ST_RECV             ; Is it a handshake DG?
             CC   14  0408          BGTR    210$                    ; No, ignore it
                      040A  ;
                      040A  ; Use the incoming DG type (or timeout) and the current state to index
                      040A  ; into the state table and determine next action to take.
                      040A  ;
   03     68 A7   91  040A  260$:   CMPB    TAB_L_STATE(R7),#OPEN   ; Is it in a legal state?
             03   15  040E          BLEQ    265$                    ; Yes, continue
           0157   31  0410          BRW     ERROR                   ; No, leave
                      0413
 50    68 B740    DE  0413  265$:   MOVAL   @TAB_L_STATE(R7)[R0],R0 ; Set up to index into
50   FD1F CF40    9A  0418          MOVZBL  STATE_TABLE[R0],R0      ;  the state table to get action offset
55   FCFF CF40    9E  041E          MOVAB   ACTION_TABLE[R0],R5     ; Point to start in action table
      50     85   98  0424  270$:   CVTBL   (R5)+,R0                ; Pick up offset to routine
             07   19  0427          BLSS    280$                    ; State change
```

```
         FC22 CF40   16 0429           JSB     ROUT_TABLE[R0]            ; Do the routine
               F4    11 042E           BRB     270$                      ; Try for more
                        0430
  68 A7   50   C0 8F 8B 0430 280$:     BICB3   #192,R0,TAB_L_STATE(R7) ; Set new state
       28 50   06    E0 0436           BBS     #6,R0,VC_IS_OPEN          ; We are finished
             FED0    31 043A L·        BRW     LOOP                      ; Else go wait for new arrival
                        043D
                        043D           .DISABLE LSB
                        043D ;
                        043D ;***********************************************************************
                        043D ;
                        043D ;                      End of wait loop and dispatcher
                        043D ;
                        043D ;***********************************************************************
                        043D
                        043D SEND_MSG:
               02    90 043D           MOVB    #PPD$C_SNDMSG,-
            0E A2       043F            PPD$B_OPC(R2)            ; Set opcode to send message
                        0441
                        0441 SEND_ANY:
               06    10 0441           BSBB    SEND                      ; Send the CONNECT and wait
               F5    11 0443           BRB     L
                        0445
               01    B0 0445 SEND_DG:MOVW      #PPD$C_SNDDG,-
            0E A2       0447            PPD$B_OPC(R2)            ; Set opcode to send datagram
            70 A7    B0 0449 SEND:     MOVW    TAB_L_RSTAID(R7),-
            0C A2       044C            PPD$B_PORT(R2)           ; Set in remote port/station id
                        044E SENDX:    $QRETRY INSQTI,(R2),PQB_Q_CMDQ0(R7),ERROR
    0908 C3    01    D0 045C           MOVL    #PA_CQ0_M_CQC,PA_CQ0(R3); Tell port its there
                     05 0461           RSB
                        0462
                        0462 ;
                        0462 ; Now the virtual circuit is establised.  Send out the CONNECT_REQ
                        0462 ;
                        0462 VC_IS_OPEN:
       08 A2   01    D0 0462           MOVL    #1,PPD$W_SIZE(R2)         ; Set software flag
             FC0D    30 0466           BSBW    DISCARD                   ; Turn the DG into a free message
             FBF3    30 0469           BSBW    ALLOC_DG                  ; Grab another datagram
          7E   52    7D 046C           MOVQ    R2,-(SP)                  ; Save some registers
    FBA8 CF    34    28 046F           MOVC3   #TEMPL_MSG_LEN,TEMPL_MSG,-
            10 A2       0474            SCS$W_LENGTH-SCS$B_PPD(R2) ; Set up the CONNECT message
          52   8E    7D 0476           MOVQ    (SP)+,R2                  ; Restore some registers
       08 A2   01    D0 0479           MOVL    #1,PPD$W_SIZE(R2)         ; Set software flag
               BE    11 047D           BRB     SEND_MSG                  ; Send out a message
                        047F ;
                        047F ; Received a SCS message - process it depending on type.
                        047F ;
       68 A7   03    D0 047F SCS_MSG:MOVL      #OPEN,TAB_L_STATE(R7)     ; Set us OPEN
            14 A2    B1 0483           CMPW    SCS$W_MTYPE-SCS$B_PPD(R2),-
               00       0486            #SCS$C_CON_REQ           ; Is it a CONNECT request?
               12    13 0487           BEQL    20$                       ; Yes
            14 A2    B1 0489           CMPW    SCS$W_MTYPE-SCS$B_PPD(R2),-
               01       048C            #SCS$C_CON_RSP           ; Is it a CONNECT response?
               1F    12 048D           BNEQ    30$                       ; No
         03 22 A2    E8 048F           BLBS    SCS$W_STATUS-SCS$B_PPD(R2),10$
             00D4    31 0493           BRW     ERROR                     ; Error out if bad status received
                        0496
          FBDD   30  0496   10$:    BSBW    DISCARD                         ; Get rid of it to free queue
            9F   11  0499           BRB     L                               ; Go wait for ACCEPT request
                     049B   ;
                     049B   ; Received a CONNECT_REQUEST - Send back a NOLISTEN status: we do not
                     049B   ; support any connections other than our own MSCP initiated ones.  It
                     049B   ; is probably the SCS directory poller.
                     049B   ;
            0A   B0  049B   20$:    MOVW    #SCS$C_STNOMAT,-                ; No matching listener
         22 A2       049D                    SCS$W_STATUS-SCS$B_PPD(R2)
  50     18 A2   D0  049F           MOVL    SCS$L_DST_CONID- -
                     04A3                    SCS$B_PPD(R2),R0           ; Reverse the SRC/DST connection ids
                 D0  04A3           MOVL    SCS$L_SRC_CONID- -
                     04A4                    SCS$B_PPD(R2),-
18 A2    1C A2       04A4                    SCS$L_DST_CONID- -
                     04A8                    SCS$B_PPD(R2)
1C A2       50   D0  04A8           MOVL    R0,SCS$L_SRC_CONID-SCS$B_PPD(R2)
            27   11  04AC           BRB     50$                             ; Send it out
                     04AE
         14 A2   B1  04AE   30$:    CMPW    SCS$W_MTYPE-SCS$B_PPD(R2),-
            02       04B1                    #SCS$C_ACCP_REQ             ; Is it an ACCEPT request?
            27   12  04B2           BNEQ    60$                             ; No
                     04B4   ;
                     04B4   ; Received a valid ACCEPT_REQUEST - turn it around to be a ACCEPT_RESPONSE.
                     04B4   ;
            12   B0  04B4           MOVW    #SCS$C_ACCP_RSPL,-
         10 A2       04B6                    SCS$W_LENGTH-SCS$B_PPD(R2) ; Set new length
                 B0  04B8           MOVW    #<PPD$M_RSP@8-
                     04B9                    !PPD$C_SNDMSG>,-
OE A2     0102 8F    04B9                     PPD$B_OPC(R2)             ; Set to get it back
            01   B0  04BE   40$:    MOVW    #SCS$C_STNORMAL,-
         22 A2       04C0                    SCS$W_STATUS-SCS$B_PPD(R2) ; Set a success status
  50     18 A2   D0  04C2           MOVL    SCS$L_DST_CONID- -
                     04C6                    SCS$B_PPD(R2),R0           ; Reverse the SRC/DST connection ids
                 D0  04C6           MOVL    SCS$L_SRC_CONID- -
                     04C7                    SCS$B_PPD(R2),-
18 A2    1C A2       04C7                    SCS$L_DST_CONID- -
                     04CB                    SCS$B_PPD(R2)
1C A2       50   D0  04CB           MOVL    R0,SCS$L_SRC_CONID-SCS$B_PPD(R2)
         18 A2   7D  04CF           MOVQ    SCS$L_DST_CONID-SCS$B_PPD(R2),-
       0088 C7       04D2                    TAB_C_RCONID(R7)           ; Store them both for later use
         14 A2   B6  04D5   50$:    INCW    SCS$W_MTYPE-SCS$B_PPD(R2) ; Change to ACP_RSP
          FF62   31  04D8           BRW     SEND_MSG                        ; Send it out and wait
                     04DB
         14 A2   B1  04DB   60$:    CMPW    SCS$W_MTYPE-SCS$B_PPD(R2),-
            08       04DE                    #SCS$C_CR_REQ               ; Is it an CREDIT request?
            06   12  04DF           BNEQ    70$                             ; No
                     04E1   ;
                     04E1   ; Received a CREDIT_REQUEST - send out a CREDIT_RESPONSE and ignore
                     04E1   ;
            0E   B0  04E1           MOVW    #SCS$C_CR_RSPL,-
         10 A2       04E3                    SCS$W_LENGTH-SCS$B_PPD(R2) ; Set new length
            D7   11  04E5           BRB     40$                             ; Send it out
                     04E7
         14 A2   B1  04E7   70$:    CMPW    SCS$W_MTYPE-SCS$B_PPD(R2),-
            0A       04EA                    #SCS$C_APPL_MSG             ; Is it an application message?
            7D   12  04EB           BNEQ    ERROR                           ; No, ERROR
                     04ED   ;
```

```
                        04ED   764 ; Received packet was not a datagram or a SCS control message.  It must
                        04ED   765 ; therefor be a MSCP packet.
                        04ED   766 ;
  05A6'CF     52    D0  04ED   767          MOVL    R2,W^NXT_MSG               ; Hold for later
  05A2'CF     57    D0  04F2   768          MOVL    R7,W^PQB_PTR
                        04F7   769          ASSUME  MSCP$V_ST_MASK EQ 0
     FFE0 8F        AB  04F7   770          BICW3   #^C<<1@MSCP$S_ST_MASK>-1>,-
  50   2A A2            04FB   771           MSCP$W_STATUS-SCS$B_PPD(R2),R0 ; Any drive errors?
          19        13  04FE   772          BEQL    100$                       ; No, continue on
     04   50        B1  0500   773          CMPW    R0,#MSCP$K_ST_AVLBL        ; Yes, is it drive available?
          05        13  0503   774          BEQL    80$                        ; Yes
     03   50        B1  0505   775          CMPW    R0,#MSCP$K_ST_OFFLN        ; No, is it no such drive?
          60        12  0508   776          BNEQ    ERROR                      ; No, error out
       64 A9        B5  050A   777 80$:     TSTW    RPB$W_UNIT(R9)             ; Is this 1st try at the shadow unit?
          07        18  050D   778          BGEQ    90$                        ; No, ordinay failure
       28 A9        B0  050F   779          MOVW    RPB$L_BOOTR3(R9),-         ; Yes, replace unit with physical
       64 A9            0512   780           RPB$Q_UNIT(R9)
          14        11  0514   781          BRB     110$                       ;  and try the ONLINE again
                        0516   782
        FC60        31  0516   783 90$:     BRW     RE_INIT                    ; Failure, start from scratch
                        0519   784
       28 A2        91  0519   785 100$:    CMPB    MSCP$B_OPCODE-SCS$B_PPD(R2),-
       84 8F            051C   786           #MSCP$K_OP_STCON!-
                        051E   787            MSCP$K_OP_END             ; Is it SET CTRL CHAR?
          24        12  051E   788          BNEQ    120$                       ; No
                        0520   789 ;
                        0520   790 ; After receipt of the end packet from the SET CTRL CHAR, it is time to
                        0520   791 ; attempt to put the unit online.  First we check to see if we are booting
                        0520   792 ; from a shadowed disk.  If so, we attempt the ONLINE to the shadowed unit.
                        0520   793 ; If that fails, we try the physical unit.  If we are booting from a single
                        0520   794 ; physical unit, we do nothing special.
                        0520   795 ;
  50   2A A9        B0  0520   796          MOVW    RPB$L_BOOTR3+2(R9),R0      ; Pick up possible shadow unit
          04        18  0524   797          BGEQ    110$                       ; Not a shadow unit
  64 A9   50        B0  0526   798          MOVW    R0,RPB$W_UNIT(R9)          ; Use the shadow unit first
  55   20 A2        DE  052A   799 110$:    MOVAL   -SCS$B_PPD(R2),R5          ; Set R5 to cover packet
    85    01        D0  052E   800          MOVL    #1,(R5)+                   ; Set command ref number
  85   64 A9        3C  0531   801          MOVZWL  RPB$W_UNIT(F9),(R5)+       ; Put unit number in cmd packet field
    85    09        9A  0535   802          MOVZBL  #MSCP$K_OP_ONLIN,(R5)+     ; Set opcode to bring drive online
          85        7C  0538   803          CLRQ    (R5)+                      ; Clear byte count, buff desc
          85        7C  053A   804          CLRQ    (R5)+                      ;  buff desc and LBN
          65        7C  053C   805          CLRQ    (R5)                       ;  resvd and copy speed
          32        B0  053E   806          MOVW    #<-SCS$B_PPD-PPD$C_LENGTH>+MSCP$W_SHDW_UNT+4,-
       10 A2            0540   807           SCS$W_LENGTH-SCS$B_PPD(R2) ; Set the message length
          51        11  0542   808          BRB     Q                          ; Send it out
                        0544   809
       28 A2        91  0544   810 120$:    CMPB    MSCP$B_OPCODE-SCS$B_PPD(R2),-
       89 8F            0547   811           #MSCP$K_OP_ONLIN!-
                        0549   812            MSCP$K_OP_END             ; Is it ONLINE?
          1B        12  0549   813          BNEQ    130$                       ; No
                        054B   814 ;
                        054B   815 ; Received packet was a successful ONLINE end packet.  Pick up the device
                        054B   816 ; name from the MEDIA_ID field.
                        054B   817 ;
          16        EF  054B   818          EXTZV   #MSCP$V_MTYP_D1,-
          05            054D   819           #MSCP$S_MTYP_D1,-
  51   3C A2            054E   820           MSCP$L_MEDIA_ID- -
```

```
                                0551   821                     SCS$B_PPD(R2),R1        ; Pull out 2nd device character
          51   51   08    78    0551   822          ASHL      #8,R1,R1                 ; Stick it in high byte
                    1B    EF    0555   823          EXTZV     #MSCP$V_MTYP_D0,-
                    05          0557   824                     #MSCP$S_MTYP_D0,-
            52   3C A2          0558   825                     MSCP$L_MEDIA_ID- -
                                055B   826                     SCS$B_PPD(R2),R2        ; Pull out 1st device character
          51   4040 8F    A8    055B   827          BISW      #^X4040,R1               ; Make ASCII characters
    FAB4 CF   52   51     A9    0560   828          BISW3     R1,R2,DEVNAME            ; Set into driver name
                                0566   829 ;
                                0566   830 ; Transfer is complete. Return with success status code.
                                0566   831 ;
               50   01    3C    0566   832 130$:    MOVZWL    #SS$_NORMAL,R0           ; Set completion code
                          05    0569   833          RSB                                ; And return
                                056A   834 ;
                                056A   835 ; Error occured during transfer.  Return and retry.
                                056A   836 ;
          50   0054 8F    3C    056A   837 ERROR:   MOVZWL    #SS$_CTRLERR,R0          ; Set failure status
                          05    056F   838          RSB                                ; Return to BOOTDRIVR
                                0570   839
                                0570   840 ;
                                0570   841 ; Now set the controller characteristics
                                0570   842 ;
               0E A2      B4    0570   843 MSG_SNT:CLRW      PPD$B_OPC(R2)            ; Clear response flags
            55   20 A2    DE    0573   844          MOVAL     -SCS$B_PPD(R2),R5        ; Set R5 to cover packet
               85   01    D0    0577   845          MOVL      #1,(R5)+                 ; Set command ref number
                    85    D4    057A   846          CLRL      (R5)+                    ; Reserved field
               85   04    9A    057C   847          MOVZBL    #MSCP$K_OP_STCON,(R5)+   ; Set opcode to SET CTRL CHAR
                    85    D4    057F   848          CLRL      (R5)+                    ; Version & flags
          85   FF 8F      9A    0581   849          MOVZBL    #255,(R5)+               ; Set time out to 255 seconds
                    85    7C    0585   850          CLRQ      (R5)+                    ; Clear time and date
                    65    D4    0587   851          CLRL      (R5)                     ; Clear controller depend params
                    04    B0    0589   852          MOVW      #PPD$C_SCS_MSG,-         ; Set PPD type to application
               12 A2            058B   853                     PPD$W_MTYPE(R2)         ;  message
                    0A    B0    058D   854          MOVW      #SCS$C_APPL_MSG,-        ; Set SCS type to application
               14 A2            058F   855                     SCS$W_MTYPE-SCS$B_PPD(R2) ;  message
                    2E    B0    0591   856          MOVW      #<-SCS$B_PPD-PPD$C_LENGTH>+MSCP$Q_TIME+12,-
               10 A2            0593   857                     SCS$W_LENGTH-SCS$B_PPD(R2) ; Set the message length
          16 A2   01      B0    0595   858 Q:       MOVW      #1,SCS$W_CREDIT-SCS$B_PPD(R2); Always give back one credit
             0088 C7      7D    0599   859          MOVQ      TAB_L_RCONID(R7),-
               18 A2            059D   860                     SCS$L_DST_CONID-SCS$B_PPD(R2)  ; Set correct xct_id
                FE9B      31    059F   861          BRW       SEND_MSG                          ; Send it out and wait
                                05A2   862 ;
                    00000000    05A2   863 PQB_PTR:.LONG   0
                    00000000    05A6   864 NXT_MSG:.LONG   0
                    00000000    05AA   865 SAVE_AP:.LONG   0
          05AE          .SBTTL  CI port bootstrap driver QIO
          05AE
          05AE  ;++
          05AE  ;
          05AE  ; Inputs:
          05AE  ;
          05AE  ;       R3      - base address of adapter's register space
          05AE  ;       R5      - lbn for current piece of transfer
          05AE  ;       R6      - contains 0
          05AE  ;       R8      - size of transfer in bytes
          05AE  ;       R9      - address of the RPB
          05AE  ;       R10     - starting address of transfer (byte offset in first
          05AE  ;                 page ORed with starting map register number)
          05AE  ;
          05AE  ;       FUNC(AP)- I/O operation (IO$_READLBLK or IO$_WRITELBLK only)
          05AE  ;       MODE(AP)- Address interpretation mode (0 = physical, 1 = virtual)
          05AE  ;
          05AE  ; Outputs:
          05AE  ;
          05AE  ;       R0 - status code
          05AE  ;               SS$_NORMAL      - successful transfer
          05AE  ;               SS$_CTRLERR     - fatal controller error
          05AE  ;
          05AE  ;       R3 - must be preserved
          05AE  ;
          05AE  ; NOTE:
          05AE  ;       This routine can be called with four combinations of mapping:
          05AE  ;
          05AE  ;       1) MODE(AP) = physical and PR$_MAPEN = physical.  This is the case
          05AE  ;               when being called from BOOTDRIVR.  We use the made up page
          05AE  ;               table that maps VA = PA for both the port and BDT page tables.
          05AE  ;
          05AE  ;       2) MODE(AP) = virtual and PR$_MAPEN = physical.  This is the case
          05AE  ;               when being called from SYSBOOT to read in SYS.EXE using
          05AE  ;               the real system page table.  We contiue to use the made up
          05AE  ;               page table for the port, but use the real system page table
          05AE  ;               for the BDT.
          05AE  ;
          05AE  ;       3) MODE(AP) = virtual and PR$_MAPEN = virtual.  This is the case
          05AE  ;               when being called from BUGCHECK to read in the non-resident
          05AE  ;               portion of the bugcheck code.  We use the real system page
          05AE  ;               table for both the port and the BDT.
          05AE  ;
          05AE  ;       4) MODE(AP) = physical and PR$_MAPEN = virtual.  This is the case
          05AE  ;               when being called from BUGCHECK to write out all of physical
          05AE  ;               memory from the descriptors in the RPB.  We use the real
          05AE  ;               system page table for the port, but make up a new pagetable
          05AE  ;               for the BDT.  This table is only a page long since the max
          05AE  ;               IOSIZE is 127 pages.  This table is created on the fly with
          05AE  ;               the first entry being the first PFN to be written out.
          05AE  ;--
          05AE
00000010  05AE  FUNC = 16
00000014  05AE  MODE = 20
          05AE
          05AE  PA_DRIVER:                              ; CI/HSC device driver.
          05AE
```

```
                                     05AE              .ENABLE LSB
                                     05AE       ;
                                     05AE       ; Translate the I/O function code into a device-dependent function
                                     05AE       ; code for this disk.
                                     05AE       ;
                                     05AE
                    57    F1 AF  D0  05AE              MOVL    PQB_PTR,R7                      ; Cover the PQB etc
                 54 A7    10 AC  D0  05B2              MOVL    FUNC(AP),TAB_L_HOLE(R7)         ; Temp store function
                    68 A7    04  D0  05B7              MOVL    #4,TAB_L_STATE(R7)              ; Show we are in message
                    52    E8 AF  D0  05BB              MOVL    NXT_MSG,R2                      ; Pick up message buffer
                                     05BF              ASSUME  MSCP$L_BYTE_CNT EQ MSCP$B_OPCODE+4
            54    5A  FE00 8F    AB  05BF              BICW3   #^C<VA$M_BYTE>,R10,R4           ; Get the byte offset
         78 A7    54  8000 8F    A9  05C5              BISW3   #^X8000,R4,-
                                     05CC               TAB_B_BDT+CIBD$W_FLAGS(R7)     ; Set the boff and valid
                 7C A7    58     D0  05CC              MOVL    R8,TAB_B_BDT+CIBD$L_BLEN(R7)    ; .et the byte count
            54    5A    15   09  EF  05D0              EXTZV   #VA$V_VPN,#VA$S_VPN,R10,R4      ; Pick up page table offset
     51   50 A9  80000000 8F     C9  05D5              BISL3   #VA$M_SYSTEM,RPB$L_SVASPT(R9),R1; Virtual
                      35 14 AC   E8  05DE              BLBS    MODE(AP),30$                    ; Pick correct page table
                                     05E2       ;
                                     05E2       ; CASE 1 or 4: MODE = physical
                                     05E2       ;
                    51    74 A7  D0  05E2              MOVL    TAB_L_PAGETBL(R7),R1            ; Physical
                       50    38  DB  05E6              MFPR    #PR$_MAPEN,R0                   ; Check the current addr mode
                         2B 50   E9  05E9              BLBC    R0,30$                          ; Normal, skip out
                                     05EC       ;
                                     05EC       ; CASE 4: MODE = physical, PR$_MAPEN = virtual
                                     05EC       ; Send an INVALIDATE TRANSLATE CACHE to port
                                     05EC       ;
                                 B0  05EC              MOVW    #<PPD$M_RSP@8-                  ; Send a INVALIDAT and
                                     05ED                       !PPD$C_INVTC>,-                ;  set to get it back
           0E A2   0118 8F           05ED                        PPD$B_OPC(R2)                 ;  but leave port alone
                    5C    B5 AF  D0  05F2              MOVL    SAVE_AP,AP                      ; SEND codes expects AP->VMB
                          FE48   31  05F6              BRW     SEND_ANY                        ; Do it
                                     05F9
                                     05F9       INVTC:                                         ; Return
                         0E A2   B4  05F9              CLRW    PPD$B_OPC(R2)                   ; Clean up packet
                    51   6C'AF   DE  05FC              MOVAL   B^PPAGTBL,R1                    ; New page table
               5A    58    F7 8F 78  0600              ASHL    #-9,R8,R10                      ; Number of pages
                       50    51  D0  0605              MOVL    R1,R0                           ; Copy start of table
      80     54  90000000 8F     C9  0608       20$:   BISL3   #<PTE$C_KW!PTE$M_VALID>,R4,(R0)+; Fake up a page table entry
                             54  D6  0610              INCL    R4                              ; Next PFN
                          F3 5A  F5  0612              SOBGTR  R10,20$                         ; Loop until done
                             54  D4  0615              CLRL    R4                              ; Use start of fake table
                                     0617       ;
                                     0617       ; All mapping done. Format the write packet and send it out
                                     0617       ;
           0080 C7   6144        DE  0617       30$:   MOVAL   (R1)[R4],-
                                     061D                       TAB_B_BDT+CIBD$L_SVAPTE(R7)    ; Set it in *** TEMP ***
                    56    28 A2  DE  061D              MOVAL   MSCP$B_OPCODE-SCS$B_PPD(R2),R6  ; Point into message
                       86    21  9A  0621              MOVZBL  #MSCP$K_OP_READ,(R6)+           ; Assume read
                    20    54 A7  B1  0624              CMPW    TAB_L_HOLE(R7),#IO$_WRITELBLK   ; Check for write function
                             04  12  0628              BNEQ    40$                             ; No, do read
                    FC A6    22  90  062A              MOVB    #MSCP$K_OP_WRITE,-4(R6)         ; Set write function code
                       86    58  D0  062E       40$:   MOVL    R8,(R6)+                        ; Set the byte count
                             86  D4  0631              CLRL    (R6)+                           ; Set no offset in buffer
                 86    01    10  9C  0633              ROTL    #16,#1,(R6)+                    ; BUffer name
                 86    0088 C7   D0  0637              MOVL    TAB_L_RCONID(R7),(R6)+          ; RCONID
                 66    55   D0  063C    981         MOVL    R5,(R6)                              ; Set the logical block number
                       2E   B0  063F    982         MOVW    #<-SCS$B_PPD-PPD$C_LENGTH>+MSCP$L_LBN+4,-
                    10 A2       0641    983                  SCS$W_LENGTH-SCS$B_PPD(R2)          ; Set the message length
            5C   FF63 CF    D0  0643    984         MOVL    SAVE_AP,AP                           ; SEND codes expects AP->VMB
                     FF4A   31  0648    985         BRW     Q                                    ; Send and wait
                                064B    986
                                064B    987         .DISABLE LSB
```

```
                      064B          .SBTTL  CI port bootstrap device disconnect
                      064B
                      064B  ;++
                      064B  ; This routine disconnect the boot device after a bugcheck dump.
                      064B  ; It sends an AVAIL packet to the controller, in effect doing a
                      064B  ; dismount of the system device.  It is designed to be called
                      064B  ; only from BUGCHECK immediately after the dump has finished.
                      064B  ; It assumes virtual mapping turned on.
                      064B  ;
                      064B  ; Inputs:
                      064B  ;
                      064B  ;       R9 -->  RPB
                      064B  ;       AP -->  VMB argument list
                      064B  ;
                      064B  ; Outputs:
                      064B  ;
                      064B  ;       R0 - status code
                      064B  ;
                      064B  ;--
                      064B          .ENABLE LSB
                      064B
                      064B  PA_DISC:
                008C  064B          .WORD   ^M<R2,R3,R7>
                      064D
  57   FF51 CF    D0  064D          MOVL    PQB_PTR,R7              ; Cover the PQB etc
  52   FF50 CF    D0  0652          MOVL    NXT_MSG,R2              ; Pick up message buffer addr
   53    60 A9    D0  0657          MOVL    RPB$L_ADPVIR(R9),R3     ; Pick up pointer to adp IO space
            08    9A  065B          MOVZBL  #MSCP$K_OP_AVAIL,-      ; Make drive AVAILable
         28 A2        065D               MSCP$B_OPCODE-SCS$B_PPD(R2)
            1A    B0  065F          MOVW    #<-SCS$B_PPD-PPD$C_LENGTH>+MSCP$B_OPCODE+4,-
         10 A2        0661               SCS$W_LENGTH-SCS$B_PPD(R2)      ; Set the message length
          FF2F    30  0663          BSBW    Q                       ; Send and wait
   04 A3    01    D0  0666          MOVL    #PA_PMC_M_MIN,PA_PMC(R3); Do maint init to shut down port
                  04  066A          RET
                      066B
                      066B  ;
                      066B  ; Data area
                      066B  ;
            0000066C  066B          .=<.+3>&-4                              ; .ALIGN LONG
            0000086C  066C  PPAGTBL:.BLKB   512
            00000EBC  086C  TABLE:  .BLKB   TAB_LEN+512
                      0EBC
            00000EBC  0EBC  PA_DRVSIZ=.-START_DRV
                      0EBC
                      0EBC          .END
```

```
$TABLE                  = 00000000 R    02      NDT$_CI                 = 00000038
ACT1                      00000122 R    03      NXT_MSG                   000005A6 R    03
ACT1A                     00000125 R    03      OPEN                    = 00000003
ACT2                      00000128 R    03      OPEN_VC                   000000AA R    03
ACT3                      0000012F R    03      OPEN_VC_CONT              000000E0 R    03
ACT3A                     0000012E R    03      PA_CNF                    00000000
ACT4                      00000131 R    03      PA_CNF_M_CRD            = 00010000
ACT5                      00000132 R    03      PA_CNF_M_MXTFLT         = 08000000
ACT6                      00000133 R    03      PA_CNF_M_PARFLT         = 80000000
ACT7                      00000135 R    03      PA_CNF_M_URDFLT         = 20000000
ACT8                      00000137 R    03      PA_CNF_M_WSQFLT         = 40000000
ACT9                      00000139 R    03      PA_CNF_M_XMTFLT         = 04000000
ACTION_TABLE              00000122 R    03      PA_CQ0                    00000908
ALLOC_DG                  0000005F R    03      PA_CQ0_M_CQC            = 00000001
BQO$L_TENUSEC           = 0000003E              PA_CQ1                    0000090C
BQO$L_UBDELAY           = 00000042              PA_CQ2                    00000910
BQO$L_UCODE             = 00000028              PA_CQ3                    00000914
BTD$K_HSCCI             = 00000020              PA_C_UCODEST            = 00000400
CIBD$C_BLEN             = 00000004              PA_C_WCSSIZ             = 00000C00
CIBD$L_SVAPTE           = 00000008              PA_DFQ                    00000928
CIBD$W_FLAGS            = 00000000              PA_DFQ_M_DFQC           = 00000001
CLOSED                  = 00000000              PA_DISC                   0000064B R    03
COPY_SYSID                00000050 R    03      PA_DRIVER                 000005AE R    03
DEVNAME                   0000001A R    03      PA_DRVSIZ               = 00000EBC
DG_SIZ                  = 00000060              PA_INIT                   00000150 R    03
DISCARD                   00000076 R    03      PA_MADR                   00000014
DSKDRVNAME                00000000 R    03      PA_MDATR                  00000018
ERROR                     0000056A R    03      PA_MFQ                    0000092C
FUNC                    = 00000010              PA_MTC                    00000930
INVTC                     000005F9 R    03      PA_MTEC                   00000934
IO$_WRITELBLK           = 00000020              PA_PDC                    00000920
L                         0000043A R    03      PA_PEC                    0000091C
LOOP                      0000030D R    03      PA_PEC_M_PEC            = 00000001
MODE                    = 00000014              PA_PESR                   0000093C
MSCP$B_OPCODE           = 00000008              PA_PFAR                   00000938
MSCP$K_OP_AVAIL         = 00000008              PA_PIC                    00000924
MSCP$K_OP_END           = 00000080              PA_PIC_M_PIC            = 00000001
MSCP$K_OP_ONLIN         = 00000009              PA_PMC                    00000004
MSCP$K_OP_READ          = 00000021              PA_PMC_M_MIF            = 00000008
MSCP$K_OP_STCON         = 00000004              PA_PMC_M_MIN            = 00000001
MSCP$K_OP_WRITE         = 00000022              PA_PMC_M_MTD            = 00000002
MSCP$K_ST_AVLBL         = 00000004              PA_PMC_M_PSA            = 00000040
MSCP$K_ST_OFFLN         = 00000003              PA_PPR                    00000940
MSCP$L_BYTE_CNT         = 0000000C              PA_PQBBR                  00000904
MSCP$L_LBN              = 0000001C              PA_PS                     00000900
MSCP$L_MEDIA_ID         = 0000001C              PA_PSR                    00000918
MSCP$Q_TIME             = 00000014              PA_PSR_M_PSC            = 00000001
MSCP$S_MTYP_D0          = 00000005              PA_PS_M_PIC             = 00000008
MSCP$S_MTYP_D1          = 00000005              PA_PS_M_RQA             = 00000001
MSCP$S_ST_MASK          = 00000005              PPAGTBL                   0000066C R    03
MSCP$V_MTYP_D0          = 0000001B              PPD$B_DEF_ST              0000001C
MSCP$V_MTYP_D1          = 00000016              PPD$B_FLAGS               0000000F
MSCP$V_ST_MASK          = 00000000              PPD$B_HWVERS              00000034
MSCP$W_SHDW_UNT         = 00000020              PPD$B_LBDATA              00000012
MSCP$W_STATUS           = 0000000A              PPD$B_LCB_0               00000012
MSG_SNT                   00000570 R    03      PPD$B_LCB_LPORT           00000010
MS_SIZ                  = 00000060              PPD$B_LCB_NPORT           0000000F
```

```
PPD$B_LCB_OPC                     00000011          PPD$M_RSP                  = 00000001
PPD$B_LCB_PORT                    0000000E          PPD$Q_CURTIME                00000048
PPD$B_OPC                         0000000E          PPD$Q_NODENAME               00000040
PPD$B_PORT                        0000000C          PPD$Q_SWINCARN               00000028
PPD$B_PROTOCOL                    0000001A          PPD$Q_XCT_ID                 00000010
PPD$B_RSTATE                      00000025          PPD$S_STSTYP               = 00000003
PPD$B_RST_PORT                    00000024          PPD$T_HWTYPE                 00000030
PPD$B_STATUS                      0000000D          PPD$T_SWTYPE                 00000020
PPD$B_SWFLAG                      0000000B          PPD$T_SWVERS                 00000024
PPD$B_SYSTEMID                    00000014          PPD$V_STSTYP               = 00000005
PPD$B_TYPE                        0000000A          PPD$W_LCB_LEN7               0000000C
PPD$C_ACK                       = 00000002          PPD$W_LENGTH                 00000010
PPD$C_ACK_LEN                   = 00000004          PPD$W_MASK                   00000010
PPD$C_DGREC                     = 00000021          PPD$W_MAXDG                  0000001C
PPD$C_IDREC                     = 0000002B          PPD$W_MAXMSG                 0000001E
PPD$C_INVTC                     = 00000018          PPD$W_MTYPE                  00000012
PPD$C_LB_LENGTH                   00000046          PPD$W_M_VAL                  00000014
PPD$C_LCB_DATA                    00000013          PPD$W_SIZE                   00000008
PPD$C_LENGTH                      00000012          PQB_L_BDT_BASE               0000003C
PPD$C_MIN_DGSIZ                   00000050          PQB_L_BDT_LEN                00000040
PPD$C_MSGREC                    = 00000022          PQB_L_DFRQ_HDR               00000028
PPD$C_REQID                     = 00000005          PQB_L_DQE_LEN                00000030
PPD$C_SCS_MSG                   = 00000004          PQB_L_GPT_BASE               0000004C
PPD$C_SETCKT                    = 00000019          PQB_L_GPT_LEN                00000050
PPD$C_SNDDG                     = 00000001          PQB_L_MFRQ_HDR               0000002C
PPD$C_SNDMSG                    = 00000002          PQB_L_MQE_LEN                00000034
PPD$C_STACK                     = 00000001          PQB_L_SPT_BASE               00000044
PPD$C_STACK_LEN                 = 0000003E          PQB_L_SPT_LEN                00000048
PPD$C_START                     = 00000000          PQB_L_VPQB_BASE              00000038
PPD$C_START_LEN                 = 0000003E          PQB_PTR                      000005A2 R      03
PPD$C_TYPNP                     = 00000005          PQB_Q_CMDQ0                  00000000
PPD$K_LB_LENGTH                   00000046          PQB_Q_CMDQ1                  00000008
PPD$K_LENGTH                      00000012          PQB_Q_CMDQ2                  00000010
PPD$L_BLINK                       00000004          PQB_Q_CMDQ3                  00000018
PPD$L_DG_DISC                     00000028          PQB_Q_RESPQ                  00000020
PPD$L_FLINK                       00000000          PR$_MAPEN                  = 00000038
PPD$L_IN_VCD                      00000018          PR$_SID                    = 0000003E
PPD$L_LBCRC                       00000042          PRTDRVNAME                   0000000D R      03
PPD$L_P0_ACK                      00000010          PTE$C_KW                   = 10000000
PPD$L_P0_NAK                      00000014          PTE$M_VALID                = 80000000
PPD$L_P0_NRSP                     00000018          Q                            00000595 R      03
PPD$L_P1_ACK                      0000001C          REM_NODE                     0000014C R      03
PPD$L_P1_NAK                      00000020          REM_NODE_INDEX               0000014E R      03
PPD$L_P1_NRSP                     00000024          RE_INIT                      00000179 R      03
PPD$L_REC_BOFF                    00000028          ROOT_TABLE                   00000050 R      03
PPD$L_REC_NAME                    00000024          RPB$L_ADPPHY               = 0000005C
PPD$L_RPORT_FCN                   00000020          RPB$L_ADPVIR               = 00000060
PPD$L_RPORT_REV                   0000001C          RPB$L_BADPGS               = 00000104
PPD$L_RPORT_TYP                   00000018          RPB$L_BOOTR2               = 00000024
PPD$L_SND_BOFF                    00000020          RPB$L_BOOTR3               = 00000028
PPD$L_SND_NAME                    0000001C          RPB$L_IOVEC                = 00000034
PPD$L_ST_ADDR                     00000018          RPB$L_PFNCNT               = 0000004C
PPD$L_XCT_LEN                     00000018          RPB$L_SBR                  = 000000AC
PPD$M_CST                       = 00008000          RPB$L_SLR                  = 000000B8
PPD$M_DQI                       = 00001000          RPB$L_SVASPT               = 00000050
PPD$M_NR                        = 00004000          RPB$W_UNIT                 = 00000064
PPD$M_NS                        = 00002000          SAVE_AP                      000005AA R      03
```

```
SCS$B_PPD               = FFFFFFE0               TAB_Q_MFRQ                  00000060
SCS$C_ACCP_REQ          = 00000002               TEMPL_MSG                   0000001C R     03
SCS$C_ACCP_RSPL         = 00000012               TEMPL_MSG_LEN             = 00000034
SCS$C_APPL_MSG          = 0000000A               TIME                      = 00061A80
SCS$C_CON_REQ           = 00000000               TIMEOUT                   = 00000003
SCS$C_CON_REQL          = 00000042               TIMOUT                      000003F3 R     03
SCS$C_CON_RSP           = 00000001               VA$M_BYTE                 = 000001FF
SCS$C_CR_REQ            = 00000008               VA$M_SYSTEM               = 80000000
SCS$C_CR_RSPL           = 0000000E               VA$S_VPN                  = 00000015
SCS$C_STNOMAT           = 0000000A               VA$V_SYSTEM               = 0000001F
SCS$C_STNORMAL          = 00000001               VA$V_VPN                  = 00000009
SCS$L_DST_CONID         = FFFFFFF8               VC_IS_OPEN                  00000462 R     03
SCS$L_SRC_CONID         = FFFFFFFC               VMB$B_SYSTEMID              00000024
SCS$W_CREDIT            = FFFFFFF6               VMB$C_ARGBYTCNT             0000003C
SCS$W_LENGTH            = FFFFFFF0               VMB$L_CI_HIPFN              00000030
SCS$W_MTYPE             = FFFFFFF4               VMB$L_FLAGS                 0000002C
SCS$W_STATUS            = 00000002               VMB$L_HI_PFN                00000010
SCS_MSG                   0000047F R     03      VMB$L_LO_PFN                0000000C
SEND                      00000449 R     03      VMB$Q_FILECACHE             00000004
SENDX                     0000044E R     03      VMB$Q_NODENAME              00000034
SEND_ACK                  00000097 R     03      VMB$Q_PFNMAP                00000014
SEND_ANY                  00000441 R     03      VMB$Q_UCODE                 0000001C
SEND_DG                   00000445 R     03      X                           00000122 R     03
SEND_MSG                  0000043D R     03
SEND_STACK                000000AE R     03
SEND_START                000000A1 R     03
SETCRT                    000002E7 R     03
SIZ...                  = 00000001
SPIN                      000000AC R     03
SPIN_CONT                 000000F3 R     03
SS$_CTRLERR             = 00000054
SS$_NORMAL              = 00000001
START_DRV                 00000000 R     03
STATE_TABLE               0000013C R     03
ST_RECV                 = 00000002
ST_SENT                 = 00000001
TABLE                     0000086C R     03
TAB_B_BDT                 00000078
TAB_LEN                 = 00000450
TAB_L_HOLE                00000054
TAB_L_LCONID              0000008C
TAB_L_PAGETBL             00000074
TAB_L_RCONID              00000088
TAB_L_RSTAID              00000070
TAB_L_STATE               00000068
TAB_L_TIMER               0000006C
TAB_PKT0                  00000090
TAB_PKT1                  000000F0
TAB_PKT2                  00000150
TAB_PKT3                  000001B0
TAB_PKT4                  00000210
TAB_PKT5                  00000270
TAB_PKT6                  000002D0
TAB_PKT7                  00000330
TAB_PKT8                  00000390
TAB_PKT9                  000003F0
TAB_Q_DFRQ                00000058
```

```
                                                       +----------------+
                                                       ! Psect synopsis !
                                                       +----------------+

PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (     0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
$ABS$                           00000944  (  2372.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
BOOTDRIVR_4                     00000028  (    40.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
BOOTDRIVR_2                     00000EBC  (  3772.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC BYTE

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

Phase                        Page faults    CPU Time       Elapsed Time
-----                        -----------    --------       ------------
Initialization                        36    00:00:00.10    00:00:00.74
Command processing                   156    00:00:00.83    00:00:03.49
Pass 1                               523    00:00:22.45    00:00:39.66
Symbol table sort                      2    00:00:03.20    00:00:05.17
Pass 2                               183    00:00:04.37    00:00:08.05
Symbol table output                   36    00:00:00.27    00:00:00.29
Psect synopsis output                  2    00:00:00.03    00:00:00.03
Cross-reference output                 0    00:00:00.00    00:00:00.00
Assembler run totals                 940    00:00:31.25    00:00:57.43

The working set limit was 2000 pages.
122365 bytes (239 pages) of virtual memory were used to buffer the intermediate code.
There were 110 pages of symbol table space allocated to hold 2020 non-local and 71 local symbols.
1033 source lines were read in Pass 1, producing 17 object records in Pass 2.
32 pages of virtual memory were used to define 29 macros.

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

Macro library name                                 Macros defined
------------------                                 --------------
_$255$DUA28:[SHRLIB]PALIB.MLB;1                                 2
_$255$DUA28:[BOOTS.OBJ]BOOTS.MLB;1                              3
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                                 10
_$255$DUA28:[SYSLIB]STARLET.MLB;2                               8
TOTALS (all libraries)                                         23

2197 GETS were required to define 23 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:PABTDRIVR/OBJ=OBJ$:PABTDRIVR MSRC$:PABTDRIVR/UPDATE=(ENH$:PABTDRIVR)+EXECML$/LIB+LIB$:BOOTS.MLB/LIB+SHRLIB$:PALIB/LIB
```

0039 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

READLBN
LIS

RTFILREAD
LIS

SHARE
LIS

QVSS
LIS

RXBTDRIVR
LIS

MBBTDRIVR
LIS

SCSLOADER
LIS

RMSCONIO
LIS

READDRIV
LIS

PABTDRIVR
LIS

PUBTDRIVR
LIS

PUTERROR
LIS

READPRMPT
LIS