```
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDD          DDD RRR          RRR       III       VVV         VVV  EEE              RRR          RRR
DDD          DDD RRR          RRR       III       VVV         VVV  EEE              RRR          RRR
DDD          DDD RRR          RRR       III       VVV         VVV  EEE              RRR          RRR
DDD          DDD RRR          RRR       III       VVV         VVV  EEE              RRR          RRR
DDD          DDD RRR          RRR       III       VVV         VVV  EEE              RRR          RRR
DDD          DDD RRR          RRR       III       VVV         VVV  EEE              RRR          RRR
DDD          DDD RRRRRRRRRRRR           III       VVV         VVV  EEEEEEEEEEE      RRRRRRRRRRRR
DDD          DDD RRRRRRRRRRRR           III       VVV         VVV  EEEEEEEEEEE      RRRRRRRRRRRR
DDD          DDD RRRRRRRRRRRR           III       VVV         VVV  EEEEEEEEEEE      RRRRRRRRRRRR
DDD          DDD RRR   RRR              III       VVV         VVV  EEE              RRR   RRR
DDD          DDD RRR   RRR              III       VVV         VVV  EEE              RRR   RRR
DDD          DDD RRR   RRR              III       VVV         VVV  EEE              RRR   RRR
DDD          DDD RRR      RRR           III          VVV   VVV     EEE              RRR      RRR
DDD          DDD RRR      RRR           III          VVV   VVV     EEE              RRR      RRR
DDD          DDD RRR      RRR           III          VVV   VVV     EEE              RRR      RRR
DDDDDDDDDDDD     RRR         RRR     IIIIIIIII          VVV       EEEEEEEEEEEEEEE  RRR         RRR
DDDDDDDDDDDD     RRR         RRR     IIIIIIIII          VVV       EEEEEEEEEEEEEEE  RRR         RRR
DDDDDDDDDDDD     RRR         RRR     IIIIIIIII          VVV       EEEEEEEEEEEEEEE  RRR         RRR
```

```
PPPPPPPP      AAAAAA      IIIIII    NN      NN    IIIIII    TTTTTTTTTT
PPPPPPPP      AAAAAA      IIIIII    NN      NN    IIIIII    TTTTTTTTTT
PP      PP   AA      AA      II     NN      NN      II          TT
PP      PP   AA      AA      II     NN      NN      II          TT
PP      PP   AA      AA      II     NNNN    NN      II          TT
PP      PP   AA      AA      II     NNNN    NN      II          TT
PPPPPPPP     AA      AA      II     NN  NN  NN      II          TT
PPPPPPPP     AA      AA      II     NN  NN  NN      II          TT
PP           AAAAAAAAAA      II     NN    NNNN      II          TT
PP           AAAAAAAAAA      II     NN    NNNN      II          TT
PP           AA      AA      II     NN      NN      II          TT          ....
PP           AA      AA      II     NN      NN      II          TT          ....
PP           AA      AA    IIIIII   NN      NN    IIIIII        TT          ....
PP           AA      AA    IIIIII   NN      NN    IIIIII        TT          ....

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

(2) 288 DEFINITIONS
(3) 325 TABLES OF INITIALIZATION DATA
(4) 501 UNIT_INIT
(5) 573 CONTROLLER INIT
(9) 651 BUILD PDT
(10) 780 BUILD TEMPLATE LOOPBACK DG
(11) 817 SCS LAYER INITIIALIZATION
(12) 835 INIT POLLER TIMER AND TRACE FUNCTION
(13) 856 INI$PORT, INITIALIZE PORT
(14) 1102 TEST_SHUTDOWN, CHECK IF PORT SHOULD
(14) 1103 - BE LEFT OFFLINE
(15) 1185 PORT INITIALIZATION ERRORS
(17) 1381 INI$FORK

```
0000     1          .TITLE  PAINIT
0000     2          .IDENT  'V04-001'
0000     3
0000     4  ;*****************************************************************************
0000     5  ;*                                                                           *
0000     6  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     7  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     8  ;*  ALL RIGHTS RESERVED.                                                     *
0000     9  ;*                                                                           *
0000    10  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    11  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    12  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    13  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    14  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    15  ;*  TRANSFERRED.                                                             *
0000    16  ;*                                                                           *
0000    17  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    18  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    19  ;*  CORPORATION.                                                             *
0000    20  ;*                                                                           *
0000    21  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    22  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    23  ;*                                                                           *
0000    24  ;*                                                                           *
0000    25  ;*****************************************************************************
0000    26  ;
0000    27  ;++
0000    28  ;
0000    29  ; FACILITY:
0000    30  ;
0000    31  ;       VAX/VMS EXECUTIVE, I/O DRIVERS
0000    32  ;
0000    33  ; ABSTRACT: CI PORT INITIALIZATION
0000    34  ;
0000    35  ; AUTHOR:  N. KRONENBERG,  MAY 1981
0000    36  ;
0000    37  ; MODIFIED BY:
0000    38  ;
0000    39  ;       V04-001 NPK3066         N. Kronenberg           9-Sep-1984
0000    40  ;               Add flags INI$CPU/PORT_REV.  Flags = 1/0 if ucode is
0000    41  ;               okay/insufficient.  Used to trigger UCODEREV bugcheck
0000    42  ;               rather than usual CIPORT bugcheck if bugcheck is needed.
0000    43  ;               Set INI$CPU_REV to okay just before CPU rev check; clear
0000    44  ;               if check fails prior to calling CLEANUP_PDT.
0000    45  ;               Set INI$PORT_REV when port is successfully init'ed assuming
0000    46  ;               its ucode is okay.  Clear in PACONFIG when we have
0000    47  ;               checked port ucode rev and determined it is bad.
0000    48  ;
0000    49  ;       V03-034 NPK3064         N. Kronenberg           21-Aug-1984
0000    50  ;               On cpu powerfail recovery (when port may still be
0000    51  ;               alive if it was an unswitched power failure), min
0000    52  ;               the port before dropping ipl to reinit.
0000    53  ;
0000    54  ;       V03-033 NPK3062         N. Kronenberg           10-Aug-1984
0000    55  ;               Fix yet another bug in leaving port offline, but
0000    56  ;               continuing to run the system.
0000    57  ;
```

```
0000     58 ;       V03-032 NPK3061         N. Kronenberg           9-Aug-1984
0000     59 ;               Fix CLUB check in CLEANUP_PDT.
0000     60 ;
0000     61 ;       V03-031 NPK3060         N. Kronenberg           1-Aug-1984
0000     62 ;               Init local port status to have loop back datagrams
0000     63 ;               enabled.
0000     64 ;
0000     65 ;       V03-030 NPK3059         N. Kronenberg           25-Jul-1984
0000     66 ;               Fix problems with deallocating PDT before deciding
0000     67 ;               to call BUGCHECK with a nonex PDT address.
0000     68 ;
0000     69 ;       V03-029 NPK3057         N. Kronenberg           23-Jul-1984
0000     70 ;               Eliminate override of max port reinit retry count
0000     71 ;               if system disk or clustering requires CI.
0000     72 ;               Now port unconditionally shutdown if retry count
0000     73 ;               exhausted.  Difference is that now, if clustering
0000     74 ;               or if system disk available via the failing port,
0000     75 ;               system bugchecks unless there is another SCS speaking
0000     76 ;               port left.
0000     77 ;               Move the above check for system bugcheck to CLEANUP_PDT--
0000     78 ;               previously the analogous check was in TEST_SHUTDOWN
0000     79 ;               which was called only on each reinit.
0000     80 ;
0000     81 ;       V03-028 NPK3055         N. Kronenberg           14-Jul-1984
0000     82 ;               Add init of PDT$W_STDGUSED/DYN in INI$PORT.
0000     83 ;               Put 11/750 SID in R1 instead of R0 and pass to new
0000     84 ;               error logging routine, ELOG$CPU_REV.
0000     85 ;               Leave port offline if 11/750 ucode not up to at least
0000     86 ;               97 (base 10.)
0000     87 ;               Make CLEANUP_PDT do maint init on port just in case.
0000     88 ;
0000     89 ;       v03-027 NPK3054         N. Kronenberg           24-Jun-1984
0000     90 ;               Log error if CPU is 11/750 and rev level is insufficient
0000     91 ;               to support ci port.  Ucode rev must be 97 (base 10)
0000     92 ;               or greater.
0000     93 ;
0000     94 ;       V03-026 NPK3048         N. Kronenberg           5-Apr-1984
0000     95 ;               In TEST_SHUTDOWN, override retry max of 10 if this
0000     96 ;               system is part a cluster.  I.e., never leave the
0000     97 ;               port offline, because it may prevent the cluster from
0000     98 ;               running and will certainly prevent this system from
0000     99 ;               doing anything useful.
0000    100 ;
0000    101 ;       V03-024 NPK3047         N. Kronenberg           15-Mar-1984
0000    102 ;               For VAX 8600, set system hardware type appropriately.
0000    103 ;               When building a PDT, add it to the list of SCS speaking
0000    104 ;               PDT's.  When removing a PDT, remove it from the list.
0000    105 ;               Init new PDT vector, PDT$L_STOP_VCS.
0000    106 ;               Near the end of port initialization call CNF$CALC_POLLSW
0000    107 ;               to compute the estimated time to do a full sweep of the
0000    108 ;               configuration poller.
0000    109 ;
0000    110 ;       V03-023 TMK0004         Todd M. Katz            07-Mar-1984
0000    111 ;               It is no longer necessary to broadcast messages to _OPA0 when
0000    112 ;               it is discovered, during controller initialization, that
0000    113 ;               SCSSYSTEMID has not been initialized to a non-zero value and
0000    114 ;               that the port is going to be left offline. This is because the
```

```
0000   115 ;               error logging of this error condition has been modified to
0000   116 ;               notice that the existance of this error should also be broadcast
0000   117 ;               to _OPA0, and does so.
0000   118 ;
0000   119 ;               I have modified the routine TEST_SHUTDOWN so that the port
0000   120 ;               re-initialization message that is broadcast to _OPA0 includes
0000   121 ;               the number of retries left.
0000   122 ;
0000   123 ;       V03-022 TMK0003         Todd M. Katz            21-Feb-1984
0000   124 ;               Change unit and port initialization so that they proceed at fork
0000   125 ;               IPL instead of at IPL$_POWER. This requires these changes:
0000   126 ;
0000   127 ;               1. Add a new routine INI$FORK. This routine is assumed to be
0000   128 ;                  called at elevated IPL with a routine address in R3 which is
0000   129 ;                  to be jumped to at fork IPL. INI$FORK will extract the fork
0000   130 ;                  block from the appropriate fork queue in an atomic fashion,
0000   131 ;                  if it has to, and create a fork process before returning
0000   132 ;                  to its caller. When the fork process resumes, it does so
0000   133 ;                  within INI$FORK, which proceeds to jump to the routine
0000   134 ;                  address passed to it as input. Throughout this procedure
0000   135 ;                  proper use is made of the fork block interlock bit.
0000   136 ;
0000   137 ;               2. If the unit initialization routine has been called and the
0000   138 ;                  port has not yet been initialized then all miscellaneous
0000   139 ;                  errors within the configuration register are cleared, device
0000   140 ;                  interrupts are disabled by placing the port in the
0000   141 ;                  un-initialized state, and the routine INI$FORK is called
0000   142 ;                  so that the remained of the unit initialization maybe done
0000   143 ;                  at fork IPL.
0000   144 ;
0000   145 ;               3. Because port initialization proceeds at fork IPL there is
0000   146 ;                  no longer any need to fork in order to print out messages
0000   147 ;                  to OPA0.
0000   148 ;
0000   149 ;               4. Because port initialization proceeds at fork IPL,
0000   150 ;                  EXE$ALONONPAGED maybe called to allocate whatever non-paged
0000   151 ;                  pool needs to be allocated. This means that the routine
0000   152 ;                  HIPL_ALLPOOL (INI$HIPLALC) maybe deleted, and that the funny
0000   153 ;                  games that were being played with the IPL value in the pool
0000   154 ;                  header area, so that the allocation of free datagrams and
0000   155 ;                  sequence messages could proceed at IPL$_POWER, can be
0000   156 ;                  stopped.
0000   157 ;
0000   158 ;               5. Because port initialization used to proceed at IPL$_POWER, it
0000   159 ;                  never mattered when device interrupts were being enabled.
0000   160 ;                  However, port initialization is now being done at fork IPL
0000   161 ;                  so it has become important not to enable device interrupts
0000   162 ;                  until after everything else has been done and just before
0000   163 ;                  port initialization terminates.
0000   164 ;
0000   165 ;               6. It becomes an implicit assumption, that INI$PORT is only
0000   166 ;                  called at fork IPL with device interrupts disabled, and with
0000   167 ;                  no outstanding interrupts.
0000   168 ;
0000   169 ;       V03-021 TMK0002         Todd M. Katz            17-Feb-1984
0000   170 ;               Change the text of the message that is printed out on the
0000   171 ;               operator's console when it is discovered that SCSSYSTEMID
```

```
0000   172 ;           is unitialized, and has a value of 0.
0000   173 ;
0000   174 ;       V03-020 NPK3044         N. Kronenberg           6-Feb-1984
0000   175 ;               Replace queuing of 3 gratituous datagrams to the port
0000   176 ;               free queue (to fill the cache) with queuing of an
0000   177 ;               additional SCS$GW_PAPPDDG datagrams.  The additional
0000   178 ;               datagrams are intended to handle error log datagrams
0000   179 ;               not associated with any particular connection.
0000   180 ;
0000   181 ;       V03-019 TMK0001         Todd M. Katz            27-Jan-1984
0000   182 ;               Before allocating the PDT, check for a SCSSYSTEMID of 0.
0000   183 ;               If such a SCSSYSTEMID is found, log the error condition,
0000   184 ;               notify the operator's console via an appropriate set of
0000   185 ;               messages, and keep the port off-line.
0000   186 ;
0000   187 ;       V03-018 NPK3039         N. Kronenberg           11-Jan-1984
0000   188 ;               Remove return of top unused portion of pool before PDT
0000   189 ;               back to pool.  No return can be done if PDT allocated
0000   190 ;               from LRP, so never return.
0000   191 ;
0000   192 ;       V03-017 NPK3037         N. Kronenberg           11-Nov-1983
0000   193 ;               Comment inputs to INI$PORT.
0000   194 ;               Add check to INI$PORT that command queues and response
0000   195 ;               queue are empty prior to starting port.  If queues
0000   196 ;               aren't empty, attempt recovery by setting them empty.
0000   197 ;
0000   198 ;       V03-016 NPK3035         N. Kronenberg           21-Oct-1983
0000   199 ;               Fix calculation of global page table length.
0000   200 ;
0000   201 ;       V03-015 TCM0002         Trudy C. Matthews       19-Aug-1983
0000   202 ;               Add SUPERSTAR-specific path to CPU-dependent code that sets
0000   203 ;               CPU type and port device type.
0000   204 ;
0000   205 ;       V03-014 NPK3029         N. Kronenberg           14-Jul-1983
0000   206 ;               Numerous enhancements for V4.0.
0000   207 ;               Add fork process call, SENDRGDG, to SCS offset table.
0000   208 ;               Set max block xfer byte count in PDT.
0000   209 ;               Allow sanity timer to be enabled.
0000   210 ;               Add routine TEST_SHUTDOWN to check if port can be
0000   211 ;               reinitialized or must be left offline and to print
0000   212 ;               operator warning if appropriate.
0000   213 ;               Add init of fork ipl for msg fork block in ucb.
0000   214 ;               Add $PAUCBDEF and $DDBDEF.
0000   215 ;
0000   216 ;       V03-013 NPK3024         N. Kronenberg           18-May-1983
0000   217 ;               Add comments explaining variable network header.
0000   218 ;
0000   219 ;       V03-012 KTA3046         Kerbey T. Altmann       03-Apr-1983
0000   220 ;               Redo for SCS/PPD split.
0000   221 ;
0000   222 ;       V03-011 TCM0001         Trudy C. Matthews       29-Feb-1983
0000   223 ;               Added an 11/790-specific path to CPUDISP macro which sets
0000   224 ;               CPU type and port device type.
0000   225 ;
0000   226 ;       V03-010 NPK3021         N. Kronenberg           28-Feb-1983
0000   227 ;               Fix setting of 'V750' cpu type.
0000   228 ;
```

```
0000   229 ;       V03-009 NPK3010         N. Kronenberg           9-Nov-1982
0000   230 ;               Modify BUILD_PDT to set CI PDT type;  modify INI$PORT
0000   231 ;               to set local port number in PDT rather than maximum
0000   232 ;               port number on this CI.
0000   233 ;
0000   234 ;       V03-008 NPK3009         N. Kronenberg           2-Nov-1982
0000   235 ;               Always fill in BDT info in newly created PDT in case
0000   236 ;               multiple ports per system.
0000   237 ;
0000   238 ;       V03-007 NPK3004         N. Kronenberg           30-Jul-1982
0000   239 ;               Add setting of CI750 device type in UCB.  Add ASCII
0000   240 ;               CPU type for start handshake.  Add check for 11/750
0000   241 ;               status, NOCI, before initializing port.
0000   242 ;
0000   243 ;       V03-006 NPK3001         N. Kronenberg           25-Jun-1982
0000   244 ;               Fix to allow loading of ucode into rom/ram ports.
0000   245 ;               Enable read back of loaded ucode to check it.
0000   246 ;
0000   247 ;       V03-005 ROW0101         Ralph O. Weber          10-JUN-1982
0000   248 ;               Change ordering of port initialization operations to that
0000   249 ;               proposed by Barry Odonoghue in his 9 June mail to Nancy.
0000   250 ;               The proposed order is as follows (the parenthetical letters
0000   251 ;               indicate the order previously employed by this driver):
0000   252 ;                 1(a)  Set PIC
0000   253 ;                 2(b)  Wait for MIF
0000   254 ;                 3(c)  Check that only PIC is set in PSR
0000   255 ;                 4(g)  Release PSR to port (this should clear MIF)
0000   256 ;                 5(e)  Enable interrupts
0000   257 ;                 6(f)  Write PECR
0000   258 ;               The intent of the new ordering is to prevent unexpected
0000   259 ;               interrupts which can occur if interrupts are enabled while MIF
0000   260 ;               is set as the result of PIC (Port Initialization Complete).
0000   261 ;               This change will be in a new driver image shipped in V3.1.
0000   262 ;
0000   263 ;       V03-004 ROW0100         Ralph O. Weber          9-JUN-1982
0000   264 ;               Add a high-IPL allocation jacket around the code which
0000   265 ;               allocates and queues extra datagrams for start handshakes
0000   266 ;               and extra message buffers to fill the port cache.  This jacket
0000   267 ;               will allow the calls to EXE$ALONONPAGED, called within the
0000   268 ;               SCS$ routines, to be made from IPL$_POWER in the same way that
0000   269 ;               a similar call is made within HIPL_ALLPOOL.
0000   270 ;               This change will be in a new driver image shipped in V3.1.
0000   271 ;
0000   272 ;       V03-003 ROW0094         Ralph O. Weber          7-JUN-1982
0000   273 ;               Add calls to error logging routines in BUILD_PDT, BUILD_TLB,
0000   274 ;               BUILD_BDT, and INI$PORT.  Add necessary reference to $PAERDEF
0000   275 ;               macro.  Correct branch destination out of range, caused by new
0000   276 ;               code, in BUILD_BDT at BNEQ INIT_CRB.
0000   277 ;               This change will be in a new driver image shipped in V3.1.
0000   278 ;
0000   279 ;       V03-002 NPK2019         N. Kronenberg           6-Apr-1982
0000   280 ;               Fixed bug in setting of device type in UCB.
0000   281 ;               Remove unit init JSB to INI$BRK.
0000   282 ;
0000   283 ;       V03-001 NPK2016         N. Kronenberg           18-Mar-1982
0000   284 ;               Fixed .TITLE
0000   285 ;
```

```
                        0000    286 ;--
```

```
                        .SBTTL  DEFINITIONS

                ;
                ; Set PSECT to driver code:
                ;

                        .PSECT  $$$115_DRIVER,LONG

                ;
                ; System definitions (LIB.MLB):
                ;

                        .nocross
                        $CRBDEF                         ; Channel Request Block format
                        $CXBDEF                         ; Complex Buffer format
                        $DCDEF                          ; Device type codes
                        $DDBDEF                         ; Device Data Block format
                        $DYNDEF                         ; Structure type codes
                        $IPLDEF                         ; Define interrupt priorities
                        $MCHKDEF                        ; Protect from machine check definitions
                        $PDTDEF                         ; Port Descriptor Table format
                        $PRDEF                          ; Internal processor register definitions
                        $SSDEF                          ; System service success codes
                        $UCBDEF                         ; Unit Control Block definitions
                        $VECDEF                         ; CRB vector dispatch block offsets

                ;
                ; PADRIVER definitions (PALIB.MLB):
                ;

                        $PAERDEF                        ; Port driver error code values
                        $PAPDTDEF                       ; Port -specific PDT extension
                        $PAREGDEF                       ; CI port register definitions
                        $PAUCBDEF                       ; PA specific extension to UCB
                        $PPDDEF                         ; PPD message layer
```

```
0000    325          .SBTTL  TABLES OF INITIALIZATION DATA
0000    326
0000    327 ;+
0000    328 ; The following table gives word offsets for fork process SCS calls.
0000    329 ; Offsets are relative to the address of the controller initialization
0000    330 ; routine, PA$CTLINIT.
0000    331 ;-
0000    332
0000    333 ;
0000    334 ; Macro to generate the table and ASSUME statements about PDT format:
0000    335 ;
0000    336
0000    337          .MACRO  SCS_OFFSET_TAB  ENTRY_LIST
0000    338
0000    339           $$$ENTRYNUM=0                                ; No entries in table yet
0000    340           .IRP    ENTRY,ENTRY_LIST                     ; For each entry in the list...
0000    341           .WORD   <FPC$'ENTRY'-PA$CTLINIT>             ;  insert offset from ctl init,
0000    342           .IF     NE $$$ENTRYNUM                       ;  and for entries after the 1st
0000    343           ASSUME $$$PREV+4 EQ PDT$L_'ENTRY'            ;  specify assumed PDT adjacency
0000    344           .ENDC                                        ;
0000    345           $$$PREV=PDT$L_'ENTRY'                        ; Set previous entry as this entry
0000    346           $$$ENTRYNUM=$$$ENTRYNUM+1                    ; Step entry count
0000    347           .ENDR                                        ;
0000    348
0000    349           ASSUME $$$PREV+4 EQ PDT$C_SCSEND             ; Final PDT assumption
0000    350           .WORD  0                                     ; Offset table terminator
0000    351
0000    352          .ENDM   SCS_OFFSET_TAB                        ;
0000    353
0000    354 ;
0000    355 ; Table itself:
0000    356 ;
0000    357
0000    358 PA$SCSOFFSET::
0000    359
0000    360          SCS_OFFSET_TAB  <-                           ; Invoke macro to define offsets
0000    361                  ACCEPT,-
0000    362                  ALLOCDG,-
0000    363                  ALLOCMSG,-
0000    364                  CONNECT,-
0000    365                  DEALLOCDG,-
0000    366                  DEALLOMSG,-
0000    367                  DEALRGMSG,-
0000    368                  DCONNECT,-
0000    369                  MAP,-
0000    370                  MAPBYPASS,-
0000    371                  MAPIRP,-
0000    372                  MAPIRPBYP,-
0000    373                  QUEUEDG,-
0000    374                  QUEUEMDGS,-
0000    375                  RCHMSGBUF,-
0000    376                  RCLMSGBUF,-
0000    377                  REJECT,-
0000    378                  REQDATA,-
0000    379                  SENDDATA,-
0000    380                  SENDDG,-
0000    381                  SENDMSG,-
```

```
0000    382                     SNDCNTMSG,-
0000    383                     UNMAP,-
0000    384                     READCOUNT,-
0000    385                     RLSCOUNT,-
0000    386                     MRESET,-
0000    387                     MSTART,-
0000    388                     MAINTFCN,-
0000    389                     SENDRGDG,-
0000    390                     STOP_VCS,-
0000    391                     >
003E    392
003E    393 ;+
003E    394 ; The following table is a list of word offsets from the CI configuration
003E    395 ; register of CI register addresses to be kept in the PDT for quick access.
003E    396 ;-
003E    397
003E    398 ;
003E    399 ; Macro to generate table:
003E    400 ;
003E    401
003E    402         .MACRO  REG_OFFSET_TAB  REG_LIST
003E    403
003E    404          $$$REGNUM=0                                ; # table entries =0
003E    405          .IRP    REG REG_LIST                       ; For each register in list,
003E    406          .WORD   <PA_'REG'-PA_CNF>                  ;  enter offset from config reg
003E    407          .IF     NE $$$REGNUM                       ;  and for entries after first,
003E    408          ASSUME $$$PREV+4 EQ PDT$L_'REG'            ;  verify PDT adjacency
003E    409          .ENDC                                      ;
003E    410
003E    411          $$$PREV =PDT$L_'REG'                       ; Set this PDT entry to previous
003E    412          $$$REGNUM=$$$REGNUM+1                      ; Step entry counter
003E    413          .ENDR                                      ;
003E    414
003E    415          ASSUME $$$PREV+4 EQ PDT$C_PAREGEND         ; Verify final PDT offset
003E    416
003E    417          .WORD  -1                                  ; Table terminator
003E    418
003E    419         .ENDM   REG_OFFSET_TAB
003E    420
003E    421 ;
003E    422 ; Table itself:
003E    423 ;
003E    424
003F    425 PA$REGOFFSET::
003E    426
003E    427         REG_OFFSET_TAB  <-                         ; Invoke macro to define offsets
003E    428                 CNF,-                              ;  Configuration register
003E    429                 PMC,-                              ;  Port maint control reg
003E    430                 PS,-                               ;  Port status register
003E    431                 CQ0,-                              ;  Command queue 0 control
003E    432                 CQ1,-                              ;  Command queue 1 control
003E    433                 PSR,-                              ;  Port status release register
003E    434                 DFQ,-                              ;  Datagram free queue control
003E    435                 MFQ,-                              ;  Message free queue control
003E    436                 MTC,-                              ;  Maint timer control
003E    437                 PFAR,-                             ;  Fail address register
003E    438                 PPR,-                              ;  Port parameter register
```

```
                                003E   439                >
                                0056   440                .cross
                                0056   441 ;
                                0056   442 ; Processor type in ASCII for start handshakes:
                                0056   443 ;
                                0056   444
                                0056   445 INI$T_HWTYPE::
                                0056   446
                      30 38 37 56 0056   447                .ASCII  'V780'                          ; Assume 'V780' to start with
                                005A   448
                      00000061  005A   449 MIN_750_REV = 97                                        ; Minimum 11/750 CPU ucode
                                005A   450                                                        ;  rev level to support ci750
                                005A   451
                                005A   452 ;
                                005A   453 ; Messages to send to _OPA0 on serious port errors:
                                005A   454 ;
                                005A   455
                      0000000D  005A   456 CR       = 13                                        ; ASCII for carriage return,
                      0000000A  005A   457 LF       = 10                                        ;  linefeed,
                      00000007  005A   458 BELL     = 7                                         ;  and bell
                                005A   459
                                005A   460 INI$MSG_INIT::
                                005A   461
43 20 2C 30 78 41 50 25 07 0A 0D 00' 005A   462                .ASCIC  <CR><LF><BELL>'%PAx0, CI Port is Reinitializing ( xxx Retries Left).
65 52 20 73 69 20 74 72 6F 50 20 49  0066
67 6E 69 7A 69 6C 61 69 74 69 6E 69  0072
69 72 74 65 52 20 78 78 78 20 28 20  007E
43 20 20 2E 29 74 66 65 4C 20 73 65  008A
72 72 45 20 65 68 74 20 6B 63 65 68  0096
         0A 0D 2E 67 6F 4C 20 72 6F  00A2
                              50     005A
                                00AB   463
                                00AB   464 INI$MSG_OFFL::
                                00AB   465
43 20 2C 30 78 41 50 25 07 0A 0D 00' 00AB   466                .ASCIC  <CR><LF><BELL>'%PAx0, CI Port is going Offline.'<CR><LF>
6F 67 20 73 69 20 74 72 6F 50 20 49  00B7
2E 65 6E 69 6C 66 66 4F 20 67 6E 69  00C3
                        0A 0D        00CF
                           25        00AB
                                00D1   467
                      00000027  00D1   468 RETRY_OFFSET    = 39                                  ; Byte offset to retry count
                                00D1   469                                                        ; numerical field in port
                                00D1   470                                                        ; re-initialization message
                                00D1   471
                      00000006  00D1   472 CTRLR_NAME = 6                                       ; Byte offset to device
                                00D1   473                                                        ;  controller letter (x)
                                00D1   474                                                        ;  in above msgs
                                00D1   475
                                00D1   476
                                00D1   477 ;
                                00D1   478 ; Polynomial table used to calculate CRC for loopback datagram:
                                00D1   479 ;
                                00D1   480
                                00D1   481 CRC_TABLE:
                                00D1   482
                      00000000  00D1   483                .LONG   0
                      1DB71064  00D5   484                .LONG   ^X1DB71064
```

```
3B6E20C8  00D9   485          .LONG   ^X3B6E20C8
26D930AC  00DD   486          .LONG   ^X26D930AC
76DC4190  00E1   487          .LONG   ^X76DC4190
6B6B51F4  00E5   488          .LONG   ^X6B6B51F4
4DB26158  00E9   489          .LONG   ^X4DB26158
5005713C  00ED   490          .LONG   ^X5005713C
EDB88320  00F1   491          .LONG   ^XEDB88320
F00F9344  00F5   492          .LONG   ^XF00F9344
D6D6A3E8  00F9   493          .LONG   ^XD6D6A3E8
CB61B38C  00FD   494          .LONG   ^XCB61B38C
9B64C2B0  0101   495          .LONG   ^X9B64C2B0
86D3D2D4  0105   496          .LONG   ^X86D3D2D4
A00AE278  0109   497          .LONG   ^XA00AE278
BDBDF21C  010D   498          .LONG   ^XBDBDF21C
          0111   499
```

```
                     0111    501          .SBTTL  UNIT_INIT
                     0111    502
                     0111    503 ;+
                     0111    504 ; The device type is set in the UCB depending on what CPU we are running
                     0111    505 ; on.  If this is an 11/780, then the device type is CI780; if 11/750,
                     0111    506 ; then the device type is CI750.  If the CPU is other than an 11/780,
                     0111    507 ; alter the ASCII CPU type accordingly.  If this is some other CPU, we really
                     0111    508 ; shouldn't ever get here, so we just exit leaving the unit offline so
                     0111    509 ; it can't be used.
                     0111    510 ;
                     0111    511 ; Then the device is initialized by calling subroutine INIT_CTLR.
                     0111    512 ;
                     0111    513 ; Inputs:
                     0111    514 ;
                     0111    515 ;       R3                      -Address of PA configuration reg
                     0111    516 ;       R4                      -Same as R3
                     0111    517 ;       R5                      -Addr of UCB
                     0111    518 ;
                     0111    519 ; Outputs:
                     0111    520 ;
                     0111    521 ;       R0-R3                   -Destroyed
                     0111    522 ;       Other registers         -Preserved
                     0111    523 ;
                     0111    524 ;       INI$T_HWTYPE            -Set to proper CPU type in ASCII
                     0111    525 ;-
                     0111    526
                     0111    527 ASSUME  DT$_CI750  EQ  DT$_CI780+1
                     0111    528
                     0111    529          .ENABL  LSB
                     0111    530
                     0111    531 PA$UNITINIT::
                     0111    532
          41 A5  01  90  0111    533          MOVB    #DT$_CI780,UCB$B_DEVTYPE(R5)  ; Set the device type to CI780
                     0115    534
                     0115    535          CPUDISP <<780,CI_780>, -        ; * Dispatch on CPU type *
                     0115    536                   <750,CI_750>, -
                     0115    537                   <730,OTHER_CPU>, -
                     0115    538                   <790,CI_790>, -
                     0115    539                   <785,CI_785>>
                     0147    540
                     0147    541 CI_750:
                     0147    542
        41 A5  96    0147    543          INCB    UCB$B_DEVTYPE(R5)       ; Step device type to 750
FF09 CF  35  90  014A    544          MOVB    #^A/57,INI$T_HWTYPE+2   ; Change CPU type to 'V750'
             10  11  014F    545          BRB     CI_780                  ; Join common code
                     0151    546
                     0151    547 CI_785:
                     0151    548
FF03 CF  35  90  0151    549          MOVB    #^A/5/,INI$T_HWTYPE+3   ; Change CPU type to 'V785'
             09  11  0156    550          BRB     CI_780
                     0158    551
                     0158    552 CI_790:
                     0158    553
FEF5 CF  30303638 8F  D0  0158    554          MOVL    #^A/8600/,INI$T_HWTYPE  ; Change CPU type to '8600'
                     0161    555                                          ; Device type = CI780
                     0161    556
                     0161    557 CI_780:                                  ; * End of CPU dependent code *
```

```
                        0161    558
      08    90          0161    559          MOVB    #IPL$_SCS,-                ; Set ipl for fork block to
00AB  C5                0163    560                  <UCB$C_MSGFKBLK+UCB$B_FIPL>(R5)
                        0166    561                                             ;  start up msg to opa0
    54  A5    B5        0166    562          TSTW    UCB$W_UNIT(R5)             ; Is this unit 0?
      02    12          0169    563          BNEQ    10$                        ; Branch if not
      02    10          016B    564          BSBB    INIT_CTLR                  ; Else init controller too
                        016D    565
                        016D    566 OTHER_CPU:
                        016D    567
                  05    016D    568 10$:     RSB                                ; Done with unit init
                        016E    569
                        016E    570
                        016E    571          .DSABL  LSB
```

```
                     016E    573          .SBTTL  CONTROLLER INIT
                     016E    574
                     016E    575 ;+
                     016E    576 ; The controller initialization entry as seen by the system, PA$CTLINIT,
                     016E    577 ; is a noop since initialization can't begin without the unit 0 UCB.
                     016E    578 ; Actual controller init is called from unit 0 unit initialization with
                     016E    579 ; the same inputs as unit init.
                     016E    580 ;
                     016E    581 ; Inputs:
                     016E    582 ;
                     016E    583 ;       R3                      -Addr of PA configuration register
                     016E    584 ;       R4                      -Same as R3
                     016E    585 ;       R5                      -Addr of UCB for unit 0
                     016E    586 ;
                     016E    587 ; Outputs:
                     016E    588 ;
                     016E    589 ;       All registers           -Preserved
                     016E    590 ;-
                     016E    591
                     016E    592 PA$CTLINIT::                            ; Controller init called by system
                     016E    593
                  05 016E    594          RSB                            ; Return
```

```
                     016F    596 ;
                     016F    597 ; Controller initialization called from unit 0 init.
                     016F    598 ;
                     016F    599
                     016F    600         .ENABL  LSB
                     016F    601 INIT_CTLR::
        0084 C5   D5 016F    602         TSTL    UCB$L_PDT(R5)           ; Built structures yet?
             32   13 0173    603         BEQL    BUILD_STRUCT            ; Branch if not
             10   AA 0175    604         BICW    #UCB$M_ONLINE,-         ; Set unit offline to show
           64 A5    0177    605                 UCB$W_STS(R5)           ;  port init being done
    54   0084 C5   D0 0179    606         MOVL    UCB$L_PDT(R5),R4        ; Get PDT addr
                     017E    607         $PRTCTINI -                     ; Protect from non-ex port
                     017E    608                 B^1$,#MCHK$M_NEXM       ;
             01   D0 018A    609         MOVL    #PA_PMC_M_MIN,-         ;  and maint init port
        00E8 D4      018C    610                 @PDT$L_PMC(R4)          ;
                     018F    611         $PRTCTEND 1$                    ; End of mcheck protection
          05 50   E9 0190    612         BLBC    R0,5$                   ; If mcheck, dont mark port
                     0193    613                                         ;  powered up
             02   A8 0193    614         BISW    #PDT$M_PUP,-            ; Set power up on this
        0110 C4      0195    615                 PDT$W_[PORT_STS(R4)     ;  port
                     0198    616
             00   E2 0198    617 5$:     BBSS    #PDT$V_PWF_CLNUP,-      ; Branch if SYSAP notification
      08 0110 C4     019A    618                 PDT$W_[PORT_STS(R4),10$ ;  underway; else set pwf recov
                     019E    619                                         ;  in progress,
    51   0364 8F   3C 019E    620         MOVZWL  #SS$_POWERFAIL,R1       ;  set aux status to give SYSAP's,
           FE5A'   30 01A3    621         BSBW    ERR$PWF_RECOV           ;  call recovery startup
                     01A6    622
                  05 01A6    623 10$:    RSB                             ; Return
                     01A7    624
                     01A7    625 BUILD_STRUCT:                           ; Complete unit init at fork IPL
          64    64   D0 01A7    626         MOVL    PA_CNF(R4),PA_CNF(R4)   ; Clear all miscellaneous errors
        04 A4    01   D0 01AA    627         MOVL    #PA_PMC_M_MIN,PA_PMC(R4); Place port in un-initialized state
    53  00000188'EF   9E 01AE    628         MOVAB   CHECK_SYSTEMID,R3       ; Address of where to resume at fork IPL
              0545   31 01B5    629         BRW     INI$FORK                ; Fork...
                     01B8    630
                     01B8    631         .DSABL  LSB
```

```
                    01B8   633
                    01B8   634 ;
                    01B8   635 ; The SYSGEN parameter SCSSYSTEMID must be sent to a non-zero value. If it
                    01B8   636 ; has not been so initialized, log this error condition and do not allow the
                    01B8   637 ; port to come on-line.
                    01B8   638 ;
                    01B8   639
                    01B8   640          .ENABL  LSB
                    01B8   641 CHECK_SYSTEMID:
00000000'GF   D5    01B8   642          TSTL    G^SCS$GB_SYSTEMID       ; Has SCSSYSTEMID been initialized?
          0E   12   01BE   643          BNEQ    BUILD_PDT               ; Branch if it has
00000004'GF   B5    01C0   644          TSTW    G^SCS$GB_SYSTEMID+4     ; Are we sure its been initialized?
          06   12   01C6   645          BNEQ    BUILD_PDT               ; Branch if it has
                    01C8   646
     50   02   9A   01C8   647          MOVZBL  #PAER$K_ES_SCSID,R0     ; Log that SCSSYSTEMID is 0
        FE32'   31   01CB   648          BRW     ELOG$INIT_SWERR
                    01CE   649          .DSABL  LSB
```

```
                                01CE   651          .SBTTL  BUILD PDT
                                01CE   652
                                01CE   653 ;
                                01CE   654 ; PDT adjacency assumptions:
                                01CE   655 ;
                                01CE   656
                                01CE   657 ASSUME  PDT$L_FLINK     EQ 0
                                01CE   658 ASSUME  PDT$B_PDT_TYPE  EQ 7
                                01CE   659 ASSUME  PDT$W_SIZE      EQ 8
                                01CE   660 ASSUME  PDT$W_SIZE+2    EQ PDT$B_TYPE
                                01CE   661 ASSUME  PDT$B_TYPE+1    EQ PDT$B_SUBTYP
                                01CE   662 ASSUME  PDT$B_SUBTYP+1  EQ PDT$C_SCSBASE
                                01CE   663
                                01CE   664          .ENABL  LSB
                                01CE   665
                                01CE   666 BUILD_PDT:
                                01CE   667
     51    0560 8F    3C      01CE   668          MOVZWL  #<PDT$C_PALENGTH+512>,R1  ; Enough for a PDT + 1 pg
      00000000'GF    16      01D3   669          JSB     G^EXE$ALONONPAGED        ; Allocate non-paged pool for PDT
            06 50    E8      01D9   670          BLBS    R0,10$                   ; Branch if success
                             01DC   671          ASSUME  PAER$K_ES_POOL EQ 0      ; Else, log a pool allocation
               50    D4      01DC   672          CLRL    R0                       ; error.
             FE1F'   30      01DE   673          BSBW    ELOG$INIT_SWERR
                     05      01E1   674          RSB                              ; Return with unit offline
                             01E2   675
        50   52    D0      01E2   676 10$:     MOVL    R2,R0                    ; Save address
     52   03E0 C2    DE      01E5   677          MOVAL   PDT$C_PQB+512(R2),R2     ; Round PQB offset up to
     52   01FF 8F    AA      01EA   678          BICW    #511,R2                  ;  next page boundary
 52   000001E0 8F    C2      01EF   679          SUBL    #PDT$C_PQB,R2            ;  and compute corresponding PDT base
     0084 C5   52    D0      01F6   680          MOVL    R2,UCB$L_PDT(R5)         ; Save PDT addr
       53   24 A5    D0      01FB   681          MOVL    UCB$L_CRB(R5),R3         ; Get CRB addr
       10 A3   52    D0      01FF   682          MOVL    R2,CRB$L_AUXSTRUC(R3)    ;  and save PDT addr in CRB
  08 A0    52   50    A3      0203   683          SUBW3   R0,R2,PDT$W_SIZE(R0)     ; Fix up size of unused memory
  0A A0    0060 8F    B0      0208   684          MOVW    #DYN$C_SCS,PDT$B_TYPE(R0) ;  and type
                82    D4      020E   685          CLRL    (R2)+                    ; Init PDT, unused longwd,
 82   01000000 8F    D0      0210   686          MOVL    #PDT$C_PA@24,(R2)+       ;  unused 3 bytes and port type
  82     51   08 A0    A3      0217   687          SUBW3   PDT$W_SIZE(R0),R1,(R2)+  ;  PDT size,
     82   0560 8F    B0      021C   688          MOVW    #<DYN$C_SCS_PDT@8 + DYN$C_SCS>,(R2)+  ;  structure subtype and type
       53   30 A3    D0      0221   689          MOVL    CRB$L_INTD+VEC$L_INITIAL(R3),R3  ; Get addr of controller
                             0225   690                                           ;  init routine
     51   FDD7 CF    3E      0225   691          MOVAW   PA$SCSOFFSET,R1          ; Get addr of table of offsets
                             022A   692                                           ;  to SCS entries in PADRIVER
                             022A   693
          50   81    32      022A   694 20$:     CVTWL   (R1)+,R0                 ; Get offset to next SCS routine
                06    13      022D   695          BEQL    30$                      ; Branch if no more
     82   53   50    C1      022F   696          ADDL3   R0,R3,(R2)+              ; Add offset from controller init
                             0233   697                                           ;  to addr of controller init
                             0233   698                                           ;  and store in PDT
                F5    11      0233   699          BRB     20$                      ; Get next offset
                             0235   700
          7E   54    7D      0235   701 30$:     MOVQ    R4,-(SP)                 ; Save R4, R5
  00    00 8F    00    2C      0238   702          MOVC5   #0,#0,#0,-               ; Zero PDT from here to
          025C 8F            023D   703                  #<PDT$L_DQELOGOUT - PDT$C_SCSEND>,-
                62           0240   704                  (R2)                     ;  to logout area
          54   8E    7D      0241   705          MOVQ    (SP)+,R4                 ; Restore  R4, R5
     52   0084 C5    D0      0244   706          MOVL    UCB$L_PDT(R5),R2         ; Get base of PDT again
          00AC C2    DE      0249   707          MOVAL   PDT$L_WAITQFL(R2),-      ; Init the pool wait
```

```
                00AC C2          024D   708                 PDT$L_WAITQFL(R2)       ;  queue
                00AC C2    DE    0250   709         MOVAL   PDT$L_WAITQFL(R2),-     ;  to empty
                00B0 C2          0254   710                 PDT$L_WAITQBL(R2)       ;
                0000'CF    C1    0257   711         ADDL3   W^SCS$GL_SCSSIZE,-      ; Set size of message header,
                     12          025B   712                 #PPD$C_LENGTH,-         ;  PPD +
                00B4 C2          025C   713                 PDT$L_MSGHDRSZ(R2)      ;  SCS header
                0000'CF    C1    025F   714         ADDL3   W^SCS$GL_SCSSIZE,-      ; Save size of datagram header,
                     12          0263   715                 #PPD$C_LENGTH,-         ;  PPD +
                0190 C2          0264   716                 PDT$L_DGHDRSZ(R2)       ;  SCS portion only
            00000048 8F    D0    0267   717         MOVL    #CXB$C_HEADER,-         ; Set size of total dg header
                00B8 C2          026D   718                 PDT$L_DGOVRHD(R2)       ;  including PPD/SCS, and net
                0190 C2    C3    0270   719         SUBL3   PDT$L_DGHDRSZ(R2),-     ; Calculate size of network header
                00B8 C2          0274   720                 PDT$L_DGOVRHD(R2),-     ;
                0194 C2          0277   721                 PDT$L_DGNETHD(R2)       ;
        00BC    C2    01    CE   027A   722         MNEGL   #1,PDT$L_MAXBCNT(R2)    ; Set max bytes per block xfer =
                                 027F   723                                         ;  2**32-1
        52      00E4  C2    DE   027F   724         MOVAL   PDT$C_PAREGBASE(R2),R2  ; Step to addr of PA device
                                 0284   725                                         ;  registers accessed via PDT
        51      FDB6  CF    3E   0284   726         MOVAW   PA$REGOFFSET,R1         ; Get addr of table of offsets
                                 0289   727                                         ;  to device registers we want
                                 0289   728
                50      81    32   0289   729 40$:  CVTWL   (R1)+,R0              ; Get next offset
                        06    19   028C   730         BLSS    50$                 ; Branch if end of offset table
        82      54      50    C1   028E   731         ADDL3   R0,R4,(R2)+         ; Add offset to config reg addr,
                                 0292   732                                         ;  and store in PDT
                        F5    11   0292   733         BRB     40$                 ; Get next offset
                                 0294   734
        52      0084    C5    D0   0294   735 50$:  MOVL    UCB$L_PDT(R5),R2        ; Get base of PDT again
        00DC    C2      55    D0   0299   736         MOVL    R5,PDT$L_UCB0(R2)       ; Save in PDT UCB 0 addr
                0174 C2    DE   029E   737         MOVAL   PDT$Q_FORMPB(R2),-      ; Init formative
                0174 C2          02A2   738                 PDT$Q_FORMPB(R2)        ;  PB list
                0174 C2    DE   02A5   739         MOVAL   PDT$Q_FORMPB(R2),-      ;  to empty
                0178 C2          02A9   740                 PDT$Q_FORMPB+4(R2)      ;
                     03    90   02AC   741         MOVB    #<PDT$M_CUR_LBS!PDT$M_PRV_LBS>,-  ; Set current/previous
                0180 C2          02AE   742                 PDT$B_PO_LBSTS(R2)              ;  loopback status to
                     03    90   02B1   743         MOVB    #<PDT$M_CUR_LBS!PDT$M_PRV_LBS>,-  ;  good,
                0181 C2          02B3   744                 PDT$B_PT_LBSTS(R2)              ;  both paths
                01D0 C2    DE   02B6   745         MOVAL   PDT$Q_DFREEQ(R2),-      ; Set up addresses
                0208 C2          02BA   746                 PDT$L_DFQHDR(R2)        ;  of datagram and
                01D8 C2    DE   02BD   747         MOVAL   PDT$Q_MFREEQ(R2),-      ;  message free queue
                020C C2          02C1   748                 PDT$L_MFQHDR(R2)        ;  headers for port
                00B8 C2    D0   02C4   749         MOVL    PDT$L_DGOVRHD(R2),-     ; Set up dg and msg queue
                0210 C2          02C8   750                 PDT$W_DQELEN(R2)        ;  entry sizes in PQB
        00000000'GF    A0    02CB   751         ADDW    G^SCS$GW_MAXDG,-        ;  for port
                0210 C2          02D1   752                 PDT$W_DQELEN(R2)        ;
                00B4 C2    D0   02D4   753         MOVL    PDT$L_MSGHDRSZ(R2),-    ; Queue entry size =
                0214 C2          02D8   754                 PDT$W_MQELEN(R2)        ;  PPD/SCS header
        00000000'GF    A0    02DB   755         ADDW    G^SCS$GW_MAXMSG,-       ;  + SYSGEN param
                0214 C2          02E1   756                 PDT$W_MQELEN(R2)        ;
                01E0 C2    DE   02E4   757         MOVAL   PDT$C_PQB(R2),-         ; Set VA of PQB within
                0218 C2          02E8   758                 PDT$L_VPQB(R2)          ;  PDT
                     0C    DB   02EB   759         MFPR    #PR$_SBR,-              ; Set PA of base of SPT
                0224 C2          02ED   760                 PDT$C_SPTBASE(R2)       ;
                     0D    DB   02F0   761         MFPR    #PR$_SLR,-              ;  and SPT length
                0228 C2          02F2   762                 PDT$C_SPTLEN(R2)        ;
        00000000'GF    D0    02F5   763         MOVL    G^MMG$GL_GPTBASE,-      ; Set VA of base of global
                022C C2          02FB   764                 PDT$L_GPTBASE(R2)       ;  page table
```

```
                00000000'GF   C1  02FE    765          ADDL3   G^SGN$GL_MAXGPGCT,-     ;  and GPT length which is
                    0228 C2       0304    766                  PDT$L_SPTLEN(R2),-      ;  the sum of the global page
                    0230 C2       0307    767                  PDT$L_GPTLEN(R2)        ;  count and spte count  (global
                                  030A    768                                          ;  page table base=spt base.)
          51    00000000'GF   DE  030A    769          MOVAL   G^SCS$GL_PDT,R1         ; Get base of SCS port list
                                  0311    770
                    50    61    D0  0311    771  60$:    MOVL    (R1),R0                 ; Get next port
                          05    13  0314    772          BEQL    70$                     ; Branch if none
                    51    50    D0  0316    773          MOVL    R0,R1                   ; Else save next PDT as previous
                          F6    11  0319    774          BRB     60$                     ; Continue down the list
                                  031B    775
                    61    52    D0  031B    776  70$:    MOVL    R2,(R1)                 ; Hook this PDT to end of list
                                  031E    777
                                  031E    778          .DSABL  LSB
```

```
                     031E   780          .SBTTL  BUILD TEMPLATE LOOPBACK DG
                     031E   781
                     031E   782 ;+
                     031E   783 ; Allocate and initialize the template loopback datagram except for local
                     031E   784 ; port number and CRC.  These are recalculated each time power is recovered.
                     031E   785 ;
                     031E   786 ; Note that the template loopback datagram need not have a network header,
                     031E   787 ; nor have PPD$W_SIZE be a negative offset.
                     031E   788 ;-
                     031E   789
                     031E   790          .ENABL  LSB
                     031E   791
                     031E   792 BUILD_TLB:
                     031E   793
     51   0046 8F    3C  031E   794          MOVZWL  # PD$C_LB_LENGTH,R1     ; Get total template size
      00000000'GF    16  0323   795          JSB     G^EXE$ALONONPAGED       ; Allocate non-paged pool for template
            08 50    E8  0329   796          BLBS    R0,10$                  ; Branch if got it
                         032C   797          ASSUME  PAER$K_ES_POOL EQ 0     ; Else, log a pool allocation
                50   D4  032C   798          CLRL    R0                      ; error.
              FCCF'  30  032E   799          BSBW    ELOG$INIT_SWERR
              0320   31  0331   800          BRW     CLEANUP_PDT             ; Go clean up allocated buffers
                         0334   801
       08 A2    51   B0  0334   802 10$:     MOVW    R1,PPD$W_SIZE(R2)       ; Save structure size and
                3B   B0  0338   803          MOVW    #DYN$C_CIDG,-           ;  type
            0A A2        033A   804                  PPD$B_TYPE(R2)          ;
            0D A2   94   033C   805          CLRB    PPD$B_STATUS(R2)        ; Init template status = 0,
              0D    9B   033F   806          MOVZBW  #PPD$C_SNDLB,-          ;  opcode = SNDLB,
            0E A2        0341   807                  PPD$B_OPC(R2)           ;
                30   B0  0343   808          MOVW    #PPD$C_LBDAT_LEN,-      ;  LB length to # of
            10 A2        0345   809                  PPD$W_LENGTH(R2)        ;  bytes of data
                51   D4  0347   810          CLRL    R1                      ; Generate LB data pattern
                         0349   811
  12 A241    51   90  0349   812 20$:     MOVB    R1,PPD$B_LBDATA(R2)[R1] ;  of bytes = 0,1,2,...
    F7 51    30   F3  034E   813          AOBLEQ  #PPD$C_LBDAT_LEN,R1,20$ ;  ... LBDAT_LEN-1
50   0084 C5   D0  0352   814          MOVL    UCB$L_PDT(R5),R0        ; Hook template to
0184 C0   52   D0  0357   815          MOVL    R2,PDT$L_LBDG(R0)       ;  PDT
```

```
                     035C   817          .SBTTL  SCS LAYER INITIIALIZATION
                     035C   818
                     035C   819 ;+
                     035C   820 ; Now call into the SCS layer so it can do any needed initializations.
                     035C   821 ;-
                     035C   822
         FCA1'  30   035C   823          BSBW    SCS$INITIAL                 ; Do it
52  0084 C5  D0   035F   824          MOVL    UCB$L_PDT(R5),R2            ; Restore PDT address
 00000000'GF  3C   0364   825          MOVZWL  G^SCS$GW_BDTCNT,-           ; Set count of BD's
     0220 C2       036A   826                  PDT$W_BDTLEN(R2)            ;  and address of
 00000000'GF  D0   036D   827          MOVL    G^SCS$GL_BDT,-              ;  BDT in case BDT
     021C C2       0373   828                  PDT$L_VBDT(R2)              ;  has already been created
       08 50   E8   0376   829          BLBS    R0,INIT_CRB                 ; Branch if success
              0379   830          ASSUME  PAER$K_ES_POOL EQ 0         ; Else, log a pool allocation
          50   D4   0379   831          CLRL    R0                          ; error.
         FC82'  30   037B   832          BSBW    ELOG$INIT_SWERR
         02D3   31   037E   833          BRW     CLEANUP_PDT                 ; Go clean up allocated pool
```

```
                     0381   835          .SBTTL  INIT POLLER TIMER AND TRACE FUNCTION
                     0381   836
                     0381   837 ;+
                     0381   838 ; Initialize timer to wake driver up and insert CRB on timer queue.
                     0381   839 ;-
                     0381   840
                     0381   841          .ENABL  LSB
                     0381   842
                     0381   843 INIT_CRB:
                     0381   844
 53    24 A5    D0   0381   845          MOVL    UCB$L_CRB(R5),R3        ; Get CRB addr
       FC78'    30   0385   846          BSBW    CNF$CALCINTDUE          ; Set to wake up basic timer
                     0388   847                                          ;  interval from now and
00000000'GF    16   0388   848          JSB     G^IOC$THREADCRB         ; put CRB on timer queue
                     038E   849
                     038E   850          .IF     DF PA$DEBUG             ; Conditional init of debug facility
                     038E   851          BSBW    TRC$INIT                ; Init trace buffer
                     038E   852          .ENDC                           ;
                     038E   853
                     038E   854          .DSABL  LSB
```

```
                     038E    856          .SBTTL  INI$PORT, INITIALIZE PORT
                     038E    857
                     038E    858 ;+
                     038E    859 ; Load the port microcode, init port hardware, complete initialization
                     038E    860 ; of the template loopback datagram (in case port number changed while
                     038E    861 ; powered down.)  Allocate and queue free datagrams and messages to
                     038E    862 ; port for future receives.  If all this is successful, set unit 0 online,
                     038E    863 ; clear power fail in progress and set port powered up.
                     038E    864 ;
                     038E    865 ; Inputs:
                     038E    866 ;
                     038E    867 ;       R4                      -Addr of port configuration register
                     038E    868 ;       R5                      -Addr of UCB of unit 0
                     038E    869 ;
                     038E    870 ;       IPL                     -IPL$_SCS
                     038E    871 ;
                     038E    872 ; It is assumed that device interrupts are disabled, that there are no
                     038E    873 ; outstanding interrupts, and that the port is in the un-initialized state.
                     038E    874 ;-
                     038E    875
                     038E    876 ASSUME  PDT$W_STDGUSED  EQ  PDT$W_STDGDYN+2
                     038E    877 ASSUME  PDT$Q_COMQH     EQ  PDT$Q_COMQBASE+8
                     038E    878 ASSUME  PDT$Q_COMQ2     EQ  PDT$Q_COMQH+8
                     038E    879 ASSUME  PDT$Q_COMQ3     EQ  PDT$Q_COMQ2+8
                     038E    880 ASSUME  PDT$Q_RSPQ      EQ  PDT$Q_COMQ3+8
                     038E    881
                     038E    882 ASSUME  PPD$C_LBDAT_LEN+7  LE   255
                     038E    883
                     038E    884          .ENABL  LSB
                     038E    885
                     038E    886 INI$PORT::
                     038E    887
          64   64    D0  038E    888          MOVL    PA_CNF(R4),PA_CNF(R4)   ; Clear any misc errors we can
          1000 8F    B3  0391    889          BITW    #PA_CNF_M_NOCI,-        ; CI750 port inaccessible?
               64          0395    890                  PA_CNF(R4)         ;
                  03   13  0396    891          BEQL    10$                     ; Branch if accessible
              0283     31  0398    892          BRW     PORT_NOTPRES            ; Else go handle error
                           039B    893
    04 A4      01    D0  039B    894 10$:     MOVL    #PA_PMC_M_MIN,PA_PMC(R4); Place port in un-initialized state
       7E      54    7D  039F    895          MOVQ    R4,-(SP)                ; Save registers destroyed by subr
            022B     30  03A2    896          BSBW    TEST_SHUTDOWN           ; Check if we are shutting down and
                         03A5    897                                          ;  if so, take operator action which
                         03A5    898                                          ;  may possibly include a bugcheck
                         03A5    899                                          ;  if we can't go on without the port
       54      8E    7D  03A5    900          MOVQ    (SP)+,R4                ; Restore registers
    52 0084    C5    D0  03A8    901          MOVL    UCB$L_PDT(R5),R2        ; Get PDT address
               01    12  03AD    902          BNEQ    15$                     ; Branch if this port is still in business
                     05  03AF    903          RSB                             ; Else return to caller without
                         03B0    904                                          ;  reinitializing it
                         03B0    905
00000652'EF    01    90  03B0    906 15$:     MOVB    #1,INI$CPU_REV          ; Assume CPU rev will be okay
30353756 8F  FC9B    CF  D1  03B7    907          CMPL    INI$T_HWTYPE,#^A/V750/  ; Running 11/750?
               11    12  03C0    908          BNEQ    CPU_REV_OK              ; Branch if not
       51      3E    DB  03C2    909          MFPR    #PR$_SID,R1             ; Read SID (copy of SID in memory
                         03C5    910                                          ;  is not good enough because the
                         03C5    911                                          ;  ucode rev level may have been
                         03C5    912                                          ;  increased by the loading of patches
```

```
                                03C5   913                                     ;  by the PCS software load mechanism
                                03C5   914                                     ;  or decreased by failure to load patches
                                03C5   915                                     ;  following pwr fail recovery.)
          50   51    F8 8F     78   03C5   916          ASHL    #-8,R1,R0       ; Get ucode rev level in low byte
               61 8F    50     91   03CA   917          CMPB    R0,#MIN_750_REV ; Is CPU ucode sufficient to run port
                        03     18   03CE   918          BGEQ    CPU_REV_OK      ; Branch if so
                      0256     31   03D0   919          BRW     CPU_REV_ERROR   ; Go handle if not sufficient
                                03D3   920
                                03D3   921 CPU_REV_OK:
                                03D3   922
          53   00000000'GF     D0   03D3   923          MOVL    G^SCS$GL_MCADR,R3 ; Get base of microcode image
                        09     12   03DA   924          BNEQ    CHECK_QUEUES    ; Branch if got microcode image in pool
                     50   01   9A   03DC   925          MOVZBL  #PAER$K_ES_CODE, R0 ; Else, log error indicating that
                        FC1E'   30   03DF   926          BSBW    ELOG$INIT_SWERR ; microcode could not be found in pool.
                                03E2   927
                      026F     31   03E2   928 20$:     BRW     CLEANUP_PDT     ; Go clean up allocated pool
                                03E5   929
                                03E5   930 CHECK_QUEUES:
                                03E5   931
                     20   00   3B   03E5   932          SKPC    #0,#<PDT$Q_RSPQ - PDT$Q_COMQBASE>,-
                     01E0 C2         03E8   933                  PDT$Q_COMQBASE(R2) ; Check for any nonzero port
                                03EB   934                                     ;  command queues or response queue
                        17   13   03EB   935          BEQL    LOAD_UCODE      ; Branch if all zero (empty queues)
                                03ED   936          $DEBUGCHECK #ERR$V_DEB_NEPQ ; Else do optional bugcheck
                                0400   937                                     ;  since on reinit all queues should
                                0400   938                                     ;  be empty
                        61   94   0400   939          CLRB    (R1)            ; For recovery, try zeroing byte
                        E1   11   0402   940          BRB     CHECK_QUEUES    ;  and check again
                                0404   941
                                0404   942 LOAD_UCODE:
                                0404   943
               52   0400 8F    3C   0404   944          MOVZWL  #^X400,R2       ; Set initial MC addr assuming  ROM port
     5F6A00A1 8F     08 A3     D1   0409   945          CMPL    8(R3),#^X5F6A00A1 ; Is this code from an all RAM
                                0411   946                                     ;  port file?
                        02   12   0411   947          BNEQ    30$             ; Branch if not -- assume ROM
                        52   D4   0413   948          CLRL    R2              ; Else set to load both PROM
                                0415   949
                     7E   52   7D   0415   950 30$:     MOVQ    R2,-(SP)        ; Save initial addresses
                                0418   951
                  14 A4   52   D0   0418   952 40$:     MOVL    R2,PA_MADR(R4)  ; Give CS addr to CI
                  18 A4   83   D0   041C   953          MOVL    (R3)+,PA_MDATR(R4) ; Write 4 bytes of ucode
14 A4    52    00001000 8F    C9   0420   954          BISL3   #^X1000,R2,-    ; Step CS addr of h.o.
                                0429   955                  PA_MADR(R4)        ;  word of data
                  18 A4   83   3C   0429   956          MOVZWL  (R3)+,PA_MDATR(R4) ; Write h.o. 2 bytes
     E3 52    00000C00 8F    F2   042D   957          AOBLSS  #^XC00,R2,40$   ; Loop till entire image loaded
                     52   8E   7D   0435   958          MOVQ    (SP)+,R2        ; Retreive initial addresses for
                                0438   959                                     ;  read back of ucode
                        52   D5   0438   960          TSTL    R2              ; Is this a possible all RAM port?
                        2A   13   043A   961          BEQL    START_UCODE     ; Branch if so to skip check of
                                043C   962                                     ;  possible ROM code that might
                                043C   963                                     ;  not agree with our ucode image.
                                043C   964
                                043C   965 CHECK_UCODE:
                                043C   966
                  14 A4   52   D0   043C   967          MOVL    R2,PA_MADR(R4)  ; Set next control store addr
                     50   83   D0   0440   968          MOVL    (R3)+,R0        ; Get next 4 bytes wcs should have
               50    18 A4   D1   0443   969          CMPL    PA_MDATR(R4),R0 ; WCS ok?
```

```
                                12  12  0447   970          BNEQ    BAD_UCODE                       ; Branch if not
      14 A4    52    00001000 8F  C9  0449   971          BISL3   #^X1000,R2,PA_MADR(R4)           ; Set to read h.o. 2 bytes of uword
                           50    83  3C  0452   972          MOVZWL  (R3)+,R0                        ; Get next 2 bytes WCS should have
                      50   18 A4  B1  0455   973          CMPW    PA_MDATR(R4),R0                 ; Next 2 bytes ok?
                                03  13  0459   974          BEQL    60$                             ; Branch if so
                                        045B   975
                                        045B   976 BAD_UCODE:
                                        045B   977
                              01D7  31  045B   978          BRW     WCS_ERROR                       ; Go handle error
                                        045E   979
         D6 52    00000C00 8F  F2  045E   980 60$:     AOBLSS  #^XC00,R2,CHECK_UCODE           ; Branch if more to check
                                        0466   981
                                        0466   982
                                        0466   983 START_UCODE:
                                        0466   984
                00000040 8F  C8  0466   985          BISL    #PA_PMC_M_PSA,-                 ; Set programmable start addr
                      04 A4      046C   986                  PA_PMC(R4)                      ;  bit
                00000400 8F  D0  046E   987          MOVL    #PA_C_UCODEST,-                 ; Set microcode start addr
                      14 A4      0474   988                  PA_MADR(R4)                     ;
                         01  D0  0476   989          MOVL    #PA_PIC_M_PIC,-                 ; Set port initialize -- move
                    0924 C4      0478   990                  PA_PIC(R4)                      ;  port state from uninit to disabled
                                 047B   991          TIMEWAIT #^10000>,#PA_PMC_M_MIF,-       ; Wait for port init done
                                 047B   992                   PA_PMC(R4),L                   ;  or 100 msec
                      07 50      E9  04A3   993          BLBC    R0,70$                          ; Branch if failed
                    0900 C4      D1  04A6   994          CMPL    PA_PS(R4),-                     ; Check that port init is done
                         08          04AA   995                  #PA_PS_M_PIC                    ;  and no errors set
                         03  13  04AB   996          BEQL    90$                             ; Branch if sucess
                              018B  31  04AD   997 70$:     BRW     INIT_PORT_FAIL                  ; Else go to failure
                                        04B0   998
                50   0084 C5  D0  04B0   999 90$:     MOVL    UCB$L_PDT(R5),R0                ; Retreive PDT addr
                     14   09  EF  04B5  1000          EXTZV   #9,#20,-                        ; Extract virtual page #
                50   0218 C0      04B8  1001                  PDT$L_VPQB(R0),R0               ;  of PQB
           51    00000000'GF  D0  04BC  1002          MOVL    G^MMG$GL_SPTBASE,R1             ; Get base of SPT
                    50   6140  D0  04C3  1003          MOVL    (R1)[R0],R0                     ; Get PTE for PQB addr
          50   50   14   00  EF  04C7  1004          EXTZV   #0,#20,R0,R0                    ; Get PFN of PQB addr
        0904 C4   50   09  78  04CC  1005          ASHL    #9,R0,PA_PQBBR(R4)              ; Convert to phys addr and
                                04D2  1006                                                  ;  store in the PQB base reg
            04 00000000'GF  E8  04D2  1007          BLBS    G^SCS$GB_PASANITY,95$           ; Branch if sanity timer wanted
                         02  C8  04D9  1008          BISL    #PA_PMC_M_MTD,-                 ; Else disable it
                      04 A4      04DB  1009                  PA_PMC(R4)                      ;
                                04DD  1010
                         01  D0  04DD  1011 95$:     MOVL    #PA_PSR_M_PSC,-                 ; Release the port
                    0918 C4      04DF  1012                  PA_PSR(R4)                      ;  status register to port
                         01  D0  04E2  1013          MOVL    #PA_PEC_M_PEC,-                 ; Finally,...
                    091C C4      04E4  1014                  PA_PEC(R4)                      ;  enable the port
                                04E7  1015
                                04E7  1016 INIT_LBDG_CRC:
                                04E7  1017
                54   0084 C5  D0  04E7  1018          MOVL    UCB$L_PDT(R5),R4                ; Get PDT address
                53   0184 C4  D0  04EC  1019          MOVL    PDT$L_LBDG(R4),R3               ; Get addr of LB template
                     010C D4  90  04F1  1020          MOVB    @PDT$L_PPR(R4),-                ; Save local port number
                        0C A3      04F5  1021                  PPD$B_PORT(R3)                  ;  in LB dg template
                51   0043 8F  3C  04F7  1022          MOVZWL  #<PPD$C_LCB_DATA + PPD$C_LBDAT_LEN>,R1
                 00000000'GF  16  04FC  1023          JSB     G^EXE$ALONONPAGED               ; Allocate temporary buffer for
                                0502  1024                                                  ;  setting up data to calc CRC
                      03 50      E8  0502  1025          BLBS    R0,100$                         ; Branch if got it
                         013E    31  0505  1026 97$:     BRW     INIT_LBDG_FAIL                  ; Else go handle error
```

```
                                0508  1027
      08 A2    51    00600000 8F   C9 0508  1028 100$:   BISL3   #DYN$C_SCS@16,R1,-       ; Set structure type
                                0511  1029                PPD$W_SIZE(R2)          ;  and size in temporary buffer
                                0511  1030                                        ; Note that the size is not CRCed
                                0511  1031                                        ;  so it can be the straight size
                                0511  1032                                        ;  here, not a negative offset to
                                0511  1033                                        ;  a net header.
                                0511  1034                                        ; Set up data to CRC:
                    0C A2       94  0511  1035          CLRB    PPD$W_LCB_LEN7(R2)      ;  H.o. lb data length + 7
                07  10 A3       81  0514  1036          ADDB3   PPD$W_LENGTH(R3),#7,-   ;  L.o. lb data length + 7
                    0D A2           0518  1037                  PPD$W_LCB_LEN7+1(R2)    ;
                50  0C A3       90  051A  1038          MOVB    PPD$B_PORT(R3),R0       ;
                0E A2   50       90  051E  1039          MOVB    R0,PPD$B_LCB_PORT(R2)   ;  own port number,
                0F A2   50       92  0522  1040          MCOMB   R0,PPD$B_LCB_NPORT(R2)  ;  NOT(own port),
                10 A2   50       90  0526  1041          MOVB    R0,PPD$B_LCB_LPORT(R2)  ;  local port,
                        0D       90  052A  1042          MOVB    #PPD$C_SNDLB,-          ;  SNDLB opcode,
                    11 A2           052C  1043                  PPD$B_LCB_OPC(R2)       ;
                    12 A2       94  052E  1044          CLRB    PPD$B_LCB_0(R2)         ;  and 0 (packing fmt = 0)
                       3C       BB  0531  1045          PUSHR   #^M<R2,R3,R4,R5>        ; Save registers
                       30       28  0533  1046          MOVC    #PPD$C_LBDAT_LEN,-      ; Copy LB data from
                    12 A3           0535  1047                  PPD$B_LBDATA(R3),-      ;  template to temporary
                    13 A2           0537  1048                  PPD$C_LCB_DATA(R2)      ;  buffer
                52      6E       7D  0539  1049          MOVQ    (SP),R2                 ; Get R2 and R3 again
                50  0D A2       9A  053C  1050          MOVZBL  PPD$W_LCB_LEN7+1(R2),R0 ; Get # bytes to crc
      FFFFFFFF 8F    FB8D     CF  0B  0540  1051          CRC     CRC_TABLE,#-1,-         ; Compute CRC from temporary
                0C A2   50          0549  1052                  R0,PPD$W_LCB_LEN7(R2)   ;  buffer
                       3C       BA  054C  1053          POPR    #^M<R2,R3,R4,R5>        ; Retreive registers
                42 A3   50       D2  054E  1054          MCOML   R0,PPD$L_LBCRC(R3)      ; Put CRC complement into template
                    50  52       D0  0552  1055          MOVL    R2,R0                   ; Copy temporary buffer addr
                00000000'GF      16  0555  1056          JSB     G^COM$DRVDEALMEM        ;  and return it to pool
                                0558  1057
                                0558  1058 INIT_DFREEQ:
                                0558  1059
          54    0084 C5       D0  0558  1060          MOVL    UCB$L_PDT(R5),R4        ; Get PDT address
      50   00000000'GF       3C  055D  1061          MOVZWL  G^SCS$GW_PAPPDDG,R0     ; Get # dg's for start handshakes
          0198 C4   50       D0  0564  1062          MOVL    R0,PDT$W_STDGDYN(R4)    ; Set dynamic count of # dgs for
                                0569  1063                                          ;  IDREC's and 0 the number curren'.y
                                0569  1064                                          ;  spoken for by known ports
                50   50       C0  0569  1065          ADDL    R0,R0                   ; Double to handle error log dgs
                        FA8E'   30  056C  1066          BSBW    SCS$ALL_FRDGS           ; Allocate and put on free queue
                      90 50     E9  056F  1067          BLBC    R0,97$                  ; Branch if insufficient memory
                50      03      3C  0572  1068          MOVZWL  #PA_C_MCACHESZ,R0       ; Get # msgs to fill cache
                        FA85'   30  0575  1069          BSBW    SCS$ALL_FRMSGS          ; Allocate and put on free queue
                      87 50     E9  0578  1070          BLBC    R0,97$                  ; Branch if insufficient memory
             017C C4    0F       90  057B  1071          MOVB    #15,PDT$B_MAX_PORT(R4)  ; Assume small capacity cluster
                010C D4    D5       0580  1072          TSTL    @PDT$L_PPR(R4)          ; Is it 16 ports max?
                        06      18  0584  1073          BGEQ    120$                    ; Branch if so
           017C C4    DF 8F    90  0586  1074          MOVB    #223,PDT$B_MAX_PORT(R4) ; Else up count to 224 ports
                                0586  1075
                010C D4    90       0586  1076 120$:    MOVB    @PDT$L_PPR(R4),-        ; Save local port number in
                017D C4             058A  1077                  PDT$B_PORT_NUM(R4)      ;  PDT
                       01       90  058D  1078          MOVB    #PPD$C_PSPO,-           ; Set poller to scan all ports
                017F C4             0590  1079                  PDT$B_REQIDPS(R4)       ;  on path A
              53    24 A5       D0  0592  1080          MOVL    UCB$L_CRB(R5),R3        ; Retreive CRB address
                        FA5E'   30  0596  1081          BSBW    CNF$CALCINTDUE          ; Calculate due time for
                                0599  1082                                          ;  next driver wakeup
                        FA5B'   30  0599  1083          BSBW    CNF$CALC_POLLSW         ; Compute the time to do a
```

```
                                05A5  1084                                    ;  full sweep by the poller
                          30    BB  05A5  1085          PUSHR   #^M<R4,R5>      ; Save registers
FF 8F    00 8F            00    2C  05A7  1086          MOVC5   #0,#0,#-1,-     ; Init port logout area
                    0080  8F        05AD  1087                  #<PDT$C_PALENGTH - PDT$L_DQELOGOUT>,-
                    02E0  C4        05B0  1088                  PDT$L_DQELOGOUT(R4)
                          30    BA  05B3  1089          POPR    #^M<R4,R5>      ; Restore registers
  00000653'EF             01    90  05B5  1090          MOVB    #1,INI$PORT_REV ; Assume port ucode rev is good
                          10    A8  05BC  1091          BISW    #UCB$M_ONLINE,- ;  set unit online
                       64 A5        05BE  1092                  UCB$W_STS(R5)   ;
                          01    AA  05C0  1093          BICW    #PDT$M_PWF_CLNUP,- ; Clear power up init
                    0110  C4        05C2  1094                  PDT$W_PORT_STS(R4) ;  in progress
                          06    A8  05C5  1095          BISW    #<PDT$M_PUP!PDT$M_LBDG>,- ; Set power up and loopback
                    0110  C4        05C7  1096                  PDT$W_PORT_STS(R4) ;  dg enabled on port
                          04    C8  05CA  1097          BISL2   #PA_PMC_M_MIE,- ; Enable interrupts
                    00E8  D4        05CC  1098                  @PDT$L_PMC(R4)  ;
                          05    05CF  1099          RSB                     ; Return from port initialization
                                05D0  1100          .DSABL  LSB
```

```
                        .SBTTL  TEST_SHUTDOWN,  CHECK IF PORT SHOULD
                        .SBTTL  -               BE LEFT OFFLINE

                ;+
                ; This routine is called each time a port is initialized.  If the port
                ; is being initialized for the first time, or if it is already shut
                ; down, then return is taken.
                ;
                ; If this is not the first port init, then it must be a reinit following
                ; a serious port error.  If the system device is not on this CI port,
                ; and the number of reinitialization tries have been exhausted, then
                ; the message, '%PAx0, CI Port is going Offline.' is printed.
                ; If there are retries left, then the message,
                ; '%PAx0, CI Port is Reinitializing ( xxx Retries Left).  Check Error Log.'
                ; is printed. Both messages are directed to _OPA0, rather than OPCOM since
                ; OPCOM needs a functioning system device to run and the system device may be
                ; accessible only through the failing port.
                ;
                ; In the case of initialization failures that result in branching back
                ; to INI$PORT for another try (e. g., ucode readback compare error),
                ; these failures count against the maximum error retry count, and a message
                ; is printed for each failure.
                ;
                ; If the port should be taken offline due to exhausted retry count, but
                ; the system device is on this CI or this port is needed to cluster,
                ; then the port driver bugchecks (in routine CLEANUP_PDT.)
                ;
                ; Inputs:
                ;
                ;       R4                      -Addr of port config register
                ;       R5                      -Addr of UCB
                ;       IPL                     -IPL$_SCS
                ;
                ; Outputs:
                ;
                ;       R0-R5                   -Destroyed
                ;       Other registers         -Preserved
                ;-

                        .ENABL  LSB

TEST_SHUTDOWN:

0084 C5   D5    05D0    TSTL    UCB$L_PDT(R5)           ; Already shutdown?
     09   13    05D4    BEQL    10$                     ; Branch if so, no reason to proceed
0080 C5   91    05D6    CMPB    UCB$B_ERTCNT(P5),-      ; Is this the first init of port?
0081 C5         05DA            UCB$B_ERTMAX(R5)        ;
     01   12    05DD    BNEQ    20$                     ; Branch if not
          05    05DF    10$:    RSB                     ; Else return to continue init
                05E0

52   FA76 CF   DE    05E0    20$:    MOVAL   INI$MSG_INIT,R2         ; Get addr of appropriate msg
0080 C5   95    05E5            TSTB    UCB$B_ERTCNT(R5)        ; Retries all used up?
     0A   18    05E9            BGEQ    30$                     ; Branch if not
        0066    30  05EB        BSBW    CLEANUP_PDT             ; Else cleanup PDT-- bugcheck
                05EE                                            ;  if can't proceed without port
52   FAB9 CF   DE    05EE    1157    MOVAL   INI$MSG_OFFL,R2     ; We can proceed
     10   11    05F3    1158    BRB     40$                     ; Go print port offline message
```

```
                        05F5  1159
                        05F5  1160 ;
                        05F5  1161 ; The port reinitialization message is going to be broadcast to _OPA0. Format
                        05F5  1162 ; that portion of the message the contains the number of retries remaining.
                        05F5  1163 ;
                        05F5  1164
            52   DD     05F5  1165 30$:    PUSHL   R2                      ; Save message address
        52  27   C0     05F7  1166         ADDL2   #RETRY_OFFSET,R2        ; Position to retry field in message
    50  0080 C5   9A     05FA  1167         MOVZBL  UCB$B_ERTCNT(R5),R0     ; Retrieve number of retries left
          F9FE'   30     05FF  1168         BSBW    ERR$CNV_HEX_DEC         ; Format the retry field
            52 8ED0     0602  1169         POPL    R2                      ; Restore message address
                        0605  1170
                        0605  1171 ;
                        0605  1172 ; Broadcast the message of choice to _OPA0 after completing the common
                        0605  1173 ; formatting of the message.
                        0605  1174 ;
                        0605  1175
        51  82   9A     0605  1176 40$:    MOVZBL  (R2)+,R1                ; Get message size and address
    50  28 A5   D0     0608  1177         MOVL    UCB$L_DDB(R5),R0        ; Get DDB addr in R0
          17 A0   90     060C  1178         MOVB    DDB$T_NAME+3(R0),-      ; Copy device controller letter from
          06 A2          060F  1179                 CTRLR_NAME(R2)          ;  DDB to ASCII msg
55  00000000'GF   9E     0611  1180         MOVAB   G^OPA$UCB0,R5           ; Set _OPA0 to get msg
    00000000'GF   17     0618  1181         JMP     G^IOC$BROADCAST         ; Send msg to terminal driver
                        061E  1182
                        061E  1183         .DSABL  LSB
```

```
                        061E  1185          .SBTTL  PORT INITIALIZATION ERRORS
                        061E  1186
                        061E  1187 ;+
                        061E  1188 ; Come to PORT_NOTPRES if the NOCI bit is set in the configuration
                        061E  1189 ; register.  The condition can only ever happen on a CI750.  It indicates
                        061E  1190 ; that the port in its external cabinet is uncabled or unpowered.
                        061E  1191 ;-
                        061E  1192
                        061E  1193          .ENABL  LSB
                        061E  1194
                        061E  1195 PORT_NOTPRES:
                        061E  1196
     50   8002 8F   32  061E  1197          CVTWL   #<PAER$K_ES_HWER ! ^X8000>,R0    ; Log as unspecified hardware
            F9DA'   30  0623  1198          BSBW    ELOG$HARDWARE                   ;  error
             002B   31  0626  1199          BRW     CLEANUP_PDT                     ; Go clean up without bothering
                        0629  1200                                                  ;  to do any retries.
                        0629  1201
                        0629  1202 ;+
                        0629  1203 ; Come to CPU_REV_ERROR if the CPU revision level is not sufficient to
                        0629  1204 ; support CI port activity.
                        0629  1205 ;
                        0629  1206 ; Inputs:
                        0629  1207 ;
                        0629  1208 ;       R1                      -Copy of current SID
                        0629  1209 ;       R5                      -UCB address
                        0629  1210 ;-
                        0629  1211
                        0629  1212 CPU_REV_ERROR:
                        0629  1213
00000652'EF     94      0629  1214          CLRB    INI$CPU_REV                     ; Clear flag that says this
                        062F  1215                                                  ;  is regular bugcheck reason --
                        062F  1216                                                  ;  this is reason for UCODEREV bugchk
            F9CE'   30  062F  1217          BSBW    ELOG$CPU_REV                    ; Log bad CPU rev
             001F   31  0632  1218          BRW     CLEANUP_PDT                     ; Go clean up pool because we
                        0635  1219                                                  ;  can't continue
                        0635  1220
                        0635  1221 ;+
                        0635  1222 ; Come to WCS_ERROR if loaded microcode could not be read back
                        0635  1223 ; correctly.
                        0635  1224 ;
                        0635  1225 ; Inputs:
                        0635  1226 ;
                        0635  1227 ;       PA_MADR(R4)             -micro addr that failed
                        0635  1228 ;       PA_MDATR(R4)            -Bad WCS contents
                        0635  1229 ;       R0                      -Copy of good data from pool
                        0635  1230 ;-
                        0635  1231
                        0635  1232 WCS_ERROR:
                        0635  1233
            F9C8'   30  0635  1234          BSBW    ELOG$UCODE_NORD                 ; Log microcode read-back error.
             00B2   31  0638  1235          BRW     RETRY_INIT                      ; Go clean up allocated pool
                        0638  1236
                        0638  1237 ;+
                        0638  1238 ; Port initialization (transition from uninitialized to disabled) did
                        0638  1239 ; not complete correctly.
                        0638  1240 ;-
                        0638  1241
```

```
                     063B  1242 INIT_PORT_FAIL:
                     063B  1243
50   8001 8F   32    063B  1244          CVTWL   #<PAER$K_ES_INIT ! ^X8000>, R0  ; Log failed to change from
        F9BD'  30    0640  1245          BSBW    ELOG$HARDWARE                    ; uninit. to disabled error.
        00A7   31    0643  1246          BRW     RETRY_INIT              ; Go clean up allocated pool
                     0646  1247
                     0646  1248 ;+
                     0646  1249 ; Unable to allocate temporary buffer in which to calculate the
                     0646  1250 ; loopback datagram CRC.
                     0646  1251 ;-
                     0646  1252
                     0646  1253 INIT_LBDG_FAIL:
                     0646  1254
                     0646  1255          ASSUME  PAER$K_ES_POOL EQ 0     ; Log a pool allocation error.
          50   D4    0646  1256          CLRL    R0
        F9B5'  30    0648  1257          BSBW    ELOG$INIT_SWERR
           01  D0    064B  1258          MOVL    #PA_PMC_M_MIN,-         ; Do maint init on port to
      00E8 D4         064D  1259                  @PDT$L_PMC(R4)          ;  make sure it's quiet
           02  11    0650  1260          BRB     CLEANUP_PDT             ; Go clean up allocated pool
                     0652  1261
                     0652  1262          .DSABL  LSB
```

```
                     0652  1264 ;+
                     0652  1265 ; Deallocate template loopback datagram (if any) and PDT (if any).
                     0652  1266 ; Return to caller with device offline, and power fail/power up
                     0652  1267 ; unchanged.
                     0652  1268 ;
                     0652  1269 ; Inputs:
                     0652  1270 ;
                     0652  1271 ;       R5                          -UCB 0 addr
                     0652  1272 ;-
                     0652  1273
                     0652  1274 INI$CPU_REV::
                     0652  1275
 01                  0652  1276         .BYTE   1                           ; 1/0 means CPU rev is okay/not okay
                     0653  1277
                     0653  1278 INI$PORT_REV::
                     0653  1279
 01                  0653  1280         .BYTE   1                           ; 1/0 means port ucode rev is okay/not okay
                     0654  1281                                            ; NOTE: use of this memory flag is not
                     0654  1282                                            ;  quite right in the case with multiple
                     0654  1283                                            ;  ports since there is a fork between
                     0654  1284                                            ;  the point where the flag is set
                     0654  1285                                            ;  in ERR$CRASHPORT and where it is
                     0654  1286                                            ;  set and here.  This means that we
                     0654  1287                                            ;  might take the UCODEREV bugcheck
                     0654  1288                                            ;  with the context for the wrong
                     0654  1289                                            ;  port in hand, not a very serious
                     0654  1290                                            ;  mishap.
                     0654  1291
                     0654  1292 ASSUME  PDT$L_FLINK     EQ 0
                     0654  1293
                     0654  1294         .ENABL  LSB
                     0654  1295
                     0654  1296 CLEANUP_PDT:
                     0654  1297
     52   0084 C5    DO   0654  1298         MOVL    UCB$L_PDT(R5),R2           ; Get PDT addr
                58    13   0659  1299         BEQL    MAYBE_SYS_DEAD             ; Branch if none allocated
 51   00000000'GF    DE   065B  1300         MOVAL   G^SCS$GL_PDT,R1            ; Get base of SCS port list
                     0662  1301
          50   61    DO   0662  1302 10$:    MOVL    (R1),R0                    ; Get next PDT
                0D    13   0665  1303         BEQL    30$                        ; Branch if none, ours wasn't linked
          52   50    D1   0667  1304         CMPL    R0,R2                      ; Is this PDT ours?
                05    13   066A  1305         BEQL    20$                        ; Branch if it is
          51   50    DO   066C  1306         MOVL    R0,R1                      ; Else, save next PDT as previous
                F1    11   066F  1307         BRB     10$                        ; Continue searching down the list
                     0671  1308
          61   62    DO   0671  1309 20$:    MOVL    (R2),(R1)                  ; Remove out PDT from the list
                     0674  1310
     50   0184 C2    DO   0674  1311 30$:    MOVL    PDT$L_LBDG(R2),R0          ; Get loopback dg addr
                06    13   0679  1312         BEQL    40$                        ; Branch if none allocated
      00000000'GF    16   067B  1313         JSB     G^COM$DRVDEALMEM           ; Else deallocate it
                     0681  1314
     51   00E4 C2    DO   0681  1315 40$:    MOVL    PDT$L_CNF(R2),R1           ; Get configuration register addr
                     0686  1316             $PRTCTINI -                        ; Ignore non responding device registers
                     0686  1317                     B^50$,MCHK$M_NEXM          ;  causing machine checks
                01    DO   0696  1318         MOVL    #PA_PMC_M_MIN,-            ;  while MINing the port once
              04 A1         0698  1319                 PA_PMC(R1)                 ;  more just to be sure it's quiet
                     069A  1320             $PRTCTEND 50$                      ;
```

```
                50  00000000'GF   D0  069B  1321          MOVL    G^EXE$GL_SYSUCB,R0      ; Get system device UCB
                    0084 C0       D1  06A2  1322          CMPL    UCB$L_PDT(R0),-         ; Is it via our PDT?
                    0084 C5           06A6  1323                  UCB$L_PDT(R5)           ;
                          08      13  06A9  1324          BEQL    MAYBE_SYS_DEAD          ; Branch if so
                00000000'GF       D5  06AB  1325          TSTL    G^CLU$GL_CLUB           ; No.  Are we clustering?
                          25      13  06B1  1326          BEQL    60$                     ; Branch if not because system can get
                                      06B3  1327                                          ;  by without port
                                      06B3  1328
                                      06B3  1329 MAYBE_SYS_DEAD:
                                      06B3  1330
                00000000'GF       D5  06B3  1331          TSTL    G^SCS$GL_PDT            ; Any SCS speaking PDTs left?
                          1D      12  06B9  1332          BNEQ    60$                     ; Branch if so -- take a chance
                                      06BB  1333                                          ;  that the remaining port(s) will
                                      06BB  1334                                          ;  carry us
            54  0084 C5           D0  06BB  1335          MOVL    UCB$L_PDT(R5),R4        ; Else set up R4 for BUGCHECK
                      8F AF       95  06C0  1336          TSTB    INI$CPU_REV             ; Is this a CPU ucode problem?
                          0C      13  06C3  1337          BEQL    UCODE_BUGCHK            ; Branch if so
                      8B AF       95  06C5  1338          TSTB    INI$PORT_REV            ; Is this a port ucode problem?
                          07      13  06C8  1339          BEQL    UCODE_BUGCHK            ; Branch if so
                                      06CA  1340          BUGCHECK CIPORT,FATAL           ; Else shut the system down with
                                      06D1  1341                                          ;  normal CIPORT bugcheck
                                      06D1  1342
                                      06D1  1343 UCODE_BUGCHK:
                                      06D1  1344
                                      06D1  1345          BUGCHECK UCODEREV,FATAL         ; Shut system down with microcode
                                      06D8  1346                                          ;  revision problem bugcheck
                                      06D8  1347
                    50    52      D0  06D8  1348 60$:     MOVL    R2,R0                   ; Copy PDT addr to R0
            00000000'GF       16  06DB  1349          JSB     G^COM$DRVDEALMEM        ; Deallocate PDT
                0084 C5       D4  06E1  1350          CLRL    UCB$L_PDT(R5)           ; Show PDT as gone
             50   24 A5       D0  06E5  1351          MOVL    UCB$L_CRB(R5),R0        ; Get CRB addr
                  10 A0       D4  06E9  1352          CLRL    CRB$L_AUXSTRUC(R0)      ;  and show no PDT here either
                                      06EC  1353
                          05      06EC  1354 70$:     RSB                             ; Return
                                      06ED  1355
                                      06ED  1356
                                      06ED  1357 ;+
                                      06ED  1358 ; RETRY_INIT is branched to on WCS load error or port init
                                      06ED  1359 ; error.  RETRY_INIT checks for retries left.  If none, it branches
                                      06ED  1360 ; to CLEANUP_PDT.  If retries are left, it branches to INI$PORT
                                      06ED  1361 ; after decrementing the retry count.
                                      06ED  1362 ;
                                      06ED  1363 ; Inputs:
                                      06ED  1364 ;
                                      06ED  1365 ;       R4                      -Cofiguration register addr
                                      06ED  1366 ;       R5                      -UCB 0 addr
                                      06ED  1367 ;-
                                      06ED  1368
                                      06ED  1369 RETRY_INIT:
                                      06ED  1370
                      01      D0  06ED  1371          MOVL    #PA_PMC_M_MIN,-         ; Do maint init on port
                  04 A4               06EF  1372                  PA_PMC(R4)              ;  to make sure it's quiet
                0080 C5       97  06F1  1373          DECB    UCB$B_ERTCNT(R5)        ; Decr # retries left
                      03      18  06F5  1374          BGEQ    90$                     ; Branch if retries left
                    FF5A      31  06F7  1375          BRW     CLEANUP_PDT             ; Else cleanup PDT
                                      06FA  1376
                    FC91      31  06FA  1377 90$:     BRW     INI$PORT                ; Else do another init
```

PORT INITIALIZATION ERRORS

06FD 1378
06FD 1379 .DSABL LSB

```
                     06FD  1381          .SBTTL  INI$FORK
                     06FD  1382
                     06FD  1383 ;+
                     06FD  1384 ; This routine oversees and participates in the creation of a fork process,
                     06FD  1385 ; and the transfer of control at fork IPL to a user supplied address. This
                     06FD  1386 ; routine will use the UCB's fork block dequeuing it from the appropriate fork
                     06FD  1387 ; queue if necessary. The dequeuing of the fork block and creation of the fork
                     06FD  1388 ; process is handled as an atomic event by disabling all interrupts before
                     06FD  1389 ; testing whether the fork block needs to be dequeued, and then re-enabling
                     06FD  1390 ; interrupts after creation of the fork process. After creation of the fork
                     06FD  1391 ; process this routine returns control to the caller.
                     06FD  1392 ;
                     06FD  1393 ; When the fork process commences execution it will do so within this routine.
                     06FD  1394 ; It immediately will transfer control to the user supplied address. The caller
                     06FD  1395 ; of this routine has available R4 in order to pass information across the
                     06FD  1396 ; creation of the fork process to the user routine which will be jumped to at
                     06FD  1397 ; fork IPL.
                     06FD  1398 ;
                     06FD  1399 ; This routine participates in proper synchronization to the fork block by the
                     06FD  1400 ; appropriate setting and clearing of the fork block interlock bit before the
                     06FD  1401 ; fork process is creating, and within the context of the fork process.
                     06FD  1402 ;
                     06FD  1403 ; Inputs:
                     06FD  1404 ;
                     06FD  1405 ;       R3                      -Address to JMP to at fork IPL
                     06FD  1406 ;       R5                      -Address of UCB
                     06FD  1407 ;       IPL                     -Device IPL or higher
                     06FD  1408 ;
                     06FD  1409 ;
                     06FD  1410 ; Outputs:
                     06FD  1411 ;
                     06FD  1412 ;       After creation of fork process but before returning to caller:
                     06FD  1413 ;
                     06FD  1414 ;       R3-R4                   -Destroyed
                     06FD  1415 ;       Other registers         -Preserved
                     06FD  1416 ;       IPL                     -Preserved
                     06FD  1417 ;
                     06FD  1418 ;       Before exit from fork process:
                     06FD  1419 ;
                     06FD  1420 ;       R0-R2                   -Unpredictable
                     06FD  1421 ;       R3                      -User address jumped to at fork IPL
                     06FD  1422 ;       R4                      -User supplied value
                     06FD  1423 ;       R5                      -Address of UCB
                     06FD  1424 ;       IPL                     -Fork IPL
                     06FD  1425 ;-
                     06FD  1426
                     06FD  1427          .ENABL  LSB
                     06FD  1428 INI$FORK::
                     06FD  1429          DSBINT                              ; Disable all interrupts
           01    E3  0703  1430          BBCS    #UCB_V_FKLOCK,-             ; Is the fork block in use?
 03 68 A5            0705  1431                  UCB$W_DEVSTS(R5),10$        ; Branch if not and set in use bit
    55    65   0F    0708  1432          REMQUE  (R5),R5                     ; Remove fork block from its queue
                     070B  1433
0000071D'EF    9F   070B  1434 10$:     PUSHAB  20$                         ; Return address for fork proc creation
00000721'EF    9F   0711  1435          PUSHAB  30$                         ; Fork process PC
00000000'GF    17   0717  1436          JMP     G^EXE$FORK                  ; Create the fork process
                     071D  1437
```

```
                 071D  1438 20$:    ENBINT                          ; Re-enable interrupts
              05 0720  1439          RSB                             ; Return to caller
                 0721  1440
                 0721  1441 ;
                 0721  1442 ; INI$FORK resumes execution at fork IPL within the context of the fork process
                 0721  1443 ; whose creation it has initiated.
                 0721  1444 ;
                 0721  1445
           02    AA 0721  1446 30$:    BICW2   #UCB_M_FKLOCK,-         ; Clear fork block in use bit
        68 A5       0723  1447                 UCB$W_DEVSTS(R5)
           63    17 0725  1448          JMP     (R3)                    ; Transfer control to user supplied addr
                 0727  1449          .DSABL  LSB
                 0727  1450
                 0727  1451          .END
```

```
$$$CURSIZ                =  000001C4
$$$ENTRYNUM              =  0000001E
$$$NEWSIZ                =  000001D0
$$$PREV                  =  0000010C
$$$REGNUM                =  0000000B
$$BASE                   =  00000001
$$DISPL                  =  0000000A
$$GENSW                  =  00000001
$$HIGH                   =  00000009
$$LIMIT                  =  00000008
$$LOW                    =  00000001
$$MNSW                   =  00000001
$$MXSW                   =  00000001
BAD_UCODE                   0000045B R      01
BELL                     =  00000007
BUG$_CIPORT                 ********   X    01
BUG$_UCODEREV               ********   X    01
BUG$_UNSUPRTCPU             ********   X    01
BUILD_PDT                   000001CE R      01
BUILD_STRUCT                000001A7 R      01
BUILD_TLB                   0000031E R      01
CHECK_QUEUES                000003E5 R      01
CHECK_SYSTEMID              000001B8 R      01
CHECK_UCODE                 0000043C R      01
CI_750                      00000147 R      01
CI_780                      00000161 R      01
CI_785                      00000151 R      01
CI_790                      00000158 R      01
CLEANUP_PDT                 00000654 R      01
CLU$GL_CLUB                 ********   X    01
CNF$CALLINTDUE              ********   X    01
CNF$CALL_POLLSW             ********   X    01
COM$DRVDEALMEM              ********   X    01
CPU_REV_ERROR               00000629 R      01
CPU_REV_OK                  000003D3 R      01
CR                       =  0000000D
CRB$L_AUXSTRUC           =  00000010
CRB$L_INTD               =  00000024
CRC_TABLE                   000000D1 R      01
CTRLR_NAME               =  00000006
CXB$C_HEADER             =  00000048
DDB$T_NAME               =  00000014
DTS_CI750                =  00000002
DTS_CI780                =  00000001
DYN$C_CIDG               =  0000003B
DYN$C_SCS                =  00000060
DYN$C_SCS_PDT            =  00000005
ELOG$CPU_REV                ********   X    01
ELOG$HARDWARE               ********   X    01
ELOG$INIT_SWERR             ********   X    01
ELOG$UCODE_NORD             ********   X    01
ERR$BUGCHECK                ********   X    01
ERR$CNV_HEX_DEC             ********   X    01
ERR$DEBUGCHECK              ********   X    01
ERR$PWF_RECOV               ********   X    01
ERR$V_DEB_NEPQ              ********   X    01
EXE$ALLONONPAGED            ********   X    01
EXE$FORK                    ********   X    01
EXE$GB_CPUDATA              ********   X    01
EXE$GB_CPUTYPE              ********   X    01
EXE$GL_SYSUCB               ********   X    01
EXE$GL_TENUSEC              ********   X    01
EXE$GL_UBDELAY              ********   X    01
EXE$MCHK_PRTCT              ********   X    01
FPC$ACCEPT                  ********   X    01
FPC$ALLOCDG                 ********   X    01
FPC$ALLOCMSG                ********   X    01
FPC$CONNECT                 ********   X    01
FPC$DCONNECT                ********   X    01
FPC$DEALLOCDG               ********   X    01
FPC$DEALLOMSG               ********   X    01
FPC$DEALRGMSG               ********   X    01
FPC$MAINTFCN                ********   X    01
FPC$MAP                     ********   X    01
FPC$MAPBYPASS               ********   X    01
FPC$MAPIRP                  ********   X    01
FPC$MAPIRPBYP               ********   X    01
FPC$MRESET                  ********   X    01
FPC$MSTART                  ********   X    01
FPC$QUEUEDG                 ********   X    01
FPC$QUEUEMDGS               ********   X    01
FPC$RCHMSGBUF               ********   X    01
FPC$RCLMSGBUF               ********   X    01
FPC$READCOUNT               ********   X    01
FPC$REJECT                  ********   X    01
FPC$REQDATA                 ********   X    01
FPC$RLSCOUNT                ********   X    01
FPC$SENDDATA                ********   X    01
FPC$SENDDG                  ********   X    01
FPC$SENDMSG                 ********   X    01
FPC$SENDRGDG                ********   X    01
FPC$SNDCNTMSG               ********   X    01
FPC$STOP_VCS                ********   X    01
FPC$UNMAP                   ********   X    01
INI$CPU_REV                 00000652 RG     01
INI$FORK                    000006FD RG     01
INI$MSG_INIT                0000005A RG     01
INI$MSG_OFFL                000000AB RG     01
INI$PORT                    0000038E RG     01
INI$PORT_REV                00000653 RG     01
INI$T_HWTYPE                00000056 RG     01
INIT_CRB                    00000381 R      01
INIT_CTLR                   0000016F RG     01
INIT_DFREEQ                 0000055B R      01
INIT_LBDG_CRC               000004E7 R      01
INIT_LBDG_FAIL              00000646 R      01
INIT_PORT_FAIL              0000063B R      01
IOC$BROADCAST               ********   X    01
IOC$THREADCRB               ********   X    01
IPL$_SCS                 =  00000008
LF                       =  0000000A
LOAD_UCODE                  0000C 04 R      01
MAYBE_SYS_DEAD              00000B83 R      01
MCHK$M_NEXM              =  00000004
```

```
MIN_750_REV                     =  00000061
MMG$GL_GPTBASE                     ********    X    01
MMG$GL_SPTBASE                     ********    X    01
OPA$UCB0                           ********    X    01
OTHER_CPU                          0000016D R       01
PA$CTLINIT                         0000016E RG      01
PA$REGOFFSET                       0000003E RG      01
PA$SCSOFFSET                       00000000 RG      01
PA$UNITINIT                        00000111 RG      01
PAER$K_ES_CODE                  =  00000001
PAER$K_ES_HWER                  =  00000002
PAER$K_ES_INIT                  =  00000001
PAER$K_ES_LST0                  =  00000003
PAER$K_ES_LST1                  =  00000009
PAER$K_ES_LST2                  =  00000007
PAER$K_ES_LST3                  =  00000009
PAER$K_ES_LST4                  =  0000000C
PAER$K_ES_POOL                  =  00000000
PAER$K_ES_SCSID                 =  00000002
PAER$K_ET_DALT                  =  00000003
PAER$K_ET_LMLT                  =  00000042
PA_CNF                             00000000
PA_CNF_M_NOCI                   =  00001000
PA_CQ0                             00000908
PA_CQ1                             0000090C
PA_CQ2                             00000910
PA_CQ3                             00000914
PA_C_MCACHESZ                   =  00000003
PA_C_UCODEST                    =  00000400
PA_DFQ                             00000928
PA_MADR                            00000014
PA_MDATR                           00000018
PA_MFQ                             0000092C
PA_MTC                             00000930
PA_MTEC                            00000934
PA_PDC                             00000920
PA_PEC                             0000091C
PA_PEC_M_PEC                    =  00000001
PA_PESR                            0000093C
PA_PFAR                            00000938
PA_PIC                             00000924
PA_PIC_M_PIC                    =  00000001
PA_PMC                             00000004
PA_PMC_M_MIE                    =  00000004
PA_PMC_M_MIF                    =  00000008
PA_PMC_M_MIN                    =  00000001
PA_PMC_M_MTD                    =  00000002
PA_PMC_M_PSA                    =  00000040
PA_PPR                             00000940
PA_PQBBR                           00000904
PA_PS                              00000900
PA_PSR                             00000918
PA_PSR_M_PSC                    =  00000001
PA_PS_M_PIC                     =  00000008
PDT$B_DGIMAP                       00000154
PDT$B_HSHUT_DG                     000001B0
PDT$B_MAX_PORT                     0000017C
PDT$B_NXT_PORT                     0000017E
PDT$B_PO_LBSTS                     00000180
PDT$B_P1_LBSTS                     00000181
PDT$B_PDT_TYPE                  =  00000007
PDT$B_PLOGMAP                      00000134
PDT$B_PORTMAP                      00000114
PDT$B_PORT_NUM                     0000017D
PDT$B_REQIDPS                      0000017F
PDT$B_SUBTYP                    =  0000000B
PDT$B_TYPE                      =  0000000A
PDT$C_LENGTH                    =  000000E4
PDT$C_PA                        =  00000001
PDT$C_PALENGTH                  =  00000360
PDT$C_PAREGBASE                    000000E4
PDT$C_PAREGEND                     00000110
PDT$C_PQB                       =  000001E0
PDT$C_SCSBASE                   =  0000000C
PDT$C_SCSEND                    =  00000084
PDT$L_ACCEPT                    =  0000000C
PDT$L_ALLOCDG                   =  00000010
PDT$L_ALLOCMSG                  =  00000014
PDT$L_CNF                          000000E4
PDT$L_CONNECT                   =  00000018
PDT$L_CQ0                          000000F0
PDT$L_CQ1                          000000F4
PDT$L_DCONNECT                  =  00000028
PDT$L_DEALLOCDG                 =  0000001C
PDT$L_DEALLOMSG                 =  00000020
PDT$L_DEALRGMSG                 =  00000024
PDT$L_DFQ                          000000FC
PDT$L_DFQHDR                       00000208
PDT$L_DGHDRSZ                      00000190
PDT$L_DGNETHD                      00000194
PDT$L_DGOVRHD                   =  000000B8
PDT$L_DQELOGOUT                    000002E0
PDT$L_FLINK                     =  00000000
PDT$L_GPTBASE                      0000022C
PDT$L_GPTLEN                       00000230
PDT$L_LBDG                         00000184
PDT$L_MAINTFCN                  =  00000078
PDT$L_MAP                       =  0000002C
PDT$L_MAPBYPASS                 =  00000030
PDT$L_MAPIRP                    =  00000034
PDT$L_MAPIRPBYP                 =  00000038
PDT$L_MAXBCNT                   =  000000BC
PDT$L_MFQ                          00000100
PDT$L_MFQHDR                       0000020C
PDT$L_MQELOGOUT                    00000320
PDT$L_MRESET                    =  00000070
PDT$L_MSGHDRSZ                  =  000000B4
PDT$L_MSTART                    =  00000074
PDT$L_MTC                          00000104
PDT$L_PFAR                         00000108
PDT$L_PMC                          000000E8
PDT$L_POLLERDUE                    0000018C
PDT$L_POOLDUE                      00000188
PDT$L_PPR                          0000010C
```

```
PDT$L_PS                        000000EC
PDT$L_PSR                       000000F8
PDT$L_QUEUEDG                 = 0000003C
PDT$L_QUEUEMDGS               = 00000040
PDT$L_RCHMSGBUF               = 00000044
PDT$L_RCLMSGBUF               = 00000048
PDT$L_READCOUNT               = 00000068
PDT$L_REJECT                  = 0000004C
PDT$L_REQDATA                 = 00000050
PDT$L_RLSCOUNT                = 0000006C
PDT$L_SENDDATA                = 00000054
PDT$L_SENDDG                  = 00000058
PDT$L_SENDMSG                 = 0000005C
PDT$L_SENDRGDG                = 0000007C
PDT$L_SNDCNTMSG               = 00000060
PDT$L_SPTBASE                   00000224
PDT$L_SPTLEN                    00000228
PDT$L_STOP_VCS                = 00000080
PDT$L_UCBO                    = 000000DC
PDT$L_UNMAP                   = 00000064
PDT$L_VBDT                      0000021C
PDT$L_VPQB                      00000218
PDT$L_WAITQBL                 = 000000B0
PDT$L_WAITQFL                 = 000000AC
PDT$M_CUR_LBS                 = 00000001
PDT$M_LBDG                    = 00000004
PDT$M_PRV_LBS                 = 00000002
PDT$M_PUP                     = 00000002
PDT$M_PWF_CLNUP               = 00000001
PDT$Q_COMQ2                     000001F0
PDT$Q_COMQ3                     000001F8
PDT$Q_COMQBASE                  000001E0
PDT$Q_COMQH                     000001E8
PDT$Q_COMQL                     000001E0
PDT$Q_DFREEQ                    000001D0
PDT$Q_FORMPB                    00000174
PDT$Q_MFREEQ                    000001D8
PDT$Q_RSPQ                      00000200
PDT$Q_TEMP_RSPQ                 0000019C
PDT$V_PWF_CLNUP               = 00000000
PDT$W_BDTLEN                    00000220
PDT$W_DQELEN                    00000210
PDT$W_LPORT_STS                 00000110
PDT$W_MQELEN                    00000214
PDT$W_PBCOUNT                   00000112
PDT$W_SIZE                    = 00000008
PDT$W_STDGDYN                   00000198
PDT$W_STDGUSED                  0000019A
PORT_NOTPRES                    0000061E  R      01
PPD$B_DEF_ST                    0000001C
PPD$B_FLAGS                     0000000F
PPD$B_HWVERS                    00000034
PPD$B_LBDATA                    00000012
PPD$B_LCB_0                     00000012
PPD$B_LCB_LPORT                 00000010
PPD$B_LCB_NPORT                 0000000F
PPD$B_LCB_OPC                   00000011
PPD$B_LCB_PORT                  0000000E
PPD$B_OPC                       0000000E
PPD$B_PORT                      0000000C
PPD$B_PROTOCOL                  0000001A
PPD$B_RSTATE                    00000025
PPD$B_RST_PORT                  00000024
PPD$B_STATUS                    0000000D
PPD$B_SWFLAG                    0000000B
PPD$B_SYSTEMID                  00000014
PPD$B_TYPE                      0000000A
PPD$C_LBDAT_LEN               = 00000030
PPD$C_LB_LENGTH                 00000046
PPD$C_LCB_DATA                  00000013
PPD$C_LENGTH                    00000012
PPD$C_MIN_DGSIZ                 00000050
PPD$C_PSPO                    = 00000001
PPD$C_SNDLB                   = 0000000D
PPD$K_LB_LENGTH                 00000046
PPD$K_LENGTH                    00000012
PPD$L_BLINK                     00000004
PPD$L_DG_DISC                   00000028
PPD$L_FLINK                     00000000
PPD$L_IN_VCD                    00000018
PPD$L_LBCRC                     00000042
PPD$L_PO_ACK                    00000010
PPD$L_PO_NAK                    00000014
PPD$L_PO_NRSP                   00000018
PPD$L_P1_ACK                    0000001C
PPD$L_P1_NAK                    00000020
PPD$L_P1_NRSP                   00000024
PPD$L_REC_BOFF                  00000028
PPD$L_REC_NAME                  00000024
PPD$L_RPORT_FCN                 00000020
PPD$L_RPORT_REV                 0000001C
PPD$L_RPORT_TYP                 00000018
PPD$L_SND_BOFF                  00000020
PPD$L_SND_NAME                  0000001C
PPD$L_ST_ADDR                   00000018
PPD$L_XCT_LEN                   00000018
PPD$Q_CURTIME                   00000048
PPD$Q_NODENAME                  00000040
PPD$Q_SWINCARN                  00000028
PPD$Q_XCT_ID                    00000010
PPD$T_HWTYPE                    00000030
PPD$T_SWTYPE                    00000020
PPD$T_SWVERS                    00000024
PPD$W_LCB_LEN7                  0000000C
PPD$W_LENGTH                    00000010
PPD$W_MASK                      00000010
PPD$W_MAXDG                     0000001C
PPD$W_MAXMSG                    0000001E
PPD$W_MTYPE                     00000012
PPD$W_M_VAL                     00000014
PPD$W_SIZE                      00000008
PR$_IPL                       = 00000012
PR$_SBR                       = 0000000C
PR$_SID                       = 0000003E
```

```
PR$_SID_TYP730                  = 00000003
PR$_SID_TYP750                  = 00000002
PR$_SID_TYP780                  = 00000001
PR$_SID_TYP785                  = 00000009
PR$_SID_TYP790                  = 00000004
PR$_SID_TYPMAX                  = 00000008
PR$_SLR                         = 0000000D
RETRY_INIT                        000006ED R      01
RETRY_OFFSET                    = 00000027
SCS$ACL_FRDGS                     ********    X   01
SCS$ALL_FRMSGS                    ********    X   01
SCS$GB_PASANITY                   ********    X   01
SCS$GB_SYSTEMID                   ********    X   01
SCS$GL_BDT                        ********    X   01
SCS$GL_MCADR                      ********    X   01
SCS$GL_PDT                        ********    X   01
SCS$GL_SCSSIZE                    ********    X   01
SCS$GW_BDTCNT                     ********    X   01
SCS$GW_MAXDG                      ********    X   01
SCS$GW_MAXMSG                     ********    X   01
SCS$GW_PAPPDDG                    ********    X   01
SCS$INITIAL                       ********    X   01
SGN$GL_MAXGPGCT                   ********    X   01
SIZ...                          = 00000001
SS$_NORMAL                      = 00000001
SS$_POWERFAIL                   = 00000364
START_UCODE                       00000466 R      01
TEST_SHUTDOWN                     000005D0 R      01
UCB$B_DEVTYPE                   = 00000041
UCB$B_ERTCNT                    = 00000080
UCB$B_ERTMAX                    = 00000081
UCB$B_FIPL                      = 0000000B
UCB$B_LMERTCNT                    000000D2
UCB$B_LMERTMAX                    000000D3
UCB$B_LMEST                       000000D0
UCB$B_LMET                        000000D1
UCB$K_ERRDGBYTS                 = 000000B4
UCB$K_LMPKTBYTS                 = 00000040
UCB$L_CICMD                       000000F0
UCB$L_CRB                       = 00000024
UCB$L_DDB                       = 00000028
UCB$L_DPC                       = 0000009C
UCB$L_MSGFKBLK                    000000A0
UCB$L_PDT                       = 00000084
UCB$M_ONLINE                    = 00000010
UCB$N_LSADDR                      000000D8
UCB$N_LSID                        000000DE
UCB$N_RSADDR                      000000E4
UCB$N_RSID                        000000EA
UCB$T_MSGDATA                     000000F8
UCB$T_OPA0_TEMP                   000000B8
UCB$W_DEVSTS                    = 00000068
UCB$W_LMERRCNT                    000000D4
UCB$W_MSGBYTCNT                   000000F4
UCB$W_MSGPPDTYP                   000000F6
UCB$W_STS                       = 00000064
UCB$W_UNIT                      = 00000054
UCB_M_FKLOCK                    = 00000002
UCB_V_FKLOCK                    = 00000001
UCODE_BUGCHK                      000006D1 R      01
VEC$L_INITIAL                   = 0000000C
WCS_ERROR                         00000635 R      01
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
$$$115_DRIVER	00000727 (1831.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
ABS	00000944 (2372.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	32	00:00:00.06	00:00:00.81
Command processing	110	00:00:00.44	00:00:05.45
Pass 1	552	00:00:17.00	00:01:07.80
Symbol table sort	0	00:00:02.03	00:00:11.66
Pass 2	264	00:00:03.70	00:00:12.64
Symbol table output	47	00:00:00.23	00:00:01.76
Psect synopsis output	2	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1009	00:00:23.49	00:01:40.39

The working set limit was 2250 pages.
134834 bytes (264 pages) of virtual memory were used to buffer the intermediate code.
There were 110 pages of symbol table space allocated to hold 1967 non-local and 52 local symbols.
1451 source lines were read in Pass 1, producing 23 object records in Pass 2.
51 pages of virtual memory were used to define 45 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[DRIVER.OBJ]PALIB.MLB;1	7
_255DUA28:[SYS.OBJ]LIB.MLB;1	17
_255DUA28:[SYSLIB]STARLET.MLB;2	10
TOTALS (all libraries)	34

2338 GETS were required to define 34 macros.

There were no errors, warnings or information messages.

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

PAINIT
LIS

PAFPCALL
LIS

PAMONIT
LIS

PAINTR
LIS