C 8

```
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  EEEEEEEEEEEE     RRRRRRRRRRRR
DDD         DDD  RRRRRRRRRRRR           III        VVV         VVV  EEEEEEEEEEEE     RRRRRRRRRRRR
DDD         DDD  RRRRRRRRRRRR           III        VVV         VVV  EEEEEEEEEEEE     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
```

```
**FILE**ID**PAINIT                                   D  6
```

```
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

```
0000            .TITLE  PAINIT
0000            .IDENT  'V04-001'
0000
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000    ;++
0000    ;
0000    ; FACILITY:
0000    ;
0000    ;       VAX/VMS EXECUTIVE, I/O DRIVERS
0000    ;
0000    ; ABSTRACT: CI PORT INITIALIZATION
0000    ;
0000    ; AUTHOR:  N. KRONENBERG,  MAY 1981
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V04-001 NPK3066         N. Kronenberg                   9-Sep-1984
0000    ;               Add flags INI$CPU/PORT_REV.  Flags = 1/0 if ucode is
0000    ;               okay/insufficient.  Used to trigger UCODEREV bugcheck
0000    ;               rather than usual CIPORT bugcheck if bugcheck is needed.
0000    ;               Set INI$CPU_REV to okay just before CPU rev check; clear
0000    ;               if check fails prior to calling CLEANUP_PDT.
0000    ;               Set INI$PORT_REV when port is successfully init'ed assuming
0000    ;               its ucode is okay.  Clear in PACONFIG when we have
0000    ;               checked port ucode rev and determined it is bad.
0000    ;
0000    ;       V03-034 NPK3064         N. Kronenberg                   21-Aug-1984
0000    ;               On cpu powerfail recovery (when port may still be
0000    ;               alive if it was an unswitched power failure), min
0000    ;               the port before dropping ipl to reinit.
0000    ;
0000    ;       V03-033 NPK3062         N. Kronenberg                   10-Aug-1984
0000    ;               Fix yet another bug in leaving port offline, but
0000    ;               continuing to run the system.
0000    ;
```

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

```
0000     ;                error logging of this error condition has been modified to
0000     ;                notice that the existance of this error should also be broadcast
0000     ;                to _OPA0, and does so.
0000     ;
0000     ;                I have modified the routine TEST_SHUTDOWN so that the port
0000     ;                re-initialization message that is broadcast to _OPA0 includes
0000     ;                the number of retries left.
0000     ;
0000     ;        V03-022 TMK0003         Todd M. Katz            21-Feb-1984
0000     ;                Change unit and port initialization so that they proceed at fork
0000     ;                IPL instead of at IPL$_POWER. This requires these changes:
0000     ;
0000     ;                1. Add a new routine INI$FORK. This routine is assumed to be
0000     ;                   called at elevated IPL with a routine address in R3 which is
0000     ;                   to be jumped to at fork IPL. INI$FORK will extract the fork
0000     ;                   block from the appropriate fork queue in an atomic fashion,
0000     ;                   if it has to, and create a fork process before returning
0000     ;                   to its caller. When the fork process resumes, it does so
0000     ;                   within INI$FORK, which proceeds to jump to the routine
0000     ;                   address passed to it as input. Throughout this procedure
0000     ;                   proper use is made of the fork block interlock bit.
0000     ;
0000     ;                2. If the unit initialization routine has been called and the
0000     ;                   port has not yet been initialized then all miscellaneous
0000     ;                   errors within the configuration register are cleared, device
0000     ;                   interrupts are disabled by placing the port in the
0000     ;                   un-initialized state, and the routine INI$FORK is called
0000     ;                   so that the remained of the unit initialization maybe done
0000     ;                   at fork IPL.
0000     ;
0000     ;                3. Because port initialization proceeds at fork IPL there is
0000     ;                   no longer any need to fork in order to print out messages
0000     ;                   to OPA0.
0000     ;
0000     ;                4. Because port initialization proceeds at fork IPL,
0000     ;                   EXE$ALONONPAGED maybe called to allocate whatever non-paged
0000     ;                   pool needs to be allocated. This means that the routine
0000     ;                   HIPL_ALLPOOL (INI$HIPLALC) maybe deleted, and that the funny
0000     ;                   games that were being played with the IPL value in the pool
0000     ;                   header area, so that the allocation of free datagrams and
0000     ;                   sequence messages could proceed at IPL$_POWER, can be
0000     ;                   stopped.
0000     ;
0000     ;                5. Because port initialization used to proceed at IPL$_POWER, it
0000     ;                   never mattered when device interrupts were being enabled.
0000     ;                   However, port initialization is now being done at fork IPL
0000     ;                   so it has become important not to enable device interrupts
0000     ;                   until after everything else has been done and just before
0000     ;                   port initialization terminates.
0000     ;
0000     ;                6. It becomes an implicit assumption, that INI$PORT is only
0000     ;                   called at fork IPL with device interrupts disabled, and with
0000     ;                   no outstanding interrupts.
0000     ;
0000     ;        V03-021 TMK0002         Todd M. Katz            17-Feb-1984
0000     ;                Change the text of the message that is printed out on the
0000     ;                operator's console when it is discovered that SCSSYSTEMID
```

```
0000   ;             is unitialized, and has a value of 0.
0000   ;
0000   ;    V03-020 NPK3044         N. Kronenberg             6-Feb-1984
0000   ;             Replace queuing of 3 gratituous datagrams to the port
0000   ;             free queue (to fill the cache) with queuing of an
0000   ;             additional SCS$GW_PAPPDDG datagrams.  The additional
0000   ;             datagrams are intended to handle error log datagrams
0000   ;             not associated with any particular connection.
0000   ;
0000   ;    V03-019 TMK0001         Todd M. Katz              27-Jan-1984
0000   ;             Before allocating the PDT, check for a SCSSYSTEMID of 0.
0000   ;             If such a SCSSYSTEMID is found, log the error condition,
0000   ;             notify the operator's console via an appropriate set of
0000   ;             messages, and keep the port off-line.
0000   ;
0000   ;    V03-018 NPK3039         N. Kronenberg             11-Jan-1984
0000   ;             Remove return of top unused portion of pool before PDT
0000   ;             back to pool.  No return can be done if PDT allocated
0000   ;             from LRP, so never return.
0000   ;
0000   ;    V03-017 NPK3037         N. Kronenberg             11-Nov-1983
0000   ;             Comment inputs to INI$PORT.
0000   ;             Add check to INI$PORT that command queues and response
0000   ;             queue are empty prior to starting port.  If queues
0000   ;             aren't empty, attempt recovery by setting them empty.
0000   ;
0000   ;    V03-016 NPK3035         N. Kronenberg             21-Oct-1983
0000   ;             Fix calculation of global page table length.
0000   ;
0000   ;    V03-015 TCM0002         Trudy C. Matthews         19-Aug-1983
0000   ;             Add SUPERSTAR-specific path to CPU-dependent code that sets
0000   ;             CPU type and port device type.
0000   ;
0000   ;    V03-014 NPK3029         N. Kronenberg             14-Jul-1983
0000   ;             Numerous enhancements for V4.0.
0000   ;             Add fork process call, SENDRGDG, to SCS offset table.
0000   ;             Set max block xfer byte count in PDT.
0000   ;             Allow sanity timer to be enabled.
0000   ;             Add routine TEST_SHUTDOWN to check if port can be
0000   ;             reinitialized or must be left offline and to print
0000   ;             operator warning if appropriate.
0000   ;             Add init of fork ipl for msg fork block in ucb.
0000   ;             Add $PAUCBDEF and $DDBDEF.
0000   ;
0000   ;    V03-013 NPK3024         N. Kronenberg             18-May-1983
0000   ;             Add comments explaining variable network header.
0000   ;
0000   ;    V03-012 KTA3046         Kerbey T. Altmann         03-Apr-1983
0000   ;             Redo for SCS/PPD split.
0000   ;
0000   ;    V03-011 TCM0001         Trudy C. Matthews         29-Feb-1983
0000   ;             Added an 11/790-specific path to CPUDISP macro which sets
0000   ;             CPU type and port device type.
0000   ;
0000   ;    V03-010 NPK3021         N. Kronenberg             28-Feb-1983
0000   ;             Fix setting of 'V750' cpu type.
0000   ;
```

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

```
                  0000   286 ;--
```

```
    0000            .SBTTL  DEFINITIONS
    0000
    0000    ;
    0000    ; Set PSECT to driver code:
    0000    ;
    0000
00000000            .PSECT  $$$115_DRIVER,LONG
    0000
    0000    ;
    0000    ; System definitions (LIB.MLB):
    0000    ;
    0000
    0000            .nocross
    0000            $CRBDEF                         ; Channel Request Block format
    0000            $CXBDEF                         ; Complex Buffer format
    0000            $DCDEF                          ; Device type codes
    0000            $DDBDEF                         ; Device Data Block format
    0000            $DYNDEF                         ; Structure type codes
    0000            $IPLDEF                         ; Define interrupt priorities
    0000            $MCHKDEF                        ; Protect from machine check definitions
    0000            $PDTDEF                         ; Port Descriptor Table format
    0000            $PRDEF                          ; Internal processor register definitions
    0000            $SSDEF                          ; System service success codes
    0000            $UCBDEF                         ; Unit Control Block definitions
    0000            $VECDEF                         ; CRB vector dispatch block offsets
    0000
    0000    ;
    0000    ; PADRIVER definitions (PALIB.MLB):
    0000    ;
    0000
    0000            $PAERDEF                        ; Port driver error code values
    0000            $PAPDTDEF                       ; Port -specific PDT extension
    0000            $PAREGDEF                       ; CI port register definitions
    0000            $PAUCBDEF                       ; PA specific extension to UCB
    0000            $PPDDEF                         ; PPD message layer
    0000
```

```
0000                   .SBTTL  TABLES OF INITIALIZATION DATA
0000
0000      ;+
0000      ; The following table gives word offsets for fork process SCS calls.
0000      ; Offsets are relative to the address of the controller initialization
0000      ; routine, PA$CTLINIT.
0000      ;-
0000
0000      ;
0000      ; Macro to generate the table and ASSUME statements about PDT format:
0000      ;
0000
0000                   .MACRO  SCS_OFFSET_TAB  ENTRY_LIST
0000
0000                    $$$ENTRYNUM=0                               ; No entries in table yet
0000                    .IRP    ENTRY ENTRY_LIST                    ; For each entry in the list...
0000                    .WORD   <FPC$'ENTRY'-PA$CTLINIT>            ;   insert offset from ctl init,
0000                    .IF     NE $$$ENTRYNUM                      ;   and for entries after the 1st
0000                    ASSUME $$$PREV+4 EQ PDT$L_'ENTRY'           ;   specify assumed PDT adjacency
0000                    .ENDC                                       ;
0000                    $$$PREV=PDT$L 'ENTRY'                       ; Set previous entry as this entry
0000                    $$$ENTRYNUM=$$$ENTRYNUM+1                   ; Step entry count
0000                    .ENDR                                       ;
0000
0000                    ASSUME $$$PREV+4 EQ PDT$C_SCSEND            ; Final PDT assumption
0000                    .WORD  0                                    ; Offset table terminator
0000
0000                   .ENDM   SCS_OFFSET_TAB                       ;
0000
0000      ;
0000      ; Table itself:
0000      ;
0000
0000      PA$SCSOFFSET::
0000
0000                   SCS_OFFSET_TAB  <-                           ; Invoke macro to define offsets
0000                           ACCEPT,-
0000                           ALLOCDG,-
0000                           ALLOCMSG,-
0000                           CONNECT,-
0000                           DEALLOCDG,-
0000                           DEALLOMSG,-
0000                           DEALRGMSG,-
0000                           DCONNECT,-
0000                           MAP,-
0000                           MAPBYPASS,-
0000                           MAPIRP,-
0000                           MAPIRPBYP,-
0000                           QUEUEDG,-
0000                           QUEUEMDGS,-
0000                           RCHMSGBUF,-
0000                           RCLMSGBUF,-
0000                           REJECT,-
0000                           REQDATA,-
0000                           SENDDATA,-
0000                           SENDDG,-
0000                           SENDMSG,-
```

```
0000                    SNDCNTMSG,-
0000                    UNMAP,-
0000                    READCOUNT,-
0000                    RLSCOUNT,-
0000                    MRESET,-
0000                    MSTART,-
0000                    MAINTFCN,-
0000                    SENDRGDG,-
0000                    STOP_VCS,-
0000                    >
003E
003E    ;+
003E    ; The following table is a list of word offsets from the CI configuration
003E    ; register of CI register addresses to be kept in the PDT for quick access.
003E    ;-
003E
003E    ;
003E    ; Macro to generate table:
003E    ;
003E
003E            .MACRO  REG_OFFSET_TAB  REG_LIST
003E
003E             $$$REGNUM=0                                    ; # table entries =0
003E             .IRP   REG REG_LIST                            ; For each register in list,
003E             .WORD  <PA_'REG'-PA_CNF>                       ;  enter offset from config reg
003E             .IF    NE $$$REGNUM                            ;  and for entries after first,
003E             ASSUME $$$PREV+4 EQ PDT$L_'REG'                ;  verify PDT adjacency
003E             .ENDC                                          ;
003E
003E             $$$PREV =PDT$L_'REG'                           ; Set this PDT entry to previous
003E             $$$REGNUM=$$$REGNUM+1                          ; Step entry counter
003E             .ENDR                                          ;
003E
003E             ASSUME $$$PREV+4 EQ PDT$C_PAREGEND             ; Verify final PDT offset
003E
003E             .WORD  -1                                      ; Table terminator
003E
003E            .ENDM   REG_OFFSET_TAB
003E
003E    ;
003E    ; Table itself:
003E    ;
003E
003E    PA$REGOFFSET::
003E
003E            REG_OFFSET_TAB  <-                              ; Invoke macro to define offsets
003E                    CNF,-                                   ;  Configuration register
003E                    PMC,-                                   ;  Port maint control reg
003E                    PS,-                                    ;  Port status register
003E                    CQ0,-                                   ;  Command queue 0 control
003E                    CQ1,-                                   ;  Command queue 1 control
003E                    PSR,-                                   ;  Port status release register
003E                    DFQ,-                                   ;  Datagram free queue control
003E                    MFQ,-                                   ;  Message free queue control
003E                    MTC,-                                   ;  Maint timer control
003E                    PFAR,-                                  ;  Fail address register
003E                    PPR,-                                   ;  Port parameter register
```

```
                                    003E                >
                                    0056                .cross
                                    0056        ;
                                    0056        ; Processor type in ASCII for start handshakes:
                                    0056        ;
                                    0056
                                    0056        INI$T_HWTYPE::
                                    0056
                        30 38 37 56 0056                .ASCII  'V780'                          ; Assume 'V780' to start with
                                    005A
                           00000061 005A        MIN_750_REV = 97                                ; Minimum 11/750 CPU ucode
                                    005A                                                        ;  rev level to support ci750
                                    005A
                                    005A        ;
                                    005A        ; Messages to send to _OPA0 on serious port errors:
                                    005A        ;
                                    005A
                           0000000D 005A        CR      = 13                                    ; ASCII for carriage return,
                           0000000A 005A        LF      = 10                                    ;  linefeed,
                           00000007 005A        BELL    = 7                                     ;  and bell
                                    005A
                                    005A        INI$MSG_INIT::
                                    005A
43 20 2C 30 78 41 50 25 07 0A 0D 00' 005A               .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
                                    00AB        INI$MSG_OFFL::
                                    00AB
43 20 2C 30 78 41 50 25 07 0A 0D 00' 00AB               .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
                           00000027 00D1        RETRY_OFFSET    = 39                            ; Byte offset to retry count
                                    00D1                                                        ; numerical field in port
                                    00D1                                                        ; re-initialization message
                                    00D1
                           00000006 00D1        CTRLR_NAME = 6                                  ; Byte offset to device
                                    00D1                                                        ;  controller letter (x)
                                    00D1                                                        ;  in above msgs
                                    00D1
                                    00D1
                                    00D1        ;
                                    00D1        ; Polynomial table used to calculate CRC for loopback datagram:
                                    00D1        ;
                                    00D1
                                    00D1        CRC_TABLE:
                                    00D1
                           00000000 00D1                .LONG   0
                           1DB71064 00D5                .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          .SBTTL  UNIT_INIT
                            0111
                            0111  ;+
                            0111  ; The device type is set in the UCB depending on what CPU we are running
                            0111  ; on.  If this is an 11/780, then the device type is CI780; if 11/750,
                            0111  ; then the device type is CI750.  If the CPU is other than an 11/780,
                            0111  ; alter the ASCII CPU type accordingly.  If this is some other CPU, we really
                            0111  ; shouldn't ever get here, so we just exit leaving the unit offline so
                            0111  ; it can't be used.
                            0111  ;
                            0111  ; Then the device is initialized by calling subroutine INIT_CTLR.
                            0111  ;
                            0111  ; Inputs:
                            0111  ;
                            0111  ;       R3                              -Address of PA configuration reg
                            0111  ;       R4                              -Same as R3
                            0111  ;       R5                              -Addr of UCB
                            0111  ;
                            0111  ; Outputs:
                            0111  ;
                            0111  ;       R0-R3                           -Destroyed
                            0111  ;       Other registers                 -Preserved
                            0111  ;
                            0111  ;       INI$T_HWTYPE                    -Set to proper CPU type in ASCII
                            0111  ;-
                            0111
                            0111  ASSUME  DT$_CI750  EQ  DT$_CI780+1
                            0111
                            0111          .ENABL  LSB
                            0111
                            0111  PA$UNITINIT::
                            0111
           41 A5    01  90  0111          MOVB    #DT$_CI780,UCB$B_DEVTYPE(R5)  ; Set the device type to CI780
                            0115
                            0115          CPUDISP <<780,CI_780>, -             ; * Dispatch on CPU type *
                            0115                   <750,CI_750>, -
                            0115                   <730,OTHER_CPU>, -
                            0115                   <790,CI_790>, -
                            0115                   <785,CI_785>>
                            0147
                            0147  CI_750:
                            0147
                41 A5   96  0147          INCB    UCB$B_DEVTYPE(R5)            ; Step device type to 750
        FF09 CF     35  90  014A          MOVB    #^A/57,INI$T_HWTYPE+2        ; Change CPU type to 'V750'
                    10  11  014F          BRB     CI_780                       ; Join common code
                            0151
                            0151  CI_785:
                            0151
        FF03 CF     35  90  0151          MOVB    #^A/5/,INI$T_HWTYPE+3        ; Change CPU type to 'V785'
                    09  11  0156          BRB     CI_780
                            0158
                            0158  CI_790:
                            0158
FEF5 CF   30303638 8F   D0  0158          MOVL    #^A/8600/,INI$T_HWTYPE       ; Change CPU type to '8600'
                            0161                                               ; Device type = CI780
                            0161
                            0161  CI_780:                                      ; * End of CPU dependent code *
```

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

```
           016E          .SBTTL  CONTROLLER INIT
           016E
           016E     ;+
           016E     ; The controller initialization entry as seen by the system, PA$CTLINIT,
           016E     ; is a noop since initialization can't begin without the unit 0 UCB.
           016E     ; Actual controller init is called from unit 0 unit initialization with
           016E     ; the same inputs as unit init.
           016E     ;
           016E     ; Inputs:
           016E     ;
           016E     ;        R3                          -Addr of PA configuration register
           016E     ;        R4                          -Same as R3
           016E     ;        R5                          -Addr of UCB for unit 0
           016E     ;
           016E     ; Outputs:
           016E     ;
           016E     ;        All registers               -Preserved
           016E     ;-
           016E
           016E     PA$CTLINIT::                                 ; Controller init called by system
           016E
     05    016E             RSB                                  ; Return
```

```
                          016F  ;
                          016F  ; Controller initialization called from unit 0 init.
                          016F  ;
                          016F
                          016F           .ENABL  LSB
                          016F  INIT_CTLR::
        0084 C5     D5    016F           TSTL    UCB$L_PDT(R5)              ; Built structures yet?
               32   13    0173           BEQL    BUILD_STRUCT               ; Branch if not
               10   AA    0175           BICW    #UCB$M_ONLINE,-            ; Set unit offline to show
            64 A5         0177                   UCB$W_STS(R5)              ;  port init being done
   54   0084 C5     D0    0179           MOVL    UCB$L_PDT(R5),R4           ; Get PDT addr
                          017E           $PRTCTINI -                        ; Protect from non-ex port
                          017E                   B^1$,#MCHK$M_NEXM          ;
               01   D0    018A           MOVL    #PA_PMC_M_MIN,-            ;  and maint init port
          00E8 D4         018C                   @PDT$L_PMC(R4)             ;
                          018F           $PRTCTEND 1$                       ; End of mcheck protection
            05 50   E9    0190           BLBC    R0,5$                      ; If mcheck, dont mark port
                          0193                                              ;  powered up
               02   A8    0193           BISW    #PDT$M_PUP,-               ; Set power up on this
          0110 C4         0195                   PDT$W_[PORT_STS(R4)        ;  port
                          0198
               00   E2    0198  5$:      BBSS    #PDT$V_PWF_CLNUP,-         ; Branch if SYSAP notification
       08 0110 C4         019A                   PDT$W_[PORT_STS(R4),10$ ;  underway; else set pwf recov
                          019E                                              ;  in progress,
   51   0364 8F     3C    019E           MOVZWL  #SS$_POWERFAIL,R1          ;  set aux status to give SYSAP's,
            FE5A'   30    01A3           BSBW    ERR$PWF_RECOV              ;  call recovery startup
                          01A6
                    05    01A6  10$:     RSB                                ; Return
                          01A7
                          01A7  BUILD_STRUCT:                               ; Complete unit init at fork IPL
            64  64  D0    01A7           MOVL    PA_CNF(R4),PA_CNF(R4)      ; Clear all miscellaneous errors
         04 A4  01  D0    01AA           MOVL    #PA_PMC_M_MIN,PA_PMC(R4); Place port in un-initialized state
 53   000001B8'EF   9E    01AE           MOVAB   CHECK_SYSTEMID,R3          ; Address of where to resume at fork IPL
             0545   31    01B5           BRW     INI$FORK                   ; Fork...
                          01B8
                          01B8           .DSABL  LSB
```

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

```
                           01CE         .SBTTL  BUILD PDT
                           01CE
                           01CE  ;
                           01CE  ; PDT adjacency assumptions:
                           01CE  ;
                           01CE
                           01CE  ASSUME  PDT$L_FLINK        EQ 0
                           01CE  ASSUME  PDT$B_PDT_TYPE     EQ 7
                           01CE  ASSUME  PDT$W_SIZE         EQ 8
                           01CE  ASSUME  PDT$W_SIZE+2       EQ PDT$B_TYPE
                           01CE  ASSUME  PDT$B_TYPE+1       EQ PDT$B_SUBTYP
                           01CE  ASSUME  PDT$B_SUBTYP+1     EQ PDT$C_SCSBASE
                           01CE
                           01CE         .ENABL  LSB
                           01CE
                           01CE  BUILD_PDT:
                           01CE
      51    0560 8F    3C  01CE         MOVZWL  #<PDT$C_PALENGTH+512>,R1   ; Enough for a PDT + 1 pg
       00000000'GF     16  01D3         JSB     G^EXE$ALONONPAGED          ; Allocate non-paged pool for PDT
              06 50    E8  01D9         BLBS    R0,10$                     ; Branch if success
                           01DC         ASSUME  PAER$K_ES_POOL EQ 0        ; Else, log a pool allocation
                 50    D4  01DC         CLRL    R0                         ; error.
               FE1F'   30  01DE         BSBW    ELOG$INIT_SWERR
                       05  01E1         RSB                                ; Return with unit offline
                           01E2
         50    52      D0  01E2  10$:   MOVL    R2,R0                      ; Save address
      52    03E0 C2    DE  01E5         MOVAL   PDT$C_PQB+512(R2),R2       ; Round PQB offset up to
      52    01FF 8F    AA  01EA         BICW    #511,R2                    ;  next page boundary
 52   000001E0 8F      C2  01EF         SUBL    #PDT$C_PQB,R2              ;  and compute corresponding PDT base
    0084 C5    52      D0  01F6         MOVL    R2,UCB$L_PDT(R5)           ; Save PDT addr
      53    24 A5      D0  01FB         MOVL    UCB$L_CRB(R5),R3           ; Get CRB addr
      10 A3    52      D0  01FF         MOVL    R2,CRB$L_AUXSTRUC(R3)      ;  and save PDT addr in CRB
 08 A0    52   50      A3  0203         SUBW3   R0,R2,PDT$W_SIZE(R0)       ; Fix up size of unused memory
 0A A0    0060 8F      B0  0208         MOVW    #DYN$C_SCS,PDT$B_TYPE(R0)  ;  and type
                 82    D4  020E         CLRL    (R2)+                      ; Init PDT, unused longwd,
 82   01000000 8F      D0  0210         MOVL    #PDT$C_PA@24,(R2)+         ;  unused 3 bytes and port type
 82   51    08 A0      A3  0217         SUBW3   PDT$W_SIZE(R0),R1,(R2)+    ;  PDT size,
   82   0560 8F        B0  021C         MOVW    #<DYN$C_SCS_PDT@8 + DYN$C_SCS>,(R2)+   ;  structure subtype and type
      53    30 A3      D0  0221         MOVL    CRB$L_INTD+VEC$L_INITIAL(R3),R3  ; Get addr of controller
                           0225                                                  ;   init routine
      51    FDD7 CF    3E  0225         MOVAW   PA$SCSOFFSET,R1            ; Get addr of table of offsets
                           022A                                            ;  to SCS entries in PADRIVER
                           022A
         50    81      32  022A  20$:   CVTWL   (R1)+,R0                   ; Get offset to next SCS routine
               06      13  022D         BEQL    30$                        ; Branch if no more
      82    53   50    C1  022F         ADDL3   R0,R3,(R2)+                ; Add offset from controller init
                           0233                                            ;  to addr of controller init
                           0233                                            ;  and store in PDT
               F5      11  0233         BRB     20$                        ; Get next offset
                           0235
         7E    54      7D  0235  30$:   MOVQ    R4,-(SP)                   ; Save R4, R5
 00      00 8F   00    2C  0238         MOVC5   #0,#0,#0,-                 ; Zero PDT from here to
         025C 8F           023D                 #<PDT$L_DQELOGOUT - PDT$C_SCSEND>,-
               62          0240                 (R2)                       ;  to logout area
         54    8E      7D  0241         MOVQ    (SP)+,R4                   ; Restore  R4, R5
      52    0084 C5    D0  0244         MOVL    UCB$L_PDT(R5),R2           ; Get base of PDT again
            00AC C2    DE  0249         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_P0_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          ADDL3   G^SGN$GL_MAXGPGCT,-       ;  and GPT length which is
                0228 C2       0304                  PDT$L_SPTLEN(R2),-        ;  the sum of the global page
                0230 C2       0307                  PDT$L_GPTLEN(R2)          ;  count and spte count  (global
                              030A                                            ;  page table base=spt base.)
      51   00000000'GF    DE  030A          MOVAL   G^SCS$GL_PDT,R1           ; Get base of SCS port list
                              0311
           50      61     D0  0311  60$:    MOVL    (R1),R0                   ; Get next port
                   05     13  0314          BEQL    70$                       ; Branch if none
           51      50     D0  0316          MOVL    R0,R1                     ; Else save next PDT as previous
                   F6     11  0319          BRB     60$                       ; Continue down the list
                              031B
           61      52     D0  031B  70$:    MOVL    R2,(R1)                   ; Hook this PDT to end of list
                              031E
                              031E          .DSABL  LSB
                              031E           .SBTTL  BUILD TEMPLATE LOOPBACK DG
                              031E
                              031E    ;+
                              031E    ; Allocate and initialize the template loopback datagram except for local
                              031E    ; port number and CRC.  These are recalculated each time power is recovered.
                              031E    ;
                              031E    ; Note that the template loopback datagram need not have a network header,
                              031E    ; nor have PPD$W_SIZE be a negative offset.
                              031E    ;-
                              031E
                              031E           .ENABL  LSB
                              031E
                              031E    BUILD_TLB:
                              031E
51    0046 8F     3C          031E           MOVZWL  #PPD$C_LB_LENGTH,R1      ; Get total template size
 00000000'GF      16          0323           JSB     G^EXE$ALONONPAGED        ; Allocate non-paged pool for template
         08 50    E8          0329           BLBS    R0,10$                   ; Branch if got it
                              032C           ASSUME  PAER$K_ES_POOL EQ 0      ; Else, log a pool allocation
            50    D4          032C           CLRL    R0                       ; error.
         FCCF'    30          032E           BSBW    ELOG$INIT_SWERR
          0320    31          0331           BRW     CLEANUP_PDT              ; Go clean up allocated buffers
                              0334
    08 A2   51    B0          0334    10$:   MOVW    R1,PPD$W_SIZE(R2)        ; Save structure size and
            3B    B0          0338           MOVW    #DYN$C_CIDG,-            ;  type
       0A A2                  033A                   PPD$B_TYPE(R2)           ;
       0D A2      94          033C           CLRB    PPD$B_STATUS(R2)         ; Init template status = 0,
            0D    9B          033F           MOVZBW  #PPD$C_SNDLB,-           ;  opcode = SNDLB,
       0E A2                  0341                   PPD$B_OPC(R2)            ;
            30    B0          0343           MOVW    #PPD$C_LBDAT_LEN,-       ;  LB length to # of
       10 A2                  0345                   PPD$W_LENGTH(R2)         ;  bytes of data
            51    D4          0347           CLRL    R1                       ; Generate LB data pattern
                              0349
 12 A241    51    90          0349    20$:   MOVB    R1,PPD$B_LBDATA(R2)[R1]  ;  of bytes = 0,1,2,...
   F7 51    30    F3          034E           AOBLEQ  #PPD$C_LBDAT_LEN,R1,20$  ;  ... LBDAT_LEN-1
50    0084 C5     D0          0352           MOVL    UCB$L_PDT(R5),R0         ; Hook template to
0184 C0     52    D0          0357           MOVL    R2,PDT$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           .SBTTL  INIT POLLER TIMER AND TRACE FUNCTION
                         0381
                         0381     ;+
                         0381     ; Initialize timer to wake driver up and insert CRB on timer queue.
                         0381     ;-
                         0381
                         0381           .ENABL  LSB
                         0381
                         0381     INIT_CRB:
                         0381
      53    24 A5    D0  0381           MOVL    UCB$L_CRB(R5),R3        ; Get CRB addr
              FC78'  30  0385           BSBW    CNF$CALCINTDUE          ; Set to wake up basic timer
                         0388                                           ;  interval from now and
     00000000'GF     16  0388           JSB     G^IOC$THREADCRB         ; put CRB on timer queue
                         038E
                         038E           .IF     DF PA$DEBUG             ; Conditional init of debug facility
                         038E           BSBW    TRC$INIT                ; Init trace buffer
                         038E           .ENDC                           ;
                         038E
                         038E           .DSABL  LSB
```

```
                           038E           .SBTTL  INI$PORT, INITIALIZE PORT
                           038E
                           038E      ;+
                           038E      ; Load the port microcode, init port hardware, complete initialization
                           038E      ; of the template loopback datagram (in case port number changed while
                           038E      ; powered down.)  Allocate and queue free datagrams and messages to
                           038E      ; port for future receives.  If all this is successful, set unit 0 online,
                           038E      ; clear power fail in progress and set port powered up.
                           038E      ;
                           038E      ; Inputs:
                           038E      ;
                           038E      ;        R4                          -Addr of port configuration register
                           038E      ;        R5                          -Addr of UCB of unit 0
                           038E      ;
                           038E      ;        IPL                         -IPL$_SCS
                           038E      ;
                           038E      ; It is assumed that device interrupts are disabled, that there are no
                           038E      ; outstanding interrupts, and that the port is in the un-initialized state.
                           038E      ;-
                           038E
                           038E      ASSUME   PDT$W_STDGUSED  EQ  PDT$W_STDGDYN+2
                           038E      ASSUME   PDT$Q_COMQH     EQ  PDT$Q_COMQBASE+8
                           038E      ASSUME   PDT$Q_COMQ2     EQ  PDT$Q_COMQH+8
                           038E      ASSUME   PDT$Q_COMQ3     EQ  PDT$Q_COMQ2+8
                           038E      ASSUME   PDT$Q_RSPQ      EQ  PDT$Q_COMQ3+8
                           038E
                           038E      ASSUME   PPD$C_LBDAT_LEN+7  LE   255
                           038E
                           038E              .ENABL  LSB
                           038E
                           038E      INI$PORT::
                           038E
           64   64    D0   038E              MOVL    PA_CNF(R4),PA_CNF(R4)     ; Clear any misc errors we can
         1000 8F      B3   0391              BITW    #PA_CNF_M_NOCI,-          ; CI750 port inaccessible?
           64              0395                      PA_CNF(R4)                ;
           03         13   0396              BEQL    10$                       ; Branch if accessible
         0283         31   0398              BRW     PORT_NOTPRES              ; Else go handle error
                           039B
      04 A4   01      D0   039B      10$:    MOVL    #PA_PMC_M_MIN,PA_PMC(R4); Place port in un-initialized state
         7E   54      7D   039F              MOVQ    R4,-(SP)                  ; Save registers destroyed by subr
         022B         30   03A2              BSBW    TEST_SHUTDOWN             ; Check if we are shutting down and
                           03A5                                                ;  if so, take operator action which
                           03A5                                                ;  may possibly include a bugcheck
                           03A5                                                ;  if we can't go on without the port
           54   8E    7D   03A5              MOVQ    (SP)+,R4                  ; Restore registers
      52   0084 C5    D0   03A8              MOVL    UCB$L_PDT(R5),R2          ; Get PDT address
                01    12   03AD              BNEQ    15$                       ; Branch if this port is still in business
                      05   03AF              RSB                               ; Else return to caller without
                           03B0                                                ;  reinitializing it
                           03B0
   00000652'EF     01   90   03B0    15$:    MOVB    #1,INI$CPU_REV            ; Assume CPU rev will be okay
30353756 8F   FC9B CF   D1   03B7            CMPL    INI$T_HWTYPE,#^A/V750/    ; Running 11/750?
                   11   12   03C0            BNEQ    CPU_REV_OK                ; Branch if not
           51      3E   DB   03C2            MFPR    #PR$_SID,R1               ; Read SID (copy of SID in memory
                             03C5                                              ;  is not good enough because the
                             03C5                                              ;  ucode rev level may have been
                             03C5                                              ;  increased by the loading of patches
                             03C5                                              ;  by the PCS software load mechanism
                             03C5                                              ;  or decreased by failure to load patches
                             03C5                                              ;  following pwr fail recovery.)
     50   51    F8 8F   78   03C5              ASHL     #-8,R1,R0              ; Get ucode rev level in low byte
          61 8F    50   91   03CA              CMPB     R0,#MIN_750_REV        ; Is CPU ucode sufficient to run port
                   03   18   03CE              BGEQ     CPU_REV_OK             ; Branch if so
                 0256   31   03D0              BRW      CPU_REV_ERROR          ; Go handle if not sufficient
                             03D3
                             03D3   CPU_REV_OK:
                             03D3
     53    00000000'GF  D0   03D3              MOVL     G^SCS$GL_MCADR,R3      ; Get base of microcode image
                   09   12   03DA              BNEQ     CHECK_QUEUES           ; Branch if got microcode image in pool
              50   01   9A   03DC              MOVZBL   #PAER$K_ES_CODE, R0    ; Else, log error indicating that
                FC1E'   30   03DF              BSBW     ELOG$INIT_SWERR        ; microcode could not be found in pool.
                             03E2
                 026F   31   03E2   20$:       BRW      CLEANUP_PDT            ; Go clean up allocated pool
                             03E5
                             03E5   CHECK_QUEUES:
                             03E5
              20   00   3B   03E5              SKPC     #0,#<PDT$Q_RSPQ - PDT$Q_COMQBASE>,-
              01E0 C2        03E8                       PDT$Q_COMQBASE(R2)     ; Check for any nonzero port
                             03EB                                              ;  command queues or response queue
                   17   13   03EB              BEQL     LOAD_UCODE             ; Branch if all zero (empty queues)
                             03ED              $DEBUGCHECK #ERR$V_DEB_NEPQ     ; Else do optional bugcheck
                             0400                                              ;  since on reinit all queues should
                             0400                                              ;  be empty
                   61   94   0400              CLRB     (R1)                   ; For recovery, try zeroing byte
                   E1   11   0402              BRB      CHECK_QUEUES           ;  and check again
                             0404
                             0404   LOAD_UCODE:
                             0404
          52   0400 8F  3C   0404              MOVZWL   #^X400,R2              ; Set initial MC addr assuming  ROM port
  5F6A00A1 8F    08 A3  D1   0409              CMPL     8(R3),#^X5F6A00A1      ; Is this code from an all RAM
                             0411                                              ;  port file?
                   02   12   0411              BNEQ     30$                    ; Branch if not -- assume ROM
                   52   D4   0413              CLRL     R2                     ; Else set to load both PROM
                             0415
              7E   52   7D   0415   30$:       MOVQ     R2,-(SP)               ; Save initial addresses
                             0418
           14 A4   52   D0   0418   40$:       MOVL     R2,PA_MADR(R4)         ; Give CS addr to CI
           18 A4   83   D0   041C              MOVL     (R3)+,PA_MDATR(R4)     ; Write 4 bytes of ucode
 14 A4   52  00001000 8F  C9  0420             BISL3    #^X1000,R2,-           ; Step CS addr of h.o.
                             0429                       PA_MADR(R4)            ;  word of data
           18 A4   83   3C   0429              MOVZWL   (R3)+,PA_MDATR(R4)     ; Write h.o. 2 bytes
    E3 52  00000C00 8F  F2   042D              AOBLSS   #^XC00,R2,40$          ; Loop till entire image loaded
              52   8E   7D   0435              MOVQ     (SP)+,R2               ; Retreive initial addresses for
                             0438                                              ;  read back of ucode
                   52   D5   0438              TSTL     R2                     ; Is this a possible all RAM port?
                   2A   13   043A              BEQL     START_UCODE            ; Branch if so to skip check of
                             043C                                              ;  possible ROM code that might
                             043C                                              ;  not agree with our ucode image.
                             043C
                             043C   CHECK_UCODE:
                             043C
           14 A4   52   D0   043C              MOVL     R2,PA_MADR(R4)         ; Set next control store addr
              50   83   D0   0440              MOVL     (R3)+,R0               ; Get next 4 bytes wcs should have
     50    18 A4        D1   0443              CMPL     PA_MDATR(R4),R0        ; WCS ok?
```

```
PA
Sy

$$
$$
$$
$$
$$
$$
$$
$$
$$
$$
$$
$$
BA
BE
BU
BU
BU
BU
BU
BU
CH
CH
CH
CI
CI
CI
CI
CL
CL
CN
CN
CO
CP
CP
CR
CR
CR
CR
CT
CX
DD
DT
DT
DY
DY
DY
EL
EL
EL
EL
ER
ER
ER
ER
ER
ER
EX
```

```
                      12  12  0447         BNEQ     BAD_UCODE                     ; Branch if not
 14 A4   52  00001000 8F  C9  0449         BISL3    #^XT000,R2,PA_MADR(R4)        ; Set to read h.o. 2 bytes of uword
                  50  83  3C  0452         MOVZWL   (R3)+,R0                      ; Get next 2 bytes WCS should have
               50   18 A4  B1  0455        CMPW     PA_MDATR(R4),R0               ; Next 2 bytes ok?
                      03  13  0459         BEQL     60$                           ; Branch if so
                              045B
                              045B BAD_UCODE:
                              045B
                    01D7  31  045B         BRW      WCS_ERROR                     ; Go handle error
                              045E
      D6 52  00000C00 8F  F2  045E 60$:    AOBLSS   #^XC00,R2,CHECK_UCODE         ; Branch if more to check
                              0466
                              0466
                              0466 START_UCODE:
                              0466
             00000040 8F  C8  0466         BISL     #PA_PMC_M_PSA,-               ; Set programmable start addr
                   04 A4      046C                  PA_PMC(R4)                    ;  bit
             00000400 8F  D0  046E         MOVL     #PA_C_UCODEST,-               ; Set microcode start addr
                   14 A4      0474                  PA_MADR(R4)                   ;
                      01  D0  0476         MOVL     #PA_PIC_M_PIC,-               ; Set port initialize -- move
                 0924 C4      0478                  PA_PIC(R4)                    ;  port state from uninit to disabled
                              047B         TIMEWAIT #<10000>,#PA_PMC_M_MIF,-      ; Wait for port init done
                              047B                   PA_PMC(R4),L                 ;  or 100 msec
                   07 50  E9  04A3         BLBC     R0,70$                        ; Branch if failed
                 0900 C4  D1  04A6         CMPL     PA_PS(R4),-                   ; Check that port init is done
                      08      04AA                  #PA_PS_M_PIC                  ;  and no errors set
                      03  13  04AB         BEQL     90$                           ; Branch if sucess
                    018B  31  04AD 70$:    BRW      INIT_PORT_FAIL                ; Else go to failure
                              04B0
            50   0084 C5  D0  04B0 90$:    MOVL     UCB$L_PDT(R5),R0              ; Retreive PDT addr
                 14   09  EF  04B5         EXTZV    #9,#20,-                      ; Extract virtual page #
            50   0218 C0      04B8                  PDT$L_VPQB(R0),R0             ;  of PQB
       51   00000000'GF   D0  04BC         MOVL     G^MMG$GL_SPTBASE,R1           ; Get base of SPT
               50   6140  D0  04C3         MOVL     (R1)[R0],R0                   ; Get PTE for PQB addr
       50   50   14   00  EF  04C7         EXTZV    #0,#20,R0,R0                  ; Get PFN of PQB addr
     0904 C4   50     09  78  04CC         ASHL     #9,R0,PA_PQBBR(R4)            ; Convert to phys addr and
                              04D2                                                ;  store in the PQB base reg
        04 00000000'GF    E8  04D2         BLBS     G^SCS$GB_PASANITY,95$         ; Branch if sanity timer wanted
                      02  C8  04D9         BISL     #PA_PMC_M_MTD,-               ; Else disable it
                   04 A4      04DB                  PA_PMC(R4)                    ;
                              04DD
                      01  D0  04DD 95$:    MOVL     #PA_PSR_M_PSC,-               ; Release the port
                 0918 C4      04DF                  PA_PSR(R4)                    ;  status register to port
                      01  D0  04E2         MOVL     #PA_PEC_M_PEC,-               ; Finally,....
                 091C C4      04E4                  PA_PEC(R4)                    ;  enable the port
                              04E7
                              04E7 INIT_LBDG_CRC:
                              04E7
            54   0084 C5  D0  04E7         MOVL     UCB$L_PDT(R5),R4              ; Get PDT address
            53   0184 C4  D0  04EC         MOVL     PDT$L_LBDG(R4),R3             ; Get addr of LB template
                 010C D4  90  04F1         MOVB     @PDT$L_PPR(R4),-              ; Save local port number
                   0C A3      04F5                  PPD$B_PORT(R3)                ;  in LB dg template
            51   0043 8F  3C  04F7         MOVZWL   #<PPD$C_LCB_DATA + PPD$C_LBDAT_LEN>,R1
             00000000'GF  16  04FC         JSB      G^EXE$ALONONPAGED             ; Allocate temporary buffer for
                              0502                                                ;  setting up data to calc CRC
                   03 50  E8  0502         BLBS     R0,100$                       ; Branch if got it
                    013E  31  0505 97$:    BRW      INIT_LBDG_FAIL                ; Else go handle error
```

```
                                0508
 08 A2   51   00600000 8F    C9 0508 100$:   BISL3   #DYN$C_SCS@16,R1,-           ; Set structure type
                                0511                 PPD$W_SIZE(R2)               ;  and size in temporary buffer
                                0511                                              ; Note that the size is not CRCed
                                0511                                              ;  so it can be the straight size
                                0511                                              ;  here, not a negative offset to
                                0511                                              ;  a net header.
                                0511                                              ; Set up data to CRC:
                     0C A2   94 0511         CLRB    PPD$W_LCB_LEN7(R2)           ;  H.o. lb data length + 7
                07   10 A3   81 0514         ADDB3   PPD$W_LENGTH(R3),#7,-        ;  L.o. lb data length + 7
                     0D A2      0518                 PPD$W_LCB_LEN7+1(R2)
                50   0C A3   90 051A         MOVB    PPD$B_PORT(R3),R0
                0E A2   50   90 051E         MOVB    R0,PPD$B_LCB_PORT(R2)        ;  own port number,
                0F A2   50   90 0522         MCOMB   R0,PPD$B_LCB_NPORT(R2)       ;  NOT(own port),
                10 A2   50   90 0526         MOVB    R0,PPD$B_LCB_LPORT(R2)       ;  local port,
                        0D   90 052A         MOVB    #PPD$C_SNDLB,-               ;  SNDLB opcode,
                     11 A2      052C                 PPD$B_LCB_OPC(R2)
                     12 A2   94 052E         CLRB    PPD$B_LCB_0(R2)              ;  and 0 (packing fmt = 0)
                        3C   BB 0531         PUSHR   #^M<R2,R3,R4,R5>             ; Save registers
                        30   28 0533         MOVC    #PPD$C_LBDAT_LEN,-           ; Copy LB data from
                     12 A3      0535                 PPD$B_LBDATA(R3),-           ;  template to temporary
                     13 A2      0537                 PPD$C_LCB_DATA(R2)           ;  buffer
                  52    6E   7D 0539         MOVQ    (SP),R2                      ; Get R2 and R3 again
               50    0D A2   9A 053C         MOVZBL  PPD$W_LCB_LEN7+1(R2),R0      ; Get # bytes to crc
 FFFFFFFF 8F   FB8D CF       0B 0540         CRC     CRC_TABLE,#-1,-              ; Compute CRC from temporary
               0C A2    50      0549                 R0,PPD$W_LCB_LEN7(R2)        ;  buffer
                        3C   BA 054C         POPR    #^M<R2,R3,R4,R5>             ; Retreive registers
               42 A3    50   D2 054E         MCOML   R0,PPD$L_LBCRC(R3)           ; Put CRC complement into template
                     50    52   D0 0552      MOVL    R2,R0                        ; Copy temporary buffer addr
            00000000'GF      16 0555         JSB     G^COM$DRVDEALMEM             ;  and return it to pool
                                055B
                                055B INIT_DFREEQ:
                                055B
           54   0084 C5      D0 055B         MOVL    UCB$L_PDT(R5),R4             ; Get PDT address
      50   00000000'GF       3C 0560         MOVZWL  G^SCS$GW_PAPPDDG,R0          ; Get # dg's for start handshakes
          0198 C4   50       D0 0567         MOVL    R0,PDT$W_STDGDYN(R4)         ; Set dynamic count of # dgs for
                                056C                                              ;  IDREC's and 0 the number currently
                                056C                                              ;  spoken for by known ports
               50   50       C0 056C         ADDL    R0,R0                        ; Double to handle error log dgs
                    FA8E'    30 056F         BSBW    SCS$ALL_FRDGS                ; Allocate and put on free queue
                  90 50      E9 0572         BLBC    R0,97$                       ; Branch if insufficient memory
               50   03       3C 0575         MOVZWL  #PA_C_MCACHESZ,R0            ; Get # msgs to fill cache
                    FA85'    30 0578         BSBW    SCS$ALL_FRMSGS               ; Allocate and put on free queue
                  87 50      E9 057B         BLBC    R0,97$                       ; Branch if insufficient memory
          017C C4   0F       90 057E         MOVB    #15,PDT$B_MAX_PORT(R4)       ; Assume small capacity cluster
          010C D4            D5 0583         TSTL    @PDT$L_PPR(R4)               ; Is it 16 ports max?
                    06       18 0587         BGEQ    120$                         ; Branch if so
       017C C4   DF 8F       90 0589         MOVB    #223,PDT$B_MAX_PORT(R4)      ; Else up count to 224 ports
                                058F
          010C D4            90 058F 120$:   MOVB    @PDT$L_PPR(R4),-             ; Save local port number in
          017D C4               0593                 PDT$B_PORT_NUM(R4)           ;  PDT
                    01       90 0596         MOVB    #PPD$C_PSP0,-                ; Set poller to scan all ports
          017F C4               0598                 PDT$B_REQIDPS(R4)            ;  on path A
          53   24 A5         D0 059B         MOVL    UCB$L_CRB(R5),R3             ; Retreive CRB address
                    FA5E'    30 059F         BSBW    CNF$CALCINTDUE               ; Calculate due time for
                                05A2                                              ;  next driver wakeup
                    FA5B'    30 05A2         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_LPORT_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$W_LPORT_STS(R4)   ;  dg enabled on port
                04    C8  05CA  1097          BISL2   #PA_PMC_M_MIE,-       ; Enable interrupts
              00E8 D4         05CC  1098                  @PDT$L_PMC(R4)        ;
                      05  05CF  1099          RSB                           ; Return from port initialization
                          05D0  1100          .DSABL  LSB
```

```
                        05D0          .SBTTL  TEST_SHUTDOWN,  CHECK IF PORT SHOULD
                        05D0          .SBTTL  -                      BE LEFT OFFLINE
                        05D0
                        05D0  ;+
                        05D0  ; This routine is called each time a port is initialized.  If the port
                        05D0  ; is being initialized for the first time, or if it is already shut
                        05D0  ; down, then return is taken.
                        05D0  ;
                        05D0  ; If this is not the first port init, then it must be a reinit following
                        05D0  ; a serious port error.  If the system device is not on this CI port,
                        05D0  ; and the number of reinitialization tries have been exhausted, then
                        05D0  ; the message, '%PAxO, CI Port is going Offline.' is printed.
                        05D0  ; If there are retries left, then the message,
                        05D0  ; '%PAxO, CI Port is Reinitializing ( xxx Retries Left).  Check Error Log.'
                        05D0  ; is printed. Both messages are directed to _OPAO, rather than OPCOM since
                        05D0  ; OPCOM needs a functioning system device to run and the system device may be
                        05D0  ; accessible only through the failing port.
                        05D0  ;
                        05D0  ; In the case of initialization failures that result in branching back
                        05D0  ; to INI$PORT for another try (e. g., ucode readback compare error),
                        05D0  ; these failures count against the maximum error retry count, and a message
                        05D0  ; is printed for each failure.
                        05D0  ;
                        05D0  ; If the port should be taken offline due to exhausted retry count, but
                        05D0  ; the system device is on this CI or this port is needed to cluster,
                        05D0  ; then the port driver bugchecks (in routine CLEANUP_PDT.)
                        05D0  ;
                        05D0  ; Inputs:
                        05D0  ;
                        05D0  ;       R4                              -Addr of port config register
                        05D0  ;       R5                              -Addr of UCB
                        05D0  ;       IPL                             -IPL$_SCS
                        05D0  ;
                        05D0  ; Outputs:
                        05D0  ;
                        05D0  ;       RO-R5                           -Destroyed
                        05D0  ;       Other registers                 -Preserved
                        05D0  ;-
                        05D0
                        05D0          .ENABL  LSB
                        05D0
                        05D0  TEST_SHUTDOWN:
                        05D0
         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$B_ERTCNT(R5),-      ; Is this the first init of port?
         0081 C5        05DA                  UCB$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          MOVAL   INI$MSG_OFFL,R2         ; We can proceed
              10    11  05F3          BRB     40$                     ; Go print port offline message
```

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

```
                        061E            .SBTTL  PORT INITIALIZATION ERRORS
                        061E
                        061E    ;+
                        061E    ; Come to PORT_NOTPRES if the NOCI bit is set in the configuration
                        061E    ; register.  The condition can only ever happen on a CI750.  It indicates
                        061E    ; that the port in its external cabinet is uncabled or unpowered.
                        061E    ;-
                        061E
                        061E            .ENABL  LSB
                        061E
                        061E    PORT_NOTPRES:
                        061E
 50   8002 8F   32      061E            CVTWL   #<PAER$K_ES_HWER ! ^X8000>,R0   ; Log as unspecified hardware
           F9DA'  30    0623            BSBW    ELOG$HARDWARE                   ;  error
           002B   31    0626            BRW     CLEANUP_PDT                     ; Go clean up without bothering
                        0629                                                    ;  to do any retries.
                        0629
                        0629    ;+
                        0629    ; Come to CPU_REV_ERROR if the CPU revision level is not sufficient to
                        0629    ; support CI port activity.
                        0629    ;
                        0629    ; Inputs:
                        0629    ;
                        0629    ;       R1                      -Copy of current SID
                        0629    ;       R5                      -UCB address
                        0629    ;-
                        0629
                        0629    CPU_REV_ERROR:
                        0629
   00000652'EF    94    0629            CLRB    INI$CPU_REV                   ; Clear flag that says this
                        062F                                                  ;  is regular bugcheck reason --
                        062F                                                  ;  this is reason for UCODEREV bugchk
           F9CE'  30    062F            BSBW    ELOG$CPU_REV                  ; Log bad CPU rev
           001F   31    0632            BRW     CLEANUP_PDT                   ; Go clean up pool because we
                        0635                                                  ;  can't continue
                        0635
                        0635    ;+
                        0635    ; Come to WCS_ERROR if loaded microcode could not be read back
                        0635    ; correctly.
                        0635    ;
                        0635    ; Inputs:
                        0635    ;
                        0635    ;       PA_MADR(R4)             -micro addr that failed
                        0635    ;       PA_MDATR(R4)            -Bad WCS contents
                        0635    ;       R0                      -Copy of good data from pool
                        0635    ;-
                        0635
                        0635    WCS_ERROR:
                        0635
           F9C8'  30    0635            BSBW    ELOG$UCODE_NORD               ; Log microcode read-back error.
           00B2   31    0638            BRW     RETRY_INIT                    ; Go clean up allocated pool
                        063B
                        063B    ;+
                        063B    ; Port initialization (transition from uninitialized to disabled) did
                        063B    ; not complete correctly.
                        063B    ;-
                        063B
```

```
                           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  ;+
                          0652  ; Deallocate template loopback datagram (if any) and PDT (if any).
                          0652  ; Return to caller with device offline, and power fail/power up
                          0652  ; unchanged.
                          0652  ;
                          0652  ; Inputs:
                          0652  ;
                          0652  ;         R5                          -UCB 0 addr
                          0652  ;-
                          0652
                          0652  INI$CPU_REV::
                          0652
                      01  0652          .BYTE   1                      ; 1/0 means CPU rev is okay/not okay
                          0653
                          0653  INI$PORT_REV::
                          0653
                      01  0653          .BYTE   1                      ; 1/0 means port ucode rev is okay/not okay
                          0654                                         ; NOTE: use of this memory flag is not
                          0654                                         ;  quite right in the case with multiple
                          0654                                         ;  ports since there is a fork between
                          0654                                         ;  the point where the flag is set
                          0654                                         ;  in ERR$CRASHPORT and where it is
                          0654                                         ;  set and here.  This means that we
                          0654                                         ;  might take the UCODEREV bugcheck
                          0654                                         ;  with the context for the wrong
                          0654                                         ;  port in hand, not a very serious
                          0654                                         ;  mishap.
                          0654
                          0654  ASSUME  PDT$L_FLINK     EQ 0
                          0654
                          0654          .ENABL  LSB
                          0654
                          0654  CLEANUP_PDT:
                          0654
     52   0084 C5     D0  0654          MOVL    UCB$L_PDT(R5),R2       ; Get PDT addr
                58    13  0659          BEQL    MAYBE_SYS_DEAD         ; Branch if none allocated
 51   00000000'GF     DE  065B          MOVAL   G^SCS$GL_PDT,R1        ; Get base of SCS port list
                          0662
          50    61    D0  0662  10$:    MOVL    (R1),R0                ; Get next PDT
                0D    13  0665          BEQL    30$                    ; Branch if none, ours wasn't linked
          52    50    D1  0667          CMPL    R0,R2                  ; Is this PDT ours?
                05    13  066A          BEQL    20$                    ; Branch if it is
          51    50    D0  066C          MOVL    R0,R1                  ; Else, save next PDT as previous
                F1    11  066F          BRB     10$                    ; Continue searching down the list
                          0671
          61    62    D0  0671  20$:    MOVL    (R2),(R1)              ; Remove out PDT from the list
                          0674
     50   0184 C2     D0  0674  30$:    MOVL    PDT$L_LBDG(R2),R0      ; Get loopback dg addr
                06    13  0679          BEQL    40$                    ; Branch if none allocated
      00000000'GF     16  067B          JSB     G^COM$DRVDEALMEM       ; Else deallocate it
                          0681
     51   00E4 C2     D0  0681  40$:    MOVL    PDT$L_CNF(R2),R1       ; Get configuration register addr
                          0686          $PRTCTINI -                    ; Ignore non responding device registers
                          0686                  B^50$,MCHK$M_NEXM      ;  causing machine checks
                01    D0  0696          MOVL    #PA_PMC_M_MIN,-        ;  while MINing the port once
             04 A1        0698                  PA_PMC(R1)             ;  more just to be sure it's quiet
                          069A          $PRTCTEND 50$                  ;
```

```
  50  00000000'GF  D0  069B             MOVL    G^EXE$GL_SYSUCB,R0         ; Get system device UCB
          0084 C0  D1  06A2             CMPL    UCB$L_PDT(R0),-            ; Is it via our PDT?
          0084 C5      06A6                     UCB$L_PDT(R5)              ;
               08  13  06A9             BEQL    MAYBE_SYS_DEAD             ; Branch if so
      00000000'GF  D5  06AB             TSTL    G^CLU$GL_CLUB              ; No.  Are we clustering?
               25  13  06B1             BEQL    60$                        ; Branch if not because system can get
                       06B3                                                ;  by without port
                       06B3
                       06B3 MAYBE_SYS_DEAD:
                       06B3
      00000000'GF  D5  06B3             TSTL    G^SCS$GL_PDT               ; Any SCS speaking PDTs left?
               1D  12  06B9             BNEQ    60$                        ; Branch if so -- take a chance
                       06BB                                                ;  that the remaining port(s) will
                       06BB                                                ;  carry us
   54     0084 C5  D0  06BB             MOVL    UCB$L_PDT(R5),R4           ; Else set up R4 for BUGCHECK
            8F AF  95  06C0             TSTB    INI$CPU_REV                ; Is this a CPU ucode problem?
               0C  13  06C3             BEQL    UCODE_BUGCHK               ; Branch if so
            8B AF  95  06C5             TSTB    INI$PORT_REV               ; Is this a port ucode problem?
               07  13  06C8             BEQL    UCODE_BUGCHK               ; Branch if so
                       06CA             BUGCHECK CIPORT,FATAL              ; Else shut the system down with
                       06D1                                                ;  normal CIPORT bugcheck
                       06D1
                       06D1 UCODE_BUGCHK:
                       06D1
                       06D1             BUGCHECK UCODEREV,FATAL            ; Shut system down with microcode
                       06D8                                                ;  revision problem bugcheck
                       06D8
           50  52  D0  06D8 60$:        MOVL    R2,R0                      ; Copy PDT addr to R0
      00000000'GF  16  06DB             JSB     G^COM$DRVDEALMEM           ; Deallocate PDT
          0084 C5  D4  06E1             CLRL    UCB$L_PDT(R5)              ; Show PDT as gone
   50        24 A5  D0  06E5             MOVL    UCB$L_CRB(R5),R0           ; Get CRB addr
            10 A0  D4  06E9             CLRL    CRB$L_AUXSTRUC(R0)         ;  and show no PDT here either
                       06EC
               05      06EC 70$:        RSB                                ; Return
                       06ED
                       06ED
                       06ED ;+
                       06ED ; RETRY_INIT is branched to on WCS load error or port init
                       06ED ; error.  RETRY_INIT checks for retries left.  If none, it branches
                       06ED ; to CLEANUP_PDT.  If retries are left, it branches to INI$PORT
                       06ED ; after decrementing the retry count.
                       06ED ;
                       06ED ; Inputs:
                       06ED ;
                       06ED ;           R4                          -Cofiguration register addr
                       06ED ;           R5                          -UCB 0 addr
                       06ED ;-
                       06ED
                       06ED RETRY_INIT:
                       06ED
               01  D0  06ED             MOVL    #PA_PMC_M_MIN,-            ; Do maint init on port
            04 A4      06EF                     PA_PMC(R4)                 ;  to make sure it's quiet
          0080 C5  97  06F1             DECB    UCB$B_ERTCNT(R5)           ; Decr # retries left
               03  18  06F5             BGEQ    90$                        ; Branch if retries left
             FF5A  31  06F7             BRW     CLEANUP_PDT                ; Else cleanup PDT
                       06FA
             FC91  31  06FA 90$:        BRW     INI$PORT                   ; Else do another init
```

```
                         06FD  1378
                         06FD  1379          .DSABL  LSB
                         06FD          .SBTTL  INI$FORK
                         06FD
                         06FD  ;+
                         06FD  ; This routine oversees and participates in the creation of a fork process,
                         06FD  ; and the transfer of control at fork IPL to a user supplied address. This
                         06FD  ; routine will use the UCB's fork block dequeuing it from the appropriate fork
                         06FD  ; queue if necessary. The dequeuing of the fork block and creation of the fork
                         06FD  ; process is handled as an atomic event by disabling all interrupts before
                         06FD  ; testing whether the fork block needs to be dequeued, and then re-enabling
                         06FD  ; interrupts after creation of the fork process. After creation of the fork
                         06FD  ; process this routine returns control to the caller.
                         06FD  ;
                         06FD  ; When the fork process commences execution it will do so within this routine.
                         06FD  ; It immediately will transfer control to the user supplied address. The caller
                         06FD  ; of this routine has available R4 in order to pass information across the
                         06FD  ; creation of the fork process to the user routine which will be jumped to at
                         06FD  ; fork IPL.
                         06FD  ;
                         06FD  ; This routine participates in proper synchronization to the fork block by the
                         06FD  ; appropriate setting and clearing of the fork block interlock bit before the
                         06FD  ; fork process is creating, and within the context of the fork process.
                         06FD  ;
                         06FD  ; Inputs:
                         06FD  ;
                         06FD  ;       R3                      -Address to JMP to at fork IPL
                         06FD  ;       R5                      -Address of UCB
                         06FD  ;       IPL                     -Device IPL or higher
                         06FD  ;
                         06FD  ;
                         06FD  ; Outputs:
                         06FD  ;
                         06FD  ;       After creation of fork process but before returning to caller:
                         06FD  ;
                         06FD  ;       R3-R4                   -Destroyed
                         06FD  ;       Other registers         -Preserved
                         06FD  ;       IPL                     -Preserved
                         06FD  ;
                         06FD  ;       Before exit from fork process:
                         06FD  ;
                         06FD  ;       R0-R2                   -Unpredictable
                         06FD  ;       R3                      -User address jumped to at fork IPL
                         06FD  ;       R4                      -User supplied value
                         06FD  ;       R5                      -Address of UCB
                         06FD  ;       IPL                     -Fork IPL
                         06FD  ;-
                         06FD
                         06FD          .ENABL  LSB
                         06FD  INI$FORK::
                         06FD          DSBINT                          ; Disable all interrupts
          01       E3    0703          BBCS    #UCB_V_FKLOCK,-         ; Is the fork block in use?
       03 68 A5          0705                  UCB$W_DEVSTS(R5),10$    ; Branch if not and set in use bit
       55    65    0F    0708          REMQUE  (R5),R5                 ; Remove fork block from its queue
                         070B
 0000071D'EF       9F    070B  10$:    PUSHAB  20$                     ; Return address for fork proc creation
 00000721'EF       9F    0711          PUSHAB  30$                     ; Fork process PC
 00000000'GF       17    0717          JMP     G^EXE$FORK              ; Create the fork process
                         071D
```

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

```
$$$CURSIZ                = 000001C4                  EXE$FORK                   *********   X   01
$$$ENTRYNUM              = 0000001E                  EXE$GB_CPUDATA             *********   X   01
$$$NEWSIZ                = 000001D0                  EXE$GB_CPUTYPE             *********   X   01
$$$PREV                  = 0000010C                  EXE$GL_SYSUCB              *********   X   01
$$$REGNUM                = 0000000B                  EXE$GL_TENUSEC             *********   X   01
$$BASE                   = 00000001                  EXE$GL_UBDELAY             *********   X   01
$$DISPL                  = 0000000A                  EXE$MCHK_PRTCT             *********   X   01
$$GENSW                  = 00000001                  FPC$ACCEPT                 *********   X   01
$$HIGH                   = 00000009                  FPC$ALLOCDG                *********   X   01
$$LIMIT                  = 00000008                  FPC$ALLOCMSG               *********   X   01
$$LOW                    = 00000001                  FPC$CONNECT                *********   X   01
$$MNSW                   = 00000001                  FPC$DCONNECT               *********   X   01
$$MXSW                   = 00000001                  FPC$DEALLOCDG              *********   X   01
BAD_UCODE                  0000045B R       01       FPC$DEALLOMSG              *********   X   01
BELL                     = 00000007                  FPC$DEALRGMSG              *********   X   01
BUG$_CIPORT                *********   X    01       FPC$MAINTFCN               *********   X   01
BUG$_UCODEREV              *********   X    01       FPC$MAP                    *********   X   01
BUG$_UNSUPRTCPU            *********   X    01       FPC$MAPBYPASS              *********   X   01
BUILD_PDT                  000001CE R       01       FPC$MAPIRP                 *********   X   01
BUILD_STRUCT               000001A7 R       01       FPC$MAPIRPBYP              *********   X   01
BUILD_TLB                  0000031E R       01       FPC$MRESET                 *********   X   01
CHECK_QUEUES               000003E5 R       01       FPC$MSTART                 *********   X   01
CHECK_SYSTEMID             000001B8 R       01       FPC$QUEUEDG                *********   X   01
CHECK_UCODE                0000043C R       01       FPC$QUEUEMDGS              *********   X   01
CI_750                     00000147 R       01       FPC$RCHMSGBUF              *********   X   01
CI_780                     00000161 R       01       FPC$RCLMSGBUF              *********   X   01
CI_785                     00000151 R       01       FPC$READCOUNT              *********   X   01
CI_790                     00000158 R       01       FPC$REJECT                 *********   X   01
CLEANUP_PDT                00000654 R       01       FPC$REQDATA                *********   X   01
CLU$GL_CLUB                *********   X    01       FPC$RLSCOUNT               *********   X   01
CNF$CALC_INTDUE            *********   X    01       FPC$SENDDATA               *********   X   01
CNF$CALC_POLLSW            *********   X    01       FPC$SENDDG                 *********   X   01
COM$DRVDEALMEM             *********   X    01       FPC$SENDMSG                *********   X   01
CPU_REV_ERROR              00000629 R       01       FPC$SENDRGDG               *********   X   01
CPU_REV_OK                 000003D3 R       01       FPC$SNDCNTMSG              *********   X   01
CR                       = 0000000D                  FPC$STOP_VCS               *********   X   01
CRB$L_AUXSTRUC           = 00000010                  FPC$UNMAP                  *********   X   01
CRB$L_INTD               = 00000024                  INI$CPU_REV                00000652 RG     01
CRC_TABLE                  000000D1 R       01       INI$FORK                   000006FD RG     01
CTRLR_NAME               = 00000006                  INI$MSG_INIT               0000005A RG     01
CXB$C_HEADER             = 00000048                  INI$MSG_OFFL               000000AB RG     01
DDB$T_NAME               = 00000014                  INI$PORT                   0000038E RG     01
DT$_CI750                = 00000002                  INI$PORT_REV               00000653 RG     01
DT$_CI780                = 00000001                  INI$T_HWTYPE               00000056 RG     01
DYN$C_CIDG               = 0000003B                  INIT_CRB                   00000381 R      01
DYN$C_SCS                = 00000060                  INIT_CTLR                  0000016F RG     01
DYN$C_SCS_PDT            = 00000005                  INIT_DFREEQ                0000055B R      01
ELOG$CPU_REV               *********   X    01       INIT_LBDG_CRC              000004E7 R      01
ELOG$HARDWARE              *********   X    01       INIT_LBDG_FAIL             00000646 R      01
ELOG$INIT_SWERR            *********   X    01       INIT_PORT_FAIL             0000063B R      01
ELOG$UCODE_NORD            *********   X    01       IOC$BROADCAST              *********   X   01
ERR$BUGCHECK               *********   X    01       IOC$THREADCRB              *********   X   01
ERR$CNV_HEX_DEC            *********   X    01       IPL$_SCS                 = 00000008
ERR$DEBUGCHECK             *********   X    01       LF                       = 0000000A
ERR$PWF_RECOV              *********   X    01       LOAD_UCODE                 00000404 R      01
ERR$V_DEB_NEPQ             *********   X    01       MAYBE_SYS_DEAD             000006B3 R      01
EXE$ALONONPAGED            *********   X    01       MCHK$M_NEXM              = 00000004
```

```
MIN_750_REV                     = 00000061                        PDT$B_NXT_PORT                    0000017E
MMG$GL_GPTBASE                    *********  X      01            PDT$B_P0_LBSTS                    00000180
MMG$GL_SPTBASE                    *********  X      01            PDT$B_P1_LBSTS                    00000181
OPA$UCB0                          *********  X      01            PDT$B_PDT_TYPE                  = 00000007
OTHER_CPU                         0000016D R        01            PDT$B_PLOGMAP                     00000134
PA$CTLINIT                        0000016E RG       01            PDT$B_PORTMAP                     00000114
PA$REGOFFSET                      0000003E RG       01            PDT$B_PORT_NUM                    0000017D
PA$SCSOFFSET                      00000000 RG       01            PDT$B_REQIDPS                     0000017F
PA$UNITINIT                       00000111 RG       01            PDT$B_SUBTYP                    = 0000000B
PAER$K_ES_CODE                  = 00000001                        PDT$B_TYPE                      = 0000000A
PAER$K_ES_HWER                  = 00000002                        PDT$C_LENGTH                    = 000000E4
PAER$K_ES_INIT                  = 00000001                        PDT$C_PA                        = 00000001
PAER$K_ES_LST0                  = 00000003                        PDT$C_PALENGTH                  = 00000360
PAER$K_ES_LST1                  = 00000009                        PDT$C_PAREGBASE                   000000E4
PAER$K_ES_LST2                  = 00000007                        PDT$C_PAREGEND                    00000110
PAER$K_ES_LST3                  = 00000009                        PDT$C_PQB                       = 000001E0
PAER$K_ES_LST4                  = 0000000C                        PDT$C_SCSBASE                   = 0000000C
PAER$K_ES_POOL                  = 00000000                        PDT$C_SCSEND                    = 00000084
PAER$K_ES_SCSID                 = 00000002                        PDT$L_ACCEPT                    = 0000000C
PAER$K_ET_DALT                  = 00000003                        PDT$L_ALLOCDG                   = 00000010
PAER$K_ET_LMLT                  = 00000042                        PDT$L_ALLOCMSG                  = 00000014
PA_CNF                            00000000                        PDT$L_CNF                         000000E4
PA_CNF_M_NOCI                   = 00001000                        PDT$L_CONNECT                   = 00000018
PA_CQ0                            00000908                        PDT$L_CQ0                         000000F0
PA_CQ1                            0000090C                        PDT$L_CQ1                         000000F4
PA_CQ2                            00000910                        PDT$L_DCONNECT                  = 00000028
PA_CQ3                            00000914                        PDT$L_DEALLOCDG                 = 0000001C
PA_C_MCACHESZ                   = 00000003                        PDT$L_DEALLOMSG                 = 00000020
PA_C_UCODEST                    = 00000400                        PDT$L_DEALRGMSG                 = 00000024
PA_DFQ                            00000928                        PDT$L_DFQ                         000000FC
PA_MADR                           00000014                        PDT$L_DFQHDR                      00000208
PA_MDATR                          00000018                        PDT$L_DGHDRSZ                     00000190
PA_MFQ                            0000092C                        PDT$L_DGNETHD                     00000194
PA_MTC                            00000930                        PDT$L_DGOVRHD                   = 000000B8
PA_MTEC                           00000934                        PDT$L_DQELOGOUT                   000002E0
PA_PDC                            00000920                        PDT$L_FLINK                     = 00000000
PA_PEC                            0000091C                        PDT$L_GPTBASE                     0000022C
PA_PEC_M_PEC                    = 00000001                        PDT$L_GPTLEN                      00000230
PA_PESR                           0000093C                        PDT$L_LBDG                        00000184
PA_PFAR                           00000938                        PDT$L_MAINTFCN                  = 00000078
PA_PIC                            00000924                        PDT$L_MAP                       = 0000002C
PA_PIC_M_PIC                    = 00000001                        PDT$L_MAPBYPASS                 = 00000030
PA_PMC                            00000004                        PDT$L_MAPIRP                    = 00000034
PA_PMC_M_MIE                    = 00000004                        PDT$L_MAPIRPBYP                 = 00000038
PA_PMC_M_MIF                    = 00000008                        PDT$L_MAXBCNT                   = 000000BC
PA_PMC_M_MIN                    = 00000001                        PDT$L_MFQ                         00000100
PA_PMC_M_MTD                    = 00000002                        PDT$L_MFQHDR                      0000020C
PA_PMC_M_PSA                    = 00000040                        PDT$L_MQELOGOUT                   00000320
PA_PPR                            00000940                        PDT$L_MRESET                    = 00000070
PA_PQBBR                          00000904                        PDT$L_MSGHDRSZ                  = 000000B4
PA_PS                             00000900                        PDT$L_MSTART                    = 00000074
PA_PSR                            00000918                        PDT$L_MTC                         00000104
PA_PSR_M_PSC                    = 00000001                        PDT$L_PFAR                        00000108
PA_PS_M_PIC                     = 00000008                        PDT$L_PMC                         000000E8
PDT$B_DQIMAP                      00000154                        PDT$L_POLLERDUE                   0000018C
PDT$B_HSHUT_DG                    000001B0                        PDT$L_POOLDUE                     00000188
PDT$B_MAX_PORT                    0000017C                        PDT$L_PPR                         0000010C
```

```
PDT$L_PS           000000EC              PPD$B_LCB_PORT    0000000E
PDT$L_PSR          000000F8              PPD$B_OPC         0000000E
PDT$L_QUEUEDG    = 0000003C              PPD$B_PORT        0000000C
PDT$L_QUEUEMDGS  = 00000040              PPD$B_PROTOCOL    0000001A
PDT$L_RCHMSGBUF  = 00000044              PPD$B_RSTATE      00000025
PDT$L_RCLMSGBUF  = 00000048              PPD$B_RST_PORT    00000024
PDT$L_READCOUNT  = 00000068              PPD$B_STATUS      0000000D
PDT$L_REJECT     = 0000004C              PPD$B_SWFLAG      0000000B
PDT$L_REQDATA    = 00000050              PPD$B_SYSTEMID    00000014
PDT$L_RLSCOUNT   = 0000006C              PPD$B_TYPE        0000000A
PDT$L_SENDDATA   = 00000054              PPD$C_LBDAT_LEN = 00000030
PDT$L_SENDDG     = 00000058              PPD$C_LB_LENGTH   00000046
PDT$L_SENDMSG    = 0000005C              PPD$C_LCB_DATA    00000013
PDT$L_SENDRGDG   = 0000007C              PPD$C_LENGTH      00000012
PDT$L_SNDCNTMSG  = 00000060              PPD$C_MIN_DGSIZ   00000050
PDT$L_SPTBASE      00000224              PPD$C_PSP0      = 00000001
PDT$L_SPTLEN       00000228              PPD$C_SNDLB     = 0000000D
PDT$L_STOP_VCS   = 00000080              PPD$K_LB_LENGTH   00000046
PDT$L_UCB0       = 000000DC              PPD$K_LENGTH      00000012
PDT$L_UNMAP      = 00000064              PPD$L_BLINK       00000004
PDT$L_VBDT         0000021C              PPD$L_DG_DISC     00000028
PDT$L_VPQB         00000218              PPD$L_FLINK       00000000
PDT$L_WAITQBL    = 000000B0              PPD$L_IN_VCD      00000018
PDT$L_WAITQFL    = 000000AC              PPD$L_LBCRC       00000042
PDT$M_CUR_LBS    = 00000001              PPD$L_P0_ACK      00000010
PDT$M_LBDG       = 00000004              PPD$L_P0_NAK      00000014
PDT$M_PRV_LBS    = 00000002              PPD$L_P0_NRSP     00000018
PDT$M_PUP        = 00000002              PPD$L_P1_ACK      0000001C
PDT$M_PWF_CLNUP  = 00000001              PPD$L_P1_NAK      00000020
PDT$Q_COMQ2        000001F0              PPD$L_P1_NRSP     00000024
PDT$Q_COMQ3        000001F8              PPD$L_REC_BOFF    00000028
PDT$Q_COMQBASE     000001E0              PPD$L_REC_NAME    00000024
PDT$Q_COMQH        000001E8              PPD$L_RPORT_FCN   00000020
PDT$Q_COMQL        000001E0              PPD$L_RPORT_REV   0000001C
PDT$Q_DFREEQ       000001D0              PPD$L_RPORT_TYP   00000018
PDT$Q_FORMPB       00000174              PPD$L_SND_BOFF    00000020
PDT$Q_MFREEQ       000001D8              PPD$L_SND_NAME    0000001C
PDT$Q_RSPQ         00000200              PPD$L_ST_ADDR     00000018
PDT$Q_TEMP_RSPQ    0000019C              PPD$L_XCT_LEN     00000018
PDT$V_PWF_CLNUP  = 00000000              PPD$Q_CURTIME     00000048
PDT$W_BDTLEN       00000220              PPD$Q_NODENAME    00000040
PDT$W_DQELEN       00000210              PPD$Q_SWINCARN    00000028
PDT$W_LPORT_STS    00000110              PPD$Q_XCT_ID      00000010
PDT$W_MQELEN       00000214              PPD$T_HWTYPE      00000030
PDT$W_PBCOUNT      00000112              PPD$T_SWTYPE      00000020
PDT$W_SIZE       = 00000008              PPD$T_SWVERS      00000024
PDT$W_STDGDYN      00000198              PPD$W_LCB_LEN7    0000000C
PDT$W_STDGUSED     0000019A              PPD$W_LENGTH      00000010
PORT_NOTPRES       0000061E R      01    PPD$W_MASK        00000010
PPD$B_DEF_ST       0000001C              PPD$W_MAXDG       0000001C
PPD$B_FLAGS        0000000F              PPD$W_MAXMSG      0000001E
PPD$B_HWVERS       00000034              PPD$W_MTYPE       00000012
PPD$B_LBDATA       00000012              PPD$W_M_VAL       00000014
PPD$B_LCB_0        00000012              PPD$W_SIZE        00000008
PPD$B_LCB_LPORT    00000010              PR$_IPL         = 00000012
PPD$B_LCB_NPORT    0000000F              PR$_SBR         = 0000000C
PPD$B_LCB_OPC      00000011              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$_SLR                         = 0000000D
RETRY_INIT                        000006ED R       01
RETRY_OFFSET                    = 00000027
SCS$ALL_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
------------------                       --------------
_$255$DUA28:[DRIVER.OBJ]PALIB.MLB;1                   7
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                       17
_$255$DUA28:[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
```

0114 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

PAINIT
LIS

PAFPCALL
LIS

PAMONIT
LIS

PAINTR
LIS