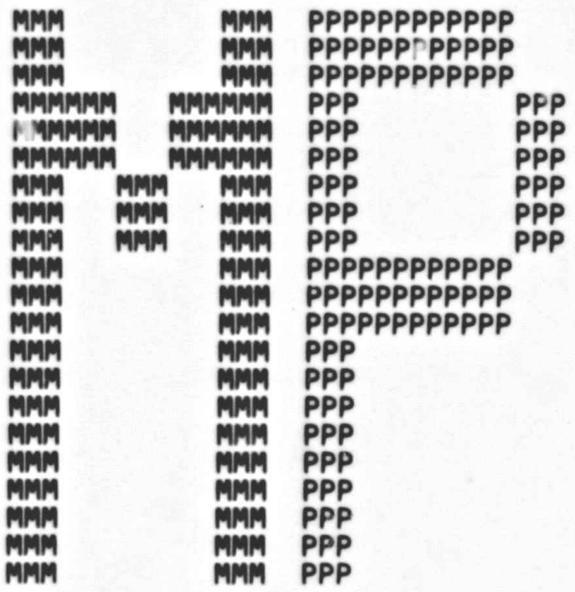
MMM MMM PPPPPPPPPPPP
MMM MMM PPPPPPPPPPPP
MMM MMM PPPPPPPPPPPP
MMMMMM MMMMMM PPP PPP
MMMMMM MMMMMM PPP PPP
MMMMMM MMMMMM PPP PPP
MMM MMM MMM PPP PPP
MMM MMM MMM PPP PPP
MMM MMM MMM PPP PPP
MMM MMM PPPPPPPPPPPP
MMM MMM PPPPPPPPPPPP
MMM MMM PPPPPPPPPPPP
MMM MMM PPP
MMM MMM PPP
MMM MMM PPP
MMM MMM PPP
MMM MMM PPP
MMM MMM PPP
MMM MMM PPP
MMM MMM PPP
MMM MMM PPP

FILEID**MPPWRFAIL

```
MM      MM  PPPPPPPP    PPPPPPPP    WW      WW  RRRRRRRR    FFFFFFFFFF    AAAAAA     IIIIII    LL
MM      MM  PPPPPPPP    PPPPPPPP    WW      WW  RRRRRRRR    FFFFFFFFFF    AAAAAA     IIIIII    LL
MMMM  MMMM  PP      PP  PP      PP  WW      WW  RR      RR  FF            AA    AA     II      LL
MMMM  MMMM  PP      PP  PP      PP  WW      WW  RR      RR  FF            AA    AA     II      LL
MM  MM  MM  PP      PP  PP      PP  WW      WW  RR      RR  FF            AA    AA     II      LL
MM  MM  MM  PP      PP  PP      PP  WW      WW  RR      RR  FF            AA    AA     II      LL
MM      MM  PPPPPPPP    PPPPPPPP    WW      WW  RRRRRRRR    FFFFFFFF      AA    AA     II      LL
MM      MM  PPPPPPPP    PPPPPPPP    WW      WW  RRRRRRRR    FFFFFFFF      AA    AA     II      LL
MM      MM  PP          PP          WW  WW  WW  RR  RR      FF            AAAAAAAAAA     II      LL
MM      MM  PP          PP          WW  WW  WW  RR  RR      FF            AAAAAAAAAA     II      LL
MM      MM  PP          PP          WWWW  WWWW  RR    RR    FF            AA    AA     II      LL              ....
MM      MM  PP          PP          WWWW  WWWW  RR    RR    FF            AA    AA     II      LL              ....
MM      MM  PP          PP          WW      WW  RR      RR  FF            AA    AA     IIIIII  LLLLLLLLLL      ....
MM      MM  PP          PP          WW      WW  RR      RR  FF            AA    AA     IIIIII  LLLLLLLLLL      ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

Table of contents

```
0000     1 ;
0000     2 ; Version:      'V04-000'
0000     3 ;
0000     4
0000     5         .MCALL  MFPR
0000     1         .TITLE  MPPWRFAIL - POWER FAIL INTERRUPT HANDLER
0000     2         .IDENT  'V04-000'
0000     3 ;
0000     4 ;******************************************************************************
0000     5 ;*                                                                            *
0000     6 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                   *
0000     7 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
0000     8 ;*  ALL RIGHTS RESERVED.                                                      *
0000     9 ;*                                                                            *
0000    10 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    11 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    12 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    13 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    14 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    15 ;*  TRANSFERRED.                                                              *
0000    16 ;*                                                                            *
0000    17 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    18 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    19 ;*  CORPORATION.                                                              *
0000    20 ;*                                                                            *
0000    21 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    22 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
0000    23 ;*                                                                            *
0000    24 ;*                                                                            *
0000    25 ;******************************************************************************
0000    26
0000    27 ;++
0000    28 ;
0000    29 ; Facility:  Executive , Hardware fault handling
0000    30 ;
0000    31 ; Abstract: POWERFAIL contains the code necessary to handle a power failure
0000    32 ;           interrupt on the secondary processor.  The secondary is set
0000    33 ;           to a state in which it restart with the normal initialization
0000    34 ;           code.
0000    35 ;
0000    36 ; Environment: MODE=Kernel , IPL=31
0000    37 ;
0000    38 ; Author:  KATHLEEN D. MORSE, Creation date:  08-JUN-1981
0000    39 ;
0000    40 ; Modified by:
0000    41 ;
0000    42 ;       V03-001 KDM0066         Kathleen D. Morse       3-Aug-1983
0000    43 ;               Change use of PR$_TODR to PR780$_TODR.
0000    44 ;
0000    45 ;--
0000    46
0000    47 ;
0000    48 ; Include files:
0000    49 ;
0000    50
0000    51 ;
0000    52 ; MACROS:
```

```
                        ;
                        ;
                        ; Equated Symbols:
                        ;
                                $IPLDEF                         ; Interrupt priority level definitions
                                $LCKDEF                         ; Interlock bit definitions
                                $MPSDEF                         ; Secondary processor states
                                $PCBDEF                         ; Process control block definitions
                                $PHDDEF                         ; Process header block definitons
                                $PR780DEF                       ; 780-specific processor register defs
                                $PSLDEF                         ; Processor status longword definitions

                                .PSECT  $AEXENONPAGED,LONG      ; Interrupt routines must be
                                                                ; longword aligned
```

```
0000     69          .SBTTL  EXE$POWERFAIL - POWER FAIL INTERRUPT SERVICE ROUTINE
0000     70 ;++
0000     71 ;
0000     72 ; Functional Description:
0000     73 ;
0000     74 ;       EXE$POWERFAIL is entered with IPL=31 as a result of a power fail
0000     75 ;       interrupt.  The objective is to fold up the current process and
0000     76 ;       get into a state from which the secondary can be re-initialized
0000     77 ;       as quickly as possible.
0000     78 ;
0000     79 ;       The secondary must be in the INIT state in order to re-initalize.
0000     80 ;       The primary processor must have exclusive rights over setting
0000     81 ;       the secondary into this state, to avoid various race conditions.
0000     82 ;       Thus, the scheduling code must check the powerfail flag and
0000     83 ;       force the primary into a state so that it can restart.
0000     84 ;
0000     85 ;       There are two possible ways in which the powerfail interrupt
0000     86 ;       can work.  Once the interrupt occurs, it can either be dismissed
0000     87 ;       or it can remain active (i.e., if an REI is issued, it will
0000     88 ;       re-occur).  This code will handle both cases.  However, it
0000     89 ;       appears that the 11/780 hardware is the former case.
0000     90 ;
0000     91 ;       When a powerfail occurs, the secondary may be in any of
0000     92 ;       its possible states:  INIT, STOP, BUSY, EXEC, IDLE, or DROP.
0000     93 ;       The following describes the flow of execution for the secondary
0000     94 ;       in each case.
0000     95 ;
0000     96 ;               INIT - Go to self-branch in powerfail code.
0000     97 ;                      In this case, the primary will not need to
0000     98 ;                      set the state of the secondary to INIT.
0000     99 ;
0000    100 ;               STOP - Go to self-branch in powerfail code.
0000    101 ;                      In this case, the primary must not set
0000    102 ;                      the state of the secondary to INIT as the
0000    103 ;                      secondary had been stopped with an explicit
0000    104 ;                      user command.
0000    105 ;
0000    106 ;                      (Restart will make secondary do busy-loop in
0000    107 ;                      EXE$MPSTART, waiting to be set to INIT state.)
0000    108 ;
0000    109 ;               BUSY - Set IPL of PSL on stack to 31 and REI.  This will
0000    110 ;                      return the secondary to the scheduling code which
0000    111 ;                      will make him do a LDPCTX and go into EXEC state.
0000    112 ;                      When the REI is done (following the LDPCTX), the
0000    113 ;                      software timer interrupt will occur and the
0000    114 ;                      secondary will fold up the process and execute
0000    115 ;                      the busy-loop in the scheduling code.
0000    116 ;
0000    117 ;               IDLE - Set IPL of PSL on stack to 31 and REI.  This will
0000    118 ;                      return the secondary to the scheduling code where
0000    119 ;                      it will loop until the restart occurs.  The primary
0000    120 ;                      will notice the powerfail flag when it next tries to
0000    121 ;                      schedule the secondary and set him to INIT state.
0000    122 ;
0000    123 ;                      (The secondary may or may not busy-loop in
0000    124 ;                      EXE$MPSTART when the restart occurs, depending
0000    125 ;                      upon when the primary sets him to the INIT state.)
```

```
                     0000   126 ;
                     0000   127 ;                    DROP - Set IPL of PSL on stack to 31 and REI.  This will
                     0000   128 ;                           cause the primary to take back the process, and the
                     0000   129 ;                           secondary to return to the scheduling code, to loop
                     0000   130 ;                           until the restart occurs.  If the primary doesn't see
                     0000   131 ;                           the powerfail flag, then the secondary will go BUSY
                     0000   132 ;                           and then EXEC, and then get the software timer
                     0000   133 ;                           interrupt after the REI (following the LDPCTX).  The
                     0000   134 ;                           secondary will fold up the process and loop in the
                     0000   135 ;                           Scheduling code until restart occurs.
                     0000   136 ;
                     0000   137 ;                    EXEC - If current mode of PSL on stack is Kernel, then
                     0000   138 ;                           set IPL of PSL on stack to 31 and REI.  This will
                     0000   139 ;                           return the secondary to the particular interrupt
                     0000   140 ;                           code that he had been executing (MA780, scheduling,
                     0000   141 ;                           hardware clock, etc.).  When that particular interrupt
                     0000   142 ;                           code finishes, the software timer interrupt will occur.
                     0000   143 ;                           Now the secondary will fold up the process, and
                     0000   144 ;                           wait in the busy-loop in the scheduling code for
                     0000   145 ;                           the restart.
                     0000   146 ;
                     0000   147 ;                           If current mode of PSL on stack is NOT Kernel, then
                     0000   148 ;                           the secondary folds up the process and interrupts the
                     0000   149 ;                           primary to take back the process.  The secondary then
                     0000   150 ;                           executes the self-branch in the powerfail code.
                     0000   151 ;
                     0000   152 ; Environment:
                     0000   153 ;
                     0000   154 ;       Executed by secondary processor.
                     0000   155 ;
                     0000   156 ; Calling Sequence:
                     0000   157 ;
                     0000   158 ;       Powerfail interrupt through Vector at offset 12 in the SCB.
                     0000   159 ;
                     0000   160 ; Input Parameters:
                     0000   161 ;
                     0000   162 ;       00(SP) - PC at time of powerfail interrupt
                     0000   163 ;       04(SP) - PSL at time of powerfail interrupt
                     0000   164 ;
                     0000   165 ; Implicit Inputs:
                     0000   166 ;
                     0000   167 ;       MPS$GL_STATE - state of secondary processor
                     0000   168 ;
                     0000   169 ;--
                     0000   170
                     0000   171          .ALIGN  LONG                    ; Exception and Interrupt routines must
                     0000   172                                          ; be longword aligned
                     0000   173 MPS$POWERFAIL::                          ;
                     0000   174          .LIST ME
                     0000   175          MFPR    #PR780$_TODR,W^MPS$GL_PFAILTIM ; Indicate powerfail occurred
                     0000                .IF IDN <#PR780$_TODR>,<#PR780$_TODR>
                     0000                PUSHR   #^M<R0,R1,R2,R3>
                     0002                30000$: MOVQ    G^EXE$GQ_SYSTIME,R0
                     0009                MOVQ    G^EXE$GQ_SYSTIME,R2
                     0010                CMPL    R0,R2
                     0013                BNEQ    30000$
                     0015                CMPL    R1,R3
```

```
                     E8   12  0018                        BNEQ    30000$
          52  00000000'GF   7D  001A                        MOVQ    G^EXE$GQ_TODCBASE,R2
                50   52    C2  0021                        SUBL    R2,R0
                51   53    D9  0024                        SBWC    R3,R1
    51    50    50  00030D40 8F  7B  0027                  EDIV    #<100*1000*2>,R0,R0,R1
                51   51    01  78  0030                    ASHL    #1,R1,R1
          51  00000000'GF   C0  0034                        ADDL    G^EXE$GL_TODR,R1
                1B   51    DA  003B                        MTPR    R1,#PR780$_TODR
                     0F    BA  003E                        POPR    #^M<R0,R1,R2,R3>
                           0040                            .ENDC
                     00000001  0040                        .MDELETE MFPR
              0000'CF   1B    DB  0040                    MFPR    #PR780$_TODR,W^MPS$GL_PFAILTIM
                           0045                            .MCALL  MFPR
                           0045
                           0045   176          .NLIST ME
                           0045   177          SOFTINT #IPL$_TIMER                 ; Force rescheduling to occur after
                           0048   178                                              ; the powerfail logic occurs
                           0048   179          ASSUME  MPS$K_INITSTATE GT MPS$K_EXECSTATE
                           0048   180          ASSUME  MPS$K_STOPSTATE GT MPS$K_EXECSTATE
       00 0000'CF   00    E6  0048   181          BBSSI   #LCK$V_INTERLOCK,W^MPS$GL_INTERLOCK,10$ ; Flush cache queue
          04   0000'CF   D1  004E   182  10$:    CMPL    W^MPS$GL_STATE,#MPS$K_EXECSTATE ; Has a LDPCTX been done?
                     11   14  0053   183          BGTR    40$                         ; Br if STOP or INIT state
                     11   12  0055   184          BNEQ    60$                         ; Br if no LDPCTX has been done
            07 AE    03   93  0057   185          BITB    #<PSL$M_CURMOD@-24>,7(SP) ; Was kernel mode code interrupted?
                     0B   13  005B   186          BEQL    60$                         ; Br if yes, can't fold up process now
                     07   FD  005D   187          SVPCTX                              ; Fold up the processs
              0000'CF   02   D0  005E   188          MOVL    #MPS$K_DROPSTATE,W^MPS$GL_STATE ; Prepare to reschedule
                   FF9A'   30  0063   189          BSBW    W^MPS$INTPRIM               ; Request primary take the process back
                     FE    11  0066   190  40$:    BRB     40$                         ; Wait for power off halt
                           0068   191                                              ; This loop is to avoid halting
                           0068   192                                              ; and confusing the console
                           0068   193                                              ; by inadvertently triggering an
                           0068   194                                              ; automatic restart too soon.
                           0068   195  60$:    SETIPL  #IPL$_POWER                 ; Prevent reserved operand fault on REI
        06 AE    1F   88  006B   196          BISB    #<PSL$M_IPL@-16>,6(SP)      ; Force saved IPL to 31
                     02   FB  006F   197          REI
                           0070   198
                           0070   199          .END                                ;
```

```
EXE$GL_TODR                       ********   X   02
EXE$GQ_SYSTIME                    ********   X   02
EXE$GQ_TODCBASE                   ********   X   02
IPL$_POWER                      = 0000001F
IPL$_TIMER                      = 00000008
LCK$V_INTERLOCK                 = 00000000
MPS$GL_INTERLOCK                  ********   X   02
MPS$GL_PFAILTIM                   ********   X   02
MPS$GL_STATE                      ********   X   02
MPS$INTPRIM                       ********   X   02
MPS$K_DROPSTATE                 = 00000002
MPS$K_EXECSTATE                 = 00000004
MPS$K_INITSTATE                 = 00000005
MPS$K_STOPSTATE                 = 00000006
MPS$POWERFAIL                     00000000 RG    02
PR$_IPL                           ********   X   02
PR$_SIRR                          ********   X   02
PR780$_TODR                     = 0000001B
PSL$M_CURMOD                    = 03000000
PSL$M_IPL                       = 001F0000
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
$AEXENONPAGED	00000070 (112.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	39	00:00:00.06	00:00:00.51
Command processing	151	00:00:00.88	00:00:06.52
Pass 1	198	00:00:04.23	00:00:14.24
Symbol table sort	0	00:00:00.49	00:00:00.56
Pass 2	58	00:00:00.97	00:00:03.94
Symbol table output	4	00:00:00.03	00:00:00.04
Psect synopsis output	1	00:00:00.03	00:00:00.04
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	453	00:00:06.70	00:00:25.87

```
The working set limit was 1350 pages.
21123 bytes (42 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 350 non-local and 4 local symbols.
204 source lines were read in Pass 1, producing 13 object records in Pass 2.
18 pages of virtual memory were used to define 17 macros.
```

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

Macro library name	Macros defined
_255DUA28:[MP.OBJ]MP.MLB;1	4
_255DUA28:[SYS.OBJ]LIB.MLB;1	5
_255DUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	15

495 GETS were required to define 15 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:MPPWRFAIL/OBJ=OBJ$:MPPWRFAIL MSRC$:MPPREFIX/UPDATE=(ENH$:MPPREFIX)+MSRC$:MPPWRFAIL/UPDATE=(ENH$:MPPWRFAIL)+EXECML$/LI
```

0248 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

MPERRLOG
LIS

MPSCBVEC
LIS

MPINT
LIS

MPPFM
LIS

MPDAT
LIS

MPPWRFAIL
LIS

MPMCHECK
LIS

MPINTEXC
LIS

MPLOG
LIS

MPERRMSG
LIS

MPSCHED
LIS

MPSHWPFM
LIS

MPLOAD
LIS

MPINIT
LIS