```
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**MPSCHED

```
MM      MM  PPPPPPPP      SSSSSSSS    CCCCCCCC  HH      HH  EEEEEEEEEE  DDDDDDDD
MM      MM  PPPPPPPP      SSSSSSSS    CCCCCCCC  HH      HH  EEEEEEEEEE  DDDDDDDD
MMMM  MMMM  PP      PP  SS          CC          HH      HH  EE          DD      DD
MMMM  MMMM  PP      PP  SS          CC          HH      HH  EE          DD      DD
MM  MM  MM  PP      PP  SS          CC          HH      HH  EE          DD      DD
MM  MM  MM  PP      PP  SS          CC          HH      HH  EE          DD      DD
MM      MM  PPPPPPPP      SSSSSS    CC          HHHHHHHHHH  EEEEEEEE    DD      DD
MM      MM  PPPPPPPP      SSSSSS    CC          HHHHHHHHHH  EEEEEEEE    DD      DD
MM      MM  PP                  SS  CC          HH      HH  EE          DD      DD
MM      MM  PP                  SS  CC          HH      HH  EE          DD      DD
MM      MM  PP                  SS  CC          HH      HH  EE          DD      DD      ....
MM      MM  PP                  SS  CC          HH      HH  EE          DD      DD      ....
MM      MM  PP          SSSSSSSS      CCCCCCCC  HH      HH  EEEEEEEEEE  DDDDDDDD        ....
MM      MM  PP          SSSSSSSS      CCCCCCCC  HH      HH  EEEEEEEEEE  DDDDDDDD        ....

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

(1)	136	MPS$RESCHEDIPL5 - MP RESCHEDULING (MA780) INTERRUPT HANDLER
(1)	291	MPS$RESCHED - RESCHEDULING INTERRUPT HANDLER
(1)	390	MPS$MPSCHED - SECONDARY PROCESSOR SCHEDULING

```
0000      ;
0000      ; Version:      'V04-000'
0000      ;

0000            .MCALL  MFPR
0000            .TITLE  MPSCHED - MULTIPROCESSOR SCHEDULER
0000            .IDENT  'V04-000'

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:  Executive , Hardware fault handling
0000      ;
0000      ; Abstract:  Multi-processing scheduler routines
0000      ;
0000      ; Environment: MODE=Kernel
0000      ;
0000      ; Author:  RICHARD I. HUSTVEDT, Creation date:  15-MAY-1979
0000      ;
0000      ; Modified by:
0000      ;
0000      ;       V03-007 KDM0032         Kathleen D. Morse       19-Nov-1982
0000      ;               Remove performance measurement for wait time between
0000      ;               reschedules on secondary.
0000      ;
0000      ;       V03-006 KDM0018         Kathleen D. Morse       19-Nov-1982
0000      ;               Implement wait for event flag system services on
0000      ;               secondary.
0000      ;
0000      ;       V03-005 KDM0014         Kathleen D. Morse       27-Sep-1982
0000      ;               Fix performance data collection of scheduling usage
0000      ;               of exec mode AST.
0000      ;
```

```
0000    ;       V03-004 KDM0008         Kathleen D. Morse       31-Aug-1982
0000    ;               Prevent race between STOP/CPU and rescheduling interrupts.
0000    ;               A SVPCTX could be done twice for the current process on
0000    ;               the secondary.
0000    ;
0000    ;       V03-003 KDM0007         Kathleen D. Morse       31-Aug-1982
0000    ;               Remove SCH$IDLE from MPS$RESCHED logic flow.  Dropping
0000    ;               IPL from 7 to 3 allowed AST queuing and if an IPL 5
0000    ;               reschedule interrupt occurred, then the same PCB address
0000    ;               ended up in MPS$GL_CURPCB and SCH$GL_CURPCB, causing problems.
0000    ;
0000    ; 01     -
0000    ;--
0000
0000    ;
0000    ; INCLUDE FILES:
0000    ;
0000
0000    ;
0000    ; MACROS:
0000    ;
0000
0000    ;
0000    ; EQUATED SYMBOLS:
0000    ;
0000
0000            $DYNDEF                         ; Structure type code definitions
0000            $IPLDEF                         ; Interrupt priority level definitions
0000            $LCKDEF                         ; Interlock bit definitions
0000            $MPSDEF                         ; Secondary processor state definitions
0000            $PCBDEF                         ; PCB definitions
0000            $PHDDEF                         ; PHD definitions
0000            $PRDEF                          ; Processor register definitions
0000            $PSLDEF                         ; Program status longword definitions
0000            $STATEDEF                       ; State definitions
0000
0000    ;
0000    ; OWN STORAGE:
0000    ;
0000
0000    ;++
0000    ;
0000    ; IPL Usages:
0000    ;
0000    ;       Primary Processor               Secondary Processor
0000    ;       -----------------               -------------------
0000    ;
0000    ;        0 - Unused                      0 - Unused
0000    ;        1 - Unused                      1 - Unused
0000    ;        2 - AST delivery                2 - AST delivery
0000    ;        3 - Rescheduling                3 - Rescheduling
0000    ;        4 - IO posting                  4 - Unused
0000    ;        5 - Multi-processor interrupt   5 - Xdelta
0000    ;        6 - Fork                        6 - Unused
0000    ;        7 - Software timer interrupt    7 - Quantum end software interrupt
0000    ;        8 - Fork                        8 - Unused
0000    ;        9 - Fork                        9 - Unused
```

```
     0000   ;        10 - Fork                      10 - Unused
     0000   ;        11 - Fork                      11 - Unused
     0000   ;        12 - Unused                    12 - Unused
     0000   ;        13 - Unused                    13 - Unused
     0000   ;        14 - Unused                    14 - Unused
     0000   ;        15 - Xdelta                    15 - Unused
     0000   ;        16 - Unused                    16 - Unused
     0000   ;        17 - Unused                    17 - Unused
     0000   ;        18 - Unused                    18 - Unused
     0000   ;        19 - Unused                    19 - Unused
     0000   ;        20 - Device interrupt          20 - Unused
     0000   ;        21 - Device interrupt          21 - Unused
     0000   ;        22 - Device interrupt          22 - Unused
     0000   ;        23 - Device interrupt          23 - Unused
     0000   ;        24 - Timer interrupt           24 - Timer interrupt
     0000   ;        25 - Unused                    25 - Unused
     0000   ;        26 - Unused                    26 - Unused
     0000   ;        27 - Unused                    27 - Unused
     0000   ;        28 - Unused                    28 - Unused
     0000   ;        29 - Unused                    29 - Unused
     0000   ;        30 - Unused                    30 - Unused
     0000   ;        31 - Unused                    31 - Unused
     0000   ;
     0000   ;--
 00000000           .PSECT  AEXENONPAGED,LONG      ; Nonpaged exec
```

```
                    0000          .SBTTL  MPS$RESCHEDIPL5 - MP RESCHEDULING (MA780) INTERRUPT HANDLER
                    0000  ;++
                    0000  ; MPS$RESCHEDIPL5 - MULTI-PROCESSOR RESCHEDULING (MA780) INTERRUPT HANDLER
                    0000  ;
                    0000  ; This routine is entered via the IPL 5 rescheduling interrupt.
                    0000  ; The vector for this interrupt is coded to cause execution
                    0000  ; on the interrupt stack.
                    0000  ;
                    0000  ; ENVIRONMENT:
                    0000  ;
                    0000  ;       IPL=5 MODE=kernel IS=1
                    0000  ;
                    0000  ;       Executed by the primary processor.
                    0000  ;
                    0000  ; INPUT:
                    0000  ;
                    0000  ;       00(SP)=PC at reschedule interrupt
                    0000  ;       04(SP)=PSL at interrupt
                    0000  ;
                    0000  ;--
                    0000          .ALIGN  LONG                    ; Interrupt handlers must be
                    0000                                          ;  longword aligned
                    0000  MPS$RESCHEDIPL5::                       ; Interrupt handler for
                    0000                                          ;  multi-processor reschedule interrupt
        3F    BB    0000          PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ; Save needed registers
                    0002
                    0002          .IF     DF,MPPFMSWT
                    0002          BSBW    MPS$PFM_RESCHD          ; Gather performance measurement data
                    0002          .ENDC
                    0002
        18    10    0002          BSBB    MPS$SCHSCND             ; Schedule secondary
                    0004
                    0004          .IF     DF,MPPFMSWT
                    0004          BSBW    MPS$PFM_SCHDSUC         ; Gather performance measurement data
                    0004          .ENDC
                    0004
        3F    BA    0004          POPR    #^M<R0,R1,R2,R3,R4,R5>  ; Restore registers
        02          0006          REI                             ; Return from interrupt
                    0007
                    0007          .ENABL  LSB
                    0007  PFAIL_EXIT:                             ; Exit when secondary is in powerfail
  0000'CF   05  D0  0007          MOVL    #MPS$K_INITSTATE,W^MPS$GL_STATE ; Set state for restart
            0D  11  000C          BRB     1$                      ; Done with rescheduling
                    000E  MPS$EXIT:                               ;
  03  0000'CF   D1  000E          CMPL    W^MPS$GL_STATE,#MPS$K_BUSYSTATE ; Check for powerfail window
            06  12  0013          BNEQ    1$                      ; where secondary was set busy after it
      0000'CF   D5  0015          TSTL    W^MPS$GL_PFAILTIM       ; saw the powerfail occur. In this case,
            13  12  0019          BNEQ    7$                      ; go drop the process owned by secondary
            05      001B  1$:     RSB                             ; All done rescheduling secondary
                    001C  MPS$SCHSCND::                           ; Schedule secondary processor
                    001C          SETIPL  #IPL$_SYNCH             ; Synch scheduler with event reporting
00 0000'CF  00  E6  001F  4$:     BBSSI   #LCK$V_INTERLOCK,W^MPS$GL_INTERLOCK,5$ ; Flush cache queue
                    0025          ASSUME MPS$K_IDLESTATE LT MPS$K_DROPSTATE
                    0025          ASSUME MPS$K_BUSYSTATE GT MPS$K_DROPSTATE
                    0025          ASSUME MPS$K_INITSTATE GT MPS$K_BUSYSTATE
                    0025          ASSUME MPS$K_STOPSTATE GT MPS$K_BUSYSTATE
  02  0000'CF   D1  0025  5$:     CMPL    W^MPS$GL_STATE,#MPS$K_DROPSTATE ; Is state already idle?
```

```
                41   19  002A        BLSS    MPS$SCHEDIPL5           ; Yes, just schedule a new one
                E0   14  002C        BGTR    MPS$EXIT                ; Br if busy
                         002E   ;
                         002E   ; The primary must check to see if the process was put into the DROP state
                         002E   ; because it needed the primary to check for an event flag wait condition
                         002E   ; or because it is ready for rescheduling.
                         002E   ;
     05 0000'CF    02   E7  002E   7$:     BBCCI   #MPS$V_SECWAITCK,W^MPS$GL_SECREQFLG,8$ ; Br if normal resched
            FFC9'      30  0034           BSBW    W^MPS$WAITCK            ; Go perform event flag wait check for
                E6   11  0037           BRB     4$                      ;  secondary processor and recheck for
                         0039   8$:                                      ;  DROP in case process was returned
                         0039                                            ;  to the secondary processor
                         0039   ;
                         0039   ; The primary processor takes the process that the secondary can
                         0039   ; no longer execute and puts it back on the appropriate scheduler queue.
                         0039   ;
        51   0000'CF    D0  0039           MOVL    W^MPS$GL_CURPCB,R1      ; Get address of current PCB
        52      0B A1   9A  003E           MOVZBL  PCB$B_PRI(R1),R2        ; Current priority
  00 00000000'GF    52   E2  0042           BBSS    R2,G^SCH$GL_COMQS,10$   ; Mark queue non-empty
           2C A1    0C   B0  004A   10$:    MOVW    #SCH$C_COM,PCB$W_STATE(R1) ; Set state to resident computable
 53   00000000'GF42   7E  004E           MOVAQ   G^SCH$AQ_COMT[R2],R3    ; Compute address of queue
            93    61   0E  0056           INSQUE  (R1),@(R3)+             ; Insert at tail of queue
     00 0000'CF    00   E6  0059           BBSSI   #LCK$V_INTERLOCK,W^MPS$GL_INTERLOCK,15$ ; Flush cache queue
           0000'CF    D5  005F   15$:     TSTL    W^MPS$GL_PFAILTIM       ; Is secondary in power fail?
                A2   12  0063           BNEQ    PFAIL_EXIT              ; Br if yes, power fail in progress
        0000'CF    01   D0  0065           MOVL    #MPS$K_IDLESTATE,W^MPS$GL_STATE ; Set secondary's state to idle
                         006A           SOFTINT #IPL$_SCHED             ; Reschedule primary processor
                         006D           .DSABL  LSB
                         006D   ;+
                         006D   ; MPS$SCHEDIPL5 - SCHEDULE NEW PROCESS FOR EXECUTION
                         006D   ;
                         006D   ; In this routine, the primary processor selects the highest priority
                         006D   ; executable process and places it in the hands of the secondary
                         006D   ; processor.  It then tells the secondary processor to exit its idle
                         006D   ; loop and start executing, by altering MPS$GL_STATE.
                         006D   ;
                         006D   ; ENVIRONMENT:
                         006D   ;
                         006D   ;       Executed by the primary processor.
                         006D   ;
                         006D   ;-
                         006D   MPS$SCHEDIPL5::                          ; Schedule for execution
           0000'CF    D5  006D           TSTL    W^MPS$GL_PFAILTIM       ; Has secondary taken powerfail?
                94   12  0071           BNEQ    PFAIL_EXIT              ; Br on yes, must set state to
                         0073                                            ; INIT so that he can restart.
52   00000000'GF    20   00  EA  0073           FFS     #0,#32,G^SCH$GL_COMQS,R2 ; Find first full state
                5C   13  007C           BEQL    MPS$IDLE                ; No executable process?
     53   00000000'GF42   7E  007E   NEXTQ:  MOVAQ   G^SCH$AQ_COMH[R2],R3    ; Compute queue head address
                54   53  D0  0086           MOVL    R3,R4                   ; Get head of queue
                54   64  D0  0089   QLOOP:  MOVL    (R4),R4                 ; Get next in queue
                54   53  D1  008C           CMPL    R3,R4                   ; Check for end of queue
                36   13  008F           BEQL    QEMPT                   ; Br if yes
           55      6C A4  D0  0091           MOVL    PCB$L_PHD(R4),R5        ; Get process header address
50   00C4 C5    02   18  EF  0095           EXTZV   #PSL$V_CURMOD,#PSL$S_CURMOD,PHD$L_PSL(R5),R0 ;Get current mode
                EB   13  009C           BEQL    QLOOP                   ; Br if yes, can't run on secondary
        00CF C5    50   91  009E           CMPB    R0,PHD$B_ASTLVL(R5)     ; Is there an AST pending for process?
```

```
                    E4  18  00A3   250          BGEQ    QLOOP                          ; Br on yes, don't sched for secondary
             0104   C5  D5  00A5   251          TSTL    PHD$L_MPINHIBIT(R5)            ; Any reason not to run on secondary?
                    DE  12  00A9   252          BNEQ    QLOOP                          ; Br on yes, don't sched for secondary
               54   64  0F  00AB   253          REMQUE  (R4),R4                        ; Remove from queue
                    08  12  00AE   254          BNEQ    20$                            ; Queue not empty
     00 00000000'GF 52  E5  00B0   255          BBCC    R2,G^SCH$GL_COMQS,20$          ; Set queue empty
                        00B8   256 20$:
            2C A4   0E  B0  00B8   257          MOVW    #SCH$C_CUR,PCB$W_STATE(R4) ; Set state to current
        0000'CF     54  D0  00BC   258          MOVL    R4,W^MPS$GL_CURPCB             ; Note current PCB loc
                        00C1   259
                        00C1   260          .IF     DF,MPPFMSWT
                        00C1   261          BSBW    W^MPS$PFM_EXCHG                ; Collect perf meas data
                        00C1   262          .ENDC
                        00C1   263
        0000'CF     03  D0  00C1   264          MOVL    #MPS$K_BUSYSTATE,W^MPS$GL_STATE ; Set state to busy
                        00C6   265
                        00C6   266          .IF     DF,MPPFMSWT
                        00C6   267          BSBW    MPS$PFM_CTXSW                  ; Gather performance measurement data
                        00C6   268          .ENDC
                        00C6   269
                    05  00C6   270          RSB                                    ; Normal return
                        00C7   271
                    52  D6  00C7   272 QEMPT:  INCL    R2                             ; Next queue
            51      20  52  C3  00C9   273          SUBL3   R2,#32,R1                      ; Compute number of remaining queues
                    0B  15  00CD   274          BLEQ    MPS$IDLE                       ; Br if none left
52  00000000'GF     51  52  EA  00CF   275          FFS     R2,R1,G^SCH$GL_COMQS,R2 ; Find next active state
                    A4  12  00D8   276          BNEQ    NEXTQ                          ; Continue if another non-empty state
                        00DA   277 MPS$IDLE:                                      ; No active, executable process
     0000'CF    00000000'GF 9E  00DA   278          MOVAB   G^SCH$GL_NULLPCB,W^MPS$GL_CURPCB ; Set to null PCB
                    05  00E3   279          RSB                                    ; And return
                        00E4   280
                        00E4   281
                        00E4   282 QEMPTY:
                        00E4   283          BUG_CHECK       QUEUEMPTY,FATAL
                        00E8   284
                        00E8   285 SCH$IDLE:
                        00E8   286          SETIPL  #IPL$_SCHED
00000000'GF         20  90  00EB   287          MOVB    #32,G^SCH$GB_PRI
            0029        31  00F2   288          BRW     MPS$SCHED                      ; And try again
                        00F5   289
```

```
                                00F5          .SBTTL  MPS$RESCHED - RESCHEDULING INTERRUPT HANDLER
                                00F5  ;++
                                00F5  ; MPS$RESCHED - RESCHEDULING INTERRUPT HANDLER
                                00F5  ;
                                00F5  ; This routine is entered via the IPL 3 rescheduling interrupt.
                                00F5  ; the vector for this interrupt is coded to cause execution on
                                00F5  ; the kernel stack.
                                00F5  ;
                                00F5  ; This routine replaces SCH$RESCHED in a vanilla VMS system.
                                00F5  ;
                                00F5  ; ENVIRONMENT:
                                00F5  ;
                                00F5  ;       IPL=3 MODE=kernel IS=0
                                00F5  ;
                                00F5  ;       Executed by the primary processor.
                                00F5  ;
                                00F5  ; INPUT:
                                00F5  ;
                                00F5  ;       00(SP)=PC at reschedule interrupt
                                00F5  ;       04(SP)=PSL at interrupt
                                00F5  ;
                                00F5  ;--
                                00F5          .ALIGN  LONG                    ; Align interrupt handler
                                00F8  MPS$RESCHED::                           ; Reschedule interrupt handler
                                00F8          SETIPL  #IPL$_SYNCH             ; Synch scheduler with event reporting
                          07    00FB          SVPCTX                          ; Save context of process
     51   00000000'GF     D0    00FC          MOVL    G^SCH$GL_CURPCB,R1      ; Get address of current PCB
             52    0B A1  9A    0103          MOVZBL  PCB$B_PRI(R1),R2        ; Current priority
  00 00000000'GF     52   E2    0107          BBSS    R2,G^SCH$GL_COMQS,10$   ; Mark queue non-empty
             2C A1     0C  B0    010F  10$:    MOVW    #SCH$C_COM,PCB$W_STATE(R1) ; Set state to resident computable
     53   00000000'GF42    7E    0113          MOVAQ   G^SCH$AQ_COMT[R2],R3    ; Compute address of queue
             93    61     0E    011B          INSQUE  (R1),@(R3)+             ; Insert at tail of queue
                                011E
                                011E  ;+
                                011E  ; MPS$SCHED - SCHEDULE NEW PROCESS FOR EXECUTION
                                011E  ;
                                011E  ; This routine selects the highest priority executable process
                                011E  ; and places it in execution.
                                011E  ;
                                011E  ; This routine replaces SCH$SCHED in a vanilla VMS system.
                                011E  ;
                                011E  ; ENVIRONMENT:
                                011E  ;
                                011E  ;       Executed by the primary processor.
                                011E  ;
                                011E  ;-
                                011E
                                011E  MPS$SCHED::                             ; Schedule for execution
                                011E          SETIPL  #IPL$_SYNCH             ; Synch scheduler with event reporting
                      FEF8      30    0121          BSBW    MPS$SCHSCND             ; Schedule secondary
52   00000000'GF    20    00     EA    0124          FFS     #0,#32,G^SCH$GL_COMQS,R2 ; Find first full state
                          B9     13    012D          BEQL    SCH$IDLE                ; No executable process?
     53   00000000'GF42    7E    012F          MOVAQ   G^SCH$AQ_COMH[R2],R3    ; Compute queue head address
                    54    93     0F    0137          REMQUE  @(R3)+,R4               ; Get head of queue
                          A8     1D    013A          BVS     QEMPTY                  ; Br if queue was empty (bugcheck)
                          08     12    013C          BNEQ    20$                     ; Queue not empty
  00 00000000'GF          52     E5    013E          BBCC    R2,G^SCH$GL_COMQS,20$   ; Set queue empty
```

```
                         0146   20$:                                    ;
        0A A4    0C   91 0146          CMPB    #DYN$C_PCB,PCB$B_TYPE(R4) ; Must be a process control block
                 98   12 014A          BNEQ    QEMPTY                    ; Otherwise fatal error
        2C A4    0E   B0 014C          MOVW    #SCH$C_CUR,PCB$W_STATE(R4) ; Set state to current
 00000000'GF     54   D0 0150          MOVL    R4,G^SCH$GL_CURPCB        ; Note current PCB location
        10    18 A4   DA 0157          MTPR    PCB$L_PHYPCB(R4),#PR$_PCBB ; Set PCB base physical address
                      06 015B          LDPCTX                            ; Restore context
                 54   DD 015C          PUSHL   R4                        ; Save register
     54  00000000'GF   D0 015E          MOVL    G^SCH$GL_CURPCB,R4        ; Get address of current PCB
      00 0000'CF      00 E6 0165       BBSSI   #LCK$V_INTERLOCK,W^MPS$GL_INTERLOCK,40$ ; Flush cache queue
         01   0000'CF D1 016B   40$:   CMPL    W^MPS$GL_STATE,#MPS$K_IDLESTATE ; Is secondary idle?
                 3A   12 0170          BNEQ    50$                       ; Br on no, already running a process
                      0172             MFPR    #PR$_ASTLVL,-(SP)         ; Get PR ASTLVL
                 8E   D5 0175          TSTL    (SP)+                     ; Is ASTLVL = KERNEL?
                 33   13 0177          BEQL    50$                       ; Br on yes, dont cause reschedule
 00000000'GF  7FFFFFFF 8F D3 0179      BITL    #^X7FFFFFFF,G^SCH$GL_COMQS ; Is any process other than null COM?
                 26   13 0184          BEQL    50$                       ; Br on no, nothing else can run
        10 A4    9F   0186             PUSHAB  PCB$L_ASTQFL(R4)          ; Are there any AST's outstanding
     10 A4       8E   D1 0189          CMPL    (SP)+,PCB$L_ASTQFL(R4)    ; for this process?  If so, don't
                 1D   12 018D          BNEQ    50$                       ; cause a reschedule AST.
     54     6C A4    D0 018F          MOVL    PCB$L_PHD(R4),R4          ; Get address of PHD
           0104 C4    D5 0193          TSTL    PHD$L_MPINHIBIT(R4)       ; Any reason not to run on secondary?
                 0C   12 0197          BNEQ    45$                       ; Br on yes, don't cause reschedule AST
                      0199
                      0199             .IF     DF,MPPFMSWT
                      0199             BSBW    W^MPS$PFM_ASTSC
                      0199             .ENDC
                      0199
        13       01   DA 0199          MTPR    #PSL$C_EXEC,#PR$_ASTLVL ; Cause reschedule of process when it
                      019C                                             ; exits from kernel mode on primary
     54  00000000'GF   D0 019C          MOVL    G^SCH$GL_CURPCB,R4        ; Get address of current PCB
                 16   11 01A3          BRB     60$                       ; Don't decrement priority
     54  00000000'GF   D0 01A5   45$:   MOVL    G^SCH$GL_CURPCB,R4        ; Get address of current PCB
     0B A4    2F A4   91 01AC   50$:   CMPB    PCB$B_PRIB(R4),PCB$B_PRI(R4) ; Check for base
                 08   13 01B1          BEQL    60$                       ; Br if Priority=current
     03 0B A4    04   E1 01B3          BBC     #4,PCB$B_PRI(R4),60$      ; Don't float real time priority
        0B A4    96   01B8             INCB    PCB$B_PRI(R4)             ; Move toward base priority
 00000000'GF     0B A4 90 01BB   60$:  MOVB    PCB$B_PRI(R4),G^SCH$GB_PRI ; Set global priority
                 54 8ED0      01C3     POPL    R4                        ; Restore register
                    02        01C6     REI                               ; Normal return
                              01C7
```

```
                        01C7          .SBTTL  MPS$MPSCHED - SECONDARY PROCESSOR SCHEDULING
                        01C7   ;++
                        01C7   ; FUNCTIONAL DESCRIPTION:
                        01C7   ;
                        01C7   ; MPS$MPSCHED is entered to drop the current process running
                        01C7   ; in the secondary processor and select a new one.
                        01C7   ;
                        01C7   ; ENVIRONMENT:
                        01C7   ;
                        01C7   ;       Executed by the secondary processor.
                        01C7   ;
                        01C7   ;--
                        01C7          .ALIGN  LONG
                        01C8   MPS$MPSCHED::                                  ;
                        01C8          SETIPL  #IPL$_SYNCH                     ; Synchronize with other events
04 AE   03000000 8F  D3 01CB          BITL    #PSL$M_CURMOD,4(SP)             ; Check for kernel
                   3E  13 01D3          BEQL    GETOUT                          ; If so, exit already in progress
                        01D5
                        01D5   MPS$MPSCHED2::
                        01D5
                        01D5          .IF     DF,MPPFMSWT
                        01D5          BSBW    MPS$PFM_SVPCTX                  ; Gather performance measurement data
                        01D5          .ENDC
                        01D5
                        01D5          SETIPL  #IPL$_POWER                     ; Prevent duplicate SVPCTX from a MA780
                        01D8                                                  ; interrupt for STOP/CPU
                   07   01D8          SVPCTX                                  ; Save current process context
        0000'CF    02 D0 01D9          MOVL    #MPS$K_DROPSTATE,W^MPS$GL_STATE ; Indicate process being dropped
                        01DE          SETIPL  #IPL$_SYNCH                     ; Now allow MA780 interrupts again
                        01E1
                        01E1   MPS$MPSCHED1::                                 ; Init entry point for startup
                        01E1
                        01E1          .IF     DF,MPPFMSWT
                        01E1          BSBW    MPS$PFM_INTP                    ; Gather performance measurement data
                        01E1          .ENDC
                        01E1
            FE1C'   30  01E1          BSBW    MPS$INTPRIM                     ; Interrupt primary processor
                        01E4   ;
                        01E4   ; Now invalidate the system half of the translation buffer because
                        01E4   ; the secondary is going into a busy-waiting loop until the primary
                        01E4   ; gives it something to do.  The primary may request that the secondary
                        01E4   ; invalidate certain system addresses from its translation buffer while
                        01E4   ; it is waiting.  Since the secondary won't have these addresses in its
                        01E4   ; translation buffer, it just keeps telling the primary that it has done
                        01E4   ; the request, i.e., by clearing MPS$GL_INVALID.  (If this clear was not
                        01E4   ; in the loop, the primary would wait indefinitely for the secondary, who
                        01E4   ; in turn was waiting indefinitely for the primary.)
                        01E4   ;
                        01E4          INVALID                                 ; Invalidate translation buffer
          0000'CF    D4 01E7   10$:    CLRL    W^MPS$GL_INVALID                ; Acknowledge TB invalidate request
00 0000'CF    00    E6 01EB          BBSSI   #LCK$V_INTERLOCK,W^MPS$GL_INTERLOCK,20$ ; Flush cache queue
   0000'CF    03    D1 01F1   20$:    CMPL    #MPS$K_BUSYSTATE,W^MPS$GL_STATE ; Is state set to busy?
              EF    12 01F6          BNEQ    10$                             ; No, loop
                        01F8   ;
                        01F8   ; The primary has scheduled a process for the secondary to execute.
                        01F8   ; The secondary need only load up the new PCB and being executing.
                        01F8   ;
```

```
                    01F8   447          SETIPL  #IPL$_POWER                    ; Disable powerfail interrupts
00 0000'CF    00    E6  01FB   448          BBSSI   #LCK$V_INTERLOCK,W^MPS$GL_INTERLOCK,30$ ; Flush cache queue
   54    0000'CF    D0  0201   449 30$:     MOVL    W^MPS$GL_CURPCB,R4             ; Get current PCB address
     52    18 A4    D0  0206   450          MOVL    PCB$L_PHYPCB(R4),R2            ; Get physical PCB address
        10    52    DA  020A   451          MTPR    R2,#PR$_PCBB                   ; Set PCB base address
                    020D   452
                    020D   453          .IF     DF,MPPFMSWT
                    020D   454          BSBW    MPS$PFM_LDPCTX                 ; Gather performance measurement data
                    020D   455          .ENDC
                    020D   456
                 06 020D   457          LDPCTX                                 ; Load process context
   0000'CF    04    D0  020E   458          MOVL    #MPS$K_EXECSTATE,W^MPS$GL_STATE ; Indicate LDPCTX is done
                 02 0213   459 GETOUT: REI                                     ; And execute
                    0214   460          .END
```

```
BUG$_QUEUEMPTY          ********   X   02
DYN$C_PCB             = 0000000C
GETOUT                  00000213 R     02
IPL$_POWER            = 0000001F
IPL$_SCHED            = 00000003
IPL$_SYNCH            = 00000008
LCK$V_INTERLOCK       = 00000000
MPS$EXIT                0000000E R     02
MPS$GL_CURPCB           ********   X   02
MPS$GL_INTERLOCK        ********   X   02
MPS$GL_INVALID          ********   X   02
MPS$GL_PFAILTIM         ********   X   02
MPS$GL_SECREQFLG        ********   X   02
MPS$GL_STATE            ********   X   02
MPS$IDLE                000000DA R     02
MPS$INTPRIM             ********   X   02
MPS$K_BUSYSTATE       = 00000003
MPS$K_DROPSTATE       = 00000002
MPS$K_EXECSTATE       = 00000004
MPS$K_IDLESTATE       = 00000001
MPS$K_INITSTATE       = 00000005
MPS$K_STOPSTATE       = 00000006
MPS$MPSCHED             000001C8 RG    02
MPS$MPSCHED1            000001E1 RG    02
MPS$MPSCHED2            000001D5 RG    02
MPS$RESCHED             000000F8 RG    02
MPS$RESCHEDIPL5         00000000 RG    02
MPS$SCHED               0000011E RG    02
MPS$SCHEDIPL5           0000006D RG    02
MPS$SCHSCND             0000001C RG    02
MPS$V_SECWAITCK       = 00000002
MPS$WAITCK              ********   X   02
NEXTQ                   0000007E R     02
PCB$B_PRI             = 0000000B
PCB$B_PRIB            = 0000002F
PCB$B_TYPE            = 0000000A
PCB$L_ASTQFL          = 00000010
PCB$L_PHD             = 0000006C
PCB$L_PHYPCB          = 00000018
PCB$W_STATE           = 0000002C
PFAIL_EXIT              00000007 R     02
PHD$B_ASTLVL          = 000000CF
PHD$L_MPINHIBIT       = 00000104
PHD$L_PSL             = 000000C4
PR$_ASTLVL            = 00000013
PR$_IPL               = 00000012
PR$_PCBB              = 00000010
PR$_SIRR              = 00000014
PR$_TBIA              = 00000039
PSL$C_EXEC            = 00000001
PSL$M_CURMOD          = 03000000
PSL$S_CURMOD          = 00000002
PSL$V_CURMOD          = 00000018
QEMPT                   000000C7 R     02
QEMPTY                  000000E4 R     02
QLOOP                   00000089 R     02
SCH$AQ_COMH             ********   X   02
SCH$AQ_COMT             ********   X   02
SCH$C_COM             = 0000000C
SCH$C_CUR             = 0000000E
SCH$GB_PRI              ********   X   02
SCH$GL_COMQS            ********   X   02
SCH$GL_CURPCB           ********   X   02
SCH$GL_NULLPCB          ********   X   02
SCH$IDLE                000000E8 R     02
```

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

PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
$ABS$                           00000000  (    0.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
AEXENONPAGED                    00000214  (  532.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC LONG

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

Phase                     Page faults    CPU Time       Elapsed Time
-----                     -----------    --------       ------------
Initialization                     30    00:00:00.06    00:00:01.02
Command processing                108    00:00:00.86    00:00:03.85
Pass 1                            252    00:00:06.96    00:00:20.38
Symbol table sort                   0    00:00:00.84    00:00:01.80
Pass 2                             97    00:00:01.76    00:00:06.14
Symbol table output                 9    00:00:00.09    00:00:00.26
Psect synopsis output               2    00:00:00.02    00:00:00.02
Cross-reference output              0    00:00:00.00    00:00:00.00
Assembler run totals              500    00:00:10.59    00:00:33.47

The working set limit was 1350 pages.
36516 bytes (72 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 597 non-local and 17 local symbols.
465 source lines were read in Pass 1, producing 15 object records in Pass 2.
22 pages of virtual memory were used to define 21 macros.

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

Macro library name                      Macros defined
------------------                      --------------
_$255$DUA28:[MP.OBJ]MP.MLB;1                         4
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                       9
_$255$DUA28:[SYSLIB]STARLET.MLB;2                    6
TOTALS (all libraries)                              19

745 GETS were required to define 19 macros.

There were no errors, warnings or information messages.

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

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

MPSWPFM
LIS

MPLOAD
LIS

MPINIT
LIS