```
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**MPWAIT

J 4

```
MM      MM  PPPPPPPP    WW      WW    AAAAAA      IIIIII    TTTTTTTTTT
MM      MM  PPPPPPPP    WW      WW    AAAAAA      IIIIII    TTTTTTTTTT
MMMM  MMMM  PP      PP  WW      WW  AA      AA      II          TT
MMMM  MMMM  PP      PP  WW      WW  AA      AA      II          TT
MM  MM  MM  PP      PP  WW      WW  AA      AA      II          TT
MM  MM  MM  PP      PP  WW      WW  AA      AA      II          TT
MM      MM  PPPPPPPP    WW      WW  AA      AA      II          TT
MM      MM  PPPPPPPP    WW      WW  AA      AA      II          TT
MM      MM  PP          WW  WW  WW  AAAAAAAAAA      II          TT
MM      MM  PP          WW  WW  WW  AAAAAAAAAA      II          TT
MM      MM  PP          WWWW  WWWW  AA      AA      II          TT          ....
MM      MM  PP          WWWW  WWWW  AA      AA      II          TT          ....
MM      MM  PP          WW      WW  AA      AA    IIIIII        TT          ....
MM      MM  PP          WW      WW  AA      AA    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
```

MPW
V04

```
          0000  ;
          0000  ; Version:       'V04-000'
          0000  ;
          0000
          0000          .MCALL  MFPR
00000001  0000  MPSWITCH = 1
          0000          .NLIST  CND
          0000          .TITLE  MPWAIT - SECONDARY EVENT FLAG WAIT SERVICES
          0000          .IDENT  'V04-000'
          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  ; FACILITY: MULTI-PROCESSING EXECUTIVE, SECONDARY EVENT FLAG SERVICES
          0000  ;
          0000  ; ABSTRACT: $WAITFR THAT SUCEEDS IS DONE ON SECONDARY.  ALL
          0000  ;           OTHER CASES ARE RETURNED TO THE PRIMARY FOR HANDLING.
          0000  ;
          0000  ;--
          0000  ;
          0000  ; AUTHOR:
          0000  ;       R.HUSTVEDT : VERSION
          0000  ;
          0000  ; MODIFIED BY:
          0000  ;
          0000  ;       V03-007 SSA0016         Stan Amway              8-Mar-1984
          0000  ;               Subtract IOTA from automatic working set adjustment
          0000  ;               time reference in PHD on any entry to SCH$WAITx code.
          0000  ;               (Acknowledgements go to Wayne Cardoza and Larry Kenah,
          0000  ;                who both collaborated on this change.)
          0000  ;
          0000  ;       V03-006 WMC0001         Wayne Cardoza           22-Feb-1984
          0000  ;               No reason to try to wake up swapper on every EFN wait.
          0000  ;
          0000  ;       V03-005 SSA0003         Stan Amway              5-Dec-1983
          0000  ;               Added support for outswap scheduling changes.
          0000  ;               Changed process wait code to store wait time in PCB
          0000  ;               as longword system absolute time.
```

```
0000     64 ;
0000     65 ;       V03-004 KDM0035         Kathleen D. Morse       14-Dec-1982
0000     66 ;               Fix assembly switch for performance collection for
0000     67 ;               kernel mode services executed on secondary processor.
0000     68 ;
0000     69 ;       V03-003 KDM0034         Kathleen D. Morse       13-Dec-1982
0000     70 ;               Correct logic for secondary continuing execution of
0000     71 ;               a process after a WAITCHK request is done by the primary.
0000     72 ;
0000     73 ;       V03-002 KDM0030         Kathleen D. Morse       18-Nov-1982
0000     74 ;               Add IFPRIMARY logic that allows primary to execute
0000     75 ;               secondary-specific code without turning into a secondary.
0000     76 ;
0000     77 ;       V03-001 KDM0018         Kathleen D. Morse       13-Oct-1982
0000     78 ;               Add multi-processing switch, which generates these
0000     79 ;               system services for the secondary processor.
0000     80 ;
0000     81 ;
```

```
                    0000              .SBTTL  DECLARATIONS
                    0000
                    0000      ;
                    0000      ; INCLUDE FILES:
                    0000      ;
                    0000
                    0000              $CEBDEF                                  ;COMMON EVENT BLOCK DEFS
                    0000              $DYNDEF                                  ;DYNAMIC STRUCTURE TYPES
                    0000              $IPLDEF                                  ;IPL DEFINITIONS
                    0000              $LCKDEF                                  ;INTERLOCK BIT DEFINITIONS
                    0000              $MPSDEF                                  ;SECONDARY REQUEST FLAG DEFS
                    0000              $RPBDEF                                  ;REBOOT PARAMETER BLOCK DEFS
                    0000              $PCBDEF                                  ;PCB DEFINITIONS
                    0000              $PHDDEF                                  ;PHD DEFINITIONS
                    0000              $PRDEF                                   ;PROCESSOR REGISTER DEFS
                    0000              $PSLDEF                                  ;PSL DEFINITIONS
                    0000              $SHBDEF                                  ;SHARED MEMORY CONTROL BLK DEFS
                    0000              $SHDDEF                                  ;SHARED MEMORY COMMON DATA PAGE
                    0000              $SSDEF                                   ;STATUS CODE DEFINITIONS
                    0000              $STATEDEF                                ;STATE DEFINITIONS
                    0000              $WQHDEF                                  ;WAIT QUEUE HEADER DEFS
                    0000      ;
                    0000      ; EQUATES:
                    0000      ;
          00000004  0000      EFN=4                                            ;EVENT FLAG NUMBER
          00000008  0000      MASK=8                                           ;WAIT MASK
                00000000              .PSECT  AEXENONPAGED                     ;NON-PAGED
```

```
                  0000   116          .SBTTL  MPS$WFLAND - WAIT FOR LOGICAL AND OF EVENT FLAGS
                  0000   118 ;++
                  0000   119 ; FUNCTIONAL DESCRIPTION:
                  0000   120 ;
                  0000   124 ;        MPS$WFLAND RETURNS TO THE CALLER WHEN THE SET OF EVENT
                  0000   126 ;        FLAGS SELECTED BY THE MASK ARE ALL SET AND RETURNS THE
                  0000   127 ;        STATE OF ALL EVENT FLAGS IN THE SPECIFIED CLUSTER.
                  0000   128 ;
                  0000   129 ; CALLING SEQUENCE:
                  0000   130 ;
                  0000   134 ;        CALLG   ARGLIST,MPS$WFLAND
                  0000   136 ;
                  0000   137 ; INPUT PARAMETERS:
                  0000   138 ;
                  0000   139 ;        04(AP) - EVENT FLAG NUMBER SELECTING CLUSTER
                  0000   140 ;        08(AP) - MASK SELECTING COMBINATION OF EVENTS
                  0000   141 ;        R4 - PCB ADDRESS OF CURRENT PROCESS
                  0000   142 ;
                  0000   143 ; OUTPUT PARAMETERS:
                  0000   144 ;
                  0000   145 ;        R0 - COMPLETION STATUS CODE
                  0000   146 ;             CONDITION IS SATISFIED.
                  0000   147 ;
                  0000   148 ; SIDE EFFECTS:
                  0000   149 ;
                  0000   150 ;        IF THE SET OF EVENT FLAGS SELECTED BY THE MASK ARE NOT
                  0000   151 ;        ALL SET, THEN THE PROCESS ISSUING THE WAIT SERVICE CALL WILL
                  0000   152 ;        BE PLACED IN A WAIT STATE.
                  0000   153 ;
                  0000   154 ; COMPLETION CODES:
                  0000   155 ;
                  0000   156 ;        SS$_NORMAL - NORMAL SUCCESSFUL COMPLETION
                  0000   157 ;        SS$_ILLEFC - ILLEGAL EVENT FLAG CLUSTER NUMBER.  EVENT NUMBER
                  0000   158 ;                     NOT IN THE RANGE 0-127.
                  0000   159 ;        SS$_UNASEFC - UNASSIGNED EVENT FLAG CLUSTER.
                  0000   161 ;
                  0000   162 ; ENVIRONMENT:
                  0000   163 ;
                  0000   164 ;        EXECUTES ON SECONDARY PROCESSOR, MODE=KERNEL.
                  0000   165 ;        IF INTERRUPTED AT ANY POINT, MAY CONTINUE ON PRIMARY.
                  0000   167 ;
                  0000   168 ;--
                  0000   169
                  0000   173 MPS$WFLAND::                                 ;WAIT FOR LOGICAL AND
            003C  0000   175          .WORD   ^M<R2,R3,R4,R5>             ;REGISTER SAVE MASK FOR R2-R5
   51    01   D0  0002   176          MOVL    #1,R1                       ;SET MODE TO WAITALL
         04   11  0005   177          BRB     WFRL                        ;AND MERGE WITH COMMON CODE
```

```
                    0007         .SBTTL  MPS$WFLOR - WAIT FOR LOGICAL OR OF EVENTS
                    0007  ;++
                    0007  ; FUNCTIONAL DESCRIPTION:
                    0007  ;
                    0007  ;      MPS$WFLOR RETURNS TO THE CALLER WHEN ANY OF THE
                    0007  ;      EVENTS SELECTED BY THE MASK WITHIN THE SPECIFIED CLUSTER
                    0007  ;      ARE SET AND RETURNS THE STATE OF ALL 32 EVENT FLAGS IN THE
                    0007  ;      CLUSTER.
                    0007  ;
                    0007  ; CALLING SEQUENCE:
                    0007  ;
                    0007  ;      CALLG   ARGLIST,MPS$WFLOR
                    0007  ;
                    0007  ; INPUT PARAMETERS:
                    0007  ;
                    0007  ;      04(AP) - EVENT FLAG NUMBER TO SELECT CLUSTER
                    0007  ;      08(AP) - MASK SELECTING DESIRED COMBINATION OF EVENTS
                    0007  ;      R4 - PCB ADDRESS OF CURRENT PROCESS
                    0007  ;
                    0007  ; OUTPUT PARAMETERS:
                    0007  ;
                    0007  ;      R0 - COMPLETION STATUS CODE
                    0007  ;           IS SATISFIED.
                    0007  ;
                    0007  ; COMPLETION CODES:
                    0007  ;
                    0007  ;      SS$_NORMAL - NORMAL SUCCESSFUL COMPLETION
                    0007  ;      SS$_ILLEFC - ILLEGAL EVENT FLAG NUMBER NOT IN THE RANGE 0-127.
                    0007  ;      SS$_UNASEFC - UNASSIGNED EVENT FLAG CLUSTER.
                    0007  ;
                    0007  ; SIDE EFFECTS:
                    0007  ;
                    0007  ;      THE PROCESS ISSUING THE SERVICE CALL IS BE PLACED IN A
                    0007  ;      WAIT STATE IF NONE OF THE SPECIFIED EVENTS ARE SET.
                    0007  ;
                    0007  ; ENVIRONMENT:
                    0007  ;
                    0007  ;      EXECUTES ON SECONDARY PROCESSOR, MODE=KERNEL.
                    0007  ;      IF INTERRUPTED AT ANY POINT, MAY CONTINUE ON PRIMARY.
                    0007  ;
                    0007  ;--
                    0007
                    0007  MPS$WFLOR::                                   ;WAIT FOR LOGICAL OR
              003C  0007         .WORD   ^M<R2,R3,R4,R5>               ;REGISTER SAVE MASK FOR R2-R5
           51   D4  0009         CLRL    R1                            ;SET MODE TO WAIT ANY
50    08 AC   D0  000B  WFRL:  MOVL    MASK(AP),R0                   ;GET WAIT MASK
           09   11  000F         BRB     MPS$WAIT                      ;MERGE WITH COMMON CODE
```

```
                          0011          .SBTTL  MPS$WAITFR - WAIT FOR SINGLE EVENT
                          0011   ;++
                          0011   ; FUNCTIONAL DESCRIPTION:
                          0011   ;
                          0011   ;          MPS$WAITFR RETURNS TO THE CALLER WHEN THE SPECIFIED SINGLE
                          0011   ;          EVENT FLAG IS SET.  UPON RETURN THE STATE OF ALL 32 EVENT FLAGS
                          0011   ;          WITHIN THE CLUSTER CONTAINING THE SPECIFIED EVENT ARE RETURN.
                          0011   ;
                          0011   ; CALLING SEQUENCE:
                          0011   ;
                          0011   ;          CALLG   ARGLIST,MPS$WAITFR
                          0011   ;
                          0011   ; INPUT PARAMETERS:
                          0011   ;
                          0011   ;          04(AP)=EVENT FLAG NUMBER
                          0011   ;          R4 - PCB ADDRESS OF CURRENT PROCESS
                          0011   ;
                          0011   ; OUTPUT PARAMETERS:
                          0011   ;
                          0011   ;          R0 - COMPLETION STATUS CODE
                          0011   ;               SATISIFIED.
                          0011   ;
                          0011   ; SIDE EFFECTS:
                          0011   ;
                          0011   ;          IF THE SPECIFIED EVENT FLAG IS NOT SET, THE PROCESS ISSUING THE
                          0011   ;          WAIT SYSTEM SERVICE WILL BE PLACED IN THE APPROPRIATE WAIT
                          0011   ;          STATE.
                          0011   ;
                          0011   ; COMPLETION CODES:
                          0011   ;
                          0011   ;          SS$_NORMAL - NORMAL SUCCESSFUL COMPLETION
                          0011   ;          SS$_ILLEFC - ILLEGAL EVENT FLAG NUMBER NOT IN THE RANGE 0-127.
                          0011   ;          SS$_UNASEFC - UNASSIGNED EVENT FLAG CLUSTER.
                          0011   ;
                          0011   ; ENVIRONMENT:
                          0011   ;
                          0011   ;          EXECUTES ON SECONDARY PROCESSOR, MODE=KERNEL.
                          0011   ;--
                          0011
                          0011   MPS$WAITFR::                                  ;WAIT FOR SINGLE EVENT
                   003C   0011          .WORD   ^M<R2,R3,R4,R5>               ;SAVE REGISTERS R2,R3,R4,R5
              51     D4   0013          CLRL    R1                            ;SET MODE
  50   01   04 AC    9C   0015          ROTL    EFN(AP),#1,R0                 ;INIT MASK
                          001A   ;      BRB     MPS$WAIT                      ;AND MERGE WITH COMMON CODE
```

```
                             001A         .SBTTL  MPS$WAIT - WAIT COMMON CODE
                             001A     ;++
                             001A     ; FUNCTIONAL DESCRIPTION:
                             001A     ;
                             001A     ;       THIS IS THE COMMON WAIT CODE FOR ALL THE EVENT FLAG WAIT
                             001A     ;       SYSTEM SERVICES.
                             001A     ;
                             001A     ; INPUT PARAMETERS:
                             001A     ;
                             001A     ;       04(AP) = EVENT FLAG NUMBER
                             001A     ;       R0 = MASK SELECTING EVENTS OF INTEREST
                             001A     ;       R1 = ANY/ALL MODE SELECTOR
                             001A     ;               0 => ANY
                             001A     ;               1 => ALL
                             001A     ;       R4 = PCB ADDRESS OF CURRENT PROCESS
                             001A     ;
                             001A     ; IMPLICIT INPUTS:
                             001A     ;
                             001A     ;       CEB IF NON-LOCAL CLUSTER.
                             001A     ;
                             001A     ; OUTPUT PARAMETERS:
                             001A     ;
                             001A     ;       R0 - COMPLETION STATUS CODE
                             001A     ;            SATISIFIED.
                             001A     ;
                             001A     ; ENVIRONMENT:
                             001A     ;
                             001A     ;       EXECUTES ON SECONDARY PROCESSOR, MODE=KERNEL.
                             001A     ;       IF INTERRUPTED AT ANY POINT, MAY CONTINUE ON PRIMARY.
                             001A     ;
                             001A     ;--
                             001A
                             001A    MPS$WAIT:                                        ;WAIT COMMON CODE
        52    04 AC     98   001A            CVTBL   EFN(AP),R2                ;GET CLUSTER NUMBER
              1A        19   001E            BLSS    10$                       ;ILLEGAL IF NOT (0,1,2,3)
  52    52    FB 8F     78   0020            ASHL    #-5,R2,R2                 ;RIGHT ALIGN CLUSTER NUMBER
     53   50 A442       DE   0025            MOVAL   PCB$L_EFCS(R4)[R2],R3     ;POINTER TO PCB EVENT CLUSTER
       2E A4    52      90   002A            MOVB    R2,PCB$B_WEFC(R4)         ;SAVE WAIT CLUSTER NUMBER
            15 52       F5   002E            SOBGTR  R2,30$                    ;BR IF COMMON CLUSTER R2 = (2,3)
  52    00000000'GF     7E   0031            MOVAQ   G^SCH$GQ_LEFWQ,R2         ;SET WAIT QUEUE POINTER
              23        11   0038            BRB     WAITCK                    ;
     50    00EC 8F      3C   003A    10$:    MOVZWL  #SS$_ILLEFC,R0            ;SET ERROR CODE FOR ILLEGAL CLUSTER
              04        04   003F            RET                               ;AND EXIT
     50    0234 8F      3C   0040    20$:    MOVZWL  #SS$_UNASEFC,R0           ;SET ERROR CODE FOR UNASSIGNED
              04        04   0045            RET                               ;AND EXIT
     52    63    10     C1   0046    30$:    ADDL3   #CEB$L_EFC,(R3),R2        ;GET CEB ADDRESS FOR EVENT FLAGS
              F4        18   004A            BGEQ    20$                       ;CEB ASSIGNED (SYSTEM SPACE ADDRESS)
       FA A2    2D      91   004C            CMPB    #DYN$C_SLAVCEB,<CEB$B_TYPE-CEB$L_EFC>(R2) ;IS THIS IN SH MEM?
              08        12   0050            BNEQ    40$                       ;BR IF IN LOCAL MEMORY
     53    30 A2        D0   0052            MOVL    <CEB$L_MASTER-CEB$L_EFC>(R2),R3 ;GET ADR OF MASTER CEB
     62    10 A3        D0   0056            MOVL    CEB$L_EFC(R3),(R2)        ;COPY EFC FROM MASTER TO SLAVE CEB
                             005A            ASSUME  <CEB$L_EFC+4> EQ CEB$L_WQFL
        53    82        DE   005A    40$:    MOVAL   (R2)+,R3                  ;GET EVENT POINTER AND WAIT QUEUE ADDR
                             005D                                              ; R3=CEB$L_EFC, R2=CEB$L_WQFL
                             005D     ;
                             005D     ;       R0 - MASK SELECTING EVENTS OF INTEREST
                             005D     ;       R1 - ANY/ALL MODE SELECTOR
```

```
                            005D   :        R2 - ADDRESS OF WAIT QUEUE HEADER
                            005D   :        R3 - ADDRESS OF EVENT FLAG VECTOR
                            005D   :        R4 - PCB ADDRESS
                            005D   :
                            005D   WAITCK:                                          ;CHECK FOR WAIT SATISFIED
                            005D            SETIPL  #IPL$_SYNCH                     ;BLOCK SCHEDULING ACTIVITY
             50   63   D3   0060            BITL    (R3),R0                         ;WAIT FOR LOGICAL OR MAY BE SATISFIED
                  13   13   0063            BEQL    WAIT                            ;NO, MUST WAIT
             0B 51     E8   0065            BLBS    R1,WAITALL                      ; 1 => WAIT FOR ALL IN MASK
                            0068   NOWAIT:                                          ;
          5D   0C AD   D0   0068            MOVL    12(FP),FP                       ;GET SAVED FRAME POINTER
             5E   00'  C0   006C            ADDL    S^#EXE$C_CMSTKSZ,SP             ;CLEAN STACK TO PC,PSL
                            006F   MPS$WAITCONT::                                   ;CONTINUE HERE AFTER PRIMARY WAITCHK
             50   01   9A   006F            MOVZBL  #SS$_NORMAL,R0                  ;RETURN SUCCESS CODE
                            0072
                            0072
                       02   0072            REI                                     ;RETURN TO CALLER
                            0073
                            0073   WAITALL:                                         ; WAIT FOR ALL SELECTED EVENTS
             50   63   CA   0073            BICL2   (R3),R0                         ; CLEAR BITS FOR ALREADY SET FLAGS
                  F0   13   0076            BEQL    NOWAIT                          ;YES, DONT WAIT
24 A4    01   0D  51   F0   0078   WAIT:    INSV    R1,#PCB$V_WALL,#1,PCB$L_STS(R4) ;SET WAIT ALL FLAG
      4C A4       50   D2   007E            MCOML   R0,PCB$L_EFWM(R4)               ;SAVE INVERTED WAIT MASK
                            0082
                            0082   ;
                            0082   ; WAIT CONDITION IS NOT SATISFIED.  PROCESS IS RETURNED TO THE PRIMARY
                            0082   ; ALONG WITH A REQUEST BIT TO CHECK IF THE WAIT CONDITION IS SATISFIED.
                            0082   ;
                            0082   ; THE SECONDARY WAITS IN ITS IDLE LOOP, WHILE THE PRIMARY CONTINUES
                            0082   ; EXECUTION AT MPS$WAITCK.
                            0082   ;
                            0082   ; THE PROCESS WILL BE IN A STATE READY FOR RESCHEDULING IF THE WAIT
                            0082   ; CONDITION IS NOT SATISFIED, AND THE PRIMARY WILL CONTINUE EXECUTION
                            0082   ; AT SCH$WAIT IN THIS CASE.
                            0082   ;
                            0082            IFPRIMARY <JMP G^SCH$WAIT>              ;IF PRIMARY, THEN CONTINUE
                            009B                                                    ;IF SECONDARY, ASK FOR PRIMARY ASSIST
          5D   0C AD   D0   009B            MOVL    12(FP),FP                       ;PUT SAVED FRAME POINTER WHERE PRIMARY
                            009F                                                    ; CAN FIND IT AS IT CAN'T TOUCH STACK
             5E   00'  C0   009F            ADDL    S^#EXE$C_CMSTKSZ,SP             ;CLEAN STACK TO PC/PSL OF CHMK INSTR
        50   6E   04   C3   00A2            SUBL3   #4,(SP),R0                      ;GET PS/PSL PAIR FOR CHMK INSTRUCTION
          51   04 AE   D0   00A6            MOVL    4(SP),R1                        ; INTO A PLACE THE PRIMARY CAN TOUCH
                  7E   DC   00AA            MOVPSL  -(SP)                           ;PUT PC/PSL PAIR ON STACK FOR SECONDARY
         0000006F'GF   9F   00AC            PUSHAB  G^MPS$WAITCONT                  ; PROCESSOR TO CONTINUE EXECUTING PROC
     00 0000'CF   02   E6   00B2            BBSSI   #MPS$V_SECWAITCK,W^MPS$GL_SECREQFLG,10$ ;IND WAIT CHECK REQUEST
               FF45'   31   00B8   10$:     BRW     MPS$MPSCHED2                    ;GO WAIT IN IDLE LOOP FOR WORK TO DO
                            00BB
```

```
00BB         .SBTTL  MPS$WAITCK - PRIMARY CHECK WAIT CONDITION FOR POTENTIAL RESCHED
00BB   ;++
00BB   ; FUNCTIONAL DESCRIPTION:
00BB   ;
00BB   ;       MPS$WAITCK IS EXECUTED BY THE PRIMARY PROCESSOR WHENEVER
00BB   ;       THE SECONDARY PROCESSOR EXECUTES AN EVENT FLAG WAIT SYSTEM
00BB   ;       SERVICE THAT CAUSES A PROCESS TO ACTUALLY WAIT.  THE PRIMARY
00BB   ;       PROCESSOR EXECUTES THIS CODE DUE TO A SECONDARY REQUEST FLAG,
00BB   ;       DISPATCHED FROM THE SECONDARY RESCHEDULING CODE.  THIS IS
00BB   ;       ENTERED VIA THE IPL 5 (MULTI-PROCESSING SECONDARY RESCHEDULE
00BB   ;       REQUEST) INTERRUPT OR THE IPL 3 (PRIMARY RESCHEDULE) INTERRUPT.
00BB   ;
00BB   ;       NOTE:  A RACE CONDITION EXISTS IF THE PRIMARY IS IN THE IPL 3
00BB   ;       HANDLER WHEN THE SECONDARY SETS THE REQUEST FLAG, AND THUS THIS
00BB   ;       CODE MAY BE EXECUTED FROM THE IPL 3 LOGIC, THOUGH IT IS INTENDED
00BB   ;       TO BE EXECUTED FROM THE IPL 5 PATH.
00BB   ;
00BB   ;       THE SECONDARY HAS DONE A SVPCTX FOR THIS PROCESS JUST PRIOR
00BB   ;       TO WHERE THE PRIMARY WOULD HAVE EXECUTED SCH$WAIT, IF THE
00BB   ;       SYSTEM SERVICE WAS BEING EXECUTED ON THE PRIMARY.  THIS
00BB   ;       ALLOWS THE PRIMARY TO LOOK IN THE HARDWARE PCB FOR THE
00BB   ;       INFORMATION NEEDED TO CHECK THE EVENT FLAG WAIT CONDITION
00BB   ;       AND IF RESCHEDULING IS NEEDED, TO CONTINUE EXECUTING
00BB   ;       THIS PROCESS AT THE EQUIVALENT OF THE SCH$WAIT LOGIC AT
00BB   ;       SECWAIT.
00BB   ;
00BB   ; CALLING SEQUENCE:
00BB   ;
00BB   ;       JSB     MPS$WAITCHK
00BB   ;
00BB   ; INPUT PARAMETERS:
00BB   ;
00BB   ;       MPS$GL_CURPCB - ADDRESS OF PCB FOR CURRENT PROCESS ON SECONDARY
00BB   ;
00BB   ;       IN THE PHD FOR THE CURRENT PROCESS ON SECONDARY.
00BB   ;
00BB   ;               R0 - PC OF CHMK INSTRUCTION
00BB   ;               R1 - PSL FOR RE-EXECUTION OF CHMK INSTRUCTION
00BB   ;               R2 - WAIT QUEUE HEADER ADDRESS
00BB   ;               R3 - ADDRESS OF EVENT FLAG VECTOR
00BB   ;               R4 - PCB ADDRESS
00BB   ;               00(SP) - PC AT WHICH TO RE-EXECUTE THE CHMK INSTRUCTION
00BB   ;               04(SP) - PSL WITH WHICH TO RE-EXECUTE THE CHMK INSTRUCTION
00BB   ;
00BB   ;       ON PRIMARY'S INTERRUPT STACK:
00BB   ;
00BB   ;         IF ENTERED FROM IPL 5 INTERRUPT HANDLER:
00BB   ;
00BB   ;               00(SP) - SAVED R0 (AT TIME OF INTERRUPT)
00BB   ;               04(SP) - SAVED R1 (AT TIME OF INTERRUPT)
00BB   ;               08(SP) - SAVED R2 (AT TIME OF INTERRUPT)
00BB   ;               0C(SP) - SAVED R3 (AT TIME OF INTERRUPT)
00BB   ;               10(SP) - SAVED R4 (AT TIME OF INTERRUPT)
00BB   ;               14(SP) - SAVED R5 (AT TIME OF INTERRUPT)
00BB   ;               18(SP) - PC AT TIME OF IPL 5 RESCHEDULE INTERRUPT
00BB   ;               1C(SP) - PSL AT TIME OF IPL 5 RESCHEDULE INTERRUPT
00BB   ;
```

```
                               00BB   ;          IF ENTERED FROM IPL 3 INTERRUPT HANDLER:
                               00BB   ;
                               00BB   ;                  00(SP) - PC AT TIME OF IPL 3 RESCHEDULE INTERRUPT
                               00BB   ;                  04(SP) - PSL AT TIME OF IPL 3 RESCHEDULE INTERRUPT
                               00BB   ;
                               00BB   ;         IPL=SYNCH.
                               00BB   ;
                               00BB   ; OUTPUTS:
                               00BB   ;
                               00BB   ;         PROCESS IS EITHER RETURNED TO THE SECONDARY TO CONTINUE EXECUTION
                               00BB   ;         OR THIS PROCESS IS PLACED INTO A WAIT STATE AND ANOTHER PROCESS
                               00BB   ;         IS SCHEDULED FOR THE SECONDARY.
                               00BB   ;
                               00BB   ;         R0-R4 MAY BE DESTROYED (IF RPTEVT AST IS DONE).
                               00BB   ;
                               00BB   ; ENVIRONMENT:
                               00BB   ;
                               00BB   ;         EXECUTES ON PRIMARY PROCESSOR, MODE=KERNEL, IPL=SYNCH.
                               00BB   ;
                               00BB   ;--
                               00BB
                               00BB  MPS$WAITCK::
     54   0000'CF        D0    00BB          MOVL     W^MPS$GL_CURPCB,R4        ;GET PCB OF CURRENT PROCESS ON SECONDARY
       55    6C A4       D0    00C0          MOVL     PCB$L_PHD(R4),R5          ;GET ADDRESS OF HARDWARE PCB
     52    0090 C5       D0    00C4          MOVL     PHD$L_R2(R5),R2           ;GET ADR OF WAIT QUEUE HEADER
     53    0094 C5       D0    00C9          MOVL     PHD$L_R3(R5),R3           ;GET ADR OF EVENT FLAG VECTOR IN CEB
  51    63    4C A4      CB    00CE          BICL3    PCB$L_EFWM(R4),(R3),R1    ;TEST WAIT MASK
                 14      13    00D3          BEQL     30$                       ;BR IF WAIT NOT SATISFIED
   06 24 A4      0D      E0    00D5          BBS      #PCB$V_WALL,PCB$L_STS(R4),20$ ;BR IF WAITING FOR ANY FLAGS
                               00DA   ;
                               00DA   ; RETURN PROCESS TO SECONDARY AS WAIT CONDITION IS SATISFIED.
                               00DA   ;
                               00DA  10$:
     0000'CF     03      D0    00DA          MOVL     #MPS$K_BUSYSTATE,W^MPS$GL_STATE ;START SECONDARY EXECUTING PROC
                               00DF
                               00DF
                         05    00DF          RSB                                ;RETURN TO RESCHEDULING LOGIC TO
                               00E0                                             ; FINISH EITHER RESTORING REGISTERS
                               00E0                                             ; (IPL 5) OR SCHED PRIMARY (IPL 3)
                               00E0
                               00E0   ;
                               00E0   ; CHECK IF ANY OF THE FLAGS WERE SET -- $WFLOR REQUEST.
                               00E0   ;
          51     51      D2    00E0  20$:    MCOML    R1,R1                     ;INVERT MASKED FLAGS
       51    4C A4       D1    00E3          CMPL     PCB$L_EFWM(R4),R1         ;CHECK FOR 'AND' OF ALL FLAGS
                 F1      13    00E7          BEQL     10$                       ;BR IF WAIT SATISFIED
                               00E9   ;      BRB      30$                       ;CONTINUE IF WAIT NOT SATISFIED
                               00E9
                               00E9   ;
                               00E9   ; PLACE PROCESS IN A WAIT STATE AS WAIT CONDITION IS NOT MET, AND
                               00E9   ; RESCHEDULE ANOTHER PROCESS FOR SECONDARY.
                               00E9   ;
                               00E9  30$:
                               00E9   ;      BRW      SECWAIT                   ;CONTINUE AT THE EQUIVALENT OF SCH$WAIT
```

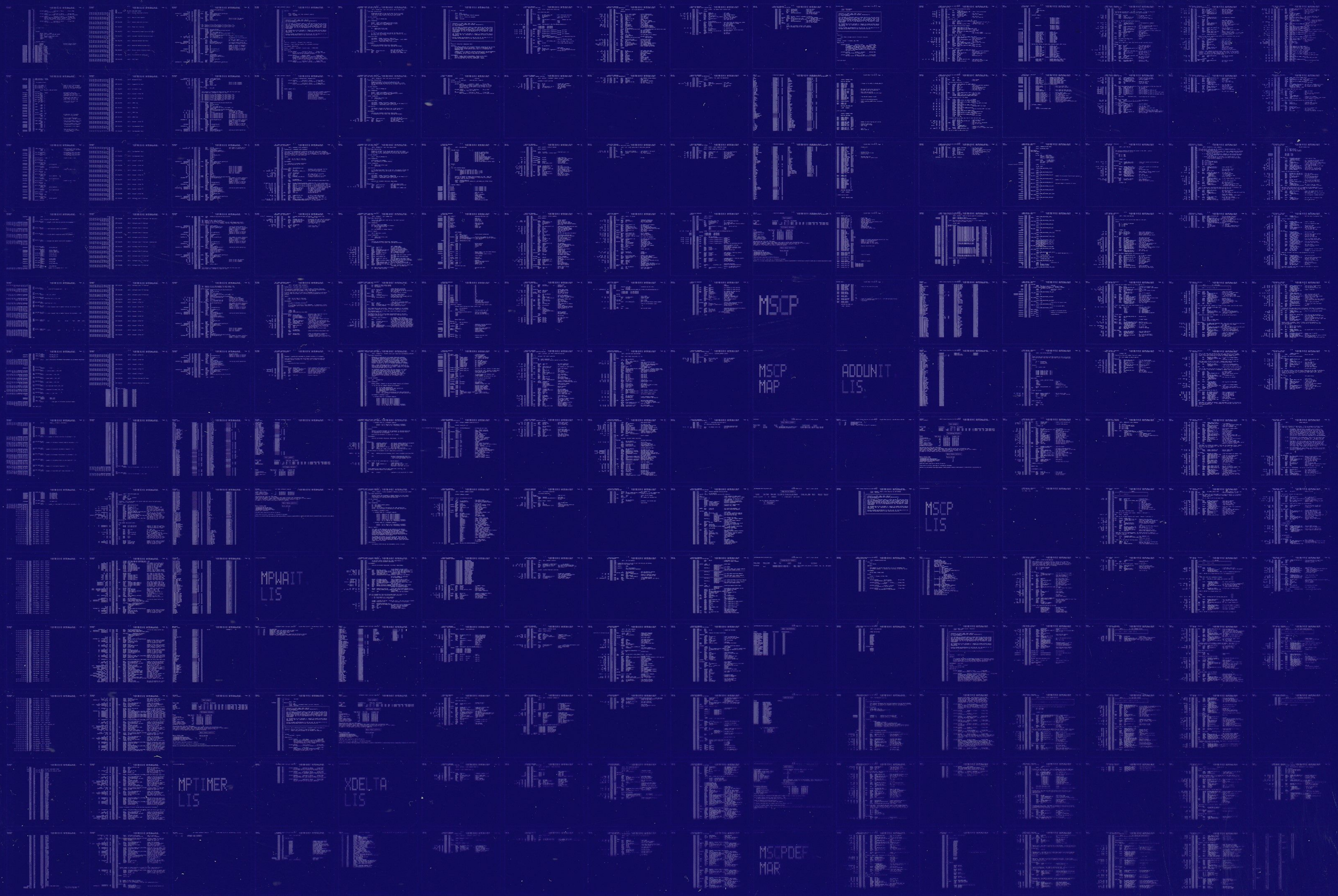
```
00E9          .SBTTL  SECWAIT - PLACES SECONDARY'S PROCESS IN SELECTED WAIT QUEUE
00E9    ;++
00E9    ; FUNCTIONAL DESCRIPTION:
00E9    ;
00E9    ;       SECWAIT PLACES THE SECONDARY'S CURRENT PROCESS IN A WAIT QUEUE
00E9    ;       SELECTED BY A WAIT QUEUE HEADER ADDRESS SUPPLIED IN A REGISTER
00E9    ;       A NEW PROCESS IS THEN SELECTED FOR EXECUTION.
00E9    ;
00E9    ; CALLING SEQUENCE:
00E9    ;
00E9    ;       JMP     SECWAIT
00E9    ;
00E9    ; INPUT PARAMETERS:
00E9    ;
00E9    ;       R2 - WAIT QUEUE HEADER ADDRESS
00E9    ;       R4 - PCB ADDRESS
00E9    ;       R5 - PHD ADDRESS
00E9    ;       00(SP) - PC AT WHICH TO RE-EXECUTE THE CHMK INSTRUCTION
00E9    ;       04(SP) - PSL WITH WHICH TO RE-EXECUTE THE CHMK INSTRUCTION
00E9    ;
00E9    ;       ON PRIMARY'S INTERRUPT STACK:
00E9    ;
00E9    ;         IF ENTERED FROM IPL 5 INTERRUPT HANDLER:
00E9    ;
00E9    ;               00(SP) - SAVED R0 (AT TIME OF INTERRUPT)
00E9    ;               04(SP) - SAVED R1 (AT TIME OF INTERRUPT)
00E9    ;               08(SP) - SAVED R2 (AT TIME OF INTERRUPT)
00E9    ;               0C(SP) - SAVED R3 (AT TIME OF INTERRUPT)
00E9    ;               10(SP) - SAVED R4 (AT TIME OF INTERRUPT)
00E9    ;               14(SP) - SAVED R5 (AT TIME OF INTERRUPT)
00E9    ;               18(SP) - PC AT TIME OF IPL 5 RESCHEDULE INTERRUPT
00E9    ;               1C(SP) - PSL AT TIME OF IPL 5 RESCHEDULE INTERRUPT
00E9    ;
00E9    ;         IF ENTERED FROM IPL 3 INTERRUPT HANDLER:
00E9    ;
00E9    ;               00(SP) - PC AT TIME OF IPL 3 RESCHEDULE INTERRUPT
00E9    ;               04(SP) - PSL AT TIME OF IPL 3 RESCHEDULE INTERRUPT
00E9    ;
00E9    ; IMPLICIT INPUTS:
00E9    ;
00E9    ;       THIS CODE IS NOT EXECUTING IN THE CONTEXT OF THE PROCESS
00E9    ;       (AS DOES DOES ITS PRIMARY PROCESSOR COUNTERPART - SCH$WAIT).
00E9    ;       THEREFORE, THE STACKS CANNOT BE ACCESSED FROM THIS CODE, ONLY
00E9    ;       THINGS CONTAINED IN THE HARDWARE PCB FOR THIS PROCESS.
00E9    ;       EVERYTHING NEEDED FOR RESCHEDULING THIS PROCESS HAS BEEN
00E9    ;       PLACED IN THE HARDWARE PCB BY THE SECONDARY PROCESSOR BEFORE
00E9    ;       PASSING THIS PROCESS TO THE PRIMARY.
00E9    ;
00E9    ;       THE PRIMARY HAS ALREADY SET R2-R5 TO BE THE VALUES FOR THE
00E9    ;       CURRENT PROCESS ON THE SECONDARY PROCESSOR.  THE PC/PSL FOR
00E9    ;       THE INSTRUCTION FOR RE-EXECUTING THE CHMK INSTRUCTION ARE ON
00E9    ;       THE STACK.  THEY ARE ALSO CONTAINED IN R0/R1, SINCE THE PRIMARY
00E9    ;       CANNOT ACCESS THE STACK.
00E9    ;
00E9    ; SIDE EFFECTS:
00E9    ;
00E9    ;       THE PROCESS SPECIFIED BY THE PCB ADDRESS IN R4 IS PLACED
```

```
                                 00E9   646 ;       IN THE WAIT QUEUE LOCATED BY R2, AND A NEW PROCESS IS
                                 00E9   647 ;       SCHEDULED FOR THE SECONDARY PROCESSOR.
                                 00E9   648 ;
                                 00E9   649 ; ENVIRONMENT:
                                 00E9   650 ;
                                 00E9   651 ;       EXECUTES ON PRIMARY PROCESSOR, IPL=SYNCH, MODE=KERNEL.
                                 00E9   653 ;
                                 00E9   654 ;--
                                 00E9   655
                                 00E9   659 SECWAIT:                                ;PLACE PROCESS IN WAIT STATE
            78 A5    08     C0   00E9   666         ADDL    #8,PHD$L_KSP(R5)        ;CLEAN PC/PSL OF CHMK INSTR OFF STACK
    00C0 C5   0088 C5     D0   00ED   667         MOVL    PHD$L_R0(R5),PHD$L_PC(R5) ;SET PC TO BE CHMK INSTRUCTION
    00C4 C5   008C C5     D0   00F4   668         MOVL    PHD$L_R1(R5),PHD$L_PSL(R5) ;SET PSL BACK FOR CHMK INSTRUCTION
                 08 A2    B6   00FB   670         INCW    WQH$W_WQCNT(R2)         ;INCREMENT COUNT FOR QUEUE
              62    64    0E   00FE   671         INSQUE  (R4),@QH$L_WQFL(R2)     ;INSERT IN QUEUE
        2C A4    0A A2    B0   0101   672         MOVW    WQH$W_WQSTATE(R2),PCB$W_STATE(R4) ;SET STATE FOR PROCESS
                                 0106   673 ;
                                 0106   674 ;                                       THE STATE NUMBER IS CONTAINED
                                 0106   675 ;                                       IN THE QUEUE HEADER
                                 0106   676 ;
       50     00000000'GF    3C   0106   683         MOVZWL  G^SCH$GW_IOTA,R0        ;Get a longword copy of IOTA
             3C A5    50    A0   010D   684         ADDW    R0,PHD$W_QUANT(R5)      ;CHARGE QUOTA FOR VOLUNTARY WAIT
           0100 C5    50    C2   0111   685         SUBL2   R0,PHD$L_TIMREF(R5)     ;Adjust AWSA time reference
  0118 C4     00000000'GF    D0   0116   686         MOVL    G^EXE$GL_ABSTIM,PCB$L_WAITIME(R4) ;RECORD TIME AT WAIT START
           00CF C5    04    91   011F   687         CMPB    #4,PHD$B_ASTLVL(R5)     ;NULL ASTLVL?
                      0C    12   0124   688         BNEQ    20$                     ;NO, DO LONG CHECK
       00 0000'CF     00    E6   0126   692 10$:    BBSSI   #LCK$V_INTERLOCK,W^MPS$GL_INTERLOCK,15$ ;FLUSH CACHE QUEUE
          0000'CF     01    D0   012C   693 15$:    MOVL    #MPS$K_IDLESTATE,W^MPS$GL_STATE ;SET SECONDARY IDLE
                            05   0131   694         RSB                             ;RETURN TO SCHEDULE SECONDARY
                                 0132   696
            50     00CF C5    9A   0132   697 20$:    MOVZBL  PHD$B_ASTLVL(R5),R0     ;FETCH AND ZERO EXTEND PENDING ASTLVL
   50    00C4 C5    02    18    ED   0137   698         CMPZV   #PSL$V_CURMOD,#PSL$S_CURMOD,PHD$L_PSL(R5),R0 ;COMPARE WITH WAIT
                                 013E   699                                         ;ACCESS MODE
                      E6    19   013E   700         BLSS    10$                     ;BRANCH IF AST NOT DELIVERABLE
                                 0140   701
                                 0140   702 ;
                                 0140   703 ;  Test for assumptions that are being made about the layout of the
                                 0140   704 ;  PSL that enables the next instruction to work correctly.
                                 0140   705 ;
                                 0140   706 ;       o  IPL field begins on a byte boundary
                                 0140   707 ;       o  IPL field fits into a single byte
                                 0140   708
                                 0140   709
                                 0140   710         ASSUME  <<<PSL$V_IPL/8>*8> - PSL$V_IPL> EQ 0 ; IPL must be byte aligned
                                 0140   711         ASSUME  PSL$S_IPL LE 8          ; IPL field must fit into byte
                                 0140   712
                      1F    93   0140   713         BITB    #<PSL$M_IPL@-PSL$V_IPL>,-
                 00C6 C5         0142   714                 <PSL$V_IPL/8>+PHD$L_PSL(R5) ;MUST BE AT IPL 0 FOR DELIVERY
                      DF    12   0145   715         BNEQ    10$                     ;BRANCH IF AST NOT DELIVERABLE
                      52    D4   0147   716         CLRL    R2                      ;SET NULL PRIORITY INCREMENT
                                 0149   720         RPTEVT  AST,JSB                 ;REPORT AST EVENT
                      D4    11   0150   722         BRB     10$                     ;GO SCHEDULE NEXT PROCESS
                                 0152   724
                                 0152   725         .END
```

```
                                                        K 5
MPWAIT                     - SECONDARY EVENT FLAG WAIT SERVICES     16-SEP-1984 02:09:57  VAX/VMS Macro V04-00       Page  13      XD
Symbol table                                                         5-SEP-1984 03:58:01  [SYS.SRC]SYSWAIT.MAR;1          (1)    V0

CEB$B_TYPE                 = 0000000A                SCH$RSE                  ********   X    02
CEB$L_EFC                  = 00000010                SCH$WAIT                 ********   X    02
CEB$L_MASTER               = 00000040                SECWAIT                  000000E9 R      02
CEB$L_WQFL                 = 00000014                SS$_ILLEFC             = 000000EC
DYN$C_SLAVCEB              = 0000002D                SS$_NORMAL             = 00000001
EFN                        = 00000004                SS$_UNASEFC            = 00000234
EVT$_AST                     ********   X    02      WAIT                     00000078 R      02
EXE$C_CMSTKSZ                ********   X    02      WAITALL                  00000073 R      02
EXE$GQ_ABSTIM                ********   X    02      WAITCK                   0000005D R      02
EXE$GL_RPB                   ********   X    02      WFRL                     0000000B R      02
IPL$_SYNCH                 = 00000008                WQH$L_WQFL             = 00000000
LCK$V_INTERLOCK            = 00000000                WQH$W_WQCNT            = 00000008
MASK                       = 00000008                WQH$W_WQSTATE          = 0000000A
MPS$GL_CURPCB                ********   X    02
MPS$GL_INTERLOCK             ********   X    02
MPS$GL_SECREQFLG             ********   X    02
MPS$GL_STATE                 ********   X    02
MPS$K_BUSYSTATE            = 00000003
MPS$K_IDLESTATE            = 00000001
MPS$MPSCHED2                 ********   X    02
MPS$V_SECWAITCK            = 00000002
MPS$WAIT                     0000001A R      02
MPS$WAITCK                   000000BB RG     02
MPS$WAITCONT                 0000006F RG     02
MPS$WAITFR                   00000011 RG     02
MPS$WFLAND                   00000000 RG     02
MPS$WFLOR                    00000007 RG     02
MPSWITCH                   = 00000001
NOWAIT                       00000068 R      02
PCB$B_WEFC                 = 0000002E
PCB$L_EFCS                 = 00000050
PCB$L_EFWM                 = 0000004C
PCB$L_PHD                  = 0000006C
PCB$L_STS                  = 00000024
PCB$L_WAITIME              = 00000118
PCB$V_WALL                 = 0000000D
PCB$W_STATE                = 0000002C
PHD$B_ASTLVL               = 000000CF
PHD$L_KSP                  = 00000078
PHD$L_PC                   = 000000C0
PHD$L_PSL                  = 000000C4
PHD$L_R0                   = 00000088
PHD$L_R1                   = 0000008C
PHD$L_R2                   = 00000090
PHD$L_R3                   = 00000094
PHD$L_TIMREF               = 00000100
PHD$W_QUANT                = 0000003C
PR$_IPL                    = 00000012
PR$_SCBB                   = 00000011
PSL$M_IPL                  = 001F0000
PSL$S_CURMOD               = 00000002
PSL$S_IPL                  = 00000005
PSL$V_CURMOD               = 00000018
PSL$V_IPL                  = 00000010
RPB$L_SCBB                 = 000000B0
SCH$GQ_LEFWQ                 ********   X    02
SCH$GW_IOTA                  ********   X    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                    00000152  (    338.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC BYTE

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

Phase                        Page faults    CPU Time       Elapsed Time
-----                        -----------    --------       ------------
Initialization                       33     00:00:00.09    00:00:01.55
Command processing                  112     00:00:01.26    00:00:09.20
Pass 1                              385     00:00:12.41    00:00:40.99
Symbol table sort                     0     00:00:01.86    00:00:02.82
Pass 2                              111     00:00:02.76    00:00:08.62
Symbol table output                   9     00:00:00.10    00:00:00.85
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                654     00:00:18.51    00:01:04.06

The working set limit was 1650 pages.
72959 bytes (143 pages) of virtual memory were used to buffer the intermediate code.
There were 70 pages of symbol table space allocated to hold 1210 non-local and 12 local symbols.
731 source lines were read in Pass 1, producing 14 object records in Pass 2.
27 pages of virtual memory were used to define 26 macros.

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

Macro library name                           Macros defined
------------------                           --------------
_$255$DUA28:[MP.OBJ]MP.MLB;1                             5
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                          12
_$255$DUA28:[SYSLIB]STARLET.MLB;2                        7
TOTALS (all libraries)                                  24

1388 GETS were required to define 24 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:MPWAIT/OBJ=OBJ$:MPWAIT MSRC$:MPPREFIX/UPDATE=(ENH$:MPPREFIX)+MSRC$:MPSWT/UPDATE=(ENH$:MPSWT)+MASD$:[SYS.SRC]SYSWAIT/U
```

0249 AH-BT13A-SE DIGITAL EQUIPMENT CORPORATION
VAX/VMS V4.0 CONFIDENTIAL AND PROPRIETARY


MSCP

MSCP
MAP

ADDUNIT
LIS

MSCP
LIS

MPWAIT
LIS

MPTIMER
LIS

XDELTA
LIS

MSCPDEF
MAR