

[illegible]

```
MM      MM  PPPPPPP  WW      WW  AAAAAA  IIIIII  TTTT?TTTTT
MM      MM  PPPPPPP  WW      WW  AAAAAA  IIIIII  TTTT?TTTTT
MMM     MMM  PP      PP  WW      WW  AA      AA  II      TT
MMM     MMM  PP      PP  WW      WW  AA      AA  II      TT
MM      MM  PP      PP  WW      WW  AA      AA  II      TT
MM      MM  PPPPPPP  WW      WW  AA      AA  II      TT
MM      MM  PPPPPPP  WW      WW  AA      AA  II      TT
MM      MM  PP      WW  WW  AAAAAAAAAA  II      TT
MM      MM  PP      WW  WW  AAAAAAAAAA  II      TT
MM      MM  PP      WWW  WWW  AA      AA  II      TT
MM      MM  PP      WWW  WWW  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
```

(1)	83
(1)	116
(1)	182
(1)	252
(1)	321
(1)	445
(1)	565

DECLARATIONS

MPSSWFLAND - WAIT FOR LOGICAL AND OF EVENT FLAGS

MPSS\$WFLOR - WAIT FOR LOGICAL OR OF EVENTS

MPSS\$WAITFR - WAIT FOR SINGLE EVENT

MPSS\$WAIT - WAIT COMMON CODE

MPSSWAITCK - PRIMARY CHECK WAIT CONDITION FOR POTENTIAL RESCHED

SECWAIT - PLACES SECONDARY'S PROCESS IN SELECTED WAIT QUEUE

MPV
Syn CEE
CEE
CEE
CEE
DYN
EFM
EVN
EXE
EXE
EXE
IPL
LCI
MAS
MPS
MPS
MPS
MPS
MPS
MPS
MPS
MPS
MPS
MPS
MPS
MPS
MPS
MPS
NOW
PCE
PCE
PCE
PCE
PCE
PCE
PCE
PHD
PHD
PHD
PHD
PHD
PHD
PHD
PHD
PRJ
PRJ
PSL
PSL
PSL
PSL
RPE
SCH
SCH


```
00000001 0000 1 :
0000 2 : Version: 'V04-000'
0000 3 :
0000 4 :
0000 5 : .MCALL MFPR
0000 6 : MPSWITCH = 1
0000 7 : .NLIST CND
0000 8 : .TITLE MPWAIT - SECONDARY EVENT FLAG WAIT SERVICES
0000 9 : .IDENT 'V04-000'
0000 10 : *****
0000 11 : *
0000 12 : * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 13 : * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 14 : * ALL RIGHTS RESERVED.
0000 15 : *
0000 16 : * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 17 : * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 18 : * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 19 : * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 20 : * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 21 : * TRANSFERRED.
0000 22 : *
0000 23 : * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 24 : * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 25 : * CORPORATION.
0000 26 : *
0000 27 : * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 28 : * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 29 : *
0000 30 : *****
0000 31 : ++
0000 32 : FACILITY: MULTI-PROCESSING EXECUTIVE, SECONDARY EVENT FLAG SERVICES
0000 33 :
0000 34 : ABSTRACT: $WAITFR THAT SUCCEEDS IS DONE ON SECONDARY. ALL
0000 35 : OTHER CASES ARE RETURNED TO THE PRIMARY FOR HANDLING.
0000 36 :
0000 37 : --
0000 38 :
0000 39 : AUTHOR:
0000 40 : R.HUSTVEDT : VERSION
0000 41 :
0000 42 : MODIFIED BY:
0000 43 :
0000 44 :
0000 45 :
0000 46 : V03-007 SSA0016 Stan Amway 8-Mar-1984
0000 47 : Subtract IOTA from automatic working set adjustment
0000 48 : time reference in PHD on any entry to SCH$WAITx code.
0000 49 : (Acknowledgements go to Wayne Cardoza and Larry Kenah,
0000 50 : who both collaborated on this change.)
0000 51 :
0000 52 : V03-006 WMC0001 Wayne Cardoza 22-Feb-1984
0000 53 : No reason to try to wake up swapper on every EFN wait.
0000 54 :
0000 55 : V03-005 SSA0003 Stan Amway 5-Dec-1983
0000 56 : Added support for outswap scheduling changes.
0000 57 : Changed process wait code to store wait time in PCB
0000 58 : as longword system absolute time.
0000 59 :
0000 60 :
0000 61 :
0000 62 :
0000 63 :
```


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

```
0000 83      .SBTTL  DECLARATIONS
0000 84
0000 85  :
0000 86  : INCLUDE FILES:
0000 87  :
0000 88
0000 89      $CEBDEF      ;COMMON EVENT BLOCK DEFS
0000 90      $DYNDEF      ;DYNAMIC STRUCTURE TYPES
0000 91      $IPLDEF      ;IPL DEFINITIONS
0000 93      $LCKDEF      ;INTERLOCK BIT DEFINITIONS
0000 94      $MPSDEF      ;SECONDARY REQUEST FLAG DEFS
0000 95      $RPBDEF      ;REBOOT PARAMETER BLOCK DEFS
0000 97      $PCBDEF      ;PCB DEFINITIONS
0000 98      $PHDDEF      ;PHD DEFINITIONS
0000 99      $PRDEF       ;PROCESSOR REGISTER DEFS
0000 100     $PSLDEF      ;PSL DEFINITIONS
0000 101     $$HBDEF      ;SHARED MEMORY CONTROL BLK DEFS
0000 102     $$HDDEF      ;SHARED MEMORY COMMON DATA PAGE
0000 103     $$$DEF       ;STATUS CODE DEFINITIONS
0000 104     $$STATEDEF   ;STATE DEFINITIONS
0000 105     $WQHDEF      ;WAIT QUEUE HEADER DEFS
0000 106  :
0000 107  : EQUATES:
0000 108  :
00000004 0000 109 EFN=4      ;EVENT FLAG NUMBER
00000008 0000 110 MASK=8    ;WAIT MASK
00000000 111      .PSECT  AEXENONPAGED ;NON-PAGED
```

```

0000 116 .SBTTL MPSS$WFLAND - WAIT FOR LOGICAL AND OF EVENT FLAGS
0000 118 :++
0000 119 : FUNCTIONAL DESCRIPTION:
0000 120 :
0000 124 : MPSS$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,MPSS$WFLAND
0000 136 :
0000 137 : INPUT PARAMETERS:
0000 138 :
0000 139 : 04(AP) - EVENT FLAG NUMBER SELECTING CLUSTER
0000 140 : 08(AP) - MASK SELECTING COMBINATION OF EVENTS
0000 141 : R4 - PCB ADDRESS OF CURRENT PROCESS
0000 142 :
0000 143 : OUTPUT PARAMETERS:
0000 144 :
0000 145 : R0 - COMPLETION STATUS CODE
0000 146 : CONDITION IS SATISFIED.
0000 147 :
0000 148 : SIDE EFFECTS:
0000 149 :
0000 150 : IF THE SET OF EVENT FLAGS SELECTED BY THE MASK ARE NOT
0000 151 : ALL SET, THEN THE PROCESS ISSUING THE WAIT SERVICE CALL WILL
0000 152 : BE PLACED IN A WAIT STATE.
0000 153 :
0000 154 : COMPLETION CODES:
0000 155 :
0000 156 : SS$_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 MPSS$WFLAND::
0000 175 .WORD *M<R2,R3,R4,R5> ;WAIT FOR LOGICAL AND
0002 176 MOVL #1,R1 ;REGISTER SAVE MASK FOR R2-R5
0005 177 BRB WFL ;SET MODE TO WAITALL
;AND MERGE WITH COMMON CODE

```



```
0007 182 .SBTTL MPSS$WFLO - WAIT FOR LOGICAL OR OF EVENTS
0007 184 :++
0007 185 : FUNCTIONAL DESCRIPTION:
0007 186 :
0007 190 : MPSS$WFLO RETURNS TO THE CALLER WHEN ANY OF THE
0007 192 : EVENTS SELECTED BY THE MASK WITHIN THE SPECIFIED CLUSTER
0007 193 : ARE SET AND RETURNS THE STATE OF ALL 32 EVENT FLAGS IN THE
0007 194 : CLUSTER.
0007 195 :
0007 196 : CALLING SEQUENCE:
0007 197 :
0007 201 : CALLG ARGLIST,MPSS$WFLO
0007 203 :
0007 204 : INPUT PARAMETERS:
0007 205 :
0007 206 : 04(AP) - EVENT FLAG NUMBER TO SELECT CLUSTER
0007 207 : 08(AP) - MASK SELECTING DESIRED COMBINATION OF EVENTS
0007 208 : R4 - PCB ADDRESS OF CURRENT PROCESS
0007 209 :
0007 210 : OUTPUT PARAMETERS:
0007 211 :
0007 212 : R0 - COMPLETION STATUS CODE
0007 213 : IS SATISFIED.
0007 214 :
0007 215 : COMPLETION CODES:
0007 216 :
0007 217 : SS$_NORMAL - NORMAL SUCCESSFUL COMPLETION
0007 218 : SS$_ILLEFC - ILLEGAL EVENT FLAG NUMBER NOT IN THE RANGE 0-127.
0007 219 : SS$_UNASEFC - UNASSIGNED EVENT FLAG CLUSTER.
0007 220 :
0007 221 : SIDE EFFECTS:
0007 222 :
0007 223 : THE PROCESS ISSUING THE SERVICE CALL IS BE PLACED IN A
0007 224 : WAIT STATE IF NONE OF THE SPECIFIED EVENTS ARE SET.
0007 226 :
0007 227 : ENVIRONMENT:
0007 228 :
0007 229 : EXECUTES ON SECONDARY PROCESSOR, MODE=KERNEL.
0007 230 : IF INTERRUPTED AT ANY POINT, MAY CONTINUE ON PRIMARY.
0007 232 :
0007 233 : --
0007 234 :
0007 238 MPSS$WFLO::
0007 240 .WORD ^M<R2,R3,R4,R5>
0009 241 CLRL R1
000B 242 WFL: MOVL MASK(AP),R0
000F 246 BRB MPSS$WAIT
0007 247 :WAIT FOR LOGICAL OR
0007 248 :REGISTER SAVE MASK FOR R2-R5
0007 249 :SET MODE TO WAIT ANY
0007 250 :GET WAIT MASK
0007 251 :MERGE WITH COMMON CODE
```

50 08 AC 09 003C D4 D0 11

```
0011 252 .SBTTL MPSS$WAITFR - WAIT FOR SINGLE EVENT
0011 254 :++
0011 255 : FUNCTIONAL DESCRIPTION:
0011 256 :
0011 260 : MPSS$WAITFR RETURNS TO THE CALLER WHEN THE SPECIFIED SINGLE
0011 262 : EVENT FLAG IS SET. UPON RETURN THE STATE OF ALL 32 EVENT FLAGS
0011 263 : WITHIN THE CLUSTER CONTAINING THE SPECIFIED EVENT ARE RETURN.
0011 264 :
0011 265 : CALLING SEQUENCE:
0011 266 :
0011 270 : CALLG ARGLIST,MPSS$WAITFR
0011 272 :
0011 273 : INPUT PARAMETERS:
0011 274 :
0011 275 : 04(AP)=EVENT FLAG NUMBER
0011 276 : R4 - PCB ADDRESS OF CURRENT PROCESS
0011 277 :
0011 278 : OUTPUT PARAMETERS:
0011 279 :
0011 280 : R0 - COMPLETION STATUS CODE
0011 281 : SATISFIED.
0011 282 :
0011 283 : SIDE EFFECTS:
0011 284 :
0011 285 : IF THE SPECIFIED EVENT FLAG IS NOT SET, THE PROCESS ISSUING THE
0011 286 : WAIT SYSTEM SERVICE WILL BE PLACED IN THE APPROPRIATE WAIT
0011 287 : STATE.
0011 288 :
0011 289 : COMPLETION CODES:
0011 290 :
0011 291 : $$$_NORMAL - NORMAL SUCCESSFUL COMPLETION
0011 292 : $$$_ILLEFC - ILLEGAL EVENT FLAG NUMBER NOT IN THE RANGE 0-127.
0011 293 : $$$_UNASEFC - UNASSIGNED EVENT FLAG CLUSTER.
0011 294 :
0011 295 : ENVIRONMENT:
0011 296 :
0011 300 : EXECUTES ON SECONDARY PROCESSOR, MODE=KERNEL.
0011 302 :--
0011 303 :
0011 307 MPSS$WAITFR::
0011 309 .WORD ^M<R2,R3,R4,R5> ;WAIT FOR SINGLE EVENT
0013 310 CLRL R1 ;SAVE REGISTERS R2,R3,R4,R5
0015 311 ROTL EFN(AP),#1,R0 ;SET MODE
001A 315 ; BRB MPSS$WAIT ;INIT MASK
;AND MERGE WITH COMMON CODE
```

50 01 04 AC 9C 003C 51 D4


```
001A 321 .SBTTL MPSS$WAIT - WAIT COMMON CODE
001A 322 :++
001A 323 : FUNCTIONAL DESCRIPTION:
001A 324 :
001A 325 : THIS IS THE COMMON WAIT CODE FOR ALL THE EVENT FLAG WAIT
001A 326 : SYSTEM SERVICES.
001A 327 :
001A 328 : INPUT PARAMETERS:
001A 329 :
001A 330 : 04(AP) = EVENT FLAG NUMBER
001A 331 : R0 = MASK SELECTING EVENTS OF INTEREST
001A 332 : R1 = ANY/ALL MODE SELECTOR
001A 333 : 0 => ANY
001A 334 : 1 => ALL
001A 335 :
001A 336 : R4 = PCB ADDRESS OF CURRENT PROCESS
001A 337 :
001A 338 : IMPLICIT INPUTS:
001A 339 :
001A 340 : CEB IF NON-LOCAL CLUSTER.
001A 341 :
001A 342 : OUTPUT PARAMETERS:
001A 343 :
001A 344 : R0 - COMPLETION STATUS CODE
001A 345 : SATISFIED.
001A 346 :
001A 347 : ENVIRONMENT:
001A 348 :
001A 349 : EXECUTES ON SECONDARY PROCESSOR, MODE=KERNEL.
001A 350 : IF INTERRUPTED AT ANY POINT, MAY CONTINUE ON PRIMARY.
001A 351 :
001A 352 : --
001A 353 :
001A 354 :
001A 355 :
001A 359 MPSS$WAIT:
52 04 AC 98 001A 361 CVTBL EFN(AP),R2 ;WAIT COMMON CODE
52 52 FB 1A 19 001E 362 BLSS 10$ ;GET CLUSTER NUMBER
53 50 A4 78 0020 363 ASHL #-5,R2,R2 ;ILLEGAL IF NOT (0,1,2,3)
2E A4 52 DE 0025 364 MOVAL PCB$EFC(R4)[R2],R3 ;RIGHT ALIGN CLUSTER NUMBER
15 52 90 002A 365 MOVB R2,PCB$B_WFC(R4) ;POINTER TO PCB EVENT CLUSTER
52 00000000 GF 7E 0031 366 SOBGTR R2,30$ ;SAVE WAIT CLUSTER NUMBER
23 11 0038 367 MOVAQ G$SCH$GQ_LEFWQ,R2 ;BR IF COMMON CLUSTER R2 = (2,3)
50 00EC 8F 3C 003A 368 BRB WAITCK ;SET WAIT QUEUE POINTER
04 003F 369 10$: MOVZWL #SS$_ILLEFC,R0 ;SET ERROR CODE FOR ILLEGAL CLUSTER
50 0234 8F 3C 0040 370 RET ;AND EXIT
04 0045 371 20$: MOVZWL #SS$_UNASEFC,R0 ;SET ERROR CODE FOR UNASSIGNED
52 63 10 C1 0046 372 RET ;AND EXIT
FA A2 2D 91 004A 373 30$: ADDL3 #CEB$_EFC,(R3),R2 ;GET CEB ADDRESS FOR EVENT FLAGS
08 12 004C 374 BGEQ 20$ ;CEB ASSIGNED (SYSTEM SPACE ADDRESS)
53 30 A2 DO 0050 375 CMPB #DYN$_SLAVCEB,<CEB$_TYPE-CEB$_EFC>(R2) ;IS THIS IN SH MEM?
62 10 A3 DO 0052 376 BNEQ 40$ ;BR IF IN LOCAL MEMORY
53 82 DE 005A 377 MOVL <CEB$_MASTER-CEB$_EFC>(R2),R3 ;GET ADR OF MASTER CEB
005D 378 MOVL CEB$_EFC(R3),(R2) ;COPY EFC FROM MASTER TO SLAVE CEB
005D 379 ASSUME <CEB$_EFC+4> EQ CEB$_WQFL
005D 380 40$: MOVAL (R2)+,R3 ;GET EVENT POINTER AND WAIT QUEUE ADDR
005D 381 ; R3=CEB$_EFC, R2=CEB$_WQFL
005D 382 :
005D 383 :
005D 384 :
R0 - MASK SELECTING EVENTS OF INTEREST
R1 - ANY/ALL MODE SELECTOR
```



```
005D 385 : R2 - ADDRESS OF WAIT QUEUE HEADER
005D 386 : R3 - ADDRESS OF EVENT FLAG VECTOR
005D 387 : R4 - PCB ADDRESS
005D 388 :
005D 389 WAITCK: ;CHECK FOR WAIT SATISFIED
50 63 D3 005D 390 SETIPL #IPL$ SYNCH ;BLOCK SCHEDULING ACTIVITY
13 13 13 0060 391 BITL (R3),R0 ;WAIT FOR LOGICAL OR MAY BE SATISFIED
OB 51 E8 0063 392 BEQL WAIT ;NO, MUST WAIT
0065 393 BLBS R1,WAITALL ; 1 => WAIT FOR ALL IN MASK
0068 394 NOWAIT: ;
5D OC AD D0 0068 395 MOVL 12(FP),FP ;GET SAVED FRAME POINTER
5E 00' C0 006C 396 ADDL S^#EXESC_CMSTKSZ,SP ;CLEAN STACK TO PC,PSL
50 01 9A 006F 398 MPSS$WAITCONT: ;CONTINUE HERE AFTER PRIMARY WAITCHK
006F 400 MOVZBL #SS$_NORMAL,R0 ;RETURN SUCCESS CODE
0072 402
0072 406
02 0072 408 REI ;RETURN TO CALLER
0073 409
0073 410 WAITALL: ; WAIT FOR ALL SELECTED EVENTS
50 63 CA 0073 411 BICL2 (R3),R0 ; CLEAR BITS FOR ALREADY SET FLAGS
FO 13 0076 415 BEQL NOWAIT ;YES, DONT WAIT
24 A4 01 OD 51 FO 0078 417 WAIT: INSV R1,#PCBSV_WALL,#1,PCBSL_STS(R4) ;SET WAIT ALL FLAG
4C A4 50 D2 007E 418 MCOML R0,PCBSL_EFWM(R4) ;SAVE INVERTED WAIT MASK
0082 420
0082 421 :
0082 422 : WAIT CONDITION IS NOT SATISFIED. PROCESS IS RETURNED TO THE PRIMARY
0082 423 : ALONG WITH A REQUEST BIT TO CHECK IF THE WAIT CONDITION IS SATISFIED.
0082 424 :
0082 425 : THE SECONDARY WAITS IN ITS IDLE LOOP, WHILE THE PRIMARY CONTINUES
0082 426 : EXECUTION AT MPSS$WAITCK.
0082 427 :
0082 428 : THE PROCESS WILL BE IN A STATE READY FOR RESCHEDULING IF THE WAIT
0082 429 : CONDITION IS NOT SATISFIED, AND THE PRIMARY WILL CONTINUE EXECUTION
0082 430 : AT SCH$WAIT IN THIS CASE.
0082 431 :
0082 432 IFPRIMARY <JMP G^SCH$WAIT> ;IF PRIMARY, THEN CONTINUE
009B 433 ;IF SECONDARY, ASK FOR PRIMARY ASSIST
5D OC AD D0 009B 434 MOVL 12(FP),FP ;PUT SAVED FRAME POINTER WHERE PRIMARY
5E 00' C0 009F 435 ADDL S^#EXESC_CMSTKSZ,SP ;CAN FIND IT AS IT CAN'T TOUCH STACK
50 6E 04 C3 00A2 437 SUBL3 #4,(SP),R0 ;CLEAN STACK TO PC/PSL OF CHMK INSTR
51 04 AE D0 00A6 438 MOVL 4(SP),R1 ;GET PS/PSL PAIR FOR CHMK INSTRUCTION
7E DC 00AA 439 MOVPSL -(SP) ; INTO A PLACE THE PRIMARY CAN TOUCH
0000006F'GF 9F 00AC 440 PUSHAB G^MPSS$WAITCONT ;PUT PC/PSL PAIR ON STACK FOR SECONDARY
00 0000'CF 02 E6 00B2 441 BBSSI #MPSSV_SECWAITCK,W^MPSS$GL SECREQFLG,10$ ;PROCESSOR TO CONTINUE EXECUTING PROC
FF45' 31 00B8 442 10$: BRW MPSS$MPSCHED2 ;IND WAIT CHECK REQUEST
00BB 443 ;GO WAIT IN IDLE LOOP FOR WORK TO DO
```

```
00BB 445 .SBTTL MPSS$WAITCK - PRIMARY CHECK WAIT CONDITION FOR POTENTIAL RESCHED
00BB 446 :++
00BB 447 : FUNCTIONAL DESCRIPTION:
00BB 448 :
00BB 449 : MPSS$WAITCK IS EXECUTED BY THE PRIMARY PROCESSOR WHENEVER
00BB 450 : THE SECONDARY PROCESSOR EXECUTES AN EVENT FLAG WAIT SYSTEM
00BB 451 : SERVICE THAT CAUSES A PROCESS TO ACTUALLY WAIT. THE PRIMARY
00BB 452 : PROCESSOR EXECUTES THIS CODE DUE TO A SECONDARY REQUEST FLAG,
00BB 453 : DISPATCHED FROM THE SECONDARY RESCHEDULING CODE. THIS IS
00BB 454 : ENTERED VIA THE IPL 5 (MULTI-PROCESSING SECONDARY RESCHEDULE
00BB 455 : REQUEST) INTERRUPT OR THE IPL 3 (PRIMARY RESCHEDULE) INTERRUPT.
00BB 456 :
00BB 457 : NOTE: A RACE CONDITION EXISTS IF THE PRIMARY IS IN THE IPL 3
00BB 458 : HANDLER WHEN THE SECONDARY SETS THE REQUEST FLAG, AND THUS THIS
00BB 459 : CODE MAY BE EXECUTED FROM THE IPL 3 LOGIC, THOUGH IT IS INTENDED
00BB 460 : TO BE EXECUTED FROM THE IPL 5 PATH.
00BB 461 :
00BB 462 : THE SECONDARY HAS DONE A SVPCTX FOR THIS PROCESS JUST PRIOR
00BB 463 : TO WHERE THE PRIMARY WOULD HAVE EXECUTED SCH$WAIT, IF THE
00BB 464 : SYSTEM SERVICE WAS BEING EXECUTED ON THE PRIMARY. THIS
00BB 465 : ALLOWS THE PRIMARY TO LOOK IN THE HARDWARE PCB FOR THE
00BB 466 : INFORMATION NEEDED TO CHECK THE EVENT FLAG WAIT CONDITION
00BB 467 : AND IF RESCHEDULING IS NEEDED, TO CONTINUE EXECUTING
00BB 468 : THIS PROCESS AT THE EQUIVALENT OF THE SCH$WAIT LOGIC AT
00BB 469 : SECWAIT.
00BB 470 :
00BB 471 : CALLING SEQUENCE:
00BB 472 :
00BB 473 : JSB MPSS$WAITCHK
00BB 474 :
00BB 475 : INPUT PARAMETERS:
00BB 476 :
00BB 477 : MPSS$GL_CURPCB - ADDRESS OF PCB FOR CURRENT PROCESS ON SECONDARY
00BB 478 :
00BB 479 : IN THE PHD FOR THE CURRENT PROCESS ON SECONDARY:
00BB 480 :
00BB 481 : R0 - PC OF CHMK INSTRUCTION
00BB 482 : R1 - PSL FOR RE-EXECUTION OF CHMK INSTRUCTION
00BB 483 : R2 - WAIT QUEUE HEADER ADDRESS
00BB 484 : R3 - ADDRESS OF EVENT FLAG VECTOR
00BB 485 : R4 - PCB ADDRESS
00BB 486 : 00(SP) - PC AT WHICH TO RE-EXECUTE THE CHMK INSTRUCTION
00BB 487 : 04(SP) - PSL WITH WHICH TO RE-EXECUTE THE CHMK INSTRUCTION
00BB 488 :
00BB 489 : ON PRIMARY'S INTERRUPT STACK:
00BB 490 :
00BB 491 : IF ENTERED FROM IPL 5 INTERRUPT HANDLER:
00BB 492 :
00BB 493 : 00(SP) - SAVED R0 (AT TIME OF INTERRUPT)
00BB 494 : 04(SP) - SAVED R1 (AT TIME OF INTERRUPT)
00BB 495 : 08(SP) - SAVED R2 (AT TIME OF INTERRUPT)
00BB 496 : 0C(SP) - SAVED R3 (AT TIME OF INTERRUPT)
00BB 497 : 10(SP) - SAVED R4 (AT TIME OF INTERRUPT)
00BB 498 : 14(SP) - SAVED R5 (AT TIME OF INTERRUPT)
00BB 499 : 18(SP) - PC AT TIME OF IPL 5 RESCHEDULE INTERRUPT
00BB 500 : 1C(SP) - PSL AT TIME OF IPL 5 RESCHEDULE INTERRUPT
00BB 501 :
```



```
00BB 502 : IF ENTERED FROM IPL 3 INTERRUPT HANDLER:
00BB 503 :
00BB 504 : 00(SP) - PC AT TIME OF IPL 3 RESCHEDULE INTERRUPT
00BB 505 : 04(SP) - PSL AT TIME OF IPL 3 RESCHEDULE INTERRUPT
00BB 506 :
00BB 507 : IPL=SYNCH.
00BB 508 :
00BB 509 : OUTPUTS:
00BB 510 :
00BB 511 : PROCESS IS EITHER RETURNED TO THE SECONDARY TO CONTINUE EXECUTION
00BB 512 : OR THIS PROCESS IS PLACED INTO A WAIT STATE AND ANOTHER PROCESS
00BB 513 : IS SCHEDULED FOR THE SECONDARY.
00BB 514 :
00BB 515 : R0-R4 MAY BE DESTROYED (IF RPTEVT AST IS DONE).
00BB 516 :
00BB 517 : ENVIRONMENT:
00BB 518 :
00BB 519 : EXECUTES ON PRIMARY PROCESSOR, MODE=KERNEL, IPL=SYNCH.
00BB 520 :
00BB 521 : --
00BB 522 :
00BB 523 MPSS$WAITCK::
54 0000'CF D0 00BB 524 MOVL W*MPSS$GL_CURPCB,R4 ;GET PCB OF CURRENT PROCESS ON SECONDARY
55 6C A4 D0 00C0 525 MOVL PCB$$_PHD(R4),R5 ;GET ADDRESS OF HARDWARE PCB
52 0090 C5 D0 00C4 526 MOVL PHD$$_R2(R5),R2 ;GET ADR OF WAIT QUEUE HEADER
53 0094 C5 D0 00C9 527 MOVL PHD$$_R3(R5),R3 ;GET ADR OF EVENT FLAG VECTOR IN CEB
51 63 4C A4 CB 00CE 528 BICL3 PCB$$_EFWM(R4),(R3),R1 ;TEST WAIT MASK
14 13 00D3 529 BEQL 30$ ;BR IF WAIT NOT SATISFIED
06 24 A4 OD E0 00D5 530 BBS #PCB$$_WALL,PCB$$_STS(R4),20$ ;BR IF WAITING FOR ANY FLAGS
00DA 531 :
00DA 532 : RETURN PROCESS TO SECONDARY AS WAIT CONDITION IS SATISFIED.
00DA 533 :
00DA 534 10$:
0000'CF 03 D0 00DA 535 MOVL #MPSS$K_BU$STATE,W*MPSS$GL_STATE ;START SECONDARY EXECUTING PROC
00DF 536 :
00DF 540 :
05 00DF 541 RSB ;RETURN TO RESCHEDULING LOGIC TO
00E0 542 : FINISH EITHER RESTORING REGISTERS
00E0 543 : (IPL 5) OR SCHED PRIMARY (IPL 3)
00E0 544 :
00E0 545 :
00E0 546 : CHECK IF ANY OF THE FLAGS WERE SET -- $WFLOR REQUEST.
00E0 547 :
00E0 548 20$: MCOML R1,R1 ;INVERT MASKED FLAGS
51 51 51 D2 00E0 549 CMPL PCB$$_EFWM(R4),R1 ;CHECK FOR 'AND' OF ALL FLAGS
51 4C A4 D1 00E3 550 BEQL 10$ ;BR IF WAIT SATISFIED
F1 13 00E7 551 BRB 30$ ;CONTINUE IF WAIT NOT SATISFIED
00E9 552 :
00E9 553 :
00E9 554 : PLACE PROCESS IN A WAIT STATE AS WAIT CONDITION IS NOT MET, AND
00E9 555 : RESCHEDULE ANOTHER PROCESS FOR SECONDARY.
00E9 556 :
00E9 557 30$:
00E9 558 BRW SECWAIT ;CONTINUE AT THE EQUIVALENT OF SCH$WAIT
```



```
00E9 565 .SBTTL SECWAIT - PLACES SECONDARY'S PROCESS IN SELECTED WAIT QUEUE
00E9 566 :++
00E9 568 : FUNCTIONAL DESCRIPTION:
00E9 569 :
00E9 573 : SECWAIT PLACES THE SECONDARY'S CURRENT PROCESS IN A WAIT QUEUE
00E9 575 : SELECTED BY A WAIT QUEUE HEADER ADDRESS SUPPLIED IN A REGISTER
00E9 576 : A NEW PROCESS IS THEN SELECTED FOR EXECUTION.
00E9 577 :
00E9 578 : CALLING SEQUENCE:
00E9 579 :
00E9 583 : JMP SECWAIT
00E9 585 :
00E9 586 : INPUT PARAMETERS:
00E9 587 :
00E9 589 : R2 - WAIT QUEUE HEADER ADDRESS
00E9 590 : R4 - PCB ADDRESS
00E9 595 : R5 - PHD ADDRESS
00E9 596 : 00(SP) - PC AT WHICH TO RE-EXECUTE THE CHMK INSTRUCTION
00E9 597 : 04(SP) - PSL WITH WHICH TO RE-EXECUTE THE CHMK INSTRUCTION
00E9 598 :
00E9 599 : ON PRIMARY'S INTERRUPT STACK:
00E9 600 :
00E9 601 : IF ENTERED FROM IPL 5 INTERRUPT HANDLER:
00E9 602 :
00E9 603 : 00(SP) - SAVED R0 (AT TIME OF INTERRUPT)
00E9 604 : 04(SP) - SAVED R1 (AT TIME OF INTERRUPT)
00E9 605 : 08(SP) - SAVED R2 (AT TIME OF INTERRUPT)
00E9 606 : 0C(SP) - SAVED R3 (AT TIME OF INTERRUPT)
00E9 607 : 10(SP) - SAVED R4 (AT TIME OF INTERRUPT)
00E9 608 : 14(SP) - SAVED R5 (AT TIME OF INTERRUPT)
00E9 609 : 18(SP) - PC AT TIME OF IPL 5 RESCHEDULE INTERRUPT
00E9 610 : 1C(SP) - PSL AT TIME OF IPL 5 RESCHEDULE INTERRUPT
00E9 611 :
00E9 612 : IF ENTERED FROM IPL 3 INTERRUPT HANDLER:
00E9 613 :
00E9 614 : 00(SP) - PC AT TIME OF IPL 3 RESCHEDULE INTERRUPT
00E9 615 : 04(SP) - PSL AT TIME OF IPL 3 RESCHEDULE INTERRUPT
00E9 617 :
00E9 618 : IMPLICIT INPUTS:
00E9 619 :
00E9 624 : THIS CODE IS NOT EXECUTING IN THE CONTEXT OF THE PROCESS
00E9 625 : (AS DOES DOES ITS PRIMARY PROCESSOR COUNTERPART - SCH$WAIT).
00E9 626 : THEREFORE, THE STACKS CANNOT BE ACCESSED FROM THIS CODE, ONLY
00E9 627 : THINGS CONTAINED IN THE HARDWARE PCB FOR THIS PROCESS.
00E9 628 : EVERYTHING NEEDED FOR RESCHEDULING THIS PROCESS HAS BEEN
00E9 629 : PLACED IN THE HARDWARE PCB BY THE SECONDARY PROCESSOR BEFORE
00E9 630 : PASSING THIS PROCESS TO THE PRIMARY.
00E9 631 :
00E9 632 : THE PRIMARY HAS ALREADY SET R2-R5 TO BE THE VALUES FOR THE
00E9 633 : CURRENT PROCESS ON THE SECONDARY PROCESSOR. THE PC/PSL FOR
00E9 634 : THE INSTRUCTION FOR RE-EXECUTING THE CHMK INSTRUCTION ARE ON
00E9 635 : THE STACK. THEY ARE ALSO CONTAINED IN R0/R1, SINCE THE PRIMARY
00E9 636 : CANNOT ACCESS THE STACK.
00E9 637 :
00E9 639 : SIDE EFFECTS:
00E9 640 :
00E9 641 : 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:
00E9 666 ADDL #8,PHD$KSP(R5) ;PLACE PROCESS IN WAIT STATE
00E9 667 MOVL PHD$R0(R5),PHD$PC(R5) ;CLEAN PC/PSL OF CHMK INSTR OFF STACK
00C0 C5 78 A5 08 C0 DO 00ED 667 MOVL PHD$R0(R5),PHD$PC(R5) ;SET PC TO BE CHMK INSTRUCTION
00C4 C5 0088 C5 DO 00F4 668 MOVL PHD$R1(R5),PHD$PSL(R5) ;SET PSL BACK FOR CHMK INSTRUCTION
00C4 C5 008C C5 DO 00FB 670 INCW WQH$W_QCNT(R2) ;INCREMENT COUNT FOR QUEUE
00C4 C5 08 A2 B6 00FE 671 INSQUE (R4),WQH$W_QFL(R2) ;INSERT IN QUEUE
2C A4 62 64 OE 00FE 672 MOVW WQH$W_QSTATE(R2),PCB$W_STATE(R4) ;SET STATE FOR PROCESS
2C A4 0A A2 B0 0101 673 :
0106 674 :
0106 675 : THE STATE NUMBER IS CONTAINED
0106 676 : IN THE QUEUE HEADER
50 00000000'GF 3C 0106 683 MOVZWL G^SCH$GW_IOTA,R0 ;Get a longword copy of IOTA
3C A5 50 A0 010D 684 ADDW R0,PHD$W_QUANT(R5) ;CHARGE QUOTA FOR VOLUNTARY WAIT
0100 C5 50 C2 0111 685 SUBL2 R0,PHD$TIMREF(R5) ;Adjust AWSA time reference
0118 C4 00000000'GF D0 0116 686 MOVL G^EXE$GL_ABSTIM,PCB$W_WAITIME(R4) ;RECORD TIME AT WAIT START
00CF C5 04 91 011F 687 CMPB #4,PHD$B_ASTLVL(R5) ;NULL ASTLVL?
00 0000'CF 0C 12 0124 688 BNEQ 20$ ;NO, DO LONG CHECK
0000'CF 01 E6 0126 692 10$: BBSSI #LCK$V_INTERLOCK,W^MPSS$GL_INTERLOCK,15$ ;FLUSH CACHE QUEUE
0000'CF 01 D0 012C 693 15$: MOVL #MPSS$K_IDLESTATE,W^MPSS$GL_STATE ;SET SECONDARY IDLE
05 0131 694 RSB ;RETURN TO SCHEDULE SECONDARY
50 50 00CF C5 9A 0132 696
50 00C4 C5 02 18 ED 0137 697 20$: MOVZBL PHD$B_ASTLVL(R5),R0 ;FETCH AND ZERO EXTEND PENDING ASTLVL
013E 698 CMPZV #PSL$V_CURMOD,#PSL$S_CURMOD,PHD$PSL(R5),R0 ;COMPARE WITH WAIT
E6 19 013E 699 ;ACCESS MODE
0140 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$S_IPC LE 8 ; IPL field must fit into byte
0140 712 :
00C6 1F 93 0140 713 BITB #<PSL$M_IPL0-PSL$V_IPL>,-
0142 714 <PSL$V_IPL/8>+PHD$C_PSL(R5) ;MUST BE AT IPL 0 FOR DELIVERY
DF 12 0145 715 BNEQ 10$ ;BRANCH IF AST NOT DELIVERABLE
52 D4 0147 716 CLRL R2 ;SET NULL PRIORITY INCREMENT
0149 720 RPTVLT AST,JSB ;REPORT AST EVENT
D4 11 0150 722 BRB 10$ ;GO SCHEDULE NEXT PROCESS
0152 724 :
0152 725 .END
```


MPWAIT
Symbol table

K 5
- SECONDARY EVENT FLAG WAIT SERVICES

16-SEP-1984 02:09:57 VAX/VMS Macro V04-00
5-SEP-1984 03:58:01 [SYS.SRC]SYSWAIT.MAR;1

Page 13
(1)

CEBSB_TYPE	= 0000000A		
CEBSL_EFC	= 00000010		
CEBSL_MASTER	= 00000040		
CEBSL_WQFL	= 00000014		
DYN\$C_SLAVCEB	= 0000002D		
EFN	= 00000004		
EVT\$ AST	*****	X	02
EXESC_CMSTKSZ	*****	X	02
EXESGL_ABSTIM	*****	X	02
EXESGL_RPB	*****	X	02
IPL\$ SYNCH	= 00000008		
LCK\$V_INTERLOCK	= 00000000		
MASK	= 00000008		
MPSSGL_CURPCB	*****	X	02
MPSSGL_INTERLOCK	*****	X	02
MPSSGL_SECREQFLG	*****	X	02
MPSSGL_STATE	*****	X	02
MPSSK_BUYSSTATE	= 00000003		
MPSSK_IDLESTATE	= 00000001		
MPSSMPSCHED2	*****	X	02
MPSSV_SECWAITCK	= 00000002		
MPSSWAIT	0000001A	R	02
MPSSWAITCK	000000BB	RG	02
MPSSWAITCONT	0000006F	RG	02
MPSSWAITFR	00000011	RG	02
MPSSWFLAND	00000000	RG	02
MPSSWFLOR	00000007	RG	02
MPSWITCH	= 00000001		
NOWAIT	00000068	R	02
PCBSB_WEFC	= 0000002E		
PCBSL_EFCS	= 00000050		
PCBSL_EFWM	= 0000004C		
PCBSL_PHD	= 0000006C		
PCBSL_SYS	= 00000024		
PCBSL_WAITIME	= 00000118		
PCBSV_WALL	= 0000000D		
PCBSW_STATE	= 0000002C		
PHDSB_ASTLVL	= 000000CF		
PHDSL_KSP	= 00000078		
PHDSL_PC	= 000000C0		
PHDSL_PSL	= 000000C4		
PHDSL_R0	= 00000088		
PHDSL_R1	= 0000008C		
PHDSL_R2	= 00000090		
PHDSL_R3	= 00000094		
PHDSL_TIMREF	= 00000100		
PHDSW_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		
RPBSL_SCBB	= 000000B0		
SCH\$GQ_LEFWQ	*****	X	02
SCH\$GW_IOTA	*****	X	02

SCH\$RSE	*****	X	02
SCH\$WAIT	*****	X	02
SECWAIT	000000E9	R	02
SS\$ ILLEFC	= 000000EC		
SS\$ NORMAL	= 00000001		
SS\$ UNASEFC	= 00000234		
WAIT	00000078	R	02
WAITALL	00000073	R	02
WAITCK	0000005D	R	02
WFRL	0000000B	R	02
WQH\$L_WQFL	= 00000000		
WQH\$W_WQCNT	= 00000008		
WQH\$W_WQSTATE	= 0000000A		

XD
VO

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

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	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
 VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY