F 10

```
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
```

```
**FILE**ID**MPAST                                   E 13




MM      MM  PPPPPPPP      AAAAAA      SSSSSSSS  TTTTTTTTTT
MM      MM  PPPPPPPP      AAAAAA      SSSSSSSS  TTTTTTTTTT
MMMM  MMMM  PP      PP  AA      AA  SS              TT
MMMM  MMMM  PP      PP  AA      AA  SS              TT
MM  MM  MM  PP      PP  AA      AA  SS              TT
MM  MM  MM  PP      PP  AA      AA  SS              TT
MM      MM  PPPPPPPP    AA      AA    SSSSSS        TT
MM      MM  PPPPPPPP    AA      AA    SSSSSS        TT
MM      MM  PP          AAAAAAAAAA          SS      TT
MM      MM  PP          AAAAAAAAAA          SS      TT
MM      MM  PP          AA      AA          SS      TT          ....
MM      MM  PP          AA      AA          SS      TT          ....
MM      MM  PP          AA      AA  SSSSSSSS        TT          ....
MM      MM  PP          AA      AA  SSSSSSSS        TT          ....


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

```
0000      ;
0000      ; Version:      'V04-000'
0000      ;
0000
0000            .MCALL  MFPR
0000            .TITLE  MPAST - MULTIPROCESSOR AST ROUTINES
0000            .IDENT  'V04-000'
0000
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      ;
0000      ; Facility:  Executive, Scheduler
0000      ;
0000      ; Abstract:  Primitives for AST queueing and delivery.
0000      ;
0000      ; Environment: MODE=Kernel
0000      ;
0000      ; Author:  RICHARD I. HUSTVEDT, Creation date:  15-MAY-1979
0000      ;
0000      ; Modified by:
0000      ;
0000            V02-009 KDM0032         Kathleen D. Morse               27-Aug-1981
0000      ;             Don't do rescheduling AST if process can not run on
0000      ;             secondary due to non-zero PHD$L_MPINHIBIT.
0000      ;
0000      ;     V02-008 KDM0015         Kathleen D. Morse       12-Jun-1981
0000      ;             Add use of hook MPH$QEMPTYCONT.
0000      ;
0000      ;     V02-007 KDM0011         Kathleen D. Morse       29-Apr-1981
0000      ;             Integrate a new performance-imoroved AST delivery routine.
0000      ;             Use new exec labels for hooks:  MPH$<name>.
0000      ;
0000      ;     V02-006 KDM0010         Kathleen D. Morse       06-Apr-1981
```

```
0000     ;               Only cause rescheduling AST if there is some other
0000     ;               process to run (not null) and there are no AST's
0000     ;               queued to this process, in addition to other checks.
0000     ;
0000     ;       V02-005 KDM0009         Kathleen D. Morse       06-Apr-1981
0000     ;               Add new routines, MPS$ASTNEWLVL and MPS$ASTSCHEDCHK,
0000     ;               that used to be in module, MPSCHED.  Rename module
0000     ;               to be MPAST.
0000     ;
0000     ;       V02-004 KDM0008         Kathleen D. Morse       01-Apr-1981
0000     ;               Set ASTLVL processor register to executive mode if
0000     ;               the secondary is idle and the PHD ASTLVL > PR ASTLVL.
0000     ;
0000     ;       V02-003 KDM0003         Kathleen D. Morse       15-Sep-1980
0000     ;               Make changes to run on VMS V2.0.
0000     ;
0000     ;       V02-002 KDM0001         Kathleen D. Morse       04-Sep-1980
0000     ;               Replace copyright.
0000     ;
0000     ; 01    -
0000     ;--
0000
0000
```

```
             0000         .SBTTL  DECLARATIONS
             0000
             0000  ;
             0000  ; INCLUDE FILES:
             0000  ;
             0000
             0000  ;
             0000  ; MACROS:
             0000  ;
             0000
             0000  ;
             0000  ; EQUATED SYMBOLS:
             0000  ;
             0000
             0000         $ACBDEF                          ; AST control block definitions
             0000         $IPLDEF                          ; IPL definitions
             0000         $LCKDEF                          ; Interlock bit definitions
             0000         $MPSDEF                          ; Secondary processor states
             0000         $PCBDEF                          ; PCB definitions
             0000         $PHDDEF                          ; PHD definitions
             0000         $PRDEF                           ; Processor register definitions
             0000         $PRIDEF                          ; Priority increment class defs
             0000         $PSLDEF                          ; PSL field definitions
             0000         $SSDEF                           ; Status code definitions
             0000         $STATEDEF                        ; Define state values
             0000
  00000000   0000  ASTEXIT = 0                             ; AST exit change mode code
             0000
             0000  ;
             0000  ; OWN STORAGE:
             0000  ;
         00000000         .PSECT  A$EXENONPAGED,LONG
             0000
```

```
                      0000          .SBTTL  MPS$QAST - ENQUEUE AST CONTROL BLOCK FOR PROCESS
                      0000
                      0000  ;++
                      0000  ;
                      0000  ; FUNCTIONAL DESCRIPTION:
                      0000  ;
                      0000  ; MPS$QAST inserts the AST control block supplied in the proper
                      0000  ; position by access mode in the AST queue of the process specified
                      0000  ; by the PID field of the AST control block.  An AST arrival event
                      0000  ; is then reported for the process to reactivate from a wait state
                      0000  ; if appropriate.  The AST control block will be released immediately
                      0000  ; if the PID specifies a non-existent process.
                      0000  ;
                      0000  ; ENVIRONMENT:
                      0000  ;
                      0000  ;       Executed by the primary processor.
                      0000  ;
                      0000  ; CALLING SEQUENCE:
                      0000  ;
                      0000  ;       BSB/JSB MPS$QAST
                      0000  ;
                      0000  ; INPUT PARAMETERS:
                      0000  ;
                      0000  ;       R2 - Priority increment class
                      0000  ;       R5 - Pointer to AST control block
                      0000  ;
                      0000  ; IMPLICIT INPUTS:
                      0000  ;
                      0000  ;       PCB of process identified by PID field
                      0000  ;
                      0000  ; OUTPUT PARAMETERS:
                      0000  ;
                      0000  ;       R0 - Completion status code
                      0000  ;       R4 - PCB address of process for which AST was queued
                      0000  ;
                      0000  ; SIDE EFFECTS:
                      0000  ;
                      0000  ;       The process identified by the PID in the AST control block
                      0000  ;       will be made executable if not suspended.
                      0000  ;
                      0000  ; COMPLETION CODES:
                      0000  ;
                      0000  ;       SS$_NORMAL  - Normal successful completion status
                      0000  ;       SS$_NONEXPR - Non-existent process
                      0000  ;
                      0000  ;--
                      0000
                      0000          .ENABL  LSB
                      0000  QNONEXPR:                                       ;
      50    55    D0  0000          MOVL    R5,R0                           ; RELEASE AST CONTROL BLOCK
  00000000'GF     16  0003          JSB     G^EXE$DEANONPAGED               ; IF NO SUCH PROCESS
 50   08E8 8F     3C  0009          MOVZWL  #SS$_NONEXPR,R0                 ; SET ERROR STATUS CODE
            56    11  000E          BRB     QEXIT                           ; AND EXIT
                      0010
                      0010  MPS$QAST::                                      ; ENQUEUE AST FOR PROCESS
   50    0C A5    3C  0010          MOVZWL  ACB$L_PID(R5),R0                ; GET PROCESS INDEX FOR AST TARGET
                      0014          DSBINT  #IPL$_SYNCH                     ; DISABLE SYSTEM EVENTS
   54  00000000'GF   D0  001A          MOVL    G^SCH$GL_PCBVEC,R4          ; GET ADDRESS OF VECTOR
         54   6440   D0  0021          MOVL    (R4)[R0],R4                 ; LOOK UP PCB ADDRESS
     60 A4    0C A5  D1  0025          CMPL    ACB$L_PID(R5),PCB$L_PID(R4) ; CHECK FOR MATCH IN PID
                D4   12  002A          BNEQ    QNONEXPR                    ; PID MISMATCHES
                50   D4  002C          CLRL    R0                          ; ASSUME KERNEL MODE AND CLEAR HIGH BITS
        10 A4   65   0E  002E          INSQUE  (R5),PCB$L_ASTQFL(R4)       ; ASSUME QUEUE IS EMPTY AND ATTEMPT INSERT
                3B   12  0032          BNEQ    50$                         ; BR IF IT WAS NOT EMPTY
             0B A5   95  0034          TSTB    ACB$B_RMOD(R5)              ; CHECK FOR SPECIAL KERNEL AST
                06   19  0037          BLSS    10$                         ; BR IF YES
50    0B A5    FC 8F   8B  0039          BICB3   #^C<3>,ACB$B_RMOD(R5),R0; GET AST MODE
                         003F     ;
                         003F     ; THE PROCESS HEADER ADDRESS IS ALWAYS A SYSTEM SPACE ADDRESS (NEGATIVE NUMBER)
                         003F     ; WHILE THE PROCESS HEADER IS RESIDENT.  DURING THE OUTSWAP TRANSITION IT IS
                         003F     ; THE BALANCE SLOT INDEX, A SMALL POSITIVE NUMBER.  FINALLY, AFTER OUTSWAP IT
                         003F     ; IS SET TO ZERO.  HENCE, THE FOLLOWING TEST COMBINES THE FETCH OF THE PHD
                         003F     ; ADDRESS WITH THE TEST FOR PROCESS RESIDENCE.
                         003F     ;
        51    6C A4  D0  003F 10$:     MOVL    PCB$L_PHD(R4),R1            ; POINT TO PROCESS HEADER
                17   18  0043          BGEQ    25$                         ; DON'T SET ASTLVL IF NOT RESIDENT
     00CF C1    50   90  0045          MOVB    R0,PHD$B_ASTLVL(R1)         ; SET ASTLVL IN PROCESS HEADER
        2C A4   0E   B1  004A 20$:     CMPW    #SCH$C_CUR,PCB$W_STATE(R4)  ; IS PROCESS A CURRENT PROCESS?
                0C   12  004E          BNEQ    25$                         ; BR ON NO, NOT A CURRENT PROCESS
   54  00000000'GF   D1  0050          CMPL    G^SCH$GL_CURPCB,R4          ; IS PROCESS ON PRIMARY?
                11   13  0057          BEQL    40$                         ; YES,
             FFA4'   30  0059          BSBW    MPS$INTSCND                 ; INTERRUPT SECONDARY PROCESSOR
                         005C          .LIST MEB
                         005C ;25$:    RPTEVT  AST                         ; REPORT AST ARRIVAL
       00000000'GF   16  005C 25$:     JSB     G^SCH$RSE                   ; REPORT AST ARRIVAL
                    00'  0062          .BYTE   EVT$_AST                    ; EVENT CODE
        50      01   3C  0063 30$:     MOVZWL  #SS$_NORMAL,R0              ; SET SUCCESS STATUS CODE
                         0066 QEXIT:   ENBINT                              ; ENABLE INTERRUPTS
        12      8E   DA  0066                  MTPR    (SP)+,S^#PR$_IPL
                     05  0069          RSB                                 ; AND RETURN
                         006A     ;
                         006A     ; IF THE AST IS BEING ENQUEUED FOR THE CURRENT PROCESS, THEN THE REPORTING
                         006A     ; OF THE AST EVENT CAN BE BYPASSED AND THE ASTLVL PROCESSOR REGISTER MUST BE
                         006A     ; SET INSTEAD.
                         006A     ;
        13      50   DA  006A 40$:     MTPR    R0,#PR$_ASTLVL              ; ALSO SET ASTLVL REGISTER
                F4   11  006D          BRB     30$                         ;
                         006F     ;
                         006F     ; THE AST QUEUE WAS NOT EMPTY (ITS USUAL CONDITION) AND THE PROPER
                         006F     ; POSITION FOR THE NEW AST MUST BE LOCATED.  SINCE THE AST CONTROL
                         006F     ; BLOCK HAS BEEN ERRONEOUSLY INSERTED ON THE QUEUE, IT MUST BE REMOVED
                         006F     ; FIRST.
                         006F     ;
        55      65   0F  006F 50$:     REMQUE  (R5),R5                     ; ELSE CORRECT MISTAKE
     51     10 A4    DE  0072          MOVAL   PCB$L_ASTQFL(R4),R1         ; POINT TO QUEUE HEADER
        53      61   D0  0076          MOVL    (R1),R3                     ; AND GET FIRST ENTRY
          0B A5      95  0079          TSTB    ACB$B_RMOD(R5)              ; CHECK FOR SPECIAL KERNEL AST
                0F   18  007C          BGEQ    70$                         ; BR IF NOT
                         007E     ;
                         007E     ; THE NEW AST IS A SPECIAL KERNEL AST.  IT WILL GO AFTER ALL OTHER SPECIAL
                         007E     ; KERNEL ASTS OR AT THE HEAD OF THE QUEUE IF THERE ARE NONE.
                         007E     ;
        53      51   D1  007E 60$:     CMPL    R1,R3                       ; CHECK FOR END OF QUEUE
                27   13  0081          BEQL    110$                        ; BR IF NOT
```

```
            0B A3   95  0083        TSTB    ACB$B_RMOD(R3)             ; CHECK FOR SPECIAL KERNEL IN QUEUE
               22   18  0086        BGEQ    110$                       ; BR IF NOT
         53    63   D0  0088        MOVL    (R3),R3                    ; FLINK ON TO NEXT ACB.
               F1   11  008B        BRB     60$                        ;
                        008D   ;
                        008D   ; THE NEW AST IS A NORMAL AST.  IT WILL GO AFTER ALL SPECIAL KERNEL ASTS
                        008D   ; AND ASTS WITH LOWER ACCESS MODE.
                        008D   ;
 50    0B A5     FC 8F  8B  008D   70$:  BICB3   #^C<3>,ACB$B_RMOD(R5),R0; GET AST MODE
         53    51   D1  0093   80$:  CMPL    R1,R3                      ; CHECK FOR END OF QUEUE
               12   13  0096        BEQL    110$                       ; INSERT IF AT END
         02    00   ED  0098        CMPZV   #ACB$V_MODE,#ACB$S_MODE,-
      50    0B A3       009B                ACB$B_RMOD(R3),R0          ; COMPARE ACCESS MODES
               05   14  009E        BGTR    100$                       ; IF GTR AT RIGHT PLACE
         53    63   D0  00A0   90$:  MOVL    (R3),R3                    ; FLINK ON TO NEXT ACB
               EE   11  00A3        BRB     80$                        ;
            0B A3   95  00A5   100$: TSTB    ACB$B_RMOD(R3)             ; IS THIS ENTRY A SPECIAL KAST?
               F6   19  00A8        BLSS    90$                        ; YES, MUST GO AFTER THIS
                        00AA   ;
                        00AA   ; NOW THE CORRECT POSITION HAS BEEN LOCATED.  INSERT THE AST CONTROL BLOCK
                        00AA   ; ON THE QUEUE AND COMPUTE THE NEW VALUE FOR ASTLVL BY INTERROGATING THE
                        00AA   ; MODE OF THE AST CONTROL BLOCK AT THE HEAD OF THE QUEUE.
                        00AA   ;
      04 B3    65   0E  00AA   110$: INSQUE  (R5),@ACB$L_ASTQBL(R3)     ; INSERT AFTER PREVIOUS
               50   D4  00AE        CLRL    R0                         ; ASSUME KERNEL MODE
      51    10 A4   D0  00B0        MOVL    PCB$L_ASTQFL(R4),R1        ; GET HEAD OF AST QUEUE
            0B A1   95  00B4        TSTB    ACB$B_RMOD(R1)             ; IS IT KAST?
               86   19  00B7        BLSS    10$                        ; BR IF YES TO SET ASTLVL
 50    0B A1     FC 8F  8B  00B9        BICB3   #^C<3>,ACB$B_RMOD(R1),R0; GET AST MODE FOR HEAD OF QUEUE
             FF7D   31  00BF        BRW     10$                        ; GO SET ASTLVL
                        00C2
                        00C2        .DSABL LSB
                        00C2        ASSUME  ACB$V_MODE EQ 0
                        00C2        ASSUME  ACB$S_MODE EQ 2
                        00C2        ASSUME  ACB$V_KAST EQ 7
                        00C2
```

```
                                 00C2          .SBTTL  MPS$ASTSCHEDCHK - CHECK FOR RESCHEDULE AT AST DELIVERY
                                 00C2      ;++
                                 00C2      ; FUNCTIONAL DESCRIPTION:
                                 00C2      ;
                                 00C2      ; MPS$ASTSCHEDCHK is entered to check whether the process that the
                                 00C2      ; primary was running, should be rescheduled to run on the secondary.
                                 00C2      ;
                                 00C2      ; ENVIRONMENT:
                                 00C2      ;
                                 00C2      ;       Executed by the primary processor.
                                 00C2      ;       Hooked in at MPH$ASTDELHK and sometimes returns to MPH$ASTDELCONT.
                                 00C2      ;
                                 00C2      ;--
                                 00C2
                                 00C2              .ALIGN  LONG
                                 00C4      MPS$ASTSCHEDCHK::
         55    10 B4      0F     00C4              REMQUE  @PCB$L_ASTQFL(R4),R5    ; Remove head of queue
                  06      1D     00C8              BVS     20$                     ; Br if queue empty
          00000000'9F     17     00CA      10$:    JMP     @#MPH$ASTDELCONT        ; Return to normal code
    00 0000'CF    00      E6     00D0      20$:    BBSSI   #LCK$V_INTERLOCK,W^MPS$GL_INTERLOCK,30$ ; Flush cache queue
       01    0000'CF      D1     00D6      30$:    CMPL    W^MPS$GL_STATE,#MPS$K_IDLESTATE ; Is secondary idle?
                  03      12     00DB              BNEQ    40$                     ; Br on no, dont bother to reschedule
                                 00DD              SOFTINT #3                      ; Request a reschedule interrupt
            14    03      DA     00DD                      MTPR    #3,S^#PR$_SIRR
          00000000'9F     17     00E0      40$:    JMP     @#MPH$QEMPTYCONT        ; Go set null ast level
```

```
                                 00E6    286           .SBTTL  MPS$ASTNEWLVL - CHECK FOR SETTING NEW ASTLVL
                                 00E6    287 ;++
                                 00E6    288 ; FUNCTIONAL DESCRIPTION:
                                 00E6    289 ;
                                 00E6    290 ; MPS$ASTNEWLVL is entered to check whether a new AST level should
                                 00E6    291 ; be placed in the processor ASTLVL register   If the AST level is kernel,
                                 00E6    292 ; it should be left alone.  Otherwise, the special indication of an idle
                                 00E6    293 ; secondary may be used.
                                 00E6    294 ;
                                 00E6    295 ; ENVIRONMENT:
                                 00E6    296 ;
                                 00E6    297 ;         Executed by the primary processor.
                                 00E6    298 ;         Hooked in at MPH$NEWLVL, replaces code from hook to end of routine.
                                 00E6    299 ;
                                 00E6    300 ;--
                                 00E6    301
                                 00E6    302           .ALIGN  LONG
                                 00E8    303 MPS$ASTNEWLVL::
             13     52     DA    00E8    304           MTPR    R2,#PR$_ASTLVL          ; Set ASTLVL register
                    2C     13    00EB    305           BEQL    20$                     ; If kernel, dont change it
     00 0000'CF     00     E6    00ED    306           BBSSI   #LCK$V_INTERLOCK,W^MPS$GL_INTERLOCK,10$ ; Flush cache queue
     01   0000'CF   D1           00F3    307 10$:      CMPL    W^MPS$GL_STATE,#MPS$K_IDLESTATE ; Is secondary idle?
                    1F     12    00F8    308           BNEQ    20$                     ; Br on no, dont change ASTLVL
00000000'GF   7FFFFFFF 8F  D3    00FA    309           BITL    #^X7FFFFFFF,G^SCH$GL_COMQS ; Is any process other than null COM?
                    12     13    0105    310           BEQL    20$                     ; Br on no, nothing else can run
            10 A4   9F           0107    311           PUSHAB  PCB$L_ASTQFL(R4)        ; Are there any AST's outstanding
         10 A4      8E     D1    010A    312           CMPL    (SP)+,PCB$L_ASTQFL(R4)  ; for this process?  If so, don't
                    09     12    010E    313           BNEQ    20$                     ; cause a reschedule AST.
          0104 C0          D5    0110    314           TSTL    PHD$L_MPINHIBIT(R0)     ; Any reason not to run on secondary?
                    03     12    0114    315           BNEQ    20$                     ; Br on yes, don't cause reschedule AST
             13     01     DA    0116    316           MTPR    #PSL$C_EXEC,#PR$_ASTLVL ; Set ASTLVL to cause reschedule
          00CF C0   52     90    0119    317 20$:      MOVB    R2,PHD$B_ASTLVL(R0)     ; Set ASTLVL in PHD
                                 011E    318           ENBINT                          ; Enable system events again
             12     8E     DA    011E                          MTPR    (SP)+,S^#PR$_IPL
                           05    0121    319           RSB                             ; Return
                                 0122    320
                                 0122    321           .END
```

```
ACB$B_RMOD                     = 0000000B
ACB$L_ASTQBL                   = 00000004
ACB$L_PID                      = 0000000C
ACB$S_MODE                     = 00000002
ACB$V_KAST                     = 00000007
ACB$V_MODE                     = 00000000
ASTEXIT                        = 00000000
EVT$_AST                         ********     X    02
EXE$DEANONPAGED                  ********     X    02
IPL$_SYNCH                     = 00000008
LCK$V_INTERLOCK                = 00000000
MPH$ASTDELCONT                   ********     X    02
MPH$QEMPTYCONT                   ********     X    02
MPS$ASTNEWLVL                    000000E8 RG       02
MPS$ASTSCHEDCHK                  000000C4 RG       02
MPS$GL_INTERLOCK                 ********     X    02
MPS$GL_STATE                     ********     X    02
MPS$INTSCND                      ********     X    02
MPS$K_IDLESTATE                = 00000001
MPS$QAST                         00000010 RG       02
PCB$L_ASTQFL                   = 00000010
PCB$L_PHD                      = 0000006C
PCB$L_PID                      = 00000060
PCB$W_STATE                    = 0000002C
PHD$B_ASTLVL                   = 000000CF
PHD$L_MPINHIBIT                = 00000104
PR$_ASTLVL                     = 00000013
PR$_IPL                        = 00000012
PR$_SIRR                       = 00000014
PSL$C_EXEC                     = 00000001
QEXIT                            00000066 R        02
QNONEXPR                         00000000 R        02
SCH$C_CUR                      = 0000000E
SCH$GL_COMQS                     ********     X    02
SCH$GL_CURPCB                    ********     X    02
SCH$GL_PCBVEC                    ********     X    02
SCH$RSE                          ********     X    02
SS$_NONEXPR                    = 000008E8
SS$_NORMAL                     = 00000001

                                            +----------------+
                                            ! 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
A$EXENONPAGED                   00000122  (  290.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC LONG
```

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

Phase                   Page faults    CPU Time       Elapsed Time
-----                   -----------    --------       ------------
Initialization                   47    00:00:00.10    00:00:00.79
Command processing              164    00:00:00.99    00:00:06.15
Pass 1                          301    00:00:08.98    00:00:26.48
Symbol table sort                 5    00:00:01.42    00:00:02.56
Pass 2                           73    00:00:01.74    00:00:05.31
Symbol table output               5    00:00:00.07    00:00:00.21
Psect synopsis output             2    00:00:00.03    00:00:00.09
Cross-reference output            0    00:00:00.00    00:00:00.00
Assembler run totals            599    00:00:13.33    00:00:41.59

The working set limit was 1500 pages.
47761 bytes (94 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 871 non-local and 18 local symbols.
326 source lines were read in Pass 1, producing 14 object records in Pass 2.
23 pages of virtual memory were used to define 22 macros.

                                       +---------------------------+
                                       ! Macrc library statistics !
                                       +---------------------------+

Macro library name                             Macros defined
------------------                             --------------
_$255$DUA28:[MP.OBJ]MP.MLB;1                                4
_$255$DUA28:[SYS.OBJ]LIB.MLR;1                              9
_$255$DUA28:[SYSLIB]STARLET.MLB;2                           7
TOTALS (all libraries)                                     20

1040 GETS were required to define 20 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:MPAST/OBJ=OBJ$:MPAST MSRC$:MPPREFIX/UPDATE=(ENH$:MPPREFIX)+MSRC$:MPAST/UPDATE=(ENH$:MPAST)+EXECML$/LIB+LIB$:MP.MLB/LI
```

0247 AH-BT13A-SE DIGITAL EQUIPMENT CORPORATION
VAX/VMS V4.0 CONFIDENTIAL AND PROPRIETARY

VMOUNT
LIS

MPCLRPFM
LIS

MPAST
LIS

MP

MP
MAP

MP
MDL

TRNLOG
LIS

MPCMOD
LIS

MPMACROS
MAR