```
MMM          MMM  TTTTTTTTTTTTTTT  HHH          HHH  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
MMM          MMM  TTTTTTTTTTTTTTT  HHH          HHH  RRRRRRRRRRRR    TTTTTTTTTTTTTTT  LLL
MMM          MMM  TTTTTTTTTTTTTTT  HHH          HHH  RRRRRRRRRRRR    TTTTTTTTTTTTTTT  LLL
MMMMMM    MMMMMM        TTT       HHH          HHH  RRR         RRR       TTT         LLL
MMMMMM    MMMMMM        TTT       HHH          HHH  RRR         RRR       TTT         LLL
MMMMMM    MMMMMM        TTT       HHH          HHH  RRR         RRR       TTT         LLL
MMM  MMM  MMM           TTT       HHH          HHH  RRR         RRR       TTT         LLL
MMM  MMM  MMM           TTT       HHH          HHH  RRR         RRR       TTT         LLL
MMM  MMM  MMM           TTT       HHH          HHH  RRR         RRR       TTT         LLL
MMM       MMM           TTT       HHHHHHHHHHHHHHH  RRRRRRRRRRR         TTT         LLL
MMM       MMM           TTT       HHHHHHHHHHHHHHH  RRRRRRRRRRRR        TTT         LLL
MMM       MMM           TTT       HHHHHHHHHHHHHHH  RRRRRRRRRRRR        TTT         LLL
MMM       MMM           TTT       HHH          HHH  RRR   RRR           TTT         LLL
MMM       MMM           TTT       HHH          HHH  RRR   RRR           TTT         LLL
MMM       MMM           TTT       HHH          HHH  RRR   RRR           TTT         LLL
MMM       MMM           TTT       HHH          HHH  RRR      RRR        TTT         LLL
MMM       MMM           TTT       HHH          HHH  RRR      RRR        TTT         LLL
MMM       MMM           TTT       HHH          HHH  RRR      RRR        TTT         LLL
MMM       MMM           TTT       HHH          HHH  RRR         RRR     TTT         LLLLLLLLLLLLLLL
MMM       MMM           TTT       HHH          HHH  RRR         RRR     TTT         LLLLLLLLLLLLLLL
MMM       MMM           TTT       HHH          HHH  RRR         RRR     TTT         LLLLLLLLLLLLLLL
```

FILEID**OTSPOWDJ

```
  OOOOOO    TTTTTTTTTT    SSSSSSSS  PPPPPPPP     OOOOOO    WW      WW  DDDDDDDD          JJ
  OOOOOO    TTTTTTTTTT    SSSSSSSS  PPPPPPPP     OOOOOO    WW      WW  DDDDDDDD          JJ
OO      OO      TT      SS          PP      PP OO      OO  WW      WW  DD      DD        JJ
OO      OO      TT      SS          PP      PP OO      OO  WW      WW  DD      DD        JJ
OO      OO      TT      SS          PP      PP OO      OO  WW      WW  DD      DD        JJ
OO      OO      TT      SS          PP      PP OO      OO  WW      WW  DD      DD        JJ
OO      OO      TT        SSSSSS    PPPPPPPP   OO      OO  WW      WW  DD      DD        JJ
OO      OO      TT        SSSSSS    PPPPPPPP   OO      OO  WW      WW  DD      DD        JJ
OO      OO      TT              SS  PP         OO      OO  WW  WW  WW  DD      DD  JJ    JJ
OO      OO      TT              SS  PP         OO      OO  WW  WW  WW  DD      DD  JJ    JJ
OO      OO      TT              SS  PP         OO      OO  WWWW  WWWW  DD      DD  JJ    JJ      ....
OO      OO      TT              SS  PP         OO      OO  WWWW  WWWW  DD      DD  JJ    JJ      ....
  OOOOOO        TT      SSSSSSSS    PP           OOOOOO    WW      WW  DDDDDDDD      JJJJJJ      ....
  OOOOOO        TT      SSSSSSSS    PP           OOOOOO    WW      WW  DDDDDDDD      JJJJJJ      ....

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

Table of contents

```
0000     1          .TITLE  OTS$POWDJ - DOUBLE PRECISION ** INTEGER*4 power routine
0000     2          .IDENT  /1-006/                 ; File: OTSPOWDJ.MAR  Edit: SBL1006
0000     3
0000     4  ;
0000     5  ;*****************************************************************************
0000     6  ;*                                                                           *
0000     7  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9  ;*  ALL RIGHTS RESERVED.                                                     *
0000    10  ;*                                                                           *
0000    11  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    12  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    13  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    14  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    15  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    16  ;*  TRANSFERRED.                                                             *
0000    17  ;*                                                                           *
0000    18  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    19  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    20  ;*  CORPORATION.                                                             *
0000    21  ;*                                                                           *
0000    22  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    23  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24  ;*                                                                           *
0000    25  ;*                                                                           *
0000    26  ;*****************************************************************************
0000    27  ;
0000    28  ;
0000    29  ; FACILITY: Language support library - user callable
0000    30  ;++
0000    31  ; ABSTRACT:
0000    32  ;
0000    33  ;       Double base to integer longword power.
0000    34  ;       Floating overflow and underflow can occur.
0000    35  ;       Undefined exponentation can occur if base is 0 and power is 0 or negative.
0000    36  ;
0000    37  ;
0000    38  ;--
0000    39  ;
0000    40  ; VERSION: 01
0000    41  ;
0000    42  ; HISTORY:
0000    43  ; AUTHOR:
0000    44  ;       Thomas N. Hastings, 5-May-77: Version 01
0000    45  ;
0000    46  ; MODIFIED BY: SUSAN HUBBARD AZIBERT
0000    47  ;
0000    48  ;
0000    49  ;
```

```
0000     51          .SBTTL  HISTORY         ; Detailed Current Edit History
0000     52
0000     53
0000     54 ; Edit History for Version 01 of OTS$POWDJ
0000     55 ; version 5 - changed module name to FORPOWDJ
0000     56 ; version 6 - added error handler and changed formal ref from 8(AP) to 12(AP)
0000     57 ; version 8 - changed error handler name from MTH$ERROR to MTH$$ERROR
0000     58 ; version 9 - removed W^ from MTH$$ERROR, saved code with MOVZBL.
0000     59 ; version 10 - changed references to MTH$$ERROR to MTH$$SIGNAL - JMT
0000     60 ; 0-14  - Change FOR$FLAG_JACKET to MTH$FLAG_JACKET.  TNH 17-July-78
0000     61 ; 1-001 - Update version number and copyright notice.  JBS 16-NOV-78
0000     62 ; 1-002 - Include MTHJACKET at assembly time and change MTH__UNDEXP to
0000     63 ;          MTH$K_UNDEXP.  JBS 07-DEC-78
0000     64 ; 1-003 - Add "_" to the PSECT directive.  JBS 22-DEC-78
0000     65 ; 1-004 - Declare externals.  SBL 17-May-1979
0000     66 ; 1-005 - Add handlers to catch SS$_FLTOVF and SS$_FLTDIV, and signal
0000     67 ;          MTH$_FLOOVEMAT or MTH$_FLOUNDMAT instead, depending on the context.
0000     68 ;          Also disable IV and change BLBS/ASHL at EXPGTR to BBSC/ROTL for
0000     69 ;          uniformity with OTS$POWRJ.  JAW 26-Feb-1980.
0000     70 ; 1-006 - Use general mode addressing.  SBL 30-Nov-1981
0000     71 ;
0000     72 ;
```

```
                     0000    74          .SBTTL  DECLARATIONS
                     0000    75
                     0000    76 ;
                     0000    77 ; INCLUDE FILES:
                     0000    78 ;
                     0000    79 ;       MTHJACKET.MAR                       ; Math jacketing macro
                     0000    80 ;
                     0000    81 ; EXTERNAL SYMBOLS:
                     0000    82 ;
                     0000    83
                     0000    84          .DSABL  GBL
                     0000    85          .EXTRN  MTH$K_UNDEXP, MTH$K_FLOOVEMAT, MTH$K_FLOUNDMAT
                     0000    86          .EXTRN MTH$$SIGNAL                  ; Math error routine
                     0000    87          .EXTRN  SS$_FLTOVF, SS$_FLTOVF_F, SS$_FLTDIV, SS$_FLTDIV_F, SS$_CONTINUE
                     0000    88
                     0000    89 ;
                     0000    90 ; MACROS:
                     0000    91 ;
                     0000    92          $CHFDEF                             ; Define condition handler symbols.
                     0000    93          $SFDEF                              ; Define stack frame symbols.
                     0000    94          $PSLDEF                             ; Define program status longword
                     0000    95                                              ; symbols.
                     0000    96
                     0000    97 ;
                     0000    98 ; EQUATED SYMBOLS:
                     0000    99 ;
00000004             0000   100          base = 4                            ; base input formal - by-value
0000000C             0000   101          exp = 12                            ; exponent intpu formal - by-value
                     0000   102                                              ; Note: double floating by-value violates
                     0000   103                                              ; calling standard, but ok since this
                     0000   104                                              ; routine is a code support routine (OTS$)
                     0000   105
                     0000   106 ;
                     0000   107 ; OWN STORAGE:
                     0000   108 ;
                     0000   109
                     0000   110 ;
                     0000   111 ; PSECT DECLARATIONS:
                     0000   112 ;
                     0000   113
00000000             0000   114          .PSECT  _OTS$CODE PIC,SHR,LONG,EXE,NOWRT
                     0000   115                                              ; program section for OTS$ code
                     0000   116
```

```
0000    118          .SBTTL  OTS$POWDJ - double to power longword giving double result
0000    119
0000    120 ;++
0000    121 ; FUNCTIONAL DESCRIPTION:
0000    122 ;
0000    123 ;       Double result = double base ** signed longword exponent
0000    124 ;       The double result is given by:
0000    125 ;
0000    126 ;       base    exponent        result
0000    127 ;
0000    128 ;       any     > 0             product (base * 2**i) where i is each
0000    129 ;                               non-zero bit position in exponent
0000    130 ;
0000    131 ;       > 0     = 0             1.0
0000    132 ;       = 0     = 0             Undefined exponentation
0000    133 ;       < 0     = 0             1.0
0000    134 ;
0000    135 ;       > 0     < 0             1.0 / product (base * 2**i)
0000    136 ;                               where i is each non-zero bit position
0000    137 ;                               in !exponent!
0000    138 ;       = 0     < 0             Undefined exponentation
0000    139 ;       < 0     < 0             1.0 / product (base * 2**i)
0000    140 ;                               where i is each non-zero bit position
0000    141 ;                               in !exponent!
0000    142 ;
0000    143 ;       Floating overflow can occur on either of the two MULD's.  If this
0000    144 ;       happens when the exponent is less than zero, the exception is caught by
0000    145 ;       a local condition handler named EXC_HNDLR_UNDER, which sets the result
0000    146 ;       to 0.0 and either signals MTH$_FLOUNDMAT (if FU is enabled in the
0000    147 ;       caller's PSW) or continues at POWDJX.  If it happens when the exponent
0000    148 ;       is greater than zero, the exception is caught by a local condition
0000    149 ;       handler named EXC_HNDLR_OVER, which sets the result to the reserved
0000    150 ;       operand (-0.0) and signals MTH$_FLOOVEMAT.
0000    151 ;
0000    152 ;       Floating overflow and floating divide by zero can occur on the DIVD.
0000    153 ;       These exceptions are caught by EXC_HNDLR_OVER, which sets the result to
0000    154 ;       the reserved operand (-0.0) and signals MTH$_FLOOVEMAT.
0000    155 ;
0000    156 ;       Undefined exponentiation occurs if base is 0 and
0000    157 ;       exponent is 0 or negative.
0000    158 ;
0000    159 ; CALLING SEQUENCE:
0000    160 ;
0000    161 ;       Power.wd.v = OTS$POWDJ (base.rd.v, exponent.rl.v)
0000    162 ;
0000    163 ; INPUT PARAMETERS:
0000    164 ;       NONE
0000    165 ;
0000    166 ; IMPLICIT INPUTS:
0000    167 ;       The setting of FU in the caller's PSW.
0000    168 ;
0000    169 ; OUTPUT PARAMETERS:
0000    170 ;       NONE
0000    171 ;
0000    172 ; IMPLICIT OUTPUTS:
0000    173 ;       NONE
0000    174 ;
```

```
                                0000   175 ; FUNCTION VALUE:
                                0000   176 ;
                                0000   177 ;       Double base **  signed longword exponent
                                0000   178 ;
                                0000   179 ; SIDE EFFECTS:
                                0000   180 ;
                                0000   181 ;       Signals MTH$_FLOOVEMAT if floating overflow occurs on either of the two
                                0000   182 ;           MULD's when exponent > 0, or if floating overflow or divide by zero
                                0000   183 ;           occurs on the DIVD.
                                0000   184 ;       Signals MTH$_FLOUNDMAT if floating overflow occurs on either of the two
                                0000   185 ;           MULD's when exponent < 0 and caller has FU enabled.
                                0000   186 ;       SIGNALs MTH$_UNDEXP (82 = ' UNDEFINED EXPONENTATION') if
                                0000   187 ;       base is 0 and exponent is 0 or negative.
                                0000   188 ;
                                0000   189 ;--
                                0000   190
                                0000   191
                                0000   192
                        001C    0000   193         .ENTRY  OTS$POWDJ, ^M<R2, R3, R4>
                                0002   194                                             ; Disable integer overflow.  (Occurs on
                                0002   195                                             ; maximum negative exponent.)
                6D   66'AF   9E 0002   196         MOVAB   B^EXC_HNDLR_OVER, (FP)       ; Translate exceptions to
                                0006   197                                             ; MTH$_FLOOVEMAT.
                50     08    70 0006   198         MOVD    #1, R0                       ; R0/R1 = initial result
                52     04 AC 70 0009   199         MOVD    base(AP), R2                 ; R2/R3 = base
                54     0C AC D0 000D   200         MOVL    exp(AP), R4                  ; R4 = exponent
                       0D    14 0011   201         BGTR    EXPGTR                       ; branch if exponent > 0
                6D   57'AF   9E 0013   202         MOVAB   B^EXC_HNDLR_UNDER, (FP)      ; Translate exceptions to
                                0017   203                                             ; MTH$_FLOUNDMAT.
                                0017   204
                          52    73 0017   205         TSTD    R2                           ; test base
                          2C    13 0019   206         BEQL    UNDEFINED                    ; undefined 0**0 or 0**(-n)
                       54     54 CE 001B   207         MNEGL   R4, R4                       ; R4 = |exponent|
                          26    13 001E   208         BEQL    POWDJX                       ; if exponent is 0, return R0 = 1.0
                                0020   209
                                0020   210 ;+
                                0020   211 ; Exponent is > 0 or (exponent is =< 0 and base is not = 0 -- use |exponent|)
                                0020   212 ;-
                                0020   213
                    0B 54       00 E4 0020   214 EXPGTR: BBSC    #0, R4, PARTIAL              ; branch if |exponent| is odd
                54     54       FF 8F 9C 0024   215 SQUAR:  ROTL    #-1, R4, R4                  ; R4 = |exponent|/2
                          52    52 64 0029   216 SQUAR1: MULD    R2, R2                       ; R2/R3 = current power of base
                                002C   217                                             ; Floating overflow will trap or fault
                                002C   218                                             ; and signal SS$_FLTOVF or SS$_FLTOVF_F.
                             F5 54 E9 002C   219         BLBC    R4, SQUAR                    ; branch if next bit in |exponent| is 0
                                002F   220
                                002F   221 ;+
                                002F   222 ; Here when bit i of |exponent| is a 1.
                                002F   223 ; Partial result = partial result *  (base * 2**i)
                                002F   224 ;-
                                002F   225
                                002F   226 PARTIAL:
                          50    52 64 002F   227         MULD    R2, R0                       ; R0/R1 = new partial result
                54     54       FF 8F 78 0032   228         ASHL    #-1, R4, R4                  ; R4 = |exponent|/2
                                F0 12 0037   229         BNEQ    SQUAR1                       ; loopback if more exponent bits are 1
                                0039   230
                             0C AC D5 0039   231         TSTL    exp(AP)                      ; test sign of exponent
```

```
                     08   14   003C   232          BGTR    POWDJX                          ; if exponent > 0, return R0
                  6D   66'AF   9E   003E   233          MOVAB   B^EXC_HNDLR_OVER, (FP)  ; Translate exceptions to
                               0042   234                                                  ; MTH$_FLOOVEMAT.
                50   08   50   67   0042   235          DIVD3   R0, #1, R0              ; R0/RT = 1.0/result
                               04   0046   236 POWDJX:  RET                             ; return, result in R0
                                    0047   237
                                    0047   238 ;+
                                    0047   239 ; Undefined exponentation error - 0**0 or 0**(-n)
                                    0047   240 ;-
                                    0047   241
                                    0047   242 UNDEFINED:
                   50   01   0F   79   0047   243          ASHQ    #15, #1, R0             ; R0/R1 = reserved floating operand
                     7E   00'8F   9A   004B   244          MOVZBL  #MTH$K_UNDEXP, -(SP)    ; Indicate undefined exponentiation.
             00000000'GF   01   FB   004F   245          CALLS   #1, G^MTH$$SIGNAL       ; convert to 32-bit condition code
                                    0056   246                                          ; and SIGNAL MTH$_UNDEXP
                                    0056   247                                          ; Note: 2nd arg not needed since no JSB OTS$
                                    0056   248                                          ; is possible.
                               04   0056   249          RET                             ; return
                                    0057   250
                                    0057   251 ;+
                                    0057   252 ; The following handler is established to process exceptions which imply
                                    0057   253 ; underflow of the final result (floating overflow in either of the two MULD's
                                    0057   254 ; when exp < 0).  On the occurrence of such an exception, the handler signals
                                    0057   255 ; MTH$_FLOUNDMAT.
                                    0057   256 ;-
                                    0057   257
                                    0057   258 EXC_HNDLR_UNDER:
                           001C     0057   259          .WORD   ^M<R2, R3, R4>          ; Entry mask
                        2B   10     0059   260          BSBB    SETUP                   ; Set up R0:R3 and identify condition.
                                    005B   261                                          ; Return only if FLTOVF or FLTDIV.
                1E 04 A2   06   E1   005B   262          BBC     #PSL$V_FU, SF$W_SAVE_PSW(R2), CON_U
                                    0060   263                                          ; Branch if caller has not enabled FU.
                   54   00'8F   9A   0060   264          MOVZBL  #MTH$K_FLOUNDMAT, R4    ; Report MTH$_FLOUNDMAT, not SS$_FLTOVF.
                               0C   11   0064   265          BRB     DO_SIG
                                    0066   266
                                    0066   267 ;+
                                    0066   268 ; The following handler is established to process exceptions which imply
                                    0066   269 ; overflow of the final result (floating overflow in either of the two MULD's
                                    0066   270 ; when exp > 0, floating overflow in the DIVD, or floating divide by zero in the
                                    0066   271 ; DIVD).  On the occurrence of such an exception, the handler signals
                                    0066   272 ; MTH$_FLOOVEMAT.
                                    0066   273 ;-
                                    0066   274
                                    0066   275 EXC_HNDLR_OVER:
                           001C     0066   276          .WORD   ^M<R2, R3, R4>          ; Entry mask
                        1C   10     0068   277          BSBB    SETUP                   ; Set up R0:R3 and identify condition.
                                    006A   278                                          ; Return only if FLTOVF or FLTDIV.
                   50   01   0F   78   006A   279          ASHL    #15, #1, R0             ; Make the default result -0.0.
                     54   00'8F   9A   006E   280          MOVZBL  #MTH$K_FLOOVEMAT, R4    ; Report MTH$_FLOOVEMAT, not SS$_FLTxxx.
                                    0072   281
                      10 A2   DD   0072   282 DO_SIG:  PUSHL   SF$L_SAVE_PC(R2)        ; Report caller's PC, not exception PC.
                         54   DD   0075   283          PUSHL   R4                      ; Report MTH$_xxx, not SS$_xxx.
             00000000'GF   02   FB   0077   284          CALLS   #2, G^MTH$$SIGNAL       ; Signal the condition.
                0C A3   50   7D   007E   285 CON_U:   MOVQ    R0, CHF$L_MCH_SAVR0(R3) ; If continued, restore R0 and R1.
                   50   00'   D0   0082   286          MOVL    S^#SS$_CONTINUE, R0     ; Continue from the original exception.
                               04   0085   287 DO_RET:  RET                             ; Exit from handler.
                                    0086   288
```

```
                     0086    289 ;+
                     0086    290 ; Common setup routine for handlers.  Returns normally if exception was FLTOVF,
                     0086    291 ; FLTOVF_F, FLTDIV, or FLTDIV_F.  If the exception was anything else, it
                     0086    292 ; executes a RET, causing an exit from the handler with R0 = 0, which is
                     0086    293 ; equivalent to SS$_RESIGNAL.  In the case of a normal return (FLTOVF, FLTOVF_F,
                     0086    294 ; FLTDIV, or FLTDIV_F) it sets up R0:R3 as follows:
                     0086    295 ;       R0/R1:  0
                     0086    296 ;       R2:     address of establisher's frame
                     0086    297 ;       R3:     address of mechanism array
                     0086    298 ;-
                     0086    299
              50  7C 0086    300 SETUP:  CLRQ    R0                              ; Set default result to 0.0.
          52  04 AC  7D 0088    301         MOVQ    CHF$L_SIGARGLST(AP), R2 ; R2 = address of signal array
                     008C    302                                             ; R3 = address of mechanism array
 0000'8F  04 A2  B1 008C    303         CMPW    CHF$L_SIG_NAME(R2), #SS$_FLTOVF
                     0092    304                                             ; Was it a floating overflow trap?
              18  13 0092    305         BEQL    DO_RSB                          ; Branch if yes.
 0000'8F  04 A2  B1 0094    306         CMPW    CHF$L_SIG_NAME(R2), #SS$_FLTOVF_F
                     009A    307                                             ; Or a floating overflow fault?
              10  13 009A    308         BEQL    DO_RSB                          ; Branch if yes.
 0000'8F  04 A2  B1 009C    309         CMPW    CHF$L_SIG_NAME(R2), #SS$_FLTDIV
                     00A2    310                                             ; Or a floating divide by zero trap?
              08  13 00A2    311         BEQL    DO_RSB                          ; Branch if yes.
 0000'8F  04 A2  B1 00A4    312         CMPW    CHF$L_SIG_NAME(R2), #SS$_FLTDIV_F
                     00AA    313                                             ; Or a floating divide by zero fault?
                  D9 12 00AA    314         BNEQ    DO_RET                          ; None of the above: return from handler
                     00AC    315                                             ; with R0 = 0.
    08 A2  97 AF  9E 00AC    316 DO_RSB: MOVAB   B^POWDJX, CHF$L_SIG_NAME+4(R2)
                     00B1    317                                             ; Change return PC to POWDJX.
          52  04 A3  D0 00B1    318         MOVL    CHF$L_MCH_FRAME(R3), R2 ; R2 = address of establisher's frame
                  05 00B5    319         RSB                                     ; Return.
                     00B6    320         .END
```

```
BASE                =  00000004
CHF$L_MCH_FRAME     =  00000004
CHF$L_MCH_SAVR0     =  0000000C
CHF$L_SIGARGLST     =  00000004
CHF$L_SIG_NAME      =  00000004
CON_U                  0000007E  R      02
DO_RET                 00000085  R      02
DO_RSB                 000000AC  R      02
DO_SIG                 00000072  R      02
EXC_HNDLR_OVER         00000066  R      02
EXC_HNDLR_UNDER        00000057  R      02
EXP                 =  0000000C
EXPGTR                 00000020  R      02
MTH$$SIGNAL            ********    X    00
MTH$K_FLOOVEMAT        ********    X    00
MTH$K_FLOUNDMAT        ********    X    00
MTH$K_UNDEXP           ********    X    00
OTS$POWDJ              00000000  RG     02
PARTIAL                0000002F  R      02
POWDJX                 00000046  R      02
PSL$V_FU            =  00000006
SETUP                  00000086  R      02
SF$L_SAVE_PC        =  00000010
SF$W_SAVE_PSW       =  00000004
SQUAR                  00000024  R      02
SQUAR1                 00000029  R      02
SS$_CONTINUE           ********    X    00
SS$_FLTDIV             ********    X    00
SS$_FLTDIV_F           ********    X    00
SS$_FLTOVF             ********    X    00
SS$_FLTOVF_F           ********    X    00
UNDEFINED              00000047  R      02
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
_OTS$CODE	000000B6 (182.)	02 (2.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	33	00:00:00.11	00:00:00.56
Command processing	140	00:00:00.72	00:00:05.89
Pass 1	136	00:00:02.09	00:00:08.66
Symbol table sort	0	00:00:00.10	00:00:00.22
Pass 2	72	00:00:00.85	00:00:03.08
Symbol table output	4	00:00:00.04	00:00:00.10
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 389 00:00:03.93 00:00:18.54

The working set limit was 900 pages.
9162 bytes (18 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 106 non-local and 0 local symbols.
380 source lines were read in Pass 1, producing 13 object records in Pass 2.
11 pages of virtual memory were used to define 10 macros.

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

Macro library name	Macros defined
_255DUA28:[SYSLIB]STARLET.MLB;2	6

148 GETS were required to define 6 macros.

There were no errors, warnings or information messages.

```
MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS$:OTSPOWDJ/OBJ=OBJ$:OTSPOWDJ MSRC$:MTHJACKET/UPDATE=(ENH$:MTHJACKET)+MSRC
```

OTSMULCD
LIS

OTSPOWCGC
LIS

OTSDIVC
LIS

OTSPOWDD
LIS

OTSPOWCC
LIS

OTSPOWGG
LIS

OTSPOWGJ
LIS

OTSPOWCDJ
LIS

MTHTAN
LIS

MTHVECTOR
LIS

OTSDIVCG
LIS

OTSPOWCJ
LIS

OTSPOWDLJ
LIS

MTHTANH
LIS

OTSMULCG
LIS

OTSPOWCGJ
LIS

OTSPOWDJ
LIS

OTSDIVCD
LIS

OTSPOWCDC
LIS