F 13

```
MMM         MMM TTTTTTTTTTTTTTT HHH         HHH RRRRRRRRRRRR    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 RRRRRRRRRRRR          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
```

```
_$

Sy
--
MT
MT
MT

MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT
MT

MT
MT
MT
MT
MT
MT
MT

MT
MT
MT
MT
MT
MT
MT
```

```
**FILE**ID**OTSPOWCDJ                         G 9
```

```
  OOOOOO    TTTTTTTTTT    SSSSSSSS  PPPPPPP       OOOOOO    WW      WW    CCCCCCCC  DDDDDDD             JJ
  OOOOOO    TTTTTTTTTT    SSSSSSSS  PPPPPPP       OOOOOO    WW      WW    CCCCCCCC  DDDDDDD             JJ
OO      OO      TT      SS          PP      PP  OO      OO  WW      WW  CC          DD      DD          JJ
OO      OO      TT      SS          PP      PP  OO      OO  WW      WW  CC          DD      DD          JJ
OO      OO      TT      SS          PP      PP  OO      OO  WW      WW  CC          DD      DD          JJ
OO      OO      TT      SS          PP      PP  OO      OO  WW      WW  CC          DD      DD          JJ
OO      OO      TT        SSSSSS    PPPPPPPP    OO      OO  WW      WW  CC          DD      DD          JJ
OO      OO      TT        SSSSSS    PPPPPPPP    OO      OO  WW      WW  CC          DD      DD          JJ
OO      OO      TT              SS  PP          OO      OO  WW  WW  WW  CC          DD      DD  JJ      JJ
OO      OO      TT              SS  PP          OO      OO  WW  WW  WW  CC          DD      DD  JJ      JJ
OO      OO      TT              SS  PP          OO      OO  WWWW  WWWW  CC          DD      DD  JJ      JJ
OO      OO      TT              SS  PP          OO      OO  WWWW  WWWW  CC          DD      DD  JJ      JJ      ....
  OOOOOO        TT      SSSSSSSS    PP            OOOOOO    WW      WW    CCCCCCCC  DDDDDDD       JJJJJ         ....
  OOOOOO        TT      SSSSSSSS    PP            OOOOOO    WW      WW    CCCCCCCC  DDDDDDD       JJJJJ         ....

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
```

```
                                             H  9
OTS$POWCDJ               - D COMPLEX*16 ** INTEGER*4 power routin 16-SEP-1984 01:55:48  VAX/VMS Macro V04-00        Page   0
Table of contents

    (2)        47        HISTORY         ; Detailed Current Edit History
    (3)        56        DECLARATIONS
    (4)        90        OTS$POWCDJ_R3 - D COMPLEX*16 ** INTEGER*4
```

```
0000          .TITLE  OTS$POWCDJ - D COMPLEX*16 ** INTEGER*4 power routine
0000          .IDENT  /1-003/           ; File OTSPOWCDJ.MAR  Edit: SBL1003
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000    ;
0000    ;
0000    ; FACILITY: Language support library - user callable
0000    ;++
0000    ; ABSTRACT:
0000    ;
0000    ;       D COMPLEX*16 base to INTEGER*4 power.
0000    ;       Floating overflow can occur.
0000    ;       Undefined exponentiation can occur if
0000    ;       base = (0.,0.) and exp <=0
0000    ;
0000    ;--
0000    ;
0000    ; VERSION: 1
0000    ;
0000    ; HISTORY:
0000    ; AUTHOR:
0000    ;       Steven B. Lionel, 27-July-1979
0000    ;
```

```
                         0000     47           .SBTTL  HISTORY            ; Detailed Current Edit History
                         0000     48
                         0000     49
                         0000     50  ; Edit History
                         0000     51  ; 1-001 - Adapted from OTS$POWCJ version 1-003.  SBL 27-July-1979
                         0000     52  ; 1-002 - Correct bug in testing for undefined result with negative powers.
                         0000     53  ;         SPR 11-35262 SBL 22-Jan-1981
                         0000     54  ; 1-003 - Use general mode addressing.  SBL 30-Nov-1981
```

```
    0000          .SBTTL  DECLARATIONS
    0000
    0000  ;
    0000  ; INCLUDE FILES:
    0000  ;
    0000
    0000  ; EXTERNAL SYMBOLS:
    0000  ;
    0000
    0000          .DSABL  GBL
    0000          .EXTRN  MTH$$SIGNAL                 ; Math error routine
    0000          .EXTRN  OTS$DIVCD_R3                ; COMPLEX division routine
    0000          .EXTRN  MTH$K_UNDEXP
    0000
    0000  ;
    0000  ; MACROS:
    0000  ;
    0000
    0000  ;
    0000  ; EQUATED SYMBOLS:
    0000  ;
    0000
    0000  ;
    0000  ; OWN STORAGE:
    0000  ;
    0000
    0000  ;
    0000  ; PSECT DECLARATIONS:
    0000  ;
    0000
00000000          .PSECT  _OTS$CODE PIC,SHR,LONG,EXE,NOWRT
    0000                                          ; program section for OTS$ code
    0000
```

```
                 0000         .SBTTL  OTS$POWCDJ_R3 - D COMPLEX*16 ** INTEGER*4
                 0000  ;++
                 0000  ; FUNCTIONAL DESCRIPTION:
                 0000  ;
                 0000  ;      D COMPLEX*16 result = D COMPLEX*16 base ** INTEGER*4 exponent
                 0000  ;      The COMPLEX result is given by:
                 0000  ;
                 0000  ;      base               exponent          result
                 0000  ;
                 0000  ;      any                  >0              PRODUCT (base * 2**i) where
                 0000  ;                                            i is each non-zero bit in
                 0000  ;                                            exponent.
                 0000  ;
                 0000  ;      (0., 0.)            <=0              Undefined exponentiation.
                 0000  ;
                 0000  ;      not (0., 0.)         <0              PRODUCT (base * 2**i) where
                 0000  ;                                            i is each non-zero bit in
                 0000  ;                                            |exponent|.
                 0000  ;
                 0000  ;      not (0., 0.)         =0              (1.0, 0.0)
                 0000  ;
                 0000  ;      Floating overflow can occur.
                 0000  ;      Undefined exponentiation occurs if base is 0 and
                 0000  ;      exponent is 0 or negative.
                 0000  ;
                 0000  ; CALLING SEQUENCE:
                 0000  ;
                 0000  ;      result.wdc.v = OTS$POWCDJ_R3 (base.rdc.v, exponent.rl.v)
                 0000  ;
                 0000  ; INPUT PARAMETERS:
       00000004  0000         base     = 4                  ; D COMPLEX*16 base passed by VALUE!
       00000014  0000         exponent = 20                 ; Longword integer exponent by value.
                 0000  ;
                 0000  ; IMPLICIT INPUTS:
                 0000  ;      NONE
                 0000  ;
                 0000  ; OUTPUT PARAMETERS:
                 0000  ;      NONE
                 0000  ;
                 0000  ; IMPLICIT OUTPUTS:
                 0000  ;      NONE
                 0000  ;
                 0000  ; FUNCTION VALUE:
                 0000  ;
                 0000  ;      THE D COMPLEX*16 result is returned in registers R0-R3.
                 0000  ;      This is a violation of the VAX calling standard, but is
                 0000  ;      excused for compiled code support routines.
                 0000
                 0000  ;
                 0000  ; SIDE EFFECTS:
                 0000  ;
                 0000  ;      Modifies registers R0-R3!
                 0000  ;      SS$_FLTOVF - Floating overflow
                 0000  ;      SIGNALs MTH$_UNDEXP (82 = ' UNDEFINED EXPONENTATION') if
                 0000  ;      base is 0 and exponent is 0 or negative.
                 0000  ;
                 0000  ;--
```

```
                     01F0 0000                .ENTRY  OTS$POWCDJ_R3, ^M<R4,R5,R6,R7,R8>
                          0002                                                  ; disable integer overflow
          54    04 AC  7D 0002                MOVQ    base(AP), R4              ; R4-R7 gets COMPLEX base
          56    0C AC  7D 0006                MOVQ    base+8(AP), R6
          58    14 AC  D0 000A                MOVL    exponent(AP), R8          ; R8 = longword exponent
                   03  18 000E                BGEQ    1$                        ; R8 = | exponent |
             58    58  CE 0010                MNEGL   R8, R8
          0F 58    00  E5 0013    1$:         BBCC    #0, R8, EVEN              ; branch if even and clear low bit
             50    54  70 0017                MOVD    R4, R0                    ; R0-R3 = initial result
             52    56  70 001A                MOVD    R6, R2
    58    58    FF 8F  9C 001D                ROTL    #-1, R8, R8               ; R8 = unsigned_exponent / 2
                   5E  13 0022                BEQL    DONE                      ; done if exponent was 1
                   2D  11 0024                BRB     SQUAR1                    ; else use rest of exponent
                          0026
                          0026    EVEN:
             50    08  70 0026                MOVD    #1, R0                    ; R0-R3 = initial result
                   52  7C 0029                CLRQ    R2                        ; (1.0, 0.0)
    58    58    FF 8F  9C 002B                ROTL    #-1, R8, R8               ; R8 = unsigned_exponent / 2
                   21  12 0030                BNEQ    SQUAR1                    ; branch if exponent not 0
                   54  73 0032                TSTD    R4                        ; exponent was 0, text RP(base)
                   4C  12 0034                BNEQ    DONE                      ; done if non-0, answer is 1.0
                   56  73 0036                TSTD    R6                        ; IP(base) better not be zero
                   48  12 0038                BNEQ    DONE                      ; it isn't return 1.0
                          003A
                          003A    UNDEFINED:
       50    01    0F  79 003A                ASHQ    #15, #1, R0               ; return R0-R3 = reserved operands
       52    01    0F  79 003E                ASHQ    #15, #1, R2
          7E    00'8F  9A 0042                MOVZBL  #MTH$K_UNDEXP, -(SP)      ; FORTRAN error number
  00000000'GF      01  FB 0046                CALLS   #1, G^MTH$$SIGNAL                 ; convert to 32-bit condition code
                          004D                                                  ; and SIGNAL MTH$_UNDEXP
                       04 004D                RET
                          004E
                          004E    SQUAR:
    58    58    FF 8F  78 004E                ASHL    #-1, R8, R8               ; R8 = |reduced exponent| / 2
                          0053    ;
                          0053    ; R4-R7 = square current base
                          0053    ;
                          0053    SQUAR1:
       7E    56    54  65 0053                MULD3   R4, R6, -(SP)             ; (SP) = tmp = RP(base)*IP(base)
             54    54  64 0057                MULD2   R4, R4                    ; R4-R5 = RP(base)**2
             56    56  64 005A                MULD2   R6, R6                    ; R6-R7 = IP(base)**2
             54    56  62 005D                SUBD2   R6, R4                    ; R4-R5 = RP(base)**2 - IP(base)**2
       56    8E    6E  61 0060                ADDD3   (SP), (SP)+, R6           ; R6-R7 = 2*(RP(base)*IP(base))
                E7 58  E9 0064                BLBC    R8, SQUAR                 ; branch if next exponent bit is 0
                          0067    ;
                          0067    ; R0-R3 = partial result * current power of base
                          0067    ;
       7E    56    50  65 0067                MULD3   R0, R6, -(SP)             ; (SP) = tmp = RP(part) * IP(base)
             50    54  64 006B                MULD2   R4, R0                    ; R0-R1 = RP(part) * RP(base)
       7E    56    52  65 006E                MULD3   R2, R6, -(SP)             ; (SP) = tmp = IP(part) * IP(base)
             50    8E  62 0072                SUBD2   (SP)+, R0                 ; R0-R1 = RP(part)*RP(base)-IP(part)*IP(base
             52    54  64 0075                MULD2   R4, R2                    ; R2-R3 = IP(part)*RP(base)
             52    8E  60 0078                ADDD2   (SP)+, R2                 ; R2-R3 = IP(part)*RP(base)+RP(part)*IP(base
    58    58    FF 8F  78 007B                ASHL    #-1, R8, R8               ; R8 = |reduced exponent| / 2
                   D1  12 0080                BNEQ    SQUAR1                    ; loop if more exponent bits left
                          0082    DONE:
                14 AC  D5 0082                TSTL    exponent(AP)              ; test exponent sign
```

```
                 1A    18  0085          BGEQ    POWCDJ              ; done if positive
                 50    73  0087          TSTD    R0                  ; test RP(result)
                 04    12  0089          BNEQ    RECIP               ; if non-0, OK to take reciprocal
                 52    73  008B          TSTD    R2                  ; RP(result) was 0, test IP(result)
                 AB    13  008D          BEQL    UNDEFINED           ; undefined (0.0+0.0i) ** -n
                           008F  RECIP:
           7E    52    7D  008F          MOVQ    R2, -(SP)           ; second arg pair is divisor
           7E    50    7D  0092          MOVQ    R0, -(SP)
                 7E    7C  0095          CLRQ    -(SP)               ; push (1.0,0.0) on stack
           7E    08    70  0097          MOVD    #1, -(SP)
  00000000'GF    08    FB  009A          CALLS   #8, G^OTS$DIVCD_R3  ; R0-R3 = reciprocal
                           00A1  POWCDJ:
                       04  00A1          RET                         ; result in R0-R3
                           00A2
                           00A2          .END
```

```
                                                        B 10
OTS$POWCDJ                      - D COMPLEX*16 ** INTEGER*4 power routin 16-SEP-1984 01:55:48  VAX/VMS Macro V04-00         Page   7
Symbol table                                                              6-SEP-1984 11:27:49  [MTHRTL.SRC]OTSPOWCDJ.MAR;1     (5)

BASE                   = 00000004
DONE                     00000082 R     01
EVEN                     00000026 R     01
EXPONENT               = 00000014
MTH$$SIGNAL              ********   X   00
MTH$K_UNDEXP             ********   X   00
OTS$DIVCD_R3             ********   X   00
OTS$POWCDJ_R3            00000000 RG    01
POWCDJ                   000000A1 R     01
RECIP                    0000008F R     01
SQUAR                    0000004E R     01
SQUAR1                   00000053 R     01
UNDEFINED                0000003A R     01

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

PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (     0.)  00 (   0.)  NOPIC    USR    CON    ABS    LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
_OTS$CODE                       000000A2  (   162.)  01 (   1.)    PIC    USR    CON    REL    LCL   SHR   EXE   RD   NOWRT NOVEC LONG

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

Phase                        Page faults    CPU Time       Elapsed Time
-----                        -----------    --------       ------------
Initialization                       29    00:00:00.09    00:00:00.97
Command processing                  124    00:00:00.45    00:00:02.59
Pass 1                               76    00:00:00.60    00:00:01.93
Symbol table sort                     0    00:00:00.00    00:00:00.04
Pass 2                               53    00:00:00.48    00:00:01.53
Symbol table output                   2    00:00:00.02    00:00:00.02
Psect synopsis output                 2    00:00:00.01    00:00:00.02
Cross-reference output                0    00:00:00.00    00:00:00.00
Assembler run totals                288    00:00:01.65    00:00:07.10

The working set limit was 900 pages.
3177 bytes (7 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 13 non-local and 1 local symbols.
219 source lines were read in Pass 1, producing 11 object records in Pass 2.
0 pages of virtual memory were used to define 0 macros.

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

Macro library name                                 Macros defined
------------------                                 --------------
_$255$DUA28:[SYSLIB]STARLET.MLB;2                        0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.
```

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

0264 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

OTSMULCD
LIS

OTSPOWCGC
LIS

OTSDIVC
LIS

OTSPOWDD
LIS

OTSPOWCC
LIS

OTSPOWGG
LIS

OTSPOWGJ
LIS

OTSPOWCDJ
LIS

MTHTAN
LIS

MTHVECTOR
LIS

OTSDIVCG
LIS

OTSPOWCJ
LIS

OTSPOWDLU
LIS

MTHTANH
LIS

OTSMULCG
LIS

OTSPOWCGJ
LIS

OTSPOWDJ
LIS

OTSDIVCD
LIS

OTSPOWCDC
LIS