```
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
```

```
**FILE**ID**OTSPOWCC                                F  8
```

```
  000000    TTTTTTTTTT    SSSSSSSS  PPPPPPPP      000000    WW      WW    CCCCCCCC    CCCCCCCC
  000000    TTTTTTTTTT    SSSSSSSS  PPPPPPPP      000000    WW      WW    CCCCCCCC    CCCCCCCC
00      00      TT      SS          PP      PP  00      00  WW      WW  CC          CC
00      00      TT      SS          PP      PP  00      00  WW      WW  CC          CC
00      00      TT      SS          PP      PP  00      00  WW      WW  CC          CC
00      00      TT      SS          PP      PP  00      00  WW      WW  CC          CC
00      00      TT        SSSSSS    PPPPPPPP    00      00  WW      WW  CC          CC
00      00      TT        SSSSSS    PPPPPPPP    00      00  WW      WW  CC          CC
00      00      TT              SS  PP          00      00  WW  WW  WW  CC          CC
00      00      TT              SS  PP          00      00  WW  WW  WW  CC          CC
00      00      TT              SS  PP          00      00  WWWW  WWWW  CC          CC              ....
00      00      TT              SS  PP          00      00  WWWW  WWWW  CC          CC              ....
  000000        TT      SSSSSSSS    PP            000000    WW      WW    CCCCCCCC    CCCCCCCC      ....
  000000        TT      SSSSSSSS    PP            000000    WW      WW    CCCCCCCC    CCCCCCCC      ....

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            .TITLE  OTS$POWCC - COMPLEX*8 ** COMPLEX*8 power routine
0000            .IDENT  /1-004/                 ; File: OTSPOWCC.MAR
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    ; FACILITY: Language support library - user callable
0000    ;
0000    ; ABSTRACT:
0000    ;
0000    ;       COMPLEX*8 base to COMPLEX*8 power giving COMPLEX*8 result.
0000    ;
0000    ; ENVIRONMENT: User Mode, AST Reentrant
0000    ;
0000    ;--
0000    ; AUTHOR: Steven B. Lionel, CREATION DATE: 24-Oct-1978: Version 0
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ; SBL 24-Oct-1978, VERSION 00
0000    ; 1-001 - Original
0000    ; 1-002 - Standardized version number format, with three digits in
0000    ;               the edit number.  JBS 16-NOV-78
0000    ; 1-003 - Add "_" to the PSECT directive.  JBS 22-DEC-78
0000    ; 1-004 - Change shared external references to G^ RNH 25-Sep-81
```

```
                  0000          .SBTTL  DECLARATIONS
                  0000    ;
                  0000    ; INCLUDE FILES:
                  0000    ;
                  0000
                  0000    ;
                  0000    ; EXTERNAL DECLARATIONS:
                  0000    ;
                  0000          .GLOBL  MTH$CEXP                ; Complex exponentiation
                  0000          .GLOBL  MTH$CLOG                ; Complex logarithm
                  0000    ;
                  0000    ; MACROS:
                  0000    ;
                  0000
                  0000    ;
                  0000    ; EQUATED SYMBOLS:
                  0000    ;
         00000004 0000          base = 4                        ; base input - by value
         0000000C 0000          exp = 12                        ; exponent input - by value
                  0000    ;
                  0000    ; OWN STORAGE:
                  0000    ;
                  0000
                  0000    ;
                  0000    ; PSECT DECLARATIONS:
                  0000    ;
              00000000          .PSECT _OTS$CODE PIC, USR, CON, REL, LCL, SHR, -
                  0000                          EXE, RD, NOWRT, LONG
                  0000
```

```
0000            .SBTTL  OTS$POWCC  - COMPLEX*8 ** COMPLEX*8 power routine
0000     ;++
0000     ; FUNCTIONAL DESCRIPTION:
0000     ;
0000     ;       OTS$POWCC evaluates the result of taking a complex base
0000     ;       to a complex power.  The ANS FORTRAN X3.9-1978 standard defines
0000     ;       complex exponentiation as:
0000     ;
0000     ;       x ** y = CEXP( y * CLOG(x))
0000     ;
0000     ;       where x and y are type COMPLEX.
0000     ;
0000     ;       The arguments of OTS$POWCC are CALL BY VALUE.
0000     ;
0000     ; CALLING SEQUENCE:
0000     ;
0000     ;       power.wfc.v = OTS$POWCC (base.rfc.v, exponent.rfc.v)
0000     ;
0000     ; INPUT PARAMETERS:
0000     ;
0C00     ;       Both base and exponent are COMPLEX*8 numbers, each consisting
0000     ;       of a REAL*4 real part and a REAL*4 imaginary part.  Both are
0000     ;       CALL BY VALUE.
0000     ;
0000     ; IMPLICIT INPUTS:
0000     ;
0000     ;       NONE
0000     ;
0000     ; OUTPUT PARAMETERS:
0000     ;
0000     ;       NONE
0000     ;
0000     ; IMPLICIT OUTPUTS:
0000     ;
0000     ;       NONE
0000     ;
0000     ; FUNCTION VALUE:
0000     ;
0000     ;       The COMPLEX*8 (REAL*4, REAL*4) result of taking the
0000     ;       COMPLEX base to the COMPLEX exponent power.
0000     ;
0000     ; SIDE EFFECTS:
0000     ;
0000     ;       Possible error signals are:
0000     ;
0000     ;        MTH$_INVARGMAT if base is (0.,0.).
0000     ;        MTH$_FLOOVEMAT if floating overflow occurs.
0000     ;        MTH$_SINCOSSIG if absolute value of the imaginary part of
0000     ;                       (exponent * CLOG(base)) > PI*2**30.
0000     ;        SS$_ROPRAND    if reserved floating operand is fetched.
0000     ;--
```

```
                     003C  0000           .ENTRY  OTS$POWCC, ^M<R2,R3,R4,R5>      ; disable integer ovflo
                           0002
                           0002           MTH$FLAG_JACKET                  ; establish math error handler
                           0002
   6D   00000000'GF   9E   0002           MOVAB   G^MTH$$JACKET_HND, (FP)
                           0009                                            ; set handler address to jacket
                           0009                                            ; handler
                           0009
                           0009
        52   0C AC    7D   0009           MOVQ    exp(AP), R2              ; put exponent in R2, R3
                           000D                                            ; later operations will check
                           000D                                            ; for reserved operands
                           000D
                           000D     ;+
                           000D     ;     Get complex logarithm of base
                           000D     ;-
             04 AC    DF   000D           PUSHAL  base(AP)                 ; address of base
   00000000'GF   01   FB   0010           CALLS   #1, G^MTH$CLOG           ; R0,R1 get LOG(base)
                           0017                                            ; call by reference
                           0017
                           0017     ;+
                           0017     ;     CLOG(base) is in R0, R1.  Multiply by exponent.
                           0017     ;     R0,R1 = CLOG(base) = a+bi
                           0017     ;     R2,R3 = exp  = c+di
                           0017     ;
                           0017     ;     Complex multiplication defined as:
                           0017     ;
                           0017     ;     real part = ac-bd
                           0017     ;     imaginary part = ad+bc
                           0017     ;-
                           0017
      54   53   50    45   0017           MULF3   R0, R3, R4               ; R4 = ad
           50   52    44   001B           MULF2   R2, R0                   ; R0 = ac
      55   53   51    45   001E           MULF3   R1, R3, R5               ; R5 = bd
           50   55    42   0022           SUBF2   R5, R0                   ; R0 = ac-bd
           51   52    44   0025           MULF2   R2, R1                   ; R1 = bc
           51   54    40   0028           ADDF2   R4, R1                   ; R1 = ad+bc
                           002B
                           002B     ;+
                           002B     ;     Now compute CEXP(product)
                           002B     ;-
                           002B
           7E   50    7D   002B           MOVQ    R0, -(SP)                ; put product (R0,R1) on stack
                5E    DD   002E           PUSHL   SP                       ; address of arguments
   00000000'GF   01   FB   0030           CALLS   #1, G^MTH$CEXP           ; R0, R1 get EXP(product)
                           0037                                            ; call by reference
                           0037
                      04   0037           RET                              ; all done, exit
                           0038           .END
```

```
BASE            = 00000004
EXP             = 0000000C
MTH$$JACKET_HND   ********   X    01
MTH$CEXP          ********   G    00
MTH$CLOG          ********   G    00
OTS$POWCC         00000000 RG     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                         00000038  (   56.)  01 (  1.)    PIC    USR    CON    REL    LCL   SHR   EXE   RD   NOWRT NOVEC LONG

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

Phase                        Page faults    CPU Time       Elapsed Time
-----                        -----------    --------       ------------
Initialization                        32    00:00:00.08    00:00:01.80
Command processing                   138    00:00:00.59    00:00:05.14
Pass 1                                87    00:00:00.65    00:00:03.71
Symbol table sort                      0    00:00:00.00    00:00:00.00
Pass 2                                45    00:00:00.48    00:00:04.60
Symbol table output                    2    00:00:00.01    00:00:00.05
Psect synopsis output                  2    00:00:00.02    00:00:00.03
Cross-reference output                 0    00:00:00.00    00:00:00.00
Assembler run totals                 308    00:00:01.85    00:00:15.34

The working set limit was 900 pages.
2597 bytes (6 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 6 non-local and 0 local symbols.
235 source lines were read in Pass 1, producing 11 object records in Pass 2.
1 page of virtual memory was used to define 1 macro.

                                            +--------------------------+
                                            ! 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$:OTSPOWCC/OBJ=OBJ$:OTSPOWCC MSRC$:MTHJACKET/UPDATE=(ENH$:MTHJACKET)+MSRC
```

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