```
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBB         BBB   AAA         AAA   SSS          RRR         RRR         TTT        LLL
BBB         BBB   AAA         AAA   SSS          RRR         RRR         TTT        LLL
BBB         BBB   AAA         AAA   SSS          RRR         RRR         TTT        LLL
BBB         BBB   AAA         AAA   SSS          RRR         RRR         TTT        LLL
BBB         BBB   AAA         AAA   SSS          RRR         RRR         TTT        LLL
BBB         BBB   AAA         AAA   SSS          RRR         RRR         TTT        LLL
BBBBBBBBBBBB      AAA         AAA     SSSSSSSSS  RRRRRRRRRRR             TTT        LLL
BBBBBBBBBBBB      AAA         AAA     SSSSSSSSS  RRRRRRRRRRR             TTT        LLL
BBBBBBBBBBBB      AAA         AAA     SSSSSSSSS  RRRRRRRRRRR             TTT        LLL
BBB         BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR               TTT        LLL
BBB         BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR               TTT        LLL
BBB         BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR               TTT        LLL
BBB         BBB   AAA         AAA           SSS  RRR      RRR            TTT        LLL
BBB         BBB   AAA         AAA           SSS  RRR   RRR               TTT        LLL
BBB         BBB   AAA         AAA           SSS  RRR      RRR            TTT        LLL
BBBBBBBBBBBB      AAA         AAA   SSSSSSSSSSS     RRR         RRR      TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB      AAA         AAA   SSSSSSSSSSS     RRR         RRR      TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB      AAA         AAA   SSSSSSSSSSS     RRR         RRR      TTT        LLLLLLLLLLLLLLL
```

FILEID**BASPOWGG

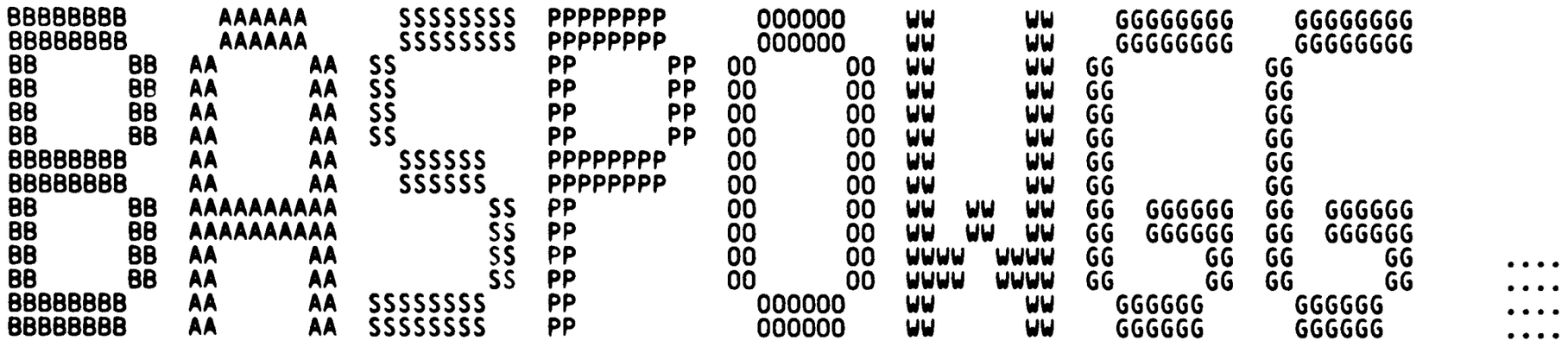

Table of contents

```
0000     1          .TITLE  BAS$POWGG               ; BASIC gfloat ** gfloat routine
0000     2          .IDENT  /1-001/                 ; File  BASPOWGG.MAR Edit:PLL1001
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.                                                             *
^000    21 ;*                                                                           *
U000    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 ;++
0000    30 ; FACILITY: Basic Support Library
0000    31 ;
0000    32 ; ABSTRACT:
0000    33 ;
0000    34 ;       This module contains entry points to support exponentiation
0000    35 ;       (** or ^) in BASIC-PLUS-2 for DOUBLE ** DOUBLE.
0000    36 ;
0000    37 ; ENVIRONMENT: User Mode, AST Reentrant
0000    38 ;
0000    39 ;--
0000    40 ; AUTHOR: P. Levesque   , CREATION DATE: 5-Oct-81
0000    41 ;
0000    42 ; MODIFIED BY:
0000    43 ;
0000    44 ; 1-001 - Original
```

```
                        46          .SBTTL  DECLARATIONS
                        47 ;
                        48 ; INCLUDE FILES:
                        49 ;
                        50
                        51 ;
                        52 ; EXTERNAL DECLARATIONS:
                        53 ;
                        54          .DSABL  GBL                     ; Prevent undeclared
                        55                                          ; symbols from being
                        56                                          ; automatically global.
                        57
                        58          .EXTRN  OTS$POWGG               ; OTS$ gfloat ** gfloat exponentiation
                        59          .EXTRN  OTS$POWGJ               ; OTS$ gfloat ** int exponentation
                        60          .EXTRN  BAS$K_DIVBY_ZER         ; Divide by Zero
                        61          .EXTRN  BAS$K_ILLARGLOG         ; Illegal argument in LOG
                        62          .EXTRN  BAS$$STOP               ; Error reporting routine
                        63
                        64 ;
                        65 ; MACROS:
                        66 ;
                        67
                        68 ;
                        69 ; EQUATED SYMBOLS:
                        70 ;
                        71
                        72 ;
                        73 ; OWN STORAGE:
                        74 ;
                        75
                        76 ;
                        77 ; PSECT DECLARATIONS:
                        78 ;
                  00000000  79          .PSECT _BAS$CODE PIC, USR, CON, REL, LCL, SHR, -
                        80                      EXE, RD, NOWRT, LONG
                        81
```

```
                0000        .SBTTL  BAS$POWGG  - BASIC gfloat ** gfloat
                0000   ;++
                0000   ; FUNCTIONAL DESCRIPTION:
                0000   ;
                0000   ;       This routine takes BASE ** EXP, using the following table
                0000   ;       for unusual cases:
                0000   ;
                0000   ;       BASE > 0                        Call OTS$POWGG, normal case.
                0000   ;       BASE = 0, EXP > 0               Return 0.0.
                0000   ;       BASE = 0, EXP = 0               Return 1.0.
                0000   ;       BASE = 0, EXP < 0               Error: divide by zero
                0000   ;       BASE < 0, EXP even integer      Call OTS$POWGJ with -BASE
                0000   ;       BASE < 0, EXP odd integer       Call OTS$POWGJ with -BASE, negate result
                0000   ;       BASE < 0, EXP not integer       Error: illegal argument in LOG.
                0000   ;
                0000   ; CALLING SEQUENCE:
                0000   ;
                0000   ;       CALL result.wg.v = BAS$POWGG (base.rg.v, exponent.rg.v)
                0000   ;
                0000   ; INPUT PARAMETERS:
                0000   ;
      00000004  0000        base = 4
      0000000C  0000        exponent = 12
                0000   ;
                0000   ; IMPLICIT INPUTS:
                0000   ;
                0000   ;       NONE
                0000   ;
                0000   ; OUTPUT PARAMETERS:
                0000   ;
                0000   ;       NONE
                0000   ;
                0000   ; IMPLICIT OUTPUTS:
                0000   ;
                0000   ;       NONE
                0000   ;
                0000   ; FUNCTION VALUE:
                0000   ; COMPLETION CODES:
                0000   ;
                0000   ;       gfloat result of exponentiation
                0000   ;
                0000   ; SIDE EFFECTS:
                0000   ;
                0000   ;       Will signal Divide By Zero or Illegal argument in LOG if its
                0000   ;       arguments are bad, and OTS$POWGG and OTS$POWGJ may also signal.
                0000   ;
                0000   ;--
                0000
         0000'  0000   BAS$POWGG::     .MASK OTS$POWGG         ; Entry point
                0002                                           ; Since this routine uses no
                0002                                           ; registers and usually transfers
                0002                                           ; control to OTS$POWGG, we copy
                0002                                           ; its register save mask and then
                0002                                           ; JMP past its save mask and only
                0002                                           ; save the registers once
   04 AC 53FD   0002        TSTG    base(AP)                   ; Test base relationship to zero
      06   15   0006        BLEQ    1$                         ; If base leq 0, do case analysis
```

```
            00000002'GF   17  0008        JMP     G^OTS$POWGG+2           ; Transfer control to the OTS$
                              000E                                        ; routine to do exponentiation
                              000E  ;+
                              000E  ; Come here if the base is less than or equal to zero.  We must filter
                              000E  ; several special cases, as described above.
                              000E  ;-
                      32  13  000E  1$:   BEQL    4$                      ; Branch if base = 0
50  50  08  00   0C AC 54FD   0010        EMODG   exponent(AP), #0, #1, R0, R0
                      1D  12  0018        BNEQ    3$                      ; Branch if exponent is not integer
                              001A  ;+
                              001A  ; The base is less than zero and the exponent is an integer.
                              001A  ; BASIC defines this as working the same way as if an integer was
                              001A  ; in the expression (making a double variable which happens to
                              001A  ; contain an integer value equivalent to an integer variable).
                              001A  ;-
               50  0C AC 4AFD 001A        CVTGL   exponent(AP), R0        ; Convert exponent to integer
                      50  DD  001F        PUSHL   R0                      ; Save for even/odd test
                      50  DD  0021        PUSHL   R0                      ; Stack as parameter to OTS$POWGJ
               7E  04 AC 52FD 0023        MNEGG   base(AP), -(SP)         ; Stack -base also
            00000000'GF   03  FB  0028    CALLS   #3, G^OTS$POWGJ         ; Call integer power routines
                   04 8E  E9  002F        BLBC    (SP)+,2$                ; Branch if exponent even
                 50   50 52FD 0032        MNEGG   R0, R0                  ; Exponent odd, negate the result
                          04  0036  2$:   RET                             ; and return with it.
                              0037  ;+
                              0037  ; Come here if the base is less than zero but the exponent is not
                              0037  ; an integer.  BASIC defines this as an error.
                              0037  ;-
               7E   00'8F  9A 0037  3$:   MOVZBL  #BAS$K_ILLARGLOG, -(SP) ; Illegal Argument in LOG
            00000000'GF   01  FB  003B    CALLS   #1, G^BAS$$STOP         ; Never return.
                              0042  ;+
                              0042  ; Come here if the base is equal to zero.  The value we return depends
                              0042  ; upon the sign of the exponent.
                              0042  ;-
                  0C AC 53FD  0042  4$:   TSTG    exponent(AP)            ; Test the exponent against zero
                      0A  19  0046        BLSS    6$                      ; Branch if exponent lss 0
                      03  13  0048        BEQL    5$                      ; Branch if exponent is 0
                              004A  ;+
                              004A  ; Come here if the base is zero and the exponent is greater than zero.
                              004A  ; BASIC defines this as 0.0.
                              004A  ;-
                      50  7C  004A        CLRG    R0                      ; R0, R1 = 0.0
                          04  004C        RET                             ; Return to caller
                              004D  ;+
                              004D  ; Come here if the base is zero and the exponent is zero.  BASIC defines
                              004D  ; this as 1.0.
                              004D  ;-
                 50   08 50FD 004D  5$:   MOVG    #1, R0                  ; R0, R1 = 1.0
                          04  0051        RET                             ; Return to caller.
                              0052  ;+
                              0052  ; Come here if the base is zero and the exponent is less than zero.
                              0052  ; BASIC defines this as an error.
                              0052  ;-
               7E   00'8F  9A 0052  6$:   MOVZBL  #BAS$K_DIVBY_ZER, -(SP) ; Divide by zero
            00000000'GF   01  FB  0056    CALLS   #1, G^BAS$$STOP         ; Report error, never return.
                              005D  ;
                              005D        .END
```

```
BAS$$STOP           ********    X   00
BAS$K_DIVBY_ZER     ********    X   00
BAS$K_ILLARGLOG     ********    X   00
BAS$POWGG           00000000 RG     01
BASE              = 00000004
EXPONENT          = 0000000C
OTS$POWGG           ********    X   00
OTS$POWGJ           ********    X   00
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
_BAS$CODE	0000005D (93.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	33	00:00:00.09	00:00:00.61
Command processing	119	00:00:00.53	00:00:02.84
Pass 1	73	00:00:00.54	00:00:01.40
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	48	00:00:00.40	00:00:01.07
Symbol table output	3	00:00:00.02	00:00:00.01
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	280	00:00:01.60	00:00:05.96

```
The working set limit was 900 pages.
2217 bytes (5 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 8 non-local and 6 local symbols.
195 source lines were read in Pass 1, producing 8 object records in Pass 2.
0 pages of virtual memory were used to define 0 macros.
```

Macro library statistics

Macro library name	Macros defined
_255DUA28:[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$:BASPOWGG/OBJ=OBJ$:BASPOWGG MSRC$:BASPOWGG/UPDATE=(ENH$:BASPOWGG)
```

BASOPEN
LIS

BASPOWJJ
LIS

BASPOS
LIS

BASPOWDJ
LIS

BASOPENDE
LIS

BASPOWGG
LIS

BASPOWHH
LIS

BASPOWRJ
LIS

BASPOWII
LIS

BASPURIOB
LIS

BASPOWDD
LIS

BASOPENZE
LIS

BASPOWDR
LIS

BASPOWGJ
LIS

BASPOWRD
LIS

BASPOWHJ
LIS

BASPOWRR
LIS