M 1

```
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRR      TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRR      TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  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  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

L 11

```
**FILE**ID**BASPOWGJ
```

```
BBBBBBBB      AAAAAA      SSSSSSSS  PPPPPPPP      OOOOOO    WW      WW    GGGGGGGG          JJ
BBBBBBBB      AAAAAA      SSSSSSSS  PPPPPPPP      OOOOOO    WW      WW    GGGGGGGG          JJ
BB      BB  AA      AA  SS          PP      PP  OO      OO  WW      WW  GG                  JJ
BB      BB  AA      AA  SS          PP      PP  OO      OO  WW      WW  GG                  JJ
BB      BB  AA      AA  SS          PP      PP  OO      OO  WW      WW  GG                  JJ
BB      BB  AA      AA  SS          PP      PP  OO      OO  WW      WW  GG                  JJ
BBBBBBBB    AA      AA    SSSSSS    PPPPPPPP    OO      OO  WW      WW  GG                  JJ
BBBBBBBB    AA      AA    SSSSSS    PPPPPPPP    OO      OO  WW      WW  GG                  JJ
BB      BB  AAAAAAAAAA          SS  PP          OO      OO  WW  WW  WW  GG  GGGGGG  JJ      JJ
BB      BB  AAAAAAAAAA          SS  PP          OO      OO  WW  WW  WW  GG  GGGGGG  JJ      JJ
BB      BB  AA      AA          SS  PP          OO      OO  WWWW  WWWW  GG      GG  JJ      JJ      ....
BB      BB  AA      AA          SS  PP          OO      OO  WWWW  WWWW  GG      GG  JJ      JJ      ....
BBBBBBBB    AA      AA  SSSSSSSS    PP              OOOOOO    WW      WW    GGGGGG      JJJJJJ          ....
BBBBBBBB    AA      AA  SSSSSSSS    PP              OOOOOO    WW      WW    GGGGGG      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
```

L

Table of contents

```
0000            .TITLE  BAS$POWGJ                 ; BASIC gfloat ** longword routine
0000            .IDENT  /1-002/                   ; File: BASPOWGJ.MAR Edit:PLL1002
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: Basic Support Library
0000    ;
0000    ; ABSTRACT:
0000    ;
0000    ;       This module contains entry points to support exponentiation
0000    ;       (** or ^) in BASIC-PLUS-2 for GFLOAT ** LONGWORD.
0000    ;
0000    ; ENVIRONMENT: User Mode, AST Reentrant
0000    ;
0000    ;--
0000    ; AUTHOR: P. Levesque   , CREATION DATE: 5-Oct-81
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ; 1-001 - Original
0000    ; 1-002 - Fix typo that references POWGJ.  PLL 23-Mar-1982
```

```
    0000           .SBTTL  DECLARATIONS
    0000  ;
    0000  ; INCLUDE FILES:
    0000  ;
    0000
    0000  ;
    0000  ; EXTERNAL DECLARATIONS:
    0000  ;
    0000           .DSABL  GBL                         ; Prevent undeclared
    0000                                               ; symbols from being
    0000                                               ; automatically global.
    0000
    0000           .EXTRN  OTS$POWGJ                   ; OTS$ gfloat ** int exponentation
    0000           .EXTRN  BAS$K_DIVBY_ZER             ; Divide by Zero
    0000           .EXTRN  BAS$$STOP                   ; Error reporting routine
    0000
    0000  ;
    0000  ; MACROS:
    0000  ;
    0000
    0000  ;
    0000  ; EQUATED SYMBOLS:
    0000  ;
    0000
    0000  ;
    0000  ; OWN STORAGE:
    0000  ;
    0000
    0000  ;
    0000  ; PSECT DECLARATIONS:
    0000  ;
00000000           .PSECT _BAS$CODE PIC, USR, CON, REL, LCL, SHR, -
    0000                           EXE, RD, NOWRT, LONG
    0000
```

```
                       0000          .SBTTL  BAS$POWGJ  - BASIC gfloat ** long
                       0000  ;++
                       0000  ; FUNCTIONAL DESCRIPTION:
                       0000  ;
                       0000  ;       This routine takes BASE ** EXP, using the following table
                       0000  ;       for unusual cases:
                       0000  ;
                       0000  ;       BASE > 0                        Call OTS$POWGJ, 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              Call OTS$POWGJ with -BASE
                       0000  ;       BASE < 0, EXP odd               Call OTS$POWGJ with -BASE, negate result
                       0000  ;
                       0000  ; CALLING SEQUENCE:
                       0000  ;
                       0000  ;       CALL result.wg.v = BAS$POWGJ (base.rg.v, exponent.rl.v)
                       0000  ;
                       0000  ; INPUT PARAMETERS:
                       0000  ;
              00000004 0000          base = 4
              0000000C 0000          exponent = 12
                       0000  ;
                       0000  ; IMPLICIT INPUTS:
                       0000  ;
                       0000  ;       NONE
                       0000  ;
                       0000  ; OUTPUT PARAMETERS:
                       0000  ;
                       0000  ;       NONE
                       0000  ;
                       0000  ; IMPLICIT OUTPUTS:
                       0000  ;
                       0000  ;       NONE
                       0000  ;
                       0000  ; FUNCTION VALUE:
                       0000  ; COMPLETION CODES:
                       0000  ;
                       0000  ;       gfloat result of exponentiation
                       0000  ;
                       0000  ; SIDE EFFECTS:
                       0000  ;
                       0000  ;       Will signal Divide By Zero if its arguments are bad,
                       0000  ;       and  OTS$POWGJ may also signal.
                       0000  ;
                       0000  ;--
                       0000
                 0000' 0000  BAS$POWGJ::     .MASK OTS$POWGJ         ; Entry point
                       0002                                          ; Since this routine uses no
                       0002                                          ; registers and usually transfers
                       0002                                          ; control to OTS$POWGJ, we copy
                       0002                                          ; its register save mask and then
                       0002                                          ; JMP past its save mask and only
                       0002                                          ; save the registers once
            04 AC 53FD 0002          TSTG    base(AP)                ; Test base relationship to zero
               06   15 0006          BLEQ    1$                      ; If base leq 0, do case analysis
       00000002'GF  17 0008          JMP     G^OTS$POWGJ+2           ; Transfer control to the OTS$
```

```
                          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      ;-
                  18   13 000E  1$:     BEQL    4$                    ; Branch if base = 0
               0C AC   DD 0010          PUSHL   exponent(AP)          ; Stack EXP as parameter to OTS$POWGJ
           7E  04 AC 52FD 0013          MNEGG   base(AP), -(SP)       ; Stack -base also
    00000000'GF   03   FB 0018          CALLS   #3, G^OTS$POWGJ       ; Call integer power routines
            04 0C AC   E9 001F          BLBC    exponent(AP),2$       ; Branch if exponent even
           50     50 52FD 0023          MNEGG   R0, R0                ; Exponent odd, negate the result
                       04 0027  2$:     RET                           ; and return with it.
                          0028      ;+
                          0028      ; Come here if the base is equal to zero.  The value we return depends
                          0028      ; upon the sign of the exponent.
                          0028      ;-
               0C AC   D5 0028  4$:     TSTL    exponent(AP)          ; Test the exponent against zero
                  0A   19 002B          BLSS    6$                    ; Branch if exponent lss 0
                  03   13 002D          BEQL    5$                    ; Branch if exponent is 0
                          002F      ;+
                          002F      ; Come here if the base is zero and the exponent is greater than zero.
                          002F      ; BASIC defines this as 0.0.
                          002F      ;-
                  50   7C 002F          CLRG    R0                    ; R0, R1 = 0.0
                       04 0031          RET                           ; Return to caller
                          0032      ;+
                          0032      ; Come here if the base is zero and the exponent is zero.  BASIC defines
                          0032      ; this as 1.0.
                          0032      ;-
           50     08 50FD 0032  5$:     MOVG    #1, R0                ; R0, R1 = 1.0
                       04 0036          RET                           ; Return to caller.
                          0037      ;+
                          0037      ; Come here if the base is zero and the exponent is less than zero.
                          0037      ; BASIC defines this as an error.
                          0037      ;-
       7E   00'8F      9A 0037  6$:     MOVZBL  #BAS$K_DIVBY_ZER, -(SP) ; Divide by zero
    00000000'GF   01   FB 003B          CALLS   #1, G^BAS$$STOP       ; Report error, never return.
                          0042      ;
                          0042          .END
```

```
BAS$$STOP          ********    X   00
BAS$K_DIVBY_ZER    ********    X   00
BAS$POWGJ          00000000 RG     01
BASE             = 00000004
EXPONENT         = 0000000C
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                        00000042  (    66.)  01 (   1.)    PIC    USR    CON    REL    LCL   SHR   EXE   RD  NOWRT NOVEC LONG

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

Pha'.e                     Page faults    CPU Time        Elapsed Time
---- --                    -----------    --------        ------------
Initialization                     29    00:00:00.08     00:00:00.32
Command processing                114    00:00:00.51     00:00:01.74
Pass 1                             73    00:00:00.45     00:00:01.18
Symbol table sort                   0    00:00:00.00     00:00:00.00
Pass 2                             44    00:00:00.38     00:00:00.97
Symbol table output                 2    00:00:00.01     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              266    00:00:01.46     00:00:04.25

The working set limit was 900 pages.
1894 bytes (4 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 5 local symbols.
177 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
------------------                                  --------------
_$255$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$:BASPOWGJ/OBJ=OBJ$:BASPOWGJ MSRC$:BASPOWGJ/UPDATE=(ENH$:BASPOWGJ)
```

0029 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

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