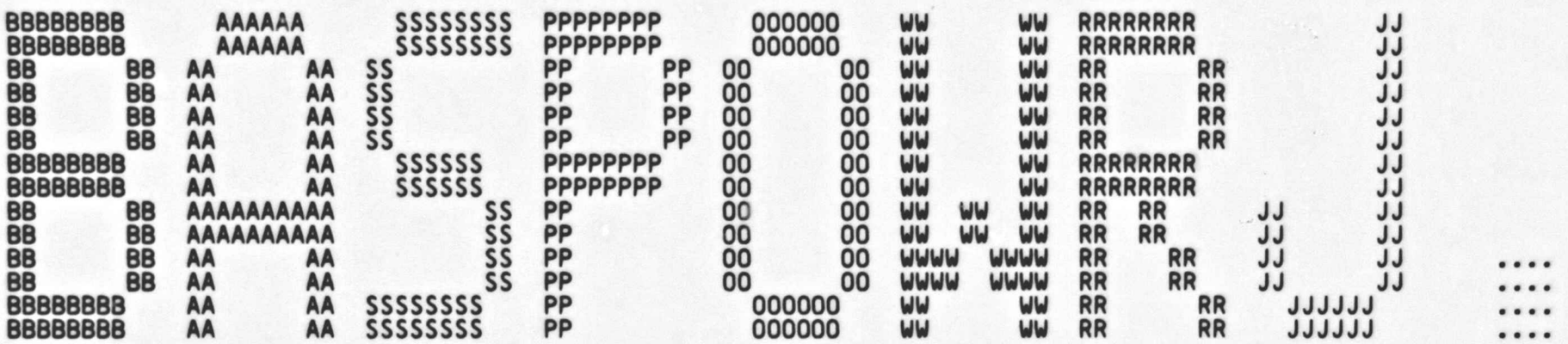
```
BBBBBBBBBBBB        AAAAAAAAA       SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA       SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA       SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBB          BBB   AAA         AAA  SSS          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**BASPOWRJ



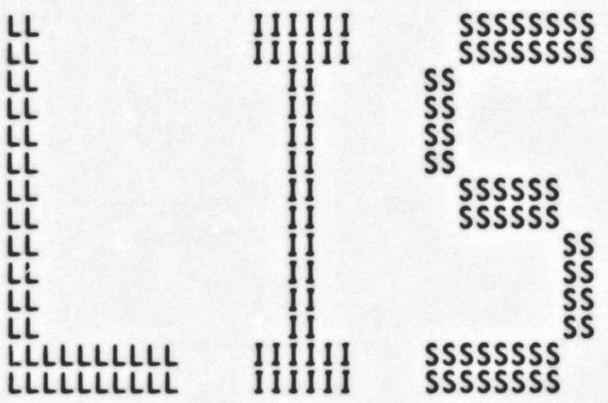


Table of contents

```
0000     1          .TITLE  BAS$POWRJ                ; BASIC real ** longword routine
0000     2          .IDENT  /1-005/          ; File: BASPOWRJ.MAR Edit: RNH1005
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 ;++
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 REAL ** LONGWORD.
0000    36 ;
0000    37 ; ENVIRONMENT: User Mode, AST Reentrant
0000    38 ;
0000    39 ;--
0000    40 ; AUTHOR: R. Will        , CREATION DATE: 22-NOV-78
0000    41 ;
0000    42 ; MODIFIED BY:
0000    43 ;
0000    44 ; R. Will,        : VERSION 01
0000    45 ; 1-01  - Original
0000    46 ; 1-02  - Redo comments, JMP instead of BRW.    RW 5-Dec-78
0000    47 ; 1-003 - Add "," to the PSECT directive.  JBS 22-DEC-78
0000    48 ; 1-004 - Redo the case analysis for base leq 0 for compatability
0000    49 ;          with the PDP-11.  JBS 24-APR-1979
0000    50 ; 1-005 - 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          .DSABL  GBL                       ; Prevent undeclared
    0000                                            ; symbols from being
    0000                                            ; automatically global.
    0000
    0000          .EXTRN  OTS$POWRJ                 ; OTS$ double ** 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     87          .SBTTL  BAS$POWRJ  - BASIC double ** long
                     0000     88 ;++
                     0000     89 ; FUNCTIONAL DESCRIPTION:
                     0000     90 ;
                     0000     91 ;       This routine takes BASE ** EXP, using the following table
                     0000     92 ;       for unusual cases:
                     0000     93 ;
                     0000     94 ;       BASE > 0                            Call OTS$POWRJ, normal case.
                     0000     95 ;       BASE = 0, EXP > 0                   Return 0.0.
                     0000     96 ;       BASE = 0, EXP = 0                   Return 1.0.
                     0000     97 ;       BASE = 0, EXP < 0                   Error: divide by zero
                     0000     98 ;       BASE < 0, EXP even                  Call OTS$POWRJ with -BASE
                     0000     99 ;       BASE < 0, EXP odd                   Call OTS$POWRJ with -BASE, negate result
                     0000    100 ;
                     0000    101 ; CALLING SEQUENCE:
                     0000    102 ;
                     0000    103 ;       CALL result.wf.v = BAS$POWRJ (base.rf.v, exponent.rl.v)
                     0000    104 ;
                     0000    105 ; INPUT PARAMETERS:
                     0000    106 ;
           00000004  0000    107          base = 4
           00000008  0000    108          exponent = 8
                     0000    109 ;
                     0000    110 ; IMPLICIT INPUTS:
                     0000    111 ;
                     0000    112 ;       NONE
                     0000    113 ;
                     0000    114 ; OUTPUT PARAMETERS:
                     0000    115 ;
                     0000    116 ;       NONE
                     0000    117 ;
                     0000    118 ; IMPLICIT OUTPUTS:
                     0000    119 ;
                     0000    120 ;       NONE
                     0000    121 ;
                     0000    122 ; FUNCTION VALUE:
                     0000    123 ; COMPLETION CODES:
                     0000    124 ;
                     0000    125 ;       floating result of exponentiation
                     0000    126 ;
                     0000    127 ; SIDE EFFECTS:
                     0000    128 ;
                     0000    129 ;       Will signal Divide By Zero if its arguments are bad,
                     0000    130 ;       and  OTS$POWRJ may also signal.
                     0000    131 ;
                     0000    132 ;--
                     0000    133
               0000' 0000    134 BAS$POWRJ::     .MASK OTS$POWRJ             ; Entry point
                     0002    135                                             ; Since this routine uses no
                     0002    136                                             ; registers and usually transfers
                     0002    137                                             ; control to OTS$POWRJ, we copy
                     0002    138                                             ; its register save mask and then
                     0002    139                                             ; JMP past its save mask and only
                     0002    140                                             ; save the registers once
            04 AC  53 0002    141          TSTF    base(AP)                   ; Test base relationship to zero
                06  15 0005    142          BLEQ    1$                         ; If base leq 0, do case analysis
00000002'GF     17  0007    143          JMP     G^OTS$POWRJ+2              ; Transfer control to the OTS$
```

```
                                000D   144                                     ; routine to do exponentiation
                                000D   145 ;+
                                000D   146 ; Come here if the base is less than or equal to zero.  We must filter
                                000D   147 ; several special cases, as described above.
                                000D   148 ;-
                          16    13   000D   149 1$:     BEQL    4$                      ; Branch if base = 0
                       08 AC    DD   000F   150         PUSHL   exponent(AP)            ; Stack EXP as parameter to OTS$POWRJ
                    7E 04 AC    52   0012   151         MNEGF   base(AP), -(SP)         ; Stack -base also
          00000000'GF    03    FB   0016   152         CALLS   #3, G^OTS$POWRJ         ; Call integer power routines
                 03 08 AC    E9   001D   153         BLBC    exponent(AP),2$         ; Branch if exponent even
                    50   50    52   0021   154         MNEGF   R0, R0                  ; Exponent odd, negate the result
                          04    04   0024   155 2$:     RET                             ; and return with it.
                                0025   156 ;+
                                0025   157 ; Come here if the base is equal to zero.  The value we return depends
                                0025   158 ; upon the sign of the exponent.
                                0025   159 ;-
                       08 AC    D5   0025   160 4$:     TSTL    exponent(AP)            ; Test the exponent against zero
                          09    19   0028   161         BLSS    6$                      ; Branch if exponent lss 0
                          03    13   002A   162         BEQL    5$                      ; Branch if exponent is 0
                                002C   163 ;+
                                002C   164 ; Come here if the base is zero and the exponent is greater than zero.
                                002C   165 ; BASIC defines this as 0.0.
                                002C   166 ;-
                          50    D4   002C   167         CLRF    R0                      ; R0 = 0.0
                                04   002E   168         RET                             ; Return to caller
                                002F   169 ;+
                                002F   170 ; Come here if the base is zero and the exponent is zero.  BASIC defines
                                002F   171 ; this as 1.0.
                                002F   172 ;-
                    50    08    50   002F   173 5$:     MOVF    #1, R0                  ; R0 = 1.0
                                04   0032   174         RET                             ; Return to caller.
                                0033   175 ;+
                                0033   176 ; Come here if the base is zero and the exponent is less than zero.
                                0033   177 ; BASIC defines this as an error.
                                0033   178 ;-
                 7E    00'8F    9A   0033   179 6$:     MOVZBL  #BAS$K_DIVBY_ZER, -(SP) ; Divide by zero
       00000000'GF    01    FB   0037   180         CALLS   #1, G^BAS$$STOP         ; Report error, never return.
                                003E   181 ;
                                003E   182         .END
```

```
BAS$$STOP           ********   X   00
BAS$K_DIVBY_ZER     ********   X   00
BAS$POWRJ           00000000 RG    01
BASE              = 00000004
EXPONENT          = 00000008
OTS$POWRJ           ********   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	0000003E (62.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.06	00:00:00.44
Command processing	110	00:00:00.48	00:00:04.06
Pass 1	72	00:00:00.43	00:00:00.85
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	48	00:00:00.34	00:00:00.85
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	265	00:00:01.34	00:00:06.25

```
The working set limit was 900 pages.
1934 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.
182 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$:BASPOWRJ/OBJ=OBJ$:BASPOWRJ MSRC$:BASPOWRJ/UPDATE=(ENH$:BASPOWRJ)
```

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