


```
BBBBBBBB      AAAAAA      SSSSSSSS      PPPPPPPP      000000      WW      WW      HH      HH      HH      HH
BBBBBBBB      AAAAAA      SSSSSSSS      PPPPPPPP      000000      WW      WW      HH      HH      HH      HH
BB      BB      AA      AA      SS      PP      PP      00      00      WW      WW      HH      HH      HH      HH
BB      BB      AA      AA      SS      PP      PP      00      00      WW      WW      HH      HH      HH      HH
BB      BB      AA      AA      SS      PP      PP      00      00      WW      WW      HH      HH      HH      HH
BB      BB      AA      AA      SS      PP      PP      00      00      WW      WW      HH      HH      HH      HH
BBBBBBBB      AA      AA      SSSSSS      PPPPPPPP      00      00      WW      WW      HHHHHHHHHH      HHHHHHHHHH
BBBBBBBB      AA      AA      SSSSSS      PPPPPPPP      00      00      WW      WW      HHHHHHHHHH      HHHHHHHHHH
BB      BB      AAAAAAAAAA      SS      PP      00      00      WW      WW      WW      HH      HH      HH      HH
BB      BB      AAAAAAAAAA      SS      PP      00      00      WW      WW      WW      HH      HH      HH      HH
BB      BB      AA      AA      SS      PP      00      00      WWW      WWW      HH      HH      HH      HH
BB      BB      AA      AA      SS      PP      00      00      WWW      WWW      HH      HH      HH      HH
BBBBBBBB      AA      AA      SSSSSSSS      PP      000000      WW      WW      HH      HH      HH      HH
BBBBBBBB      AA      AA      SSSSSSSS      PP      000000      WW      WW      HH      HH      HH      HH
```

```
....
....
....
....
```

```
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
```

(2) 47
(3) 84

DECLARATIONS
BASSPOWHH - BASIC hfloat ** hfloat

```
0000 1 .TITLE BASSPOWHH ; BASIC hfloat ** hfloat routine
0000 2 .IDENT /1-002/ ; File: BASPOWHH.MAR Edit:PLL1002
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 hfloat ** hfloat.
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
0000 45 : 1-002 - fix some typos (wrong OTSS name). PLL 7-Oct-81
```

```
0000 47      .SBTTL  DECLARATIONS
0000 48      :
0000 49      : INCLUDE FILES:
0000 50      :
0000 51      :
0000 52      :
0000 53      : EXTERNAL DECLARATIONS:
0000 54      :
0000 55      .DSABL  GBL                ; Prevent undeclared
0000 56      :                               ; symbols from being
0000 57      :                               ; automatically global.
0000 58      :
0000 59      .EXTRN  OTSSPOWHH_R3      ; OTSS hfloat ** hfloat exponentiation
0000 60      .EXTRN  OTSSPOWHJ_R3     ; OTSS hfloat ** int exponentiation
0000 61      .EXTRN  BASSK_DIVBY_ZER   ; Divide by Zero
0000 62      .EXTRN  BASSK_ILLARGLOG  ; Illegal argument in LOG
0000 63      .EXTRN  BASS$STOP        ; Error reporting routine
0000 64      :
0000 65      :
0000 66      : MACROS:
0000 67      :
0000 68      :
0000 69      :
0000 70      : EQUATED SYMBOLS:
0000 71      :
0000 72      :
0000 73      :
0000 74      : OWN STORAGE:
0000 75      :
0000 76      :
0000 77      :
0000 78      : PSECT DECLARATIONS:
0000 79      :
00000000 80      .PSECT _BASS$CODE PIC, USR, CON, REL, LCL, SHR, -
0000 81      EXE, RD, NOWRT, LONG
0000 82
```

```

0000 84      .SBTTL BASSPOWHH - BASIC hfloat ** hfloat
0000 85      :++
0000 86      : FUNCTIONAL DESCRIPTION:
0000 87      :
0000 88      : This routine takes BASE ** EXP, using the following table
0000 89      : for unusual cases:
0000 90      :
0000 91      : BASE > 0                Call OTSSPOWHH, normal case.
0000 92      : BASE = 0, EXP > 0    Return 0.0.
0000 93      : BASE = 0, EXP = 0    Return 1.0.
0000 94      : BASE = 0, EXP < 0  Error: divide by zero
0000 95      : BASE < 0, EXP even integer Call OTSSPOWHJ with -BASE
0000 96      : BASE < 0, EXP odd integer  Call OTSSPOWHJ with -BASE, negate result
0000 97      : BASE < 0, EXP not integer Error: illegal argument in LOG.
0000 98      :
0000 99      : CALLING SEQUENCE:
0000 100     :
0000 101     : CALL result.wh.v = BASSPOWHH (base.rh.v, exponent.rh.v)
0000 102     :
0000 103     : INPUT PARAMETERS:
0000 104     :
00000004 0000 105     : base = 4
00000014 0000 106     : exponent = 20
0000 107     :
0000 108     : IMPLICIT INPUTS:
0000 109     :
0000 110     : NONE
0000 111     :
0000 112     : OUTPUT PARAMETERS:
0000 113     :
0000 114     : NONE
0000 115     :
0000 116     : IMPLICIT OUTPUTS:
0000 117     :
0000 118     : NONE
0000 119     :
0000 120     : FUNCTION VALUE:
0000 121     : COMPLETION CODES:
0000 122     :
0000 123     : hfloat result of exponentiation
0000 124     :
0000 125     : SIDE EFFECTS:
0000 126     :
0000 127     : Will signal Divide By Zero or Illegal argument in LOG if its
0000 128     : arguments are bad, and OTSSPOWHH and OTSSPOWHJ may also signal.
0000 129     :
0000 130     :--
0000 131     :
0000 132     BASSPOWHH:: .MASK OTSSPOWHH_R3 ; Entry point
0002 133     ; Since this routine uses no
0002 134     ; registers and usually transfers
0002 135     ; control to OTSSPOWHH, we copy
0002 136     ; its register save mask and then
0002 137     ; JMP past its save mask and only
0002 138     ; save the registers once
04 AC 73FD 0002 139     TSTH base(AP)
06 15 0006 140     BLEQ 1$ ; Test base relationship to zero
; If base leq 0, do case analysis

```

```

00000002'GF 17 0008 141 JMP G^OTSSPOWHH_R3+2 ; Transfer control to the OTSS
; routine to do exponentiation
000E 142
000E 143 ;+
000E 144 ; Come here if the base is less than or equal to zero. We must filter
000E 145 ; several special cases, as described above.
000E 146 ;-
50 50 08 00 14 32 13 000E 147 1$: BEQL 4$ ; Branch if base = 0
AC 74FD 0010 148 EMODH exponent(AP), #0, #1, R0, R0
1D 12 0018 149 BNEQ 3$ ; Branch if exponent is not integer
001A 150 ;+
001A 151 ; The base is less than zero and the exponent is an integer.
001A 152 ; BASIC defines this as working the same way as if an integer was
001A 153 ; in the expression (making a hfloat variable which happens to
001A 154 ; contain an integer value equivalent to an integer variable).
001A 155 ;-
50 14 AC 6AFD 001A 156 CVTHL exponent(AP), R0 ; Convert exponent to integer
50 DD 001F 157 PUSHL R0 ; Save for even/odd test
50 DD 0021 158 PUSHL R0 ; Stack as parameter to OTSSPOWHJ
7E 04 AC 72FD 0023 159 MNEGH base(AP), -(SP) ; Stack -base also
00000000'GF 03 FB 0028 160 CALLS #3, G^OTSSPOWHJ_R3 ; Call integer power routines
04 8E E9 002F 161 BLBC (SP)+, 2$ ; Branch if exponent even
50 50 72FD 0032 162 MNEGH R0, R0 ; Exponent odd, negate the result
04 0036 163 2$: RET ; and return with it.
0037 164 ;+
0037 165 ; Come here if the base is less than zero but the exponent is not
0037 166 ; an integer. BASIC defines this as an error.
0037 167 ;-
7E 00'8F 9A 0037 168 3$: MOVZBL #BAS$K ILLARGLOG, -(SP) ; Illegal Argument in LOG
00000000'GF 01 FB 003B 169 CALLS #1, G^BAS$$STOP ; Never return.
0042 170 ;+
0042 171 ; Come here if the base is equal to zero. The value we return depends
0042 172 ; upon the sign of the exponent.
0042 173 ;-
14 AC 73FD 0042 174 4$: TSTH exponent(AP) ; Test the exponent against zero
OB 19 0046 175 BLSS 6$ ; Branch if exponent lss 0
04 13 0048 176 BEQL 5$ ; Branch if exponent is 0
004A 177 ;+
004A 178 ; Come here if the base is zero and the exponent is greater than zero.
004A 179 ; BASIC defines this as 0.0.
004A 180 ;-
50 7CFD 004A 181 CLRH R0 ; R0, R1 = 0.0
04 004D 182 RET ; Return to caller
004E 183 ;+
004E 184 ; Come here if the base is zero and the exponent is zero. BASIC defines
004E 185 ; this as 1.0.
004E 186 ;-
50 08 70FD 004E 187 5$: MOVH #1, R0 ; R0, R1 = 1.0
04 0052 188 RET ; Return to caller.
0053 189 ;+
0053 190 ; Come here if the base is zero and the exponent is less than zero.
0053 191 ; BASIC defines this as an error.
0053 192 ;-
7E 00'8F 9A 0053 193 6$: MOVZBL #BAS$K DIVBY ZER, -(SP) ; Divide by zero
00000000'GF 01 FB 0057 194 CALLS #1, G^BAS$$STOP ; Report error, never return.
005E 195 ;
005E 196 .END

```

BAS\$POWHH ; BASIC hfloat ** hfloat routine L 12
 Symbol table

15-SEP-1984 23:59:46 VAX/VMS Macro V04-00 Page 5
 6-SEP-1984 10:34:15 [BASRTL.SRC]BASPOWHH.MAR;1 (3)

```

BAS$$STOP ***** X 00
BAS$K_DIVBY_ZER ***** X 00
BAS$K_ILLARGLOG ***** X 00
BAS$POWHH 00000000 RG 01
BASE = 00000004
EXPONENT = 00000014
OT$$POWHH_R3 ***** X 00
OT$$POWHJ_R3 ***** 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	0000005E (94.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

 ! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.08	00:00:00.78
Command processing	104	00:00:00.44	00:00:02.23
Pass 1	73	00:00:00.52	00:00:01.31
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	48	00:00:00.37	00:00:00.96
Symbol table output	2	00:00:00.01	00:00:00.02
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	260	00:00:01.45	00:00:05.33

The working set limit was 900 pages.
 2225 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.
 196 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\$:BASPOWHH/OBJ=OBJ\$:BASPOWHH MSRC\$:BASPOWHH/UPDATE=(ENH\$:BASPOWHH)

