

BBBBBBBBBBBB		AAAAAAAAA		SSSSSSSSSS		RRRRRRRRRR		TTTTTTTTTTTT		LLL
BBBBBBBBBBBB		AAAAAAAAA		SSSSSSSSSS		RRRRRRRRRR		TTTTTTTTTTTT		LLL
BBBBBBBBBBBB		AAAAAAAAA		SSSSSSSSSS		RRRRRRRRRR		TTTTTTTTTTTT		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	SSS	SSSSSSSS	RRR	RRR	TTT		LLL
BBBBBBBBBBBB		AAA	AAA	SSS	SSSSSSSS	RRR	RRR	TTT		LLL
BBBBBBBBBBBB		AAA	AAA	SSS	SSSSSSSS	RRR	RRR	TTT		LLL
BBB	BBB	AAAAAAAAAAAA			SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAAAAAAAAAAA			SSS	RRR	RRR	TTT		LLL
BBB	BBB	AAAAAAAAAAAA			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	SSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLLLLLL
BBBBBBBBBBBB		AAA	AAA	SSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLLLLLL
BBBBBBB		AAA	AAA	SSSSSSSSSS		RRR	RRR	TTT		LLLLLLLLLLLLLLLL

```

BBBBBBBB      AAAAAA      SSSSSSSS      PPPPPPPP      000000      WW      WW      HH      HH      JJ
BBBBBBBB      AAAAAA      SSSSSSSS      PPPPPPPP      000000      WW      WW      HH      HH      JJ
JB      BB      AA      AA      SS      PP      PP      00      00      WW      WW      HH      HH      JJ
BB      BB      AA      AA      SS      PP      PP      00      00      WW      WW      HH      HH      JJ
BB      BB      AA      AA      SS      PP      PP      00      00      WW      WW      HH      HH      JJ
BB      BB      AA      AA      SS      PP      PP      00      00      WW      WW      HH      HH      JJ
BBBBBBBB      AA      AA      SSSSSS      PPPPPPPP      00      00      WW      WW      HHHHHHHHHH      JJ
BBBBBBBB      AA      AA      SSSSSS      PPPPPPPP      00      00      WW      WW      HHHHHHHHHH      JJ
BB      BB      AAAAAAAAAA      SS      PP      PP      00      00      WW      WW      HH      HH      JJ
BB      BB      AAAAAAAAAA      SS      PP      PP      00      00      WW      WW      HH      HH      JJ
BB      BB      AA      AA      SS      PP      PP      00      00      WWW      WWW      HH      HH      JJ
BB      BB      AA      AA      SS      PP      PP      00      00      WWW      WWW      HH      HH      JJ
BBBBBBBB      AA      AA      SSSSSSSS      PP      000000      WW      WW      HH      HH      JJJJJJ
BBBBBBBB      AA      AA      SSSSSSSS      PP      000000      WW      WW      HH      HH      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
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS

```

BASSPOWHJ
Table of contents

; BASIC hfloat ** longword routine ^{N 12}

15-SEP-1984 23:59:57 VAX/VMS Macro V04-00

Page 0

(2) 47
(3) 82

DECLARATIONS
BASSPOWHJ - BASIC hfloat ** long

```
0000 1 .TITLE BAS$POWHJ ; BASIC hfloat ** longword routine
0000 2 .IDENT /1-002/ ; File: BASPOWHJ.MAR Edit:PLL1002
0000 3
0000 4
0000 5 :*****
0000 6 :*
0000 7 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 :* ALL RIGHTS RESERVED.
0000 10 :*
0000 11 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 :* TRANSFERRED.
0000 17 :*
0000 18 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 :* CORPORATION.
0000 21 :*
0000 22 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 :*
0000 25 :*
0000 26 :*****
0000 27 :
0000 28 :
0000 29 :++
0000 30 : FACILITY: Basic Support Library
0000 31 :
0000 32 : ABSTRACT:
0000 33 :
0000 34 : This module contains entry points to support exponentiation
0000 35 : (** or ^) in BASIC-PLUS-2 for hfloat ** LONGWORD.
0000 36 :
0000 37 : ENVIRONMENT: User Mode, AST Reentrant
0000 38 :
0000 39 :--
0000 40 : AUTHOR: P. Levesque , CREATION DATE: 5-Oct-81
0000 41 :
0000 42 : MODIFIED BY:
0000 43 :+
0000 44 : 1-001 - Original
0000 45 : 1-002 - Fix some typos (wrong OTS$ name). PLL 7-Oct-81
```

```
0000 47      .SBTTI  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  OTS$POWHJ R3      ; OTS$ hfloat ** int exponentation
0000 60      .EXTRN  BAS$K_DIVBY_ZER  ; Divide by Zero
0000 61      .EXTRN  BAS$$STOP        ; Error reporting routine
0000 62      :
0000 63      :
0000 64      : MACROS:
0000 65      :
0000 66      :
0000 67      :
0000 68      : EQUATED SYMBOLS:
0000 69      :
0000 70      :
0000 71      :
0000 72      : OWN STORAGE:
0000 73      :
0000 74      :
0000 75      :
0000 76      : PSECT DECLARATIONS:
0000 77      :
0000 78      .PSECT _BAS$CODE PIC, USR, CON, REL, LCL, SHR, -
0000 79      EXE, RD, NOWRT, LONG
0000 80
```

```

0000 82      .SBTTL BASSPOWHJ - BASIC hfloat ** long
0000 83      :++
0000 84      : FUNCTIONAL DESCRIPTION:
0000 85      :
0000 86      : This routine takes BASE ** EXP, using the following table
0000 87      : for unusual cases:
0000 88      :
0000 89      : BASE > 0                Call OTSSPOWHJ, normal case.
0000 90      : BASE = 0, EXP > 0    Return 0.0.
0000 91      : BASE = 0, EXP = 0    Return 1.0.
0000 92      : BASE = 0, EXP < 0   Error: divide by zero
0000 93      : BASE < 0, EXP even  Call OTSSPOWHJ with -BASE
0000 94      : BASE < 0, EXP odd   Call OTSSPOWHJ with -BASE, negate result
0000 95      :
0000 96      : CALLING SEQUENCE:
0000 97      :
0000 98      : CALL result.wh.v = BASSPOWHJ (base.rh.v, exponent.rl.v)
0000 99      :
0000 100     : INPUT PARAMETERS:
0000 101     :
0000 102     : base = 4
0000 103     : exponent = 20
0000 104     :
0000 105     : IMPLICIT INPUTS:
0000 106     :
0000 107     : NONE
0000 108     :
0000 109     : OUTPUT PARAMETERS:
0000 110     :
0000 111     : NONE
0000 112     :
0000 113     : IMPLICIT OUTPUTS:
0000 114     :
0000 115     : NONE
0000 116     :
0000 117     : FUNCTION VALUE:
0000 118     : COMPLETION CODES:
0000 119     :
0000 120     : hfloat result of exponentiation
0000 121     :
0000 122     : SIDE EFFECTS:
0000 123     :
0000 124     : Will signal Divide By Zero if its arguments are bad,
0000 125     : and OTSSPOWHJ may also signal.
0000 126     :
0000 127     :--
0000 128     :
0000 129     BASSPOWHJ::      .MASK OTSSPOWHJ_R3      ; Entry point
0002 130     ; Since this routine uses no
0002 131     ; registers and usually transfers
0002 132     ; control to OTSSPOWHJ, we copy
0002 133     ; its register save mask and then
0002 134     ; JMP past its save mask and only
0002 135     ; save the registers once
0002 136     TSTH      base(AP)      ; Test base relationship to zero
0006 137     BLEQ      1$           ; If base leq 0, do case analysis
0008 138     JMP      G^OTSSPOWHJ_R3+2 ; Transfer control to the OTSS
00000004 0000 102
00000014 0000 103
00000004 0002 136
00000006 0006 137
00000002'GF 0008 138

```

```

000E 139 ; routine to do exponentiation
000E 140 ;+
000E 141 ; Come here if the base is less than or equal to zero. We must filter
000E 142 ; several special cases, as described above.
000E 143 ;-
000E 144 1$: BEQL 4$ ; Branch if base = 0
14 AC DD 0010 145 PUSHL exponent(AP) ; Stack EXP as parameter to OTS$POWHJ
7E 14 AC 72FD 0013 146 MNEGH base(AP), -(SP) ; Stack -base also
00000000'GF 03 FB 0018 147 CALLS #3, G^OTS$POWHJ_R3 ; Call integer power routines
04 14 AC E9 001F 148 BLBC exponent(AP), 2$ ; Branch if exponent even
50 50 72FD 0023 149 MNEGH R0, R0 ; Exponent odd, negate the result
04 0027 150 2$: RET ; and return with it.
0028 151 ;+
0028 152 ; Come here if the base is equal to zero. The value we return depends
0028 153 ; upon the sign of the exponent.
0028 154 ;-
14 AC D5 0028 155 4$: TSTL exponent(AP) ; Test the exponent against zero
0B 19 002B 156 BLSS 6$ ; Branch if exponent lss 0
04 13 002D 157 BEQL 5$ ; Branch if exponent is 0
002F 158 ;+
002F 159 ; Come here if the base is zero and the exponent is greater than zero.
002F 160 ; BASIC defines this as 0.0.
002F 161 ;-
50 7CFD 002F 162 CLRH R0 ; R0, R1 = 0.0
04 0032 163 RET ; Return to caller
0033 164 ;+
0033 165 ; Come here if the base is zero and the exponent is zero. BASIC defines
0033 166 ; this as 1.0.
0033 167 ;-
50 08 70FD 0033 168 5$: MOVH #1, R0 ; R0, R1 = 1.0
04 0037 169 RET ; Return to caller.
0038 170 ;+
0038 171 ; Come here if the base is zero and the exponent is less than zero.
0038 172 ; BASIC defines this as an error.
0038 173 ;-
7E 00'8F 9A 0038 174 6$: MOVZBL #BAS$K_DIVBY_ZER, -(SP) ; Divide by zero
00000000'GF 01 FB 003C 175 CALLS #1, G^BAS$$STOP ; Report error, never return.
0043 176 ;
0043 177 .END

```

BAS\$POWHJ ; BASIC hfloat ** longword routine F 13
 Symbol table

15-SEP-1984 23:59:57 VAX/VMS Macro V04-00 Page 5
 6-SEP-1984 10:34:20 [BASRTL.SRC]BASPOWHJ.MAR;1 (3)

```

BAS$$STOP      ***** X 00
BAS$K_DIVBY_ZER ***** X 00
BAS$POWHJ      00000000 RG 01
BASE           = 00000004
EXPONENT       = 00000014
OT$$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	00000043 (67.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

 ! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.08	00:00:00.37
Command processing	131	00:00:00.51	00:00:01.90
Pass 1	70	00:00:00.47	00:00:01.01
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	46	00:00:00.31	00:00:00.66
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	284	00:00:01.41	00:00:04.04

The working set limit was 750 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\$:BASPOWHJ/OBJ=OBJ\$:BASPOWHJ MSRC\$:BASPOWHJ/UPDATE=(ENH\$:BASPOWHJ)

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

BASOPEN
LIS

BASPOWJ
LIS

BASPOS
LIS

BASPOWJ
LIS

BASOPENDE
LIS

BASPOWGG
LIS

BASPOWH
LIS

BASPOWRJ
LIS

BASPOWII
LIS

BASPURJOB
LIS

BASPOWDD
LIS

BASOPENZE
LIS

BASPOWR
LIS

BASPOWJ
LIS

BASPOWR
LIS

BASPOWH
LIS

BASPOWRR
LIS