


```

BBBBBBBB      AAAAAA      SSSSSSSS      MM      MM      UU      UU      LL      DDDDDDDD      DDDDDDDD      IIIIII
BBBBBBBB      AAAAAA      SSSSSSSS      MM      MM      UU      UU      LL      DDDDDDDD      DDDDDDDD      IIIIII
BB      BB      AA      AA      SS      MMMM      MMMM      UU      UU      LL      DD      DD      DD      DD      II
BB      BB      AA      AA      SS      MMMM      MMMM      UU      UU      LL      DD      DD      DD      DD      II
BB      BB      AA      AA      SS      MM      MM      UU      UU      LL      DD      DD      DD      DD      II
BB      BB      AA      AA      SS      MM      MM      UU      UU      LL      DD      DD      DD      DD      II
BBBBBBBB      AA      AA      SSSSSS      MM      MM      UU      UU      LL      DD      DD      DD      DD      II
BBBBBBBB      AA      AA      SSSSSS      MM      MM      UU      UU      LL      DD      DD      DD      DD      II
BB      BB      AAAAAAAAAA      SS      MM      MM      UU      UU      LL      DD      DD      DD      DD      II
BB      BB      AAAAAAAAAA      SS      MM      MM      UU      UU      LL      DD      DD      DD      DD      II
BB      BB      AA      AA      SS      MM      MM      UU      UU      LL      DD      DD      DD      DD      II
BB      BB      AA      AA      SS      MM      MM      UU      UU      LL      DD      DD      DD      DD      II
BBBBBBBB      AA      AA      SSSSSSSS      MM      MM      UUUUUUUUU      LLLLLLLLLL      DDDDDDDD      DDDDDDDD      IIIIII
BBBBBBBB      AA      AA      SSSSSSSS      MM      MM      UUUUUUUUU      LLLLLLLLLL      DDDDDDDD      DDDDDDDD      IIIIII

```

```

....
....
....
....

```

```

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)	51	HISTORY	: Detailed Current Edit History
(3)	58	DECLARATIONS	
(4)	89	BAS\$\$MULD	Multiply D_floating numbers
(5)	145	BAS\$\$DIVD	Divide D_floating numbers

```
0000 1 .TITLE BAS$$MULD_DIVD BASIC Multiply and Divide D_floating
0000 2 .IDENT /1-001/ ; File: BASMULDDI.MAR
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 : FACILITY: BASIC-PLUS-2 Miscellaneous
0000 30 :++
0000 31 : ABSTRACT:
0000 32 :
0000 33 : This module contains routines to multiply and divide D_floating
0000 34 : values. They are used by BASIC Scale and Descale routines.
0000 35 :
0000 36 :
0000 37 :--
0000 38 :
0000 39 : VERSION: 1
0000 40 :
0000 41 : HISTORY:
0000 42 :
0000 43 : AUTHOR:
0000 44 : John Sauter, 08-MAY-1979
0000 45 :
0000 46 : MODIFIED BY:
0000 47 :
0000 48 :
0000 49 :
```

BAS\$MULD_DIVD
1-001

E 12

BASIC Multiply and Divide D_floating 15-SEP-1984 23:57:49 VAX/VMS Macro V04-00 Page 2
HISTORY ; Detailed Current Edit History 6-SEP-1984 10:32:18 [BASRTL.SRC]BASMULDDI.MAR;1 (2)

0000 51 .SBTTL HISTORY ; Detailed Current Edit History
0000 52
0000 53
0000 54 ; Edit History for Version 1 of BASMULDDI
0000 55 ;
0000 56 ; 1-001 - Original. JBS 08-MAY-1979

```
0000 58      .SBTTL  DECLARATIONS
0000 59
0000 60 :
0000 61 : INCLUDE FILES:
0000 62 :
0000 63 :
0000 64 :
0000 65 : EXTERNAL SYMBOLS:
0000 66 :
0000 67 :      .DSABL  GBL                      ; No automatic globals
0000 68 :
0000 69 :
0000 70 :
0000 71 : MACROS:
0000 72 :      NONE
0000 73 :
0000 74 :
0000 75 :
0000 76 : PSECT DECLARATIONS:
0000 77 :      .PSECT  _BAS$CODE          PIC, SHR, LONG, EXE, NOWRT
0000 78 :
0000 79 :
0000 80 : EQUATED SYMBOLS:
0000 81 :      NONE
0000 82 :
0000 83 :
0000 84 :
0000 85 : OWN STORAGE:
0000 86 :
0000 87 :      NONE
```

```

0000 89      .SBTTL  BAS$$MULD      Multiply D_floating numbers
0000 90
0000 91      :++
0000 92      : FUNCTIONAL DESCRIPTION:
0000 93      :
0000 94      : Multiply two D_floating numbers, producing a D_floating product.
0000 95      : Underflow is made zero.
0000 96      :
0000 97      : CALLING SEQUENCE:
0000 98      :
0000 99      : BAS$$MULD (VAL1.rd.r, VAL2.rd.r, PRODUCT.wd.r)
0000 100     :
0000 101     : INPUT PARAMETERS:
0000 102     :
0000 103     : VAL1.rd.r      Number to multiply
0000 104     : VAL2.rd.r      Number to multiply
0000 105     :
0000 106     : IMPLICIT INPUTS:
0000 107     :
0000 108     : NONE
0000 109     :
0000 110     : OUTPUT PARAMETERS:
0000 111     :
0000 112     : PRODUCT.wd.r    VAL1 * VAL2
0000 113     :
0000 114     : IMPLICIT OUTPUTS:
0000 115     :
0000 116     : NONE
0000 117     :
0000 118     : FUNCTION VALUE:
0000 119     :
0000 120     : NONE
0000 121     :
0000 122     : SIDE EFFECTS:
0000 123     :
0000 124     : May get floating Reserved Operand hardware trap.
0000 125     : May get an arithmetic fault doing the multiply.
0000 126     :
0000 127     : --
0000 128
0000 129
4000 0000 130     .ENTRY  BAS$$MULD, *M<IV>
0002 131     :+
0002 132     : Define symbols for the arguments
0002 133     :-
00000004 0002 134 VAL1=4
00000008 0002 135 VAL2=8
0000000C 0002 136 PRODUCT=12
0002 137
0002 138     :+
0002 139     : Do the multiply
0002 140     :-
0002 141
0C BC 08 BC 04 BC 65 0002 142     MULD3  @VAL1(AP), @VAL2(AP), @PRODUCT(AP)
04 0009 143     RET      ; Return to caller.

```

```

000A 145      .SBTTL  BAS$$DIVD      Divide D_floating numbers
000A 146
000A 147      :++
000A 148      : FUNCTIONAL DESCRIPTION:
000A 149      :
000A 150      :     Divide two D_floating numbers, producing a D_floating quotient.
000A 151      :     Underflow is made zero.
000A 152      :
000A 153      : CALLING SEQUENCE:
000A 154      :
000A 155      :     BAS$$DIVD (DIVISOR.rd.r, DIVIDEND.rd.r, QUOTIENT.wd.r)
000A 156      :
000A 157      : INPUT PARAMETERS:
000A 158      :
000A 159      :     DIVISOR.rd.r      Divisor
000A 160      :     DIVIDEND.rd.r     Dividend
000A 161      :
000A 162      : IMPLICIT INPUTS:
000A 163      :
000A 164      :     NONE
000A 165      :
000A 166      : OUTPUT PARAMETERS:
000A 167      :
000A 168      :     QUOTIENT.wd.r  DIVIDEND / DIVISOR
000A 169      :
000A 170      : IMPLICIT OUTPUTS:
000A 171      :
000A 172      :     NONE
000A 173      :
000A 174      : FUNCTION VALUE:
000A 175      :
000A 176      :     NONE
000A 177      :
000A 178      : SIDE EFFECTS:
000A 179      :
000A 180      :     May get Floating Reserved Operand hardware trap.
000A 181      :     May get an arithmetic fault doing the divide.
000A 182      :
000A 183      :--
000A 184
000A 185
4000 000A 186      .ENTRY  BAS$$DIVD, ^M<IV>
000C 187      :+
000C 188      : Define symbols for the arguments
000C 189      :-
00000004 000C 190  DIVISOR=4
00000008 000C 191  DIVIDEND=8
0000000C 000C 192  QUOTIENT=12
000C 193
000C 194      :+
000C 195      : Do the divide.
000C 196      :-
000C 197
0C BC 08 BC 04 BC 67 000C 198      DIVD3  @VAL1(AP), @VAL2(AP), @PRODUCT(AP)
04 0013 199      RET
0014 200      :
0014 201      .END

```


BAS\$\$MULD DIVD
Symbol table

BASIC Multiply and Divide D_floating^{1 12}

15-SEP-1984 23:57:49 VAX/VMS Macro V04-00 Page 6
6-SEP-1984 10:32:18 [BASRTL.SRC]BASMULDDI.MAR;1 (5)

BAS\$\$DIVD 0000000A RG 01
BAS\$\$MULD 00000000 RG 01
PRODUCT = 0000000C
VAL1 = 00000004
VAL2 = 00000008

↑-----↑
! 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	00000014 (20.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

↑-----↑
! Performance indicators !
↑-----↑

Phase	Page faults	CPU Time	Elapsed Time
Initialization	28	00:00:00.08	00:00:00.65
Command processing	103	00:00:00.54	00:00:02.78
Pass 1	66	00:00:00.40	00:00:01.57
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	46	00:00:00.38	00:00:01.87
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	249	00:00:01.43	00:00:06.90

The working set limit was 1050 pages.
1972 bytes (4 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 0 local symbols.
201 source lines were read in Pass 1, producing 13 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\$:BASMULDDI/OBJ=OBJ\$:BASMULDDI MSRC\$:BASMULDDI/UPDATE=(ENH\$:BASMULDDI)

BASMTD
LIS

BASMLD01
LIS

BASNOTIMP
LIS

BASMOVEAR
LIS

BASMSGDEF
LIS

BASMSGGEN
LIS

BASONECHR
LIS

BASMOVE
LIS

BASNUM
LIS

BASNAMEAS
LIS

BASNUM
LIS