

LL	IIIIII	BBBBBBBB	EEEEEEEEEE	MM	MM	000000	DDDDDDDD	FFFFFFFFFF
LL	IIIIII	BBBBBBBB	EEEEEEEEEE	MM	MM	000000	DDDDDDDD	FFFFFFFFFF
LL	II	BB	BB	EEEE	MMMM	00	DD	FF
LL	II	BB	BB	EEEE	MMMM	00	DD	FF
LL	II	BB	BB	EEEE	MM	00	DD	FF
LL	II	BB	BB	EEEE	MM	00	DD	FF
LL	II	BBBBBBBB	EEEEEEEE	MM	MM	00	DD	FFFFFFFF
LL	II	BBBBBBBB	EEEEEEEE	MM	MM	00	DD	FFFFFFFF
LL	II	BB	BB	EEEE	MM	00	DD	FF
LL	II	BB	BB	EEEE	MM	00	DD	FF
LL	II	BB	BB	EEEE	MM	00	DD	FF
LL	II	BB	BB	EEEE	MM	00	DD	FF
LL	II	BB	BB	EEEE	MM	00	DD	FF
LL	II	BB	BB	EEEE	MM	00	DD	FF
LLLLLLLLLL	IIIIII	BBBBBBBB	EEEEEEEEEE	MM	MM	000000	DDDDDDDD	FF
LLLLLLLLLL	IIIIII	BBBBBBBB	EEEEEEEEEE	MM	MM	000000	DDDDDDDD	FF

• • • •

```

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
LLLLLLLLLLL IIIIIIII   SSSSSSSS
LLLLLLLLLLL IIIIIIII   SSSSSSSS

```

(2) 52
(3) 94

DECLARATIONS
LIB\$MODF - Extended multiply and integerize

```
0000 1      .TITLE  LIB$EMODF - Extended multiply and integerize floating
0000 2      .IDENT  /1-005/      ; File: LIBEMODF.MAR  Edit: SBL1005
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: General Utility Library
0000 31
0000 32      ABSTRACT:
0000 33
0000 34      Extend precision of multiplier, multiply by multiplicand
0000 35      and extract integer and fractional portion of result.
0000 36
0000 37      ENVIRONMENT: User Mode, AST Reentrant
0000 38
0000 39      --
0000 40      AUTHOR: Steven B. Lionel, CREATION DATE: 04-Oct-78
0000 41
0000 42      MODIFIED BY:
0000 43
0000 44      SBL, 04-OCT-78 : VERSION 00
0000 45      1-001 - Original
0000 46      1-002 - Put version number in standard format: one digit of version
0000 47      number and three digits of edit number. JBS 16-NOV-78
0000 48      1-003 - Add " " to PSECT directive. JBS 21-DEC-78
0000 49      1-004 - Minor code improvements. SBL 05-Feb-79
0000 50      1-005 - Use local handler that lets all but documented signals through. SBL 17-Se
```

```
0000 52      .SBTTL  DECLARATIONS
0000 53      :
0000 54      : INCLUDE FILES:
0000 55      :
0000 56      :
0000 57      :
0000 58      : EXTERNAL SYMBOLS:
0000 59      :
0000 60      :
0000 61      .EXTRN  LIB$SIG_TO_RET      ; Library routine to convert a signal
0000 62      :                          ; to error return to caller
0000 63      :                          ; of LIB$EMODF.
0000 64      :                          ; R0 = signaled condition
0000 65      :
0000 66      :
0000 67      : MACROS:
0000 68      :
0000 69      :
0000 70      $CHFDEF      ; Condition handling facility symbols
0000 71      $SSDEF       ; System symbols
0000 72      :
0000 73      :
0000 74      : EQUATED SYMBOLS:
0000 75      :
0000 76      :
00000004 0000 77      mulr = 4      ; multiplier
00000008 0000 78      mulrx = 8     ; multiplier extension
0000000C 0000 79      muld = 12     ; multiplicand
00000010 0000 80      int = 16      ; integer portion returned
00000014 0000 81      fract = 20    ; fractional portion returned
0000 82      :
0000 83      :
0000 84      : OWN STORAGE:
0000 85      :
0000 86      :
0000 87      :
0000 88      : PSECT DECLARATIONS:
0000 89      :
00000000 90      .PSECT _LIB$CODE      PIC, USR, CON, REL, LCL, SHR, -
0000 91      EXE, RD, NOWRT, LONG
0000 92
```

```
0000 94 .SBTTL LIB$EMODF - Extended multiply and integerize
0000 95 :++
0000 96 : FUNCTIONAL DESCRIPTION:
0000 97 :
0000 98 : LIB$EMODF provides the functionality of the VAX hardware
0000 99 : instruction EMODF to high-level language users.
0000 100 :
0000 101 : The floating point multiplier extension operand (second operand)
0000 102 : is concatenated with the floating point multiplier (first
0000 103 : operand) to gain 8 additional low order fraction bits.
0000 104 : The multiplicand operand is multiplied by the extended
0000 105 : multiplier operand. After multiplication, the integer
0000 106 : portion is extracted and a 32 bit floating point number is
0000 107 : formed from the fractional part of the product by truncating
0000 108 : extra bits. The multiplication is such that the result is
0000 109 : equivalent to the exact product truncated to a fraction
0000 110 : field of 32 bits. Regarding the result as the sum of an
0000 111 : integer and fraction of the same sign, the integer operand
0000 112 : is replaced by the integer part of the result and the
0000 113 : fraction operand is replaced by the rounded fractional
0000 114 : part of the result.
0000 115 :
0000 116 : CALLING SEQUENCE:
0000 117 :
0000 118 : status.wlc.v = LIB$EMODF (mulr.rf.r, mulrx.rb.r, muld.rf.r,
0000 119 : int.wl.r, fract.wf.r)
0000 120 :
0000 121 : INPUT PARAMETERS:
0000 122 :
0000 123 : mulr.rf.r - floating point multiplier
0000 124 : mulrx.rb.r - byte to be appended to multiplier fraction
0000 125 : muld.rf.r - floating point multiplicand
0000 126 :
0000 127 : IMPLICIT INPUTS:
0000 128 :
0000 129 : NONE
0000 130 :
0000 131 : OUTPUT PARAMETERS:
0000 132 :
0000 133 : int.wl.r - integer portion of result
0000 134 : fract.wf.r - fractional portion of result
0000 135 :
0000 136 : IMPLICIT OUTPUTS:
0000 137 :
0000 138 : NONE
0000 139 :
0000 140 : FUNCTION VALUE:
0000 141 :
0000 142 : $$$_NORMAL - successful execution
0000 143 : $$$_INTOVF - integer overflow or floating overflow
0000 144 : $$$_FLTUND - floating underflow
0000 145 : $$$_ROPRAND - reserved operand
0000 146 :
0000 147 : All other exceptions are resgnaled.
0000 148 :
0000 149 : SIDE EFFECTS:
0000 150 :
```

```
0000 151 : Any exceptions other than those listed above are signaled.
0000 152 :
0000 153 :--
0000 154 :
4000 0000 155 .ENTRY LIB$EMODF, ^M<IV> ; Entry point
6D 15'AF 9E 0002 156 ;
0006 157 MOVAB B^HANDLER, (FP) ; Enable local handler to
0006 158 ; process exceptions
10 BC 0C BC 08 BC 04 BC 54 0006 159 ;
14 BC 000F 160 EMODF @mulr(AP), - ; perform multiplication
0011 161 @mulrx(AP), - ; trap on exception to
0011 162 @muld(AP), - ; handler which will
0011 163 @int(AP), - ; unwind a return error
0011 164 @fract(AP) ; condition in R0 to
0011 165 ; caller of LIB$EMODF.
50 01 9A 0011 166 MOVZBL #1, R0 ; success status code
04 0014 167 RET ; return
0000 0015 170
0015 171 HANDLER:
0015 172 .WORD 0
0017 173
0017 174 :+
0017 175 : If the exception is one of the documented exceptions for this routine,
0017 176 : call LIB$SIG_TO_RET to return it as a status. Otherwise, resignal.
0017 177 : Also, resignal if the depth is not zero.
0017 178 :-
0017 179
50 08 AC D0 0017 180 MOVL CHF$M_MCHARGLST(AP), R0 ; Get mechanism vector address
08 A0 D5 001B 181 TSTL CHF$M_MCH_DEPTH(R0) ; Is depth zero?
32 12 001E 182 BNEQ 90$ ; If not, resignal
51 04 AC D0 0020 183 MOVL CHF$M_SIGARGLST(AP), R1 ; Get signal vector address
50 04 A1 D0 0024 184 MOVL CHF$M_SIG_NAME(R1), R0 ; Get signalled condition
047C 8F 50 B1 0028 185 CMPW R0, #$$$_INTOVF ; Compare conditions
1B 13 002D 186 BEQL 10$ ; If it matches, don't resignal
049C 8F 50 B1 002F 187 CMPW R0, #$$$_FLTUND
14 13 0034 188 BEQL 10$
0454 8F 50 B1 0036 189 CMPW R0, #$$$_ROPRAND
0D 13 003B 190 BEQL 10$
04C4 8F 50 B1 003D 191 CMPW R0, #$$$_FLTUND_F
0E 12 0042 192 BNEQ 90$
04 A1 049C 8F 3C 0044 193 MOVZWL #$$$_FLTUND, CHF$M_SIG_NAME(R1) ; Change fault code to trap code
00000000'GF 6C FA 004A 194 10$: CALLG (AP), G^LIB$SIG_TO_RET ; Return signal as a status
50 0918 8F 3C 0051 195 RET
04 0052 196 90$: MOVZWL #$$$_RESIGNAL, R0 ; Resignal condition
0057 197 RET
0058 198
0058 199 .END
```

LIB\$EMODF
Symbol table

D 5

- Extended multiply and integerize float 16-SEP-1984 00:00:31 VAX/VMS Macro V04-00 Page 5
6-SEP-1984 11:06:12 [LIBRTL.SRC]LIBEMODF.MAR;1 (3)

CHFSL_MCHARGLST	=	00000008		
CHFSL_MCH_DEPTH	=	00000008		
CHFSL_SIGARGLST	=	00000004		
CHFSL_SIG_NAME	=	00000004		
FRACT	=	00000014		
HANDLER		00000015	R	02
INT	=	00000010		
LIB\$EMODF		00000000	RG	02
LIB\$SIG_TO_RET		*****	X	00
MULD	=	0000000C		
MULR	=	00000004		
MULRX	=	00000008		
SS\$FLTUND	=	0000049C		
SS\$FLTUND_F	=	000004C4		
SS\$INTOVF	=	0000047C		
SS\$RESIGNAL	=	00000918		
SS\$ROPRAND	=	00000454		

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes														
. ABS .	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE				
\$AB\$\$	00000000 (0.)	01 (1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE				
_LIB\$CODE	00000058 (88.)	02 (2.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG				

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.04	00:00:01.07
Command processing	131	00:00:00.29	00:00:01.93
Pass 1	189	00:00:02.76	00:00:08.66
Symbol table sort	0	00:00:00.46	00:00:00.62
Pass 2	50	00:00:00.56	00:00:03.82
Symbol table output	4	00:00:00.02	00:00:00.02
Psect synopsis output	2	00:00:00.02	00:00:00.38
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	407	00:00:04.15	00:00:16.61

The working set limit was 1200 pages.
21537 bytes (43 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 428 non-local and 2 local symbols.
199 source lines were read in Pass 1, producing 13 object records in Pass 2.
9 pages of virtual memory were used to define 8 macros.

LIB
1-0

+-----+
! Macro library statistics !
+-----+

Macro library name

Macros defined

_S255\$DUA28:[SYSLIB]STARLET.MLB;2

5

486 GETS were required to define 5 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS\$:LIBEMODF/OBJ=OBJ\$:LIBEMODF MSRC\$:LIBEMODF/UPDATE=(ENH\$:LIBEMODF)

0206 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY