D 8

```
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

FILEID**LIBEMODF

K 4

```
LL            IIIIII    BBBBBBBB    EEEEEEEEEE  MM      MM    OOOOOO    DDDDDDDD    FFFFFFFFFF
LL            IIIIII    BBBBBBBB    EEEEEEEEEE  MM      MM    OOOOOO    DDDDDDDD    FFFFFFFFFF
LL              II      BB      BB  EE          MMMM  MMMM  OO      OO  DD      DD  FF
LL              II      BB      BB  EE          MMMM  MMMM  OO      OO  DD      DD  FF
LL              II      BB      BB  EE          MM  MM  MM  OO      OO  DD      DD  FF
LL              II      BB      BB  EE          MM  MM  MM  OO      OO  DD      DD  FF
LL              II      BBBBBBBB    EEEEEEEE    MM      MM  OO      OO  DD      DD  FFFFFFFF
LL              II      BBBBBBBB    EEEEEEEE    MM      MM  OO      OO  DD      DD  FFFFFFFF
LL              II      BB      BB  EE          MM      MM  OO      OO  DD      DD  FF
LL              II      BB      BB  EE          MM      MM  OO      OO  DD      DD  FF
LL              II      BB      BB  EE          MM      MM  OO      OO  DD      DD  FF              ....
LL              II      BB      BB  EE          MM      MM  OO      OO  DD      DD  FF              ....
LLLLLLLLLL    IIIIII    BBBBBBBB    EEEEEEEEEE  MM      MM    OOOOOO    DDDDDDDD    FF              ....
LLLLLLLLLL    IIIIII    BBBBBBBB    EEEEEEEEEE  MM      MM    OOOOOO    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
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
0000            .TITLE  LIB$EMODF - Extended multiply and integerize floating
0000            .IDENT  /1-005/         ; File: LIBEMODF.MAR  Edit: SBL1005
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000
0000    ;++
0000    ; FACILITY: General Utility Library
0000    ;
0000    ; ABSTRACT:
0000    ;
0000    ;       Extend precision of multiplier, multiply by multiplicand
0000    ;       and extract integer and fractional portion of result.
0000    ;
0000    ; ENVIRONMENT: User Mode, AST Reentrant
0000    ;
0000    ;--
0000    ; AUTHOR: Steven B. Lionel, CREATION DATE: 04-Oct-78
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       SBL, 04-OCT-78 : VERSION 00
0000    ; 1-001 - Original
0000    ; 1-002 - Put version number in standard format: one digit of version
0000    ;            number and three digits of edit number.  JBS 16-NOV-78
0000    ; 1-003 - Add "_" to PSECT directive.  JBS 21-DEC-78
0000    ; 1-004 - Minor code improvements.  SBL 05-Feb-79
0000    ; 1-005 - Use local handler that lets all but documented signals through.  SBL 17-Se
```

```
LIB
Sym

CHF
CHF
CHF
CHF
FRA
HAN
INT
LIB
LIB
MUL
MUL
MUL
SS$
SS$
SS$
SS$
SS$

PSE
---

$AB
_LI

Pha
---
Ini
Com
Pas
Sym
Pas
Sym
Pse
Cro
Ass

The
215
The
196
9 p
```

```
                  0000          .SBTTL  DECLARATIONS
                  0000    ;
                  0000    ; INCLUDE FILES:
                  0000    ;
                  0000
                  0000    ;
                  0000    ; EXTERNAL SYMBOLS:
                  0000    ;
                  0000
                  0000          .EXTRN  LIB$SIG_TO_RET          ; Library routine to convert a signal
                  0000                                          ; to error return to caller
                  0000                                          ; of LIB$EMODF.
                  0000                                          ; R0 = signaled condition
                  0000
                  0000    ;
                  0000    ; MACROS:
                  0000    ;
                  0000
                  0000          $CHFDEF                         ; Condition handling facility symbols
                  0000          $SSDEF                          ; System symbols
                  0000
                  0000    ;
                  0000    ; EQUATED SYMBOLS:
                  0000    ;
                  0000
        00000004  0000          mulr = 4                        ; multiplier
        00000008  0000          mulrx = 8                       ; multiplier extension
        0000000C  0000          muld = 12                       ; multiplicand
        00000010  0000          int = 16                        ; integer portion returned
        00000014  0000          fract = 20                      ; fractional portion returned
                  0000
                  0000    ;
                  0000    ; OWN STORAGE:
                  0000    ;
                  0000
                  0000    ;
                  0000    ; PSECT DECLARATIONS:
                  0000    ;
              00000000          .PSECT _LIB$CODE        PIC, USR, CON, REL, LCL, SHR, -
                  0000                          EXE, RD, NOWRT, LONG
                  0000
```

```
0000              .SBTTL  LIB$EMODF  - Extended multiply and integerize
0000        ;++
0000        ; FUNCTIONAL DESCRIPTION:
0000        ;
0000        ;       LIB$EMODF provides the functionality of the VAX hardware
0000        ;       instruction EMODF to high-level language users.
0000        ;
0000        ;       The floating point multiplier extension operand (second operand)
0000        ;       is concatenated with the floating point multiplier (first
0000        ;       operand) to gain 8 additional low order fraction bits.
0000        ;       The multiplicand operand is multiplied by the extended
0000        ;       multiplier operand.  After multiplication, the integer
0000        ;       portion is extracted and a 32 bit floating point number is
0000        ;       formed from the fractional part of the product by truncating
0000        ;       extra bits.  The multiplication is such that the result is
0000        ;       equivalent to the exact product truncated to a fraction
0000        ;        ield of 32 bits.  Regarding the result as the sum of an
0000        ;       integer and fraction of the same sign, the integer operand
0000        ;       is replaced by the integer part of the result and the
0000        ;       fraction operand is replaced by the rounded fractional
0000        ;       part of the result.
0000        ;
0000        ; CALLING SEQUENCE:
0000        ;
0000        ;       status.wlc.v = LIB$EMODF (mulr.rf.r, mulrx.rb.r, muld.rf.r,
0000        ;                                int.wl.r, fract.wf.r)
0000        ;
0000        ; INPUT PARAMETERS:
0000        ;
0000        ;       mulr.rf.r       - floating point multiplier
0000        ;       mulrx.rb.r      - byte to be appended to multiplier fraction
0000        ;       muld.rf.r       - floating point multiplicand
0000        ;
0000        ; IMPLICIT INPUTS:
0000        ;
0000        ;       NONE
0000        ;
0000        ; OUTPUT PARAMETERS:
0000        ;
0000        ;       int.wl.r        - integer portion of result
0000        ;       fract.wf.r      - fractional portion of result
0000        ;
0000        ; IMPLICIT OUTPUTS:
0000        ;
0000        ;       NONE
0000        ;
0000        ; FUNCTION VALUE:
0000        ;
0000        ;       SS$_NORMAL      - successful execution
0000        ;       SS$_INTOVF      - integer overflow or floating overflow
0000        ;       SS$_FLTUND      - floating underflow
0000        ;       SS$_ROPRAND     - reserved operand
0000        ;
0000        ;       All other exceptions are resignaled.
0000        ;
0000        ; SIDE EFFECTS:
0000        ;
```

```
                              0000  ;       Any exceptions other than those listed above are signaled.
                              0000  ;
                              0000  ;--
                              0000
                        4000  0000          .ENTRY LIB$EMODF, ^M<IV>                ; Entry point
                              0002
            6D   15'AF    9E  0002          MOVAB   B^HANDLER, (FP)                 ; Enable local handler to
                              0006                                                  ; process exceptions
                              0006
 10 BC   0C BC   08 BC   04 BC    54  0006          EMODF   @mulr(AP), -                    ; perform multiplication
                 14 BC        000F
                              0011                  @mulrx(AP), -                   ; trap on exception to
                              0011                  @muld(AP), -                    ; handler which will
                              0011                  @int(AP), -                     ; unwind a return error
                              0011                  @fract(AP)                      ; condition in R0 to
                              0011                                                  ; caller of LIB$EMODF.
                              0011
            50      01    9A  0011          MOVZBL  #1, R0                          ; success status code
                              0014
                          04  0014          RET                                     ; return
                              0015
                              0015  HANDLER:
                        0000  0015          .WORD   0
                              0017
                              0017  ;+
                              0017  ;   If the exception is one of the documented exceptions for this routine,
                              0017  ;   call LIB$SIG_TO_RET to return it as a status.  Otherwise, resignal.
                              0017  ;   Also, resignal if the depth is not zero.
                              0017  ;-
                              0017
            50   08 AC    D0  0017          MOVL    CHF$L_MCHARGLST(AP), R0         ; Get mechanism vector address
                 08 A0    D5  001B          TSTL    CHF$L_MCH_DEPTH(R0)             ; Is depth zero?
                    32    12  001E          BNEQ    90$                             ; If not, resignal
            51   04 AC    D0  0020          MOVL    CHF$L_SIGARGLST(AP), R1         ; Get signal vector address
            50   04 A1    D0  0024          MOVL    CHF$L_SIG_NAME(R1), R0          ; Get signalled condition
       047C 8F      50    B1  0028          CMPW    R0, #SS$_INTOVF                 ; Compare conditions
                    1B    13  002D          BEQL    10$                             ; If it matches, don't resignal
       049C 8F      50    B1  002F          CMPW    R0, #SS$_FLTUND
                    14    13  0034          BEQL    10$
       0454 8F      50    B1  0036          CMPW    R0, #SS$_ROPRAND
                    0D    13  003B          BEQL    10$
       04C4 8F      50    B1  003D          CMPW    R0, #SS$_FLTUND_F
                    0E    12  0042          BNEQ    90$
    04 A1    049C 8F      3C  0044          MOVZWL  #SS$_FLTUND, CHF$L_SIG_NAME(R1) ; Change fault code to trap code
   00000000'GF      6C    FA  004A  10$:    CALLG   (AP), G^LIB$SIG_TO_RET          ; Return signal as a status
                          04  0051          RET
       50    0918 8F      3C  0052  90$:    MOVZWL  #SS$_RESIGNAL, R0               ; Resignal condition
                          04  0057          RET
                              0058
                              0058          .END
```

```
CHF$L_MCHARGLST                  = 00000008
CHF$L_MCH_DEPTH                  = 00000008
CHF$L_SIGARGLST                  = 00000004
CHF$L_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
$ABS$                             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.
```

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

Macro library name                       Macros defined
------------------                       --------------
_$255$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

LIBEMODH
LIS

LIBEMODD
LIS

LIBESTEMU
LIS

LIBEF
LIS

LIBEMULAT
LIS

LIBFFS
LIS

LIBFINCVT
LIS

LIBE2AREV
LIS

LIBFAO
LIS

LIBEMODG
LIS

LIBEXTV
LIS

LIBEDIV
LIS

LIBESTABL
LIS

LIBFFC
LIS

LIBEMODF
LIS

LIBEMUL
LIS

LIBFILSCA
LIS

LIBEXTZV
LIS

LIBEBCASC
LIS

LIBFAOL
LIS