```
LLL                 IIIIIIIIII   BBBBBBBBBBBB   RRRRRRRRRRRR   TTTTTTTTTTTTTTTT   LLL
LLL                 IIIIIIIIII   BBBBBBBBBBBB   RRRRRRRRRRRR   TTTTTTTTTTTTTTTT   LLL
LLL                 IIIIIIIIII   BBBBBBBBBBBB   RRRRRRRRRRRR   TTTTTTTTTTTTTTTT   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       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
LLLLLLLLLLLLLLLL       III       BBBBBBBBBBBB   RRR      RRR         TTT          LLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLL    IIIIIIIIII   BBBBBBBBBBBB   RRR      RRR         TTT          LLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLL    IIIIIIIIII   BBBBBBBBBBBB   RRR      RRR         TTT          LLLLLLLLLLLLLLLL
```

```
LL              IIIIII     BBBBBBBB    EEEEEEEEEE   MM      MM    UU      UU    LL
LL              IIIIII     BBBBBBBB    EEEEEEEEEE   MM      MM    UU      UU    LL
LL                II       BB      BB  EE           MMMM  MMMM    UU      UU    LL
LL                II       BB      BB  EE           MMMM  MMMM    UU      UU    LL
LL                II       BB      BB  EE           MM  MM  MM    UU      UU    LL
LL                II       BB      BB  EE           MM  MM  MM    UU      UU    LL
LL                II       BBBBBBBB    EEEEEEE      MM      MM    UU      UU    LL
LL                II       BBBBBBBB    EEEEEEE      MM      MM    UU      UU    LL
LL                II       BB      BB  EE           MM      MM    UU      UU    LL
LL                II       BB      BB  EE           MM      MM    UU      UU    LL
LL                II       BB      BB  EE           MM      MM    UU      UU    LL
LL                II       BB      BB  EE           MM      MM    UU      UU    LL        ....
LLLLLLLLLL    IIIIII     BBBBBBBB    EEEEEEEEEE   MM      MM    UUUUUUUUUU    LLLLLLLLLL   ....
LLLLLLLLLL    IIIIII     BBBBBBBB    EEEEEEEEEE   MM      MM    UUUUUUUUUU    LLLLLLLLLL   ....

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
```

```
0000     1               .TITLE  LIB$EMUL - Execute EMUL instruction
0000     2               .IDENT  /1-001/                   ; File: LIBEMUL.MAR Edit: SBL1001
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     ;     This module contains LIB$EMUL, which makes the VAX EMUL instruction
0000    35     ;     available as a callable procedure.
0000    36     ;
0000    37     ; ENVIRONMENT: Runs at any access mode, AST Reentrant
0000    38     ;
0000    39     ; AUTHOR: Steven B. Lionel, CREATION DATE: 8-July-1981
0000    40     ;
0000    41     ; MODIFIED BY:
0000    42     ;
0000    43     ; 1-001 - Original. SBL 8-July-1981
0000    44     ;--
```

LIB$EMUL
1-001

N 6

- Execute EMUL instruction
DECLARATIONS

16-SEP-1984 00:06:36   VAX/VMS Macro V04-00   Page 2
6-SEP-1984 11:06:25   [LIBRTL.SRC]LIBEMUL.MAR;1   (2)

```
            0000       46              .SBTTL  DECLARATIONS
            0000       47 ;
            0000       48 ; LIBRARY MACRO CALLS:
            0000       49 ;
            0000       50     $SSDEF                      ; Define SS$ symbols
            0000       51 ;
            0000       52 ; EXTERNAL DECLARATIONS:
            0000       53 ;
            0000       54     .DSABL  GBL                 ; Force all external symbols to be declared
            0000       55 ;  NONE
            0000       56 ;
            0000       57 ; MACROS:
            0000       58 ;
            0000       59     NONE
            0000       60 ;
            0000       61 ; EQUATED SYMBOLS:
            0000       62 ;
            0000       63     NONE
            0000       64 ;
            0000       65 ; OWN STORAGE:
            0000       66 ;
            0000       67     NONE
            0000       68 ;
            0000       69 ; PSECT DECLARATIONS:
            0000       70 ;
        00000000       71     .PSECT _LIB$CODE PIC, USR, CON, REL, LCL, SHR, -
            0000       72              EXE, RD, NOWRT, LONG
            0000       73
```

LIB$EMUL
1-001

B 7

- Execute EMUL instruction   16-SEP-1984 00:06:36 VAX/VMS Macro V04-00   Page   3
LIB$EMUL - Execute EMUL instruction   6-SEP-1984 11:06:25 [LIBRTL.SRC]LIBEMUL.MAR;1   (3)

```
                              0000      75              .SBTTL  LIB$EMUL - Execute EMUL instruction
                              0000      76     ;++
                              0000      77     ; FUNCTIONAL DESCRIPTION:
                              0000      78     ;
                              0000      79     ;       This procedure makes the VAX EMUL instruction available as
                              0000      80     ;       a callable procedure.
                              0000      81     ;
                              0000      82     ;       The multiplicand argument is multiplied by the multiplier
                              0000      83     ;       argument giving a double-length result.  The addend argument
                              0000      84     ;       is sign-extended to double-length and added to the result, and
                              0000      85     ;       then the product argument is replaced by the final result.
                              0000      86     ;
                              0000      87     ;       For more information, see the VAX-11 Architecture Handbook.
                              0000      88     ;
                              0000      89     ; CALLING SEQUENCE:
                              0000      90     ;
                              0000      91     ;       status.wlc.v = LIB$EMUL (multiplier.rl.r, multiplicand.rl.r,
                              0000      92     ;                               addend.rl.r, product.wq.r)
                              0000      93     ;
                              0000      94     ; FORMAL PARAMETERS:
                              0000      95     ;
                    00000004  0000      96     ;       multiplier = 4           ; The address of the longword integer multiplier.
                              0000      97     ;
                    00000008  0000      98     ;       multiplicand = 8         ; The address of the longword integer multiplicand.
                              0000      99     ;
                    0000000C  0000     100     ;       addend = 12              ; The address of the longword integer addend.
                              0000     101     ;
                    00000010  0000     102     ;       product = 16             ; The address of a quadword integer for the
                              0000     103     ;                                ; product result.
                              0000     104     ;
                              0000     105     ; IMPLICIT INPUTS:
                              0000     106     ;
                              0000     107     ;       NONE
                              0000     108     ;
                              0000     109     ; IMPLICIT OUTPUTS:
                              0000     110     ;
                              0000     111     ;       NONE
                              0000     112     ;
                              0000     113     ; COMPLETION STATUS:
                              0000     114     ;
                              0000     115     ;       SS$_NORMAL, normal successful completion
                              0000     116     ;
                              0000     117     ; SIDE EFFECTS:
                              0000     118     ;
                              0000     119     ;       NONE
                              0000     120     ;
                              0000     121     ;--
                              0000     122
                         4000 0000     123              .ENTRY LIB$EMUL, ^M<IV>         ; Entry point
                              0002     124
10 BC  0C BC  04 BC  08 BC  7A 0002    125              EMUL    @multiplicand(AP), @multiplier(AP), -
                              000B     126                      @addend(AP), @product(AP)
                              000B     127
                  50  01  D0 000B      128              MOVL    #SS$_NORMAL, R0        ; The EMUL can not fail
                         04  000E      129              RET
                              000F     130
                              000F     131              .END                          ; End of module LIB$EMUL
```

C 7

LIB$EMUL   - Execute EMUL instruction   16-SEP-1984 00:06:36 VAX/VMS Macro V04-00   Page   4
Symbol table   6-SEP-1984 11:06:25 [LIBRTL.SRC]LIBEMUL.MAR;1   (3)

```
ADDEND                           = 0000000C
LIB$EMUL                           00000000 RG    02
MULTIPLICAND                     = 00000008
MULTIPLIER                       = 00000004
PRODUCT                          = 00000010
SS$_NORMAL                       = 00000001
```

```
                                 +------------------+
                                 ! 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                        0000000F (     15.)  02 (   2.)   PIC   USR  CON  REL  LCL  SHR    EXE    RD    NOWRT  NOVEC  LONG
```

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

```
Phase                       Page faults   CPU Time      Elapsed Time
-----                       -----------   --------      ------------
Initialization                      29    00:00:00.05   00:00:01.48
Command processing                 102    00:00:00.28   00:00:02.20
Pass 1                             179    00:00:02.41   00:00:10.72
Symbol table sort                    0    00:00:00.42   00:00:02.64
Pass 2                              40    00:00:00.47   00:00:03.02
Symbol table output                  3    00:00:00.02   00:00:00.02
Psect synopsis output                2    00:00:00.01   00:00:00.01
Cross-reference output               0    00:00:00.00   00:00:00.00
Assembler run totals               357    00:00:03.67   00:00:20.10
```

The working set limit was 1200 pages.
19637 bytes (39 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 412 non-local and 0 local symbols.
131 source lines were read in Pass 1, producing 12 object records in Pass 2.
8 pages of virtual memory were used to define 7 macros.

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

```
Macro library name                             Macros defined
------------------                             --------------
_$255$DUA28:[SYSLIB]STARLET.MLB;2                     4
```

469 GETS were required to define 4 macros.

There were no errors, warnings or information messages.

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

LIBEMODH
LIS

LIBEMODD
LIS

LIBESTEMU
LIS

LIBEF
LIS

LIBEMULAT
LIS

LIBFFS
LIS

LIBFINCVT
LIS

LIBE2AREV
LIS

LIBFAO
LIS

LIBEDIV
LIS

LIBEMODG
LIS

LIBEXTV
LIS

LIBESTABL
LIS

LIBFFC
LIS

LIBEMODF
LIS

LIBEMUL
LIS

LIBFILSCA
LIS

LIBEXTZV
LIS

LIBEBCASC
LIS

LIBFAOL
LIS