```
MMM        MMM    000000000      UUU         UUU  NNN         NNN  TTTTTTTTTTTTTTT
MMM        MMM    000000000      UUU         UUU  NNN         NNN  TTTTTTTTTTTTTTT
MMM        MMM    000000000      UUU         UUU  NNN         NNN  TTTTTTTTTTTTTTT
MMMMMM  MMMMMM   000       000    UUU         UUU  NNN         NNN        TTT
MMMMMM  MMMMMM   000       000    UUU         UUU  NNN         NNN        TTT
MMMMMM  MMMMMM   000       000    UUU         UUU  NNN         NNN        TTT
MMM  MMM  MMM   000       000    UUU         UUU  NNNNNN       NNN        TTT
MMM  MMM  MMM   000       000    UUU         UUU  NNNNNN       NNN        TTT
MMM  MMM  MMM   000       000    UUU         UUU  NNNNNN       NNN        TTT
MMM        MMM   000       000    UUU         UUU  NNN   NNN    NNN        TTT
MMM        MMM   000       000    UUU         UUU  NNN   NNN    NNN        TTT
MMM        MMM   000       000    UUU         UUU  NNN   NNN    NNN        TTT
MMM        MMM   000       000    UUU         UUU  NNN      NNNNNN        TTT
MMM        MMM   000       000    UUU         UUU  NNN      NNNNNN        TTT
MMM        MMM   000       000    UUU         UUU  NNN      NNNNNN        TTT
MMM        MMM   000       000    UUU         UUU  NNN         NNN        TTT
MMM        MMM   000       000    UUU         UUU  NNN         NNN        TTT
MMM        MMM   000       000    UUU         UUU  NNN         NNN        TTT
MMM        MMM    000000000      UUUUUUUUUUUUUUU  NNN         NNN        TTT
MMM        MMM    000000000      UUUUUUUUUUUUUUU  NNN         NNN        TTT
MMM        MMM    000000000      UUUUUUUUUUUUUUU  NNN         NNN        TTT
```

```
LL        EEEEEEEEEE  FFFFFFFFFF  TTTTTTTTTT   000000    NN      NN  EEEEEEEEEE
LL        EEEEEEEEEE  FFFFFFFFFF  TTTTTTTTTT   000000    NN      NN  EEEEEEEEEE
LL        EE          FF             TT      00      00  NN      NN  EE
LL        EE          FF             TT      00      00  NN      NN  EE
LL        EE          FF             TT      00      00  NNNN    NN  EE
LL        EE          FF             TT      00      00  NNNN    NN  EE
LL        EEEEEEEE    FFFFFFFF       TT      00      00  NN  NN  NN  EEEEEEEE
LL        EEEEEEEE    FFFFFFFF       TT      00      00  NN  NN  NN  EEEEEEEE
LL        EE          FF             TT      00      00  NN    NNNN  EE
LL        EE          FF             TT      00      00  NN    NNNN  EE
LL        EE          FF             TT      00      00  NN      NN  EE          ....
LL        EE          FF             TT      00      00  NN      NN  EE          ....
LLLLLLLLL EEEEEEEEEE  FF             TT       000000     NN      NN  EEEEEEEEEE  ....
LLLLLLLLL EEEEEEEEEE  FF             TT       000000     NN      NN  EEEEEEEEEE  ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
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
LLLLLLLLL     IIIIII      SSSSSSSS
LLLLLLLLL     IIIIII      SSSSSSSS
```

LEFTONE
V04-000

H 9
- FIND LEFTMOST ONE BIT IN LONGWORD   16-SEP-1984 00:59:43 VAX/VMS Macro V04-00   Page 1
                                         5-SEP-1984 15:11:23 [MOUNT.SRC]FCPDEF.MAR;1   (2)

```
                0000     55 ;
                0000     56 ; ASSORTED MACROS USED IN FCP CODE
                0000     57 ;
                0000     58
                0000     59         .MACRO  SET_IPL LEVEL              ; SET PROCESSOR IPL (DUMMY NOW)
                0000     60         .ENDM   SET_IPL
                0000     61
                0000     62 ; MACRO USED TO SIGNAL FATAL ERRORS (INTERNAL CONSISTENCY CHECKS).
                0000     63 ;
                0000     64         .MACRO  BUG_CHECK       CODE, TYPE, MESSAGE
                0000     65         HALT                    ; SIMPLY CALL A HALT FOR NOW
                0000     66         .ENDM   BUG_CHECK
                0000     67
                0000     68 ; MACRO TO SIGNAL AN ERROR STATUS AND CONTINUE.
                0000     69 ;
                0000     70         .MACRO  ERROR   CODE
                0000     71         MOVL    #CODE,USER_STATUS
                0000     72         .ENDM   ERROR
                0000     73 ;
                0000     74 ; MACRO TO SIGNAL AN ERROR STATUS AND EXIT.
                0000     75 ;
                0000     76         .MACRO  ERR_EXIT        CODE
                0000     77         MOVZWL  CODE,-(SP)
                0000     78         HALT                    ; UNTIL WE FIGURE THIS OUT
                0000     79         .ENDM   ERR_EXIT
                0000     80 ;
                0000     81 ; TYPE CODES USED TO IDENTIFY BLOCKS BEING READ BY READ_BLOCK.
                0000     82 ; NOTE THAT READ_BLOCK CONTAINS A TABLE INDEXED BY THESE CODES.
                0000     83 ;
00000000        0000     84         HEADER_TYPE     = 0             ; FILE HEADER
00000001        0000     85         BITMAP_TYPE     = 1             ; STORAGE BITMAP
00000002        0000     86         DIRECTORY_TYPE  = 2             ; DIRECTORY BLOCK
00000003        0000     87         INDEX_TYPE      = 3             ; OTHER INDEX FILE BLOCKS
                0000     88 ;
                0000     89 ; TYPE CODES USED TO IDENTIFY BLOCKS OF MEMORY REQUESTED FROM THE
                0000     90 ; ALLOCATOR. NOTE THAT THESE CODES INDEX INTO A TABLE IN ALLOCATE.
                0000     91 ;
00000000        0000     92         FCB_TYPE        = 0             ; FILE CONTROL BLOCK
00000001        0000     93         WCB_TYPE        = 1             ; WINDOW BLOCK
```

LEFTONE
V04-000

I 9
- FIND LEFTMOST ONE BIT IN LONGWORD   16-SEP-1984 00:59:43   VAX/VMS Macro V04-00   Page 2
                                            5-SEP-1984 02:03:49   [MOUNT.SRC]LEFTONE.MAR;1   (1)

```
                    0000      1           .TITLE   LEFTONE - FIND LEFTMOST ONE BIT IN LONGWORD
                    0000      2           .IDENT   'V04-000'
                    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:  F11ACP STRUCTURE LEVEL 1
                    0000     31    ;
                    0000     32    ; ABSTRACT:
                    0000     33    ;
                    0000     34    ;       THIS ROUTINE RETURNS THE BIT NUMBER PLUS ONE OF THE LEFTMOST BIT
                    0000     35    ;       SET IN THE ARGUMENT LONGWORD. IF THE ARGUMENT IS ZERO, ZERO IS
                    0000     36    ;       RETURNED.
                    0000     37    ;
                    0000     38    ; ENVIRONMENT:
                    0000     39    ;
                    0000     40    ;       STARLET OPERATING SYSTEM, INCLUDING PRIVILEGED SYSTEM SERVICES
                    0000     41    ;       AND INTERNAL EXEC ROUTINES. THIS ROUTINE MUST BE CALLED IN
                    0000     42    ;       KERNEL MODE.
                    0000     43    ;
                    0000     44    ;--
                    0000     45    ;
                    0000     46    ; AUTHOR:  ANDREW C. GOLDSTEIN  2-FEB-78  14:53
                    0000     47    ;
                    0000     48    ; MODIFIED BY:
                    0000     49    ;
                    0000     50    ;       V02-000 ACG0167         Andrew C. Goldstein,    18-Apr-1980  13:40
                    0000     51    ;               Previous revision history moved to MOUNT.REV
                    0000     52    ;**
                    0000     53
                    0000     54    ;
                    0000     55    ; EQUATED SYMBOLS:
                    0000     56    ;
00000004            0000     57    ARG     = 4                                     ; INPUT ARGUMENT
```

```
                       0000        59  ;++
                       0000        60  ;
                       0000        61  ; FUNCTIONAL DESCRIPTION:
                       0000        62  ;
                       0000        63  ;       THIS ROUTINE RETURNS THE BIT NUMBER PLUS ONE OF THE LEFTMOST BIT
                       0000        64  ;       SET IN THE INPUT ARGUMENT.
                       0000        65  ;
                       0000        66  ; CALLING SEQUENCE:
                       0000        67  ;       CALL    LEFT_ONE (ARG1)
                       0000        68  ;
                       0000        69  ; INPUT PARAMETERS:
                       0000        70  ;       ARG1: RANDOM LONGWORD
                       0000        71  ;
                       0000        72  ; IMPLICIT INPUTS:
                       0000        73  ;       NONE
                       0000        74  ;
                       0000        75  ; OUTPUT PARAMETERS:
                       0000        76  ;       NONE
                       0000        77  ;
                       0000        78  ; IMPLICIT OUTPUTS:
                       0000        79  ;       NONE
                       0C00        80  ;
                       0000        81  ; ROUTINE VALUE:
                       0000        82  ;       BIT POSITION PLUS 1
                       0000        83  ;
                       0000        84  ; SIDE EFFECTS:
                       0000        85  ;       NONE
                       0000        86  ;
                       0000        87  ;--
                       0000        88
                   00000000        89          .PSECT  $CODE$,NOWRT,LONG
                       0000        90
                       0000        91  LEFT_ONE::
              0000     0000        92          .WORD   ^M<>                    ; NO REGISTERS SAVED
        50 20    D0    0002        93          MOVL    #32,R0                  ; ASSUME NEGATIVE
  09 04 AC 1F    E0    0005        94          BBS     #31,ARG(AP),10$         ; BRANCH TO END IF NEGATIVE
     50    04 AC 6E    000A        95          CVTLD   ARG(AP),R0              ; CONVERT TO DOUBLE FLOATING
50 50 07 07    EF    000E        96          EXTZV   #7,#7,R0,R0             ; GET EXPONENT, LESS EXCESS 128
           04    0013        97  10$:    RET                             ; AND THAT'S ALL
                       0014        98
                       0014        99
                       0014       100
                       0014       101          .END
```

```
ARG            = 00000004
BITMAP_TYPE    = 00000001
DIRECTORY_TYPE = 00000002
FCB_TYPE       = 00000000
HEADER_TYPE    = 00000000
INDEX_TYPE     = 00000003
LEFT_ONE         00000000 RG    01
WCB_TYPE       = 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
$CODE$	00000014 (20.)	01 (1.)	NOPIC	US?	CON	REL	LCL	NOSHR	EXE	RD	NOWRT NOVEC LONG

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	32	00:00:00.07	00:00:01.12
Command processing	127	00:00:00.78	00:00:04.88
Pass 1	78	00:00:00.54	00:00:53.16
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	39	00:00:00.48	00:00:03.00
Symbol table output	3	00:00:00.02	00:00:00.02
Psect synopsis output	1	00:00:00.03	00:00:00.13
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	283	00:00:01.92	00:01:02.40

The working set limit was 900 pages.
1849 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 1 local symbols.
195 source lines were read in Pass 1, producing 11 object records in Pass 2.
4 pages of virtual memory were used to define 4 macros.

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

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	0
_255DUA28:[SYSLIB]STARLET.MLB;2	0
TOTALS (all libraries)	0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:LEFTONE/OBJ=OBJ$:LEFTONE MSRC$:FCPDEF/UPDATE=(ENH$:FCPDEF)+MSRC$:LEFTONE/UPDATE=(ENH$:LEFTONE)+EXECML$/LIB

CHKHM2
LIS

CHNUCB
LIS

GETUIC
LIS

ERASE
LIS

LEFTONE
LIS

MOUDK2
LIS

CHKHM1
LIS

CHKSM2
LIS

CLUSTRMNT
LIS

INIFC2
LIS

MOUDK1
LIS

MAKRUT
LIS

MAKLOG
LIS