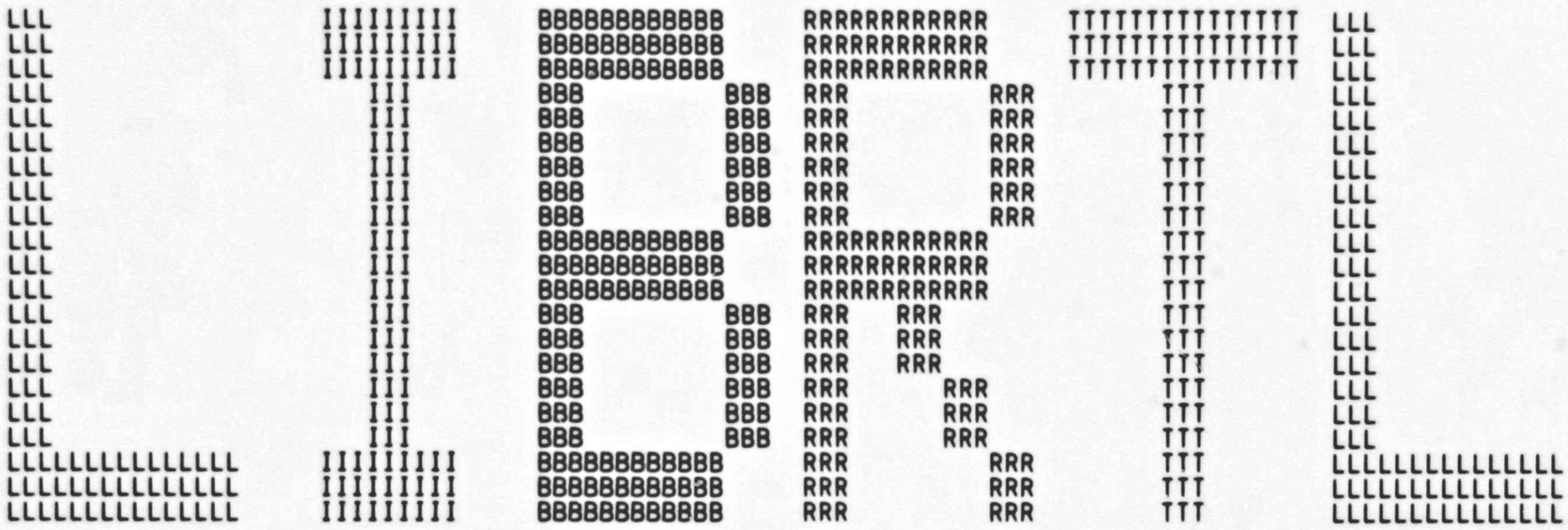
LLL IIIIIIIII BBBBBBBBBBB RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
LLL IIIIIIIII BBBBBBBBBBBB RRRRRRRRRRRR TTTTTTTTTTTTTTT LLL
LLL IIIIIIIII BBBBBBBBBBBB RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBB BBB RRR RRR TTT LLL
LLL III BBBBBBBBBBB RRRRRRRRRRR 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
LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBBB RRR RRR TTT LLLLLLLLLLLLLLL

FILEID**LIBGETOPC

```
LL          IIIIII   BBBBBBBB     GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT    OOOOOO    PPPPPPPP     CCCCCCCC
LL          IIIIII   BBBBBBBB     GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT    OOOOOO    PPPPPPPP     CCCCCCCC
LL            II     BB      BB   GG        EE              TT      OO      OO  PP      PP   CC
LL            II     BB      BB   GG        EE              TT      OO      OO  PP      PP   CC
LL            II     BB      BB   GG        EE              TT      OO      OO  PP      PP   CC
LL            II     BB      BB   GG        EE              TT      OO      OO  PP      PP   CC
LL            II     BBBBBBBB     GG        EEEEEEEE        TT      OO      OO  PPPPPPPP     CC
LL            II     BBBBBBBB     GG        EEEEEEEE        TT      OO      OO  PPPPPPPP     CC
LL            II     BB      BB   GG  GGGGGG  EE              TT      OO      OO  PP           CC
LL            II     BB      BB   GG  GGGGGG  EE              TT      OO      OO  PP           CC
LL            II     BB      BB   GG      GG  EE              TT      OO      OO  PP           CC                    ....
LL            II     BB      BB   GG      GG  EE              TT      OO      OO  PP           CC                    ....
LLLLLLLLLL  IIIIII   BBBBBBBB     GGGGGG    EEEEEEEEEE      TT        OOOOOO    PP             CCCCCCCC            ....
LLLLLLLLLL  IIIIII   BBBBBBBB     GGGGGG    EEEEEEEEEE      TT        OOOOOO    PP             CCCCCCCC            ....

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     1          .TITLE  LIB$GET_OPCODE - Get opcode from debugger
0000     2          .IDENT  /1-001/                 ; File: LIBGETOPC.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 a procedure which asks the debugger if
0000    35 ;       a particular instruction has been modified by it.
0000    36 ;
0000    37 ; ENVIRONMENT: Runs at any access mode, AST Reentrant
0000    38 ;
0000    39 ; AUTHOR: Steven B. Lionel, CREATION DATE: 04-Dec-1981
0000    40 ;
0000    41 ; MODIFIED BY:
0000    42 ;
0000    43 ; 1-001 - Original. SBL 04-DEC-1981
0000    44 ;--
```

```
                        .SBTTL  DECLARATIONS
                ;
                ; LIBRARY MACRO CALLS:
                ;
                        $SSDEF                  ; SS$_ symbols
                        $CHFDEF                 ; Condition handling facility symbols
                ;
                ; EXTERNAL DECLARATIONS:
                ;
                        .DSABL  GBL             ; Force all external symbols to be declared
                        .EXTRN  LIB$SIGNAL      ; Signal exception
                ;
                ; MACROS:
                ;
                        NONE
                ;
                ; EQUATED SYMBOLS:
                ;
                        NONE
                ;
                ; OWN STORAGE:
                ;
                        NONE
                ;
                ; PSECT DECLARATIONS:
                ;
00000000                .PSECT _LIB$CODE PIC, USR, CON, REL, LCL, SHR, -
                                EXE, RD, NOWRT, LONG
```

```
                     .SBTTL  LIB$GET_OPCODE
                ;++
                ; FUNCTIONAL DESCRIPTION:
                ;
                ;       LIB$GET__OPCODE returns as its function value the opcode of
                ;       an instruction which may have been replaced by a debugger.
                ;       For example, VAX-11 DEBUG replaces instructions for which
                ;       breakpoints have been set with BPT. It is designed to be
                ;       used from condition handlers which understand instruction
                ;       faults and which need to know the original contents of the
                ;       instruction stream.
                ;
                ;       LIB$GET__OPCODE is called implicitly from LIB$DECODE__FAULT,
                ;       LIB$EMULATE, LIB$FIXUP__FLT and LIB$SIM__TRAP.  Therefore,
                ;       it should only be used from fault handlers which do not
                ;       employ LIB$DECODE__FAULT.
                ;
                ;       LIB$GET__OPCODE determines the original opcode by signalling
                ;       the special exception "SS$__DBGOPCREQ, debugger opcode
                ;       request".  This success-severity exception is signalled with
                ;       two FAO arguments: the first is the PC of the instruction
                ;       for which the request is being made, the second is the
                ;       address of a 16-bit word where the original instruction is
                ;       to be placed.  If the debugger is being used, it has a
                ;       handler in the primary exception vector.  This handler
                ;       recognizes SS$__DBGOPCREQ as a request for the original
                ;       opcode for the indicated PC.  If the debugger has changed
                ;       the instruction at that PC, it stores the original opcode at
                ;       the location given as the second FAO argument. If the
                ;       debugger has modified only one byte of the instruction
                ;       stream, it will only write one byte to the destination. The
                ;       debugger's handler then returns SS$__CONTINUE, causing
                ;       execution to continue after the signal.
                ;
                ;       If no debugger is present, the error will be resignalled and
                ;       will be intercepted by a handler inside LIB$GET__OPCODE,
                ;       which will then return SS$__CONTINUE.  LIB$GET__OPCODE
                ;       copies the instruction to the destination location before
                ;       signalling so that the original instruction is returned if
                ;       not modified by the debugger.
                ;
                ; CALLING SEQUENCE:
                ;
                ;       opcode.wwu.v = LIB$GET__OPCODE (instruction.rzi.r)
                ;
                ; FORMAL PARAMETERS:
                ;
00000004        ;       instruction = 4         ; The PC of the instruction
                ;                               ; which is to be inquired about.
                ;
                ; IMPLICIT INPUTS:
                ;
                ;       NONE
                ;
                ; IMPLICIT OUTPUTS:
                ;
                ;       NONE
```

```
                                0000   134 ;
                                0000   135 ; ROUTINE VALUE:
                                0000   136 ;
                                0000   137 ;       The original instruction.
                                0000   138 ;
                                0000   139 ; SIDE EFFECTS:
                                0000   140 ;
                                0000   141 ;       NONE
                                0000   142 ;
                                0000   143 ;--
                                0000   144
                          0000  0000   145          .ENTRY  LIB$GET_OPCODE, ^M<>    ; Entry point
                50    04 AC     D0  0002   146          MOVL    instruction(AP), R0     ; Get instruction PC
                   7E     60     9A  0006   147          MOVZBL  (R0), -(SP)             ; Push instruction in stack
                FD 8F     6E     91  0009   148          CMPB    (SP), #^XFD             ; 2-byte opcode?
                          03     1F  000D   149          BLSSU   10$                     ; Skip if not
                   6E     60     3C  000F   150          MOVZWL  (R0), (SP)              ; Get 2-byte instruction
                          5E     DD  0012   151 10$:     PUSHL   SP                      ; Push address of saved instruction
                          50     DD  0014   152          PUSHL   R0                      ; Push instruction PC
                000006A1  8F     DD  0016   153          PUSHL   #SS$_DBGOPCREQ          ; Push "debugger opcode request" message
                  6D    2B'AF     9E  001C   154          MOVAB   B^HANDLER, (FP)         ; Enable local handler
            00000000'GF    03     FB  0020   155          CALLS   #3, G^LIB$SIGNAL        ; Signal SS$_DBGOPCREQ
                   50     8E     D0  0027   156          MOVL    (SP)+, R0               ; Get "original" opcode in R0
                          04     05  002A   157          RET                             ; Return to caller
```

```
                        002B   159 ;++
                        002B   160 ; Local handler which does a SS$_CONTINUE if SS$_DBGOPCREQ seen.
                        002B   161 ;--
                        002B   162
                        002B   163 HANDLER:
                  0000  002B   164         .WORD   0               ; Save nothing
              50  01    9A 002D   165         MOVZBL  #SS$_CONTINUE, R0       ; Assume continue
          51  04 AC    D0 0030   166         MOVL    CHF$L_SIGARGLST(AP), R1 ; Get signal arguments list
000006A1 8F   04 A1    D1 0034   167         CMPL    CHF$L_SIG_NAME(R1), #SS$_DBGOPCREQ ; Is it SS$_DBGOPCREQ?
                05    13 003C   168         BEQL    10$                     ; Skip if yes
          50  0918 8F   3C 003E   169         MOVZWL  #SS$_RESIGNAL, R0       ; Resignal all other exceptions
                04    0043   170 10$:     RET                             ; Return to condition handling
                      0044   171
                      0044   172         .END                            ; End of module LIB$GET_OPCODE
```

```
CHF$L_SIGARGLST                 = 00000004
CHF$L_SIG_NAME                  = 00000004
HANDLER                           0000002B R      02
INSTRUCTION                     = 00000004
LIB$GET_OPCODE                    00000000 RG     02
LIB$SIGNAL                        ******** X      00
SS$_CONTINUE                    = 00000001
SS$_DBGOPCREQ                   = 000006A1
SS$_RESIGNAL                    = 00000918
```

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	00000044 (68.)	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	00:00:01.79
Command processing	114	00:00:00.32	00:00:01.61
Pass 1	191	00:00:02.59	00:00:11.68
Symbol table sort	0	00:00:00.42	00:00:02.05
Pass 2	47	00:00:00.55	00:00:02.72
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	388	00:00:03.97	00:00:19.90

The working set limit was 900 pages.
21140 bytes (42 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 424 non-local and 2 local symbols.
172 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
_255DUA28:[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$:LIBGETOPC/OBJ=OBJ$:LIBGETOPC MSRC$:LIBGETOPC/UPDATE=(ENH$:LIBGETOPC)
```

LIBFLTUND
LIS

LIBGETSYI
LIS

LIBICHAR
LIS

LIBINITIA
LIS

LIBFIXUPF
LIS

LIBGETFOR
LIS

LIBGETINP
LIS

LIBINISHR
LIS

LIBGETDVI
LIS

LIBGETOPC
LIS

LIBFNDIMG
LIS

LIBGETMSG
LIS

LIBINDEX
LIS

LIBINSQHI
LIS

LIBGETJPI
LIS

LIBGETTAB
LIS