```
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSS  RRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBB         BBB   AAA         AAA   SSS          RRR         RRR         TTT        LLL
BBB         BBB   AAA         AAA   SSS          RRR         RRR         TTT        LLL
BBB         BBB   AAA         AAA   SSS          RRR         RRR         TTT        LLL
BBB         BBB   AAA         AAA   SSS          RRR         RRR         TTT        LLL
BBB         BBB   AAA         AAA   SSS          RRR         RRR         TTT        LLL
BBB         BBB   AAA         AAA   SSS          RRR         RRR         TTT        LLL
BBBBBBBBBBBB      AAA         AAA     SSSSSSSSS  RRRRRRRRRRR             TTT        LLL
BBBBBBBBBBBB      AAA         AAA     SSSSSSSSS  RRRRRRRRRRR             TTT        LLL
BBBBBBBBBBBB      AAA         AAA     SSSSSSSSS  RRRRRRRRRRR             TTT        LLL
BBB         BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR               TTT        LLL
BBB         BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR               TTT        LLL
BBB         BBB   AAAAAAAAAAAAAAA           SSS  RRR   RRR               TTT        LLL
BBB         BBB   AAA         AAA           SSS  RRR      RRR            TTT        LLL
BBB         BBB   AAA         AAA           SSS  RRR   RRR               TTT        LLL
BBB         BBB   AAA         AAA           SSS  RRR      RRR            TTT        LLL
BBBBBBBBBBBB      AAA         AAA   SSSSSSSSSSS     RRR         RRR      TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB      AAA         AAA   SSSSSSSSSSS     RRR         RRR      TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB      AAA         AAA   SSSSSSSSSSS     RRR         RRR      TTT        LLLLLLLLLLLLLLL
```

FILEID**BPAMOVTUC

```
BBBBBBBB    PPPPPPPP      AAAAAA    MM      MM    000000    VV      VV  TTTTTTTTTT  UU      UU    CCCCCCCC
BBBBBBBB    PPPPPPPP      AAAAAA    MM      MM    000000    VV      VV  TTTTTTTTTT  UU      UU    CCCCCCCC
BB      BB  PP      PP   AA    AA   MMMM  MMMM  00    00  VV      VV      TT      UU      UU  CC
BB      BB  PP      PP   AA    AA   MMMM  MMMM  00    00  VV      VV      TT      UU      UU  CC
BB      BB  PP      PP   AA    AA   MM  MM  MM  00    00  VV      VV      TT      UU      UU  CC
BB      BB  PP      PP   AA    AA   MM  MM  MM  00    00  VV      VV      TT      UU      UU  CC
BBBBBBBB    PPPPPPPP     AA    AA   MM      MM  00    00  VV      VV      TT      UU      UU  CC
BBBBBBBB    PPPPPPPP     AA    AA   MM      MM  00    00  VV      VV      TT      UU      UU  CC
BB      BB  PP           AAAAAAAAAA MM      MM  00    00  VV      VV      TT      UU      UU  CC
BB      BB  PP           AAAAAAAAAA MM      MM  00    00  VV      VV      TT      UU      UU  CC
BB      BB  PP           AA    AA   MM      MM  00    00    VV  VV        TT      UU      UU  CC              ....
BB      BB  PP           AA    AA   MM      MM  00    00    VV  VV        TT      UU      UU  CC              ....
BBBBBBBB    PP           AA    AA   MM      MM    000000      VV          TT      UUUUUUUUUU    CCCCCCCC      ....
BBBBBBBB    PP           AA    AA   MM      MM    000000      VV          TT      UUUUUUUUUU    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  BPA$$MOVTUC     ; MOVTUC routine
0000     2          .IDENT  /1-001/         ; File: BPAMOVTUC.MAR
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 ; FACILITY: VAX-11 BASIC Miscellaneous
0000    30 ;++
0000    31 ; ABSTRACT:
0000    32 ;
0000    33 ;       This module contains a MOVTUC instruction, and enough parameters
0000    34 ;       to allow it to be used effectively from BLISS.
0000    35 ;
0000    36 ;--
0000    37 ;
0000    38 ; VERSION: 1
0000    39 ;
0000    40 ; HISTORY:
0000    41 ;
0000    42 ; AUTHOR:
0000    43 ;       John Sauter, 03-DEC-1979
0000    44 ;
0000    45 ; MODIFIED BY:
0000    46 ;
0000    47 ;
0000    48 ;
```

```
                                .SBTTL  HISTORY         ; Detailed Current Edit History
                        ;
                        ; Edit History for Version 1 of BPA$$MOVTUC
                        ;
                        ; 1-001 - Original.  JBS 03-DEC-1979
```

```
                     .SBTTL  DECLARATIONS

;
; INCLUDE FILES:
;
;            NONE
;
; EXTERNAL SYMBOLS:
;
;            NONE
                     .DSABL  GBL
;
; MACROS:
;
;            NONE
;
; PSECT DECLARATIONS:
00000000             .PSECT  _BPA$CODE       PIC, SHR, LONG, EXE, NOWRT

;
; EQUATED SYMBOLS:
;
;            NONE
;
;
; OWN STORAGE:
;
;            NONE
```

```
                     0000     86          .SBTTL  BPA$$MOVTUC
                     0000     87
                     0000     88 ;++
                     0000     89 ; FUNCTIONAL DESCRIPTION:
                     0000     90 ;
                     0000     91 ;       Allows a BLISS program access to all of the fa    ties of
                     0000     92 ;       the MOVTUC instruction.
                     0000     93 ;
                     0000     94 ; CALLING SEQUENCE: (assuming the translate is of ASCII text)
                     0000     95 ;
                     0000     96 ;       CALL BPA$$MOVTUC (SRCLEN.rw.r, SRCADDR.rt.r, ESC_CHR.rt.r, TBLADR.wt.r, DSTL
                     0000     97 ;               DSTADDR.wt.r, ESC_FLAG.wl.r, SRC_LEFT.ww.r, SRC_NEXT.wa.r, DEST_LEFT
                     0000     98 ;
                     0000     99 ; INPUT PARAMETERS:
                     0000    100 ;
00000004             0000    101         SRCLEN = 4;     Length of the source string
00000008             0000    102         SRCADDR = 8;    Base of the source string
0000000C             0000    103         ESC_CHR = 12;   The escape character
00000010             0000    104         TBLADR = 16;    Translation table
00000014             0000    105         DSTLEN = 20;    Length of the destination string
00000018             0000    106         DSTADDR = 24;   Base of the destination string
                     0000    107 ;
                     0000    108 ; IMPLICIT INPUTS:
                     0000    109 ;
                     0000    110 ;       NONE
                     0000    111 ;
                     0000    112 ; OUTPUT PARAMETERS:
                     0000    113 ;
0000001C             0000    114         ESC_FLAG = 28;  1 if the translation was terminated by escape, 0 if not.
00000020             0000    115         SRC_LEFT = 32;  0 if the string was translated without escape, otherwise the
                     0000    116 ;                       number of untranslated bytes in the source string, including
                     0000    117 ;                       byte which caused the escape
00000024             0000    118         SRC_NEXT = 36;  Address of the byte in the source string which caused the es
                     0000    119 ;                       or destination string exhaustion, or, if no exhaustion or es
                     0000    120 ;                       the address one byte beyond the source string
00000028             0000    121         DEST_LEFT = 40; Number of bytes remaining in the destination string
0000002C             0000    122         DEST_NEXT = 44; Address of the byte in the destination string which would ha
                     0000    123 ;                       received the translated byte that caused the escape, or woul
                     0000    124 ;                       have received a translated byte if the source string were no
                     0000    125 ;                       exhausted; or, if no exhaustion or escape, the address of on
                     0000    126 ;                       byte beyond the destination string.
                     0000    127 ;
                     0000    128 ; IMPLICIT OUTPUTS:
                     0000    129 ;
                     0000    130 ;       NONE
                     0000    131 ;
                     0000    132 ; FUNCTION VALUE:
                     0000    133 ;
                     0000    134 ;       NONE
                     0000    135 ;
                     0000    136 ; SIDE EFFECTS:
                     0000    137 ;
                     0000    138 ;       NONE
                     0000    139 ;
                     0000    140 ;--
                     003C    141         .ENTRY  BPA$$MOVTUC, ^M <R2, R3, R4, R5>        ; Save registers through R5
10 BC   0C BC   08 BC   04 BC   2F    0002    142         MOVTUC  @SRCLEN(AP), @SRCADDR(AP), @ESC_CHR(AP), @TBLADR(AP), @DSTLEN(AP), @
```

```
                18 BC    14 BC      000B
                            05  1D  000F   143          BVS     10$
                         1C BC  D4  0011   144          CLRL    @ESC_FLAG(AP)                   ; Not terminated by escape
                            04  11  0014   145          BRB     20$
                   1C BC    01  D0  0016   146 10$:     MOVL    #1, @ESC_FLAG(AP)               ; Terminated by escape
                   20 BC    50  D0  001A   147 20$:     MOVL    R0, @SRC_LEFT(AP)
                   24 BC    51  D0  001E   148          MOVL    R1, @SRC_NEXT(AP)
                   28 BC    54  D0  0022   149          MOVL    R4, @DEST_LEFT(AP)
                   2C BC    55  D0  0026   150          MOVL    R5, @DEST_NEXT(AP)
                               04  002A   151          RET                                     ; Return to caller
                                   002B   152 ;
                                   002B   153          .END
```

```
BPA$$MOVTUC      00000000 RG    01
DEST_LEFT      = 00000028
DEST_NEXT      = 0000002C
DSTADDR        = 00000018
DSTLEN         = 00000014
ESC_CHR        = 0000000C
ESC_FLAG       = 0000001C
SRCADDR        = 00000008
SRCLEN         = 00000004
SRC_LEFT       = 00000020
SRC_NEXT       = 00000024
TBLADR         = 00000010
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
_BPA$CODE	0000002B (43.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.08	00:00:00.50
Command processing	118	00:00:00.49	00:00:03.02
Pass 1	68	00:00:00.44	00:00:01.29
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	43	00:00:00.31	00:00:01.00
Symbol table output	3	00:00:00.02	00:00:00.02
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	271	00:00:01.36	00:00:05.85

```
The working set limit was 750 pages.
1772 bytes (4 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 12 non-local and 2 local symbols.
153 source lines were read in Pass 1, producing 10 object records in Pass 2.
0 pages of virtual memory were used tc define 0 macros.
```

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYSLIB]STARLET.MLB;2	0

```
0 GETS were required to define 0 macros.

There were no error    arnings or information messages.

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

BOOTBLOCK
MAP

STACONFIG
MAP

SYSBOOT
MAP

BPAMOVTUC
LIS

BOOTS

BPASSDEAS
LIS

BOOT58
MAP

STANDCONF
MAP

CONFIGURE
MAP

BPAWAKEUP
LIS

STASYSGEN
MAP

BPASETPRI
LIS