```
RRRRRRRRRRRR     UUU         UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF  FFFFFFFFFFFFFFF
RRRRRRRRRRRR     UUU         UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF  FFFFFFFFFFFFFFF
RRRRRRRRRRRR     UUU         UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF  FFFFFFFFFFFFFFF
RRR         RRR  UUU         UUU  NNN         NNN  000         000  FFF              FFF
RRR         RRR  UUU         UUU  NNN         NNN  000         000  FFF              FFF
RRR         RRR  UUU         UUU  NNN         NNN  000         000  FFF              FFF
RRR         RRR  UUU         UUU  NNNNNN      NNN  000         000  FFF              FFF
RRR         RRR  UUU         UUU  NNNNNN      NNN  000         000  FFF              FFF
RRR         RRR  UUU         UUU  NNNNNN      NNN  000         000  FFF              FFF
RRRRRRRRRRRR     UUU         UUU  NNN   NNN   NNN  000         000  FFFFFFFFFFFFF    FFFFFFFFFFFFF
RRRRRRRRRRRR     UUU         UUU  NNN   NNN   NNN  000         000  FFFFFFFFFFFFF    FFFFFFFFFFFFF
RRRRRRRRRRRR     UUU         UUU  NNN   NNN   NNN  000         000  FFFFFFFFFFFFF    FFFFFFFFFFFFF
RRR   RRR        UUU         UUU  NNN      NNNNNN  000         000  FFF              FFF
RRR   RRR        UUU         UUU  NNN      NNNNNN  000         000  FFF              FFF
RRR   RRR        UUU         UUU  NNN      NNNNNN  000         000  FFF              FFF
RRR      RRR     UUU         UUU  NNN         NNN  000         000  FFF              FFF
RRR      RRR     UUU         UUU  NNN         NNN  000         000  FFF              FFF
RRR      RRR     UUU         UUU  NNN         NNN  000         000  FFF              FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN         NNN     000000000     FFF              FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN         NNN     000000000     FFF              FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN         NNN     000000000     FFF              FFF
```

```
_$2

Sym
---
NDX
NDX
NUM
NUM
OPE
OUT
PAC
PAC
PAC
PAC
PAC
PAC
PAC
PAD
PAG
PAG
PAG
PAG
PAG
PAG
PAG
PER
PUT
RCO
RIN

RLI
RNO
RNO
RTY
SAV
STR
STR
STR
STR
STR
STR

STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
```

FILEID**CONVBR   G 16

```
  CCCCCCCC    OOOOOO    NN      NN  VV      VV  BBBBBBBB    RRRRRRRR
  CCCCCCCC    OOOOOO    NN      NN  VV      VV  BBBBBBBB    RRRRRRRR
CC          OO      OO  NN      NN  VV      VV  BB      BB  RR      RR
CC          OO      OO  NN      NN  VV      VV  BB      BB  RR      RR
CC          OO      OO  NNNN    NN  VV      VV  BB      BB  RR      RR
CC          OO      OO  NNNN    NN  VV      VV  BB      BB  RR      RR
CC          OO      OO  NN  NN  NN  VV      VV  BBBBBBBB    RRRRRRRR
CC          OO      OO  NN  NN  NN  VV      VV  BBBBBBBB    RRRRRRRR
CC          OO      OO  NN    NNNN  VV      VV  BB      BB  RR  RR
CC          OO      OO  NN    NNNN  VV      VV  BB      BB  RR  RR
CC          OO      OO  NN      NN    VV  VV    BB      BB  RR    RR        ....
CC          OO      OO  NN      NN    VV  VV    BB      BB  RR    RR        ....
  CCCCCCCC    OOOOOO    NN      NN      VV      BBBBBBBB    RR      RR      ....
  CCCCCCCC    OOOOOO    NN      NN      VV      BBBBBBBB    RR      RR      ....

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
```

```
  0 MODULE CONVBR (                                         !
  0                 IDENT = 'V04-000'
P 0 %BLISS32[,
P 0         ADDRESSING_MODE(EXTERNAL=LONG_RELATIVE,NONEXTERNAL=LONG_RELATIVE)
  0         ]
  0                 ) =
  1 BEGIN
  1 !
  1 !****************************************************************************
  1 !*                                                                          *
  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
  1 !*  ALL RIGHTS RESERVED.                                                    *
  1 !*                                                                          *
  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
  1 !*  TRANSFERRED.                                                            *
  1 !*                                                                          *
  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
  1 !*  CORPORATION.                                                            *
  1 !*                                                                          *
  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
  1 !*                                                                          *
  1 !*                                                                          *
  1 !****************************************************************************
  1 !
  1 !++
  1 ! FACILITY:      DSR (Digital Standard RUNOFF) / DSRPLUS
  1 !
  1 ! ABSTRACT: Convert a binary number into a vector of roman numerals and
  1 !           return the result and character count.
  1 !
  1 !
  1 ! ENVIRONMENT: Transportable
  1 !
  1 ! AUTHOR: R.W.Friday     CREATION DATE: April, 1979
  1 !
```

```
;     44          0043  1 %SBTTL 'Revision History'
;     45          0044  1 !
;     46          0045  1 ! MODIFIED BY:
;     47          0046  1 !
;     48          0047  1 !       003     KFA00003        Ken Alden       07-Mar-1983
;     49          0048  1 !               Global edit of all modules. Updated module names, idents,
;     50          0049  1 !               copyright dates. Changed require files to BLISS library.
;     51          0050  1 !
;     52          0051  1 !--
```

```
    1 %SBTTL 'Module Level Declarations'
    1
    1 !
    1 ! MACROS:
    1 !
    1 MACRO
M   1     R10(C) =
M   1         IF C EQL %C'i' THEN %C'x'
M   1         ELSE
M   1         IF C EQL %C'x' THEN %C'c'
M   1         ELSE
M   1         IF C EQL %C'c' THEN %C'm'
M   1         ELSE
M   1         IF C EQL %C'v' THEN %C'l'
M   1         ELSE
M   1         IF C EQL %C'l' THEN %C'd'
M   1         ELSE
    1                 %C'*'                     %;
    1
    1 MACRO
M   1     RPLIT (S) =
    1         CH$PTR( UPLIT(%STRING(%CHAR(%CHARCOUNT(S)),S)) ) %;
    1
    1 !
    1 ! EQUATED SYMBOLS:
    1 !
    1
    1 !
    1 ! OWN STORAGE:
    1 !
    1
    1 BIND
    1     ROM_TAB = UPLIT (
    1                           RPLIT('0'),             !0
    1                           RPLIT('i'),             !1
    1                           RPLIT('ii'),            !2
    1                           RPLIT('iii'),           !3
    1                           RPLIT('iv'),            .4
    1                           RPLIT('v'),             !5
    1                           RPLIT('vi'),            !6
    1                           RPLIT('vii'),           !7
    1                           RPLIT('viii'),          !8
    1                           RPLIT('ix')             .9
    1                                   ): VECTOR;
```

```
GLOBAL ROUTINE CONVBR (BINARY_NUMBER, KHARACTERS, KHARACTER_COUNT, ULM) : NOVALUE =

!++
! FUNCTIONAL DESCRIPTION:
!
!       Converts 'binary_number' to a vector of roman numerals,
!       returning them in 'kharacters'; kharacter_count is the
!       number of characters that result.
!       The absolute value of 'binary_number' is converted,
!       so that the user is responsible for handling negative numbers.
!       ULM is as follows: -1 means return all characters in upper case.
!                           0 means return all characters in lower case.
!                          +1 means first character in upper case, rest in lower case.
!
!                                   NOTE
!       The algorithm used here does not take into account many
!       special cases for which shorter Roman forms are possible.
!       For example, 45 is translated to XLV, even though VL is
!       obviously shorter.  This is not a bug, but is well within
!       the tradition of how Roman numerals were formulated.
!       The Romans themselves did not always use the "subtractive"
!       principle in its fullest;  it is possible to find Roman
!       inscriptions that use IIXX for 18, for example.  Similarily,
!       VIIII is not uncommon for 9.
!
! FORMAL PARAMETERS:
!
!       See FUNCTIONAL DESCRIPTION
!
! IMPLICIT INPUTS:      None
!
! IMPLICIT OUTPUTS:     None
!
! ROUTINE VALUE:
! COMPLETION CODES:     None
!
! SIDE EFFECTS: None
!
!--

    BEGIN

    BIND
        ROM_DIGITS = .KHARACTERS : VECTOR;

    LOCAL
        DEC_DIGITS : VECTOR [20],
        DEC_DIG_COUNT,
        ROM_DIG_COUNT,
        T,
        X,
        XC;

    !First get all the decimal digits separated out.
    DEC_DIG_COUNT = 0;                          !Assume user supplied zero.
    ROM_DIG_COUNT = 0;                          !...

```

```
2      !Force number into correct range.
2      T = ABS(.BINARY_NUMBER) MOD 4000;
2
2      !Special case, if user supplied zero.  In such a case return '0'.
2      IF  .T EQL 0
2      THEN
3          BEGIN
3          .KHARACTER_COUNT = 1;
3          ROM_DIGITS [0] = %C'0';
3          RETURN
2          END;
2
2      !And now do strip off the digits, one by one.
2      WHILE (.T NEQ 0) DO
3          BEGIN
3          DEC_DIGITS [.DEC_DIG_COUNT] = .T MOD 10;
3          T = .T/10;
3          DEC_DIG_COUNT = .DEC_DIG_COUNT + 1;
2          END;
2
2      !Convert decimal digits to roman numerals.
2      DECR I FROM (.DEC_DIG_COUNT - 1) TO 0  DO
3          BEGIN
3          !Prior to converting the next decimal digit, do the equivalent
3          !of multiplying the partial roman numeral result by 10.
3          INCR J FROM 0 TO (.ROM_DIG_COUNT - 1) DO
3              ROM_DIGITS [.J] = (R10(.ROM_DIGITS [.J]));
3
3          !Now convert the next decimal digit.  This is done by
3          !a simple table lookup, followed by copying into ROM_DIGITS.
3          X = .ROM_TAB[.DEC_DIGITS[.I]];           !Look up the roman equivalent of this digit.
3          XC = CH$RCHAR_A(X);                      !Get the digit count into XC.
3          !Discard zeroes (i.e., 10, 20, etc) but continue in the loop so
3          !what's already been converted gets multiplied by 10.
3          IF .DEC_DIGITS [.I] NEQ 0
3          THEN
3              !Not zero, so convert it as usual
3              INCR J FROM (.ROM_DIG_COUNT + 1) TO (.ROM_DIG_COUNT + .XC) DO
4                  BEGIN
4                  ROM_DIGITS [.J - 1] = CH$RCHAR_A(X);
4                  ROM_DIG_COUNT = .J;                          !Update current length.
3                  END;
2          END;
2
2      !Set up the length for the user.
2      .KHARACTER_COUNT = .ROM_DIG_COUNT;
2
2      !Apply case conversion rules.
2      IF  .ULM EQL 0
2      THEN
2          !User is content with lower case, so just return.
2          RETURN;
2
2      !Compute how many characters need to be converted to upper case.
2      !The result is saved in T.
2      IF .ULM EQL -1
2      THEN
```

```
;  213        0210  2         T = .ROM_DIG_COUNT - 1                     !Upper case
;  214        0211  2     ELSE
;  215        0212  2         T = 0;                                     !Mixed case
;  216        0213  2
;  217        0214  2     !Now loop over the characters to be converted and make them upper case.
;  218        0215  2     INCR I FROM 0 TO .T DO
;  219        0216  2         ROM_DIGITS [.I] = .ROM_DIGITS [.I] - %C'a' + %C'A';
;  220        0217  2
;  221        0218  2     RETURN;
;  222        0219  1     END;                                           !End of CONVBR


                                                                   .TITLE  CONVBR
                                                                   .IDENT  \V04-000\

                                                                   .PSECT  $PLIT$,NOWRT,NOEXE,2

                                     00  00  30  01  00000 P.AAB:  .ASCII  <1>\0\<0><0>                                   ;
                                     00  00  69  01  00004 P.AAC:  .ASCII  <1>\i\<0><0>                                   ;
                                     00  69  69  02  00008 P.AAD:  .ASCII  <2>\ii\<0>                                     ;
                                     69  69  69  03  0000C P.AAE:  .ASCII  <3>\iii\                                       ;
                                     00  76  69  02  00010 P.AAF:  .ASCII  <2>\iv\<0>                                     ;
                                     00  00  76  01  00014 P.AAG:  .ASCII  <1>\v\<0><0>                                   ;
                                     00  69  76  02  00018 P.AAH:  .ASCII  <2>\vi\<0>                                     ;
                                     69  69  76  03  0001C P.AAI:  .ASCII  <3>\vii\                                       ;
                     00  00  00  69  69  69  76  04  00020 P.AAJ:  .ASCII  <4>\viii\<0><0><0>                             ;
                                     00  78  69  02  00028 P.AAK:  .ASCII  <2>\ix\<0>                                     ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 0002C P.AAA:  .ADDRESS P.AAB, P.AAC, P.AAD, P.AAE, P.AAF, -   ;
                    00000000' 00000000' 00000000' 00000000' 00044          P.AAG, P.AAH, P.AAI, P.AAJ, P.AAK              ;

                                                        ROM_TAB=          P.AAA


                                                                   .PSECT  $CODE$,NOWRT,2

                                              00FC 00000           .ENTRY  CONVBR, Save R2,R3,R4,R5,R6,R7                  ; 0096
                          5E      B0      AE    9E 00002           MOVAB   -80(SP), SP                                     ;
                                          51    7C 00006           CLRQ    DEC_DIG_COUNT                                   ; 0150
                          50      04      AC    D0 00008           MOVL    BINARY_NUMBER, R0                               ; 0154
                                          03    18 0000C           BGEQ    1$                                              ;
                          50              50    CE 0000E           MNEGL   R0, R0                                          ;
       7E        00       50              01    7A 00011 1$:       EMUL    #1, R0, #0, -(SP)                               ;
       54        54       8E 00000FA0     8F    7B 00016           EDIV    #4000, (SP)+, T, T                              ;
                                          54    D5 0001F           TSTL    T                                               ; 0157
                                          09    12 00021           BNEQ    2$                                              ;
                     0C   BC              01    D0 00023           MOVL    #1, @KHARACTER_COUNT                            ; 0160
                     08   BC              30    D0 00027           MOVL    #48, @KHARACTERS                                ; 0161
                                                04 0002B           RET                                                     ; 0159
                                          54    D5 0002C 2$:       TSTL    T                                               ; 0166
                                          03    12 0002E           BNEQ    3$                                              ;
                                        00A2    31 00030           BRW     15$                                             ;
       7E        00       54              01    7A 00033 3$:       EMUL    #1, T, #0, -(SP)                                ; 0168
       50        50       8E              0A    7B 00038           EDIV    #10, (SP)+, R0, R0                              ;
                        6E41              50    D0 0003D           MOVL    R0, DEC_DIGITS[DEC_DIG_COUNT]                   ;
                          54              0A    C6 00041           DIVL2   #10, T                                          ; 0169
                                          51    D6 00044           INCL    DEC_DIG_COUNT                                   ; 0170
                                          E4    11 00046           BRB     2$                                              ; 0166
```

```
                              50          01  CE 00048 4$:      MNEGL   #1, J                                   ; 0178
                                          58  11 0004B          BRB     12$
                              53    08 BC40  D0 0004D 5$:       MOVL    @KHARACTERS[J], R3                      ; 0179
                  00000069    8F          53  D1 00052          CMPL    R3, #105
                                          06  12 00059          BNEQ    6$
                              53       78 8F  9A 0005B          MOVZBL  #120, R3
                                          3F  11 0005F          BRB     11$
                  00000078    8F          53  D1 00061 6$:      CMPL    R3, #120
                                          06  12 00068          BNEQ    7$
                              53       63 8F  9A 0006A          MOVZBL  #99, R3
                                          30  11 0006E          BRB     11$
                  00000063    8F          53  D1 00070 7$:      CMPL    R3, #99
                                          06  12 00077          BNEQ    8$
                              53       6D 8F  9A 00079          MOVZBL  #109, R3
                                          21  11 0007D          BRB     11$
                  00000076    8F          53  D1 0007F 8$:      CMPL    R3, #118
                                          06  12 00086          BNEQ    9$
                              53       6C 8F  9A 00088          MOVZBL  #108, R3
                                          12  11 0008C          BRB     11$
                  0000006C    8F          53  D1 0008E 9$:      CMPL    R3, #108
                                          06  12 00095          BNEQ    10$
                              53       64 8F  9A 00097          MOVZBL  #100, R3
                                          03  11 0009B          BRB     11$
                              53          2A  D0 0009D 10$:     MOVL    #42, R3
                        08 BC40           53  D0 000A0 11$:     MOVL    R3, @KHARACTERS[J]
              A4              50          52  F2 000A5 12$:     AOBLSS  ROM_DIG_COUNT, J, 5$
                              50        6E41  D0 000A9          MOVL    DEC_DIGITS[I], R0                       ; 0183
                              56 00000000'EF40 D0 000AD         MOVL    ROM_TAB[R0], X
                              57          86  9A 000B5          MOVZBL  (X)+, XC                                ; 0184
                                          50  D5 000B8          TSTL    R0                                      ; 0187
                                          19  13 000BA          BEQL    15$
              55              52          57  C1 000BC          ADDL3   XC, ROM_DIG_COUNT, R5                   ; 0190
                              50          52  D0 000C0          MOVL    ROM_DIG_COUNT, J
                                          0C  11 000C3          BRB     14$
                              53    08 BC40  DE 000C5 13$:      MOVAL   @KHARACTERS[J], R3                      ; 0192
                        FC    A3          86  9A 000CA          MOVZBL  (X)+, -4(R3)
                              52          50  D0 000CE          MOVL    J, ROM_DIG_COUNT                        ; 0193
              F0              50          55  F3 000D1 14$:     AOBLEQ  R5, J, 13$                              ; 0190
                              02          51  F4 000D5 15$:     SOBGEQ  I, 16$                                  ; 0174
                                          03  11 000D8          BRB     17$
                                        FF6B  31 000DA 16$:     BRW     4$
                        0C    BC          52  D0 000DD 17$:     MOVL    ROM_DIG_COUNT, @KHARACTER_COUNT         ; 0198
                                     10   AC  D5 000E1          TSTL    ULM                                     ; 0201
                                          20  13 000E4          BEQL    22$
                  FFFFFFFF    8F     10   AC  D1 000E6          CMPL    ULM, #-1                                ; 0208
                                          06  12 000EE          BNEQ    18$
                              54     FF   A2  9E 000F0          MOVAB   -1(R2), T                               ; 02[illegible]
                                          02  11 000F4          BRB     19$
                                          54  D4 000F6 18$:     CLRL    T                                       ; 0212
                              50          01  CE 000F8 19$:     MNEGL   #1, I                                   ; 0215
                                          05  11 000FB          BRB     21$
                        08 BC40           20  C2 000FD 20$:     SUBL2   #32, @KHARACTERS[I]                     ; 0216
              F7              50          54  F3 00102 21$:     AOBLEQ  T, I, 20$
                                          04 00106 22$:         RET                                             ; 0219

; Routine Size:  263 bytes,    Routine Base:  $CODE$ + 0000
;  223          0220  1
;  224          0221  1 END                                              End of module
;  225          0222  0 ELUDOM



;                                     PSECT SUMMARY
;
;       Name                        Bytes                      Attributes
;
;   $PLIT$                            84  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                           263  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                                     COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:CONVBR/OBJ=OBJ$:CONVBR MSRC$:CONVBR/UPDATE=(ENH$:CONVBR)

; Size:          263 code + 84 data bytes
; Run Time:         00:05.2
; Elapsed Time:     00:14.5
; Lines/CPU Min:     2566
; Lexemes/CPU-Min: 16647
; Memory  sed:  74 pages
; Compilation Complete
```

0338 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CONVBB
LIS

CNVDAT
LIS

CONVBR
LIS

CNTVMSMSG
LIS

CONTENTS
LIS

CNTVMS
LIS

CNTVRS
LIS

CONTRL
LIS

CONVBL
LIS

0339 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

DSPHL
LIS

CONVLB
LIS

DOFLG
LIS

DSPENT
LIS

DSRLIB
LIS

DOCASE
LIS

DOCM
LIS

DOOPTS
LIS

DSPPAG
LIS

DLE
LIS