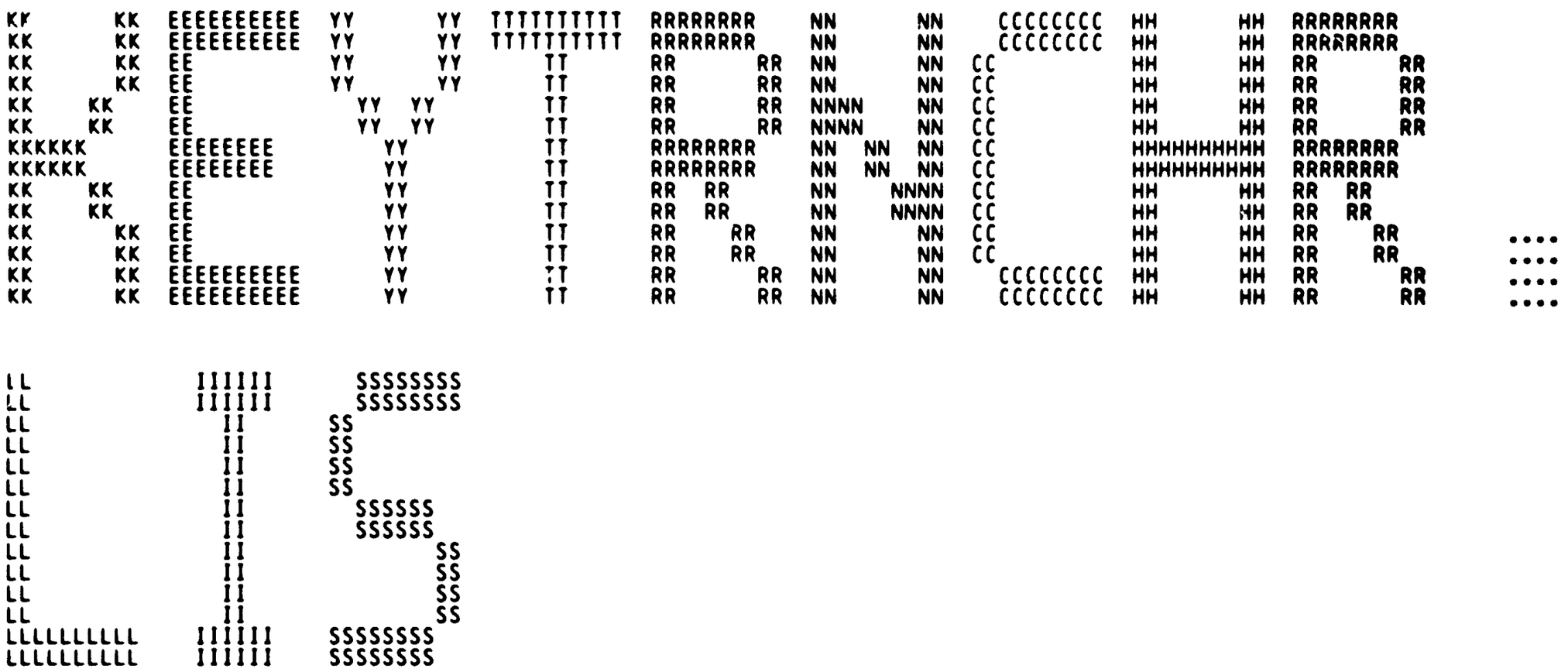
```
EEEEEEEEEEEEEEE  DDDDDDDDDDD     TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD    TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD    TTTTTTTTTTTTTTT
EEE              DDD         DDD       TTT
EEE              DDD          DDD      TTT
EEE              DDD          DDD      TTT
EEE              DDD          DDD      TTT
EEE              DDD          DDD      TTT
EEE              DDD          DDD      TTT
EEEEEEEEEEEE     DDD          DDD      TTT
EEEEEEEEEEEE     DDD          DDD      TTT
EEEEEEEEEEEE     DDD          DDD      TTT
EEE              DDD          DDD      TTT
EEE              DDD          DDD      TTT
EEE              DDD          DDD      TTT
EEE              DDD          DDD      TTT
EEE              DDD          DDD      TTT
EEE              DDD          DDD      TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDD          TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD         TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD         TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD         TTT
```

FILEID**KEYTRNCHR



EDT
V04

```
;  1     0001  0 %TITLE 'EDT$KEYTRNCHR - translate keypad key'
;  2     0002  0 MODULE EDT$KEYTRNCHR (                          ! Translate keypad key
;  3     0003  0                 IDENT = 'V04-000'                       ! File: KEYTRNCHR.BLI Edit: JBS1005
;  4     0004  0                 ) =
;  5     0005  1 BEGIN
;  6     0006  1 !
;  7     0007  1 !**************************************************************************
;  8     0008  1 !*                                                                        *
;  9     0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
; 10     0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
; 11     0011  1 !*  ALL RIGHTS RESERVED.                                                   *
; 12     0012  1 !*                                                                        *
; 13     0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
; 14     0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 15     0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
; 16     0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
; 17     0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
; 18     0018  1 !*  TRANSFERRED.                                                           *
; 19     0019  1 !*                                                                        *
; 20     0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
; 21     0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
; 22     0022  1 !*  CORPORATION.                                                           *
; 23     0023  1 !*                                                                        *
; 24     0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
; 25     0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
; 26     0026  1 !*                                                                        *
; 27     0027  1 !*                                                                        *
; 28     0028  1 !**************************************************************************
; 29     0029  1 !
; 30     0030  1
; 31     0031  1 !++
; 32     0032  1 ! FACILITY:     EDT -- The DEC Standard Editor
; 33     0033  1 !
; 34     0034  1 ! ABSTRACT:
; 35     0035  1 !
; 36     0036  1 !       Translate keypad key.
; 37     0037  1 !
; 38     0038  1 ! ENVIRONMENT:  Runs at any access mode - AST reentrant
; 39     0039  1 !
; 40     0040  1 ! AUTHOR: Bob Kushlis, CREATION DATE: April 7, 1979
; 41     0041  1 !
; 42     0042  1 ! MODIFIED BY:
; 43     0043  1 !
; 44     0044  1 ! 1-001 - Original.  DJS 24-Feb-1981.  This module was created by
; 45     0045  1 !       extracting routine EDT$$TRN_KPADK  from module KEYTRAN.
; 46     0046  1 ! 1-002 - Regularize headers.  JBS 10-Mar-1981
; 47     0047  1 ! 1-003 - Add return values.  JBS 02-Oct-1981
; 48     0048  1 ! 1-004 - New implementation of defined keys.  JBS 13-Aug-1982
; 49     0049  1 ! 1-005 - Add conditional for VT220 support.  JBS 11-Feb-1983
; 50     0050  1 !--
; 51     0051  1
```

```
;    53          0052  1 %SBTTL 'Declarations'
;    54          0053  1 !
;    55          0054  1 ! TABLE OF CONTENTS:
;    56          0055  1 !
;    57          0056  1
;    58          0057  1 REQUIRE 'EDTSRC:TRAROUNAM';
;    59          0496  1
;    60          0497  1 FORWARD ROUTINE
;    61          0498  1     EDT$$TRN_KPADK;                         ! Translate a keypad key
;    62          0499  1
;    63          0500  1 !
;    64          0501  1 ! INCLUDE FILES:
;    65          0502  1 !
;    66          0503  1
;    67          0504  1 REQUIRE 'EDTSRC:EDTREQ';
;    68          0639  1
;    69          0640  1 LIBRARY 'EDTSRC:KEYPADDEF';
;    70          0641  1
;    71          0642  1 LIBRARY 'EDTSRC:SUPPORTS';
;    72          0643  1
;    73          0644  1 !
;    74          0645  1 ! MACROS:
;    75          0646  1 !
;    76          0647  1 !       NONE
;    77          0648  1 !
;    78          0649  1 ! EQUATED SYMBOLS:
;    79          0650  1 !
;    80          0651  1 !+
;    81          0652  1 ! The following table is searched to find the meaning of
;    82          0653  1 ! any keypad key.  The ordinal position in the table
;    83          0654  1 ! corresponds to the definitions for special keypad keys.
;    84          0655  1 !-
;    85          0656  1
;    86          0657  1 BIND
;    87          0658  1     KEY_PAD = UPLIT BYTE
;    88          0659  1             ('pqrstuvwxyQRABCDnSmlPM') : VECTOR [22, BYTE];
;    89          0660  1
;    90          0661  1 !
;    91          0662  1 ! OWN STORAGE:
;    92          0663  1 !
;    93          0664  1 !       NONE
;    94          0665  1 !
;    95          0666  1 ! EXTERNAL REFERENCES:
;    96          0667  1 !
;    97          0668  1 !       In the routine
```

```
;   99          0669  1 %SBTTL 'EDT$$TRN_KPADK  - translate a keypad key'
;  100          0670  1
;  101          0671  1 GLOBAL ROUTINE EDT$$TRN_KPADK (                    ! Translate a keypad key
;  102          0672  1     C                                              ! Where to store the translation
;  103          0673  1     ) =
;  104          0674  1
;  105          0675  1 !++
;  106          0676  1 ! FUNCTIONAL DESCRIPTION:
;  107          0677  1 !
;  108          0678  1 !       This routine is called when an escape character has been seen.  The escape
;  109          0679  1 !       or control sequence is read up to the terminator, and, if it is a sequence generated
;  110          0680  1 !       by a keypad or function key, the numeric value for that key is returned; otherwise,
;  111          0681  1 !       the terminator character is returned.
;  112          0682  1 !
;  113          0683  1 ! FORMAL PARAMETERS:
;  114          0684  1 !
;  115          0685  1 !  C                    The address of a fullword to receive the translated character.
;  116          0686  1 !
;  117          0687  1 ! IMPLICIT INPUTS:
;  118          0688  1 !
;  119          0689  1 !       NONE
;  120          0690  1 !
;  121          0691  1 ! IMPLICIT OUTPUTS:
;  122          0692  1 !
;  123          0693  1 !       NONE
;  124          0694  1 !
;  125          0695  1 ! ROUTINE VALUE:
;  126          0696  1 !
;  127          0697  1 !       1 = ok, 0 = end of journal file
;  128          0698  1 !
;  129          0699  1 ! SIDE EFFECTS:
;  130          0700  1 !
;  131          0701  1 !       Calls EDT$$TI_INPCH  to read from the keyboard.
;  132          0702  1 !
;  133          0703  1 !--
;  134          0704  1
;  135          0705  2     BEGIN
;  136          0706  2
;  137          0707  2     EXTERNAL ROUTINE
;  138          0708  2         EDT$$TI_INPCH;
;  139          0709  2
;  140        L 0710  2 %IF SUPPORT_VT220
;  141          0711  2 %THEN
;  142          0712  2
;  143          0713  2     EXTERNAL
;  144          0714  2         EDT$$B_CHAR_INFO : BLOCKVECTOR [256, 1, BYTE];  ! Information about characters
;  145          0715  2
;  146          0716  2 %FI
;  147          0717  2
;  148          0718  2     LOCAL
;  149          0719  2         MY_C,
;  150          0720  2         FUN_VAL;
;  151          0721  2
;  152          0722  2     FUN_VAL = 0;
;  153          0723  2 !+
;  154          0724  2 ! Keep reading characters as long as they are intermediate characters of
;  155          0725  2 ! VT52 or VT100 keypad sequences.
```

```
;  156          0726  2 !-
;  157          0727  2
;  158          0728  2     DO
;  159          0729  3         BEGIN
;  160          0730  3
;  161          0731  3         IF (EDT$$TI_INPCH (MY_C) EQL 0) THEN RETURN (0);
;  162          0732  3
;  163        L 0733  3 %IF SUPPORT_VT220
;  164          0734  3 %THEN
;  165          0735  3
;  166          0736  4         IF (.EDT$$B_CHAR_INFO [.MY_C, 0, 0, 8, 0] EQL %X'F0')
;  167          0737  3         THEN
;  168          0738  4             BEGIN
;  169          0739  4 !+
;  170          0740  4 ! Accumulate a number.
;  171          0741  4 !-
;  172          0742  4             FUN_VAL = .FUN_VAL*10;
;  173          0743  4             FUN_VAL = .FUN_VAL + (.MY_C - %C'0');
;  174          0744  4
;  175          0745  4             IF (.FUN_VAL GTR K_MAX_FUN_VAL) THEN FUN_VAL = 0;
;  176          0746  4
;  177          0747  3             END;
;  178          0748  3
;  179          0749  3 %FI
;  180          0750  3
;  181          0751  3         END
;  182          0752  3     UNTIL ((.MY_C NEQ %C'?') AND                    !
;  183          0753  3         (.MY_C NEQ %C'O') AND                       !
;  184          0754  4         (.MY_C NEQ %C'[')
;  185          0755  4
;  186        L 0756  4 %IF SUPPORT_VT220
;  187          0757  4 %THEN
;  188          0758  4         AND (.EDT$$B_CHAR_INFO [.MY_C, 0, 0, 8, 0] NEQ %X'F0')
;  189          0759  4 %FI
;  190          0760  4
;  191          0761  2         );
;  192          0762  2
;  193          0763  2 !+
;  194          0764  2 ! If this is a function key, we know the value from the number accumulated.
;  195          0765  2 !-
;  196          0766  2
;  197        L 0767  2 %IF SUPPORT_VT220
;  198          0768  2 %THEN
;  199          0769  2
;  200          0770  3     IF (.MY_C EQL %C'~')
;  201          0771  2     THEN
;  202          0772  2         MY_C = K_FUN_BASE + .FUN_VAL
;  203          0773  2     ELSE
;  204          0774  2 %FI
;  205          0775  2
;  206          0776  3         BEGIN
;  207          0777  3 !+
;  208          0778  3 ! Not a function key, search our table of keypad sequence terminators.
;  209          0779  3 !-
;  210          0780  3
;  211          0781  3         INCR I FROM 0 TO 21 DO
;  212          0782  3
```

```
        0783 4               IF (.MY_C EQL .KEY_PAD [.I])
        0784 3               THEN
        0785 4                   BEGIN
        0786 4  !+
        0787 4  ! The terminator was found, return it's equivalent.
        0788 4  !-
        0789 4                   MY_C = K_KPAD_BASE + .I;
        0790 4                   EXITLOOP;
        0791 3                   END;
        0792 3
        0793 2           END;
        0794 2
        0795 2  !+
        0796 2  ! Return the coded character.
        0797 2  !-
        0798 2      .C = .MY_C;
        0799 2      RETURN (T);
        0800 1      END;                                   ! of routine EDT$$TRN_KPADK

                                                       .TITLE  EDT$KEYTRNCHR EDT$KEYTRNCHR - translate keypad
;                                                                                key
                                                       .IDENT  \V04-000\

                                                       .PSECT  _EDT$CODE,NOWRT,  SHR,  PIC,2

43 42 41 52 51 79 78 77 76 75 74 73 72 71 70 00000 P.AAA:  .ASCII  \pqrstuvwxyQRABCDnSmlPM\                    ;
                        4D 50 6C 6D 53 6E 44 0000F                                                            ;

                                        KEY_PAD=              P.AAA
                                                       .EXTRN  EDT$$TI_INPCH, EDT$$B_CHAR_INFO

                                         0004 00000            .ENTRY  EDT$$TRN_KPADK, Save R2               ; 0671
                      5E            04   C2 00002            SUBL2   #4, SP                                 ;
                                    52   D4 00005            CLRL    FUN_VAL                                ; 0722
                                    5E   DD 00007 1$:        PUSHL   SP                                     ; 0731
       00000000G      00            01   FB 00009            CALLS   #1, EDT$$TI_INPCH                      ;
                                    50   D5 00010            TSTL    R0                                     ;
                                    70   13 00012            BEQL    7$                                     ;
                      'O            6E   D0 00014            MOVL    MY_C, R0                               ; 0736
                      51 00000000G0040   9A 00017            MOVZBL  EDT$$B_CHAR_INFO[R0], R1               ;
                 F0   8F            51   91 0001F            CMPB    R1, #240                               ;
                                    13   12 00023            BNEQ    2$                                     ;
                      52            0A   C4 00025            MULL2   #10, FUN_VAL                           ; 0742
                      52       D0 A042   9E 00028            MOVAB   -48(R0)[FUN_VAL], FUN_VAL              ; 0743
       00000063       8F            52   D1 0002D            CMPL    FUN_VAL, #99                           ; 0745
                                    02   15 00034            BLEQ    2$                                     ;
                                    52   D4 00036            CLRL    FUN_VAL                                ;
                      3F            50   D1 00038 2$:        CMPL    R0, #63                                ; 0752
                                    CA   13 0003B            BEQL    1$                                     ;
       0000004F       8F            50   D1 0003D            CMPL    R0, #79                                ; 0753
                                    C1   13 00044            BEQL    1$                                     ;
       0000005B       8F            50   D1 00046            CMPL    R0, #9?                                ; 0754
                                    B8   13 0004D            BEQL    1$                                     ;
                 F0   8F            51   91 0004F            CMPB    R1, #240                               ; 0758
                                    B2   13 00053            BEQL    1$                                     ;
       0000007E       8F            50   D1 00055            CMPL    R0, #126                               ; 0770
```

```
                                     07  12 0005C        BNEQ    3$                              ;
                      6E      0190   C2  9E 0005E        MOVAB   400(R2), MY_C                   ; 0772
                                     17  11 00063        BRB     6$                              ;
                                     50  D4 00065 3$:    CLRL    I                               ; 0781
      6E    FF7C CF40         08     00  ED 00067 4$:    CMPZV   #0, #8, KEY_PAD[I], MY_C        ; 0783
                                     07  12 0006F        BNEQ    5$                              ;
                      6E      012C   C0  9E 00071        MOVAB   300(R0), MY_C                   ; 0789
                                     04  11 00076        BRB     6$                              ; 0785
              EB              50     15  F3 00078 5$:    AOBLEQ  #21, I, 4$                      ; 0783
                        04    BC     6E  D0 0007C 6$:    MOVL    MY_C, @C                        ; 0798
                              50     01  D0 00080        MOVL    #1, R0                          ; 0799
                                         04 00083        RET                                     ;
                                     50  D4 00084 7$:    CLRL    R0                              ; 0800
                                         04 00086        RET                                     ;

; Routine Size:  135 bytes,    Routine Base:  _EDT$CODE + 0016


;  231          0801  1
;  232          0802  1 !<BLF/PAGE>
```

```
1 END                                          . of module EDT$KEYTRNCHR
1
0 ELUDOM
```

PSECT SUMMARY

Name	Bytes	Attributes
_EDT$CODE	157	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	-------- Symbols -------- Total	Loaded	Percent	Pages Mapped	Processing Time
_255DUA28:[EDT.SRC]EDT.L32;1	377	0	0	40	00:00.2
_255DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1
_255DUA28:[EDT.SRC]KEYPADDEF.L32;1	34	3	8	7	00:00.1
_255DUA28:[EDT.SRC]SUPPORTS.L32;1	2	1	50	5	00:00.1

COMMAND QUALIFIERS

```
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS$:KEYTRNCHR/OBJ=OBJ$:KEYTRNCHR MSRC$:KEYTRNCHR.BLI/UPDATE=(ENH$:K
;       EYTRNCHR)

; Size:          135 code + 22 data bytes
; Run Time:         00:12.8
; Elapsed Time:     00:15.8
; Lines/CPU Min:     3782
; Lexemes/CPU-Min: 10675
; Memory Used:  80 pages
; Compilation Complete
```

KEYDEFKEY
LIS

KEYFMTSTR
LIS

KEYPADDEF
LIS

LDIVISION
LIS

LDEFK
LIS

LFILL
LIS

KEYCHR
LIS

KEYIMMINP
LIS

LDELETE
LIS

KEYCOM
LIS

KEYPUTCHR
LIS

LCLEAR
LIS

LPCOUNT
LIS

KEYPAD
LIS

LDEFM
LIS

LFLNO
LIS

KEYTRNCHR
LIS