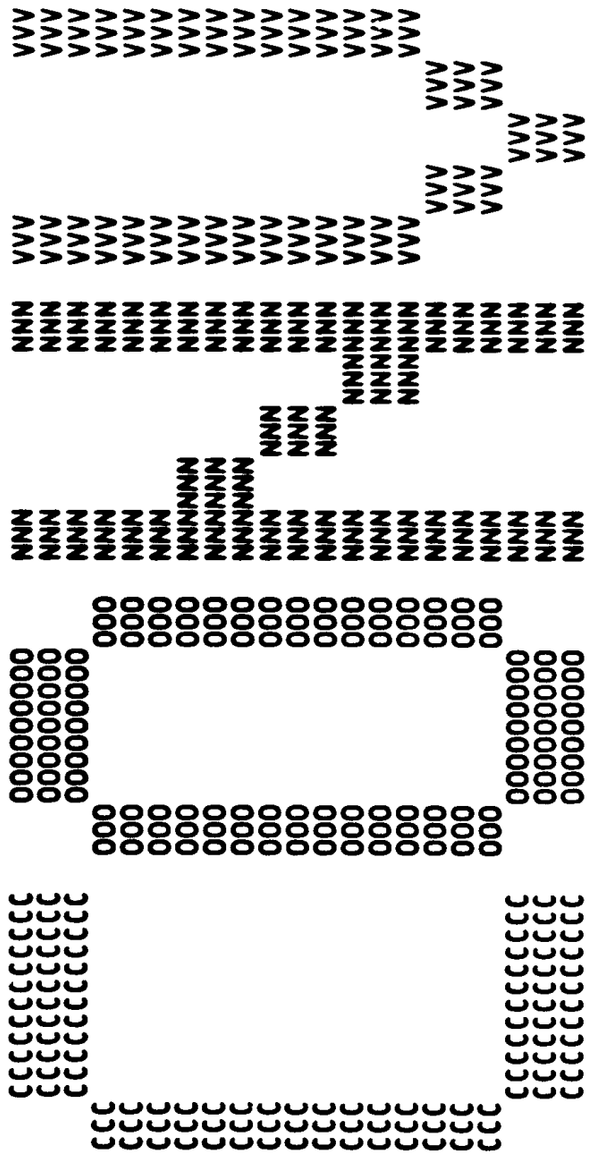

FILEID**RECLDCL

```
RRRRRRRR    EEEEEEEEEE    CCCCCCCC  LL          DDDDDDDD      CCCCCCCC  LL
RRRRRRRR    EEEEEEEEEE    CCCCCCCC  LL          DDDDDDDD      CCCCCCCC  LL
RR      RR  EE            CC        LL          DD      DD    CC        LL
RR      RR  EE            CC        LL          DD      DD    CC        LL
RR      RR  EE            CC        LL          DD      DD    CC        LL
RR      RR  EE            CC        LL          DD      DD    CC        LL
RRRRRRRR    EEEEEEEE      CC        LL          DD      DD    CC        LL
RRRRRRRR    EEEEEEEE      CC        LL          DD      DD    CC        LL
RR  RR      EE            CC        LL          DD      DD    CC        LL
RR  RR      EE            CC        LL          DD      DD    CC        LL
RR    RR    EE            CC        LL          DD      DD    CC        LL              ....
RR    RR    EE            CC        LL          DD      DD    CC        LL              ....
RR      RR  EEEEEEEEEE    CCCCCCCC  LLLLLLLLLL  DDDDDDDD      CCCCCCCC  LLLLLLLLLL      ....
RR      RR  EEEEEEEEEE    CCCCCCCC  LLLLLLLLLL  DDDDDDDD      CCCCCCCC  LLLLLLLLLL      ....

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
```

```
;  1          0001  0 %TITLE  'VAX-11 CONVERT/RECLAIM'
;  2          0002  0 MODULE  RECL$DCL        ( IDENT='V04-000',
;  3          0003  0                         MAIN=START
;  4          0004  0                         ) =
;  5          0005  0
;  6          0006  1 BEGIN
;  7          0007  1
;  8          0008  1 !**************************************************************************
;  9          0009  1 !*                                                                        *
; 10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
; 11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                *
; 12          0012  1 !*  ALL RIGHTS RESERVED.                                                  *
; 13          0013  1 !*                                                                        *
; 14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
; 15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
; 17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
; 18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
; 19          0019  1 !*  TRANSFERRED.                                                           *
; 20          0020  1 !*                                                                        *
; 21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
; 22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
; 23          0023  1 !*  CORPORATION.                                                           *
; 24          0024  1 !*                                                                        *
; 25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
; 26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
; 27          0027  1 !*                                                                        *
; 28          0028  1 !*                                                                        *
; 29          0029  1 !**************************************************************************
```

```
;    31          0030  1 !++
;    32          0031  1 !
;    33          0032  1 ! Facility:     VAX-11 CONVERT/RECLAIM
;    34          0033  1 !
;    35          0034  1 ! Environment:
;    36          0035  1 !
;    37          0036  1 !               VAX/VMS Operating System
;    38          0037  1 !
;    39          0038  1 ! Abstract:     Main routines
;    40          0039  1 !
;    41          0040  1 ! Contents:
;    42          0041  1 !               main
;    43          0042  1 !
;    44          0043  1 !--
;    45          0044  1
;    46          0045  1 !
;    47          0046  1 ! Author:       Keith B Thompson
;    48          0047  1 !               Peter Lieberwirth       Creation date:  August-1981
;    49          0048  1 !
;    50          0049  1 !
;    51          0050  1 ! Modified by:
;    52          0051  1 !
;    53          0052  1 !       V03-003 KBT0373         Keith B. Thompson       20-Oct-1982
;    54          0053  1 !               Remove the flags ref. stuff and use the new flags
;    55          0054  1 !               parameters on the call interface
;    56          0055  1 !
;    57          0056  1 !       V03-002 KBT0043         Keith Thompson          3-Apr-1982
;    58          0057  1 !               Change the refrence to conv$ab_flags to use the vector stuff
;    59          0058  1 !
;    60          0059  1 !       V03-001 KBT0020         Keith Thompson          23-Mar-1982
;    61          0060  1 !               Correct the display of CPU time
;    62          0061  1 !
;    63          0062  1 !****
```

```
;    65          0063  1
;    66          0064  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    67          0065  1 REQUIRE 'SRC$:CONVERT';
;    68          0300  1 REQUIRE 'SRC$:CONVDEF';
;    69          0461  1
;    70          0462  1 EXTERNAL ROUTINE
;    71          0463  1          LIB$INIT_TIMER          : ADDRESSING_MODE ( GENERAL ),
;    72          0464  1          CLI$GET_VALUE           : ADDRESSING_MODE ( GENERAL ),
;    73          0465  1          CLI$PRESENT             : ADDRESSING_MODE ( GENERAL ),
;    74          0466  1          CONV$RECLAIM            : ADDRESSING_MODE ( GENERAL ),
;    75          0467  1          LIB$STAT_TIMER          : ADDRESSING_MODE ( GENERAL ),
;    76          0468  1          LIB$PUT_OUTPUT          : ADDRESSING_MODE ( GENERAL );
;    77          0469  1
;    78          0470  1 FORWARD ROUTINE
;    79          0471  1          MULQ    : NOVALUE;
;    80          0472  1
;    81          0473  1 OWN
;    82          0474  1          FILE_NAME       : DESC_BLK
;    83          0475  1                                  PRESET( [ DSC$B_CLASS ] = DSC$K_CLASS_D ),
;    84          0476  1
;    85          0477  1          STATISTICS_BLOCK        : VECTOR [ 5,LONG ] INITIAL ( 4,0,0,0,0 ),
;    86          0478  1
;    87          0479  1          FLAGS           : LONG INITIAL( CONV$M_SIGNAL ),
;    88          0480  1
;    89          0481  1          !       FAO Processing
;    90          0482  1          !
;    91          0483  1          FAO_BUFFER      : VECTOR [ 132,BYTE ],
;    92          0484  1          FAO_DESC        : DESC_BLK
;    93          0485  1                                  PRESET( [ DSC$B_CLASS ] = DSC$K_CLASS_D,
;    94          0486  1                                          [ DSC$W_LENGTH ] = 132,
;    95          0487  1                                          [ DSC$A_POINTER ] = FAO_BUFFER ),
;    96          0488  1          PUT_DESC        : DESC_BLK
;    97          0489  1                                PRESET( [ DSC$B_CLASS ] = DSC$K_CLASS_D,
;    98          0490  1                                        [ DSC$W_LENGTH ] = 132,
;    99          0491  1                                        [ DSC$A_POINTER ] = FAO_BUFFER ),
;   100          0492  1
;   101          0493  1          TIMER_BLK       : LONG,
;   102          0494  1
;   103          0495  1          ELP_TIME        : VECTOR [ 2,LONG ],
;   104          0496  1          CPU_TIME        : VECTOR [ 2,LONG ],
;   105          0497  1
;   106          0498  1          ELP_TIM_BUF     : VECTOR [ 16,BYTE ],
;   107          0499  1          CPU_TIM_BUF     : VECTOR [ 16,BYTE ],
;   108          0500  1
;   109          0501  1          ELP_DESC        : DESC_BLK      INITIAL ( 16,ELP_TIM_BUF ),
;   110          0502  1          CPU_DESC        : DESC_BLK      INITIAL ( 16,CPU_TIM_BUF ),
;   111          0503  1
;   112          0504  1          ONE             : INITIAL(1),
;   113          0505  1          TWO             : INITIAL(2);
;   114          0506  1
;   115          0507  1      BIND
;   116          0508  1          STATS = UPLIT(
;   117          0509  1
;   118          0510  1          DESCRIPTOR( ' !/ RECLAIM Statistics' ),
;   119          0511  1          DESCRIPTOR( 'Total Buckets Scanned:     !8UL' ),
;   120          0512  1          DESCRIPTOR( 'Data Buckets Reclaimed:    !8UL' ),
;   121          0513  1          DESCRIPTOR( 'Index Buckets Reclaimed:   !8UL' ),
```

```
;  122          0514  1          DESCRIPTOR( 'Total Buckets Reclaimed:    !8UL' ),
;  123          0515  1          DESCRIPTOR( 'Elapsed Time:       !AS!_CPU Time:       !AS' )
;  124          0516  1
;  125          0517  1          ) : VECTOR;
;  126          0518  1
```

```
;  128          0519  1 %SBTTL  'Main Routine'
;  129          0520  1 ROUTINE START    =
;  130          0521  1 !++
;  131          0522  1 !
;  132          0523  1 ! Functional Description:
;  133          0524  1 !
;  134          0525  1 !       DCL executable inage
;  135          0526  1 !
;  136          0527  1 ! Calling Sequence:
;  137          0528  1 !
;  138          0529  1 !       By dcl
;  139          0530  1 !
;  140          0531  1 ! Input Parameters:
;  141          0532  1 !       none
;  142          0533  1 !
;  143          0534  1 ! Implicit Inputs:
;  144          0535  1 !       none
;  145          0536  1 !
;  146          0537  1 ! Output Parameters:
;  147          0538  1 !       none
;  148          0539  1 !
;  149          0540  1 ! Implicit Outputs:
;  150          0541  1 !       none
;  151          0542  1 !
;  152          0543  1 ! Routine Value:
;  153          0544  1 !       none
;  154          0545  1 !
;  155          0546  1 ! Routines Called:
;  156          0547  1 !
;  157          0548  1 !       LIB$INIT_TIMER
;  158          0549  1 !       CLI$PRESENT
;  159          0550  1 !       CLI$GET_VALUE
;  160          0551  1 !       CONV$RECLAIM
;  161          0552  1 !       LIB$STAT_TIMER
;  162          0553  1 !       $ASCTIM
;  163          0554  1 !       $FAO
;  164          0555  1 !       LIB$PUT_OUTPUT
;  165          0556  1 !
;  166          0557  1 ! Side Effects:
;  167          0558  1 !       none
;  168          0559  1 !
;  169          0560  1 !--
;  170          0561  1
;  171          0562  2     BEGIN
;  172          0563  2
;  173          0564  2     BUILTIN
;  174          0565  2         SUBM;
;  175          0566  2
;  176          0567  2     LOCAL
;  177          0568  2         STATISTICS,
;  178          0569  2         LENGTH,
;  179          0570  2         TOTAL_COUNT;
;  180          0571  2
;  181          0572  2     ! Start the timer
;  182          0573  2     !
;  183          0574  2     LIB$INIT_TIMER( TIMER_BLK );
;  184          0575  2
```

```
;  185          0576  2      ! Get the switch from DCL
;  186          0577  2      !
;  187          0578  2      STATISTICS = CLI$PRESENT( DESCRIPTOR( 'STATISTICS' ) );
;  188          0579  2
;  189          0580  2      ! Get the input file name
;  190          0581  2      !
;  191          0582  2      CLI$GET_VALUE( DESCRIPTOR( 'FILE_NAME' ),FILE_NAME );
;  192          0583  2
;  193          0584  2      ! Reclaim the file
;  194          0585  2      !
;  195          0586  2      RET_ON_ERROR( CONV$RECLAIM ( FILE_NAME,STATISTICS_BLOCK,FLAGS ) );
;  196          0587  2
;  197          0588  2      ! Output some stats if wanted
;  198          0589  2      !
;  199          0590  2      IF .STATISTICS
;  200          0591  2      THEN
;  201          0592  2          BEGIN
;  202          0593  2
;  203          0594  2          OWN
;  204          0595  2              ZERO_Q      : VECTOR [ 2,LONG ] INITIAL( 0,0 ),     ! Used for
;  205          0596  2              TEMP_TIME   : VECTOR [ 2,LONG ],                    ! conversion
;  206          0597  2              MUL100K     : VECTOR [ 2,LONG ] INITIAL( 100000,0 );! of times
;  207          0598  2
;  208          0599  2          ! Get Preformance Stats
;  209          0600  2          !
;  210          0601  2          LIB$STAT_TIMER( ONE,    ELP_TIME,       TIMER_BLK );
;  211          0602  2          LIB$STAT_TIMER( TWO,    TEMP_TIME,      TIMER_BLK );
;  212          0603  2
;  213          0604  2          ! Convert to delta time
;  214          0605  2          !
;  215          0606  2          SUBM( 2,ELP_TIME,ZERO_Q,ELP_TIME );
;  216          0607  2
;  217          0608  2          ! Convert internal times to ASCII
;  218          0609  2          !
;  219        P 0610  2          $ASCTIM( TIMLEN = 0,
;  220        P 0611  2                   TIMBUF = ELP_DESC,
;  221        P 0612  2                   TIMADR = ELP_TIME,
;  222          0613  2                   CVTFLG = 0 );
;  223          0614  2
;  224          0615  2          ! The CPU time is given in 10msec ticks so we need to convert it to
;  225          0616  2          ! system delta time
;  226          0617  2          !
;  227          0618  2          ! Convert to 10nsec ticks
;  228          0619  2          !
;  229          0620  2          MULQ( TEMP_TIME,MUL100K,CPU_TIME );
;  230          0621  2
;  231          0622  2          ! Convert to delta time
;  232          0623  2          !
;  233          0624  2          SUBM( 2,CPU_TIME,ZERO_Q,CPU_TIME );
;  234          0625  2
;  235          0626  2          ! Conver to ascii
;  236          0627  2          !
;  237        P 0628  2          $ASCTIM( TIMLEN = 0,
;  238        P 0629  2                   TIMBUF = CPU_DESC,
;  239        P 0630  2                   TIMADR = CPU_TIME,
;  240          0631  2                   CVTFLG = 0 );
;  241          0632  2
```

```
;  242          0633  3          ! Loop to output the first 5 lines of the display
;  243          0634  3          !
;  244          0635  3          INCR I FROM 0 TO 4 BY 1
;  245          0636  3          DO
;  246          0637  4              BEGIN
;  247          0638  4
;  248          0639  4              ! FAO the line
;  249          0640  4              !
;  250          0641  4              $FAO( .STATS [ .I ],LENGTH,FAO_DESC,.STATISTICS_BLOCK [ .I ] );
;  251          0642  4
;  252          0643  4              PUT_DESC [ DSC$W_LENGTH ] = .LENGTH;
;  253          0644  4
;  254          0645  4              ! Output the line
;  255          0646  4              !
;  256          0647  4              LIB$PUT_OUTPUT( PUT_DESC )
;  257          0648  4
;  258          0649  3              END;
;  259          0650  3
;  260          0651  3          ! Elasped Time and CPU Time
;  261          0652  3          !
;  262          0653  3          $FAO( .STATS [ 5 ],LENGTH,FAO_DESC,ELP_DESC,CPU_DESC );
;  263          0654  3          PUT_DESC [ DSC$W_LENGTH ] = .LENGTH;
;  264          0655  3          LIB$PUT_OUTPUT( PUT_DESC );
;  265          0656  3
;  266          0657  2          END;
;  267          0658  2
;  268          0659  2      RETURN SS$_NORMAL
;  269          0660  2
;  270          0661  1      END;

                                                                .TITLE  RECL$DCL VAX-11 CONVERT
                                                                .IDENT  \V04-000\

                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

61 74 53 20 4D 49 41 4C 43 45 52 20 2F 21 20  00000 P.AAC:   .ASCII  \ !/ RECLAIM Statistics\
                        73 63 69 74 73 69 74  0000F
                                              00016          .BLKB   2
                                 00000016     00018 P.AAB:   .LONG   22
                                 00000000'    0001C          .ADDRESS P.AAC
53 20 73 74 65 6B 63 75 42 20 6C 61 74 6F 54  00020 P.AAE:   .ASCII  \Total Buckets Scanned:     !8UL\
55 38 21 20 20 20 20 20 3A 64 65 6E 6E 61 63  0002F
                                           4C 0003E
                                              0003F          .BLKB   1
                                 0000001F     00040 P.AAD:   .LONG   31
                                 00000000'    00044          .ADDRESS P.AAE
65 52 20 73 74 65 6B 63 75 42 20 61 74 61 44  00048 P.AAG:   .ASCII  \Data Buckets Reclaimed:    !8UL\
55 38 21 20 20 20 20 3A 64 65 6D 69 61 6C 63  00057
                                           4C 00066
                                              00067          .BLKB   1
                                 0000001F     00068 P.AAF:   .LONG   31
                                 00000000'    0006C          .ADDRESS P.AAG
52 20 73 74 65 6B 63 75 42 20 78 65 64 6E 49  00070 P.AAI:   .ASCII  \Index Buckets Reclaimed:   !8UL\
55 38 21 20 20 20 3A 64 65 6D 69 61 6C 63 65  0007F
                                           4C 0008E
                                              0008F          .BLKB   1
```

```
                                                  0000001F  00090 P.AAH:  .LONG   31
                                                  00000000' 00094        .ADDRESS P.AAI
52 20 73 74 65 6B 63 75 42 20 6C 61 74 6F 54      00098 P.AAK:  .ASCII  \Total Buckets Reclaimed:   !8UL\
55 38 21 20 20 20 3A 64 65 6D 69 61 6C 63 65      000A7
                                              4C  000B6
                                                  000B7        .BLKB   1
                                                  0000001F  000B8 P.AAJ:  .LONG   31
                                                  00000000' 000BC        .ADDRESS P.AAK
20 20 3A 65 6D 69 54 20 64 65 73 70 61 6C 45      000C0 P.AAM:  .ASCII  \Elapsed Time:      !AS!_CPU Time:      !\
69 54 20 55 50 43 5F 21 53 41 21 20 20 20 20      000CF
                  21 20 20 20 20 20 20 3A 65 6D   000DE
                                        53 41     000E8        .ASCII  \AS\
                                              0A  000EA        .BLKB   2
                                                  0000002A  000EC P.AAL:  .LONG   42
                                                  00000000' 000F0        .ADDRESS P.AAM
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 000F4 P.AAA:  .ADDRESS P.AAB, P.AAD, P.AAF, P.AAH, P.AAJ, P.AAL
                  53 43 49 54 53 49 54 41 54 53   0010C P.AAO:  .ASCII  \STATISTICS\
                                                  00116        .BLKB   2
                                                  0000000A  00118 P.AAN:  .LONG   10
                                                  00000000' 0011C        .ADDRESS P.AAO
                     45 4D 41 4E 5F 45 4C 49 46   00120 P.AAQ:  .ASCII  \FILE_NAME\
                                                  00129        .BLKB   3
                                                  00000009  0012C P.AAP:  .LONG   9
                                                  00000000' 00130        .ADDRESS P.AAQ

                                                                 .PSECT  $OWN$,NOEXE,2

                                                  00# 00000 FILE_NAME:
                                                                 .BYTE   0[3]
                                                  02  00003         .BYTE   2
                                                      00004         .BLKB   4
00000000  00000000  00000000  00000000  00000004  00008 STATISTICS_BLOCK:
                                                                 .LONG   4, 0, 0, 0, 0
                                                  00000001  0001C FLAGS:  .LONG   1
                                                            00020 FAO_BUFFER:
                                                                 .BLKB   132
                                                      0084  000A4 FAO_DESC:
                                                                 .WORD   132
                                                        00  000A6         .BYTE   0
                                                        02  000A7         .BYTE   2
                                                  00000000' 000A8         .ADDRESS FAO_BUFFER
                                                      0084  000AC PUT_DESC:
                                                                 .WORD   132
                                                        00  000AE         .BYTE   0
                                                        02  000AF         .BYTE   2
                                                  00000000' 000B0         .ADDRESS FAO_BUFFER
                                                            000B4 TIMER_BLK:
                                                                 .BLKB   4
                                                            000B8 ELP_TIME:
                                                                 .BLKB   8
                                                            000C0 CPU_TIME:
                                                                 .BLKB   8
                                                            000C8 ELP_TIM_BUF:
                                                                 .BLKB   16
                                                            000D8 CPU_TIM_BUF:
                                                                 .BLKB   16
                                                  00000010  000E8 ELP_DESC:
```

```
                                                   .LONG    16                                   ;
                     00000000'  000EC              .ADDRESS ELP_TIM_BUF                          ;
                     00000010   000F0 CPU_DESC:
                                                   .LONG    16                                   ;
                     00000000'  000F4              .ADDRESS CPU_TIM_BUF                          ;
                     00000001   000F8 ONE:         .LONG    1                                    ;
                     00000002   000FC TWO:         .LONG    2                                    ;
          00000000   00000000   00100 ZERO_Q:      .LONG    0, 0                                 ;
                                00108 TEMP_TIME:
                                                   .BLKB    8
          00000000   000186A0   00110 MUL100K:     .LONG    100000, 0                            ;

                                      STATS=                   P.AAA
                                                   .EXTRN   LIB$INIT_TIMER, CLI$GET_VALUE
                                                   .EXTRN   CLI$PRESENT, CONV$RECLAIM
                                                   .EXTRN   LIB$STAT_TIMER, LIB$PUT_OUTPUT
                                                   .EXTRN   SYS$ASCTIM, SYS$FAO

                                                   .PSECT   $CODE$,NOWRT,2

                           00FC 00000 START:       .WORD    Save R2,R3,R4,R5,R6,R7               ; 0520
          57 00000000G   00   9E 00002             MOVAB    LIB$PUT_OUTPUT, R7                   ;
          56 00000000G   00   9E 00009             MOVAB    SYS$FAO, R6                          ;
          55 00000000G   00   9E 00010             MOVAB    SYS$ASCTIM, R5                       ;
          54 00000000G   00   9E 00017             MOVAB    LIB$STAT_TIMER, R4                   ;
          53      0000'  CF   9E 0001E             MOVAB    ELP_TIME, R3                         ;
          5E             04   C2 00023             SUBL2    #4, SP                               ;
                    FC   A3   9F 00026             PUSHAB   TIMER_BLK                            ; 0574
 00000000G  00           01   FB 00029             CALLS    #1, LIB$INIT_TIMER                   ;
                  0000'  CF   9F 00030             PUSHAB   P.AAN                                ; 0578
 00000000G  00           01   FB 00034             CALLS    #1, CLI$PRESENT                      ;
            52           50   D0 0003B             MOVL     R0, STATISTICS                       ;
                  FF48   C3   9F 0003E             PUSHAB   FILE_NAME                            ; 0582
                  0000'  CF   9F 00042             PUSHAB   P.AAP                                ;
 00000000G  00           02   FB 00046             CALLS    #2, CLI$GET_VALUE                    ;
                  FF64   C3   9F 0004D             PUSHAB   FLAGS                                ; 0586
                  FF50   C3   9F 00051             PUSHAB   STATISTICS_BLOCK                     ;
                  FF48   C3   9F 00055             PUSHAB   FILE_NAME                            ;
 00000000G  00           03   FB 00059             CALLS    #3, CONV$RECLAIM                     ;
            01           50   E8 00060             BLBS     STATUS, 1$                           ;
                              04 00063             RET                                           ;
            03           52   E8 00064 1$:         BLBS     STATISTICS, 2$                       ; 0590
                       00A2   31 00067             BRW      4$                                   ;
                    FC   A3   9F 0006A 2$:         PUSHAB   TIMER_BLK                            ; 0601
                         53   DD 0006D             PUSHL    R3                                   ;
                    40   A3   9F 0006F             PUSHAB   ONE                                  ;
            64           03   FB 00072             CALLS    #3, LIB$STAT_TIMER                   ;
                    FC   A3   9F 00075             PUSHAB   TIMER_BLK                            ; 0602
                    50   A3   9F 00078             PUSHAB   TEMP_TIME                            ;
                    44   A3   9F 0007B             PUSHAB   TWO                                  ;
            64           03   FB 0007E             CALLS    #3, LIB$STAT_TIMER                   ;
 63       48  A3         63   C3 00081             SUBL3    ELP_TIME, ZERO_Q, ELP_TIME           ; 0606
            50      4C   A3   D0 00086             MOVL     ZERO_Q+4, R0                         ;
            50      04   A3   D9 0008A             SBWC     ELP_TIME+4, R0                       ;
      04    A3           50   D0 0008E             MOVL     R0, ELP_TIME+4                       ;
                         7E   D4 00092             CLRL     -(SP)                                ; 0613
                         53   DD 00094             PUSHL    R3                                   ;
```

```
                                    30   A3   9F 00096          PUSHAB  ELP_DESC                                                ;
                                         7E   D4 00099          CLRL    -(SP)                                                   ;
                                    65        04 FB 0009B          CALLS   #4, SYS$ASCTIM                                          ;
                                    08   A3   9F 0009E          PUSHAB  CPU_TIME                                                ; 0620
                                    58   A3   9F 000A1          PUSHAB  MULTOOK                                                 ;
                                    50   A3   9F 000A4          PUSHAB  TEMP_TIME                                               ;
                               0000V CF        03 FB 000A7          CALLS   #3, MULQ                                                ;
                          08   A3    48   A3   08   A3   C3 000AC  SUBL3   CPU_TIME, ZERO_Q, CPU_TIME                           ; 0624
                                         50   4C   A3   D0 000B3  MOVL    ZERO_Q+4, R0                                            ;
                                         50   0C   A3   D9 000B7  SBWC    CPU_TIME+4, R0                                          ;
                                    0C   A3        50   D0 000BB  MOVL    R0, CPU_TIME+4                                          ;
                                         7E   D4 000BF          CLRL    -(SP)                                                   ; 0631
                                    08   A3   9F 000C1          PUSHAB  CPU_TIME                                                ;
                                    38   A3   9F 000C4          PUSHAB  CPU_DESC                                                ;
                                         7E   D4 000C7          CLRL    -(SP)                                                   ;
                                    65        04 FB 000C9          CALLS   #4, SYS$ASCTIM                                          ;
                                         52   D4 000CC          CLRL    I                                                       ; 0635
                               FF50 C342   DD 000CE 3$:       PUSHL   STATISTICS_BLOCK[I]                                     ; 0641
                                    EC   A3   9F 000D3          PUSHAB  FAO_DESC                                                ;
                                    08   AE   9F 000D6          PUSHAB  LENGTH                                                  ;
                               0000'CF42   DD 000D9          PUSHL   STATS[I]                                                ;
                                    66        04 FB 000DE          CALLS   #4, SYS$FAO                                             ;
                               F4   A3        6E   B0 000E1          MOVW    LENGTH, PUT_DESC                                        ; 0643
                                    F4   A3   9F 000E5          PUSHAB  PUT_DESC                                                ; 0647
                                    67        01 FB 000E8          CALLS   #1, LIB$PUT_OUTPUT                                      ;
                          DF             52   04   F3 000EB          AOBLEQ  #4, I, 3$                                               ;
                                    38   A3   9F 000EF          PUSHAB  CPU_DESC                                                ; 0653
                                    30   A3   9F 000F2          PUSHAB  ELP_DESC                                                ;
                                    EC   A3   9F 000F5          PUSHAB  FAO_DESC                                                ;
                                    0C   AE   9F 000F8          PUSHAB  LENGTH                                                  ;
                               0000'    CF   DD 000FB          PUSHL   STATS+20                                                ;
                                    66        05 FB 000FF          CALLS   #5, SYS$FAO                                             ;
                               F4   A3        6E   B0 00102          MOVW    LENGTH, PUT_DESC                                        ; 0654
                                    F4   A3   9F 00106          PUSHAB  PUT_DESC                                                ; 0655
                                    67        01 FB 00109          CALLS   #1, LIB$PUT_OUTPUT                                      ;
                                    50        01   D0 0010C 4$:       MOVL    #1, R0                                                  ; 0659
                                              04 0010F          RET                                                             ; 0661

; Routine Size:  272 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  272          0662  1 %SBTTL  'MULQ'
;  273          0663  1 ROUTINE MULQ ( MUL1 : REF VECTOR [ 2,LONG ],
;  274          0664  1                MUL2 : REF VECTOR [ 2,LONG ],
;  275          0665  1                PROD : REF VECTOR [ 2,LONG ] ) : NOVALUE =
;  276          0666  1
;  277          0667  1 !++
;  278          0668  1 !
;  279          0669  1 ! Functional Description:
;  280          0670  1 !
;  281          0671  1 !       Multiplies two quadwords.  This routine was converted from the example
;  282          0672  1 !       of the EMUL instruction in the VAX Architecture Handbook
;  283          0673  1 !
;  284          0674  1 ! Calling Sequence:
;  285          0675  1 !
;  286          0676  1 !       MULQ( mul1,mul2,prod )
;  287          0677  1 !
;  288          0678  1 ! Input Parameters:
;  289          0679  1 !
;  290          0680  1 !       mul1    - quadword multiplier
;  291          0681  1 !       mul2    - quadword multiplier
;  292          0682  1 !
;  293          0683  1 ! Implicit Inputs:
;  294          0684  1 !       none
;  295          0685  1 !
;  296          0686  1 ! Output Parameters:
;  297          0687  1 !
;  298          0688  1 !       prod    - quadword product (note: output cannot be same as either input)
;  299          0689  1 !
;  300          0690  1 ! Implicit Outputs:
;  301          0691  1 !       none
;  302          0692  1 !
;  303          0693  1 ! Routine Value:
;  304          0694  1 !       none
;  305          0695  1 !
;  306          0696  1 ! Routines Called:
;  307          0697  1 !       none
;  308          0698  1 !
;  309          0699  1 ! Side Effects:
;  310          0700  1 !       none
;  311          0701  1 !
;  312          0702  1 !--
;  313          0703  1
;  314          0704  2     BEGIN
;  315          0705  2
;  316          0706  2     BUILTIN
;  317          0707  2         EMUL;
;  318          0708  2
;  319          0709  2     BIND
;  320          0710  2         MUL1S = MUL1 [ 0 ] : SIGNED,
;  321          0711  2         MUL2S = MUL2 [ 0 ] : SIGNED;
;  322          0712  2
;  323          0713  2     LOCAL
;  324          0714  2         ZERO : INITIAL( 0 ),
;  325          0715  2         TEMP;
;  326          0716  2
;  327          0717  2     ! Multiply low half
;  328          0718  2     !
```

```
;  329          0719  2     EMUL( .MUL1,.MUL2,ZERO,.PROD );
;  330          0720  2
;  331          0721  2     ! High half = A[high] * B[low] + A[low] * B[high]
;  332          0722  2     !
;  333          0723  2     TEMP = ( .MUL1 [ 1 ] * .MUL2 [ 0 ] ) + ( .MUL1 [ 0 ] * .MUL2 [   ] );
;  334          0724  2
;  335          0725  2     ! If A[low]<0 then compensate of unsigned bias of 2**32
;  336          0726  2     !
;  337          0727  2     IF .MUL1S LSS 0
;  338          0728  2     THEN
;  339          0729  2         TEMP = .TEMP + .MUL2 [ 0 ];
;  340          0730  2
;  341          0731  2     ! If B[low]<0 then compensate of unsigned bias of 2**32
;  342          0732  2     !
;  343          0733  2     IF .MUL2S LSS 0
;  344          0734  2     THEN
;  345          0735  2         TEMP = .TEMP + .MUL1 [ 0 ];
;  346          0736  2
;  347          0737  2     ! Combine with high half of A[low] * B[low]
;  348          0738  2     !
;  349          0739  2     PROD [ 1 ] = .PROD [ 1 ] + .TEMP;
;  350          0740  2
;  351          0741  2     RETURN
;  352          0742  2
;  353          0743  1     END;


                                        001C 00000 MULQ:    .WORD   Save R2,R3,R4                   ; 0663
                                53      04  AC  D0 00002        MOVL    MUL1, R3                    ; 0710
                                52      08  AC  D0 00006        MOVL    MUL2, R2                    ; 0711
                                        51  D4  0000A           CLRL    ZERO                        ;
                                50      0C  AC  D0 0000C        MOVL    PROD, R0                    ; 0719
        60              51              62  63  7A 00010        EMUL    (R3), (R2), ZERO, (R0)      ;
                        54      04      A3  62  C5 00015        MULL3   (R2), 4(R3), R4             ; 0723
                        51      04      63  A2  C5 0001A        MULL3   4(R2), (R3), R1             ;
                                51      54  C0  0001F          ADDL2   R4, TEMP                    ;
                                        63  D5  00022          TSTL    (R3)                        ; 0727
                                        03  18  00024          BGEQ    1$                          ;
                                51      62  C0  00026          ADDL2   (R2), TEMP                  ; 0729
                                        62  D5  00029 1$:       TSTL    (R2)                        ; 0733
                                        03  18  0002B          BGEQ    2$                          ;
                                51      63  C0  0002D          ADDL2   (R3), TEMP                  ; 0735
                            04  A0          51  C0  00030 2$:   ADDL2   TEMP, 4(R0)                ; 0739
                                            04  00034          RET                                 ; 0743

; Routine Size:  53 bytes,    Routine Base:  $CODE$ + 0110

;  354          0744  1
;  355          0745  0 END     ELUDOM
```

```
;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $OWN$                          280  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                         308  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                         325  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;
;                                 Library Statistics
;                                          -------- Symbols --------      Pages       Processing
;       File                               Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1           18619         9         0        1000        00:01.8
;
;                                   COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:RECLDCL/OBJ=OBJ$:RECLDCL MSRC$:RECLDCL/UPDATE=(ENH$:RECLDCL)
;
; Size:         325 code + 588 data bytes
; Run Time:         00:10.9
; Elapsed Time:     00:38.6
; Lines/CPU Min:     4097
; Lexemes/CPU-Min: 21195
; Memory Used:  125 pages
; Compilation Complete
```

0066 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

RECLDCL
LIS

RECLREC
LIS

RECLCTRL
LIS

RECLRMSIO
LIS

CONVMSG
LIS

CONVMAIN
LIS

CONVSORT
LIS

CONVVEC
LIS