M 1

```
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     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     RRRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           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  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
**FILE**ID**BASFIND
```

K 3

```
BBBBBBBB      AAAAAA      SSSSSSSS  FFFFFFFFFF    IIIIII    NN      NN  DDDDDDDD
BBBBBBBB      AAAAAA      SSSSSSSS  FFFFFFFFFF    IIIIII    NN      NN  DDDDDDDD
BB      BB  AA      AA  SS          FF              II      NN      NN  DD      DD
BB      BB  AA      AA  SS          FF              II      NN      NN  DD      DD
BB      BB  AA      AA  SS          FF              II      NNNN    NN  DD      DD
BB      BB  AA      AA  SS          FF              II      NNNN    NN  DD      DD
BBBBBBBB    AA      AA    SSSSSS    FFFFFFFF        II      NN  NN  NN  DD      DD
BBBBBBBB    AA      AA    SSSSSS    FFFFFFFF        II      NN  NN  NN  DD      DD
BB      BB  AAAAAAAAAA          SS  FF              II      NN    NNNN  DD      DD
BB      BB  AAAAAAAAAA          SS  FF              II      NN    NNNN  DD      DD
BB      BB  AA      AA          SS  FF              II      NN      NN  DD      DD        ....
BB      BB  AA      AA          SS  FF              II      NN      NN  DD      DD        ....
BBBBBBBB    AA      AA  SSSSSSSS    FF            IIIIII    NN      NN  DDDDDDDD          ....
BBBBBBBB    AA      AA  SSSSSSSS    FF            IIIIII    NN      NN  DDDDDDDD          ....

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 BAS$FIND (                            ! Basic FIND construct
0               IDENT = '1-009'          ! File: BASFIND.B32 Edit: MDL1009
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 !++
1 ! FACILITY:
1 !       Basic support library - user callable
1 !
1 ! ABSTRACT:
1 !
1 !       This module is the UPI level of the Basic FIND construct.  Initially,
1 !       it contains only the code for sequential I/O.  This module will set
1 !       up the I/O data base for the LUN and go directly to the REC level.
1 !
1 !
1 ! ENVIRONMENT:
1 !       User access mode - AST reentrant.
1 !
1 ! AUTHOR: Donald G. Petersen, CREATION DATE: 27-Feb-79
1 !
1 ! MODIFIED BY:
1 !
1 !       DGP, 27-Feb-79 : VERSION 01
1 ! 1-001 - original.  DGP 27-Feb-79
1 ! 1-002 - Add BAS$FIND_RECORD.  DGP 02-Mar-79
1 ! 1-003 - More work on relative I/O.  DGP 05-Mar-79
1 ! 1-004 - Add BAS$FIND_KEY.  DGP 06-Apr-79
1 ! 1-005 - Set up ISB$A_USER_FP.  JBS 25-JUL-1979
1 ! 1-006 - Check for virtual use of this file; set block use.  DGP 16-Oct-79
1 ! 1-007 - Add support for RFA access and manual record locking.  PLL 1-Jun-82
1 ! 1-008 - RFAs are passed by ref.  PLL 4-Jun-1982
1 ! 1-009 - allow REGARDLESS without manual record locking.  MDL 14-Feb-1984
```

```
;    58          0058  1 !--
;    59          0059  1
;    60          0060  1 !<BLF/PAGE>
;    62          0061  1 !
;    63          0062  1 ! SWITCHES:
;    64          0063  1 !
;    65          0064  1
;    66          0065  1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;    67          0066  1
;    68          0067  1 !
;    69          0068  1 ! LINKAGES
;    70          0069  1 !
;    71          0070  1
;    72          0071  1 REQUIRE 'RTLIN:OTSLNK';                          ! Define all linkages
;    73          0500  1
;    74          0501  1 !
;    75          0502  1 ! TABLE OF CONTENTS:
;    76          0503  1 !
;    77          0504  1
;    78          0505  1 FORWARD ROUTINE
;    79          0506  1     BAS$FIND_KEY : NOVALUE,                     ! UPI level Indexed FIND
;    80          0507  1     BAS$FIND_RECORD : NOVALUE,                  ! UPI level Relative FIND
;    81          0508  1     BAS$FIND : NOVALUE,                         ! UPI level Sequential FIND
;    82          0509  1     BAS$FIND_RFA : NOVALUE;                     ! UPI level RFA FIND
;    83          0510  1
;    84          0511  1 !
;    85          0512  1 ! INCLUDE FILES:
;    86          0513  1 !
;    87          0514  1
;    88          0515  1 REQUIRE 'RTLML:OTSISB';                          ! ISB definitions
;    89          0683  1
;    90          0684  1 REQUIRE 'RTLML:OTSLUB';                          ! LUB definitions
;    91          0824  1
;    92          0825  1 REQUIRE 'RTLIN:RTLPSECT';                        ! Define DECLARE_PSECTS macro
;    93          0920  1
;    94          0921  1 LIBRARY 'RTLSTARLE';                             ! Starlet system macros
;    95          0922  1
;    96          0923  1 !
;    97          0924  1 ! MACROS:
;    98          0925  1 !
;    99          0926  1 !       NONE
;   100          0927  1 !
;   101          0928  1 !
;   102          0929  1 ! EQUATED SYMBOLS:
;   103          0930  1 !       NONE
;   104          0931  1 !
;   105          0932  1 !
;   106          0933  1 ! PSECT DECLARATIONS:
;   107          0934  1 !
;   108          0935  1 DECLARE_PSECTS (BAS);
;   109          0936  1 !
;   110          0937  1 ! OWN STORAGE:
;   111          0938  1 !
;   112          0939  1 !       NONE
;   113          0940  1 !
;   114          0941  1 !
;   115          0942  1 ! EXTERNAL REFERENCES:
;   116          0943  1 !
;   117          0944  1
;   118          0945  1 EXTERNAL ROUTINE
;  119        0946  1       BAS$$STOP_IO : NOVALUE,                     ! Signal fatal BASIC I/O error
;  120        0947  1       BAS$$REC_FIN : JSB_REC_IND1 NOVALUE,        ! REC level - FIND indexed
;  121        0948  1       BAS$$REC_FRE : JSB_REC2 NOVALUE,            ! REC level - RMS interface FIND relative
;  122        0949  1       BAS$$REC_FSE : JSB_REC2 NOVALUE,            ! REC level processing - RMS interface
;  123        0950  1                                                   ! FIND sequential
;  124        0951  1       BAS$$REC_FRFA : JSB_REC2 NOVALUE,           ! REC level - FIND by RFA
;  125        0952  1       BAS$$CB_PUSH : JSB_CB_PUSH NOVALUE,         ! Load register CCB
;  126        0953  1       BAS$$CB_POP : JSB_CB_POP NOVALUE;           ! Done with register CCB
;  127        0954  1
;  128        0955  1   !+
;  129        0956  1   ! The following are the error codes used in this module.
;  130        0957  1   !-
;  131        0958  1
;  132        0959  1   EXTERNAL LITERAL
;  133        0960  1       BAS$K_ILLILLACC : UNSIGNED (8),             ! Illegal or illogical access
;  134        0961  1       BAS$K_IO_CHANOT : UNSIGNED (8),             ! I/O channel not open
;  135        0962  1       BAS$K_ILLRECLOC : UNSIGNED (8);             ! Illegal record locking clause
;  136        0963  1
```

```
1 GLOBAL ROUTINE BAS$FIND (                              ! FIND sequential
1         UNIT,                                          ! logical unit number
1         LOCK_FLAGS                                     ! manual locking bits
1     ) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine will set up the I/O data base for this LUN if necessary
1 !       and then go directly to the REC level.  When control is returned to
1 !       this routine, it pops the CCB off of the I/O system.  The actual inter-
1 !       face to RMS is done at the REC level.  The next record is located.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       UNIT.rlu.v          logical unit number
1 !       [LOCK_FLAGS.rlu.v] if present, bits to pass on to the record level
1 !                          to control manual record locking
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       LUB$V_VA_USE             virtual use of this file
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       ISB$B_STTM_TYPE          the statement type
1 !       LUB$V_BLK_USE            non-virtual array use of this file
1 !
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Signals:
1 !       BAS$K_IO_CHANOT (I/O channel not open)
1 !       BAS$K_ILLILLACC (illegal or illogical access)
1 !
1 !--
1
2     BEGIN
2
2     BUILTIN
2         FP,
2         ACTUALCOUNT;
2
2     GLOBAL REGISTER
2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
2
2     LOCAL
2         FMP : REF BLOCK [, BYTE],
2         FLAGS;
2
2     LITERAL
2         K_LOCK_ARG = 2;
2
2     FMP = .FP;
```

```
2 !+
2 ! Allocate the LUB/ISB/RAB for this unit if necessary.  Store new CB (con-
2 ! trol block) in OTS$$A_CUR_LUB.  Store signed unit number in LUB$W_LUN.
2 !-
2     BAS$$CB_PUSH (.UNIT, LUB$K_ILUN_MIN);
2     CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
2 !+
2 ! If the channel is not open, give an error.
2 ! FIND is not permitted on channel 0.
2 !-
2
2     IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP_IO (BAS$K_IO_CHANOT);
2
2 !+
2 ! Now that the data base is in place, store the statement type and  go
2 ! directly to the REC level.
2 !-
2     CCB [ISB$B_STTM_TYPE] = ISB$K_ST_TY_FSE;
2 !+
2 ! Check for virtual array usage and set block usage
2 !-
2     IF .CCB [LUB$V_VA_USE] THEN BAS$$STOP_IO(BAS$K_ILLILLACC);
2     CCB [LUB$V_BLK_USE] = 1;
2
2     IF ACTUALCOUNT () LSS K_LOCK_ARG
2     THEN
2         FLAGS = 0
2     ELSE
3         BEGIN
3         !+
3         ! The ULK bit must set unless this is a REGARDLESS clause.
3         !-
3         CASE .CCB [RAB$V_ULK] FROM 0 TO 1 OF
3         SET
3             [0]:
3             IF (.LOCK_FLAGS AND RAB$M_RRL) NEQ 0
3             THEN
3                 FLAGS = .LOCK_FLAGS
3             ELSE
3                 BAS$$STOP_IO (BAS$K_ILLRECLOC);
3
3             [1]:
3             FLAGS = .LOCK_FLAGS;
3         TES;
2         END;
2
2     BAS$$REC_FSE (.FLAGS);
2 !+
2 ! Now that the FIND has been done, pop the CCB off the I/O system.
2 !-
2     BAS$$CB_POP ();
1     END;                                        !End of BAS$FIND

                                                    .TITLE  BAS$FIND
                                                    .IDENT  \1-009\
                                                                      .EXTRN  BAS$$STOP_IO, BAS$$REC_FIN
                                                                      .EXTRN  BAS$$REC_FRE, BAS$$REC_FSE
                                                                      .EXTRN  BAS$$REC_FRFA, BAS$$CB_PUSH
                                                                      .EXTRN  BAS$$CB_POP, BAS$K_ILLILLACC
                                                                      .EXTRN  BAS$K_IO_CHANOT
                                                                      .EXTRN  BAS$K_ILLRECLOC

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

                                                083C 00000            .ENTRY  BAS$FIND, Save R2,R3,R4,R5,R11            ; 0964
                           54 00000000G  00   9E 00002            MOVAB   BAS$$STOP_IO, R4                          ;
                           53            5D   D0 00009            MOVL    FP, FMP                                   ; 1020
                           50            08   CE 0000C            MNEGL   #8, R0                                    ; 1025
                           52       04   AC   D0 0000F            MOVL    UNIT, R2                                  ;
                              00000000G  00   16 00013            JSB     BAS$$CB_PUSH                              ;
                    FF4C   CB       0C   A3   D0 00019            MOVL    12(FMP), -180(CCB)                        ; 1026
                           07       FC   AB   E8 0001F            BLBS    -4(CCB), 1$                               ; 1032
                           7E      00G   8F   9A 00023            MOVZBL  #BAS$K_IO_CHANOT, -(SP)                   ;
                           64            01   FB 00027            CALLS   #1, BAS$$STOP_IO                          ;
                    FF71   CB            33   90 0002A  1$:       MOVB    #51, -143(CCB)                            ; 1038
                           07       FF   AB   E9 0002F            BLBC    -1(CCB), 2$                               ; 1042
                           7E      00G   8F   9A 00033            MOVZBL  #BAS$K_ILLILLACC, -(SP)                   ;
                           64            01   FB 00037            CALLS   #1, BAS$$STOP_IO                          ;
                      FF   AB            02   88 0003A  2$:       BISB2   #2, -1(CCB)                               ; 1043
                           02            6C   91 0003E            CMPB    (AP), #2                                  ; 1045
                                         04   1E 00041            BGEQU   3$                                        ;
                                         52   D4 00043            CLRL    FLAGS                                     ; 1047
                                         20   11 00045            BRB     7$                                        ;
     53       06      AB           01    02   EF 00047  3$:       EXTZV   #2, #1, 6(CCB), R3                        ; 1053
                      01           00    53   CF 0004D            CASEL   R3, #0, #1                                ;
                         0012          0004      00051  4$:       .WORD   5$-4$,-                                   ;
                                                                          6$-4$
                09      08   AC            03   E0 00055  5$:       BBS     #3, LOCK_FLAGS, 6$                        ; 1056
                           7E      00G   8F   9A 0005A            MOVZBL  #BAS$K_ILLRECLOC, -(SP)                   ; 1060
                           64            01   FB 0005E            CALLS   #1, BAS$$STOP_IO                          ;
                                         04   11 00061            BRB     7$                                        ; 1056
                           52       08   AC   D0 00063  6$:       MOVL    LOCK_FLAGS, FLAGS                         ; 1063
                           50            52   D0 00067  7$:       MOVL    FLAGS, R0                                 ; 1067
                              00000000G  00   16 0006A            JSB     BAS$$REC_FSE                              ;
                              00000000G  00   16 00070            JSB     BAS$$CB_POP                               ; 1071
                                            04 00076            RET                                               ; 1072

; Routine Size:  119 bytes,    Routine Base:  _BAS$CODE + 0000


;  247          1073  1
```

```
1 GLOBAL ROUTINE BAS$FIND_KEY (                         ! FIND indexed
1         UNIT,                                         ! logical unit number
1         KEY_NO,                                       ! key of reference number
1         REL_OP,                                       ! relational operator
1         KEY,                                          ! the key
1         LOCK_FLAGS                                    ! manual locking flags
1    ) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine will set up the I/O data base for this LUN if necessary
1 !       and then go directly to the REC level.  When control is returned to
1 !       this routine, it pops the CCB off of the I/O system.  The actual inter-
1 !       face to RMS is done at the REC level.  The next record is located
1 !       based on the key of reference specified.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       UNIT.rlu.v      logical unit number
1 !       KEY_NO.rl.v     key of reference number
1 !       REL_OP.rl.v     relational operator
1 !       KEY.rt.dx       the key desired
1 !       [LOCK_FLAGS.rlu.v]  if present, specifies bits to pass on to record level
1 !                           to control manual record locking
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       LUB$V_VA_USE            virtual array use of this file
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       ISB$B_STTM_TYPE         the statement type
1 !       LUB$V_BLK_USE           non-virtual use of this file
1 !
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Signals:
1 !       BAS$K_IO_CHANOT (I/O channel not open)
1 !       BAS$K_ILLILLACC (Illegal or illogical access)
1 !
1 !--
1
2    BEGIN
2
2    BUILTIN
2        FP,
2        ACTUALCOUNT;
2
2    GLOBAL REGISTER
2        CCB = K_CCB_REG : REF BLOCK [, BYTE];
2
2    LOCAL
```

```
:  306          1131  2            FMP : REF BLOCK [, BYTE],
:  307          1132  2            FLAGS;
:  308          1133  2
:  309          1134  2        LITERAL
:  310          1135  2            K_LOCK_ARG = 5;
:  311          1136  2
:  312          1137  2        FMP = .FP;
:  313          1138  2    !+
:  314          1139  2    ! Allocate the LUB/ISB/RAB for this unit if necessary.  Store new CB (con-
:  315          1140  2    ! trol block) in OTS$$A_CUR_LUB.  Store signed unit number in LUB$W_LUN.
:  316          1141  2    !-
:  317          1142  2        BAS$$CB_PUSH (.UNIT, LUB$K_ILUN_MIN);
:  318          1143  2        CCB [ISB$A_USER_FP] = .FMP [SF$C_SAVE_FP];
:  319          1144  2    !+
:  320          1145  2    ! If the channel is not open, give an error.
:  321          1146  2    !-
:  322          1147  2
:  323          1148  2        IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP_IO (BAS$K_IO_CHANOT);
:  324          1149  2
:  325          1150  2    !+
:  326          1151  2    ! Now that the data base is in place, store the statement type and  go
:  327          1152  2    ! directly to the REC level.
:  328          1153  2    !-
:  329          1154  2        CCB [ISB$B_STTM_TYPE] = ISB$K_ST_TY_FIN;
:  330          1155  2    !+
:  331          1156  2    ! Check for virtual array usage and set block usage
:  332          1157  2    !-
:  333          1158  2        IF .CCB [LUB$V_VA_USE] THEN BAS$$STOP_IO(BAS$K_ILLILLACC);
:  334          1159  2        CCB [LUB$V_BLK_USE] = 1;
:  335          1160  2
:  336          1161  2        IF ACTUALCOUNT () LSS K_LOCK_ARG
:  337          1162  2        THEN
:  338          1163  2            FLAGS = 0
:  339          1164  2        ELSE
:  340          1165  3            BEGIN
:  341          1166  3            !+
:  342          1167  3            ! The ULK bit must set unless this is a REGARDLESS clause.
:  343          1168  3            !-
:  344          1169  3            CASE .CCB [RAB$V_ULK] FROM 0 TO 1 OF
:  345          1170  3            SET
:  346          1171  3                [0]:
:  347          1172  3                IF (.LOCK_FLAGS AND RAB$M_RRL) NEQ 0
:  348          1173  3                THEN
:  349          1174  3                    FLAGS = .LOCK_FLAGS
:  350          1175  3                ELSE
:  351          1176  3                    BAS$$STOP_IO (BAS$K_ILLRECLOC);
:  352          1177  3
:  353          1178  3                [1]:
:  354          1179  3                FLAGS = .LOCK_FLAGS;
:  355          1180  3            TES;
:  356          1181  2            END;
:  357          1182  2
:  358          1183  2        BAS$$REC_FIN (.KEY_NO, .REL_OP, .KEY, .FLAGS);
:  359          1184  2    !+
:  360          1185  2    ! Now that the FIND has been done, pop the CCB off the I/O system.
:  361          1186  2    !-
:  362          1187  2        BAS$$CB_POP ();
```

```
;  363           1188  1       END;                                                     !End of BASSFIND_KEY



                                                       083C 00000           .ENTRY   BAS$FIND_KEY, Save R2,R3,R4,R5,R11                 ; 1074
                                  54  00000000G   00     9E 00002           MOVAB    BAS$$STOP_IO, R4                                   ;
                                  53              5D     D0 00009           MOVL     FP, FMP                                            ; 1137
                                  50              08     CE 0000C           MNEGL    #8, R0                                             ; 1142
                                  52        04    AC     D0 0000F           MOVL     UNIT, R2                                           ;
                                      00000000G   00     16 00013           JSB      BAS$$CB_PUSH                                       ;
                         FF4C     CB        0C    A3     D0 00019           MOVL     12(FMP), -180(CCB)                                 ; 1143
                                  07        FC    AB     E8 0001F           BLBS     -4(CCB), 1$                                        ; 1148
                                  7E       00G    8F     9A 00023           MOVZBL   #BAS$K_IO_CHANOT, -(SP)                            ;
                                  64              01     FB 00027           CALLS    #1, BAS$$STOP_IO                                   ;
                         FF71     CB              2F     90 0002A 1$:       MOVB     #47, -143(CCB)                                     ; 1154
                                  07        FF    AB     E9 0002F           BLBC     -1(CCB), 2$                                        ; 1158
                                  7E       00G    8F     9A 00033           MOVZBL   #BAS$K_ILLILLACC, -(SP)                            ;
                                  64              01     FB 00037           CALLS    #1, BAS$$STOP_IO                                   ;
                           FF     AB              02     88 0003A 2$:       BISB2    #2, -1(CCB)                                        ; 1159
                                  05              6C     91 0003E           CMPB     (AP), #5                                           ; 1161
                                                  04     1E 00041           BGEQU    3$                                                 ;
                                                  53     D4 00043           CLRL     FLAGS                                              ; 1163
                                                  20     11 00045           BRB      7$                                                 ;
          52        06     AB                     01     02 EF 00047 3$:       EXTZV    #2, #1, 6(CCB), R2                                 ; 1169
                           01                     00     52 CF 0004D           CASEL    R2, #0, #1                                         ;
                                0012              0004      00051 4$:       .WORD    5$-4$,-                                            ;
                                                                                     6$-4$                                              ;
                           09        14    AC              03 E0 00055 5$:       BBS      #3, LOCK_FLAGS, 6$                                 ; 1172
                                  7E       00G    8F     9A 0005A           MOVZBL   #BAS$K_ILLRECLOC, -(SP)                            ; 1176
                                  64              01     FB 0005E           CALLS    #1, BAS$$STOP_IO                                   ;
                                                  04     11 00061           BRB      7$                                                 ; 1172
                                  53        14    AC     D0 00063 6$:       MOVL     LOCK_FLAGS, FLAGS                                  ; 1179
                                  51        0C    AC     7D 00067 7$:       MOVQ     REL_OP, R1                                         ; 1183
                                  50        08    AC     D0 0006B           MOVL     KEY_NO, R0                                         ;
                                      00000000G   00     16 0006F           JSB      BAS$$REC_FIN                                       ;
                                      00000000G   00     16 00075           JSB      BAS$$CB_POP                                        ; 1187
                                                         04 0007B           RET                                                         ; 1188

; Routine Size:  124 bytes,    Routine Base:  _BAS$CODE + 0077


;  364           1189  1
```

```
1 GLOBAL ROUTINE BAS$FIND_RECORD (                      ! FIND relative
1         UNIT,                                         ! logical unit number
1         RECORD_NUM,                                   ! relative record number
1         LOCK_FLAGS                                    ! manual locking flags
1     ) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine will set up the I/O data base for this LUN if necessary
1 !       and then go directly to the REC level.  When control is returned to
1 !       this routine, it pops the CCB off of the I/O system.  The actual inter-
1 !       face to RMS is done at the REC level.  The next record is located.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       UNIT.rlu.v              logical unit number
1 !       RECORD_NUM.rl.v         relative record number
1 !       [LOCK_FLAGS.rlu.v]      if present, specifies bits to pass on to record
1 !                               level to control manual record locking
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       LUB$V_VA_USE            virtual array use of this file
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       ISB$B_STTM_TYPE         the statement type
1 !       LUB$V_BLK_USE           non-virtual array use of this file
1 !
1 ! COMPLETION CODES:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Signals:
1 !       BAS$K_IO_CHANOT (I/O channel not open)
1 !       BAS$K_ILLILLACC (Illegal or illogical access)
1 !
1 !--
1
2     BEGIN
2
2     BUILTIN
2         FP,
2         ACTUALCOUNT;
2
2     GLOBAL REGISTER
2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
2
2     LOCAL
2         FMP : REF BLOCK [, BYTE],
2         FLAGS;
2
2     LITERAL
2         K_LOCK_ARG = 3;
```

```
;  423        1247  2
;  424        1248  2      FMP = .FP;
;  425        1249  2  !+
;  426        1250  2  ! Allocate the LUB/ISB/RAB for this unit if necessary.  Store new CB (con-
;  427        1251  2  ! trol block) in OTS$$A_CUR_LUB.  Store signed unit number in LUB$W_LUN.
;  428        1252  2  !-
;  429        1253  2      BAS$$CB_PUSH (.UNIT, LUB$K_ILUN_MIN);
;  430        1254  2      CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  431        1255  2  !+
;  432        1256  2  ! Give an error if the channel is not open.
;  433        1257  2  !-
;  434        1258  2
;  435        1259  2      IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP_IO (BAS$K_IO_CHANOT);
;  436        1260  2
;  437        1261  2  !+
;  438        1262  2  ! Now that the data base is in place, store the statement type, store the index, and  go
;  439        1263  2  ! directly to the REC level.
;  440        1264  2  !-
;  441        1265  2      CCB [LUB$L_LOG_RECNO] = .RECORD_NUM;
;  442        1266  2      CCB [ISB$B_STTM_TYPE] = ISB$K_ST_TY_FRE;
;  443        1267  2  !+
;  444        1268  2  ! Check for virtual array usage and set block usage
;  445        1269  2  !-
;  446        1270  2      IF .CCB [LUB$V_VA_USE] THEN BAS$$STOP_IO(BAS$K_ILLILLACC);
;  447        1271  2      CCB [LUB$V_BLK_USE] = 1;
;  448        1272  2
;  449        1273  2      IF ACTUALCOUNT () LSS K_LOCK_ARG
;  450        1274  2      THEN
;  451        1275  2          FLAGS = 0
;  452        1276  2      ELSE
;  453        1277  3          BEGIN
;  454        1278  3          IF (.CCB [RAB$L_ROP] AND RAB$M_ULK) EQL 0
;  455        1279  3          THEN
;  456        1280  3              BAS$$STOP_IO (BAS$K_ILLRECLOC)
;  457        1281  3          ELSE
;  458        1282  3              FLAGS = .LOCK_FLAGS;
;  459        1283  2          END;
;  460        1284  2      BAS$$REC_FRE (.FLAGS);
;  461        1285  2  !+
;  462        1286  2  ! Now that the FIND has been done, pop the CCB off the I/O system.
;  463        1287  2  !-
;  464        1288  2      BAS$$CB_POP ();
;  465        1289  1      END;                                        !End of BAS$FIND_RECORD




                                083C 00000          .ENTRY  BAS$FIND_RECORD, Save R2,R3,R4,R5,R11       ; 1190
                 54 00000000G   00   9E 00002          MOVAB   BAS$$STOP_IO, R4                            ;
                 53             5D   D0 00009          MOVL    FP, FMP                                     ; 1248
                 50             08   CE 0000C          MNEGL   #8, R0                                      ; 1253
                 52        04   AC   D0 0000F          MOVL    UNIT, R2                                    ;
                    00000000G   00   16 00013          JSB     BAS$$CB_PUSH                                ;
            FF4C CB        0C   A3   D0 00019          MOVL    12(FMP), -180(CCB)                          ; 1254
                 07        FC   AB   E8 0001F          BLBS    -4(CCB), 1$                                 ; 1259
                 7E       00G   8F   9A 00023          MOVZBL  #BAS$K_IO_CHANOT, -(SP)                     ;
```

```
                          64                01  FB 00027         CALLS   #1, BAS$$STOP_IO                               ;
                  E0      AB          08    AC  D0 0002A 1$:     MOVL    RECORD_NUM, -32(CCB)                           ; 1265
                FF71      CB                29  90 0002F         MOVB    #41, -143(CCB)                                 ; 1266
                          07          FF    AB  E9 00034         BLBC    -1(CCB), 2$                                    ; 1270
                          7E          00G   8F  9A 00038         MOVZBL  #BAS$K_ILLILLACC, -(SP)                        ;
                          64                01  FB 0003C         CALLS   #1, BAS$$STOP_IO                               ;
                  FF      AB                02  88 0003F 2$:     BISB2   #2, -1(CCB)                                    ; 1271
                          03                6C  91 00043         CMPB    (AP), #3                                       ; 1273
                                            04  1E 00046         BGEQU   3$                                             ;
                                            52  D4 00048         CLRL    FLAGS                                          ; 1275
                                            11  11 0004A         BRB     5$                                             ;
         09               6B                32  E0 0004C 3$:     BBS     #50, (CCB), 4$                                 ; 1278
                          7E          00G   8F  9A 00050         MOVZBL  #BAS$K_ILLRECLOC, -(SP)                        ; 1280
                          64                01  FB 00054         CALLS   #1, BAS$$STOP_IO                               ;
                                            04  11 00057         BRB     5$                                             ;
                          52          0C    AC  D0 00059 4$:     MOVL    LOCK_FLAGS, FLAGS                              ; 1282
                          50                52  D0 0005D 5$:     MOVL    FLAGS, R0                                      ; 1284
                               00000000G    00  16 00060         JSB     BAS$$REC_FRE                                   ;
                               00000000G    00  16 00066         JSB     BAS$$CB_POP                                    ; 1288
                                                04 0006C         RET                                                    ; 1289

; Routine Size:  109 bytes,    Routine Base:  _BAS$CODE + 00F3
```

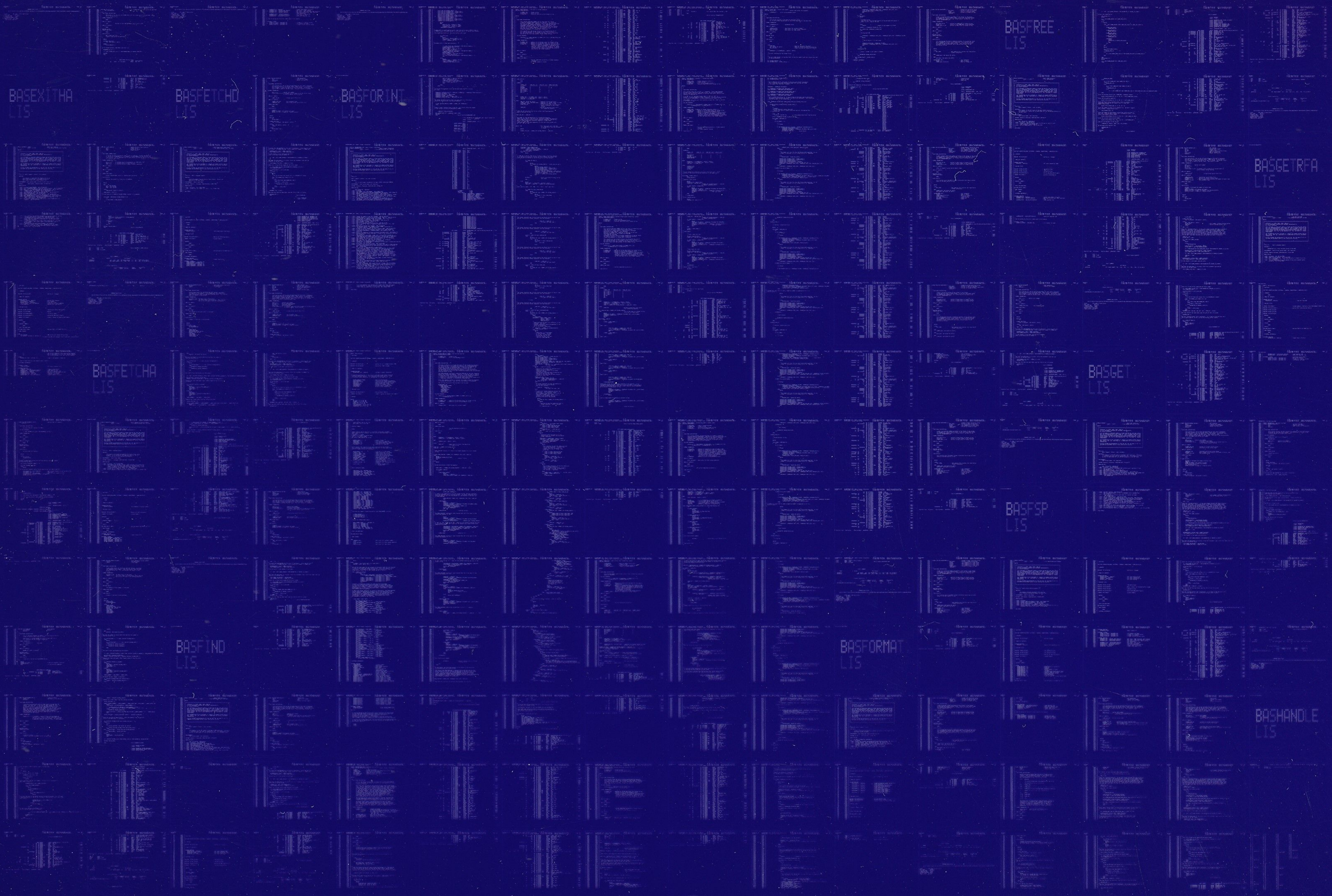
```
GLOBAL ROUTINE BAS$FIND_RFA (                           ! FIND by RFA
        UNIT,                                           ! logical unit number
        RFA,                                            ! RFA
        LOCK_FLAGS                                      ! manual locking flags
    ) : NOVALUE =

!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine will set up the I/O data base for this LUN if necessary
!       and then go directly to the REC level.  When control is returned to
!       this routine, it pops the CCB off of the I/O system.  The actual inter-
!       face to RMS is done at the REC level.  The record specified by the
!       RFA is located.
!
! FORMAL PARAMETERS:
!
!       UNIT.rlu.v              logical unit number
!       RFA.rx.r                RFA address
!       [LOCK_FLAGS.rlu.v]      if present, specifies bits to pass on to record
!                               level to control manual record locking
!
! IMPLICIT INPUTS:
!
!       LUB$V_VA_USE            virtual array use of this file
!
! IMPLICIT OUTPUTS:
!
!       ISB$B_STTM_TYPE         the statement type
!       LUB$V_BLK_USE           non-virtual array use of this file
!
! COMPLETION CODES:
!
!       NONE
!
! SIDE EFFECTS:
!
!       Signals:
!       BAS$K_IO_CHANOT (I/O channel not open)
!       BAS$K_ILLILLACC (Illegal or illogical access)
!
!--

    BEGIN

    BUILTIN
        FP,
        ACTUALCOUNT;

    GLOBAL REGISTER
        CCB = K_CCB_REG : REF BLOCK [, BYTE];

    LOCAL
        FMP : REF BLOCK [, BYTE],
        FLAGS;

    LITERAL
```

```
;  524        1347  2            K_LOCK_ARG = 3;
;  525        1348  2
;  526        1349  2        FMP = .FP;
;  527        1350  2    !+
;  528        1351  2    ! Allocate the LUB/ISB/RAB for this unit if necessary.  Store new CB (con-
;  529        1352  2    ! trol block) in OTS$$A_CUR_LUB.  Store signed unit number in LUB$W_LUN.
;  530        1353  2    !-
;  531        1354  2        BAS$$CB_PUSH (.UNIT, LUB$K_ILUN_MIN);
;  532        1355  2        CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  533        1356  2    !+
;  534        1357  2    ! Give an error if the channel is not open.
;  535        1358  2    !-
;  536        1359  2
;  537        1360  2        IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP_IO (BAS$K_IO_CHANOT);
;  538        1361  2
;  539        1362  2    !+
;  540        1363  2    ! Now that the data base is in place, store the statement type, store the RFA, and  go
;  541        1364  2    ! directly to the REC level.
;  542        1365  2    !-
;  543        1366  2        CH$MOVE (6, .RFA, CCB [RAB$W_RFA]);
;  544        1367  2        CCB [ISB$B_STTM_TYPE] = ISB$K_ST_TY_FRFA;
;  545        1368  2    !+
;  546        1369  2    ! Check for virtual array usage and set block usage
;  547        1370  2    !-
;  548        1371  2        IF .CCB [LUB$V_VA_USE] THEN BAS$$STOP_IO(BAS$K_ILLILLACC);
;  549        1372  2        CCB [LUB$V_BLK_USE] = 1;
;  550        1373  2
;  551        1374  2        IF ACTUALCOUNT () LSS K_LOCK_ARG
;  552        1375  2        THEN
;  553        1376  2            FLAGS = 0
;  554        1377  2        ELSE
;  555        1378  3            BEGIN
;  556        1379  3            IF (.CCB [RAB$L_ROP] AND RAB$M_ULK) EQL C
;  557        1380  3            THEN
;  558        1381  3                BAS$$STOP_IO (BAS$K_ILLRECLOC)
;  559        1382  3            ELSE
;  560        1383  3                FLAGS = .LOCK_FLAGS;
;  561        1384  2            END;
;  562        1385  2        BAS$$REC_FRFA (.FLAGS);
;  563        1386  2    !+
;  564        1387  2    ! Now that the FIND has been done, pop the CCB off the I/O system.
;  565        1388  2    !-
;  566        1389  2        BAS$$CB_POP ();
;  567        1390  1        END;                                          !End of BAS$FIND_RFA



                                        087C 00000          .ENTRY  BAS$FIND_RFA, Save R2,R3,R4,R5,R6,R11       ; 1290
                  56 00000000G   00  9E 00002          MOVAB   BAS$$STOP_IO, R6                             ;
                  53             5D  D0 00009          MOVL    FP, FMP                                      ; 1349
                  50             08  CE 0000C          MNEGL   #8, R0                                       ; 1354
                  52        04   AC  D0 0000F          MOVL    UNIT, R2                                     ;
                     00000000G   00  16 00013          JSB     BAS$$CB_PUSH                                 ;
           FF4C   CB        0C   A3  D0 00019          MOVL    12(FMP), -180(CCB)                           ; 1355
                  07        FC   AB  E8 0001F          BLBS    -4(CCB), 1$                                  ; 1360
```

```
                            7E        00G  8F   9A 00023         MOVZBL  #BAS$K_IO_CHANOT, -(SP)        ;
                            66             01   FB 00027         CALLS   #1, BAS$$STOP_IO               ;
          10    AB      08  BC             06   28 0002A 1$:     MOVC3   #6, @RFA, 16(CCB)              ; 1366
                      FF71  CB             38   90 00030         MOVB    #56, -143(CCB)                 ; 1367
                            07        FF   AB   E9 00035         BLBC    -1(CCB), 2$                    ; 1371
                            7E        00G  8F   9A 00039         MOVZBL  #BAS$K_ILLILLACC, -(SP)        ;
                            66             01   FB 0003D         CALLS   #1, BAS$$STOP_IO               ;
                        FF  AB             02   88 00040 2$:     BISB2   #2, -1(CCB)                    ; 1372
                            03             6C   91 00044         CMPB    (AP), #3                       ; 1374
                                           04   1E 00047         BGEQU   3$                             ;
                                           52   D4 00049         CLRL    FLAGS                          ; 1376
                                           11   11 0004B         BRB     5$                             ;
                09          6B             32   E0 0004D 3$:     BBS     #50, (CCB), 4$                 ; 1379
                            7E        00G  8F   9A 00051         MOVZBL  #BAS$K_ILLRECLOC, -(SP)        ; 1381
                            66             01   FB 00055         CALLS   #1, BAS$$STOP_IO               ;
                                           04   11 00058         BRB     5$                             ;
                            52        0C   AC   D0 0005A 4$:     MOVL    LOCK_FLAGS, FLAGS              ; 1383
                            50             52   D0 0005E 5$:     MOVL    FLAGS, R0                      ; 1385
                              00000000G    00   16 00061         JSB     BAS$$REC_FRFA                  ;
                              00000000G    00   16 00067         JSB     BAS$$CB_POP                    ; 1389
                                                04 0006D         RET                                    ; 1390

; Routine Size:  110 bytes,    Routine Base:  _BAS$CODE + 0160


;  568          1391  1
;  569          1392  1
;  570          1393  1 END                                           !End of module - BAS$FIND
;  571          1394  1
;  572          1395  0 ELUDOM




;                                PSECT SUMMARY
;
;       Name                      Bytes                    Attributes
;
;    _BAS$CODE                      462  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)



;                           Library Statistics
;
;                                         -------- Symbols --------      Pages       Processing
;       File                              Total    Loaded   Percent      Mapped      Time
;
;    _$255$DUA28:[SYSLIB]STARLET.L32;1    9776        6         0        581         00:01.1
;                                       COMMAND QUALIFIERS

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

; Size:          462 code + 0 data bytes
; Run Time:         00:14.7
; Elapsed Time:     00:32.0
; Lines/CPU Min:     5709
; Lexemes/CPU-Min: 32476
; Memory Used:  129 pages
; Compilation Complete
```

0023 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY


BASFREE
LIS

BASEXITHA
LIS

BASFETCHD
LIS

BASFORINT
LIS

BASGETRFA
LIS

BASFETCHA
LIS

BASGET
LIS

BASFSP
LIS

BASFIND
LIS

BASFORMAT
LIS

BASHANDLE
LIS