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

```
BBBBBBBB      AAAAAA      SSSSSSSS    CCCCCCCC    OOOOOO    PPPPPPPP    YY      YY  FFFFFFFFFF  DDDDDDDD
BBBBBBBB      AAAAAA      SSSSSSSS    CCCCCCCC    OOOOOO    PPPPPPPP    YY      YY  FFFFFFFFFF  DDDDDDDD
BB      BB  AA      AA  SS          CC          OO      OO  PP      PP  YY      YY  FF          DD      DD
BB      BB  AA      AA  SS          CC          OO      OO  PP      PP  YY      YY  FF          DD      DD
BB      BB  AA      AA  SS          CC          OO      OO  PP      PP    YY  YY    FF          DD      DD
BB      BB  AA      AA  SS          CC          OO      OO  PP      PP    YY  YY    FF          DD      DD
BBBBBBBB    AA      AA    SSSSSS    CC          OO      OO  PPPPPPPP        YY      FFFFFFFF    DD      DD
BBBBBBBB    AA      AA    SSSSSS    CC          OO      OO  PPPPPPPP        YY      FFFFFFFF    DD      DD
BB      BB  AAAAAAAAAA          SS  CC          OO      OO  PP              YY      FF          DD      DD
BB      BB  AAAAAAAAAA          SS  CC          OO      OO  PP              YY      FF          DD      DD
BB      BB  AA      AA          SS  CC          OO      OO  PP              YY      FF          DD      DD
BB      BB  AA      AA          SS  CC          OO      OO  PP              YY      FF          DD      DD
BBBBBBBB    AA      AA  SSSSSSSS      CCCCCCCC    OOOOOO    PP              YY      FF          DDDDDDDD
BBBBBBBB    AA      AA  SSSSSSSS      CCCCCCCC    OOOOOO    PP              YY      FF          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
```

Table of contents

```
0000     1          .TITLE  BAS$$COPY_FD    Copy floating or double
0000     2          .IDENT  /1-005/         ; File: BASCOPYFD.MAR Edit: PLL1005
0000     3
0000     4  ;
0000     5  ;*****************************************************************************
0000     6  ;*                                                                           *
0000     7  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9  ;*  ALL RIGHTS RESERVED.                                                     *
0000    10  ;*                                                                           *
0000    11  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    12  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    13  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    14  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    15  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    16  ;*  TRANSFERRED.                                                             *
0000    17  ;*                                                                           *
0000    18  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    19  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    20  ;*  CORPORATION.                                                             *
0000    21  ;*                                                                           *
0000    22  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    23  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24  ;*                                                                           *
0000    25  ;*                                                                           *
0000    26  ;*****************************************************************************
0000    27  ;
0000    28  ;
0000    29  ;++
0000    30  ; ABSTRACT:
0000    31  ;
0000    32  ;       This module contains routines to copy floating ,double, gfloat,
0000    33  ;       and hfloat numbers from one place in memory to another.  This
0000    34  ;       is done with special instructions to check for reserved
0000    35  ;       operands.  It is done by subroutine because BLISS does not have
0000    36  ;       floating point manipulation.
0000    37  ;
0000    38  ;
0000    39  ;--
0000    40  ;
0000    41  ; VERSION: 1
0000    42  ;
0000    43  ; HISTORY:
0000    44  ;
0000    45  ; AUTHOR:
0000    46  ;       John Sauter, 03-FEB-1979
0000    47  ;
0000    48  ; MODIFIED BY:
0000    49  ;
0000    50  ;
0000    51  ;
```

```
0000     53          .SBTTL  HISTORY         ; Detailed Current Edit History
0000     54
0000     55
0000     56 ; Edit History for Version 1 of BASCOPYFD
0000     57 ;
0000     58 ; 1-001 - Original.  JBS 03-FEB-1979
0000     59 ; 1-002 - Correct a typo in a comment.  JBS 08-MAY-1979
0000     60 ; 1-003 - Make JSB entry points end in _R1.  JBS 23-AUG-1979
0000     61 ; 1-004 - Add entry points for g & h floating.  PLL 9-Sep-81
0000     62 ; 1-005 - Fix bug in h entry point.  PLL 8-Apr-1982
```

```
                        .SBTTL  DECLARATIONS

;
; INCLUDE FILES:
;
;       NONE
;
; EXTERNAL SYMBOLS:
;
;       NONE
;
;
; MACROS:
;
;       NONE
;
;
; PSECT DECLARATIONS:
00000000                .PSECT  _BAS$CODE       PIC, SHR, LONG, EXE, NOWRT
;
;
; EQUATED SYMBOLS:
;
;       NONE
;
;
; OWN STORAGE:
;
;       NONE
```

```
                        0000     94          .SBTTL  BAS$$COPY_F_R1
                        0000     95
                        0000     96 ;++
                        0000     97 ; FUNCTIONAL DESCRIPTION:
                        0000     98 ;
                        0000     99 ;       Copys a single-precision (32-bit) floating number
                        0000    100 ;
                        0000    101 ; CALLING SEQUENCE:
                        0000    102 ;
                        0000    103 ;       JSB BAS$$COPY_F_R1 (src.rf.r, dst.wf.r)
                        0000    104 ;
                        0000    105 ; INPUT PARAMETERS:
                        0000    106 ;
                        0000    107 ;       SRC.rf.r                The number to be copied
                        0000    108 ;
                        0000    109 ; IMPLICIT INPUTS:
                        0000    110 ;
                        0000    111 ;       NONE
                        0000    112 ;
                        0000    113 ; OUTPUT PARAMETERS:
                        0000    114 ;
                        0000    115 ;       DST.wf.r                The place to which to copy the number
                        0000    116 ;
                        0000    117 ; IMPLICIT OUTPUTS:
                        0000    118 ;
                        0000    119 ;       NONE
                        0000    120 ;
                        0000    121 ; FUNCTION VALUE:
                        0000    122 ;
                        0000    123 ;       NONE
                        0000    124 ;
                        0000    125 ; SIDE EFFECTS:
                        0000    126 ;
                        0000    127 ;       May get Floating Reserved Operand hardware trap.
                        0000    128 ;
                        0000    129 ;--
                        0000    130
                        0000    131
                        0000    132 BAS$$COPY_F_R1::
7E   60   D0            0000    133         MOVL    (R0),-(SP)              ; Get the number without trapping
61   8E   50            0003    134         MOVF    (SP)+,(R1)              ; Trap if reserved and store
                        05  0006    135         RSB                             ; Return to caller.
```

```
                        0007   137          .SBTTL  BAS$$COPY_D_R1
                        0007   138
                        0007   139 ;++
                        0007   140 ; FUNCTIONAL DESCRIPTION:
                        0007   141 ;
                        0007   142 ;       Copys a double-precision (64-bit) floating number
                        0007   143 ;
                        0007   144 ; CALLING SEQUENCE:
                        0007   145 ;
                        0007   146 ;       JSB BAS$$COPY_D_R1 (src.rd.r, dst.wd.r)
                        0007   147 ;
                        0007   148 ; INPU  PARAMETERS:
                        0007   149 ;
                        0007   150 ;       SRC.rd.r                The number to be copied
                        0007   151 ;
                        0007   152 ; IMPLICIT INPUTS:
                        0007   153 ;
                        0007   154 ;       NONE
                        0007   155 ;
                        0007   156 ; OUTPUT PARAMETERS:
                        0007   157 ;
                        0007   158 ;       DST.wd.r                The place to which to copy the number
                        0007   159 ;
                        0007   160 ; IMPLICIT OUTPUTS:
                        0007   161 ;
                        0007   162 ;       NONE
                        0007   163 ;
                        0007   164 ; FUNCTION VALUE:
                        0007   165 ;
                        0007   166 ;       NONE
                        0007   167 ;
                        0007   168 ; SIDE EFFECTS:
                        0007   169 ;
                        0007   170 ;       May get Floating Reserved Operand hardware trap.
                        0007   171 ;
                        0007   172 ;--
                        0007   173
                        0007   174 BAS$$COPY_D_R1::
7E   60   7D            0007   175          MOVQ    (R0),-(SP)              ; Get the number without trapping
61   8E   70            000A   176          MOVD    (SP)+,(R1)              ; Trap if reserved and store
                  05    000D   177          RSB                             ; Return to caller.
                        000E   178 ;
```

```
                     000E   180          .SBTTL  BAS$$COPY_G_R1
                     000E   181
                     000E   182 ;++
                     000E   183 ; FUNCTIONAL DESCRIPTION:
                     000E   184 ;
                     000E   185 ;       Copys a g floating number
                     000E   186 ;
                     000E   187 ; CALLING SEQUENCE:
                     000E   188 ;
                     000E   189 ;       JSB BAS$$COPY_G_R1 (src.rg.r, dst.wg.r)
                     000E   190 ;
                     000E   191 ; INPUT PARAMETERS:
                     000E   192 ;
                     000E   193 ;       SRC.rg.r                The number to be copied
                     000E   194 ;
                     000E   195 ; IMPLICIT INPUTS:
                     000E   196 ;
                     000E   197 ;       NONE
                     000E   198 ;
                     000E   199 ; OUTPUT PARAMETERS:
                     000E   200 ;
                     000E   201 ;       DST.wg.r                The place to which to copy the number
                     000E   202 ;
                     000E   203 ; IMPLICIT OUTPUTS:
                     000E   204 ;
                     000E   205 ;       NONE
                     000E   206 ;
                     000E   207 ; FUNCTION VALUE:
                     000E   208 ;
                     000E   209 ;       NONE
                     000E   210 ;
                     000E   211 ; SIDE EFFECTS:
                     000E   212 ;
                     000E   213 ;       May get Floating Reserved Operand hardware trap.
                     000E   214 ;
                     000E   215 ;--
                     000E   216
                     000E   217 BAS$$COPY_G_R1::
7E    60    7D       000E   218          MOVQ    (R0),-(SP)              ; Get the number without trapping
61    8E  50FD       0011   219          MOVG    (SP)+,(R1)              ; Trap if reserved and store
                05     0015   220          RSB                             ; Return to caller.
                       0016   221 ;
```

```
                     0016   223          .SBTTL  BAS$$COPY_H_R3
                     0016   224
                     0016   225 ;++
                     0016   226 ; FUNCTIONAL DESCRIPTION:
                     0016   227 ;
                     0016   228 ;       Copys an h floating number
                     0016   229 ;
                     0016   230 ; CALLING SEQUENCE:
                     0016   231 ;
                     0016   232 ;       JSB BAS$$COPY_H_R3 (src.rh.r, dst.wh.r)
                     0016   233 ;
                     0016   234 ; INPUT PARAMETERS:
                     0016   235 ;
                     0016   236 ;       SRC.rh.r                The number to be copied
                     0016   237 ;
                     0016   238 ; IMPLICIT INPUTS:
                     0016   239 ;
                     0016   240 ;       NONE
                     0016   241 ;
                     0016   242 ; OUTPUT PARAMETERS:
                     0016   243 ;
                     0016   244 ;       DST.wh.r                The place to which to copy the number
                     0016   245 ;
                     0016   246 ; IMPLICIT OUTPUTS:
                     0016   247 ;
                     0016   248 ;       NONE
                     0016   249 ;
                     0016   250 ; FUNCTION VALUE:
                     0016   251 ;
                     0016   252 ;       NONE
                     0016   253 ;
                     0016   254 ; SIDE EFFECTS:
                     0016   255 ;
                     0016   256 ;       May get Floating Reserved Operand hardware trap.
                     0016   257 ;
                     0016   258 ;--
                     0016   259
                     0016   260 BAS$$COPY_H_R3::
7E    08 A0    7D    0016   261          MOVQ    B^8(R0),-(SP)
   7E    60    7D    001A   262          MOVQ    (R0),-(SP)              ; Get the number without trapping
   61    8E 70FD     001D   263          MOVH    (SP)+,(R1)              ; Trap if reserved and store
               05     0021   264          RSB                             ; Return to caller.
                      0022   265 ;
                      0022   266          .END
```

```
BAS$$COPY_D_R1   00000007 RG    01
BAS$$COPY_F_R1   00000000 RG    01
BAS$$COPY_G_R1   00000C0E RG    01
BAS$$COPY_H_R3   00000016 RG    01
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
_BAS$CODE	00000022 (34.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.08	00:00:00.74
Command processing	117	00:00:00.53	00:00:04.72
Pass 1	66	00:00:00.53	00:00:01.62
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	56	00:00:00.48	00:00:01.79
Symbol table output	1	00:00:00.01	00:00:00.01
Psect synopsis output	2	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	273	00:00:01.66	00:00:08.91

```
The working set limit was 750 pages.
2522 bytes (5 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 4 non-local and 0 local symbols.
266 source lines were read in Pass 1, producing 8 object records in Pass 2.
0 pages of virtual memory were used to define 0 macros.
```

Macro library statistics

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

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

There were no errors, warnings or information messages.

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

BASCLOSE
LIS

BASCONCAT
LIS

BASCTRLO
LIS

BASCHANGE
LIS

BASCTRLC
LIS

BASCHAIN
LIS

BASCOPYFD
LIS

BASCHR
LIS

BASCMPAPP
LIS

BASCUTOUT
LIS

BASCCPOS
LIS