DP FIXED POINT LOAD

DATE: February 1968

ID CODE: BPX

DRAWING: 391079 (Rev B)

LABEL: DLD

AUTHOR: STVL

SOURCE: SYM I Assembly Language

OBJECT: Relocatable

PURPOSE

To load a double precision fixed point argument from memory into the software registers MNT2, MNT3.

USAGE

Calling Sequence

```
L-1    SMB    DLD
L      JSX    DLD
L+1    D      DARG
```

Where DARG is the first of two words containing the number to be stored into the software registers. The routine will return to L+2 with the contents of MNT2 in the hardware accumulator.

Argument Description

The argument will be two consecutive words of memory.

Storage Requirements

Three words of common storage: RET1, MNT2, MNT3.

METHOD

Indexed loads and direct stores constitute the entire logic.

RESTRICTIONS

Entries

DLD

Other Routines

None.

External Constants

None.

Space Used

8 words.

Timing

15 cycles

APPENDIX A

ASSEMBLY LISTING

of

DP FIXED POINT LOAD

Drawing No.

391079 (Revision B)

ID Code

BPX

```
                                        2       'DP FIXED POINT LOAD DN3910/9 B'        BPX 0004
                                        3               BLK     MATH                    BPX 0005
                                        4               LIBR    DLD                     BPX 0006
                                        5       DLD     STX     RET1                    BPX 0007
*       0 000 0  67FF  6 0 7FF          6               LDX * 0                         BPX 0008
        0 001 0  9800  9 1 000          7               LDW * 1                         BPX 0009
        0 002 0  8801  8 1 001          8               STW     MNT3                    BPX 0010
*       0 003 0  77FF  7 0 7FF          9               LDW * 0                         BPX 0011
        0 004 0  8800  8 1 000          10              STW     MNT2                    BPX 0012
*       0 005 0  77FF  7 0 7FF          11              LDX     RET1                    BPX 0013
*       0 006 0  9000  9 0 000          12              JMP * 1                         BPX 0014
        0 007 0  1801  1 1 001          13              NTRY    DLD                     BPX 0015
                                        14              END                             BPX 0016
        0 007 0*******7
```

X-REF

```
LIB 0 000 0  DLD
EXT 0005     MNT2       0 005 0
EXT 0003     MNT3       0 003 0
EXT 0006     RET1       0 000 0     0 006 0
```

NO ERRORS

CARDS	SYMBOLS	LITR	STACK
14	4	615	0	2

DATE: February 1968
ID CODE: BPY
DRAWING: 391081 (Rev B)
LABEL: DST
AUTHOR: STVL
SOURCE: SYM I Assembly Language
OBJECT: Relocatable

PURPOSE

To store in two consecutive memory words a double precision fixed point number fetched from the software registers MNT2, MNT3.

USAGE

Calling Sequence

Where DARG is the first of two words into which the contents of the software registers will be stored. The routine will return to L+2 with the contents of MNT2 in the hardware accumulator.

```
L-1   SMB   DST
L     JSX   DST
L+1   D     DARG
```

Argument Description

The argument will be two consecutive words in memory.

Storage Requirements

Three words of common storage RET1, MNT2, MNT3.

METHOD

Direct loads and indexed stores constitute the entire logic.

RESTRICTIONS

Entries

DST

Other Routines

None

External Constants

None

Space Used

8 words

Timing

15 cycles

APPENDIX A

ASSEMBLY LISTING

of

DP FIXED POINT STORE

Drawing No.
391081 (Revision B)

ID Code
BPY

```
                                        2       'DP FIXED POINT STORE DN391081 B'                BPY 0004
                                        3               BLK   MATH                               BPY 0005
                                        4               LIBR  DST                                BPY 0006
   *    0 000 0  67FF  6 0 7FF          5       DST     STX   RET1                               BPY 0007
        0 001 0  9800  9 1 000          6               LDX * 0                                  BPY 0008
   *    0 002 0  87FF  8 0 7FF          7               LDW   MNT3                               BPY 0009
        0 003 0  7801  7 1 001          8               STW * 1                                  BPY 0010
   *    0 004 0  87FF  8 0 7FF          9               LDW   MNT2                               BPY 0011
        0 005 0  7800  7 1 000          10              STW * 0                                  BPY 0012
   *    0 006 0  9000  9 0 000          11              LDX   RET1                               BPY 0013
        0 007 0  1801  1 1 001          12              JMP * 1                                  BPY 0014
                                        13              NTRY  DST                                BPY 0015
        0 007 0*******7                 14              END                                      BPY 0016
```

X*REF

```
LIB 0 000 0  DST
EXT 0004     MNT2      0 004 0
EXT 0002     MNT3      0 002 0
EXT 0006     RET1      0 000 0    0 006 0
```

NO ERRORS

CARDS	SYMBOLS	LITR	STACK
14	4	626	0	2

RAYTHEON

QUALITY SOFTWARE

700 PROGRAMMING SYSTEMS

DP FIXED POINT ADD, SUBTRACT

DATE: February 1968

ID CODE: BPZ

DRAWING: 391083 (Rev C)

LABEL: DAD, DSUB

AUTHOR: STVL

SOURCE: SYM I Assembly Language

OBJECT: Relocatable

PURPOSE

To form the algebraic sum or difference of two double precision fixed point numbers set in the software registers MNT2, MNT3 and in two consecutive words of memory.

USAGE

Calling Sequence

The routine will return to L+2 with the result in the software registers MNT2, MNT3.

```
L-1 SMB     DAD    (or DSUB)
L   JSX     DAD    (or DSUB)
L+1 D       DARG
L+2 return
```

Argument Description

DARG is the first of two consecutive memory locations containing a fixed point double precision number.

Both arguments must be in double precision format: The sign bit of the second word of both arguments must be set to zero.

Storage Requirements

RET1 and pseudo-registers MNT2, MNT3

METHOD

DAD replaces the double register with the sum double register plus memory.

DSUB replaces the double register with the differences double register minus memory.

ERROR CONDITIONS

An overflow will turn on the overflow flip flop, and carry the most significant bit of the sum or difference in the sign bit of the high word (MNT2).

RESTRICTIONS:

Entries

DAD, DSUB

Other Routines

None

External Constants

D1 (decimal 1)
M15R (15 bit mask)

Space Used

34 words of core.

Timing

Excluding calling sequence.
DAD 24 $\pm$ 1 cycles
DSUB 23 $\pm$ 1 cycles

APPENDIX A

ASSEMBLY LISTING

of

DP FIXED POINT ADD, SUBTRACT

Drawing No.

391083 (Revision C)

ID Code

BPZ

```
                                  2     'DP FIXED POINT ADD, SUBTRACT DN391083 B'                                  BPZ 0004
                                  3             BLK   MATH                                                         BPZ 0005
                                  4             LIBR  DAD,DSUB                                                     BPZ 0006
                                  5     *                                                                          BPZ 0007
                                  6     *  A D D  ****************************************                         BPZ 0008
                                  7     *                                                                          BPZ 0009
*     0 000 0  67FF  6 0 7FF      8     DAD     STX   RET1     SAVE RETURN                                         BPZ 0010
      0 001 0  9800  9 1 000      9             LDX * 0        GET 2ND TERM                                        BPZ 0011
      0 002 0  8801  8 1 001     10             LDW * 1                                                            BPZ 0012
*     0 003 0  A7FF  A 0 7FF     11             ADD   MNT3     ADD LOWER WORDS                                     BPZ 0013
*     0 004 0  E7FF  E 0 7FF     12             AND   M15R     CLEAN SIGN BIT                                      BPZ 0014
*     0 005 0  7003  7 0 003     13             STW   MNT3                                                         BPZ 0015
      0 006 0  8800  8 1 000     14             LDW * 0                                                            BPZ 0016
      0 007 0  08A0  08A0        15             SNO                                                                BPZ 0017
*     0 008 0  A7FF  A 0 7FF     16             ADD   D1       CARRY                                               BPZ 0018
      0 009 0  08A0  08A0        17             SNO            OVERFLOW FROM CARRY                                 BPZ 0019
      0 00A 0  100F  1 0 00F     18             JMP   DAD0     YES - IT'S *2**15                                   BPZ 0020
*     0 00B 0  A7FF  A 0 7FF     19     DAD2    ADD   MNT2     ADD HIGHER WORDS                                    BPZ 0021
*     0 00C 0  700B  7 0 00B     20     EXIT    STW   MNT2                                                         BPZ 0022
*     0 00D 0  9000  9 0 000     21             LDX   RET1                                                         BPZ 0023
      0 00E 0  1801  1 1 001     22             JMP * 1                                                            BPZ 0024
*     0 00F 0  D00C  D 0 00C     23     DAD0    ORE   MNT2     (A) * 2 ** 15                                       BPZ 0025
*     0 010 0  A7FF  A 0 7FF     24             ADD   B0       OVERFLOW IF (A) POSITIVE                            BPZ 0026
*     0 011 0  D010  D 0 010     25             ORE   B0       RESTORE SIGN                                        BPZ 0027
      0 012 0  100C  1 0 00C     26             JMP   EXIT                                                         BPZ 0028
                                 27     *                                                                          BPZ 0029
                                 28     *  S U B T R A C T  ****************************                         BPZ 0030
                                 29     *                                                                          BPZ 0031
*     0 013 0  600D  6 0 00D     30     DSUB    STX   RET1     SAVE RETURN                                         BPZ 0032
      0 014 0  9800  9 1 000     31             LDX * 0        GET 2ND TERM                                        BPZ 0033
*     0 015 0  8005  8 0 005     32             LDW   MNT3                                                         BPZ 0034
      0 016 0  B801  B 1 001     33             SUB * 1        LOWER WORD                                          BPZ 0035
      0 017 0  0820  0820        34             SAM                                                                BPZ 0036
      0 018 0  101E  1 0 01E     35             JMP   DS82                                                         BPZ 0037
*     0 019 0  E004  E 0 004     36             AND   M15R     CARRY - CLEAN SIGN BIT                              BPZ 0038
*     0 01A 0  7015  7 0 015     37             STW   MNT3                                                         BPZ 0039
      0 01B 0  8800  8 1 000     38             LDW * 0                                                            BPZ 0040
      0 01C 0  0120  0120        39             INV                                                                BPZ 0041
      0 01D 0  100B  1 0 00B     40             JMP   DAD2                                                         BPZ 0042
*     0 01E 0  701A  7 0 01A     41     DS82    STW   MNT3     NO CARRY                                            BPZ 0043
*     0 01F 0  800F  8 0 00F     42             LDW   MNT2                                                         BPZ 0044
      0 020 0  B800  B 1 000     43             SUB * 0                                                            BPZ 0045
      0 021 0  100C  1 0 00C     44             JMP   EXIT                                                         BPZ 0046
                                 45     ************************************************************               BPZ 0047
                                 46             NTRY  DAD,DSUB                                                     BPZ 0048
      0 021 0*******33           47             END                                                                BPZ 0049
```

X-REF

```
EXT 0011      B0          0 010 0     0 011 0
LIB 0 000 0   DAD
    0 00F 0   DADO        0 00A 0
    0 008 0   DAD2        0 01D 0
    0 01E 0   DSB2        0 018 0
LIB 0 013 0   DSUB
EXT 0008      D1          0 008 0
    0 00C 0   EXIT.       0 012 0     0 021 0
EXT 001F      MNT2        0 008 0     0 00C 0     0 00F 0     0 01F 0
EXT 001E      MNT3        0 003 0     0 005 0     0 015 0     0 01A 0     0 01E 0
EXT 0019      M15R        0 004 0     0 019 0
EXT 0013      RET1        0 000 0     0 00D 0     0 013 0
```

NO ERRORS

CARDS	SYMBOLS	LITR	STACK
47	12	624	0	2

RAYTHEON

QUALITY SOFTWARE

700 PROGRAMMING SYSTEMS

DP FIXED POINT TWO'S COMPLEMENT

DATE: January 1968

ID CODE: BLJ

DRAWING: 390664 (Rev C)

LABEL: D2C

AUTHOR: STVL

SOURCE: SYM 1 Assembly Language

OBJECT: Relocatable

PURPOSE

To form the two's complement of a two-word number set in the software Registers MNT2, MNT3.

USAGE

Calling Sequence

The routine returnes to L+1 after two's complementing the double precision word in the software registers MNT2, MNT3.

```
    SMB   D2C
L   JSX   D2C
L+1 Return
```

Argument Description

The argument is in the software registers MNT2, MNT3.

Storage Requirements

External storage in the software registers MNT2, MNT3.

METHOD

The contents of MNT3 is changed to its 2's complement, while the contents of MNT2 is changed to its 1's complement, unless the lower word is zero, in which case the first word is set to its 2's complement.

RESTRICTIONS

Entries

D2C

Other Routines

None

External Constants

M15R (a mask of 15 bits, right adjusted)

Space Used

13 words

Timing

14 cycles

APPENDIX A

ASSEMBLY LISTING

of

DP FIXED POINT TWO'S COMPLEMENT

Drawing No.

390664 (Revision C)

ID Code

BLJ

```
                                        2       'DP FIXED POINT TWOS COMPLEMENT DN390664 C'                      BLJ 0004
                                        3               BLK   MATH                                              BLJ 0005
                                        4               LIBR  D2C                                               BLJ 0006
        0 000 0                         5       D2C     RES   0                                                 BLJ 0007
   *    0 000 0  87FF  8 0 7FF          6               LDW   MNT3                                              BLJ 0008
        0 001 0  0110  0110             7               CMP                                                     BLJ 0009
        0 002 0  0820  0820             8               SAM                                                     BLJ 0010
        0 003 0  100A  1 0 00A          9               JMP   LOWZ                                              BLJ 0011
   *    0 004 0  E7FF  E 0 7FF          10              AND   M15R                                              BLJ 0012
   *    0 005 0  7000  7 0 000          11              STW   MNT3                                              BLJ 0013
   *    0 006 0  87FF  8 0 7FF          12              LDW   MNT2                                              BLJ 0014
        0 007 0  0120  0120             13              INV                                                     BLJ 0015
   *    0 008 0  7006  7 0 006          14      EXIT    STW   MNT2                                             BLJ 0016
        0 009 0  1800  1 1 000          15              JMP * 0                                                 BLJ 0017
        0 00A 0                         16      LOWZ    RES   0                                                 BLJ 0018
   *    0 00A 0  8008  8 0 008          17              LDW   MNT2                                              BLJ 0019
        0 00B 0  0110  0110             18              CMP                                                     BLJ 0020
        0 00C 0  1008  1 0 008          19              JMP   EXIT                                              BLJ 0021
                                        20      **************************************************              BLJ 0022
                                        21              NTRY  D2C                                               BLJ 0023
        0 00C 0*******12                22              END                                                     BLJ 0024
```

X-REF

```
LIB 0 000 0  D2C
    0 008 0  EXIT       0 00C 0
    0 00A 0  LOWZ       0 003 0
EXT 000A     MNT2       0 006 0    0 008 0    0 00A 0
EXT 0005     MNT3       0 000 0    0 005 0
EXT 0004     M15R       0 004 0
```

NO ERRORS

```
CARDS  SYMBOLS    LITR  STACK
22     6      625  0     2
```

RAYTHEON

QUALITY SOFTWARE

700 PROGRAMMING SYSTEMS

DOUBLE SHIFT ARITHMETIC

DATE:	February 1968
ID CODE:	BRA
DRAWING:	391085 (Rev B)
LABEL:	DSR, DSL
AUTHOR:	STVL
SOURCE:	SYM I Assembly Language
OBJECT:	Relocatable

PURPOSE

To perform the arithmetic shift operation on a two-word argument.

USAGE

Calling Sequences

```
      SMB   DSR
      JSX   DSR, COUNT, ARGUMENT    for a shift right

      SMB   DSL
or    JSX   DSL, COUNT, ARGUMENT    for a shift left
```

Where COUNT is any integer and ARGUMENT is the address of the first word of the two-word argument to be shifted.

Argument Description

The first argument must be an integer. The second argument will be treated as a double precision integer.

Storage Requirements

Two words of external storage labeled RET1, TMP1 are used.

METHOD

The Double Shift Routine creates two shift instructions depending upon the size of the count. The two-word argument is loaded into the ACR and the IXR (hardware accumulator and index registers) and the shifts are executed. The shift argument is stored back in memory. A shift count greater than 30 is replaced by 30.

RESTRICTIONS

Entries

DSL, DSR

Other Routines

None

External Constants

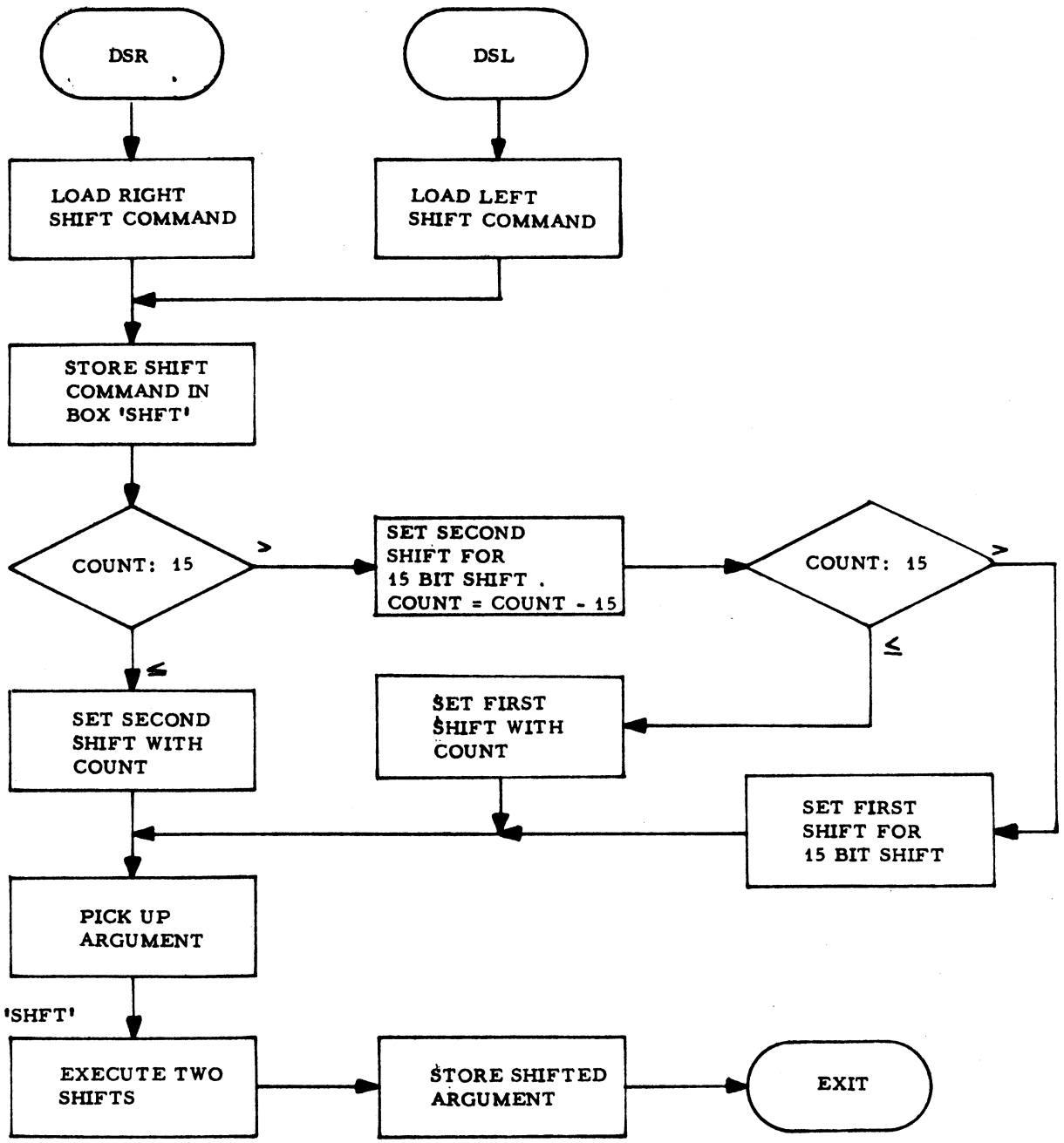
D15 (a constant equal to decimal 15)

Space Used

40 (X' 28') words

Timing

DSR	54 cycles minimum
	70 cycles maximum
	62 cycles average
DSL	55 cycles minimum
	71 cycles maximum
	63 cycles average


DSR
DSL
LOAD RIGHT SHIFT COMMAND
LOAD LEFT SHIFT COMMAND
STORE SHIFT COMMAND IN BOX 'SHFT'
COUNT: 15
>
SET SECOND SHIFT FOR 15 BIT SHIFT . COUNT = COUNT - 15
COUNT: 15
>
≤
≤
SET SECOND SHIFT WITH COUNT
SET FIRST SHIFT WITH COUNT
SET FIRST SHIFT FOR 15 BIT SHIFT
PICK UP ARGUMENT
'SHFT'
EXECUTE TWO SHIFTS
STORE SHIFTED ARGUMENT
EXIT

APPENDIX A

ASSEMBLY LISTING

of

DOUBLE SHIFT ARITHMETIC

Drawing No.	ID Code
391085 (Revision B)	BRA

```
                                    2       'DOUBLE SHIFT ARITHMETIC DN391085 B'                                BRA 0005
                                    3               BLK   MATH                                                  BRA 0006
                                    4               LIBR  DSL,DSR                                               BRA 0007
                                    5       *                                                                   BRA 0008
                                    6       *       DSR WILL SHIFT THE ARGUMENT RIGHT ARITHEMTIC                BRA 0009
                                    7       *       DSL WILL SHIFT THE ARGUMENT LEFT ARITHMETIC                 BRA 0010
                                    8       *       CALL WITH   JSX  DS-                                        BRA 0011
                                    9       *                   DATA COUNT    (A CONSTANT)                  BRA 0012
                                    10      *                   DATA  ARGUMENT(AN ADDRESS)                  BRA 0013
                                    11      *       RETURN HERE **                                              BRA 0014
                                    12      *                                                                   BRA 0015
  0 000 0                           13      DSL     RES   0                                                     BRA 0016
  0 000 0  8027  8 0 027            14              LDW   LSFT                                                  BRA 0017
  0 001 0  1003  1 0 003            15              JMP   $+2                                                   BRA 0018
  0 002 0                           16      DSR     RES   0                                                     BRA 0019
  0 002 0  8026  8 0 026            17              LDW   RSFT                                                  BRA 0020
  0 003 0  701A  7 0 01A            18              STW   SHFT                                                  BRA 0021
  0 004 0  701B  7 0 01B            19              STW   SHFT+1                                                BRA 0022
* 0 005 0  67FF  6 0 7FF            20              STX   RET1                                                  BRA 0023
  0 006 0  9800  9 1 000            21              LDX * 0                                                     BRA 0024
  0 007 0  0140  0140               22              CXA                                                         BRA 0025
  0 008 0  050F  05 0F              23              DXS   15                                                    BRA 0026
  0 009 0  8028  8 0 028            24              LDW   D15                                                   BRA 0027
  0 00A 0  C01B  C 0 01B            25              ORI   SHFT+1                                                BRA 0028
  0 00B 0  701B  7 0 01B            26              STW   SHFT+1                                                BRA 0029
  0 00C 0  0140  0140               27              CXA                                                         BRA 0030
  0 00D 0  0400  04 00              28              IXS   0                                                     BRA 0031
  0 00E 0  1013  1 0 013            29              JMP   $+5                                                   BRA 0032
  0 00F 0  050F  05 0F              30              DXS   15                                                    BRA 0033
  0 010 0  8028  8 0 028            31              LDW   D15                                                   BRA 0034
  0 011 0  C01A  C 0 01A            32              ORI   SHFT                                                  BRA 0035
  0 012 0  701A  7 0 01A            33              STW   SHFT                                                  BRA 0036
* 0 013 0  9005  9 0 005            34              LDX   RET1                                                  BRA 0037
  0 014 0  9801  9 1 001            35              LDX * 1                                                     BRA 0038
  0 015 0  8801  8 1 001            36              LDW * 1                                                     BRA 0039
  0 016 0  0A11  0A1 1              37              SLL   1                                                     BRA 0040
* 0 017 0  77FF  7 0 7FF            38              STW   TMP1                                                  BRA 0041
  0 018 0  8800  8 1 000            39              LDW * 0                                                     BRA 0042
* 0 019 0  9017  9 0 017            40              LDX   TMP1                                                  BRA 0043
  0 01A 0                           41      SHFT    RES   2                                                     BRA 0044
* 0 01C 0  7019  7 0 019            42              STW   TMP1                                                  BRA 0045
  0 01D 0  0140  0140               43              CXA                                                         BRA 0046
  0 01E 0  0A01  0A0 1              44              SRL   1                                                     BRA 0047
* 0 01F 0  9013  9 0 013            45              LDX   RET1                                                  BRA 0048
  0 020 0  9801  9 1 001            46              LDX * 1                                                     BRA 0049
  0 021 0  7801  7 1 001            47              STW * 1                                                     BRA 0050
* 0 022 0  801C  8 0 01C            48              LDW   TMP1                                                  BRA 0051
  0 023 0  7800  7 1 000            49              STW * 0                                                     BRA 0052
* 0 024 0  901F  9 0 01F            50              LDX   RET1                                                  BRA 0053
  0 025 0  1802  1 1 002            51              JMP * 2                                                     BRA 0054
  0 026 0  0920  092 0              52      RSFT    SRA D 0                                                     BRA 0055
  0 027 0  0930  093 0              53      LSFT    SLA D 0                                                     BRA 0056
  0 028 0  000F  000F               54      D15     D     15                                                    BRA 0057
```

```
                                    55              NTRY  DSR                                   BRA 0058
                                    56              NTRY  DSL                                   BRA 0059
        0 028 0*******40            57              END                                         BRA 0060
```

X-REF

```
LIB 0 000 0  DSL
LIB 0 002 0  DSR
    0 028 0  D15        0 009 0    0 010 0
    0 027 0  LSFT       0 000 0
EXT 0024     RET1       0 005 0    0 013 0    0 01F 0    0 024 0
    0 026 0  RSFT       0 002 0
    0 01A 0  SHFT       0 003 0    0 004 0    0 00A 0    0 00B 0    0 011 0    0 012 0
EXT 0022     TMP1       0 017 0    0 019 0    0 01C 0    0 022 0
```

NO ERRORS

CARDS	SYMBOLS	LITR	STACK
57	8	625	0	6

DATE: May 1968

ID CODE: BSP

DRAWING: 390014 (Rev B)

LABEL: M.ZE

AUTHOR: JACQ

SOURCE: SYM I

OBJECT: Relocatable in Block "MATH"

PURPOSE

To clear the three software registers MNT1, MNT2, MNT3.

USAGE

M.ZE is called by the library routines dealing with the mid-precision floating point format, when the software registers must be set to zero, as in the case of an underflow condition.

Calling Sequence

```
L-1     SMB     M.ZE
L       JSX     M.ZE
L+1     Return
```

RESTRICTIONS

Loading

M.ZE must be loaded in the same 2K block as the "MATH POOL".

Other Routines

None

Space Used

5 words

Timing

8 cycles

APPENDIX A

ASSEMBLY LISTING

of

MP FLOATING UNDERFLOW

Drawing No.

390014 (Revision B)

ID Code

BSP

```
                                        2      'MP FLOATING UNDERFLOW DN690014 B'                BSP 0005
                                        3      *                                                 BSP 0006
                                        4      *                                                 BSP 0007
                                        5               BLK    MATH                              BSP 0008
                                        6               LIBR   M,ZE                              BSP 0009
                    0000  0000          7      M.ZE     EQU    $                                 BSP 0010
      0 000 0       0100  0100          8               CLR                                      BSP 0011
*     0 001 0       77FF  7 0 7FF       9               STW    MNT1                              BSP 0012
*     0 002 0       77FF  7 0 7FF      10               STW    MNT2                              BSP 0013
*     0 003 0       77FF  7 0 7FF      11               STW    MNT3                              BSP 0014
      0 004 0       1800  1 1 000      12               JMP *  0                                 BSP 0015
                                       13               NTRY   M,ZE                              BSP 0016
      0 004 0*******4                  14               END                                      BSP 0017
```

X-REF

```
LIB 0 000 0  M,ZE        0 000 0
EXT 0001     MNT1        0 001 0
EXT 0002     MNT2        0 002 0
EXT 0003     MNT3        0 003 0
```

NO ERRORS

CARDS	SYMBOLS	LITR	STACK
14	4	626	0	2

RAYTHEON

QUALITY SOFTWARE

700 PROGRAMMING SYSTEMS

MP FLOATING OVERFLOW

DATE: May 1968

ID CODE: BSR

DRAWING: 390015 (Rev B)

LABEL: M.OV

AUTHOR: JACQ

SOURCE: SYM I

OBJECT: Relocatable in Block "MATH"

PURPOSE

To flag an overflow condition in the overflow flag word OVFL, and to set the software registers MNT1, MNT2, MNT3 to the maximum magnitude of the mid-precision floating point format, keeping the sign as found in MNT2.

USAGE

M.OV is called by the library routines dealing with the mid-precision floating point format, when an overflow condition occurs.

Calling Sequence

```
L-1     SMB     M.OV
L       JSX     M.OV
L+1     Return
```

METHOD

The flag word OVFL of the "MATH POOL" area is set to non-zero.

RESTRICTIONS

Loading

M.OV must be loaded in the same 2k block as the "MATH POOL".

Other Routines

D2C

Space Used

11 words

Timing

21 cycles

APPENDIX A

ASSEMBLY LISTING

of

MP FLOATING OVERFLOW

Drawing No.	ID Code
390015 (Revision B)	BSR

```
                                                2       'MP FLOATING OVERFLOW DN390015 B'                                                BSR 0005
                                                3       *                                                                               BSR 0006
                                                4       *                                                                               BSR 0007
                                                5               BLK     MATH                                                            BSR 0008
                                                6               LIBR    M,OV                                                            BSR 0009
                          0000  0000            7       M,OV    EQU     $                                                               BSR 0010
*       0 000 0  67FF   6 0 7FF                 8               STX     OVFL    SAVE RETURN AND SET FLAG WORD                           BSR 0011
*       0 001 0  97FF   9 0 7FF                 9               LDX     MNT2    SAVE SIGN                                               BSR 0012
*       0 002 0  87FF   8 0 7FF                 10              LDW     D255                                                            BSR 0013
*       0 003 0  77FF   7 0 7FF                 11              STW     MNT1                                                            BSR 0014
*       0 004 0  87FF   8 0 7FF                 12              LDW     M15R                                                            BSR 0015
*       0 005 0  7001   7 0 001                 13              STW     MNT2                                                            BSR 0016
*       0 006 0  77FF   7 0 7FF                 14              STW     MNT3                                                            BSR 0017
        0 007 0  0400   04 00                   15              IXS     0                                                               BSR 0018
*       0 008 0  27FF   2 0 7FF                 16              JSX     D2C                                                             BSR 0019
*       0 009 0  9000   9 0 000                 17              LDX     OVFL                                                            BSR 0020
        0 00A 0  1800   1 1 000                 18              JMP * 0                                                                 BSR 0021
                                                19              NTRY    M,OV                                                            BSR 0022
        0 00A 0*******10                        20              END                                                                     BSR 0023
```

X-REF

```
EXT 0008        D2C         0 008 0
EXT 0002        D255        0 002 0
LIB 0 000 0     M,OV        0 000 0
EXT 0003        MNT1        0 003 0
EXT 0005        MNT2        0 001 0     0 005 0
EXT 0006        MNT3        0 006 0
EXT 0004        M15R        0 004 0
EXT 0009        OVFL        0 000 0     0 009 0
```

NO ERRORS

```
CARDS  SYMBOLS    LITR  STACK
20     8          625   0     2
```

MP FLOATING, DP FIXED MPY AND DIV.

DATE: March 1968

ID CODE: BRB

DRAWING: 391096 (Revision D)

LABEL: FMP, FDV, DMP, DDV

AUTHOR: JACQ

SOURCE: SYM I Assembly Language

OBJECT: Relocatable

PURPOSE

DMP: To multiply two double precision fixed point numbers and return a double precision fixed point product.

DDV: To divide a double precision fixed point number by another and return a double precision fixed point quotient.

FMP: To multiply two mid-precision floating point numbers and return a mid-precision floating point

FDV: To divide a mid-precision floating point number by another and return a mid-precision floating point quotient.

USAGE

Calling Sequence

These routines will return to L+2 with the result in the software registers MNT1, MNT2, MNT3.

```
L-1     SMB     DMP (DDV, FMP, FDV)
L       JSX     DMP (DDV, FMP, FDV)
L+1     DATA    ARG
L+2     Return
```

Argument Description

Two arguments are necessary, one in the software registers and the other starting at the location ARG.

Storage Requirements

External storage in RET2, RET3, TMP1, TMP2, TMP3, TMP4, TMP5, TMP6, OVFL, and the software registers MNT1, MNT2, MNT3.

METHOD

In an initial sequence common to the four entries, the algebraic sign of the result is secured, then both arguments are set to their absolute value.

In a final sequence also common to all four entries, the result, quotient or product, is given its proper sign.

Two-word division

Let the dividend, divisor and quotient mantissae be N, D and Q.

$$N = N_1 + 2^{-15} \cdot N_2$$
$$D = Y_1 + 2^{-15} \cdot Y_2$$

In the floating point mode the exponent of the quotient is first computed.

$$E_Q = E_N - E_D + Bias$$

In the fixed point mode the divisor is normalized to bit 1 of its high word, and the dividend is shifted the same amount to the left.

The following condition is now satisfied:

$$.5 \leqslant Y_1 < 1$$

The dividend is then checked against the divisor. If it is not found smaller it is replaced by its difference with the divisor.

$N \geqslant D \quad X = N - D \quad Q = \frac{X}{D} + 1$

$N < D \quad X = N \quad Q = \frac{X}{D}$

The condition: $X < Y$ is now satisfied.

Let $k = 2^{-15}$

$$\frac{X}{D} = \frac{X_1 + k \cdot X_2}{Y_1} \cdot \frac{1}{1 + k \frac{Y_2}{Y_1}}$$

$$\frac{X}{D} = \frac{X_1 + k \cdot X_2}{Y_1} \left[1 - k \frac{Y_2}{Y_1} + k^2 \frac{Y_2^2}{Y_1^2} - k^3 \ldots \right]$$

The remaining terms of the series are dropped.

$$\text{Let } Q_1 + k \frac{R_1}{Y_1} = \frac{X_1 + k \cdot X_2}{Y_1} \qquad 0 \leq Q_1 < 1 \qquad 0 \leq R_1 < Y_1$$

$$\frac{X}{D} = Q_1 + k \frac{R_1}{Y_1} \cdot \left[1 - k \frac{Y_2}{Y_1} + k^2 \frac{Y_2^2}{Y_1^2} \right]$$

$$\frac{X}{D} = Q_1 + k \frac{R_1 - Q_1 \cdot Y_2}{Y_1} \left[1 - k \frac{Y_2}{Y_1} \right]$$

$$\text{Let } Q_2 + k \frac{R_2}{Y_1} = \frac{R_1 - Q_1 \cdot Y_2}{Y_1} \qquad 0 \leq Q_2 < 1 \qquad 0 \leq R_2 < Y_1$$

$$\frac{X}{D} = Q_1 + k \cdot Q_2 + k^2 \frac{R_2 - Q_2 \cdot Y_2}{Y_1}$$

The expression $(R_1 - Q_1 \cdot Y_2)$ is kept positive by decrementing Q_1 by k or 2k if necessary.

The expression $(R_2 - Q_2 \cdot Y_2)$ is not computed but only approximated to $\pm \frac{k^2}{4}$. If it is found negative, Q_2 is decremented by k^2.

$$\frac{X}{D} = Q_1 + k \cdot Q_2 \pm e \qquad e < k^2$$

The final quotient:

$Q = Q_1 + 2^{-15} Q_2 \pm e$ If $N < D$

$Q = 1 + Q_1 + 2^{-15} Q_2 \pm e$ If $N \geq D$

The unit bit is the sign bit of the high word. If it is used in the floating mode, the quotient mantissa is shifted 1 bit to the right. In that case the quotient exponent is incremented by 1.

Before return to the caller the quotient is set to the sign secured initially.

Two-word Multiplication

Let the factor mantissae be X and Y, the product mantissa be P.

$$X = X_1 + 2^{-15} X_2$$

$$Y = Y_1 + 2^{-15} Y_2$$

In the floating point mode the exponent of the product is first computed.

$$E_p = E_x + E_y - \text{Bias}$$

Let $k = 2^{-15}$

$$P = X \cdot Y = X_1 \cdot Y_1 + k\left(X_1 \cdot Y_2 + X_2 \cdot Y_1\right) + k^2 X_2 \cdot Y_2$$

The last term is approximated to $\pm k^2/4$ to force a carry into the lower word of the product.

$$P = P_1 + k \cdot P_2 \pm e \qquad e < k^2$$

In the floating mode the product is normalized to bit 1 of its high word. The product exponent is decremented by 1 if the mantissa has to be shifted.

Before return to the caller the product is set to the sign secured initially.

ERROR CONDITIONS

Error conditions are detected in the floating mode only.

1) An attempt to divide by zero or by an unnormalized divisor will leave the dividend unaltered in the software register.

2) An exponent overflow or underflow will give a quotient or product off by ± 256.

In both cases the flag word "OVFL" is set to non-zero. No error message is given.

RESTRICTIONS

Except for zero, the routine cannot properly process data equal to their own 2's complement, which is the case of the negative limit:

1st word 8000 (hexadecimal)
2nd word 0000

Entries

DMP, DDV, FMP, FDV

Other Routines

ACMY, DAD, DIVS, D2C, MPYS

External Constants

B0, D1, D128, D256, M8R, M15R, N1, N128

Space Used

205 words

Timing (in cycles) Average

	without hardware multiply and divide	with hardware multiply and divide
DMP	332	165
DDV	578	241
FMP	370	203
FDV	584	247

Deduct 20 cycles if both arguments are positive.

ACCURACY

29 bits

APPENDIX A

ASSEMBLY LISTING

of

MP FLOATING, DP FIXED MPY AND DIV.

Drawing No.

391096 (Revision C)

ID Code

BRB

```
                                    2     'MP FLOATING, DP FIXED MPY AND DIV DN391096 C'                    BRB 0005
                                    3     *                                                                 BRB 0006
                                    4     *                                                                 BRB 0007
                                    5             BLK   MATH                                                BRB 0008
                                    6             LIBR  FMP,FDV,DMP,DDV                                     BRB 0009
                                    7     ************************************************************      BRB 0010
                                    8     * CONSTANTS                                                      BRB 0011
      0 000 0  8031  8031           9     SH1     DATA  X'8031'                                             BRB 0012
                                   10     ************************************************************      BRB 0013
                 0001  0001        11     FDV     EQU   $                                                   BRB 0014
*     0 001 0  67FF  6 0 7FF       12             STX   RET3      SAVE RETURN                               BRB 0015
      0 002 0  9800  9 1 000       13             LDX * 0         GET TO DIVISOR                            BRB 0016
*     0 003 0  87FF  8 0 7FF       14             LDW   D128      EXPONENT BIAS                             BRB 0017
*     0 004 0  77FF  7 0 7FF       15             STW   RET2      SAVE AS FDV FLAG                          BRB 0018
      0 005 0  8800  8 1 000       16             SUB * 0         EXPONENT OF QUOTIENT                      BRB 0019
      0 006 0  100C  1 0 00C       17             JMP   FCOM      TO BE = XN - XD + 128 (+1)                BRB 0020
                 0007  0007        18     FMP     EQU   $                                                   BRB 0021
*     0 007 0  6001  6 0 001       19             STX   RET3      SAVE RETURN                               BRB 0022
      0 008 0  9800  9 1 000       20             LDX * 0         GET TO 2ND FACTOR                         BRB 0023
*     0 009 0  87FF  8 0 7FF       21             LDW   N128      - EXP, BIAS                               BRB 0024
*     0 00A 0  7004  7 0 004       22             STW   RET2      SAVE AS FMP FLAG                          BRB 0025
      0 00B 0  A800  A 1 000       23             ADD * 0         EXPONENT OF PRODUCT                       BRB 0026
*     0 00C 0  A7FF  A 0 7FF       24     FCOM    ADD   MNT0      TO BE = XF1 + XF2 - 128 (-1)              BRB 0027
*     0 00D 0  700C  7 0 00C       25             STW   MNT0                                                BRB 0028
      0 00E 0  8801  8 1 001       26             LDW * 1         GET ARGUMENT MANTISSA                     BRB 0029
      0 00F 0  9802  9 1 002       27             LDX * 2                                                   BRB 0030
      0 010 0  1019  1 0 019       28             JMP   COM                                                 BRB 0031
                                   29     ************************************************************      BRB 0032
                 0011  0011        30     DDV     EQU   $                                                   BRB 0033
*     0 011 0  87FF  8 0 7FF       31             LDW   D1        DDV FLAG                                  BRB 0034
      0 012 0  1014  1 0 014       32             JMP   DCOM                                                BRB 0035
                 0013  0013        33     DMP     EQU   $                                                   BRB 0036
*     0 013 0  87FF  8 0 7FF       34             LDW   N1        DMP FLAG                                  BRB 0037
*     0 014 0  700A  7 0 00A       35     DCOM    STW   RET2                                               BRB 0038
*     0 015 0  6007  6 0 007       36             STX   RET3      SAVE RETURN                               BRB 0039
      0 016 0  9800  9 1 000       37             LDX * 0         GET TO ARGUMENT                           BRB 0040
      0 017 0  8800  8 1 000       38             LDW * 0                                                   BRB 0041
      0 018 0  9801  9 1 001       39             LDX * 1                                                   BRB 0042
                                   40     ************************************************************      BRB 0043
*     0 019 0  D7FF  D 0 7FF       41     COM     ORE   MNT2                                                BRB 0044
*     0 01A 0  77FF  7 0 7FF       42             STW   TMP5      SIGN OF RESULT                            BRB 0045
*     0 01B 0  D019  D 0 019       43             ORE   MNT2                                                BRB 0046
      0 01C 0  0810  0810          44             SAP             GET ABSOLUTE VALUES                       BRB 0047
      0 01D 0  1047  1 0 047       45             JMP   NEG       OF BOTH ARGUMENTS                         BRB 0048
*     0 01E 0  77FF  7 0 7FF       46             STW   TMP3      2ND ARG, Y = Y1 + Y2 *2**-15              BRB 0049
*     0 01F 0  67FF  6 0 7FF       47             STX   TMP4                                                BRB 0050
*     0 020 0  801B  8 0 01B       48     COM1    LDW   MNT2      1ST ARG, X = X1 + X2 *2**-15              BRB 0051
      0 021 0  0810  0810          49             SAP                                                       BRB 0052
*     0 022 0  27FF  2 0 7FF       50             JSX   D2C                                                 BRB 0053
*     0 023 0  9014  9 0 014       51             LDX   RET2      BRANCH TO PROPER SEQUENCE                 BRB 0054
      0 024 0  0501  05 01         52             DXS   1                                                   BRB 0055
      0 025 0  1072  1 0 072       53             JMP   QUOT                                                BRB 0056
                                   54     ************************************************************      BRB 0057
```

```
                                  55     * WORK OUT TWO-WORD PRODUCT                                    BRB 0058
*   0 026 0  77FF  7 0 7FF        56     PROD     STW   TMP1      SAVE X1                              BRB 0059
*   0 027 0  87FF  8 0 7FF        57              LDW   MNT3      X2                                   BRB 0060
*   0 028 0  E01F  E 0 01F        58              AND   TMP4      Y2                                   BRB 0061
*   0 029 0  7020  7 0 020        59              STW   MNT2      X2 * Y2                              BRB 0062
*   0 02A 0  801E  8 0 01E        60              LDW   TMP3      Y1                                   BRB 0063
*   0 02B 0  27FF  2 0 7FF        61              JSX   ACMY      * X2 * Y1                            BRB 0064
*   0 02C 0  9029  9 0 029        62              LDX   MNT2                                           BRB 0065
*   0 02D 0  702C  7 0 02C        63              STW   MNT2                                           BRB 0066
*   0 02E 0  67FF  6 0 7FF        64              STX   TMP2      2ND WORD OF PARTIAL PRODUCT          BRB 0067
*   0 02F 0  8026  8 0 026        65              LDW   TMP1      X1                                   BRB 0068
*   0 030 0  7027  7 0 027        66              STW   MNT3                                           BRB 0069
*   0 031 0  8028  8 0 028        67              LDW   TMP4      Y2                                   BRB 0070
*   0 032 0  202B  2 0 02B        68              JSX   ACMY      * X1 * Y2                            BRB 0071
*   0 033 0  902F  9 0 02F        69              LDX   TMP1      X1                                   BRB 0072
*   0 034 0  7033  7 0 033        70              STW   TMP1      3RD WORD OF FINAL PRODUCT            BRB 0073
*   0 035 0  6030  6 0 030        71              STX   MNT3                                           BRB 0074
*   0 036 0  802A  8 0 02A        72              LDW   TMP3      Y1                                   BRB 0075
*   0 037 0  2032  2 0 032        73              JSX   ACMY      * X1 * Y1                            BRB 0076
*   0 038 0  A02E  A 0 02E        74              ADD   TMP2      SUM UP 2ND WORD OF PRODUCT           BRB 0077
*   0 039 0  E7FF  E 0 7FF        75              AND   M15R                                           BRB 0078
*   0 03A 0  7035  7 0 035        76              STW   MNT3                                           BRB 0079
*   0 03B 0  802D  8 0 02D        77              LDW   MNT2                                           BRB 0080
    0 03C 0  08A0  08A0           78              SNO                                                  BRB 0081
*   0 03D 0  A011  A 0 011        79              ADD   D1        CARRY INTO 1ST WORD                  BRB 0082
                                  80     ********************************************************      BRB 0083
*   0 03E 0  703B  7 0 03B        81     END      STW   MNT2                                          BRB 0084
*   0 03F 0  8023  8 0 023        82              LDW   RET2                                           BRB 0085
    0 040 0  0830  0830           83              SAO             FLOATING POINT                       BRB 0086
    0 041 0  1050  1 0 050        84              JMP   FEND      YES                                  BRB 0087
*   0 042 0  801A  8 0 01A        85     EXIT     LDW   TMP5      SIGN OF RESULT                       BRB 0088
    0 043 0  0810  0810           86              SAP                                                  BRB 0089
*   0 044 0  2022  2 0 022        87              JSX   D2C                                            BRB 0090
*   0 045 0  9015  9 0 015        88              LDX   RET3                                           BRB 0091
    0 046 0  1801  1 1 001        89              JMP * 1                                              BRB 0092
                                  90     ********************************************************      BRB 0093
                                  91     * NEGATIVE ARGUMENT - GET ABSOLUTE VALUE                     BRB 0094
    0 047 0  0501  05 01          92     NEG      DXS   1                                            BRB 0095
*   0 048 0  A03D  A 0 03D        93              ADD   D1                                             BRB 0096
    0 049 0  0110  0110           94              CMP                                                  BRB 0097
*   0 04A 0  7036  7 0 036        95              STW   TMP3                                           BRB 0098
    0 04B 0  0140  0140           96              CXA                                                  BRB 0099
    0 04C 0  0120  0120           97              INV                                                  BRB 0100
*   0 04D 0  E039  E 0 039        98              AND   M15R                                           BRB 0101
*   0 04E 0  7031  7 0 031        99              STW   TMP4                                           BRB 0102
    0 04F 0  1020  1 0 020       100              JMP   COM1                                           BRB 0103
                                 101     ********************************************************      BRB 0104
                                 102     * FLOATING POINT - NORMALIZE AND CHECK EXPONENT              BRB 0105
*   0 050 0  903A  9 0 03A       103     FEND     LDX   MNT3                                          BRB 0106
    0 051 0  0A31  0A3 1         104              SLL D 1                                              BRB 0107
*   0 052 0  803E  8 0 03E       105              LDW   MNT2                                           BRB 0108
    0 053 0  0810  0810          106              SAP                                                  BRB 0109
    0 054 0  105B  1 0 05B       107              JMP   FIXQ      OVERFLOW IN SIGN BIT                 BRB 0110
```

```
                0 055 0  0A31  0A3 1       108            SLL D 1                                          BRB 0111
                0 056 0  0810  0810        109            SAP                                              BRB 0112
                0 057 0  106A  1 0 06A     110            JMP   CHKE      ALL SET                          BRB 0113
                0 058 0  0800  0800        111            SAZ                                              BRB 0114
                0 059 0  1061  1 0 061     112            JMP   FIXP      TO NORMALIZE                     BRB 0115
                0 05A 0  1069  1 0 069     113            JMP   FIXZ                                       BRB 0116
                0 05B 0  0A21  0A2 1       114  FIXQ      SRL D 1         SHIFT 1 RIGHT = QUOTIENT         BRB 0117
              * 0 05C 0  7052  7 0 052     115            STW   MNT2                                       BRB 0118
                0 05D 0  0100  0100        116            CLR                                              BRB 0119
                0 05E 0  0A21  0A2 1       117            SRL D 1                                          BRB 0120
              * 0 05F 0  8048  8 0 048     118            LDW   D1        INCREMENT QUOTIENT EXPONENT      BRB 0121
                0 060 0  1067  1 0 067     119            JMP   FIXN                                       BRB 0122
              * 0 061 0  705C  7 0 05C     120  FIXP      STW   MNT2      SHIFT 1 LEFT = PRODUCT           BRB 0123
                0 062 0  0100  0100        121            CLR                                              BRB 0124
                0 063 0  0A63  0A6 3       122            SRC D 3                                          BRB 0125
              * 0 064 0  C034  C 0 034     123            ORI   TMP1                                       BRB 0126
                0 065 0  0A72  0A7 2       124            SLC D 2                                          BRB 0127
              * 0 066 0  8013  8 0 013     125            LDW   N1        DECREMENT PRODUCT EXPONENT       BRB 0128
              * 0 067 0  6050  6 0 050     126  FIXN      STX   MNT3                                       BRB 0129
              * 0 068 0  A00D  A 0 00D     127            ADD   MNT0                                       BRB 0130
              * 0 069 0  7068  7 0 068     128  FIXZ      STW   MNT0                                       BRB 0131
              * 0 06A 0  8069  8 0 069     129  CHKE      LDW   MNT0      CHECK EXPONENT                   BRB 0132
                0 06B 0  0600  06 00       130            LLB   0                                          BRB 0133
                0 06C 0  0800  0800        131            SAZ                                              BRB 0134
                         006D  006D        132  DVCK      EQU   $                                          BRB 0135
                0 06D 0  206F  2 0 06F     133            JSX   OUTE      EXPONENT OUT OF RANGE            BRB 0136
                0 06E 0  1042  1 0 042     134            JMP   EXIT                                       BRB 0137
                0 06F 0  0810  0810        135  OUTE      SAP                                              BRB 0138
              * 0 070 0  17FF  1 0 7FF     136            JMP   M,ZE      UNDERFLOW                        BRB 0139
              * 0 071 0  17FF  1 0 7FF     137            JMP   M,OV      OVERFLOW                         BRB 0140
                                           138  ********************************************************  BRB 0141
                                           139  * DIVIDE                                                  BRB 0142
              * 0 072 0  804A  8 0 04A     140  QUOT      LDW   TMP3      HIGH WORD OF DIVISOR             BRB 0143
                0 073 0  0A11  0A1 1       141            SLL   1                                          BRB 0144
                0 074 0  0810  0810        142            SAP             IS DIVISOR NORMALIZED            BRB 0145
                0 075 0  1095  1 0 095     143            JMP   NUM       YES                              BRB 0146
                                           144  * DENOMINATOR IS NOT NORMALIZED                          BRB 0147
                0 076 0  0850  0850        145  NORM      SXE             FLOATING POINT                   BRB 0148
                0 077 0  106D  1 0 06D     146            JMP   DVCK      YES - UNNORMALIZED OR ZERO       BRB 0149
                0 078 0  0800  0800        147            SAZ             FIXED POINT                      BRB 0150
                0 079 0  107C  1 0 07C     148            JMP   NRM1                                       BRB 0151
                0 07A 0  063F  06 3F       149            LLB   X'3F'     SHIFT COUNT = 15                 BRB 0152
                0 07B 0  1083  1 0 083     150            JMP   NRM2                                       BRB 0153
                0 07C 0  9000  9 0 000     151  NRM1      LDX   SH1                                        BRB 0154
                0 07D 0  0A11  0A1 1       152  NRM       SLL   1                                          BRB 0155
                0 07E 0  0810  0810        153            SAP                                              BRB 0156
                0 07F 0  1082  1 0 082     154            JMP   NRMX                                       BRB 0157
                0 080 0  0401  04 01       155            IXS   1                                          BRB 0158
                0 081 0  107D  1 0 07D     156            JMP   NRM                                        BRB 0159
                0 082 0  0140  0140        157  NRMX      CXA                                              BRB 0160
                0 083 0  3111  3 0 088 1   158  NRM2      STB   SFT1+1                                     BRB 0161
                0 084 0  3121  3 0 090 1   159            STB   SFT2+1                                     BRB 0162
              * 0 085 0  904E  9 0 04E     160            LDX   TMP4                                       BRB 0163
```

```
                0 086 0  0A31  0A3 1      161              SLL D 1                                                  BRB 0164
*               0 087 0  8072  8 0 072    162              LDW   TMP3                                               BRB 0165
                0 088 0  0930  093 0      163   SFT1       SLA D 0       **SHIFT DENOMINATOR                        BRB 0166
*               0 089 0  7087  7 0 087    164              STW   TMP3                                               BRB 0167
                0 08A 0  0140  0140       165              CXA                                                      BRB 0168
                0 08B 0  0A01  0A0 1      166              SRL   1                                                  BRB 0169
*               0 08C 0  7085  7 0 085    167              STW   TMP4                                               BRB 0170
*               0 08D 0  9067  9 0 067    168              LDX   MNT3                                               BRB 0171
                0 08E 0  0A31  0A3 1      169              SLL D 1                                                  BRB 0172
*               0 08F 0  8061  8 0 061    170              LDW   MNT2                                               BRB 0173
                0 090 0  0930  093 0      171   SFT2       SLA D 0       **SHIFT NUMERATOR                          BRB 0174
*               0 091 0  708F  7 0 08F    172              STW   MNT2                                               BRB 0175
                0 092 0  0140  0140       173              CXA                                                      BRB 0176
                0 093 0  0A01  0A0 1      174              SRL   1                                                  BRB 0177
*               0 094 0  708D  7 0 08D    175              STW   MNT3                                               BRB 0178
                                          176   * DENOMINATOR IS NORMALIZED                                        BRB 0179
                                          177   * CHECK DIVIDEND AGAINST DIVISOR                                   BRB 0180
*               0 095 0  8091  8 0 091    178   NUM        LDW   MNT2                                              BRB 0181
*               0 096 0  B089  B 0 089    179              SUB   TMP3        X1 - Y1                                BRB 0182
*               0 097 0  77FF  7 0 7FF    180              STW   TMP6                                               BRB 0183
                0 098 0  0810  0810       181              SAP             DIVIDEND TO DIVISOR                      BRB 0184
                0 099 0  10A8  1 0 0A8    182              JMP   DVD       LESS                                     BRB 0185
                0 09A 0  0130  0130       183              CAX                                                      BRB 0186
*               0 09B 0  8094  8 0 094    184              LDW   MNT3      MAY BE GREATER                           BRB 0187
*               0 09C 0  B08C  B 0 08C    185              SUB   TMP4      GET X - Y                                BRB 0188
                0 09D 0  0810  0810       186              SAP                                                      BRB 0189
                0 09E 0  0501  05 01      187              DXS   1                                                  BRB 0190
                0 09F 0  10A5  1 0 0A5    188              JMP   BIGN      GREATER                                  BRB 0191
*               0 0A0 0  7095  7 0 095    189              STW   MNT2      LESS, BUT EQUAL HIGH WORDS               BRB 0192
                0 0A1 0  0100  0100       190              CLR                                                      BRB 0193
*               0 0A2 0  709B  7 0 09B    191              STW   MNT3                                               BRB 0194
*               0 0A3 0  804D  8 0 04D    192              LDW   M15R      SET HIGH WORD OF QUOTIENT                BRB 0195
                0 0A4 0  10BA  1 0 0BA    193              JMP   BIGQ                                               BRB 0196
*               0 0A5 0  60A0  6 0 0A0    194   BIGN       STX   MNT2      DIVISOR LESS THAN DIVIDEND               BRB 0197
*               0 0A6 0  E0A3  E 0 0A3    195              AND   M15R      USE  Q = (X-Y)/Y + 1                     BRB 0198
*               0 0A7 0  70A2  7 0 0A2    196              STW   MNT3                                               BRB 0199
                                          197   * WORK OUT TWO-WORD QUOTIENT                                       BRB 0200
*               0 0A8 0  8096  8 0 096    198   DVD        LDW   TMP3      Y1                                       BRB 0201
*               0 0A9 0  27FF  2 0 7FF    199              JSX   DIVS      QUOTIENT HIGHER WORD Q1                  BRB 0202
*               0 0AA 0  9097  9 0 097    200              LDX   TMP6                                               BRB 0203
                0 0AB 0  0500  05 00      201              DXS   0                                                  BRB 0204
*               0 0AC 0  C7FF  C 0 7FF    202              ORI   80        ADD EXTRA 1 IF (X-Y) IS USED             BRB 0205
*               0 0AD 0  70AA  7 0 0AA    203              STW   TMP6      SAVE Q1 OR 1+Q1                          BRB 0206
*               0 0AE 0  80A5  8 0 0A5    204              LDW   MNT2                                               BRB 0207
*               0 0AF 0  7064  7 0 064    205              STW   TMP1      SAVE REMAINDER R1                        BRB 0208
*               0 0B0 0  809C  8 0 09C    206              LDW   TMP4      Y2                                       BRB 0209
                0 0B1 0  0110  0110       207              CMP                                                      BRB 0210
*               0 0B2 0  27FF  2 0 7FF    208              JSX   MPYS                                               BRB 0211
*               0 0B3 0  80AE  8 0 0AE    209              LDW   MNT2                                               BRB 0212
*               0 0B4 0  A0AF  A 0 0AF    210              ADD   TMP1      R1 - Q1 * Y2                             BRB 0213
*               0 0B5 0  70B3  7 0 0B3    211              STW   MNT2      NEXT DIVIDEND                            BRB 0214
                0 0B6 0  0820  0820       212   DVD1       SAM             IS QUOTIENT TOO BIG                      BRB 0215
                0 0B7 0  10BE  1 0 0BE    213              JMP   DVD2      NO                                       BRB 0216
```

```
    *   0 0B8 0  80AD   8 0 0AD    214             LDW   TMP6     YES                                  BRB 0217
    *   0 0B9 0  B05F   B 0 05F    215             SUB   D1       DECREMENT Q1                         BRB 0218
    *   0 0BA 0  70B8   7 0 0B8    216   BIGQ      STW   TMP6                                          BRB 0219
    *   0 0BB 0  27FF   2 0 7FF    217             JSX   DAD      ADD Y TO NEW DIVIDEND                BRB 0220
    *   0 0BC 0  30A8   30A8       218             D     TMP3                                          BRB 0221
        0 0BD 0  10B6   1 0 0B6    219             JMP   DVD1                                          BRB 0222
    *   0 0BE 0  80BC   8 0 0BC    220   DVD2      LDW   TMP3                                          BRB 0223
    *   0 0BF 0  20A9   2 0 0A9    221             JSX   DIVS                                          BRB 0224
        0 0C0 0  08A0   08A0       222             SNO                                                 BRB 0225
        0 0C1 0  10CB   1 0 0CB    223             JMP   DVD4                                          BRB 0226
        0 0C2 0  0130   0130       224             CAX           QUOTIENT LOWER WORD Q2                BRB 0227
    *   0 0C3 0  E0B0   E 0 0B0    225             AND   TMP4    Q2 * Y2                               BRB 0228
    *   0 0C4 0  B0B5   B 0 0B5    226             SUB   MNT2    REMAINDER R2                          BRB 0229
        0 0C5 0  0110   0110       227             CMP           R2 - Q2 * Y2                          BRB 0230
        0 0C6 0  0810   0810       228             SAP           IS QUOTIENT TOO BIG                   BRB 0231
        0 0C7 0  0501   05 01      229             DXS   1       YES - DECREMENT Q2                     BRB 0232
    *   0 0C8 0  60A7   6 0 0A7    230   DVD3      STX   MNT3    STORE QUOTIENT LOWER WORD             BRB 0233
    *   0 0C9 0  80BA   8 0 0BA    231             LDW   TMP6    Q1                                    BRB 0234
        0 0CA 0  103E   1 0 03E    232             JMP   END                                           BRB 0235
    *   0 0CB 0  90A6   9 0 0A6    233   DVD4      LDX   M15R    LIMIT FOR Q2                          BRB 0236
        0 0CC 0  10C8   1 0 0C8    234             JMP   DVD3                                          BRB 0237
                                   235   **************************************************          BRB 0238
                                   236             NTRY  FMP,FDV                                       BRB 0239
                                   237             NTRY  DMP,DDV                                       BRB 0240
        0 0CC 0*******204          238             END                                                 BRB 0241
```

X-REF

```
EXT 0037      ACMY      0 02B 0    0 032 0    0 037 0
    0 0A5 0   BIGN      0 09F 0
    0 0BA 0   BIGQ      0 0A4 0
EXT 00AC      B0        0 0AC 0
    0 06A 0   CHKE      0 057 0
    0 019 0   COM       0 010 0
    0 020 0   COM1      0 04F 0
EXT 0088      DAD       0 088 0
    0 014 0   DCOM      0 012 0
LIB 0 011 0   DDV       0 011 0
EXT 008F      DIVS      0 0A9 0    0 0BF 0
LIB 0 013 0   DMP       0 013 0
    0 06D 0   DVCK      0 06D 0    0 077 0
    0 0A8 0   DVD       0 099 0
    0 0B6 0   DVD1      0 0BD 0
    0 0BE 0   DVD2      0 0B7 0
    0 0C8 0   DVD3      0 0CC 0
    0 0CB 0   DVD4      0 0C1 0
EXT 0089      D1        0 011 0    0 03D 0    0 048 0    0 05F 0    0 0B9 0
EXT 0003      D128      0 003 0
EXT 0044      D2C       0 022 0    0 044 0
    0 03E 0   END       0 0CA 0
    0 042 0   EXIT      0 06E 0
    0 00C 0   FCOM      0 006 0
LIB 0 001 0   FDV       0 001 0
```

```
     0 050 0  FEND       0 041 0
     0 067 0  FIXN       0 060 0
     0 061 0  FIXP       0 059 0
     0 058 0  FIXQ       0 054 0
     0 069 0  FIXZ       0 05A 0
LIB  0 007 0  FMP        0 007 0
EXT  0071     M,0V       0 071 0
EXT  0070     M,±E       0 070 0
EXT  006A     MNT0       0 00C 0    0 00D 0    0 068 0    0 069 0    0 06A 0
EXT  00C4     MNT2       0 019 0    0 01B 0    0 020 0    0 029 0    0 02C 0    0 02D 0    0 038 0    0 03E 0
                         0 052 0    0 05C 0    0 061 0    0 08F 0    0 091 0    0 095 0    0 0A0 0    0 0A5 0
                         0 0AE 0    0 0B3 0    0 0B5 0    0 0C4 0
EXT  00C8     MNT3       0 027 0    0 030 0    0 035 0    0 03A 0    0 050 0    0 067 0    0 08D 0    0 094 0
                         0 098 0    0 0A2 0    0 0A7 0    0 0C8 0
EXT  00B2     MPYS       0 082 0
EXT  00CB     M15R       0 039 0    0 04D 0    0 0A3 0    0 0A6 0    0 0CB 0
     0 047 0  NEG        0 01D 0
     0 076 0  NORM
     0 07D 0  NRM        0 081 0
     0 082 0  NRMX       0 07F 0
     0 07C 0  NRM1       0 079 0
     0 083 0  NRM2       0 07B 0
     0 095 0  NUM        0 075 0
EXT  0066     N1         0 013 0    0 066 0
EXT  0009     N128       0 009 0
     0 06F 0  OUTE       0 06D 0
     0 026 0  PROD
     0 072 0  QUOT       0 025 0
EXT  003F     RET2       0 004 0    0 00A 0    0 014 0    0 023 0    0 03F 0
EXT  0045     RET3       0 001 0    0 007 0    0 015 0    0 045 0
     0 088 0  SFT1       0 083 0
     0 090 0  SFT2       0 084 0
     0 000 0  SH1        0 07C 0
EXT  00B4     TMP1       0 026 0    0 02F 0    0 033 0    0 034 0    0 064 0    0 0AF 0    0 0B4 0
EXT  0038     TMP2       0 02E 0    0 038 0
EXT  00BE     TMP3       0 01E 0    0 02A 0    0 036 0    0 04A 0    0 072 0    0 087 0    0 089 0    0 096 0
                         0 0A8 0    0 0BC 0    0 0BE 0
EXT  00C3     TMP4       0 01F 0    0 028 0    0 031 0    0 04E 0    0 085 0    0 08C 0    0 09C 0    0 0B0 0
                         0 0C3 0
EXT  0042     TMP5       0 01A 0    0 042 0
EXT  00C9     TMP6       0 097 0    0 0AA 0    0 0AD 0    0 0B8 0    0 0BA 0    0 0C9 0
```

NO ERRORS

CARDS	SYMBOLS	LITR	STACK
238	61	611	0	6

MP FLOATING, DP FIXED COMPARE

DATE: March 1968

ID CODE: BRC

DRAWING: 391088 (Rev B)

LABEL: FCM, DCM

AUTHOR: STVL

SOURCE: SYM I Assembly Language

OBJECT: Relocatable

PURPOSE

To compare an argument in memory, set either in mid precision floating point (FCM), or in double precision fixed point (DCM) form, to the number set, in the same format, in the pseudo-registers MNT1, MNT2, MNT3, (FCM) or MNT2, MNT3 (DCM).

USAGE

Calling Sequence

```
L-1   SMB     FCM (DCM)
L     JSX     FCM (DCM)
L+1   DATA    ARG
L+2   Return
```

Storage Requirements

External words RET1, MNT1, MNT2, MNT3

METHOD

DCM: compare most significant words and if equal compare least significant words.

FCM: if signs are different call DCM, otherwise compare exponents, if they are the same call DCM.

Both routines exit with the machine comparison flip-flops set to indicate the results of the compare. Skip instructions (SEQ, SLE, SGR, etc.) should be used to test the results of the compare.

RESTRICTIONS

Entries

FCM, DCM

Other Routines

None

External Constants

None

Space Used

28 words

Timing

FCM	26 cycles
DCM	15 cycles

APPENDIX A

ASSEMBLY LISTING

of

MP FLOATING, DP FIXED COMPARE

Drawing No.
391088 (Revision B)

ID Code
BRC

```
                                    2     'MP FLOATING, DP DIXED COMPARE DN391088 B'                    BRC 0004
                                    3             BLK   MATH                                           BRC 0005
                                    4             LIBR  FCM,DCM                                        BRC 0006
                                    5     *                                                            BRC 0007
                                    6     *       FCM COMPARES THE FLOATING REGISTER WITH MEMORY       BRC 0008
                                    7     *       DCM COMPARES THE DOUBLE REGISTER WITH MEMORY         BRC 0009
                                    8     *                                                            BRC 0010
  0 000 0                           9     FCM     RES   0                                              BRC 0011
* 0 000 0  67FF  6 0 7FF            10            STX   RET1                                           BRC 0012
  0 001 0  9800  9 1 000            11            LDX * 0                                              BRC 0013
  0 002 0  8801  8 1 001            12            LDW * 1                                              BRC 0014
* 0 003 0  D7FF  D 0 7FF            13            ORE   MNT2                                           BRC 0015
  0 004 0  0820  0820               14            SAM                                                  BRC 0016
  0 005 0  1012  1 0 012            15            JMP   EXCP                                           BRC 0017
  0 006 0  0401  04 01              16    FCMP    IXS   1                                              BRC 0018
  0 007 0  1008  1 0 008            17            JMP   $+4                                            BRC 0019
  0 008 0  1008  1 0 008            18            JMP   $+3                                            BRC 0020
  0 009 0                           19    DCM     RES   0                                              BRC 0021
* 0 009 0  6000  6 0 000            20            STX   RET1                                           BRC 0022
  0 00A 0  9800  9 1 000            21            LDX * 0                                              BRC 0023
* 0 00B 0  8003  8 0 003            22            LDW   MNT2                                           BRC 0024
  0 00C 0  F800  F 1 000            23            CMW * 0                                              BRC 0025
* 0 00D 0  87FF  8 0 7FF            24            LDW   MNT3                                           BRC 0026
  0 00E 0  0870  0870               25            SNE                                                  BRC 0027
  0 00F 0  F801  F 1 001            26            CMW * 1                                              BRC 0028
* 0 010 0  9009  9 0 009            27    FCXT    LDX   RET1                                           BRC 0029
  0 011 0  2801  2 1 001            28            JSX * 1                                              BRC 0030
  0 012 0                           29    EXCP    RES   0                                              BRC 0031
* 0 012 0  87FF  8 0 7FF            30            LDW   MNT1                                           BRC 0032
  0 013 0  F800  F 1 000            31            CMW * 0                                              BRC 0033
  0 014 0  0870  0870               32            SNE                                                  BRC 0034
  0 015 0  1006  1 0 006            33            JMP   FCMP                                           BRC 0035
* 0 016 0  800B  8 0 00B            34            LDW   MNT2                                           BRC 0036
  0 017 0  0820  0820               35            SAM                                                  BRC 0037
  0 018 0  1010  1 0 010            36            JMP   FCXT                                           BRC 0038
  0 019 0  8800  8 1 000            37            LDW * 0                                              BRC 0039
* 0 01A 0  F012  F 0 012            38            CMW   MNT1                                           BRC 0040
  0 01B 0  1010  1 0 010            39            JMP   FCXT                                           BRC 0041
                                    40            NTRY  FCM                                            BRC 0042
                                    41            NTRY  DCM                                            BRC 0043
  0 01B 0********27                 42            END                                                  BRC 0044
```

X-REF

```
LIB 0 009 0  DCM
    0 012 0  EXCP      0 005 0
LIB 0 000 0  FCM
    0 006 0  FCMP      0 015 0
    0 010 0  FCXT      0 018 0    0 01B 0
EXT 001A     MNT1      0 012 0    0 01A 0
EXT 0016     MNT2      0 003 0    0 00B 0    0 016 0
EXT 000D     MNT3      0 00D 0
EXT 0010     RET1      0 000 0    0 009 0    0 010 0
```

NO ERRORS

CARDS	SYMBOLS	LITR	STACK
42	9	624	0	6