DATE: February 1968
ID CODE: BPX
DRAWING: 394079 (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.  ID Code
391079   (Revision B)   BPX
MATH  DP FIXED POINT LOAD DN3910/9 R*

2   'DP FIXED POINT LOAD DN3910/9 R*
3   BLK  MATH
4   LDR  DLD
5   DLD  STX  RET1
*  0000  0  00 FF  0  00 7F
6   LDX  0
0010  9800  91 000
7   LDW  1
0020  8801  61 001
8   STW  MNT3
0030  77FF  70 7FF
9   LDW  0
0040  6800  81 000
10  STW  MNT2
0050  77FF  70 7FF
11  LDX  RET1
0060  9000  90 000
12  JMP  1
0070  1841  11 001
13  NTRY  DLD
0070  0000  00 00 FF
14  END

X*HEF

LIB 0 000 0 DLU
EXT 0005  MNT2  0 005 0
EXT 0003  MNT3  0 003 0
EXT 0006  RET1  0 000 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
'DP FIXED POINT STORE DN391081 B'

2       BLK MATH
3 67FF 6 7FF
4 DSTRBLKR DST
5 9800 9 1000
6 LDX 0
7 87FF 8 0 7FF
8 STW 1
9 7B01 7 1 001
10 STW 0
11 7B00 7 1 000
12 LDX RET1
13 1801 1 1 001
14 JMP 1
15 NTHY DST
16 END

X=REF
LIB 0 0 0 0 DST
EXT 0 0 0 4 MNT2 0 0 0 4 0
EXT 0 0 0 2 MNT3 0 0 0 2 0
EXT 0 0 0 6 RET1 0 0 0 6 0

NO ERRORS
CARDS SYMBOLS LITR STACK
14 4 626 0 2
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 $24 + 1$ cycles
DSUB $23 + 1$ cycles
APPENDIX A

ASSEMBLY LISTING

of

DP FIXED POINT ADD, SUBTRACT
DP FIXED POINT ADD, SUBTRACT DN391083 B'

          BLK     0005
LIBR     0006
A D D     0007
          0008
          0009
          0010
          0011
          0012
          0013
          0014
          0015
          0016
          0017
          0018
          0019
          0020
          0021
          0022
          0023
          0024
          0025
          0026
          0027
          0028
          0029
          0030
          0031
          0032
          0033
          0034
          0035
          0036
          0037
          0038
          0039
          0040
          0041
          0042
          0043
          0044
          0045
          0046
          0047
          0048
          0049

0000 0 67FF 6 0 7FF
0001 0 9800 9 1 000
0002 0 8901 8 1 001
0003 0 A7FF A 0 7FF
0004 0 A7FF E 0 7FF
0005 0 7003 7 0 003
0006 0 8800 8 1 000
0007 0 0800 0 800
0008 0 A7FF A 0 7FF
0009 0 0800 0 800
000A 0 1000 1 0 00F
000B 0 A7FF A 0 7FF
000C 0 7008 7 0 00B
000D 0 9000 9 0 000
000E 0 1001 1 1 001
000F 0 000C D 0 00C
0010 0 A7FF A 0 7FF
0011 0 2010 0 0 010
0012 0 100C 1 0 00C
0013 0 6000 6 0 00D
0014 0 9800 9 1 000
0015 0 8005 8 0 005
0016 0 8801 H 1 001
0017 0 0820 0820
0018 0 101E 1 0 01E
0019 0 E004 E 0 004
001A 0 7015 7 0 015
001B 0 8800 8 1 000
001C 0 0120 0120
001D 0 1008 1 0 00B
001E 0 1007 1 0 01A
001F 0 800F 8 0 00F
0020 0 8800 8 1 000
0021 0 100C 1 0 00C
0021 0******33

DSB2
J M P     0042
          0043
          0044
          0045
          0046
          0047
          0048
          0049

SUBTRACT
STX     0004
            0005
            0006
            0007
            0008
            0009
            0010
            0011
            0012
            0013
            0014
            0015
            0016
            0017
            0018
            0019
            0020
            0021
            0022
            0023
            0024
            0025
            0026
            0027
            0028
            0029
            0030
            0031
            0032
            0033
            0034
            0035
            0036
            0037
            0038
            0039
            0040
            0041
            0042
            0043
            0044
            0045
            0046
            0047
            0048
            0049

DSUB
STX     0004
            0005
            0006
            0007
            0008
            0009
            0010
            0011
            0012
            0013
            0014
            0015
            0016
            0017
            0018
            0019
            0020
            0021
            0022
            0023
            0024
            0025
            0026
            0027
            0028
            0029
            0030
            0031
            0032
            0033
            0034
            0035
            0036
            0037
            0038
            0039
            0040
            0041
            0042
            0043
            0044
            0045
            0046
            0047
            0048
            0049

END
D A D,D S U B   0004
            0005
            0006
            0007
            0008
            0009
            0010
            0011
            0012
            0013
            0014
            0015
            0016
            0017
            0018
            0019
            0020
            0021
            0022
            0023
            0024
            0025
            0026
            0027
            0028
            0029
            0030
            0031
            0032
            0033
            0034
            0035
            0036
            0037
            0038
            0039
            0040
            0041
            0042
            0043
            0044
            0045
            0046
            0047
            0048
            0049
X=REF

EXT 0011    B0    0 010 0    0 011 0
LIB 0 000 0    DAU
    0 00F 0    DAV0    0 00A 0
    0 009 0    DAV2    0 010 0
    0 01E 0    DSW2    0 018 0
LIB 0 013 0    DSUR
EXT 0008    DI
    0 00C 0    EXIT
    0 012 0    0 021 0
EXT 001F    MNT2
    0 00E 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    M19R
    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
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 returns 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.  ID Code
390664 (Revision C)  BLJ
MATH DP FIXED POINT TWOS COMPLEMENT DN390664 C' 09/27/68 PASS 2 PAGE 2

2 'DP FIXED POINT TWOS COMPLEMENT DN390664 C'
3 BLK MATH
4 LIBR D2C
5 D2C RES 0
6 LDW MNT3
7 CMP
8 SAM
9 JMP LOWZ
10 ANU M1SR
11 STW MNT3
12 LDW MNT2
13 INV
14 EXIT STW MNT2
15 JMP * 0
16 LOWZ RES 0
17 LDW MNT2
18 CMP
19 JMP EXIT
20 *******************
21 MTRY D2C
22 END

XREF

LIB 0 000 0 D2C
0 008 0 EXIT 0 00C 0
0 00A 0 LOWZ 0 005 0
EXT 000A MNT2 0 006 0 0 008 0 00A 0
EXT 0005 MNT3 0 000 0 0 005 0
EXT 0004 M1SR 0 004 0

NO ERRORS

CARDS SYMBOLS LITR STACK
22 6 625 0 2
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
.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
DSL  
55 cycles minimum
71 cycles maximum
63 cycles average

DSR  
54 cycles minimum
70 cycles maximum
62 cycles average
DSR

LOAD RIGHT SHIFT COMMAND

STORE SHIFT COMMAND IN BOX 'SHFT'

COUNT: 15

SET SECOND SHIFT WITH COUNT

PICK UP ARGUMENT

'SHFT'

EXECUTE TWO SHIFTS

DSL

LOAD LEFT SHIFT COMMAND

COUNT: 15

SET SECOND SHIFT FOR 15 BIT SHIFT

COUNT = COUNT - 15

SET FIRST SHIFT WITH COUNT

STORE SHIFTED ARGUMENT

EXIT

SET FIRST SHIFT FOR 15 BIT SHIFT

COUNT = COUNT - 15

SET FIRST SHIFT FOR 15 BIT SHIFT
APPENDIX A

ASSEMBLY LISTING

of

DOUBLE SHIFT ARITHMETIC

Drawing No. 391085 (Revision B)  ID Code BRA
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
MP FLOATING UNDERFLOW DN39U014 B'

2  'MP FLOATING UNDERFLOW DN39U014 B'
3 *
4 *
5 BLK MATH
6 LIBR M,2E
7 M.ZE EQU $00000000
8 CLK
9 STM MNT1
10 STM MNT2
11 STM MNT3
12 JMP #0
13 NTRY M,2E
14 END

X•REF

LIB 0 000 0 M,2E 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 LITH STACK
14 4 826 0 2
DATE: May 1968
ID CODE: BSR
DRAWING: 390015 (Rev B)
LABEL: M.OV
AUTHOR: JACQ
SOURCE: SYM 1
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. 390015 (Revision B)
MATH  HP FLOATING OVERFLOW DN390015 8
09/27/68  PASS 2  PAGE 2

2  'HP FLOATING OVERFLOW DN390015 8'
3   
4   
5   
6   
7   
8   
9   
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  

X-REF

EXT 0008  D2C  0 008 0
EXT 0012  D255  0 002 0
LIB 0 000  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
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.

\[ N = N_1 + 2^{-15}.N_2 \]
\[ D = Y_1 + 2^{-15}.Y_2 \]

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

\[ E_Q = E_N - E_D + \text{Bias} \]

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:

\[ 0.5 \leq Y_1 < 1 \]

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

\[ N \geq D \quad X = N - D \quad Q = \frac{X}{D} + 1 \]
\[ N < D \quad X = N \quad Q = \frac{X}{D} \]

The condition: \( X < Y \) is now satisfied.
Let $k = 2^{-15}$

\[
\frac{X}{D} = \frac{X_4 + k \cdot X_2}{Y_1} \cdot \frac{1}{1 + k \frac{Y_2}{Y_1}}
\]

\[
\frac{X}{D} = \frac{X_4 + k \cdot X_2}{Y_1} \left[ 1 - k \frac{Y_2}{Y_1} + k^2 \frac{Y_2}{Y_1}^2 - k^3 \ldots \right]
\]

The remaining terms of the series are dropped.

Let $Q_1 + k \frac{R_4}{Y_1} = \frac{X_4 + k \cdot X_2}{Y_1} \quad 0 \leq Q_1 < 1 \quad 0 \leq R_1 < Y_1$

\[
\frac{X}{D} = Q_1 + k \frac{R_4}{Y_1} \left[ 1 - k \frac{Y_2}{Y_1} + k^2 \frac{Y_2}{Y_1}^2 \right]
\]

\[
\frac{X}{D} = Q_1 + k \frac{R_4 - Q_1 \cdot Y_2}{Y_1} \left[ 1 - k \frac{Y_2}{Y_1} \right]
\]

Let $Q_2 + k \frac{R_2}{Y_1} = \frac{R_4 - Q_1 \cdot Y_2}{Y_1} \quad 0 \leq Q_2 < 1 \quad 0 \leq R_2 < Y_1$

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

The expression $(R_4 - Q_1 \cdot Y_2)$ 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 $\pm k^2$. If it is found negative, $Q_2$ is decremented by $k^2$.

\[
\frac{X}{D} = Q_1 + k \cdot Q_2 \pm e \quad e \leq k^2
\]
The final quotient:

\[ Q = Q_1 + 2^{-15} Q_2 + e \quad \text{if } N < D \]
\[ Q = 1 + Q_1 + 2^{-15} Q_2 + e \quad \text{if } N \geq D \]

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 \( + k^2 / 4 \) to force a carry into the lower word of the product.

\[ P = P_1 + k \cdot P_2 + e \quad e < k^2 \]

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 $\pm 256$.

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

<table>
<thead>
<tr>
<th></th>
<th>without hardware multiply and divide</th>
<th>with hardware multiply and divide</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMP</td>
<td>332</td>
<td>165</td>
</tr>
<tr>
<td>DDV</td>
<td>578</td>
<td>241</td>
</tr>
<tr>
<td>FMP</td>
<td>370</td>
<td>203</td>
</tr>
<tr>
<td>FDV</td>
<td>584</td>
<td>247</td>
</tr>
</tbody>
</table>

Deduct 20 cycles if both arguments are positive.

**ACCURACY**

29 bits
APPENDIX A

ASSEMBLY LISTING

of

MP FLOATING, DP FIXED MPY AND DIV.
55  * WORK OUT TWO-WORD PRODUCT  
56  PHQD  STW  TMP1  SAVE X1  
57  LDW  MNT3  X2  
58  AND  TMP4  Y2  
59  STW  MNT2  X2 * Y2  
60  LDW  TMP3  Y1  
61  JSX  ACMY  * X2 * Y1  
62  LDX  MNT2  
63  STW  MNT2  
64  STX  TMP2  2ND WORD OF PARTIAL PRODUCT  
65  LDW  TMP1  X1  
66  STW  MNT3  
67  LDW  MNT4  Y2  
68  JSX  ACMY  * X1 * Y2  
69  LDX  MNT1  X1  
70  STW  TMP1  3RD WORD OF FINAL PRODUCT  
71  STX  MNT3  
72  LDW  TMP3  Y1  
73  JSX  ACMY  * X1 * Y1  
74  ADD  TMP2  SUM UP 2ND WORD OF PRODUCT  
75  AND  M15R  
76  STW  MNT3  
77  LDW  MNT2  
78  SND  
79  AND  D1  CARRY INTO 1ST WORD  
80  END  STW  MNT2  
81  LDW  RET2  
82  SAO  FLOATING POINT  
83  JMP  FEND  YES  
84  EXIT  LDW  TMP5  SIGN OF RESULT  
85  SAP  
86  044  0202  2 0 022  
87  JSX  D2C  
88  LDX  RET3  
89  JMP  * 1  
90  * NEGATIVE ARGUMENT - GET ABSOLUTE VALUE  
91  UXS  1  
92  ADD  D1  
93  CMP  
94  STW  TMP3  
95  LDW  MNT3  
96  CXA  
97  INV  
98  AND  M15R  
99  STW  TMP4  
100  JMP  COM1  
101  * FLOATING POINT - NORMALIZE AND CHECK EXPONENT  
102  FEND  LDX  MNT3  
103  SLL  1  
104  SAP  
105  LDA  MNT2  
106  SAP  
107  JMP  FIX0  OVERFLOW IN SIGN BIT
MATH, MP FLOATING, MP FIXED MPY AND DIV BN391096 C!

09/27/68

PASS 2

PAGE 6

- 0 00H 0 80AD 8 0 0AD 214
  LD W TMP6
  YES
- 0 00H 0 80BF 8 0 05F 215
  SUB D1
  DECREMENT D1
- 0 00H 0 7088 7 0 0H8 216
  HIGQ
  STW TMP6
- 0 00H 0 07FF 2 0 7FF 217
  JSX DAD
  ADD Y TO NEW DIVIDEND
- 0 00H 0 30AB 30AB 218
  D TMP3
- 0 00H 0 1006 1 0 0B6 219
  JMP DVD1
- 0 00H 0 0086 8 0 086 220
  DVD2
  LD W TMP3
- 0 00H 0 20A9 2 0 0A9 221
  JSX DIV5
- 0 00H 0 08AD 08AD 222
  SNO
- 0 00H 0 10CB 1 0 0CB 223
  JMP DVD4
  QUOTIENT LOWER WORD Q2
- 0 00H 0 1300 1300 224
  CA X
- 0 00H 0 0090 9 0 090 225
  AND TMP4
  Q2 * Y2
- 0 00H 0 0858 8 0 085 226
  SUB MNT2
  REMAINDER R2
- 0 00H 0 0110 10 0 010 227
  CMP
  R2 - Q2 * Y2
- 0 00H 0 0810 8010 228
  SAP
  IS QUOTIENT TOO BIG
- 0 00H 0 0501 05 0 01 229
  DXS 1
  YES - DECREMENT Q2
- 0 00H 0 0A76 6 0 0A7 230
  DVD3
  STX MNT3
  STORE QUOTIENT LOWER WORD
- 0 00H 0 08A8 8 0 08A 231
  LD W TMP6
  Q3
- 0 00H 0 1036 1 0 036 232
  JMP EXIT
- 0 00H 0 0A69 9 0 0A6 233
  DVD4
  LD X M15R
  LIMIT FOR Q2
- 0 00H 0 10CB 1 0 0CB 234
  JMP DVD3

******************************************************

X*REF

EXT 0037  ACHY  0 028 0 0 032 0 0 037 0
  0 0A5 0 HIUW  0 09F 0
  0 09A 0 HIUW  0 04 0

EXT 0041  HI  0 0AC 0 1 0 07 0
  0 060 0 CMK  0 057 0
  0 019 0 CMK  0 10 0
  0 020 0 CMK  0 04F 0

EXT 0086  DIVS  0 086 0
  0 014 0 DCM  0 012 0

LIB 0 011 0 DDV  0 011 0

EXT 008F  DIVS  0 049 0
  0 0BF 0

LIB 0 013 0 JMP  0 013 0
  0 060 0 DVCK  0 06D 0
  0 0A8 0 DVU  0 099 0
  0 06E 0 DVU  0 08D 0
  0 0BF 0 DVU  0 087 0
  0 0C8 0 DVD  0 0C0 0
  0 0CB 0 DVD  0 0C1 0

EXT 009H  D1  0 011 0
  0 03D 0 0 048 0 0 05F 0 0 089 0

EXT 0093  D12R  0 03D 0

EXT 0044  D2C  0 022 0 0 044 0
  0 03E 0 ENU  0 0CA 0
  0 042 0 EXIT  0 06E 0
  0 000 0 FCMD  0 06 0

LIB 0 001 0 FDV  0 001 0
MATH MP FLOATING, DP FIXED MPY AND DIV DN391096 C1

0 050 0 FEND 0 041 0
0 067 0 FIXN 0 060 0
0 061 0 FIXP 0 059 0
0 059 0 FIXQ 0 054 0
0 069 0 FIXZ 0 05A 0
LIB 0 007 0 TPM 0 007 0
EXT 0071 0 M,OV 0 071 0
EXT 0070 0 M,KE 0 070 0
EXT 006A 0 MNT0 0 00C 0 0 0D0 0 0 068 0 0 069 0 0 06A 0
EXT 00C4 0 MNT2 0 019 0 0 018 0 0 020 0 0 029 0 0 02C 0 0 02D 0 0 038 0 0 03E 0
EXT 00C6 0 MNT3 0 052 0 0 05C 0 0 061 0 0 08F 0 0 091 0 0 095 0 0 0A0 0 0 0A5 0
EXT 00C8 0 MNT4 0 0AE 0 0 0A3 0 0 0B5 0 0 0C4 0
EXT 00C9 0 MNT5 0 027 0 0 030 0 0 035 0 0 03A 0 0 03C 0 0 050 0 0 05D 0 0 05E 0
EXT 00B2 0 MABS 0 082 0
EXT 00B4 0 MDEL 0 039 0 0 04D 0 0 0A3 0 0 0A6 0 0 0CB 0
0 047 0 NEW 0 01D 0
0 076 0 NRM 0 081 0
0 07D 0 NRM 0 083 0
0 07F 0 NRM1 0 079 0
0 083 0 NRM2 0 078 0
0 095 0 NUM 0 075 0
EXT 0066 0 N1 0 013 0 0 066 0
EXT 0009 0 N12B 0 009 0
0 06F 0 OUTE 0 0D0 0
0 026 0 PRED 0
0 072 0 QUOT 0 025 0
EXT 003F 0 REF2 0 004 0 0 00A 0 0 014 0 0 023 0 0 03F 0
EXT 0045 0 REF3 0 001 0 0 007 0 0 019 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 0 TMP1 0 026 0 0 02F 0 0 033 0 0 034 0 0 064 0 0 0AF 0 0 0B4 0
EXT 0038 0 TMP2 0 02E 0 0 038 0
EXT 00RE 0 TMP3 0 016 0 0 02A 0 0 036 0 0 04A 0 0 072 0 0 087 0 0 089 0 0 096 0
EXT 00C3 0 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
EXT 0042 0 TMP5 0 014 0 0 042 0
EXT 00C9 0 TMP6 0 097 0 0 0AA 0 0 0AD 0 0 088 0 0 0BA 0 0 0C9 0

NO ERRORS

CARDS  SYMBOLS  LITH  STACK
238  61  911  0  6
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