

mathfp

COLLABORATORS

	<i>TITLE :</i> mathfp		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		March 28, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	mathffp	1
1.1	mathffp.doc	1
1.2	mathffp.library/SPAbs	1
1.3	mathffp.library/SPAdd	1
1.4	mathffp.library/SPCeil	2
1.5	mathffp.library/SPCmp	2
1.6	mathffp.library/SPDiv	3
1.7	mathffp.library/SPFix	4
1.8	mathffp.library/SPFloor	4
1.9	mathffp.library/SPFlt	5
1.10	mathffp.library/SPMul	5
1.11	mathffp.library/SPNeg	6
1.12	mathffp.library/SPSub	6
1.13	mathffp.library/SPTst	7

Chapter 1

mathffp

1.1 mathffp.doc

SPAbs ()	SPCmp ()	SPFloor ()	SPNeg ()
SPAdd ()	SPDiv ()	SPFlt ()	SPSub ()
SPCeil ()	SPFix ()	SPMul ()	SPTst ()

1.2 mathffp.library/SPAbs

NAME

SPAbs - obtain the absolute value of the fast floating point number

C USAGE

```
fnum2 = SPAbs(fnum1);  
      d0
```

FUNCTION

Accepts a floating point number and returns the absolute value of said number.

INPUTS

fnum1 - floating point number

RESULT

fnum2 - floating point absolute value of fnum1

BUGS

None

SEE ALSO

`_LVOSPAbs`, `abs`

1.3 mathffp.library/SPAdd

NAME

SPAdd - add two floating point numbers

C USAGE

```
fnum3 = SPAdd(fnum1, fnum2);
           d1      d0
```

FUNCTION

Accepts two floating point numbers and returns the arithmetic sum of said numbers.

INPUTS

fnum1 - floating point number
fnum2 - floating point number

RESULT

fnum3 - floating point number

BUGS

None

SEE ALSO

`_LVOSPCAdd`, `faddi`

1.4 mathffp.library/SPCeil

NAME

SPCeil -- compute Ceil function of a number

SYNOPSIS

```
x = SPCeil( y );
d0      d0
```

```
float x,y;
```

FUNCTION

Calculate the least integer greater than or equal to x and return it. This identity is true. $\text{Ceil}(x) = -\text{Floor}(-x)$.

INPUTS

y -- Motorola Fast Floating Point Format Number

RESULT

x -- Motorola Fast Floating Point Format Number

BUGS

SEE ALSO

`SPFloor`

1.5 mathffp.library/SPCmp

NAME

SPCmp - compares two floating point numbers and sets appropriate condition codes

C USAGE

```
if (SPCmp(fnum1, fnum2)) {...}
    d1    d0
```

FUNCTION

Accepts two floating point numbers and returns the condition codes set to indicate the result of said comparison. Additionally, the integer functional result is returned to indicate the result of said comparison.

INPUTS

fnum1 - floating point number
fnum2 - floating point number

RESULT

Condition codes set to reflect the following branches:

```
GT - fnum2 > fnum1
GE - fnum2 >= fnum1
EQ - fnum2 = fnum1
NE - fnum2 != fnum1
LT - fnum2 < fnum1
LE - fnum2 <= fnum1
```

Integer functional result as:

```
+1 => fnum1 > fnum2
-1 => fnum1 < fnum2
 0 => fnum1 = fnum2
```

BUGS

None

SEE ALSO

`_LVOSPCmp`, `fcmpi`

1.6 mathffp.library/SPDiv

NAME

SPDiv - divide two floating point numbers

C USAGE

```
fnum3 = SPDiv(fnum1, fnum2);
        d1    d0
```

FUNCTION

Accepts two floating point numbers and returns the arithmetic division of said numbers.

INPUTS

fnum1 - floating point number
fnum2 - floating point number

RESULT

fnum3 - floating point number

BUGS

None

SEE ALSO

`_LVOSPDiv`, `fdivi`

1.7 mathffp.library/SPFix

NAME

SPFix - convert fast floating point number to integer

C USAGE

```
inum = SPFix(fnum);  
      d0
```

FUNCTION

Accepts a floating point number and returns the truncated integer portion of said number.

INPUTS

fnum - floating point number

RESULT

inum - signed integer number

BUGS

None

SEE ALSO

`_LVOSPFix`, `ffixi`

1.8 mathffp.library/SPFloor

NAME

SPFloor -- compute Floor function of a number

SYNOPSIS

```
x    = SPFloor( y );  
d0   d0
```

```
float  x,y;
```

FUNCTION

Calculate the largest integer less than or equal to x and return it.

INPUTS

y -- Motorola Fast Floating Point number

RESULT

x -- Motorola Fast Floating Point number

BUGS

SEE ALSO

SPCeil

1.9 mathffp.library/SPFlt

NAME

SPFlt - convert integer number to fast floating point

C USAGE

```
fnum = SPFlt(inum);  
      d0
```

FUNCTION

Accepts an integer and returns the converted floating point result of said number.

INPUTS

inum - signed integer number

RESULT

fnum - floating point number

BUGS

None

SEE ALSO

_LVOSPFlt, fflti

1.10 mathffp.library/SPMul

NAME

SPMul - multiply two floating point numbers

C USAGE

```
fnum3 = SPMul(fnum1, fnum2);  
          d1      d0
```

FUNCTION

Accepts two floating point numbers and returns the arithmetic multiplication of said numbers.

INPUTS

fnum1 - floating point number
fnum2 - floating point number

RESULT
fnum3 - floating point number

BUGS
None

SEE ALSO
_LVOSPMul, fmul_i

1.11 mathffp.library/SPNeg

NAME
SPNeg - negate the supplied floating point number

C USAGE
fnum2 = SPNeg(fnum1);
 d0

FUNCTION
Accepts a floating point number and returns the value of said number after having been subtracted from 0.0

INPUTS
fnum1 - floating point number

RESULT
fnum2 - floating point negation of fnum1

BUGS
None

SEE ALSO
_LVOSPNeg, fnegi

1.12 mathffp.library/SPSub

NAME
SPSub - subtract two floating point numbers

C USAGE
fnum3 = SPSub(fnum1, fnum2);
 d1 d0

FUNCTION
Accepts two floating point numbers and returns the arithmetic subtraction of said numbers.

INPUTS
fnum1 - floating point number
fnum2 - floating point number

RESULT

fnum3 - floating point number

BUGS

None

SEE ALSO

`_LVOSPSub`, `fsubi`

1.13 mathffp.library/SPTst

NAME

SPTst - compares a fast floating point number against the value zero (0.0) and sets the appropriate condition codes

C USAGE

```
if (!(SPTst(fnum))) {...}
    dl
```

FUNCTION

Accepts a floating point number and returns the condition codes set to indicate the result of a comparison against the value of zero (0.0). Additionally, the integer functional result is returned.

INPUTS

fnum - floating point number

RESULT

Condition codes set to reflect the following branches:

```
EQ - fnum = 0.0
NE - fnum != 0.0
PL - fnum >= 0.0
MI - fnum < 0.0
```

Integer functional result as:

```
+1 => fnum > 0.0
-1 => fnum < 0.0
0 => fnum = 0.0
```

BUGS

None

SEE ALSO

`_LVOSPTst`, `ftsti`
