Floating Point Addition / Subtraction

\# A and B both are floating point number.

$\Rightarrow$ A + B (Make sure the number is in binary)

i) Normalize both A and B.

ii) Align the bin point so that the lower exponent match with the higher exponent.

iii) Now add / sub accordingly.

iv) Normalize the result.

iv) Round if necessary.

Ex: $9.999 \times 10^{1} + 1.610 \times 10^{-1}$; size of exponent field is 3 bits

$= 99.99 + 0.1610$

$= 1100011.1111\ 1101\ 01 + 0.0010\ 1001\ 00$

$= 1.1000\ 1111\ 1111\ 0101 \times 2^{6} + 1.0100\ 100 \times 2^{-3}$

$= 1.1000\ 1111\ 1111\ 0101 \times 2^{6} + 0.0000\ 0000\ 1\ 0100\ 100 \times 2^{6}$

$= 1.10010 \times 2^{6}$ (Ans)

Bias $= 2^{3-1} - 1 = 3$

Biased Exp. $= 3 + 6 = 9$

Range $= 0$ to $2^{3} - 1$

$= 0$ to 7

$= 1$ to 6 [reserved 0 and 7]

upper range

$9 > 6 \Rightarrow$ So, overflow

\# $110100.111011 \times 2^{8} + 10110.11111 \times 2^{7}$

$= 1.1010\ 0111\ 011 \times 2^{13} + 1.0110\ 1111\ 1 \times 2^{11}$

$= 1.1010\ 0111\ 011 \times 2^{13} + 0.0101\ 1011\ 111 \times 2^{13}$

$= 10.0000\ 0011\ 010 \times 2^{13}$

$= 1.0000\ 0001\ 1010 \times 2^{14}$ (Ans)

Given number is too small to represent using the mentioned system.

Overflow / Underflow detection:

Step 1: Find the biased exponent of the answer.

Step 2: " " range of the biased exponent of the given system. (1 to upper Range)

Step 3: Detection:

if (Biased exponent < 1):

underflow

else if (Biased exponent > upper Range)

overflow

else : [1 ≤ Biased Exp ≤ upper Range]

No over/under flow

Floating Point Multiplication

A and B both are floating point number.

$\Rightarrow$ A × B (Make sure the number is in binary)

i) Normalize both A and B.

ii) Add the exponents.

iii) Now multiply accordingly.

iv) Normalize the result.

iv) Round if necessary.

v) Determine the sign from the operation.

Ex: $1.110 \times 2^{5} \times 1.11 \times 2^{-5}$

$= 1.110 \times 1.11 \times 2^{5+(-5)}$

$= 11.0001 \times 2^{0}$

$= 1.10001 \times 2^{1}$ (Ans)

F.P instructions in RiscV

Suppose, two single Preci. floating point numbers A, B are stored in memory. The memory locations are directly stored in register x_{10}, x_{11}. Write necessary code to store the result of A+B in the memory address that is stored in x_{13}.

Solⁿ:

```
flw     f1, 0(x10)    ; f1 = A
flw     f2, 0(x11)    ; f2 = B
fadd.s  f3, f1, f2    ; f3 = A+B
fsw     f3, 0(x13)
```