1 Float Point Design

The IEEE 754 standard is

signal S	exponent E	mantissa M
1 bit	8 bits	23 bits

which means

$$N = (-1)^s 2^{(e-127)} * 1.M$$

For instance:

0	00000000	00000000000000000000000	$+0.0$
1	00000000	00000000000000000000000	-0.0
0	01111111	00000000000000000000000	$+1.0$
1	01111111	00000000000000000000000	-1.0
0	10000000	00000000000000000000000	$+2.0$
0	10000001	11100000000000000000000	$+7.5$
0	11111111	01001010100010000000000	NaN

Where NaN =Not a Number. The number 2.0 is generated by:

$$\begin{aligned}
N &= (-1)^s 2^{(e-127)} * 1.0 && (1)\\
&= (-1)^0 2^{(10000000_2-127)} * 1.0 && (2)\\
&= 2^{(128-127)} * 1.0 && (3)\\
&= 2^{(1)} * 1.0 && (4)\\
&= 2.0 && (5)
\end{aligned}$$

The mantissa can be compute by using fractions, $2^{-1}, 2^{-2}, \ldots, 2^{-23}$, or $M = \sum_{i=-1}^{-23} D_i * 2^i$, or $2^{-1}, 2^{-2}, \ldots, 2^{-23} = 1/2, 1/4, \ldots, 1/2^{23}$. Let us consider 7.5, for instance:

$$\begin{aligned}
N &= (-1)^s 2^{(e-127)} * 1.M && (6)\\
&= (-1)^0 2^{(10000001_2-127)} * (1 + D_1 * 1/2 + \ldots + D_{23} * 1/2^{23}) && (7)\\
&= 2^{(129-127)} * (1 + 1/2 + 1/4 + 1/8) && (8)\\
&= 2^{(2)} * (\frac{8+4+2+1}{8}) && (9)\\
&= 4 * (\frac{15}{8}) && (10)\\
&= \frac{15}{2} && (11)\\
&= 7.5 && (12)
\end{aligned}$$

Let us consider a more simple Float-Point representation:

$$N = 2^{(e-3)} * 1.M$$

where 3 bits for the exponent and mantissa with 4 bits. For instance, $x_1 = 3$ and $x_2 = 3.25$.

Let us consider a float point adder operation

	$exponent$	$1Implicit$	$mantissa$
x_1	100	1	1000
x_2	100	1	1010
$s = x_1 + x_2$	100	11	0010

Or, $s = 2^{4-3} * (1 * 2^1 + 1 * 2^0 + 0 * 2^{-1} + 0 * 2^{-2} + 1 * 2^{-3})$ which is equal to $2 * (2 + 1 + 0.125) = 6.25$. However, the number have to be normalized to $N = 2^{5-3} * (1 + M)$, then $x_1 + x_2 = 2^2 * (1 + 1/2 + 1/16) = 6.25$, or

	exponent	1*Implicit*	*mantissa*
x_1	100	1	1000
x_2	100	1	1010
$s = x_1 + x_2$	100	11	0010
aligns =	101	1	1001

2 Multiplier

Let us consider $x_1 = 2^{e_1-3} \cdot (1 + M_1)$ e $x_2 = 2^{e_2-3} \cdot (1 + M_2)$. The multiplier result p will be $p = x_1 \cdot x_2 = 2^{e_1-3+e_2-3} \cdot (1.M_1 \cdot 1.M_2)$. To normalized, the exponent must be 2^{e_p-3}, then $e_p = e_1 + e_2 - 3$ and the mantissa will be multiplier.

For instance $x_1 = 3$ e $x_2 = 3.25$.

	exponent	1*Implicit*	*mantissa*
x_1	100	1	1000
x_2	100	1	1010
m	110	1	0011

Then $m = 2^{6-3} \cdot (1 + 2^{-3} + 2^{-4}) = 8 \cdot (\frac{16+2+1}{16}) = 9,5$. But, $3 \cdot 3.25 = 9.75$! What's happening ???

let us detail it...

					1	1	0	0	0
					1	1	0	1	0
–	–	–	–	–	–	–	–	–	–
					0	0	0	0	0
				1	1	0	0	0	
			0	0	0	0	0		
		1	1	0	0	0			
	1	1	0	0	0				
–	–	–	–	–	–	–	–	–	–
1	0	0	1	1	1	0	0	0	0

What is the weight of least significant bit ? $2^{-4} \cdot 2^{-4} = 2^{-8}$, so

	exponent	1*Implicit*	*mantissa*
x_1	100	1	1000
x_2	100	1	1010
m	?	10	01110000

To align the exponent: $e_p = e_1 + e_2 - 3$, so $e_p = 4 + 4 - 3 = 5$, that is, 2^5. To put inside our format $2^{e_p-3} = 2^{5-3} = 2^2$.

the result is $2^2 \cdot (2^2 + 2^{-2} + 2^{-3} + 2^{-4}) = 4 \cdot (\frac{32+4+2+1}{16}) = 9,75$.

		exponent	1*Implicit*	*mantissa*	*willlost...*
	x_1	100	1	1000	
after normalize it:	x_2	100	1	1010	
	m	101	10	0111	0000
	Normalize	110	1	0011	10000

The final result is rounded to $2^6 - 3 \cdot (1 + 2^{-3} + 2^{-4}) = 8 \cdot (\frac{16+2+1}{16}) = 9,5$.

3 Build a multiplier

A possible flow is

1. Compute the new exponent: $e_p = e_1 + e_2 - 3$
2. Use a 10 bit integer multiplier to compute the mantissa product. The input will be a 5 bit mantissa, DO NOT FORGET THE IMPLICIT 1 !! Simplify the multiplier by consider only the 6 most significative bits.
3. If the most significative bit is set, shift the result and align