{30-09-24}

* We handle things based on theory/principle (for now and leave the design decisions for later)

Floating point multipliers

operands $\left\{ \begin{array}{l} X = (-1)^{X_S} \cdot (1 \cdot \underbrace{\ldots X_m}_{\text{mantissa}}) \cdot 2^{X_E - 127} \\ Y = (-1)^{Y_S} \cdot (1 \cdot \ldots Y_m) \cdot 2^{Y_E - 127} \end{array} \right.$

result $Z = (-1)^{Z_S} \cdot (1 \cdot \underbrace{Z_m}_{\text{mantissa}}) \cdot 2^{Z_E - 127}$

We need to determine Z_S, Z_m, Z_E

$$\boxed{Z_S = X_S \oplus Y_S}$$

designer of multiplier is simpler (than adder)

Z_E also depends on multiplication of mantissa

* For mantissa multiplication, you essentially have a 24-bit multiplier (unsigned) if we ignore the binary point.
* We get a 48-bit result (with binary point after 46-bits)

Say $1.X_m = 1.000$ $\quad 1.Y_m = 1.111$

$1 \le 1.X_m < 2 \implies 1 \le 1./Z_m \quad (1.X_m)\cdot(1.Y_m) < 4$

$1 \le 1.Y_m < 2$ $\quad$ result can have 3 possibilities

* The leading two bits can be 10, 11 or 01 — shifting by 1 will be required and increasing exp by 1.

X_E and Y_E are biased exponents

(They're actually $X_E - 127$ & $Y_E - 127$)

So before normalization: resulting exponent (biased one) is $X_E + Y_E - 127$

$\Downarrow$

$((X_E - 127) + (Y_E - 127) + 127$

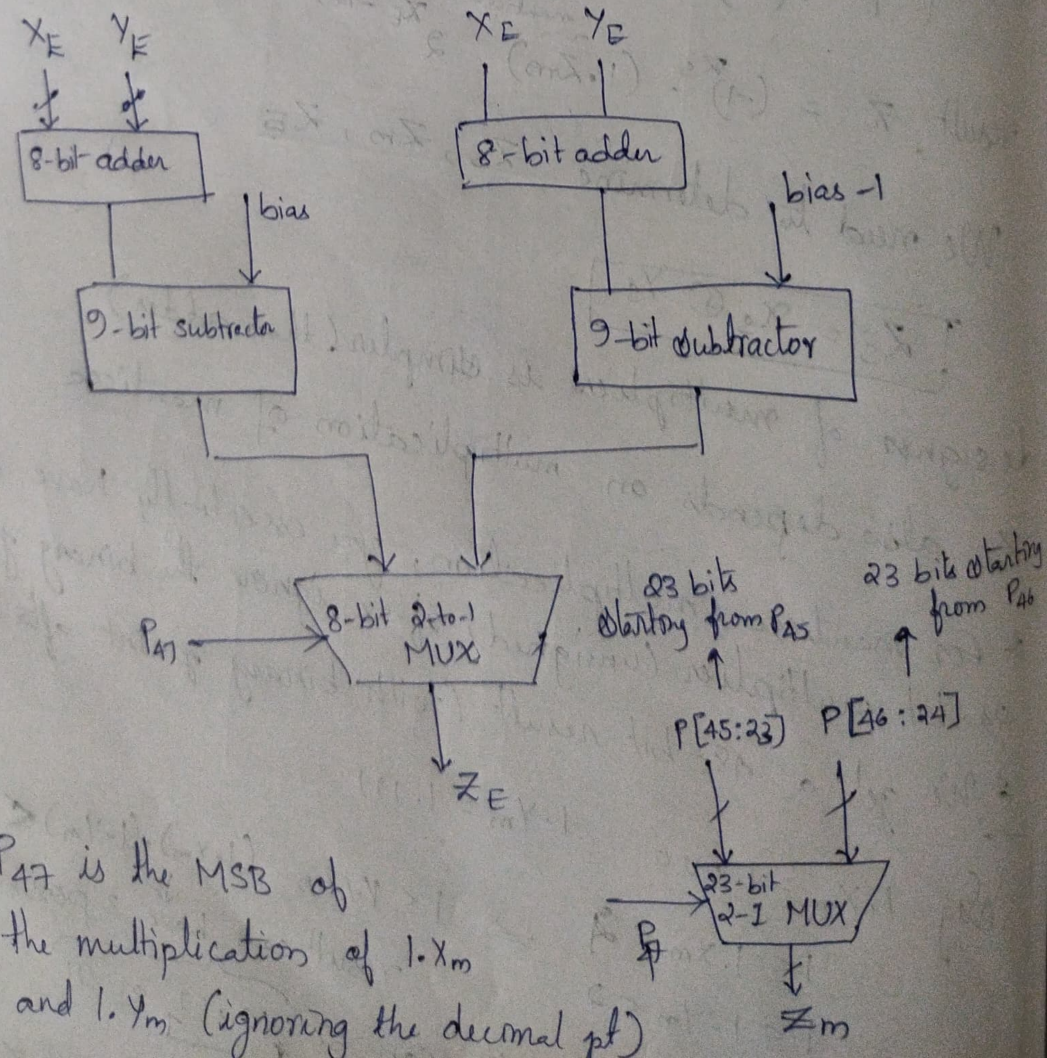
post normalization,

depending on the situation,

the biased exponent might increase by 1.

(either ways you can just add the shift value (0 or 1))

* Just pass 48^{th} bit into a multiplexer (2-to-1)



* P_{47} is the MSB of the multiplication of $1.X_m$ and $1.Y_m$ (ignoring the decimal pt)