Floating-Point numbers

1. Representation (MIPS-32)

a. 1-bit sign, 8-bit exponent, 23-bit fraction
b. IEEE 754 floating-point standard: 23-bit fraction → 24-bit significand. (To make the 1 leading bit in normalized binary numbers implicit.)
c. F = (-1)^s*(1+fraction)*2^(exponent-bias)
d. In exponent, 000…00 is the most negative (-127), 111…11 is the most positive (128). So we have a bias of 127
e. The smallest and largest unreserved biased exponents: 1~254. The difference between d and e is happening because all-0 is reserved for floating representation of 0 and all-1 is reserved for indicating values and situations outside the scope of normal floating point numbers.
f. For single precision, the maximum exponent is 127, and the minimum exponent is -126.

2. Addition

a. Compare exponents of the 2 numbers, and shift the smaller to right until it matches the larger one.
b. Add the significands.
c. Normalize the sum, either shift right and incrementing the exponent or shifting left and decrementing the exponent.
d. Check overflow or underflow.
e. (Round until the result is normalized.)

3. Multiplication

a. Add the biased exponents of the two numbers, subtraction the bias from the sum to get the new biased exponent.
b. Multiply the significands
c. Normalize the product if necessary, shifting it right and incrementing the exponent.
d. Check overflow or underflow
e. (Round until the result is normalized.)
f. Set the sign of the product. + if same original operands; - if different.

4. Reference:

1. Patterson and Hennessy, *Computer Organization and Design*, chapter 3.