CS61C — Precheck: C, Floating Point

Fall 2025 — Discussion 3

1 Discussion Pre-Check

1.1 The idea of floating point is to use the ability to move the radix (decimal) point wherever to represent a large range of real numbers as exact as possible.

1.2 Floating Point and Two's Complement can represent the same total amount of numbers (any reals, integer, etc.) given the same number of bits.

1.3 The distance between floating point numbers increases as the absolute value of the numbers increase.

1.4 Floating Point addition is associative.

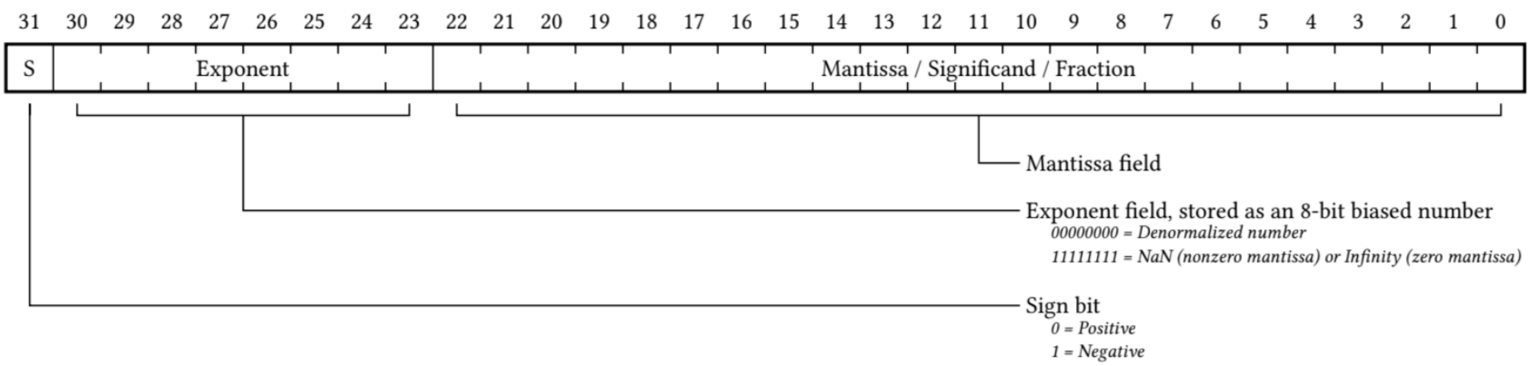
1.5 Why does normalized scientific notation always start with a 1 in base-2?

2 Floating Point

The IEEE 754 standard defines a binary representation for floating point values using three fields.

- The *sign* determines the sign of the number (0 for positive, 1 for negative).
- The *exponent* is in **biased notation**. For instance, the bias is −127, which comes from -($2^{\{8-1\}}$ − 1) for single-precision floating point numbers. For double-precision floating point numbers, the bias is −1023
- The *significand* (or *mantissa*) is akin to unsigned integers but used to store a fraction instead of an integer and refers to the bits to the right of the leading "`1`" when normalized. For example, the significand of `1.010011` is `010011`.

The table below shows the bit breakdown for the single-precision (32-bit) representation. The leftmost bit is the MSB, and the rightmost bit is the LSB.



For normalized floats:

$$\textbf{Value} = (-1)^{\text{Sign}} \times 2^{\text{Exp+Bias}} \times 1.\text{Significand}_2)$$

For denormalized floats:

$$\textbf{Value} = (-1)^{\text{Sign}} \times 2^{\text{Exp+Bias+1}} \times 0.\text{Significand}_2)$$

Exponent (Pre-bias)	**Significand**	**Meaning**
0	Anything	Denorm
1-254	Anything	Normal
255	0	± Infinity
255	Nonzero	NaN

Note that in the above table, our exponent has values from `0` to `255`. When translating between binary and decimal floating point values, we must remember that there is a bias for the exponent.