Floating Point Numbers

How do we represent numbers such as 5.75 in binary
In general to represent a real number in binary

- Find the binary representation of the int part
- Find the representation of the float part
- put a decimal point in between

$5_{10} = 101_2$
$0.75_{10} = 0.11_2$
$5.75_{10} = 101.11_2$

> *It is not however stored in this manner, but this can be used if done on paper or just for example.*

You must multiply by two and collect the whole number parts, as opposed to dividing by 2 and collecting the remainders.

4.25_{10}
$4_{10} = 100_2$

Fractional Part	Multiply by 2, collect the whole
$0.25 * 2$	0.5- bit is 0, remainder is 0.5
$0.5 * 2$	1- bit is 1, remainder is 0. stop

Write In the FORWARD direction
$4.25_{10} = 100.01_2$

12.375_{10}
$12_{10} = 1100_2$

Fractional Part	Multiply by 2, collect the whole
$.375 * 2$	.75 -> 0
$.75 * 2$	1.5 -> 1
$.5 * 2$	1. Stop

$12.375_{10} = 1100.011_2$

- Floating point numbers have an acceptable level of conversion, however it is not exact. EX. `3.10000000012143` for the value `3.1`
  For other Conversion View Binary Number Systems & Binary Arithmetic

Normalized Sign-Magnitude Form

- Represents floating point numbers
- $Num = F * 10^E$m, where F is the mantissa and E is the exponent
    - $251.38 = 2.5138 * 10^2$
- This is "normalized" is F's "whole number part" is a single digit number
    - scientific notation: $37.381 * 10^4$
    - $3.7381 * 10^5$ is normalized

The formula for the normalized notation is shown below:

$$1.x_2 * 2^y$$

We must because of limited bit storage weigh the trade-off of precision and size.

IEEE 754 Floating Point Standard

For Single Precision:

Bits	31 (1 bit)	30-23 (8bits)	22-0 (23 bits)
Purpose	Sign	Exponent (biased)	Mantissa

- Since the leading 1 bit is the most significant it is normalized in binary numbers as 1. It is not explicitly represented
- Defines standards for single and double precision floating point numbers

Mantissa is every number that follows the decimal point in the normalized notation

- Most negative numbers are represented as 00...00 rather than the positive alternative of 111...11
- This is done by adding 127 to the actual exponent to make sure the value stored is always positive.

The value of a number in IEEE 754 single precision is thus:

$$(-1)^{Sign} * x(1 + 0.Mantissa) * 2^{(Exponent-127)}$$

Double Precision

- Uses 64 bits (two 32-bit words)
- 1 bit for the sign
- 11 bits for the exponent
- 52 bits for the fraction
- 1023 as the bias

Bits	63 (1bit)	62-52 (11 bits)	51-0 (52 bits)
Purpose	Sign	Exponent (biased)	Mantissa

Overflow/underflow happens when a number is outside the range of a particular representation

- For example the exponent might be smaller than -38 (underflow) or larger than 38 (overflow)

The mantissa can be adjusted with the exponent for loss of precision, or might result in a denormalized number
Note that arithmetic operations can result in overflow/underflow

Procedure for converting decimals to float

1. Write in binary
2. Convert to standard form
3. The power of 2 is the exponent and the fractional part is the mantissa
4. Add the exponent to the bias (127 for 32 bit, 1023 for 64 bit) and convert the sum to binary
5. Write down the number as sign bit, exponent mantissa. Fill out the remaining bits with 0s.
6. Spilt the number into groups of 4. For each of the 4 bits write Hex equivalent