2.10 Floating-point representation

You have probably already have met standard scientific notation; e.g.

$$6.023 \times 10^{23}$$

- Here a number is represented as an integer place followed by a number of significant digits.
- We can do this in binary by simply moving the binary point. (commonly referred to as the decimal point) to a position to maximise the number of significant (digits) bits. Then the position of the binary point is recorded in an exponent representation
- Hence the name *floating point*; Here one plays off *efficiency* with *range* and *accuracy*.
- Since the Real numbers are continuous, we can only approximate these in a computer. Floating point maximises the resolution within a given set of space constraints.
- *Floating point* is in fact a form of sign magnitude. Actually *sign exponent magnitude*.

  As a general rule for binary
  $$\pm 0.f \times 2^{\pm e}$$

  or more formally write

  $$X = (-1)^S \times fraction \times 2^{\{exponent - K\}}$$

2.10.1 Sign S

0 for a positive number, 1 for a negative number.

2.10.2 Fraction

Just like you are used to, e.g.,
.0100011010

- The accuracy depends on number of bits. The idea is to maximise the number of significant bits, using the exponent to record the position of the point.
- We can always shift (except in the case of zero) till the MSB is a 1. Thus we can assume MSB to be a 1 (or 0) and then save space by not showing it.