Floating-Point Binary Data Representation

With fixed-point notation it is possible to represent a range of positive and negative values including a fractional component.
This method has limitations however, as we saw in the last hand-out. Very large numbers cannot be represented (the range is reduced) and very small fractions cannot be precisely represented.

Scientific Notation
In denary, we get around the limitations of fixed-point representation by using *scientific notation*. For example, the value 450,000,000,000 can be expressed as 4.5×10^{11}. Similarly, the value 0.000000000045 can be expressed as 45×10^{-11}. What we have done is to *move the point* to a more convenient place, as though it can slide. Then we keep note of the number of points it has moved, and indicate this in the exponent of 10. Because the decimal point can be moved, this method is referred to as *floating-point* representation. The same can be achieved in binary.

A floating-point number has the following form:

$$\pm \mathbf{S \times B^{\pm E}}$$

Such a binary number has three fields:

- Sign : Plus or minus
- Significand S
- Exponent E

In binary, B is always 2 and is never stored, since there is no need.
A number of standards exist for floating-point representation (see hand-out). Generally, the sign is indicated by a single bit (sign and magnitude notation); the significand is a "normalised" binary fraction, and the exponent is usually a signed binary value. Twos complement could be used for the exponent, but other negative number notations are generally used. The most common is called "biased" form.

Normalised Notation
Normalised scientific notation, also called "standard form" ensures that the significand begins **o.xxxx**.
The following scientific notation values are in *standard*, or *normalised* form:

0.3×10^{6}
0.75×10^{-12}
0.123456×10^{23}

Binary representations of the significand always use standard form.

Biased Form
This is another way of representing negative numbers in binary. Essentially, given **k** bits, the biased value is calculated by subtracting $(2^{k-1}-1)$ from the value given.
Eg: If we assume an 8-bit biased value, **k** = 8. The bias is therefore $2^{7}-1$, which is **127**. Hence the binary value **11011101** in biased form corresponds to 221-127 = +94.

The diagram below indicates a simple format for binary floating-point number representation: