Computer System Architecture
COMP201Th
Lecture: 19
Fixed and Floating Point Representation

In ordinary arithmetic, a negative number is indicated by a minus sign and a positive number by a plus sign. Because of hardware limitations, computers must represent everything with 1's and 0's including the sign of a number.

→ it is customary to represent the sign with a bit placed in the leftmost position of the number. *The convention is to make the sign bit equal to 0 for positive and 1 for negative.*

There are two ways of specifying the position of binary point (or decimal point) in a register:

1. **Using Fixed Position Representation**
   Binary point is fixed e.g. 1101101.0001001
2. **Using Floating Point Representation**
   Binary point floats to the right of the most significant 1 and an exponent is used e.g. $1.1011010001001 * 2^6$

Fixed Point Numbers:

The binary point is not a part of the representation but is implied. The number of integer and fraction bits must be agreed upon by those generating and those reading the number.

e.g. Fixed point representation using 4 integer bits and 3 fraction bits: 0110110 will be interpreted as 0110.110.

Floating-Point Representation:

The floating point representation of a number has two parts:

- the first part represents a signed, fixed-point number called **the mantissa.**
- the second part designates the position of the decimal (or binary) point and is called **the exponent.**

The fixed point mantissa may be a fraction or an integer.

Floating point is always interpreted to represent a number in the following form:

$$\mathbf{m * r^e}$$

Only the mantissa m and the exponent e are physically represented in the register (including their signs). The radix r and the radix point position of the mantissa are always assumed.

e.g. the decimal number +6132.789 is represented in floating point with a fraction and an exponent as follows:

Fraction	Exponent
+0.6132789	+04

The value of the exponent indicates that the actual position of the decimal point is four positions to the right of the indicated decimal point in the fraction.

This representation is equivalent to the scientific notation $+0.6132789 * 10^{+4}$.

Similarly, the binary number +1001.11 is represented with an 8-bit fraction and 6-bit exponent as follows:

Fraction	Exponent
01001110	000100

The fraction has a 0 in the leftmost position to denote positive.

The binary point of the fraction follows the sign bit. The exponent has the equivalent binary number +4. The floating point number is equivalent to:

$m*2^e = +(.1001110)_2*2^{+4}$

Normalization of Floating Point Number:

A floating point number is said to be normalized if the most significant digit of the mantissa is nonzero.

e.g. the decimal number 350 is normalized but 00035 is not.

8-bit binary number 00011010 is not normalized because the three leading 0's. The number can be normalized by shifting it three positions to the left and discarding the leading 0's to obtain 11010000. The three shift means multiply the number by 23 8. To keep the same value for the floating point number, the exponent must be subtracted by 3.

Normalized numbers provide the maximum possible precision for the floating-point number.

A zero cannot be normalized because it does not have a non zero digit. It is usually represented in floating point by all 0's in the mantissa and exponent.