Representation of Numbers

I.Nelson
SSN College of Engineering

Representation of Numbers:

- The main characteristic of digital arithmetic is the limited (usually fixed) number of digits used to represent numbers.

- This constraint leads to finite numerical precision in computations, which leads to round – off errors and nonlinear effects in the performance of digital filters.

Binary Fixed-point representation of Numbers:

- A real number X can be represented as

$X = (b_{-A}, \ldots\ldots, b_{-1}, b_0, b_1, \ldots. b_B)_r$

$$X = \sum_{i=-A}^{B} b_i r^{-i} \qquad 0 \le b_i \le (r-1)$$

where b_i represents the digit, r is the radix or base, A is the no. of integer digits and B is the no. of fractional digits.

(b_{-A} is the most significant bit and b_B is the least significant bit)

Note:

1. The fraction format with a binary point between b_0 and b_1 is usually used which permits numbers in the range from 0 to $1-2^{-n}$.

2. Any integer or mixed number can be represented in a fraction format by factoring out the term r^A.

3. Mixed numbers are difficult to multiply and the number of bits representing an integer cannot be reduced by truncation or rounding.

4. Therefore fractional format is used.

- There are 3 ways to represent the negative numbers or signed binary fractions.

- The format for positive fractions is the same in all three representations, i.e.,

$$X = 0.b_1 b_2 .....b_B = \sum_{i=1}^{B} b_i 2^{-i} \qquad X \geq 0$$

- Let us consider the negative fraction number be,

$$X = -0.b_1 b_2 .....b_B = -\sum_{i=1}^{B} b_i 2^{-i} \qquad X \le 0$$

- This number can be represented using any one of the following three formats.

1. Sign – magnitude format:

$$X_{SM} = 1.b_1 b_2 .....b_B \qquad X \le 0$$

2. One's – complement format:

$$X_{1C} = 1.\overline{b_1}\,\overline{b_2}\,.....\overline{b_B} \qquad X \le 0$$

$$\overline{b_i} = 1 - b_i$$

$$X_{1C} = 1 * 2^0 + \sum_{i=1}^{B} (1 - b_i)\, 2^{-i}$$

3. Two's – complement format:

$$X_{2C} = 1.\overline{b_1}\,\overline{b_2}\,.....\overline{b_B} + 0.00.....01 \qquad X \le 0$$

$$\overline{b_i} = 1 - b_i$$

$$X_{2C} = 1 * 2^0 + \sum_{i=1}^{B} (1 - b_i)\, 2^{-i} + 0.00.....01 = X_{1C} + 2^{-B}$$

Note:

- Addition is carried out by adding the numbers bit by bit in 1's and 2's complement. But in sign magnitude, it is more complex and can involve sign checks, complementing and generation of a carry, which can be done by some algorithms.

- Multiplication in Signed magnitude format is straightforward, whereas a special algorithm is usually employed for 1's and 2's complement multiplication.

- In general, the multiplication of two fixed – point numbers each of 'b' bits in length results in a product of '2b' bits in length. In fixed – point arithmetic, the product is either truncated or rounded back to 'b' bits. As a result, we have a truncation or round – off error in the 'b' least significant bits.

Binary Floating – point representation of numbers:

- A fixed point representation of numbers allows us to cover a range of numbers, say, $x_{max} - x_{min}$ with a resolution

$$\Delta = \frac{x_{max} - x_{min}}{m-1} = \frac{x_{max} - x_{min}}{2^b - 1}$$

i.e., the resolution is fixed and Δ increased in direct proportion to an increase in the dynamic range.

- A floating – point representation covers larger dynamic range. It consists of mantissa (M), which is the fractional part of the number and falls in the range $0.5 \leq M < 1$, multiplied by the exponential factor 2^E, where the exponent E is either a positive or negative integer.

  $X = M \cdot 2^E$

- Both mantissa and exponent requires a sign bit for representing positive and negative numbers.

Note:

- If the two numbers are to be multiplied, the mantissas are multiplied and the exponents are added.
- If the two numbers are to be added, the exponent of the smaller number is adjusted such that it is equal to the exponent of the larger number. This can be accomplished by shifting the mantissa of the smaller number to the right and compensating by increasing the corresponding exponent.
- Shifting operation required to equalize the exponent of two numbers results in loss of precision.
- Overflow occurs in the multiplication of two floating – point numbers when the sum of the exponents exceeds the dynamic range of the fixed – point representation of the exponent.
- For dynamic range, fixed-point representation has uniform resolution and floating-point representation provides fine resolution for small numbers and coarser resolution for the larger numbers.

Eg: For a 32 – bit computer

Representation		Range	Resolution
Fixed point	Positive integer	0 to $2^{32} - 1$ 0 to 4,294,967,295	1
	1 sign bit and 31 integer bits	$-(2^{31} - 1)$ to $(2^{31} - 1)$ -2,147,483,647 to 2,147,483,647	1
	1 sign bit, 21 integer bits and 10 fractional bits	$-(2^{31} - 1)*2^{-10}$ to $(2^{31} - 1) *2^{-10}$ $-(2^{21} - 2^{-10})$ to $(2^{21} - 2^{-10})$ -2,097,151.999 to 2,097,151.999	2^{-10}
Floating point	1 sign bit, 23 mantissa bits, 1 sign bit and 7 exponents bits	$1.11\ldots11 * 2^{0\ 1111111}$ to $0.11\ldots11 * 2^{0\ 1111111}$ $-1.701411632*10^{38}$ to $1.701411632*10^{38}$	Dynamic range $\approx 10^{38}$ Resolution is varying