Lecture 5 (10/4/2019)

Last time, we learned an IEEE format to represent numbers in computers, known as double-precision floating-point format.

$$\underbrace{c_0}_{\text{Sign}} \quad \underbrace{c_1 \cdots c_{11}}_{E} \quad \underbrace{a_1 \cdots a_{52}}_{\bar{x}}$$

This sequence of 64 bits represents the following number:

$$\sigma \bar{x} 2^e$$

where

$$\sigma = \begin{cases} 1 & \text{if } c_0 = 0 \\ -1 & \text{if } c_0 = 1 \end{cases}$$

$e = E - 1023$ if $1 \le E \le 2046$. (The cases $E=0$ and $E = 2047$ are special.)

$$\bar{x} = (1.a_1 a_2 \cdots a_{52})_2$$

- The case $E = 0$: $e = -1022$ and $\bar{x} = (0.a_1 a_2 \cdots a_{52})_2$
- The case $E = 2047$ represents $\pm\infty$ or NaN (not a number).
  If $a_1 = a_2 = \cdots = a_{52} = 0$: $\pm\infty$ (sign determined by σ)
  Otherwise, NaN.

This is called floating-point format to distinguish with fixed-point format.

Fixed-point	Floating-point
$(0.000001)_2$	$(1.00)_2 \times 2^{-6}$
$-(0.1100001)_2$	$-(1.10)_2 \times 2^{-1}$

The advantage of fixed-point format is that it can represent equally spaced numbers. (Think of example last lecture). The disadvantage is that the range of numbers it can represent is narrow. Floating-point format is the opposite. It is more preferred because the range of numbers it can represent is much wider.

What is the largest number that can be represented by the double precision floating-point format (don't count ∞)?

$$(1.\underbrace{11\ldots1}_{52})_2 \times 2^{1023}$$

$$\approx 8.988 \times 10^{307}$$

What is the smallest positive number?

$$(0.\underbrace{00\ldots01}_{51})_2 \times 2^{-1022}$$

$$= 2^{-52} \times 2^{-1022}$$

$$= 2^{-1074}$$

The ratio of the largest number to the smallest positive number is called dynamic range. In this case, it is about

$$\frac{2^{1024}}{2^{-1074}} = 2^{2098} \approx 10^{631}$$

Ex:

This is a toy model of IEEE double-precision floating-point format to demonstrate how machine arithmetic is done. Suppose a number is represented by 8 bits.

$$\underbrace{c_0}_{\text{sign}} \; \underbrace{c_1\, c_2\, c_3\, c_4}_{E} \; \underbrace{a_1\, a_2\, a_3}_{\bar{x}}$$

This sequence of 8 bits represents the following number:

$$\sigma\, \bar{x}\, 2^e$$

where

$$\sigma = \begin{cases} 1 & \text{if } c_0 = 0 \\ -1 & \text{if } c_0 = 1 \end{cases}$$

$e = E - 7$ if $1 \le E \le 14$. (The cases $E=0$ and $E=15$ are special.)

$$\bar{x} = (1.a_1 a_2 a_3)_2$$

The case $E = 0$: $e = -6$ and $\bar{x} = (0.a_1 a_2 a_3)_2$

The case $E = 15$ represents $\pm\infty$ (depending on the sign s)

1) What is the largest number (not counting ∞), smallest positive number, and dynamic range of this format?

2) If the bit sequence $c_0 c_1 \ldots a_3$ represented number $(c_0 c_1 \ldots a_3)_2$, what would be the answers to Question 1?

3) What number does the bit sequence 01101001 represent?

4) Do the following arithmetic operations using roundoff (not chopping).

To round off means $(1.101\underline{1})_2 \rightsquigarrow (1.110)_2$

add 1 to this digit

$(1.110\underline{0})_2 \rightsquigarrow (1.110)_2$

don't add 1 to this digit

- $(1.101)_2 \times 2^6 + (1.111)_2 \times 2^6$
- $(1.011)_2 \times 2^2 - (0.111)_2 \times 2^2$
- $-(1.010)_2 \times 2^1 - (1.011)_2 \times 2^{-1}$