Computer Organization & Architecture

2-8 Floating-point Numbers & IEEE 754 Standard

Wang Guohua

School of Software Engineering

Contents of this lecture

- Fixed-point Representation
- Floating-point Representation
- IEEE 754 Standard

Number Formats

- According to whether the position of binary point is fixed, there are two number formats:
  - Fixed-point numbers
    - E.g., integers, have an implied binary point at the right end of them.
  - Floating-point numbers

Fixed-point Representation (1)

- A fixed-point notation is any number in which the number of bits to the right of the binary point does not change.
  - Unsigned integers: with no bits to the right of the binary point.
  - Signed integers: with no bits to the right of the binary point.
  - Signed fractions: the binary point is to the right of the sign bit.

Fixed-point Representation (2)

- Let X = $x_n \ldots x_0$ be a fixed-point number

$$\underbrace{\boxed{x_n}}_{\text{sign}}\ \underbrace{\boxed{x_{n-1}}\ \boxed{\quad\cdots\quad}\ \boxed{x_0}}_{\text{magnitude}}$$

  - If X is an integer
    - The binary point is to the right of x_0
    - Range: $-2^n \le V(X) \le 2^n - 1$
  - If X is a pure fraction
    - The binary point is between x_n and x_{n-1}
    - Range: $-1 \le V(X) \le 1 - 2^{-n}$

Fixed-point Representation (3)

- Limitation
  - Very large integers can not be represented, nor can very small fractions.
  - Example: Consider the range of values representable in a 32-bit, signed, fixed-point format.
    - Interpreted as integers $-2^{31} \le V(X) \le 2^{31} - 1$

      $$V(X) \in [-2.15 \times 10^{9}, 0], [0, +2.15 \times 10^{9}]$$

    - Interpreted as fractions $-1 \le V(X) \le 1 - 2^{-31}$

      $$V(X) \in [-1, -4.55 \times 10^{-10}], [+4.55 \times 10^{-10}, +1]$$

Fixed-point Representation (4)

- Limitation (ctd.)
  - Example: Consider the range of values representable in a 32-bit, signed, fixed-point format. (ctd.)
    - In scientific calculations

      Avogadro's constant $6.02214076 \times 10^{23}\ \text{mol}^{-1} = 0.602214076 \times 10^{24}\ \text{mol}^{-1}$

      Planck's constant $6.62607015 \times 10^{-34}\ \text{J.s} = 0.62607015 \times 10^{-33}\ \text{J.s}$

Floating-point Representation (1)

- Floating-point Representation
  - The position of the binary point is variable and is automatically adjusted as computation proceeds.
  - The position of the binary point must be given explicitly in the floating-point representation.
  - Similar to scientific notation

Floating-point Representation (2)

- Floating-point Numbers in Computers
  - Encoding

S	E	M

  - Numerical Form
    - $(-1)^S\ M\ 2^e$
      - Sign bit *S* determines whether number is negative or positive
      - Mantissa *M*, a fraction in sign-magnitude or 2's complement representation, containing the significant digits
      - Exponent *E*
        - » In 2's complement or biased notation, the power of base
        - » Is not the actual exponent
        - » Actual exponent *e*

Floating-point Representation (3)

- Floating-point Numbers in Computers (ctd.)
  - Excess or Biased Notation
    - A negative exponent in 2's complement looks like a large exponent.
    - A fixed value is subtracted from the exponent field to get the true exponent.
    - $E = e + (2^{k-1} - 1)$
      - e is the actual exponent
      - k is the number of bits in the exponent
    - Note
      - When the bits of a biased representation are treated as unsigned integers, the relative magnitudes of the numbers do not change.

Decimal Representation	2’s complement representation	Biased Representation
+8	-	1111
+7	0111	1110
+6	0110	1101
+5	0101	1100
+4	0100	1011
+3	0011	1010
+2	0010	1001
+1	0001	1000
$\pm$0	0000	0111
-1	1111	0110
-2	1110	0101
-3	1101	0100
-4	1100	0011
-5	1011	0010
-6	1010	0001
-7	1001	0000
-8	1000	-

Floating-point Representation (4)

- Normalization
  - By convention, the number which decimal point is placed to the right of the first (nonzero) significant digit is called to be normalized.
    - 1.0×10^{-9} √ (a normalized scientific notation)
    - 0.1×10^{-10} $\times$
  - In normalized binary, the most significant bit of the mantissa is always equal to 1.
    - $\pm 0.1bbb\ldots b \times 2^{E}$ (b is either 0 or 1)
    - Example
      - 0.0110×2^{6} $\times$
      - 0.110×2^{5} √

IEEE 754 Standard (1)

- Introduction
  - Institute of Electrical and Electronics Engineers
  - Most common standard for representing floating point numbers.
  - Established in 1985 as uniform standard for floating point arithmetic
  - This standard was developed to facilitate the portability of programs from one processor to another and encourage the development of sophisticated, numerically oriented programs.
  - Supported by all major CPUs

IEEE 754 Standard (2)

- Single Precision Floating-point Number Format

single precision
(32 bits, float in C)

	1	8	23
sign	S	E	M

exponent:
excess 127
binary integer

Mantissa or significand
sign + magnitude, normalized
binary significand w/ hidden
one bit: 1.M

actual exponent is
e = E – 127 (bias)

$0 < E < 255$

$$X = (-1)^S \, 2^{E-127} \, (1.M)$$

Magnitude of numbers that can be represented is in the range:

$$2^{-126} (1.0) \quad \text{to} \quad 2^{127} (2 - 2^{-23})$$

which is approximately:

$$1.8 \times 10^{-38} \quad \text{to} \quad 3.40 \times 10^{38}$$

IEEE 754 Standard (3)

- Double Precision Floating-point Number Format

double precision
(64 bits, double in C)

	1	11	52
sign	S	E	M

exponent:
excess 1023
binary integer

Mantissa or significand
sign + magnitude, normalized
binary significand w/ hidden
one bit: 1.M

actual exponent is
e = E – 1023 (bias)

$0 < E < 2047$

$X = (-1)^{S}\, 2^{E-1023} (1.M)$

Magnitude of numbers that can be represented is in the range:

$2^{-1022}(1.0)$ to $2^{1023}(2 - 2^{-52})$

which is approximately:

2.2×10^{-308} to 1.8×10^{308}

IEEE 754 Standard (4)

- Special Values
  - Zero
    - S = 0/1, E = 0, M = 0 (0.M)   Value = $\pm$ 0
    - An exponent field of zero is special; it indicates that there is no implicit leading 1 on the mantissa.
  - Infinity
    - Operation that overflows
      - E.g., 1.0/0.0 = 1.0/0.0 = +infinity
    - S = 0/1, E = 255 or 2047, M = 0   Value = $\pm$infinity
  - NaN (Not a Number)
    - Represents case when no numeric value can be determined
      - E.g., sqrt(–1),
    - S = 0/1, E = 255 or 2047, M $\neq$ 0 Value = NaN

IEEE 754 Standard (5)

- Special Values (ctd.)
  - Denormal Numbers
    - There is no implied 1 to the left of the binary point.
    - All denormalized numbers are assumed to have an exponent field of 1 – bias.
    - Numbers very close to 0.0
    - Note that we <u>cannot</u> normalize this value.
    - Zero is effectively a denormal number.
    - Lose precision as get smaller
    - "Gradual underflow"
    - S = 0/1, E = 0, M $\neq$ 0
      - Value = $\pm$ 0.M $\times 2^{-126}$
      - Value = $\pm$ 0.M $\times 2^{-1022}$

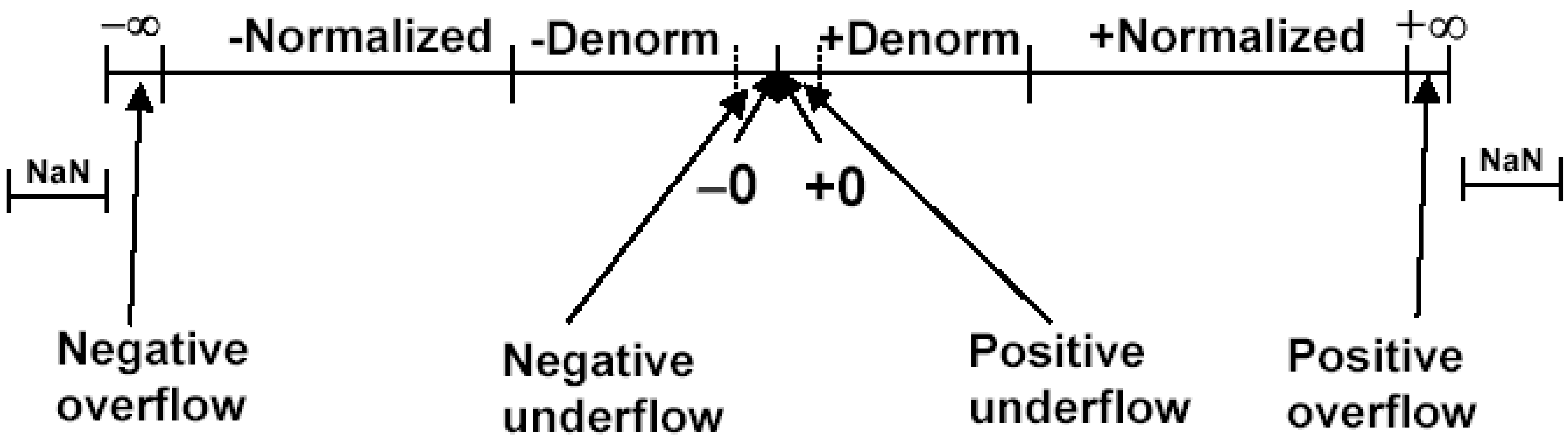
IEEE 754 Standard (6)

- Special Values Summary

Normalized:	±	0<E<max	Any bit pattern
Denormalized:	±	0	Any nonzero bit pattern
zero:	±	0	0
Infinity:	±	11...1	0
NaN:	±	11...1	Any nonzero bit pattern

IEEE 754 Standard (7)

- Special Values Summary (ctd.)



IEEE 754 Standard (8)

- Summary
  - A computer must provide at least single-precision representation to conform to the IEEE standard.
  - Double-precision representation is optional.
  - Extended single-precision (more than 32 bits) /Extended double-precision (more than 64 bits)
    - Help to reduce the size of the accumulated round-off error in a sequence of calculations.
    - Enhance the accuracy of evaluation of elementary functions such as sine, cosine, and so on.
  - Trade-off between “accuracy” and “range”
    - Increasing the size of **mantissa** enhances **accuracy.**
    - Increasing the size of **exponent** increases the **range.**

Quiz (1)

1. In IEEE754 standard for representing floating-point numbers of 32 bits, the sign of the number is given 1 bit, the exponent of the scale factor is allocated 8 bits, and the mantissa is assigned 23 bits. What is the maximum normalized positive number that 32-bit representation can represent?

A. $+(2-2^{-23})\times 2^{+127}$

B. $+(1-2^{-23})\times 2^{+127}$

C. $+(2-2^{-23})\times 2^{+255}$

D. $2^{+127}-2^{-23}$

最大的正单精度浮点数符号必为0，尾数部分取23个1，故为1.111…1，

指数部分取E=254，实际的指数e=127，

所以是$1.111\ldots1\times 2^{+127}=+(2-2^{-23})\times 2^{+127}$

Quiz (2)

2. In single-precision format of IEEE 754 floating point number standard, instead of the signed exponent E, what is the value actually stored in the exponent field?

A. E=e+255 B. E=e+127

C. E=e+256 D. E=e+128

IEEE 754标准规定单精度浮点数的指数部分占8位，$E=e+(2^{8-1}-1)$

Quiz (3)

3. In double-precision format of IEEE 754 floating point number standard, instead of the signed exponent e, what is the value E actually stored in the exponent field?

   A. E=e+2047　　　　B. E=e+1023

   C. E=e+2048　　　　D. E=e+1024

IEEE 754标准规定双精度浮点数的指数部分占11位，$E=e+(2^{11-1}-1)$

Quiz (4)

4. *True or False?* A computer must provide at least single-precision representation to conform to the IEEE standard.

   IEEE 754标准规定至少要支持单精度的浮点数格式

Quiz (5)

5. Using 32-bit IEEE 754 single precision floating point format, show the representation of -0.6875.

Solution:

$0.6875 = 0.1011 \times 2^{0} = 1.011 \times 2^{-1}$

M = 01100000000000000000000

E = e + 127 = -1 + 127 = +126, 表示为: 01111110

所以-0.6875表示为: 1 01111110 01100000000000000000000