BIỂU DIỄN SỐ NGUYÊN

Hệ cơ số q tổng quát

- Tổng quát số nguyên có n chữ số thuộc hệ cơ số q bất kỳ được biểu diễn.

$$\mathbf{X = a_{n-1} \ldots a_1 a_0 = a_{n-1} q^{n-1} + \ldots + a_1 q^1 + a_0 q^0}$$

- Trong đó:
  - q = 2, X = {0, 1}: hệ nhị phân (binary)
  - q = 8, X = {0, 1, 2,…, 7}: hệ bát phân (octal)
  - q = 10, X = {0, 1, 2,…, 9}: hệ thập phân (decimal)
  - q = 16, X = {0, 1, 2,…,9, A, B,…, F}: hệ thập lục phân (hexadecimal)

Hệ nhị phân

$$\mathbf{X= a_{n-1} \ldots a_1 a_0 = a_{n-1} 2^{n-1} + \ldots + a_1 2^1 + a_0 2^0}$$

- Được dùng nhiều trong máy tính để biểu diện các giá trị lưu trong các thanh ghi hoặc trong các ô nhớ. Thanh ghi hoặc ô nhớ có kích thước 1 byte (8 bit) hoặc 1 word (16 bit).
- n được gọi là chiều dài bit của số đó
- Bit trái nhất $\mathbf{a_{n-1}}$ là bit có giá trị (nặng) nhất MSB (Most Significant Bit)
- Bit phải nhất $\mathbf{a_0}$ là bit ít giá trị (nhẹ) nhất LSB (Less Significant Bit)

Chuyển đổi giữa các hệ cơ số

- Decimal (10) → Binary (2)
- Decimal (10) → Hexadecimal(16)
- Binary (2) → Decimal (16)
- Decimal (10) → Hexadecimal(16)
- Hexadecimal(16) → Binary(2)
- Hexadecimal(16) → Decimal(10)

Số nguyên không dấu

- Biểu diễn các đại lương luôn dương
  - Ví dụ: chiều cao, cân nặng, mã ASCII…
- Tất cả bit đều được sử dụng để biểu diễn giá trị (không quan tâm đến dấu âm, dương)
- Số nguyên không dấu 1 byte lớn nhất là:

  $1111\ 1111_2 = 2^8 - 1 = 255_{10}$
- Số nguyên không dấu 1 word lớn nhất là:

  $1111\ 1111\ 1111\ 1111_2 = 2^{16} - 1 = 65535_{10}$
- Tùy nhu cầu có thể sử dụng số 2, 3… word.
- LSB = 1 thì số đó là số đó là số lẻ

Số nguyên có dấu

- Lưu các nguyên âm hoặc dương
- Có 4 cách phổ biến:
  - Dấu lượng
  - Bù 1
  - Bù 2
  - Số quá (thừa) K

Biểu diễn số nguyên có dấu-Dấu lượng

- Bit trái nhất (MSB): bit đánh dấu âm / dương
  - 0: số dương
  - 1: số âm
- Các bit còn lại: biểu diễn độ lớn của số (hay giá trị tuyệt đối của số)

Biểu diễn số nguyên có dấu-Bù 1

- Bit MSB dùng làm bit dấu
  - 0: Số dương
  - 1: Số âm
- Các bit còn lại dùng làm độ lớn
- Số âm: Thực hiện phép đảo bit tất cả các bit không phải bit dấu

Biểu diễn số nguyên có dấu-Bù 2

- Biểu diễn giống như số bù 1 và cộng thêm số 1 vào kết quả (dạng nhị phân)
- Số bù 2 khắc phục được vấn đề đối với hai phương pháp dấu lượng và bù 1:
  - Có hai cách biểu diễn cho số 0 (+0 và -0)
  - Bit nhớ phát sinh sau khi đã thực hiện phép tính phải được cộng tiếp vào kết quả, dễ gây nhầm lẫn

Biểu diễn số nguyên có dấu-Quá (thừa) K

- Còn gọi là biểu diễn số dịch (biased representation)
- Chọn một số nguyên dương K cho trước làm giá trị dịch
- Biểu diễn số N:
  - +N (dương): lấy K + N, với K được chọn sao cho tổng của K và một số âm bất kỳ trong miền giá trị luôn luôn dương
  - -N (âm): có được bằng cách lấy K - N
- Ví dụ: dùng 8 bit, chọn K=128 (biễu diễn từ -128 đến +127)
  - Biểu diễn N=27: 27+128=155=10011011_2
  - Biểu diễn N=-27: -27+128=101=64+32+4+1=01100101_2

Các phép toán

- Phép cộng (+)
- Phép trừ (-)
- Phép nhân (*)
- Phép chia (/)
- Phép dịch, quay:
  - Dịch trái (SHL), dịch phải (SHR)
  - Quay trái (ROL), quay phải (ROR)
- Phép logic: AND, OR, NOT, XOR

Bài tập lớn

- **Tìm hiểu Giải thuật nhân số nguyên nhị phân**
- Tham khảo: *Chapter 10 (Phần 10.2: Interger Arithmetic-Multiplication), sách William Stallings*

BIỂU DIỄN SỐ THỰC

Đặt vấn đề

- Biểu diễn số 123.625_{10} ở hệ nhị phân?
  - Ý tưởng: Biểu diễn phần nguyên và phần thập phân riêng lẻ
    - Phần nguyên: Dùng 8 bit
      $123_{10} = 64 + 32 + 16 + 8 + 2 + 1 = 0111\ 1011_2$
    - Phần thập phân: Dùng 8 bit
      $0.625_{10} = 0.5 + 0.125 = 2^{-1} + 2^{-3} = 1010\ 0000_2$
  - Vậy: $123.625_{10} = 0111\ 1011.1010\ 0000_2$

  Tổng quát công thức chuyển số thực biểu diễn ở hệ nhị phân thành thập phân:

$$\boxed{x_{n-1}x_{n-2}\ldots x_0.x_{-1}x_{-2}\ldots x_{-m} = x_{n-1}.2^{n-1}+x_{n-2}.2^{n-2}\ldots + x_0.2^0+x_{-1}.2^{-1}+x_{-2}.2^{-2}+\ldots+x_{-m}.2^{-m}}$$

Nhận xét

- Phạm vi biểu diễn tùy thuộc vào số bit sử dụng
- Ví dụ sử dụng 8 bit: số nhỏ nhất biểu diễn được là 2^{-8}
- Đối với các số nhỏ hơn, chẳng hạn 0.00001(10^{-5})?

→ Tăng số bit

- Giải pháp hiệu quả hơn

Floating Point Number
(Số thực dấu chấm động)

Biểu diễn số thực dấu chấm động

- Tổng quát:

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

  - S: phần định trị không dâu (Significand)
  - E: phần số mũ (Exponent)
  - B: cơ số (Base) – không cần lưu
- Đối với hệ nhị phân:

$$\pm 1.F * 2E$$

Sign	Exponent(biased)	Significant
1 bit	8 bit	23 bit

Số thực dấu chấm động 32 bit

Ví dụ

- Biểu diễn số thực sau theo dạng số chấm động 32 bit:
  - $X = -5.75$
- *Bước 1*: Đổi X sang hệ nhị phân
  - $X = -5.75_{10} = -101.11_2$
- *Bước 2*: Chuẩn hóa theo dạng $\pm 1.F * 2^E$
  - $X = -5.75 = -101.11 = -1.0111 * 2^2$
- *Bước 3*: Biểu diễn Floating Point
  - Số âm: bit dấu Sign = 1
  - Số mũ E = 2, phần mũ exponent với số thừa K=127 được biểu diễn:
    - Exponent = E + 127 = 2 + 127 = $129_{10} = 1000\ 0001_2$
  - Phần định trị: 0111 0000 0000 0000 0000 000 (Thêm 19 số 0 cho đủ 23 bit)
- **Kết quả: 1 1000 0001 0111 0000 0000 0000 0000 000**

Chuẩn IEEE 754

Parameter	Format		
	Binary32	Binary64	Binary128
Storage width (bits)	32	64	128
Exponent width (bits)	8	11	15
Exponent bias	127	1023	16383
Maximum exponent	127	1023	16383
Minimum exponent	−126	−1022	−16382
Approx normal number range (base 10)	$10^{-38}, 10^{+38}$	$10^{-308}, 10^{+308}$	$10^{-4932}, 10^{+4932}$
Trailing significand width (bits)*	23	52	112
Number of exponents	254	2046	32766
Number of fractions	2^{23}	2^{52}	2^{112}
Number of values	1.98×2^{31}	1.99×2^{63}	1.99×2^{128}
Smallest positive normal number	2^{-126}	2^{-1022}	2^{-16362}
Largest positive normal number	$2^{128} - 2^{104}$	$2^{1024} - 2^{971}$	$2^{16384} - 2^{16271}$
Smallest subnormal magnitude	2^{-149}	2^{-1074}	2^{-16494}

Bài tập

- Biểu diễn nhị phân số thực dấu chấm động 32 bit
  a) +10.675
  b) -1080