

# CTT009

## Lưu trữ dữ liệu

Lê Thị Nhân  
[lt Nhan@fit.hcmus.edu.vn](mailto:lt Nhan@fit.hcmus.edu.vn)



KHOA CÔNG NGHỆ THÔNG TIN  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

# Nội dung

- ☐ Nhắc lại
- ☐ Biểu diễn thông tin
  - ☐ Văn bản, số, hình ảnh, âm thanh
- ☐ Lưu trữ số nguyên
- ☐ Lưu trữ phân số



# Nhắc lại

- ☐ Chuỗi bits
- ☐ Phép toán Boolean
- ☐ Flip-flop
- ☐ Ô nhớ và địa chỉ
- ☐ Volatile memory
- ☐ Non-volatile memory



# BIỂU DIỄN THÔNG TIN



# Biểu diễn văn bản

- ☐ Từng *ký tự* được biểu diễn bởi 1 chuỗi các bits duy nhất
  - ☐ Chữ cái, dấu chấm câu, ...
  
- ☐ Tiêu chuẩn mã hóa
  - ☐ ASCII
  - ☐ ISO
  - ☐ Unicode



# Biểu diễn văn bản

## □ **ASCII** - American Standard Code for Information Interchange

□ Sử dụng tổ hợp 7 bits để biểu diễn các ký tự được sử dụng trong văn bản tiếng Anh

- 0 ~ 31 & 127: mã điều khiển thiết bị
- 32 ~ 126: mã ký hiệu thông thường
  - (A..Z, a..z, 0..9, ? ! ...)

□ Ví dụ

- A:65    B:66    ...
- a:97    b:98    ...

# Biểu diễn văn bản

## □ Chữ “Hello” trong ASCII

01001000

**H**

01100101

**e**

01101100

**l**

01101100

**l**

01101111

**o**

00101110

**.**

# Biểu diễn văn bản

## □ **ISO**

- Tổ chức ISO phát triển nhiều mở rộng cho bảng mã ASCII, sử dụng tổ hợp 8 bits
  - 128 ~ 255 : mã ký hiệu mở rộng ( $\alpha$ ,  $\beta$ ,  $\gamma$ , ...)
- Mỗi mở rộng là một nhóm ngôn ngữ chính
  - ISO Latin-1 (ISO 8859-1)

## □ **Unicode**

- Sử dụng tổ hợp 16 bits để biểu diễn các ký tự được sử dụng trong các ngôn ngữ trên toàn thế giới (UTF-8, UTF-16)



# Biểu diễn giá trị số

- Ký hiệu nhị phân
  - Sử dụng các bits để biểu diễn một số trong hệ cơ số 2
- Hạn chế của biểu diễn số trong máy tính
  - Tràn số trên (overflow)
    - Xảy ra khi biểu diễn 1 số quá lớn
  - Tràn số dưới (underflow)
    - Xảy ra khi biểu diễn 1 số quá nhỏ
  - Cắt bỏ (truncation)
    - Xảy ra khi không thể biểu diễn chính xác một giá trị (một số nằm giữa 2 cách biểu diễn)



# Biểu diễn hình ảnh

## □ Kỹ thuật bit map

- Điểm ảnh (pixel : picture element)
- Điểm ảnh được mã hóa
  - RGB
  - Cường độ sáng (luminance) và độ đậm/nhạt của màu (chrominance)

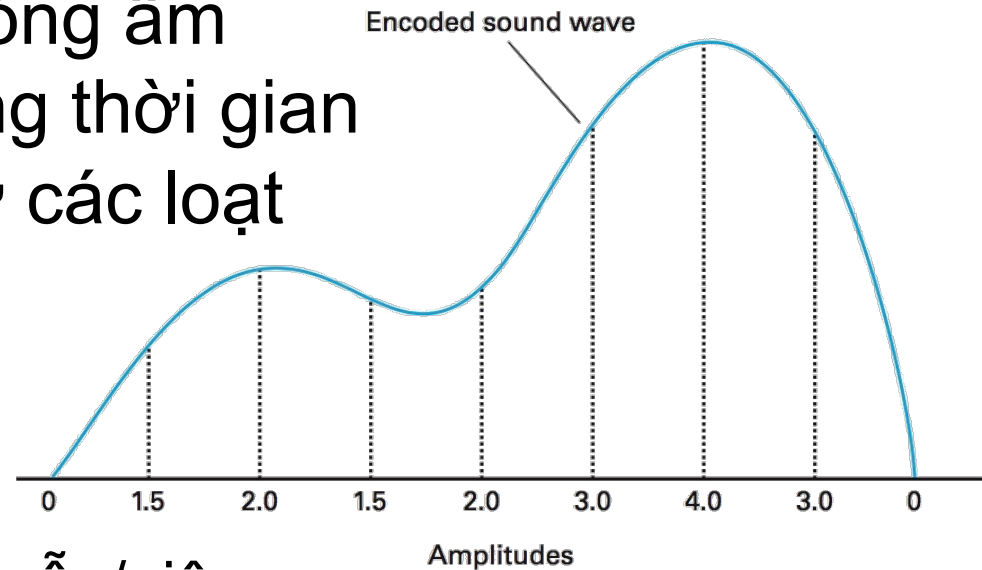
## □ Kỹ thuật vector

- Cấu trúc hình học (đường kẻ, đường cong...)
- Có thể thay đổi kích thước (phóng to, thu nhỏ) mà không bị biến dạng
  - TrueType và PostScript

# Biểu diễn âm thanh

## □ Kỹ thuật lấy mẫu (sampling)

- Lấy biên độ của sóng âm thanh ở các khoảng thời gian đều đặn và lưu trữ các loạt giá trị này



## □ Tỷ lệ mẫu

- Điện thoại : 8,000 mẫu/giây
- Nhạc : 44,100 mẫu/giây
  - Độ trung thực cao, 16 bits/mẫu
  - Nhạc nổi (stereo), 32 bits/mẫu

Nguồn: Computer Science - An Overview, 12e

# Biểu diễn âm thanh

## □ MIDI - Musical Instrument Digital Interface

□ Mã hóa dụng cụ nào đang chơi nốt nhạc nào trong khoảng thời gian nào

□ Ví dụ

- Kèn clarinet chơi nốt rê (note D) trong 2 giây
- Được mã hóa bởi 3 bytes



# LƯU TRỮ SỐ NGUYÊN



# Giới thiệu

- ☐ Giá trị số được biểu diễn bởi ký hiệu nhị phân
- ☐ Ngoài ra còn có
  - ☐ Ký hiệu số bù 2 (two's complement)
  - ☐ Ký hiệu số vượt quá (excess)



# Ký hiệu số bù 2

□ Bit trái nhất ~ bit dấu

□ 1 : âm

□ 0 : dương

□ Ví dụ

(4) 0100 → 1011 → 1100 (-4)

# Ký hiệu số bù 2

**a. Using patterns of length three**

Bit pattern	Value represented
011	3
010	2
001	1
000	0
111	-1
110	-2
101	-3
100	-4

**b. Using patterns of length four**

Bit pattern	Value represented
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	-1
1110	-2
1101	-3
1100	-4
1011	-5
1010	-6
1001	-7
1000	-8

Nguồn: Computer Science - An Overview, 12e



# Ký hiệu số bù 2

- Mỗi liên hệ giữa hai số có cùng độ lớn
  - Chuỗi bits giống nhau từ phải qua trái cho đến khi gặp bit 1 đầu tiên
  - Phần chuỗi còn lại của số này là *chuỗi đối* của số kia
    - Chuỗi đảo bits từ 0 sang 1, từ 1 sang 0



# Ký hiệu số bù 2

Two's complement notation for 6 using four bits

[0 1 1 0]

Copy the bits from right to left until a 1 has been copied

Complement the remaining bits

Two's complement notation for -6 using four bits

[1 0 1 0]

# Ký hiệu số bù 2 - Phép cộng

Problem in base 10		Problem in two's complement		Answer in base 10
$\begin{array}{r} 3 \\ + 2 \\ \hline \end{array}$	→	$\begin{array}{r} 0011 \\ + 0010 \\ \hline 0101 \end{array}$	→	5
$\begin{array}{r} -3 \\ + -2 \\ \hline \end{array}$	→	$\begin{array}{r} 1101 \\ + 1110 \\ \hline 1011 \end{array}$	→	-5
$\begin{array}{r} 7 \\ + -5 \\ \hline \end{array}$	→	$\begin{array}{r} 0111 \\ + 1011 \\ \hline 0010 \end{array}$	→	2

# Quiz

☐ Nếu dùng 4 bits để lưu số không dấu

☐ Ta lưu được các số nào?

☐ Nếu dùng 4 bits để lưu số có dấu

☐ Ta lưu được các số nào?



# Tóm lại

## □ Lưu trữ số nguyên

	8 bits	16 bits	32 bits	64 bits
<b>UMax</b>	255	65,535	4,294,967,295	18,446,744,073,709,551,615
<b>TMax</b>	127	32,767	2,147,483,647	9,223,372,036,854,775,807
<b>TMin</b>	-128	-32,768	-2,147,483,648	-9,223,372,036,854,775,808

- UMax: giá trị lớn nhất của số không dấu
- TMax: giá trị lớn nhất của số có dấu
- TMin: giá trị nhỏ nhất của số có dấu



# Ký hiệu số vượt quá

*Bù 2*

Bit pattern	Value represented
011	3
010	2
001	1
000	0
111	-1
110	-2
101	-3
100	-4

*Vượt quá*

Bit pattern	Value represented
111	3
110	2
101	1
100	0
011	-1
010	-2
001	-3
000	-4

# Ký hiệu số vượt quá

*Bù 2*

Bit pattern	Value represented
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	-1
1110	-2
1101	-3
1100	-4
1011	-5
1010	-6
1001	-7
1000	-8

*Vượt quá*

Bit pattern	Value represented
1111	7
1110	6
1101	5
1100	4
1011	3
1010	2
1001	1
1000	0
0111	-1
0110	-2
0101	-3
0100	-4
0011	-5
0010	-6
0001	-7
0000	-8

*Nguồn: Computer Science -  
An Overview, 12e*

# Ký hiệu số vượt quá

*Thập phân*

Binary	Decimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

*Vượt quá*

Bit pattern	Value represented
1111	7
1110	6
1101	5
1100	4
1011	3
1010	2
1001	1
1000	0
0111	-1
0110	-2
0101	-3
0100	-4
0011	-5
0010	-6
0001	-7
0000	-8

*Nguồn: Computer Science -  
An Overview, 12e*



# Ký hiệu số vượt quá

- Giả sử cố định chiều dài chuỗi bits là 4
  - 0000 ở hệ 10, biểu diễn giá trị 0
  - 0000 ở hệ vượt quá, biểu diễn giá trị -8
  - 1100 ở hệ 10, biểu diễn giá trị 12
  - 1100 ở vượt quá, biểu diễn giá trị 4
- Vượt quá 8 ( $2^{4-1}$ )



# Ký hiệu số vượt quá

- ☐ Giả sử cố định chiều dài chuỗi bits là 5
- ☐ Vượt quá là bao nhiêu?
- ☐ Tổng quát
  - ☐ Ký hiệu vượt quá  $2^{N-1}$
  - ☐ Ký hiệu vượt quá N-bit
  - ☐ Với N là chiều dài chuỗi bits



# LƯU TRỮ PHÂN SỐ



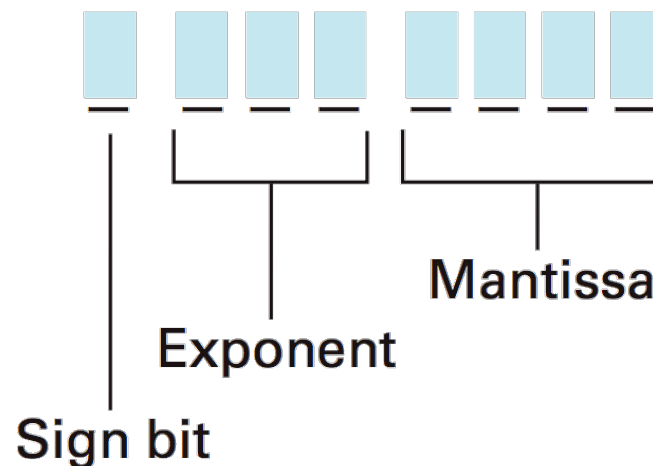
# Giới thiệu

- Phân số không chỉ lưu chuỗi bits mà còn lưu vị trí của ***radix point***
  - Ký hiệu dùng để tách phần nguyên (bên trái) và phần phân số (bên phải) của 1 số
- Ký hiệu dấu chấm động (floating-point)



# Dấu chấm động

- Giả sử có 8 bits
  - Bit trái nhất là ***bit dấu*** (sign bit)
    - 0 (dương) và 1 (âm)
  - 3 bits tiếp theo là ***phần mũ*** (exponent)
  - 4 bits sau cùng là ***phần định trị*** (mantissa)



Nguồn: Computer Science - An Overview, 12e

# Ví dụ 1

□ Chuỗi bits 01101011

□ Bit dấu : 0

□ Phần mũ : 110

□ Phần định trị : 1011

□ .1011

□  $110 = 2$  (hệ vượt quá 3-bit)

□  $10.11 = 2 \frac{3}{4}$

□  $01101011 = 2 \frac{3}{4}$

## Ví dụ 2

□ Chuỗi bits 00111100

□ Bit dấu : 0

□ Phần mũ : 011

□ Phần định trị : 1100

□ .1100

□ 011 = -1 (hệ vượt quá 3-bit)

□ .01100 = 3/8

□ 00111100 = 3/8

## Ví dụ 3

### □ Mã hóa $1\frac{1}{8}$

□ Chuyển sang hệ nhị phân: 1.001

□ Phần định trị:             1 0 0 1

■ Từ trái sang phải, bắt đầu với bit 1 trái nhất

□ Radix point: từ .1001 thành 1.001  $\rightarrow +1$

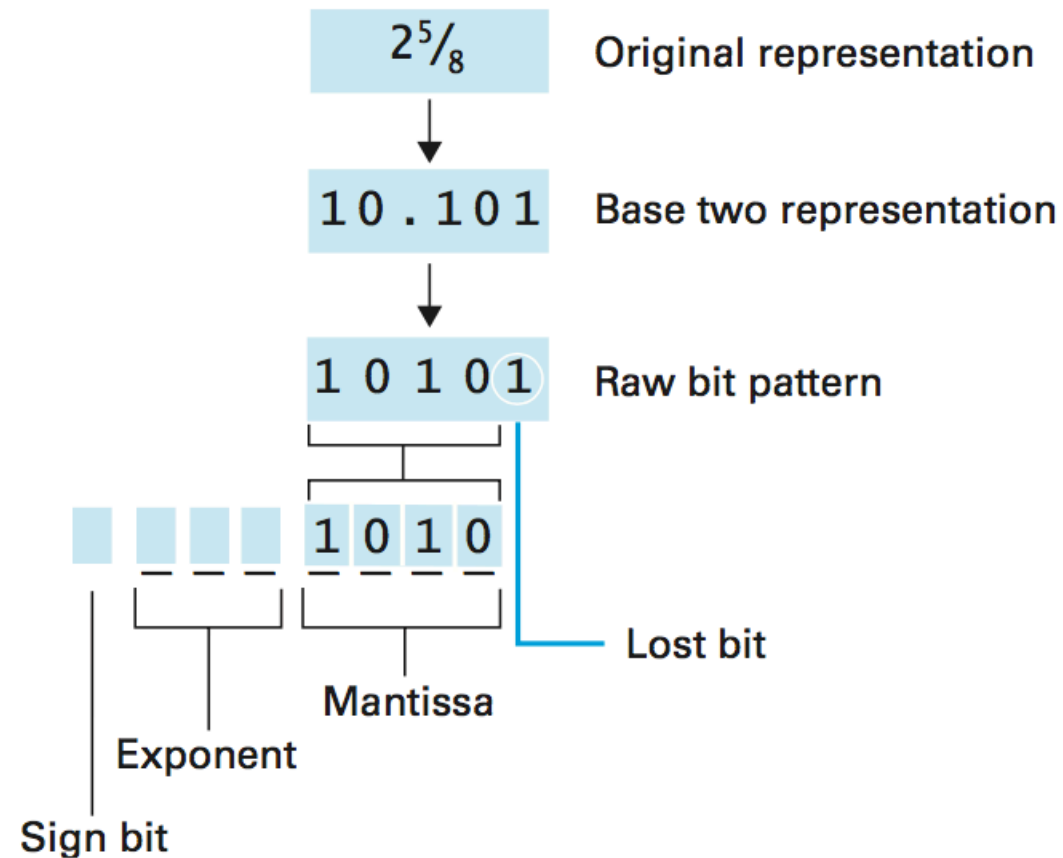
□ Phần mũ:  $+1 = 101$  (hệ vượt quá 3-bit)

□ Bit dấu: 0

□  $1\frac{1}{8} = 01011001$

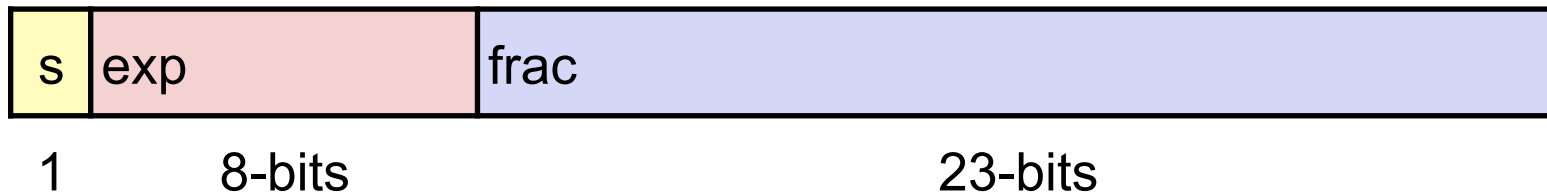


# Lỗi cắt bỏ



# Đề biểu diễn chính xác

☐ Chính xác đơn : 32 bits



☐ Chính xác kép : 64 bits



☐ Chính xác mở rộng : 80 bits (Intel only)



# Chính xác đơn 32-bit

□ Chuẩn IEEE-754

$$N = (-1)^S \times 1.F \times 2^{(E-127)}$$

- S : bit dấu
- F : fractional part
- E : exponent part
- $127 = 2^8 - 1$  (vượt quá 8-bit)

# Chính xác đơn 32-bit

□ Ví dụ

□ 0 10000000 110 0000 0000 0000 0000 0000

Sign bit  $S = 0 \Rightarrow$  positive number

$E = 1000\ 0000B = 128D$

Fraction is  $1.11B$  (with an implicit leading 1)  $= 1 + 1 \times 2^{-1} + 1 \times 2^{-2} = 1.75D$

The number is  $+1.75 \times 2^{(128-127)} = +3.5D$

# Chính xác đơn 32-bit

□ Ví dụ

□ 1 01111110 100 0000 0000 0000 0000 0000

Sign bit  $S = 1 \Rightarrow$  negative number

$E = 0111\ 1110_B = 126_D$

Fraction is  $1.1_B$  (with an implicit leading 1)  $= 1 + 2^{-1} = 1.5_D$

The number is  $-1.5 \times 2^{(126-127)} = -0.75_D$



# TÓM TẮT



# Bài giảng hôm nay

- ☐ Biểu diễn thông tin
  - ☐ Ý tưởng làm sao máy tính lưu được văn bản, số, hình ảnh, âm thanh
  - ☐ Tên gọi của các kỹ thuật
  
- ☐ Lưu trữ số nguyên
- ☐ Lưu trữ phân số

# Bài giảng tuần tới

- ☐ Lưu trữ dữ liệu (chapter 1)
  - ☐ Nén dữ liệu
  - ☐ Lỗi giao tiếp
  - ☐ Hệ thống tập tin





