

# NHẬP MÔN TIN HỌC

---

## Chương 4

# BIỂU DIỄN DỮ LIỆU

# Tài liệu tham khảo

- J. Glenn Brookshear, ***Computer Science: An Overview***, Pearson, 2015.
- N. Dell, J. Lewis, ***Computer Science Illuminated (6<sup>th</sup> Edition)***, Jones & Bartlett Learning, 2016.
- Tập bài giảng Nhập môn tin học – Ninh Xuân Hương – ĐH Mở Tp HCM

# Nội dung chương 4

---

**I. Khái niệm về biểu diễn dữ liệu**

**II. Biểu diễn số nguyên**

**III. Biểu diễn số thực**

**IV. Biểu diễn ký tự**

**V. Dữ liệu âm thanh, hình ảnh**



# I. Khái niệm về biểu diễn dữ liệu

---

1. Dữ liệu trên máy tính
2. Hệ đếm theo vị trí
3. Các hệ đếm thông dụng
4. Chuyển đổi giữa các hệ đếm
5. Các phép toán

# 1. Dữ liệu trên máy tính

- Máy tính là thiết bị đa phương tiện (multimedia device)
- Máy tính lưu trữ, xử lý, hiển thị các dạng dữ liệu:
  - Giá trị số (numbers)
  - Văn bản (text)
  - Hình ảnh (images, graphics)
  - Âm thanh (audio)
  - Hình ảnh động (video)

# Nén dữ liệu (Data Compression)

- Mục tiêu: giảm kích thước lưu trữ dữ liệu
  - Tỷ số nén (Compression ratio)
- Có hai kỹ thuật chính:
  - Nén không mất dữ liệu (lossless): dữ liệu có thể được phục hồi nguyên vẹn từ dữ liệu nén
  - Nén có mất dữ liệu (lossy): có một phần dữ liệu bị mất khi nén

# Dữ liệu dạng nhị phân

- Máy tính được thiết kế để sử dụng dữ liệu dạng nhị phân:
  - Giá thành thấp
  - Độ tin cậy cao
- BIT (Binary digiT): chữ số nhị phân

## 2. Hệ đếm theo vị trí

---

- Hệ đếm là tập các ký hiệu và quy tắc nhằm biểu diễn các giá trị số.



## 2. Hệ đếm theo vị trí

Xét ví dụ

**642 trong hệ đếm theo vị trí cơ số 10:**

$$\begin{array}{rcl} 6 \times 10^2 & = & 6 \times 100 = 600 \\ + 4 \times 10^1 & = & 4 \times 10 = 40 \\ + 2 \times 10^0 & = & 2 \times 1 = 2 \\ & & = 642_{10} \\ & & \text{(642 trong hệ đếm 10)} \end{array}$$

Hệ đếm  
cơ số 10

Số mũ thể hiện  
vị trí của chữ số

# Biểu thức tổng quát hệ đếm theo vị trí

$$X = d_{n-1}d_{n-2}\dots d_1d_0$$

R là cơ số  
của hệ đếm

$$d_{n-1} * R^{n-1} + d_{n-2} * R^{n-2} + \dots + d_1 * R^1 + d_0$$

n là số chữ số  
trong giá trị X

$d_i$  là chữ số  
tại vị trí thứ i  
trong giá trị X

$$642 = 6 * 10^2 + 4 * 10 + 2$$

Ví dụ: 642 trong hệ đếm 12 ( $642_{12}$ )

$$\begin{aligned} 642_{12} &= 6 \cdot 12^2 + 4 \cdot 12 + 2 \\ &= 864 + 48 + 2 \\ &= 914_{10} \end{aligned}$$

### 3. Các hệ đếm thông dụng

- Hệ thập phân (decimal) dùng 10 chữ số:
  - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Hệ nhị phân (binary) dùng 2 chữ số:
  - 0, 1
- Hệ bát phân (octal) dùng 8 chữ số:
  - 0, 1, 2, 3, 4, 5, 6, 7
- Hệ thập lục phân (hexadecimal) dùng 16 chữ số:
  - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

# Một số giá trị cần nhớ

Dec	Bin	Oct	Hex
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7

Dec	Bin	Oct	Hex
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

# Một số giá trị cần nhớ (tt)

	<b>Dec</b>
$2^{-2}$	0.25
$2^{-1}$	0.5
$2^0$	1
$2^1$	2
$2^2$	4
$2^3$	8
$2^4$	16
$2^5$	32

	<b>Dec</b>
$2^6$	64
$2^7$	128
$2^8$	256
$2^9$	512
$2^{10}$	1024
$2^{11}$	2048
$2^{12}$	4096
$2^{20}$	1048576

## 4. Chuyển đổi giữa các hệ đếm

- Chuyển từ hệ 2, hệ 8, hệ 16 sang hệ 10
- Chuyển đổi giữa hệ 2 và hệ 8
- Chuyển đổi giữa hệ 2 và hệ 16
- Chuyển từ hệ 10 sang hệ 2
- Chuyển từ hệ 10 sang hệ đếm cơ số K

# Chuyển từ hệ 2, hệ 8, hệ 16 sang hệ 10

■ Dùng định nghĩa hệ đếm theo vị trí

■ Ví dụ

$$\bullet \mathbf{10101_2} = 2^4 + 2^2 + 1 = 21_{10}$$

$$\bullet \mathbf{345_8} = 3*8^2 + 4*8 + 5 = 229_{10}$$

$$\bullet \mathbf{AB1_{16}} = 10*16^2 + 11*16 + 1 = 2737_{10}$$



# Chuyển đổi giữa hệ 2 và hệ 8

■ Ba bit (nhị phân) tương đương 1 chữ số hệ 8

■ Ví dụ:

$$\bullet \ 101001110 = \underline{101} \ \underline{001} \ \underline{110} = 516_8$$

$$\bullet \ 10101 = \underline{10} \ \underline{101} = 25_8$$

$$\bullet \ 175_8 = \underline{001} \ \underline{111} \ \underline{101} = 1111101_2$$

# Chuyển đổi giữa hệ 2 và hệ 16

- Bốn bit (nhị phân) tương đương 1 chữ số hệ 16
- Ví dụ:
  - $10100111 = \underline{1010} \underline{0111} = A7_{16}$
  - $110110 = \underline{11} \underline{0110} = 36_{16}$
  - $175_{16} = \underline{0001} \underline{0111} \underline{0101} = 101110101_2$

# Chuyển từ hệ 10 sang hệ 2

---

Hai phương pháp thông dụng:

- Phân tích thành tổng các lũy thừa của 2
- Thực hiện các phép chia cho 2

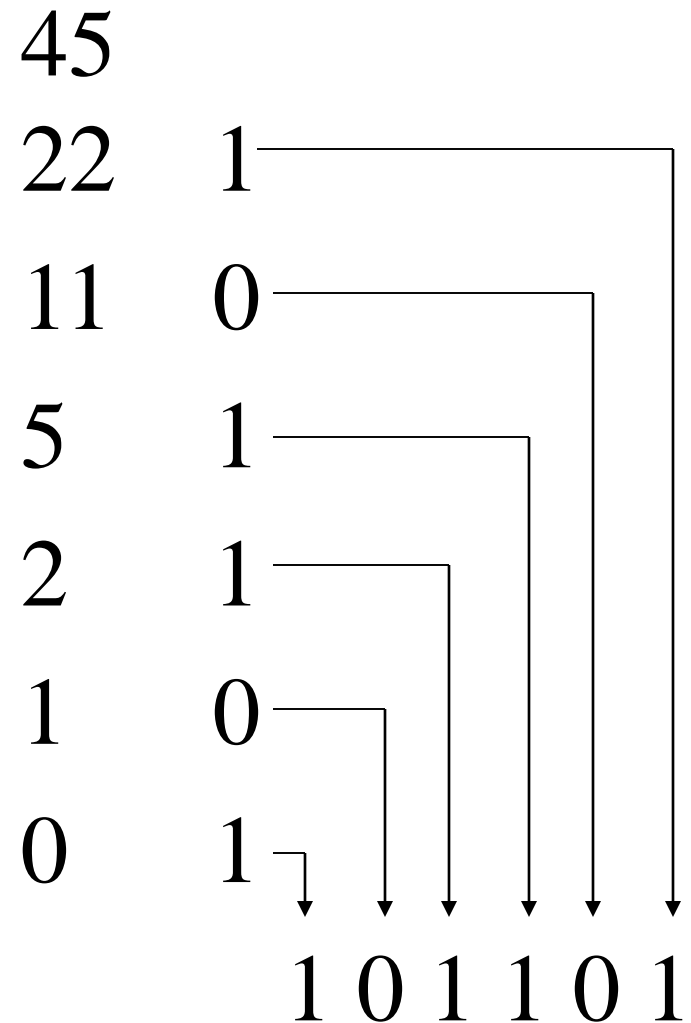
# Phân tích thành tổng các lũy thừa của 2

■ Ví dụ:  $45_{10} \rightarrow ?_2$

$$\begin{aligned} \blacksquare 45 &= 32 + 8 + 4 + 1 \\ &= 2^5 + 2^3 + 2^2 + 2^0 \\ &= 101101_2 \end{aligned}$$

# Thực hiện các phép chia cho 2

■ Ví dụ:  $45_{10} \rightarrow ?_2$

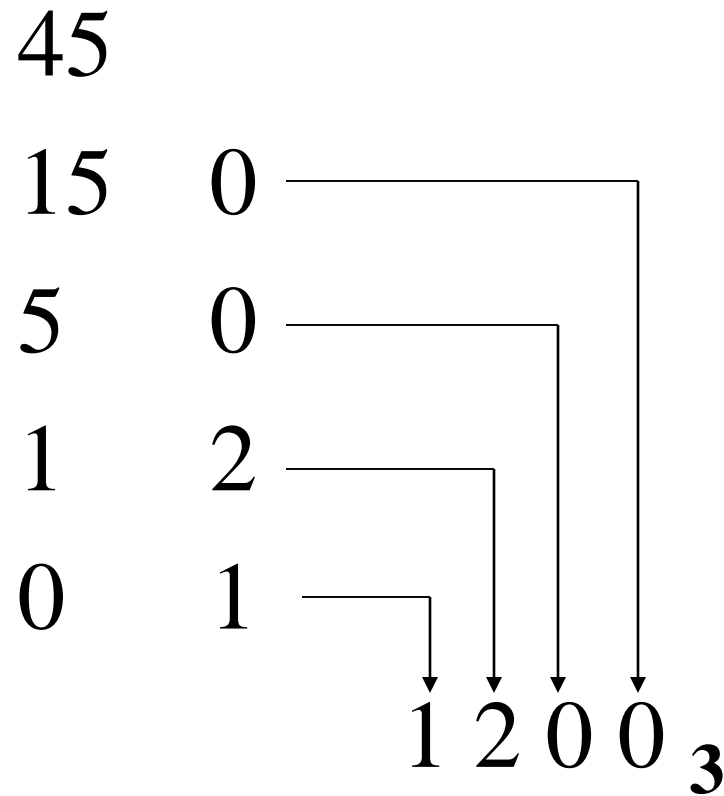


# Chuyển từ hệ 10 sang hệ đếm cơ số K

Giá trị thập phân X

- Thực hiện các phép chia X và các thương số có được cho K, cho đến khi thương số là 0
- Kết quả là các dư số được lấy theo chiều ngược lại (từ đáy lên đỉnh.)

Ví dụ:  $45_{10} \rightarrow ?_3$



# Bài tập 1

---

- Đổi các số thập phân sau đây sang hệ 2, hệ 8, hệ 16:
  - 50, 129, 200, 257, 300
- Đổi các số thập phân sau sang hệ 3, hệ 7:
  - 50, 150



## 5. Các phép toán

- Phép toán số học gồm: cộng, trừ, nhân, chia, chia nguyên, lũy thừa, đồng dư (mod).
- Phép toán so sánh gồm:  $>$ ,  $\geq$ ,  $<$ ,  $\leq$ ,  $=$ ,  $\langle \rangle$
- Phép toán logic gồm: AND, OR, NOT

# Phép toán số học

- Thứ tự ưu tiên thực hiện các phép toán:
  - Ưu tiên cao nhất: phép toán trong cặp dấu ngoặc.
  - Ưu tiên 2: lũy thừa, đồng dư.
  - Ưu tiên 3: Nhân, chia, chia nguyên.
  - Ưu tiên sau cùng: Cộng, trừ.

# Các phép toán số học

- Phép cộng
- Phép trừ
- Phép nhân
- Phép chia

# Các phép toán số học

## ❖ Phép cộng

- Là phép tính làm cơ sở cho các phép tính khác.

# Các phép toán số học

## ❖ Phép cộng

■ Khi thực hiện phép cộng cần lưu ý:

$$0 + 0 = 0 ;$$

$$0 + 1 = 1 ;$$

$$1 + 1 = 0 \text{ nhớ } 1 \text{ (đem qua bit cao hơn).}$$

# Các phép toán số học

## ❖ Phép cộng

### ■ Ví dụ:

$$\begin{array}{r} 1001 \\ + 0101 \\ \hline 1110 \end{array}$$

# Các phép toán số học

## ❖ Phép cộng

### ■ Lưu ý:

- ☐ Nếu số bit 1 chẵn, kết quả là 0;
- ☐ Nếu số bit 1 lẻ kết quả là 1
- ☐ Và cứ 1 cặp số 1 cho 1 số nhớ (bỏ qua số 1 dư, thí dụ với 5 số 1 ta kể là 2 cặp)

# Các phép toán số học

## ❖ Phép trừ

- Phép trừ 2 số nguyên  $X - Y = X + (-Y)$ .
- Nguyên tắc: Lấy bù 2 của  $Y$  để được  $-Y$ , rồi cộng với  $X$
- Khi thực hiện phép trừ cần lưu ý:

$$0 - 0 = 0 ;$$

$$1 - 1 = 0 ;$$

$$1 - 0 = 1 ;$$

$$0 - 1 = 1 \text{ nhớ 1 cho bit cao hơn}$$



# Các phép toán số học

## ❖ Phép trừ

### ■ Ví dụ:

$$\begin{array}{r} 1001 \\ - 0101 \\ \hline 0100 \end{array}$$

# Các phép toán số học

## ❖ Phép nhân

■ Khi thực hiện phép nhân cần lưu ý:

$$0 \times 0 = 0 ;$$

$$0 \times 1 = 0 ;$$

$$1 \times 1 = 1$$

# Các phép toán số học

## ❖ Phép nhân

### ■ Ví dụ:

$$\begin{array}{r} 1001 \\ \times 0101 \\ \hline 1001 \\ 000 \\ 1001 \\ 0000 \\ \hline 0101101 \end{array}$$

# Các phép toán số học

## ❖ Phép chia

- Thực hiện giống phép chia bình thường

# Các phép toán số học

## ❖ Phép chia

■ Ví dụ:

$$\begin{array}{r} 101101 \overline{) 101} \\ - 101 \phantom{0000} \\ \hline 0101 \\ 0101 \\ \hline 0 \end{array}$$

## Bài tập 2

### ■ Thực hiện các phép toán sau

1.  $10101 + 1011; 10110 + 1110$
2.  $11001 - 101; 11010 - 1110$
3.  $10101 * 101$
4.  $1101010 : 111$

# Nội dung chương 2

---

I. Khái niệm về biểu diễn dữ liệu

**II. Biểu diễn số nguyên**

III. Biểu diễn số thực

IV. Biểu diễn ký tự

V. Dữ liệu âm thanh, hình ảnh

## II. Biểu diễn số nguyên

---

1. Các khái niệm
2. Số nguyên không dấu
3. Số nguyên có dấu



# 1. Các khái niệm

- Giá trị  $X$  được biểu diễn trên  $n$  bit
  - $X$ : giá trị cần biểu diễn
  - $n$ : kích thước biểu diễn
  - Với  $n$  bit chỉ biểu diễn được các giá trị  $X$  trong một khoảng biểu diễn
- Số nguyên không dấu (unsigned integer)
- Số nguyên có dấu (signed integer)

## 2. Số nguyên không dấu

Dùng biểu diễn nhị phân của giá trị  $X$

Ví dụ:

$$X = 10_{10}, \quad n = 4 \quad \rightarrow X = 1010_2$$

$$X = 10_{10}, \quad n = 8 \quad \rightarrow X = 00001010_2$$

$$X = 100_{10}, \quad n = 8 \quad \rightarrow X = 01100100_2$$

Khoảng biểu diễn:  $0 \rightarrow 2^n - 1$

- $n = 8 : \quad 0 \rightarrow 255$

- $n = 16 : \quad 0 \rightarrow 65535$

# Bài tập 3

- Tìm biểu diễn nhị phân, bát phân, thập lục phân của các số thập phân không dấu sau đây:
  - 81, 102, 250 với  $n = 8$
  - 1000, 2050 với  $n = 16$

### 3. Số nguyên có dấu

---

- a. Mã độ lớn có dấu (dấu lượng) (signed magnitude)
- b. Mã bù 1 (one's complement)
- c. Mã bù 2 (two's complement)
- d. Mã quá N (excess N, biased)

## a. Mã độ lớn có dấu

- Dùng 1 bit thể hiện dấu:
  - 0: số dương (+)
  - 1: số âm (-)
- $(n-1)$  bit còn lại là biểu diễn nhị phân của độ lớn  $X$  (trị tuyệt đối của  $X$ )
- Ví dụ:
  - $X = 5, n = 8 \rightarrow X = 00000101$
  - $X = -5, n = 8 \rightarrow X = 10000101$

# Mã độ lớn có dấu(tt)

## ■ Khoảng biểu diễn:

$$-(2^{n-1} - 1) \rightarrow +(2^{n-1} - 1)$$

- $n = 8$ :  $-127_{10} \rightarrow +127_{10}$

- $n = 16$ :  $-32767_{10} \rightarrow +32767_{10}$

## ■ Nhận xét: với $n = 8$

- $00000000 = +0$

- $10000000 = -0$

## ■ Được sử dụng trên các máy tính thế hệ đầu, ví dụ IBM 7090

## b. Mã bù 1

- Dùng phép toán **bitwise NOT** (đổi bit 1 thành bit 0, bit 0 thành bit 1) để tạo số âm.
- Ví dụ: với  $n = 4$ 
  - $0101 = +5$
  - $1010 = -5 \rightarrow$  mã bù 1 của  $-5$  là  $1010$
- Nhận xét: có 1 bit dấu (0:dương, 1:âm)
- Khoảng biểu diễn:
  - $(2^{n-1} - 1) \rightarrow +(2^{n-1} - 1)$
  - $n = 8: \quad -127_{10} \rightarrow +127_{10}$
  - $n = 16: \quad -32767_{10} \rightarrow +32767_{10}$

# Mã bù 1 (tt)

- Nhận xét: với  $n = 8$ 
  - $00000000 = +0$
  - $11111111 = -0$
- Được sử dụng trên các máy tính thế hệ đầu, ví dụ PDP-1, UNIVAC 1100



## c. Mã bù 2

- Mã bù 2 của một số âm bằng mã bù 1 cộng thêm 1
- Ví dụ: với  $n = 4$ 
  - $0101 = +5$
  - $1010 = -5$  (mã bù 1 của -5 là 1010)
  - $1011$  (mã bù 2 của -5 là 1011)
- Nhận xét: có 1 bit dấu (0:dương, 1:âm)

## Mã bù 2 (tt)

### ■ Khoảng biểu diễn:

$$- 2^{n-1} \rightarrow +(2^{n-1} - 1)$$

- $n = 8$ :  $-128_{10} \rightarrow +127_{10}$

- $n = 16$ :  $-32768_{10} \rightarrow +32767_{10}$

### ■ Được sử dụng trên các máy tính hiện đại (trong CPU, ngôn ngữ lập trình)

# Tìm mã bù 2

$X$ : giá trị cần biểu diễn,  $n$ : số bit dùng biểu diễn

■ **X dương:** mã bù 2 là biểu diễn nhị phân dùng  $n$  bit của  $X$

- $X=5$ ,  $n=8$ , mã bù 2 của 5 là 00000101

■ **X âm:**

- Biểu diễn trị tuyệt đối  $X$  dùng  $n$  bit
- Đổi bit 1 thành bit 0 và bit 0 thành bit 1
- Cộng thêm 1

# Tìm mã bù 2 (tt)

■ Ví dụ:  $X = -10$ ,  $n = 8$

- $10 = 00001010_2$
- Đổi  $1 \rightarrow 0, 0 \rightarrow 1$ :  $11110101$
- Cộng thêm 1: mã bù 2 của  $-10$  là  $11110110$

# Tìm mã bù 2 (tt)

## ■ Phương pháp đơn giản cho X âm

- Biểu diễn trị tuyệt đối của X dùng n bit
- Tìm bit 1 đầu tiên từ bên phải
- Đổi  $1 \rightarrow 0$ ,  $0 \rightarrow 1$  tất cả các bit bên trái bit 1 đã tìm

## ■ Ví dụ: $X = -10$ , $n = 8$

- $10 = 00001010$
- Tìm bit 1 đầu tiên từ bên phải  $000010\mathbf{1}0$
- Mã bù 2 của  $-10$  là  $\mathbf{11110110}$

## d. Mã quá N

- Còn gọi là phương pháp di chuyển (biased)
- Sử dụng một số nguyên  $N$  cho trước làm giá trị dịch.
- Một giá trị thập phân (tức giá trị cần biểu diễn) sẽ được biểu diễn bằng dạng nhị phân của một số dương nào đó sao cho, giá trị của số dương này lớn hơn giá trị cần biểu diễn  $N$  đơn vị.

## d. Mã quá N

- Ví dụ: giả sử cần biểu diễn giá trị  $2_{10}$  theo số quá 5 (mẫu 8 bit):
- Bước 1: ta có:
  - Giá trị cần biểu diễn: 2
  - $N = 5$
- Bước 2: xác định số dương lớn hơn  $2_{10}$  nằm đơn vị, đó là số 7.

Vậy  $2_{10}$  được biểu diễn bằng dạng nhị phân của 7: 00000111.

# Mã quá N (tt)

## ■ Khoảng biểu diễn

$$-N \rightarrow +(2^n - 1 - N)$$

- $n = 8, N = 127: -127_{10} \rightarrow +128_{10}$

## ■ Mã quá N dùng trong biểu diễn số dấu chấm động theo tiêu chuẩn IEEE 754



# Các dạng mã hoá số nguyên với $n = 4$

Decimal	Unsigned	Sign and Magnitude	Ones' Complement	Two's Complement	Excess-7 (Biased)
+8	1000	N/A	N/A	N/A	1111
+7	0111	0111	0111	0111	1110
+6	0110	0110	0110	0110	1101
+5	0101	0101	0101	0101	1100
+4	0100	0100	0100	0100	1011
+3	0011	0011	0011	0011	1010
+2	0010	0010	0010	0010	1001
+1	0001	0001	0001	0001	1000
(+)0	0000	0000	0000	0000	0111
(-)0	N/A	1000	1111	N/A	N/A
-1	N/A	1001	1110	1111	0110
-2	N/A	1010	1101	1110	0101
-3	N/A	1011	1100	1101	0100
-4	N/A	1100	1011	1100	0011
-5	N/A	1101	1010	1011	0010
-6	N/A	1110	1001	1010	0001
-7	N/A	1111	1000	1001	0000
-8	N/A	N/A	N/A	1000	N/A

## Bài tập 4

1. Biểu diễn các số nguyên có dấu sau đây bằng 8 bit:  $B = -92$
2. Xác định giá trị của các số nguyên có dấu biểu diễn dưới đây (dùng dấu - độ lớn):
  - $C = 0110\ 1010$ ;  $D = 1100\ 0011$  (bù 2)
3. Xác định giá trị của các số nguyên có dấu được biểu diễn dưới đây (dùng mã bù 2):  
 $F = 1101\ 1010$

## Bài tập 4

- Cho  $n = 8$ , tìm mã bù 1, mã bù 2, mã quá 127 của các số thập phân sau đây:
  - 10, 61, 102
  - -22, -55, -100

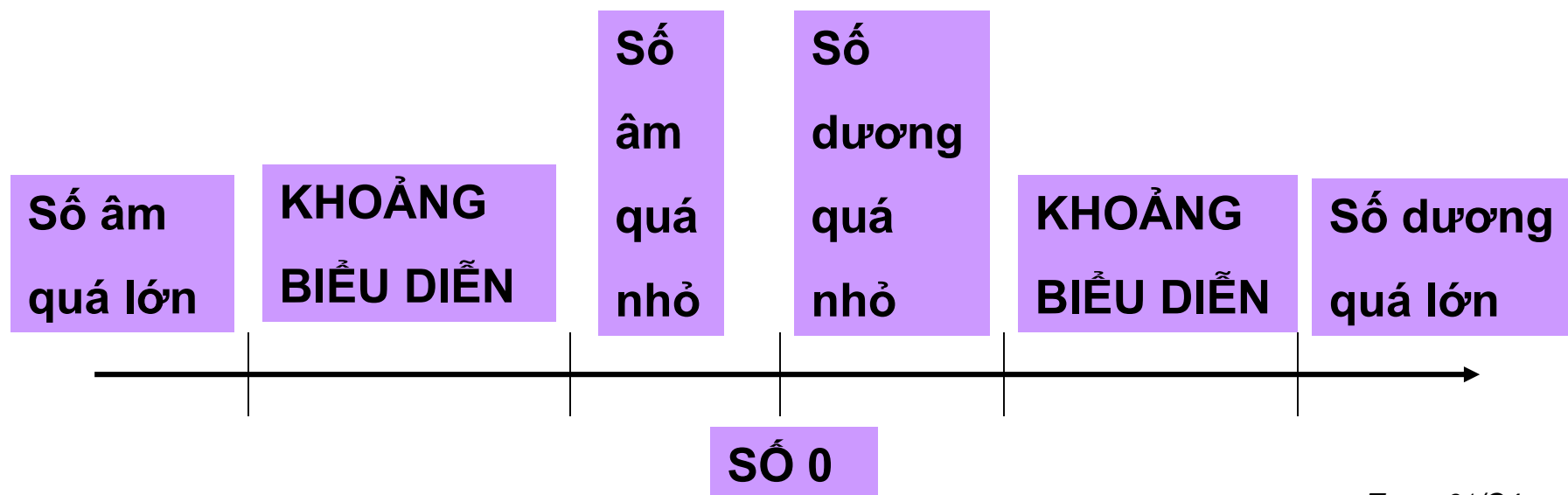
# III. Biểu diễn số thực

---

1. Khái niệm về số thực
2. Số dấu chấm tĩnh
3. Số dấu chấm động
4. Tiêu chuẩn số dấu chấm động IEEE 754

# 1. Khái niệm về số thực

- Trục số thực: chia làm 7 vùng
- Trong khoảng biểu diễn có vô số số thực
  - Ví dụ:  $20/3 = 6.666\dots$
  - Có thể có sai số khi biểu diễn số thực



## 2. Số dấu chấm tĩnh

### ■ Ví dụ:

- $+3.25_{10}$  ,  $-11.01_2$

### ■ Dùng $n+m+1$ bit để biểu diễn:

- 1 bit dấu (0: dương, 1: âm)
- $n$  bit: phần nguyên
- $m$  bit: phần phân số đúng

1 bit

$n$  bit

$m$  bit

Bit dấu

Phần nguyên

Phần phân số đúng

# Ví dụ số dấu chấm tĩnh

- $X = 3.375_{10}$ ,  $n = 7$ ,  $m = 8$ 
  - $00000011.01100000_2 = 03.60_{16}$
- $X = -7.25_{10}$ ,  $n = 7$ ,  $m = 8$ 
  - $10000111.01000000_2 = 87.40_{16}$

# Ứng dụng số dấu chấm tĩnh

- Dùng trong các chương trình tính toán thương mại (các ứng dụng spreadsheet), và được hỗ trợ trên một số ngôn ngữ lập trình



## Bài tập 5

- Cho  $n=7$ ,  $m=8$ , tìm biểu diễn số dấu chấm  
tĩnh của các số thập phân sau đây:
  - 100.625,      -70.3125,      -120.4375

### 3. Số dấu chấm động

- X là số bất kỳ, có thể phân tích dưới dạng:

$$X = m * a^e \quad (1)$$

trong đó:

a gọi là cơ số (radix)

m gọi là phần định trị (mantissa)

e gọi là phần bậc (exponent)

# Số dấu chấm động (tt)

## ■ Ví dụ:

- $a = 10, X = 3.14$

$$X = 3.14 = 0.314 * 10^1 = 31.4 * 10^{-1}$$

- $a = 2, X = 11.101$

$$X = (11.101)_2 = (1.1101 * 10^1)_2 = (0.11101 * 10^{10})_2$$

## ■ Với cơ số $a$ cho trước, có nhiều cách phân tích $X$ theo dạng (1)

## ■ Vị trí ngăn cách giữa phần nguyên và phần phân số phụ thuộc phần bậc $e$ không cố định như trong biểu diễn số dấu chấm tĩnh $\rightarrow$ biểu diễn số dấu chấm động

# Số dấu chấm động (tt)

- Nếu đặt thêm điều kiện  $a^{-1} \leq |m| < 1$  thì cách phân tích là duy nhất: dạng chuẩn
- Xét lại ví dụ trên:  
 $a = 10, X = 3.14_{10} = \underline{0.314 * 10^1}$   
 $a = 2, X = (11.101)_2 = \underline{(0.11101 * 10^{10})}_2$

## 4. Tiêu chuẩn số dấu chấm động IEEE 754

- Do tổ chức IEEE (Institute of Electrical and Electronic Engineers)
  - Mở rộng thành tiêu chuẩn IEEE 854
- Được sử dụng phổ biến trên các đơn vị số dấu chấm động (FPU, Floating-Point Unit) trong các bộ xử lý, và trên các ngôn ngữ lập trình

# Tiêu chuẩn IEEE754 (tt)

- Tiêu chuẩn IEEE 754 gồm các dạng số chấm động tiêu chuẩn (normalized):
  - Số chính xác đơn – Single-precision 32 bit (Số single)
  - Số chính xác kép – Double-precision 64 bit (Số double)
  - Số chính xác kép mở rộng – Double-Extended precision 80 bit (Số extended)
- Số extended dùng để giảm các lỗi khi làm tròn số và chỉ dùng bên trong các FPU

# Tiêu chuẩn IEEE754 (tt)

- Các số dấu chấm động IEEE 754 có dạng:

$$X = S * 2^e$$

trong đó: S là phần định trị (significand)

e là phần bậc (exponent)

với  $1.0 \leq |S| < 2$

S có thể viết thành  $S = 1.f$

trong đó f là phần phân số (fraction)

$$X = \pm 1.f * 2^e$$

# Số chính xác đơn (số single)

## ■ Số chính xác đơn có 32 bit:

- 1 bit dấu (0:dương, 1:âm)
- 8 bit phần bậc dùng mã quá 127
- 23 bit phần phân số (thuộc phần định trị)

**1 bit**

**8 bit**

**23 bit**

**Bit dấu**

**Phần bậc**

**Phần phân số**



# Ví dụ số single

■  $X = 0.5_{10}$

- $0.5 = 1.0 * 2^{-1}$

- Bit dấu: 0

- Phần bậc: -1

mã quá 127 của -1 là  $-1+127 = 126$

$$126_{10} = 01111110_2$$

- Phần định trị  $S = 1.0$ , phần phân số = 0

■ Biểu diễn dạng số chính xác đơn của 0.5:

- $0\ 01111110\ \underline{00..00}_2 = 3F000000_{16}$

**23 BIT 0**

# Ví dụ số single (tt)

■  $X = -20_{10}$

- $-20 = -1.25 * 2^4$

- Bit dấu: 1

- Phần bậc: 4

mã quá 127 của 4 là  $4+127 = 131$

$$131_{10} = 10000011_2$$

- Phần định trị  $S = 1.25$ , phần phân số  $= 0.25$

$$0.25_{10} = 0.01_2$$

■ Biểu diễn dạng số chính xác đơn của  $-20_{10}$ :

- $1 \ 10000011 \ 010\underline{00}_2 = C1A00000_{16}$   
21 BIT 0

# Số chính xác kép (số double)

## ■ Số chính xác kép có 64 bit:

- 1 bit dấu (0:dương, 1:âm)
- 11 bit phần bậc dùng mã quá 1023
- 52 bit phần phân số (thuộc phần định trị)

**1 bit**

**11 bit**

**52 bit**

**Bit dấu**

**Phần bậc**

**Phần phân số**

# Số chính xác kép mở rộng (Số extended)

- Số chính xác kép mở rộng có 80 bit:
  - 1 bit dấu (0:dương, 1:âm)
  - 15 bit phần bậc dùng mã quá 16383
  - 64 bit phần phân số (thuộc phần định trị)

**1 bit**

**15 bit**

**64 bit**

**Bit dấu**

**Phần bậc**

**Phần phân số**

# Các đặc điểm chính của số single, số double

	Số single	Số double
Số bit dấu	1	1
Số bit phần bậc	8	11
Số bit phần định trị	23	52
Tổng số bit	32	64
Biểu diễn phần bậc	Mã quá 127	Mã quá 1023
Khoảng phần bậc	-126 $\rightarrow$ +127	-1022 $\rightarrow$ 1023

## Các đặc điểm chính của số single, số double (tt)

	Số single	Số double
Giá trị nhỏ nhất	$2^{-126}$	$2^{-1022}$
Giá trị lớn nhất	$\approx 2^{+128}$	$\approx 2^{+1024}$
Khoảng biểu diễn (theo thập phân)	$\approx 10^{-38} \rightarrow 10^{+38}$	$\approx 10^{-308} \rightarrow 10^{+308}$
Giá trị nhỏ nhất (dạng số đặc biệt)	$\approx 10^{-45}$	$\approx 10^{-324}$

# Các dạng số đặc biệt

Số tiêu chuẩn (Normalized)	+/-	$0 < \text{exponent} < \text{max}$	Tùy ý
Số không chuẩn (Denormalized)	+/-	0	Tùy ý khác 0
Zero	+/-	0	0
Vô cực / Vô cùng (Infinity)	+/-	11...11	0
NaN (Not a Number)	+/-	11...11	Tùy ý khác 0

# Các dạng số đặc biệt (tt)

- Số dạng denormalized dùng để biểu diễn số rất nhỏ
  - Số nhỏ nhất là  $2^{-23} * 2^{-127} = 2^{-150}$
- Có thể biểu diễn 2 số 0 (+0, -0)
- Số vô cực có thể dùng làm toán hạng tuân theo các quy tắc toán học cho số vô cực
- Khi kết quả phép toán không xác định, ví dụ  $\infty/\infty$  thì dùng dạng NaN (Not a Number)



# Làm tròn số (rounding)

## ■ Tiêu chuẩn IEEE 754 có các dạng làm tròn số:

- Unbiased: round to nearest

Làm tròn về số gần nhất

Nếu số cần làm tròn ở giữa 2 giá trị thì làm tròn về số có bit cuối bên phải là 0

- Toward zero: làm tròn về zero
- Toward positive infinity: làm tròn về  $+\infty$
- Toward negative infinity: làm tròn về  $-\infty$

## Bài tập 6

- Đổi các giá trị thập phân sau đây sang dạng số single (IEEE 754), trình bày kết quả ở dạng hệ 16:
  - -15.5,          20.5,          -34
- Đổi các số dạng single sau đây về dạng thập phân:
  - 42E48000H,          3F880000H



## IV. Biểu diễn ký tự

---

1. Bộ mã ASCII
2. Bộ mã Unicode

# 1. Bộ mã ASCII

American Standard Code for Information Interchange

- Do ANSI (American National Standards Institute) công bố năm 1967, cập nhật năm 1986
- Bảng mã ASCII dùng biểu diễn ký tự trên máy tính và các thiết bị truyền thông

# Mã ASCII (tt)

- Mã ASCII chuẩn dùng 7 bit, biểu diễn được 128 ký tự, bao gồm:
  - Các ký tự điều khiển (control characters) có giá trị (mã) từ 0 đến 1Fh
  - Các ký tự in được (printable characters) có giá trị (mã) từ 20h đến 7Fh
- Mã ASCII mở rộng dùng 8 bit, bao gồm:
  - Phần ASCII chuẩn
  - Các ký tự đặc biệt có giá trị (mã) từ 80h đến FFh

# Mã ASCII chuẩn?

Ctrl	Dec	Hex	Char	Code
^@	0	00		NUL
^A	1	01		SOH
^B	2	02		STX
^C	3	03		ETX
^D	4	04		EOT
^E	5	05		ENQ
^F	6	06		ACK
^G	7	07		BEL
^H	8	08		BS
^I	9	09		HT
^J	10	0A		LF
^K	11	0B		VT
^L	12	0C		FF
^M	13	0D		CR
^N	14	0E		SO
^O	15	0F		SI
^P	16	10		DLE
^Q	17	11		DC1
^R	18	12		DC2
^S	19	13		DC3
^T	20	14		DC4
^U	21	15		NAK
^V	22	16		SYN
^W	23	17		ETB
^X	24	18		CAN
^Y	25	19		EM
^Z	26	1A		SUB
^[	27	1B		ESC
^\	28	1C		FS
^]	29	1D		GS
^^	30	1E	▲	RS
^-	31	1F	▼	US

Dec	Hex	Char
32	20	
33	21	!
34	22	..
35	23	#
36	24	\$
37	25	%
38	26	&
39	27	'
40	28	(
41	29	)
42	2A	*
43	2B	+
44	2C	,
45	2D	-
46	2E	.
47	2F	/
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?

Dec	Hex	Char
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	^
95	5F	_

Dec	Hex	Char
96	60	'
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	
125	7D	}
126	7E	~
127	7F	␣

# Mã ASCII chuẩn<sup>2</sup>

## Standard ASCII

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	□

# Mã ASCII mở rộng

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	€	□	‚	ƒ	„	…	†	‡	^	‰	Š	<	Œ	□	Ž	□
9	□	ˆ	‚	“	”	•	–	—	~	™	š	>	œ	□	ž	ÿ
A		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯
B	°	±	²	³	´	µ	¶	·	,	¹	º	»	¼	½	¾	¿
C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	ö	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ





## 2. Bộ mã Unicode

---

- a. Giới thiệu Unicode
- b. Các đặc điểm chính

## a. Giới thiệu Unicode

- Unicode là tiêu chuẩn mã hóa ký tự của Hiệp hội Unicode (Unicode Consortium) bao gồm các nhà sản xuất IBM, Apple, HP, MicroSoft, Adobe, ...
- Unicode là một hiện thực của chuẩn ISO 10646 UCS 2 (Universal Character Set)
- Unicode được hỗ trợ trên các hệ điều hành, trình duyệt web, các tiêu chuẩn phần mềm hiện đại như XML, Java, LDAP, CORBA, ...

# Giới thiệu Unicode (tt)

- Mục tiêu của Unicode là cung cấp mã (code point) duy nhất cho ký tự, ký hiệu trên tất cả các ngôn ngữ, hệ thống chữ viết (writing systems)
  - Việc hiển thị ký tự (font) do chương trình ứng dụng (web, word processor, ... ) thực hiện
- Unicode là mã 16 bit biểu diễn được 65536 ký tự, và có thể mở rộng đến trên 1 triệu ký tự

## b. Các đặc điểm chính

- Có các phiên bản Unicode: 1.0, 1.1, 2.0, ..., 7.0
- Cách thể hiện mã ký tự theo Unicode: U+xxxx, với x là 1 chữ số hệ 16, ví dụ:
  - Latin -1 {U+0080..U+00FF}
  - Latin Extended A, B {U+0100..U+024F}
  - Combining Diacritical Marks {U+0300..U+036F}
  - Latin Extended Additional {U+1E00..U+1EFF}
  - Đơn vị tiền tệ Việt Nam (Đồng): đ U+20AB

# Các sơ đồ mã hóa chính

## UTF (Unicode Transformation Format)

### ■ UTF – 16:

- Dạng mã hóa chuẩn, dùng 1 hay 2 số nguyên 16 bit
- Được dùng trong Windows API, .NET, Java

### ■ UTF – 8:

- Gồm 1 đến 4 byte
- Được thiết kế để tương thích với mã ASCII và các giao thức trên byte
- Dùng trên Web Browsers, E-Mail

# Bài tập 8

- Cho biết giá trị các ký tự trong bảng mã ASCII dưới dạng hệ 16 là:

Ký tự	A	B	C	D	E	F	G	H	L
ASCII hệ 16	41	42	43	44	45	46	47	48	49
	J	K	L	M	N	O	P	Q	R
	4A	4B	4C	4D	4E	4F	50	51	52
	S	T	U	V	W	X	Y	Z	space
	53	54	55	56	57	58	59	5A	20

- Hãy mã hóa thông điệp: “BITCOIN TANG” theo *bảng mã ASCII mở rộng* (8 bit) dưới dạng hệ 10.



## V. Dữ liệu âm thanh, hình ảnh

---

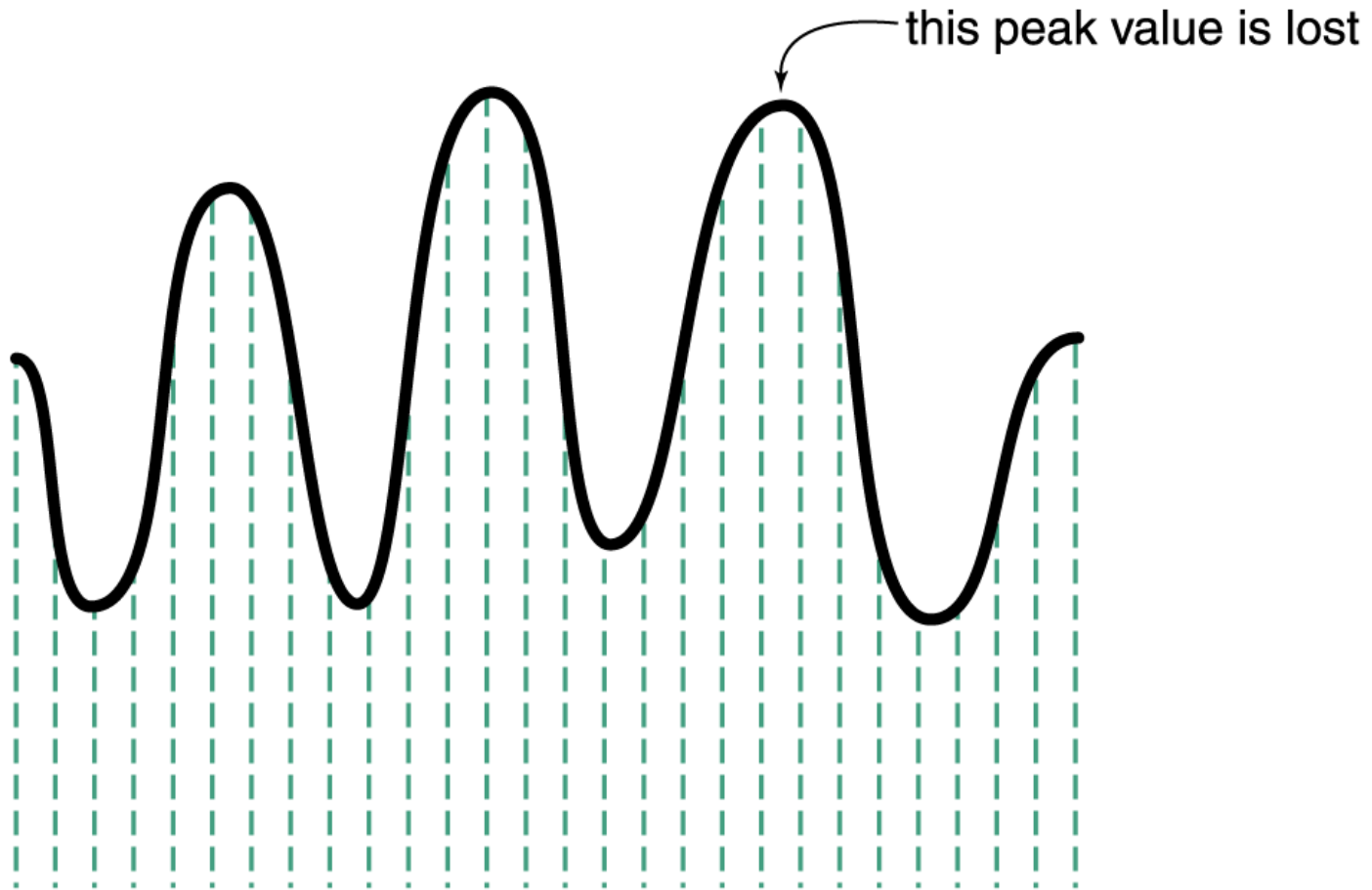
1. Dữ liệu âm thanh (audio)
2. Dữ liệu hình ảnh (images, graphics)
3. Dữ liệu hình ảnh động (video)

# 1. Dữ liệu âm thanh

- Con người cảm nhận âm thanh do sóng âm tác động trên tai, và tai truyền tín hiệu lên não
- Thiết bị loa tạo âm thanh do các tín hiệu điện dạng tương tự (analog)
- Khi số hóa dữ liệu âm thanh thì sẽ định kỳ đo điện áp tín hiệu và lưu lại trên các giá trị số thích hợp
  - Sampling: Quá trình lấy mẫu



# Lấy mẫu âm thanh



# Lấy mẫu âm thanh (tt)

- Âm thanh nghe được
  - $20\text{Hz} \rightarrow 20000\text{Hz}$
- Về nguyên tắc, tốc độ lấy mẫu khoảng 40000 lần/giây là có thể lưu trữ, phục hồi âm thanh với chất lượng tốt

# Các định dạng file âm thanh thông dụng

- Dạng không nén (uncompressed)
  - PCM (Pulse Code Modulation)
  - Dạng file .wav
- Dạng nén mất dữ liệu (lossy compression)
  - MPEG-1 layer 3 (MP3), Vorbis, lossy Windows Media Audio (WMA)
- Dạng nén không mất dữ liệu (lossless compression)
  - Apple lossless, lossless WMA

## 2. Dữ liệu hình ảnh

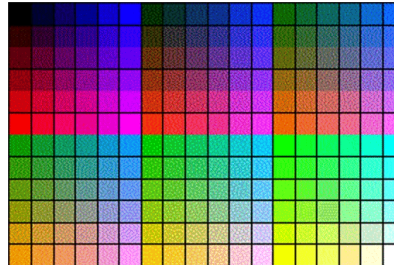
- Mắt người phân biệt màu sắc dựa trên tần số ánh sáng tác động trên võng mạc
- Võng mạc có các tế bào nhận màu sắc theo ba nhóm chính: Red, Green, Blue
- Màu sắc biểu diễn trên máy tính thường dựa trên các giá trị R,G,B thể hiện thành phần của các màu cơ bản

# Ví dụ màu theo RGB

RGB Value			Actual Color
Red	Green	Blue	
0	0	0	black
255	255	255	white
255	255	0	yellow
255	130	255	pink
146	81	0	brown
157	95	82	purple
140	0	0	maroon

# Biểu diễn màu sắc

- Color depth: số lượng dữ liệu dùng để biểu diễn màu
  - Hi Color: 16 bit, True Color: 32 bit
- Một chương trình ứng dụng có thể chỉ thể hiện một số lượng màu hạn chế theo color palette (bảng màu)

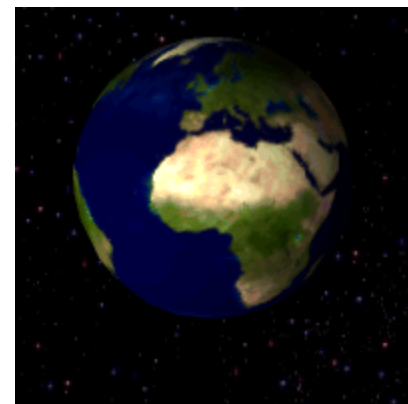


# Dữ liệu hình ảnh

- Bao gồm tập hợp các điểm ảnh (pixel)
- Resolution (độ phân giải): số điểm ảnh thể hiện một hình ảnh
  - 800x600
  - 1024x800

# Raster graphic format

- Dữ liệu hình ảnh được lưu dựa trên thông tin các điểm ảnh: raster graphic format
- Các định dạng file thông dụng:
  - BMP (BitMaP), dạng uncompressed
  - GIF (Graphic Interchange Format) dạng lossless data compression
  - PNG (Portable Network Graphics) dạng lossless data compression, thay thế dạng GIF
  - JPEG (Joint Photographic Experts Group) dạng lossy data compression





# Vector graphic format

- Dữ liệu hình ảnh được lưu theo các dạng hình học (điểm, đường thẳng, đường cong, ...) và màu sắc: vector graphic format
- Hình ảnh dạng vector graphic có thể thay đổi kích thước theo toán học
- Không phù hợp cho hình ảnh thực

# Vector graphic format (tt)

---

## ■ Các định dạng file thông dụng

- swf (Small Web Format/ShockWave Flash)
- ps (PostScript)
- wmf (Windows Metafile)
- pdf (Portable Document Format)

### 3. Dữ liệu hình ảnh động

- Hình ảnh động (video) bao gồm các hình ảnh liên tục
- Thông số chính: frames per second
  - Phim: 24 frames/sec
- Video Codec  
(C**O**mpressor/D**E**C**O**mpressor): là phương pháp chuyển hình ảnh động thành dữ liệu trên máy tính

# Dữ liệu hình ảnh động (tt)

- Hầu hết các video codec đều là dạng nén mất dữ liệu (lossless compression)
- Có 2 dạng nén chính
  - Temporal: dựa trên sự khác nhau giữa các frame liên tục
  - Spatial: dựa trên dữ liệu có trên từng frame (tương tự nén ảnh tĩnh)

# Các định dạng file thông dụng

---

- Container format: định dạng file nén theo các codec
- Các định dạng container thông dụng:
  - avi (chuẩn trên Windows)
  - asf (dùng cho WMA, WMV)
  - mov (Quicktime container)
  - Mpeg-2, Mpeg4 (Moving Picture Experts Group)
  - RealMedia (dùng cho Real Video, Real audio)
  - DivX media format

# Bài tập

1. Đổi số thập phân sau đây sang **Hệ 2, Hệ 8, Hệ 16**: 187
2. Đổi các số sau sang **HỆ NHỊ PHÂN**  
 $210_8, 6D2_{16}$
3. Đổi các số sau sang **HỆ THẬP PHÂN**  
 $101101, 131_8, C2_{16}$

# Bài tập

1. Xác định giá trị của số nguyên có dấu được biểu diễn dưới dạng mã bù 2:

$$F = 11101100$$

2. Cho  $n = 8$ , hãy biểu diễn dưới các dạng: Độ lớn có dấu, Mã bù 1, Mã bù 2, Mã quá  $N = 127$  của số nguyên có dấu -97.

# Bài tập

1. Cho  $n=7$ ,  $m=8$ , tìm biểu diễn số dấu chấm tĩnh của các số thập phân: 85.625
2. Đổi giá trị thập phân sau đây sang dạng số single (IEEE 754), trình bày kết quả ở dạng hệ 16: -72
3. Đổi số dạng single sau đây về dạng thập phân:  
 $9E42000_H$



# Bài tập

- Cho biết giá trị các ký tự trong bảng mã ASCII dưới dạng hệ 16 là:

Ký tự	A	B	C	D	E	F	G	H	L
ASCII hệ 16	41	42	43	44	45	46	47	48	49
	J	K	L	M	N	O	P	Q	R
	4A	4B	4C	4D	4E	4F	50	51	52
	S	T	U	V	W	X	Y	Z	space
	53	54	55	56	57	58	59	5A	20

- Hãy mã hóa thông điệp: “Khoa CNTT” theo *bảng mã ASCII mở rộng* (8 bit) dưới dạng hệ 10.