

25 YEARS ANNIVERSARY  
SOICT

ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# Tin học đại cương

## IT1110

# Nội dung học phần

- Phần I: Tin học căn bản (3 chương)
  - Khái niệm thông tin và biểu diễn thông tin trong máy tính
  - Giới thiệu hệ thống máy tính và ứng dụng
  - Làm quen với thuật toán
- Phần II: Lập trình C (8 chương)
  - Giới thiệu ngôn ngữ lập trình C
  - Xây dựng chương trình đơn giản với ngôn ngữ C
- Thực hành (15 tiết): theo lịch của Trung tâm máy tính
- Thi trên máy (trắc nghiệm + lập trình)

# Phần I – Tin học căn bản

## Chương 1 Thông tin và biểu diễn thông tin

- Các khái niệm cơ bản về thông tin và tin học
- Biểu diễn dữ liệu trong máy tính

# Nội dung chính

1. Các khái niệm cơ bản về thông tin và tin học
  1. Thông tin và xử lý thông tin
  2. Máy tính điện tử và phân loại
  3. Tin học và các ngành liên quan
2. Biểu diễn dữ liệu trong máy tính
  1. Biểu diễn số trong các hệ đếm
  2. Biểu diễn dữ liệu trong máy tính & đơn vị thông tin
  3. Biểu diễn số nguyên
  4. Biểu diễn số thực
  5. Biểu diễn ký tự

# 1.1 Thông tin và xử lý thông tin

- Thông tin
- Dữ liệu
- Tri thức
- Hệ thống thông tin
- Xử lý thông tin

# Thông tin

## Thông tin (information) là gì ?

- Là khái niệm trừu tượng mô tả tất cả những gì đem lại cho con người sự *hiểu biết, nhận thức* về các sự vật, hiện tượng.
- Thông tin tồn tại khách quan. Có thể tạo ra, phát sinh truyền đi, lưu trữ,...
- Là ngữ cảnh trong đó dữ liệu được xem xét.

# Dữ liệu

## Dữ liệu (data)

- Là biểu diễn của thông tin được thể hiện bằng các tín hiệu vật lý.
- Là vật liệu thô mang tin,
  - Dữ liệu sau khi được tập hợp và xử lý sẽ cho ra thông tin.

## Dữ liệu trong thực tế

- Các số liệu: Dữ liệu số như trong các bảng biểu.
- Các ký hiệu quy ước; ví dụ chữ viết...
- Các tín hiệu vật lý; ví dụ như ánh sáng, âm thanh, nhiệt độ, áp suất,...



# Dữ liệu

- Thông tin chứa đựng hiểu biết về sự vật, hiện tượng.
- Dữ liệu chỉ là các sự kiện không có cấu trúc và không có ý nghĩa nếu không được tổ chức và xử lý.

## Ví dụ

Nhiệt độ cơ thể - dữ liệu dạng số

- $39^{\circ}\text{C}$   $\rightarrow$  thông tin đang bị sốt
- $36.7^{\circ}\text{C}$   $\rightarrow$  thông tin nhiệt độ cơ thể bình thường

# Tri thức

## ▪ Tri thức (Knowledge)

- Tri thức theo nghĩa thường là thông tin ở mức trừu tượng hơn

## ▪ Tri thức rất đa dạng

- Có thể là sự kiện, là thông tin
- Là cách mà một người thu thập được qua kinh nghiệm hoặc qua đào tạo.
- Có thể là sự hiểu biết chung hay về một lĩnh vực cụ thể nào đó.

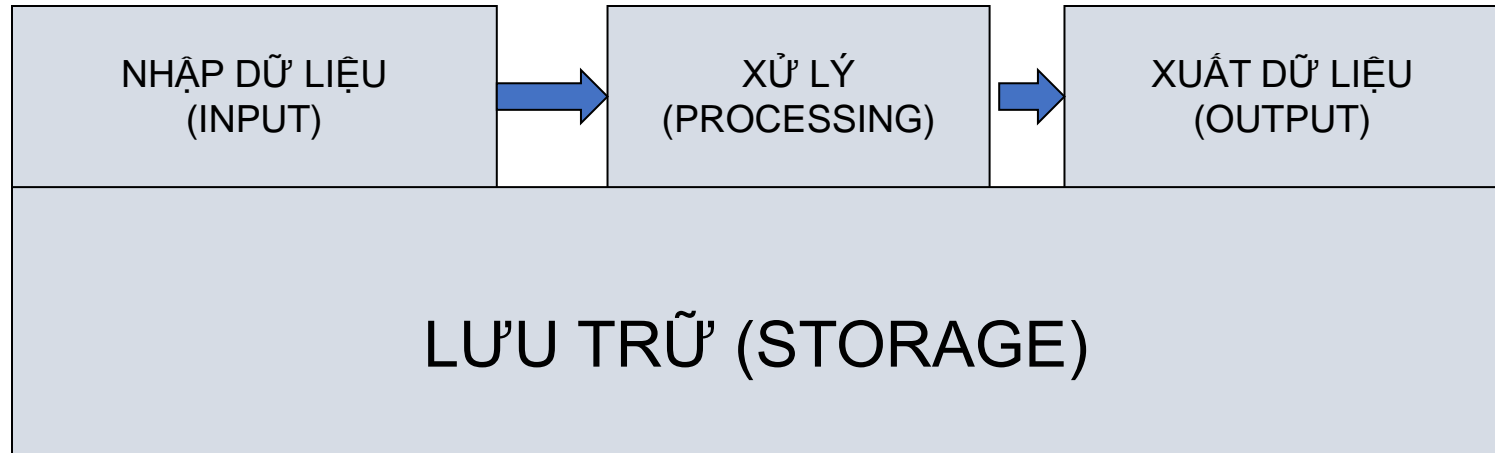
# Hệ thống thông tin

Hệ thống thông tin (*information system*) là một hệ thống thu nhận dữ liệu, xử lý chúng để tạo nên thông tin có ý nghĩa hoặc dữ liệu mới.



# Xử lý thông tin

## Quy trình xử lý thông tin



## Xử lý thông tin bằng máy tính điện tử

- Tiết kiệm rất nhiều thời gian, công sức
- Đạt độ chính xác cao nhờ tự động một phần hay toàn bộ quá trình xử lý thông tin.

# 1.2 Máy tính và phân loại

- Khái niệm
  - Máy tính và chương trình máy tính
  - Các thể hệ của máy tính
- Phân loại máy tính

# Khái niệm

- **Máy tính (*computer*):** Là thiết bị điện tử thực hiện các công việc
  - Nhận thông tin vào
  - Xử lý thông tin theo chương trình được lưu trữ sẵn bên trong
  - Đưa thông tin ra
- **Chương trình (*program*):** Là một dãy các lệnh trong bộ nhớ nhằm yêu cầu máy tính thực hiện công việc cụ thể.  
⇒ Máy tính hoạt động theo chương trình.

# Lịch sử hình thành và phát triển

## Các thế hệ của máy tính điện tử

- Thế hệ 1 (1950-1958)
- Thế hệ 2( 1958-1964)
- Thế hệ 3 (1965-1974)
- Thế hệ 4 (1974 – nay)
- Thế hệ 5 (1990 – nay)

# Thế hệ 1

## Von Neumann Machine (1950-1958)

- Sử dụng các bóng đèn điện tử chân không
- Mạch riêng rẽ, vào số liệu bằng phiếu đục lỗ
- Điều khiển bằng tay, kích thước rất lớn
- Tiêu thụ năng lượng nhiều, tốc độ tính chậm khoảng 300 - 3.000 phép tính/s.

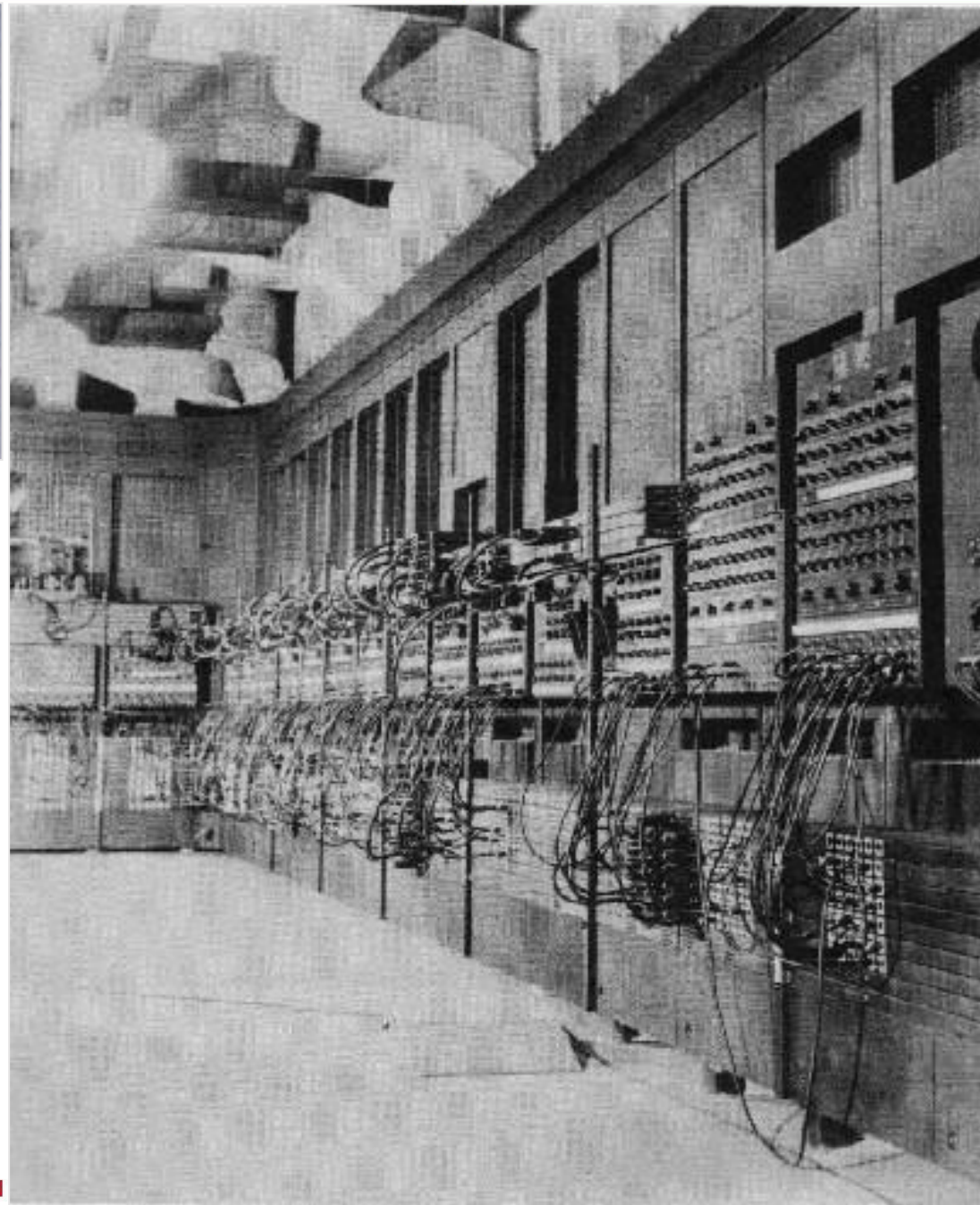
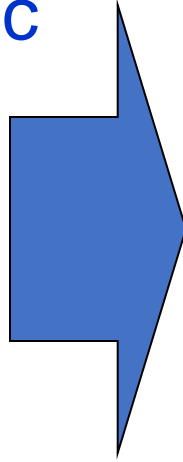
**Ví dụ:** EDVAC (Mỹ), BESEM (Liên xô cũ)

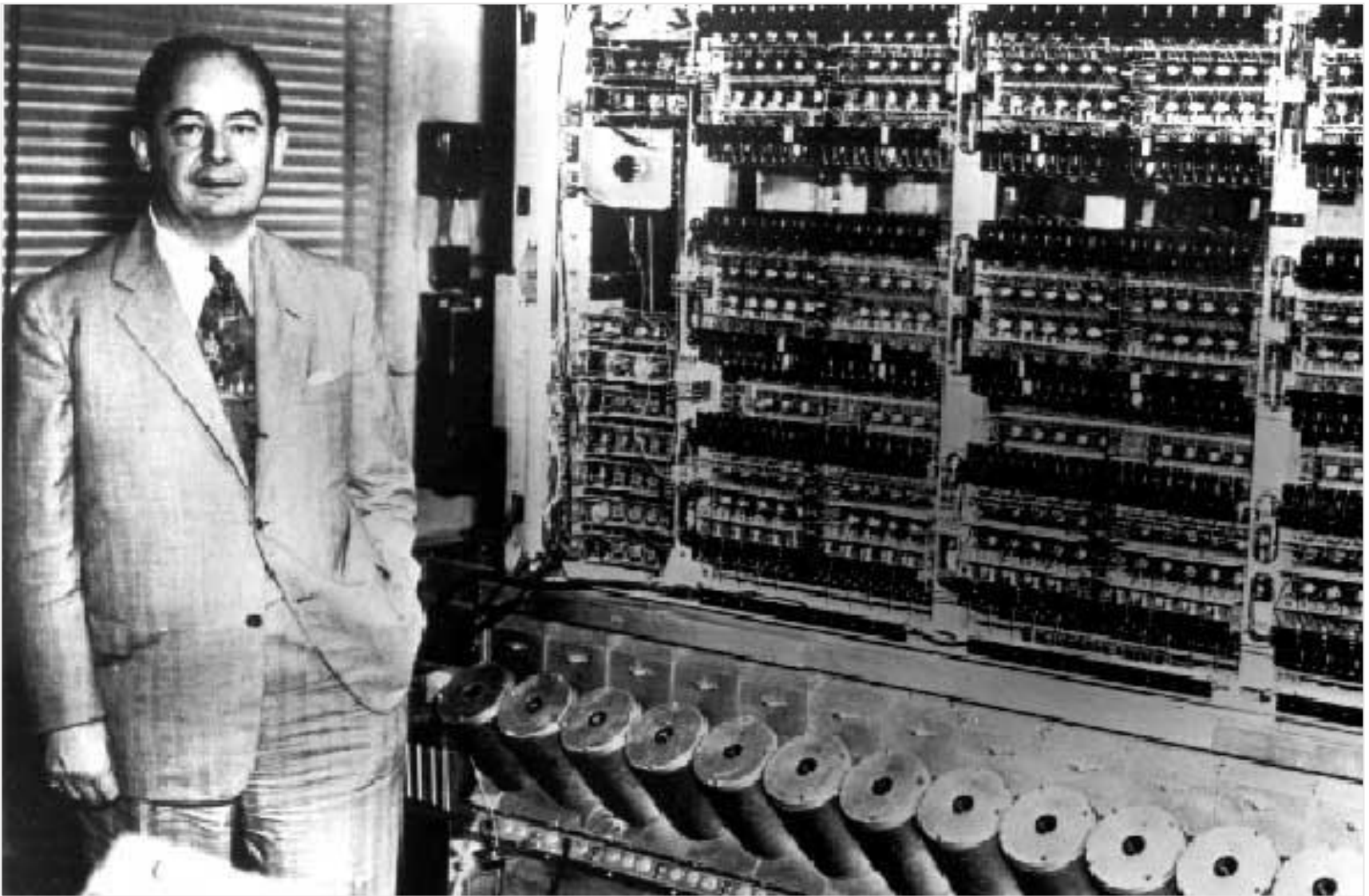




Bóng đèn chân không

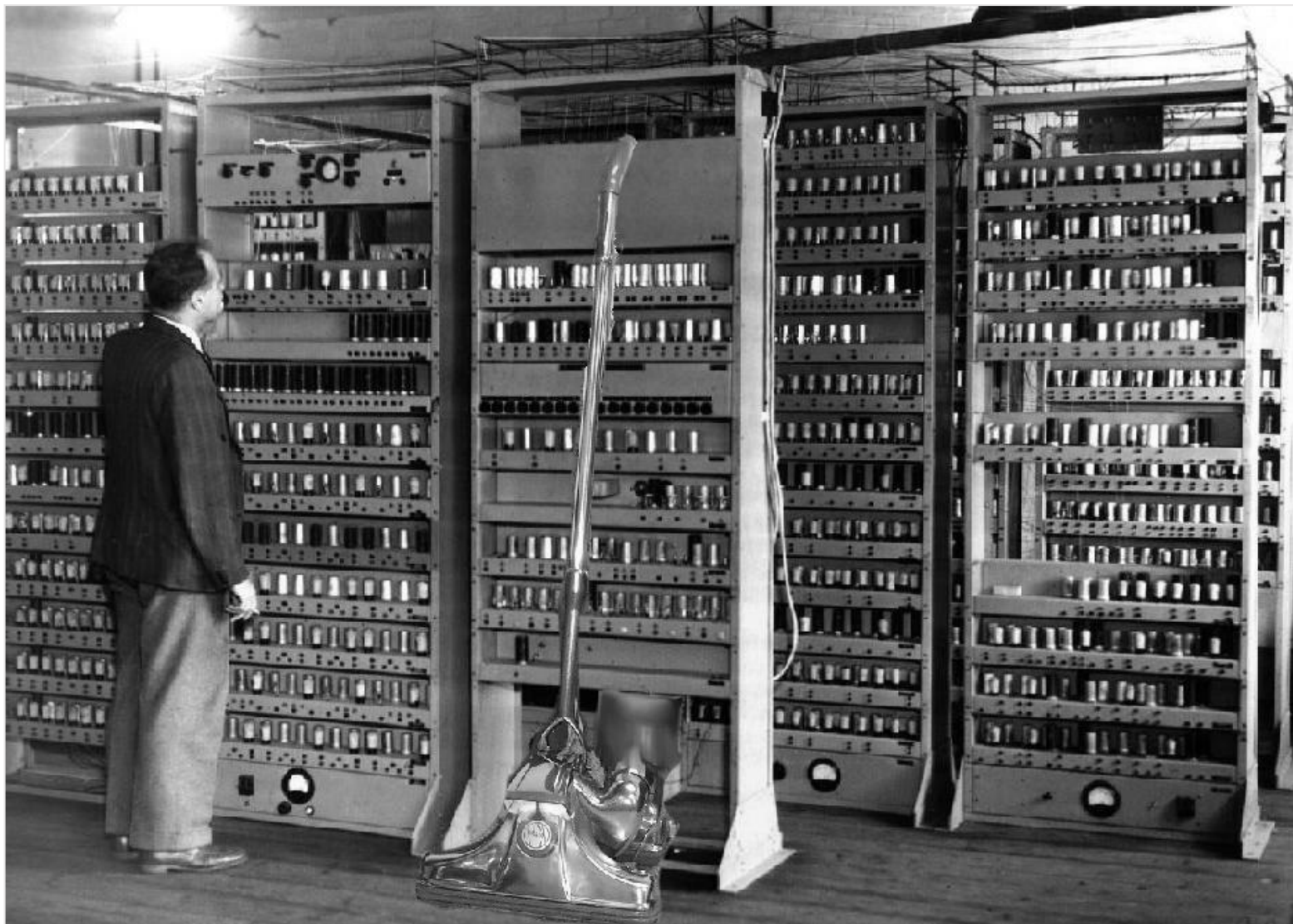
Máy tính đầu tiên:  
ENIAC (Electronic  
Numerical  
Integrator And  
Computer)





Von Neumann & **UNIVAC** (Universal Automatic Computer)





## EDVAC: Electronic Discrete Variable Automatic Computer

# Thế hệ 2

## Transistors (1958 - 1964):

- Sử dụng **bộ xử lý bằng đèn bán dẫn, mạch in**
- Đã có chương trình dịch như Cobol, Fortran và hệ điều hành đơn giản.
- Kích thước máy còn lớn
- Tốc độ tính khoảng 10.000 - 100.000 phép tính/s

## Ví dụ

- IBM 7000 series (Mỹ)
- MINSK (Liên Xô cũ)



## IBM 7030





## MINSK (Liên Xô cũ)

# Thế hệ 3

## Integrated Circuits (1965 - 1974):

- Các bộ vi xử lý được gắn vi mạch điện tử cỡ nhỏ
- Tốc độ tính khoảng 100.000 - 1 triệu phép tính/s.
- Có các hệ điều hành đa chương trình, nhiều người đồng thời theo kiểu phân chia thời gian.
- Kết quả từ máy tính có thể in trực tiếp từ máy in.

## Ví dụ

- IBM-360 (Mỹ)
- EC (Liên xô cũ)



## IBM 360/91



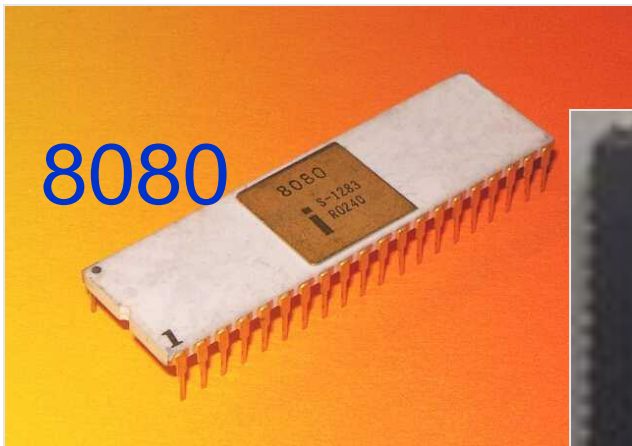
# Thế hệ 4

## **LSI**(Large Scale Integration), **Multiprocessors**:

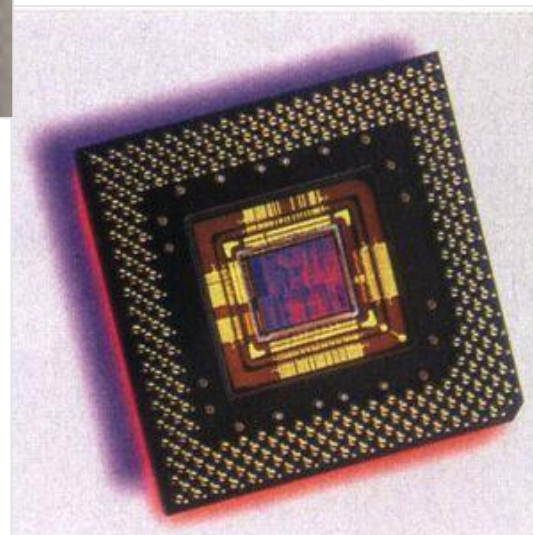
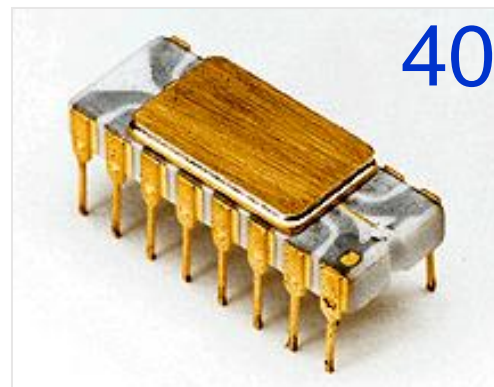
- Máy tính có các vi mạch đa xử lý
- Tốc độ: hàng chục triệu đến hàng tỷ phép tính/s.
- Hai loại máy tính chính:
  - Máy tính cá nhân để bàn (Personal Computer - PC) hoặc xách tay (Laptop hoặc Notebook computer)
  - Các loại máy tính chuyên nghiệp thực hiện đa chương trình, đa xử lý,...
- Hình thành các hệ thống mạng máy tính (Computer Networks).
- Các ứng dụng phong phú, đa phương tiện

# Vi mạch Intel

8080



4004



80386

Pentium

# Thế hệ 5

## **VLSI** (Very Large Scale Integration), **ULSI** (Ultra), Artificial Intelligence (**AI**)

- Công nghệ vi điện tử với tốc độ tính toán cao và khả năng xử lý song song.
- Mô phỏng các hoạt động của não bộ và hành vi con người
- Có trí khôn nhân tạo với khả năng tự suy diễn phát triển các tình huống nhận được
- Hệ quản lý kiến thức cơ bản để giải quyết các bài toán đa dạng.

# Phân loại máy tính

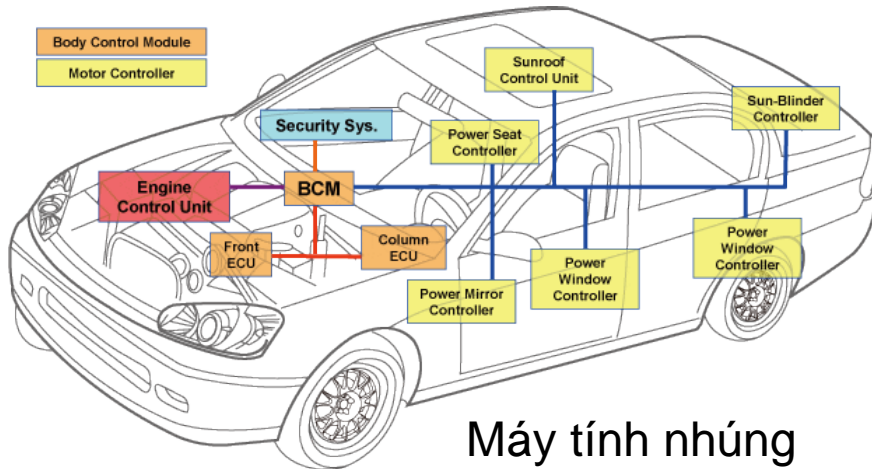
- **Cách phân loại truyền thống**
  - Máy vi tính (Microcomputer)
  - Máy tính nhỏ (Minicomputer)
  - Máy tính lớn (Mainframe Computer)
  - Siêu máy tính (Supercomputer)

# Phân loại máy tính hiện đại

- Thiết bị di động cá nhân (Personal Mobile Devices):
  - Smartphones, Tablet
- Máy tính cá nhân, máy trạm (PC, Workstation)
  - Desktop computers, Laptop computers
- Máy chủ (Servers)
  - Thực chất là Máy phục vụ
  - Dùng trong mạng theo mô hình Client/Server
- Máy tính cụm/máy tính qui mô lớn
  - Sử dụng tại các trung tâm tính toán, trung tâm dữ liệu
- Siêu máy tính
- Máy tính nhúng (Embedded Computers)
  - Đặt ẩn trong thiết bị khác
  - Được thiết kế chuyên dụng



# Một số loại máy tính



Máy tính nhúng



PC



Server



Super computer

# 1.3 Tin học và các ngành liên quan

- Thuật ngữ tin học
- Công nghệ thông tin
- Công nghệ thông tin và truyền thông

# Thuật ngữ tin học

- Nguồn gốc từ tiếng Đức
  - Năm 1957 [Karl Steinbuch](#) đề xướng trong bài báo *Informatik: Automatische Informationsverarbeitung* (*Informatics: automatic information processing*).
- Năm 1962, [Philippe Dreyfus](#) người Pháp gọi là “*informatique*”, tiếp theo là [Walter F.Bauer](#) cũng sử dụng tên này.
- Phần lớn các nước Tây Âu, đều chấp nhận.
  - Tại Anh, sử dụng thuật ngữ ‘**computer science**’, hay ‘**computing science**’,
- Nga cũng chấp nhận tên *informatika* (1966).



# Tin học (informatic)

- Tin học là ngành khoa học nghiên cứu các phương pháp, công nghệ và kỹ thuật xử lý thông tin một cách tự động.
- Công cụ chủ yếu sử dụng trong tin học là máy tính điện tử và một số thiết bị truyền tin.
- Nội dung nghiên cứu của tin học gồm :
  - **Kỹ thuật phần cứng** (Hardware engineering)
  - **Kỹ thuật phần mềm** (Software engineering)

# Công nghệ thông tin: **I**nformation **T**echnology

- Thuật ngữ *Công nghệ thông tin* xuất hiện ở Việt nam vào những năm 90.
- Theo ITAA: *Information Technology Association of America*
  - **CNTT** là ngành nghiên cứu các hệ thống thông tin dựa vào máy tính, đặc biệt là các phần mềm ứng dụng và phần cứng máy tính.
  - **CNTT** xử lý với các máy tính điện tử và các phần mềm máy tính nhằm chuyển đổi, lưu trữ, bảo vệ, truyền tin và trích rút thông tin một cách an toàn.

# Ứng dụng của Công nghệ thông tin

- Các bài toán khoa học kỹ thuật
    - Bài toán phức tạp, cần hàng triệu phép tính/giây
  - Các bài toán quản lý
    - Quản lý thông tin, CSDL, hỗ trợ quyết định
  - Y tế, Giáo dục
    - Học tập trực tuyến, chuẩn đoán bệnh dựa vào AI...
  - Thương mại điện tử
    - Hỗ trợ mua bán, thanh toán qua mạng
  - Các ứng dụng trong đời sống thường ngày
    - Máy móc, đồ điện tử...
  - Giải trí: game, mạng xã hội...
- ➔ *CNTT là thành phần không thể thiếu của xã hội hiện đại*

# Công nghệ thông tin: **I**nformation **T**echnology

- Xu hướng phát triển
  - Sử dụng "information" thay thế cho "data"
  - Mở rộng cho lĩnh vực truyền thông và CNTT trở thành **CNTT&TT**
    - ICT: *Information and Communication Technology*



Trường Đại học Bách Khoa Hà nội

## 2. Biểu diễn dữ liệu trong máy tính

- ❖ Biểu diễn số trong các hệ đếm
- ❖ Biểu diễn dữ liệu trong máy tính và đơn vị thông tin
- ❖ Biểu diễn số nguyên
- ❖ Biểu diễn số thực
- ❖ Biểu diễn ký tự

# Hệ đếm

- Mỗi hệ đếm sử dụng một số ký hiệu (ký tự, chữ số,.. ) hữu hạn
  - Tổng số chữ số của một hệ đếm được gọi là **cơ số** (*base, radix*) của hệ đếm đó, ký hiệu là  $b$ .
  - **Ví dụ:** Hệ đếm cơ số 10,
    - 10 chữ số từ 0 đến 9.

# Hệ đếm

- Trên lý thuyết, có thể biểu diễn một giá trị theo hệ đếm cơ số bất kì.
- Các hệ đếm sử dụng trong máy tính
  - **Hệ cơ số 10 (Hệ thập phân - Decimal System)**  
→ Con người sử dụng
  - **Hệ cơ số 2 (Hệ nhị phân - Binary System)**  
→ Máy tính sử dụng
  - **Hệ cơ số 8 (Octal System)**  
→ Dùng để viết gọn số nhị phân.
  - **Hệ cơ số 16 (Hexadecimal System)**  
→ Dùng để viết gọn số nhị phân

# Hệ thập phân (cơ số $b = 10$ )

- Gồm 10 chữ số: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**
- Dùng  $n$  chữ số thập phân có thể biểu diễn được  $10^n$  giá trị khác nhau:

$$00\dots000 = 0$$

....

$$99\dots999 = 10^n - 1$$

- Một biểu diễn **A** :  **$a_{n-1} \dots a_1 a_0 . a_{-1} a_{-2} \dots a_{-m}$**  sẽ xác định giá trị:

$$A = a_{n-1}10^{n-1} + \dots + a_110^1 + a_010^0 + a_{-1}10^{-1} + \dots + a_{-m}10^{-m}$$

$$= \sum_{i=-m}^{n-1} a_i 10^i$$



# Hệ thập phân (cơ số $b = 10$ )

- Biểu diễn: **5246** có giá trị là

$$5246 = 5 \times 10^3 + 2 \times 10^2 + 4 \times 10^1 + 6 \times 10^0$$

- Biểu diễn: **254.68** có giá trị là

$$254.68 = 2 \times 10^2 + 5 \times 10^1 + 4 \times 10^0 + 6 \times 10^{-1} + 8 \times 10^{-2}$$

# Hệ đếm cơ số b

- **Điều kiện:** b nguyên, và  $b \geq 2$
- Có b chữ số, giá trị từ 0 đến (b-1)
- Số có **n** chữ số có thể nhận  **$b^n$**  giá trị khác nhau
- Biểu diễn số  $A_b$  :  **$a_{n-1} \dots a_1 a_0 \cdot a_{-1} a_{-2} \dots a_{-m}$** 
  - **n** chữ số biểu diễn cho phần nguyên và **m** chữ số biểu diễn cho phần lẻ
  - Cách xác định giá trị số A

$$A = a_{n-1}b^{n-1} + \dots + a_1b^1 + a_0b^0 + a_{-1}b^{-1} + \dots + a_{-m}b^{-m}$$

$$= \sum_{i=-m}^{n-1} a_i b^i$$

# Hệ đếm nhị phân (binary system, $b=2$ )

- Sử dụng 2 chữ số (*nhị phân*): **0,1**
- Chữ số nhị phân gọi là **bit** (binary digit)
  - bit là đơn vị thông tin nhỏ nhất
- Sử dụng  $n$  bit biểu diễn được  $2^n$  giá trị khác nhau
$$00\dots000_2 \Leftrightarrow 0_{10}$$
$$\dots$$
$$11\dots111_2 \Leftrightarrow 2^n-1_{10}$$
  - Ví dụ, sử dụng 3 bit biểu diễn được 8 giá trị khác nhau

000	001	010	011
100	101	110	111

# Hệ đếm nhị phân (binary system, $b = 2$ )

- Biểu diễn A :  $\mathbf{a_{n-1} \dots a_1 a_0 . a_{-1} a_{-2} \dots a_{-m}}$ 
  - $\mathbf{a_i}$  là các chữ số nhị phân (**0,1**), giá trị số A là

$$\begin{aligned} A &= a_{n-1} 2^{n-1} + \dots + a_1 2^1 + a_0 2^0 + a_{-1} 2^{-1} + \dots + a_{-m} 2^{-m} \\ &= \sum_{i=-m}^{n-1} a_i 2^i \end{aligned}$$

Ví dụ: Số nhị phân A:  $1101001.1011_2$  có giá trị

$$\begin{aligned} A &= 2^6 + 2^5 + 2^3 + 2^0 + 2^{-1} + 2^{-3} + 2^{-4} \\ &= 64 + 32 + 8 + 1 + 0.5 + 0.125 + 0.0625 \\ &= \mathbf{105.6875}_{(10)} \end{aligned}$$

# Hệ cơ số 8 (Octal system, $b=8$ )

- Gồm 8 chữ số: **0, 1, 2, 3, 4, 5, 6, 7**
- Dùng  $n$  chữ số có thể biểu diễn được  $8^n$  giá trị khác nhau:

$$00\dots000 = 0 \text{ (giá trị trong hệ 10)}$$

....

$$77\dots777 = 8^n - 1 \text{ (giá trị trong hệ 10)}$$

# Hệ cơ số 8 (Octal system, b=8)

- Biểu diễn A :  $\mathbf{a_{n-1} \dots a_1 a_0 . a_{-1} a_{-2} \dots a_{-m}}$ 
  - $\mathbf{a_i}$  là các chữ số  $\mathbf{(0,1,...,7)}$ , giá trị của A là

$$\begin{aligned} A &= a_{n-1} 8^{n-1} + \dots + a_1 8^1 + a_0 8^0 + a_{-1} 8^{-1} + \dots + a_{-m} 8^{-m} \\ &= \sum_{i=-m}^{n-1} a_i 8^i \end{aligned}$$

Ví dụ: Số A:  $235.64_8$  có giá trị

$$\begin{aligned} A &= 2 \times 8^2 + 3 \times 8^1 + 5 \times 8^0 + 6 \times 8^{-1} + 4 \times 8^{-2} \\ &= 157.8125(10) \end{aligned}$$

# Hệ cơ số 16 (Hexadecimal, b=16)

- Sử dụng 16 chữ số:  
0,1,2,3,4,5,6,7,8,9,A,  
B,C,D,E,F
- Các chữ số  
A, B, C, D, E, F  
nhận giá trị tương  
ứng (trong hệ 10) là  
10, 11, 12, 13, 14,  
15

Hệ thập phân	Hệ nhị phân	Hệ mười sáu
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F



# Hệ cơ số 16 (Hexadecimal, b=16)

- Giả sử có số A được biểu diễn theo hệ 16 như sau:

$$A = a_n a_{n-1} \dots a_1 a_0 . a_{-1} a_{-2} \dots a_{-m}$$

Với  $a_i$  là các chữ số trong hệ 16, khi đó giá trị của A là:

$$A = a_n 16^n + a_{n-1} 16^{n-1} + \dots + a_1 16^1 + a_0 16^0 + a_{-1} 16^{-1} + a_{-2} 16^{-2} + \dots + a_{-m} 16^{-m}$$

$$A = \sum_{i=-m}^n a_i 16^i$$

# Chuyển đổi giữa các hệ đếm

- Hệ đếm bất kỳ ( $b$ )  $\rightarrow$  Hệ thập phân
- Hệ thập phân  $\rightarrow$  Hệ đếm bất kỳ ( $b$ )
- Từ hệ đếm  $b$  sang  $b^k$  và ngược lại ( $b = 2$ )

# Chuyển đổi hệ cơ số **b** → hệ thập phân

- Biểu diễn  $A_b$  :  $\mathbf{a_{n-1} \dots a_1 a_0 . a_{-1} a_{-2} \dots a_{-m}}$ 
  - n chữ số biểu diễn cho phần nguyên
  - m chữ số biểu diễn cho phần lẻ

Giá trị tương đương trong hệ 10 là

$$\begin{aligned} A &= a_{n-1}b^{n-1} + \dots + a_1b^1 + a_0b^0 + a_{-1}b^{-1} + \dots + a_{-m}b^{-m} \\ &= \sum_{i=-m}^{n-1} a_i b^i \end{aligned}$$

$$\text{VD } 11010.11_2 = 2^4 + 2^3 + 2^1 + 2^{-1} + 2^{-2} = 26.75_{10}$$

$$1A.C_{16} = 1 \times 16^1 + 10 \times 16^0 + 12 \times 16^{-1} = 26.75_{10}$$

# Chuyển từ hệ thập phân sang hệ cơ số b

- Trường hợp tổng quát, một số  $N$  trong hệ thập phân ( $N_{(10)}$ ) gồm phần nguyên và phần thập phân.
- Chuyển 1 số từ hệ thập phân sang 1 số ở hệ cơ số  $b$  bất kỳ gồm 2 bước:
  - Đổi phần nguyên (của số đó) từ hệ thập phân sang hệ  $b$
  - Đổi phần thập phân (của số đó) từ hệ thập phân sang hệ cơ số  $b$

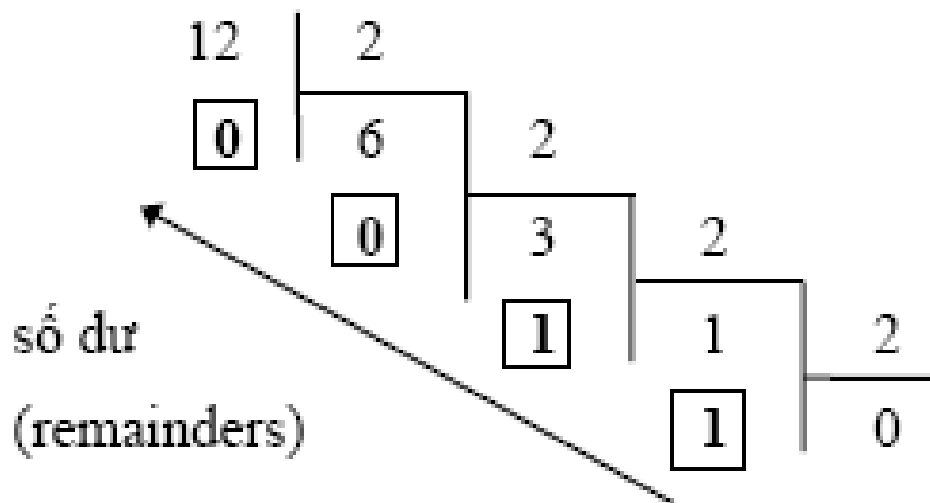
# Chuyển đổi phần nguyên

- **Bước 1:** Lấy phần nguyên của  $N_{(10)}$  chia cho  $b$ , ta được thương là  $T_1$  số **dư**  $d_1$ .
- **Bước 2:** Nếu  $T_1$  khác 0, Lấy  $T_1$  chia tiếp cho  $b$ , ta được thương số là  $T_2$ , số **dư** là  $d_2$   
(Cứ làm như vậy cho tới bước thứ  $n$ , khi ta được  $T_n=0$ )
- **Bước  $n$ :** Nếu  $T_{n-1}$  khác 0, lấy  $T_{n-1}$  chia cho  $b$ , ta được thương số là  $T_n=0$ , số **dư** là  $d_n$
- Kết quả ta được số  $N_{(b)}$  là số tạo bởi các số dư (**được viết theo thứ tự ngược lại**) trong các bước trên

$$\text{Phần nguyên của } N_{(10)} = d_n d_{n-1} \dots d_1 (b)$$

# Chuyển đổi phần nguyên (2)

- Ví dụ: Cách chuyển phần nguyên của số  $12.6875_{(10)}$  sang số trong hệ nhị phân:
  - Dùng phép chia cho 2 liên tiếp thu được một loạt các số dư như sau



Kết quả:  $12_{(10)} = 1100_{(2)}$

# Chuyển đổi phần thập phân

- Bước 1: Lấy phần thập phân của  $N_{(10)}$  nhân với  $b$ , ta được một số có dạng  $x_1.y_1$  ( $x$  là phần nguyên,  $y$  là phần thập phân)
- Bước 2: Nếu  $y_1$  khác 0, tiếp tục lấy  $0.y_1$  nhân với  $b$ , ta được một số có dạng  $x_2.y_2$   
...*(cứ làm như vậy cho đến khi  $y_n=0$ )*
- Bước n: Nếu  $y_{n-1}$  khác 0, nhân  $0.y_{n-1}$  với  $b$ , ta được  $x_n.0$
- Kết quả ta được số sau khi chuyển đổi là:  
**Phần thập phân của  $N_{(10)} = 0.x_1x_2\dots x_n (b)$**



# Chuyển đổi phần thập phân (2)

- Ví dụ: Cách chuyển phần thập phân của số  $12.6875_{(10)}$  sang hệ nhị phân:

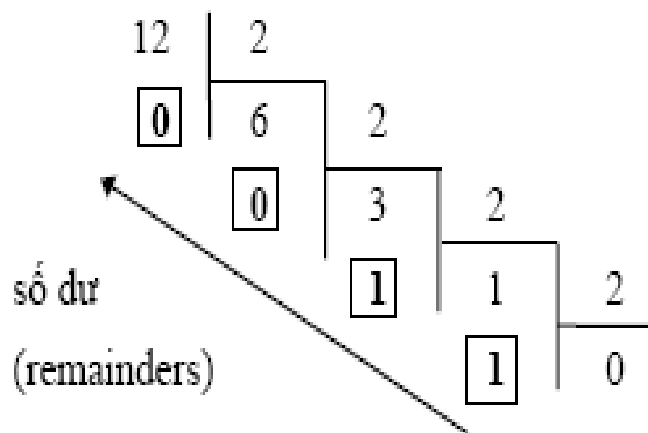
Ví dụ 3.11:  $0.6875_{(10)} = ?_{(2)}$

$0.6875$	$\times 2 =$	$1.375$	← phần nguyên của tích
$0.3750$	$\times 2 =$	$0.75$	← phần thập phân của tích
$0.75$	$\times 2 =$	$1.5$	
$0.5$	$\times 2 =$	$1.0$	

Kết quả:  $0.6875_{(10)} = 0.1011_{(2)}$

# Ví dụ: Chuyển từ thập phân sang nhị phân

- $12.6875_{(10)} = 1100.1011_{(2)}$



Kết quả:  $12_{(10)} = 1100_{(2)}$

$0.6875_{(10)} = ?_{(2)}$

0.6875	x 2	=	1	375
0.3750	x 2	=	0	.75
0.75	x 2	=	1	.5
0.5	x 2	=	1	.0

phần nguyên của tích

phần thập phân của tích

$0.6875_{(10)} = 0.1011_{(2)}$

- $69.25_{(10)} = ?_{(2)}$

## Cách 2: Tính nhẩm khi chuyển về hệ 2

- Phân tích số đó thành tổng các lũy thừa của 2, sau đó dựa vào các số mũ để xác định dạng biểu diễn nhị phân

→ Nhanh hơn với các giá trị nhỏ

- Ví dụ:  $69.25_{(10)} = 64 + 4 + 1 + \frac{1}{4}$   
 $= 2^6 + 2^2 + 2^0 + 2^{-2}$   
 $= 1000101.01_{(2)}$

# Chuyển giữa hệ 2, 8, và 16

- Chuyển về số trung gian ở hệ cơ số 2
- Mỗi chữ số trong hệ 8 hoặc 16 tương đương một nhóm gồm 3 hoặc 4 chữ số hệ 2

# Ví dụ

- Nhị phân  $\rightarrow$  Hexa:  $11\ 1011\ 1110\ 0110_{(2)} = ?$

$$11\ 1011\ 1110\ 0110_{(2)} = 3BE6_{(16)}$$

- Hexa  $\rightarrow$  Nhị phân:  $AB7_{(16)} = ?$

$$AB7_{(16)} = 1010\ 1011\ 0111_{(2)}$$

- Hexa  $\rightarrow$  Thập phân:  $3A8C \rightarrow ?$

$$3A8C_{(16)} = 3 \times 16^3 + 10 \times 16^2 + 8 \times 16^1 + 12 \times 16^0$$

$$= 12288 + 2560 + 128 + 12$$

$$= 14988_{(10)}$$

# Một số ví dụ (tiếp)

- Thập phân  $\rightarrow$  Hexa:  $14988 \rightarrow ?$   
 $14988 : 16 = 936$  dư  $12$  tức là C  
 $936 : 16 = 58$  dư  $8$   
 $58 : 16 = 3$  dư  $10$  tức là A  
 $3 : 16 = 0$  dư  $3$   
Như vậy, ta có:  $14988_{(10)} = 3A8C_{(16)}$

# Bài tập

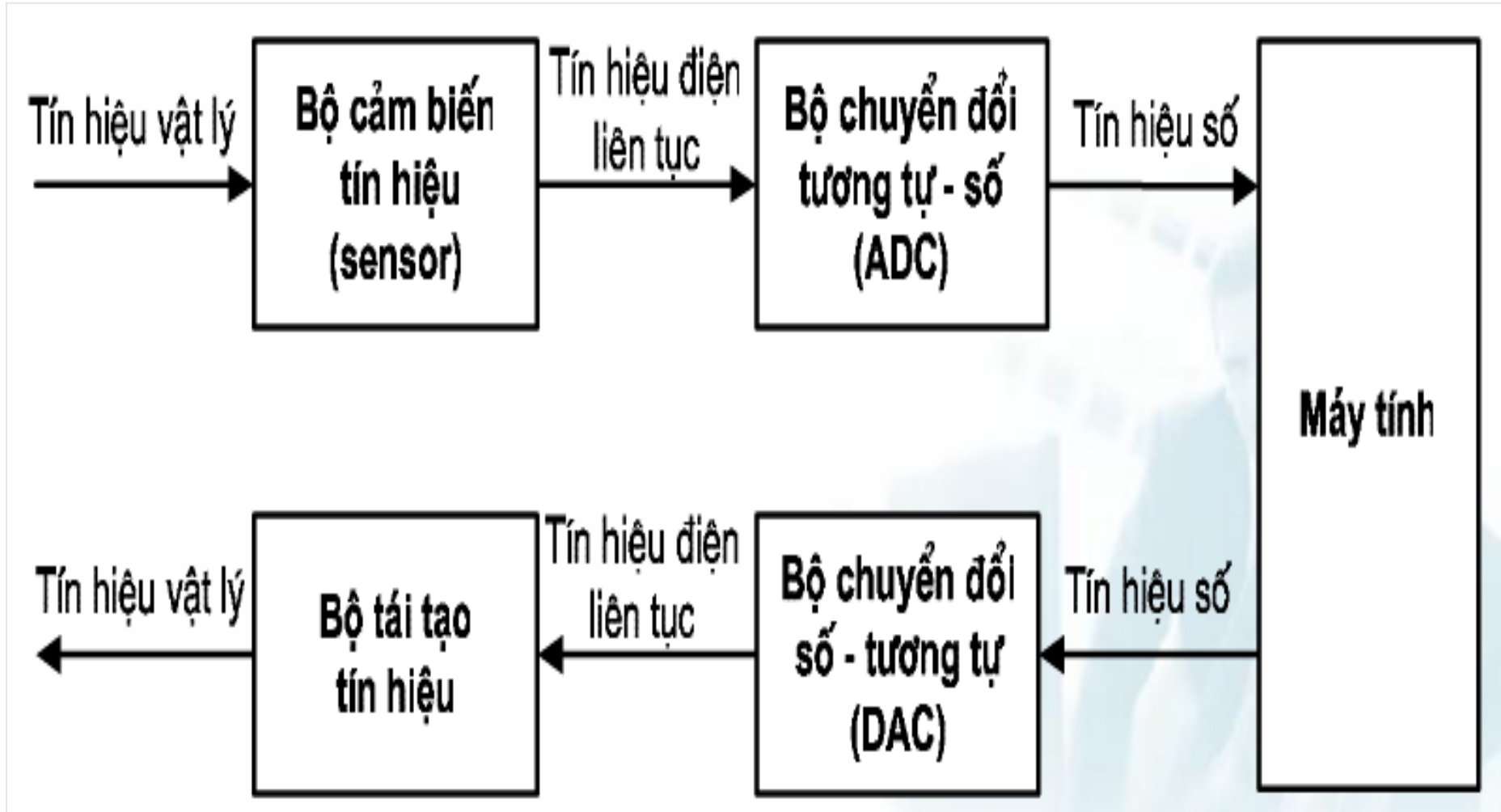
- Chuyển sang hệ nhị phân
  - 124.75
  - 65.125



# Biểu diễn dữ liệu trong máy tính

- Dữ liệu đưa vào máy tính phải được mã hóa thành số nhị phân
- Các loại dữ liệu:
  - Dữ liệu nhân tạo: Do con người quy ước
  - Dữ liệu tự nhiên:
    - Tồn tại khác quan với con người.
    - Phổ biến là các tín hiệu vật lý: âm thanh, hình ảnh,...
- Nguyên tắc mã hóa dữ liệu
  - Dữ liệu nhân tạo:
    - Dữ liệu số: Mã hóa theo các chuẩn quy ước
    - Dữ liệu ký tự: Mã hóa theo bộ mã ký tự
  - Dữ liệu tự nhiên:
    - Cần phải số hóa trước khi đưa vào máy tính

# Sơ đồ mã hóa và tái tạo tín hiệu vật lý



# Dữ liệu trong máy tính

- Dữ liệu cơ bản
  - Số nguyên, số thực, ký tự
- Dữ liệu có cấu trúc
  - Mảng, cấu trúc ký tự, tập hợp, bản ghi

# Dữ liệu cơ bản

- Số nguyên:
  - Không dấu: Biểu diễn theo mã nhị phân
  - Có dấu: Biểu diễn dưới dạng mã bù hai.
- Số thực:
  - Biểu diễn bằng số dấu chấm (phẩy) động.
- Ký tự:
  - Biểu diễn bằng mã ký tự trên các bộ mã ký tự.

## Độ dài từ dữ liệu:

- Số bit được sử dụng để mã hóa loại dữ liệu tương ứng
- Thực tế, độ dài từ dữ liệu thường là bội số của 8.

# Dữ liệu có cấu trúc

- Là tập hợp các loại dữ liệu cơ bản được cấu thành theo một cách nào đó.
  - Ví dụ: kiểu dữ liệu mảng, kiểu xâu ký tự, kiểu tập hợp, bản ghi,...
  - *Các dữ liệu có cấu trúc sẽ được nghiên cứu cụ thể trong phần học về ngôn ngữ lập trình.*

# Đơn vị đo thông tin

- Bit (**B**inary digi**T**)
  - Là đơn vị thông tin nhỏ nhất
  - Nhận một trong hai giá trị nhị phân 0/1
- Byte (**B**): Chuỗi 8 bit
- Các đơn vị dẫn xuất
  - Kilobyte (**KB**):  $1 \text{ KB} = 2^{10} \text{ B} = 1024 \text{ B}$
  - Megabyte (**MB**):  $1 \text{ MB} = 2^{10} \text{ KB} = 2^{20} \text{ B}$
  - Gigabyte (**GB**):  $1 \text{ GB} = 2^{10} \text{ MB} = 2^{20} \text{ KB} = 2^{30} \text{ B}$
  - Terabyte (**TB**):  $1 \text{ TB} = 2^{10} \text{ GB} = \dots = 2^{40} \text{ B}$
  - Petabyte (**PB**):  $1 \text{ PB} = 2^{10} \text{ TB} = \dots = 2^{50} \text{ B}$

# Số nguyên không dấu

- Dùng  $n$  bit để biểu diễn cho một số nguyên không dấu **A**:  $a_{n-1}a_{n-2}\dots a_2a_1a_0$ 
  - Trong đó  $a_i$  là các chữ số nhị phân (0,1)
- Giá trị của  $A$  được tính theo công thức

$$\begin{aligned} A &= a_{n-1}2^{n-1} + \dots + a_12^1 + a_02^0 \\ &= \sum_{i=0}^{n-1} a_i 2^i \end{aligned}$$

- Dải giá trị của  $A$ :  $[0..2^n-1]$



# Ví dụ 1

Tìm biểu diễn của các số nguyên không dấu 8 bit có giá trị sau

- **A = 45**

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

$$\rightarrow A = 0010\ 1101$$

- **B = 156**

$$\begin{aligned} B = 156 &= 128 + 16 + 8 + 4 \\ &= 2^7 + 2^4 + 2^3 + 2^2 \end{aligned}$$

$$\rightarrow B = 1001\ 1100$$

# Ví dụ 2

Tính giá trị các số nguyên không dấu 8 bit có biểu diễn nhị phân như dưới đây

$$\begin{aligned} X = 0010\ 1011 &= 2^5 + 2^3 + 2^1 + 2^0 \\ &= 32 + 8 + 2 + 1 = 43 \end{aligned}$$

$$\begin{aligned} Y = 1001\ 0110 &= 2^7 + 2^4 + 2^2 + 2^1 \\ &= 128 + 16 + 4 + 2 = 150 \end{aligned}$$

# Phạm vi biểu diễn: với $n = 8$ bit

- Dải biểu diễn là  $[0, 255]$

$$0000\ 0000 = 0$$

$$0000\ 0001 = 1$$

$$0000\ 0010 = 2$$

$$0000\ 0011 = 3$$

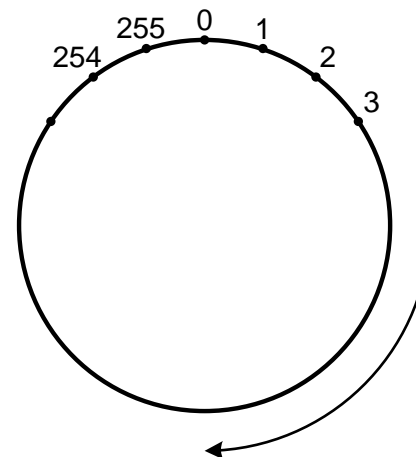
.....

$$1111\ 1111 = 255$$

- Trục số học tự nhiên:



- Trục số học máy tính:



# Phạm vi biểu diễn

- $n = 16$  bit:
  - Dải biểu diễn là  $[0, 65535]$
- $n = 32$  bit:
  - Dải biểu diễn là  $[0, 2^{32}-1]$
- $n = 64$  bit:
  - Dải biểu diễn là  $[0, 2^{64}-1]$

# Số nguyên có dấu

- Số nguyên có dấu được dùng để biểu diễn số nguyên với cả giá trị âm và dương
- Sử dụng bit đầu tiên làm bit dấu:
  - Bit 0 cho số dương
  - Bit 1 cho số âm
- Biểu diễn số âm bằng số bù 2 của số dương tương ứng

# Số bù một và số bù hai

- Giả sử có một số nhị phân được biểu diễn bởi  $n$  bit.
- Định nghĩa:
  - Số bù một của  $A = (2^n - 1) - A$
  - Số bù hai của  $A = 2^n - A$
  - NX: Số bù hai = Số bù một + 1
- Ví dụ
  - Xét  $n = 4$  bit,  $A = 0110$
  - Số bù một của  $A = (2^4 - 1) - 0110 = 1001$
  - Số bù hai của  $A = 2^4 - 0110 = 1010$

# Nhận xét

- Xét  $n = 4$  bit,  $A = 0110$
- Số bù một của  $A = (2^4 - 1) - 0110 = 1001$
- Số bù hai của  $A = 2^4 - 0110 = 1010$

→ Có thể tìm số bù một của  $A$  bằng cách đảo ngược tất cả các bit

→ Số bù hai = Số bù một + 1

→  $A + \text{Số bù hai của } A = 0$  nếu bỏ qua bit nhớ ra khỏi bit cao nhất

# Biểu diễn số nguyên có dấu

- Biểu diễn số âm bằng số bù 2
- Ví dụ: Biểu diễn số nguyên có dấu sau đây bằng 8 bit nhị phân:

$$A = -70_{(10)}$$

Biểu diễn  $70 = 0100\ 0110$

$$\begin{array}{r} \text{Bù 1:} \quad 1011\ 1001 \text{ (nghịch đảo các bit)} \\ + \quad \quad 1 \\ \hline \end{array}$$

$$\text{Bù 2:} \quad 1011\ 1010$$

$$\text{Vậy: } A = 1011\ 1010_{(2)}$$



# Biểu diễn số nguyên có dấu (tiếp)

- Dạng tổng quát của số nguyên có dấu A:

$$A = a_{n-1}a_{n-2}\dots a_2a_1a_0$$

- Giá trị của A được xác định như sau:

$$A = -a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i$$

- Dải biểu diễn:  $[-2^{n-1}, 2^{n-1}-1]$

# Ví dụ

- Xác định giá trị của các số nguyên có dấu 8 bit sau đây:

$$A = 0101\ 0110$$

$$B = 1101\ 0010$$

Giải:

$$A = 2^6 + 2^4 + 2^2 + 2^1 = 64 + 16 + 4 + 2 = +86$$

$$\begin{aligned} B &= -2^7 + 2^6 + 2^4 + 2^1 \\ &= -128 + 64 + 16 + 2 = -46 \end{aligned}$$

# Phạm vi biểu diễn: với $n = 8$ bit

- Dải biểu diễn là  $[-128, 127]$

$$0000\ 0000 = 0$$

$$0000\ 0001 = +1$$

$$0000\ 0010 = +2$$

.....

$$01111111 = +127$$

$$10000000 = -128$$

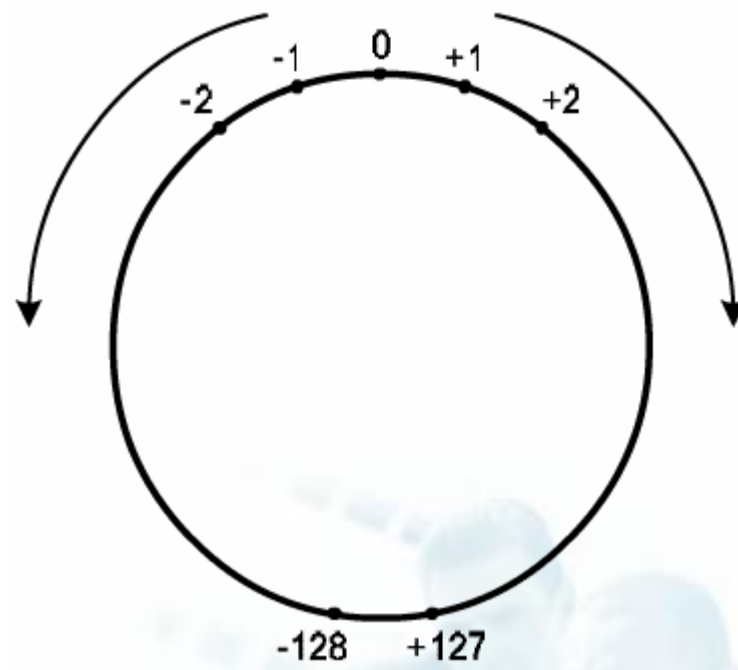
$$10000001 = -127$$

.....

$$1111\ 1110 = -2$$

$$1111\ 1111 = -1$$

- Trục số học máy tính



# Tính toán số học với số nguyên

- Cộng trừ số nguyên không dấu
- Cộng/trừ số nguyên có dấu
- Nhân chia các số nguyên không dấu
- Các phép toán logic với số nhị phân

# Tính toán trong hệ nhị phân

- Phép cộng:

$$1+0=0+1=1;$$

$$0+0=0;$$

$$1+1=10;$$

- Phép trừ:

$$0-1=1; \text{ (vay 1)}$$

$$1-1=0;$$

$$0-0=0;$$

$$1-0=1$$

$$\begin{array}{r} 110 \\ + 111 \\ \hline 1101 \end{array}$$
$$\begin{array}{r} 1101 \\ - 111 \\ \hline 110 \end{array}$$

# Cộng/ trừ số nguyên không dấu

- Tiến hành cộng/trừ lần lượt từng bit từ phải qua trái.
- Khi cộng/trừ hai số nguyên không dấu  $n$  bit ta thu được một số nguyên không dấu  $n$  bit.
  - Nếu tổng của hai số đó lớn hơn  $2^n - 1$  thì khi đó sẽ tràn số và kết quả sẽ là sai.
  - Trừ số không dấu thì ta chỉ trừ được số lớn cho số nhỏ. Trường hợp ngược lại sẽ sai

# Ví dụ

- Thực hiện phép toán với nguyên không dấu 8 bit
- Trường hợp không xảy ra tràn số (carry-out):

$$X = 1001\ 0110 = 150$$

$$Y = 0001\ 0011 = 19$$

$$S = 1010\ 1001 = 169$$

$$\text{Cout} = 0$$

- Trường hợp có xảy ra tràn số (carry-out):

$$X = 1100\ 0101 = 197$$

$$Y = 0100\ 0110 = 70$$

$$S = 0000\ 1011 \neq 267$$

$$\text{Cout} = 1 \rightarrow \text{carry-out}$$

- (KQ sai =  $2^3 + 2^1 + 2^0 = 11$ )

# Cộng số nguyên có dấu

- Cộng lần lượt các cặp bit từ phải qua trái, bỏ qua bit nhớ cao nhất (nếu có).
- Cộng hai số khác dấu: kết quả luôn đúng
- Cộng hai số cùng dấu:
  - Nếu tổng nhận được cùng dấu với 2 số hạng thì kết quả là đúng
  - Nếu tổng nhận được khác dấu với 2 số hạng thì đã xảy ra hiện tượng *tràn số học* (overflow) và kết quả nhận được là sai



# Ví dụ: Cộng/trừ số nguyên có dấu

- VD: không tràn số

$$X = 0100 \ 0110 = +70$$

$$+ \underline{Y = 0010 \ 1010 = +42}$$

$$S = 0111 \ 0000 = +112$$

$$X = 0110 \ 0001 = +97$$

$$+ \underline{Y = 1100 \ 1100 = -52}$$

$$S = 0010 \ 1101 = +45$$

$$C_{out} = 1 \rightarrow \text{bỏ qua}$$

$$X = 1010 \ 0110 = -90$$

$$+ \underline{Y = 0010 \ 0100 = +36}$$

$$S = 1100 \ 1010 = -54$$

$$X = 1011 \ 0110 = -74$$

$$+ \underline{Y = 1110 \ 0010 = -30}$$

$$S = 1001 \ 1000 = -104$$

$$C_{out} = 1 \rightarrow \text{bỏ qua}$$

# Ví dụ: Cộng/trừ số nguyên có dấu

- Có xảy ra tràn số:

$$\begin{array}{r} X = 0100 \ 1011 = +75 \\ + Y = 0101 \ 0001 = +81 \\ \hline S = 1001 \ 1100 \neq +156 \\ (S = -2^7 + 2^4 + 2^3 + 2^2 = -100) \end{array}$$

$$\begin{array}{r} X = 1001 \ 1000 = -104 \\ + Y = 1011 \ 0110 = -74 \\ \hline S = 0100 \ 1110 \neq -178 \\ C_{out} = 1 \rightarrow \text{bỏ qua} \\ (S = 2^6 + 2^3 + 2^2 + 2^1 = 78) \end{array}$$

# Trừ số nguyên có dấu

- Nhận xét

$$X - Y = X + (-Y)$$

- Kết quả chính là tổng X với bù 2 của Y
- Ví dụ:

$$97 - 52 = 97 + (-52)$$

$$\begin{array}{rcl} + & 97 & = & 0110\ 0001 \\ & -52 & = & 1100\ 1100 \end{array}$$

---

$$45 = \textcircled{1}0010\ 1101$$

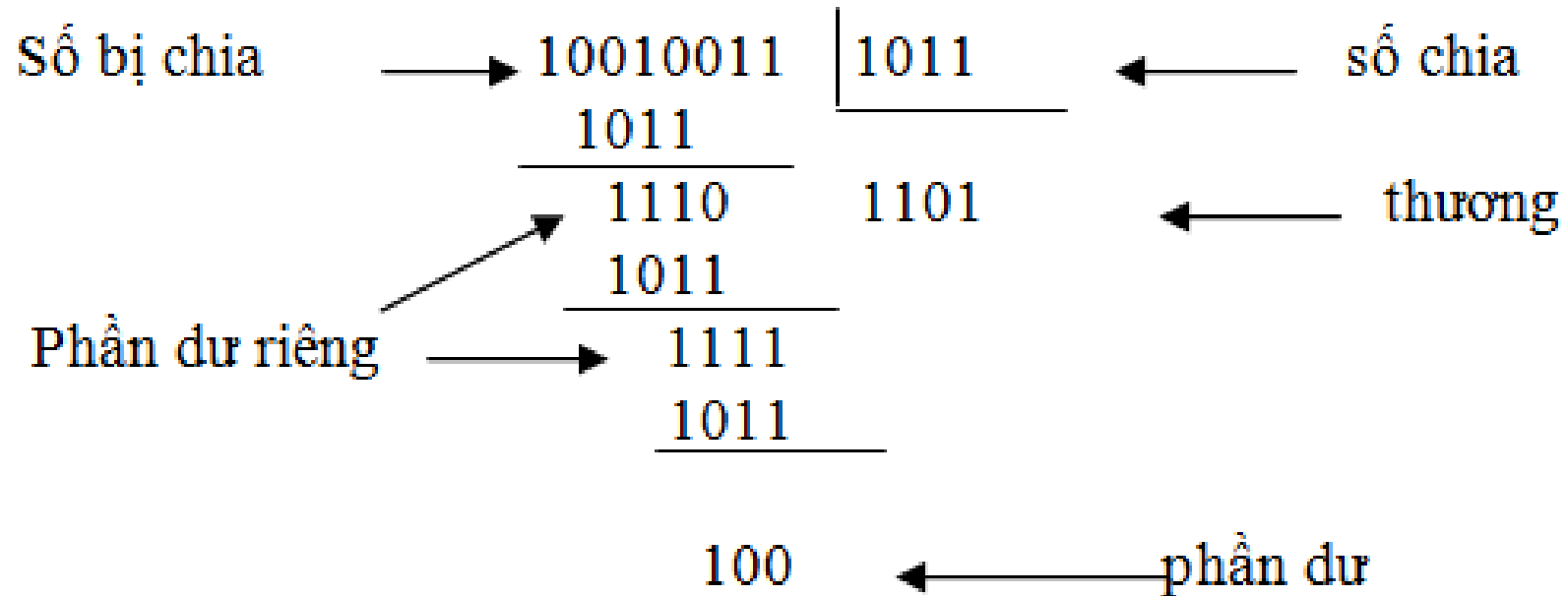
Bỏ qua

# Nhân/chia số nguyên không dấu

- Các bước thực hiện như trong hệ 10
- VD: Phép nhân

$$\begin{array}{r} 1011 \quad (11 \text{ cơ số } 10) \\ \times 1101 \quad (13 \text{ cơ số } 10) \\ \hline 1011 \\ 0000 \\ 1011 \\ 1011 \\ \hline 10001111 \quad (143 \text{ cơ số } 10) \end{array}$$

# Phép chia số nguyên không dấu



# Nhân/chia số nguyên có dấu

- Bước 1: Chuyển đổi thành số dương tương ứng
- Bước 2: Thực hiện nhân/chia với số nguyên không dấu
- Bước 3: Hiệu chỉnh dấu của kết quả.

# Các phép toán logic với số nhị phân

		<b>AND</b>	<b>OR</b>	<b>XOR</b>
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

	<b>NOT</b>
0	1
1	0

- Thực hiện các phép toán logic với 2 số nhị phân:
  - Thực hiện các phép toán logic với từng cặp bit của 2 số
    - Các phép toán logic chỉ tác động lên từng cặp bit, không ảnh hưởng đến bit khác.
  - Kết quả là 1 số nhị phân

## Ví dụ

		NOT
A	1010 1010	01010101
B	0000 1111	11110000
AND	<b>00001010</b>	
OR	<b>10101111</b>	
XOR	<b>10100101</b>	

- Có thể dùng phép **AND** để xóa một số bit và giữ nguyên các bit còn lại (Ví dụ: **AND 11011011**)
- Có thể dùng phép **OR** để thiết lập 1 số bit và giữ nguyên các bit khác. (Ví dụ: **OR 00100100**)



# Biểu diễn số thực

- Số thực trong máy tính được biểu diễn thông qua ký pháp dấu phẩy động (*Floating Point Number*)
- Một số thực  $V$  được biểu diễn theo dạng dấu phẩy động như sau:  $V = M * R^E$

Trong đó:

- $R$  là cơ số (*Radix*) thường là 2 hoặc 10.
- $M$  là phần định trị (*Mantissa*) thỏa mãn  $-R < M < R$
- $E$  là phần mũ (*Exponent*)
- Nhận xét:
  - Nếu  $R$  cố định, để lưu trữ  $V$  chỉ cần lưu trữ  $M$  và  $E$  (*dưới dạng số nguyên*)

# Ví dụ

Với cơ số  $R = 10$

$$V = -52.5 = -5.2500 \times 10^1$$

- Phần định trị:  $M = -5.2500$  / Phần số mũ:  $E = 1$

## Ghi chú

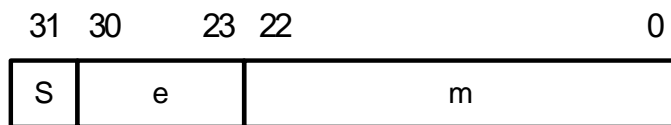
- Để chuẩn hóa, phần định trị luôn nằm trong khoảng  $(-R, R)$  và số đầu tiên là khác 0 (*nếu được*)
  - $-52.5 \times 10^0$  và  $-0.5250 \times 10^2$  tương đương với  $-5.25 \times 10^1$  nhưng không phải số chuẩn hóa
  - Việc chuẩn hóa nhằm duy trì độ chính xác và biểu diễn các số rất lớn/bé
    - $0.000162$  biểu diễn với 4 chữ số thập phân  $\Rightarrow 0.0001$
    - $\pm 9.999 \times 10^9 = \pm 9\,999\,000\,000$
    - $\pm 0.001 \times 10^{-9} = \pm 0.000\,000\,000\,001$

# Phép toán với dấu phẩy động

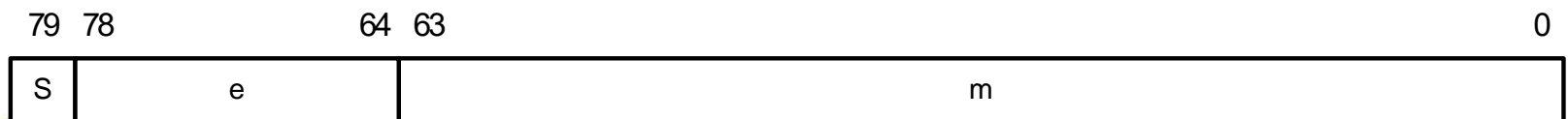
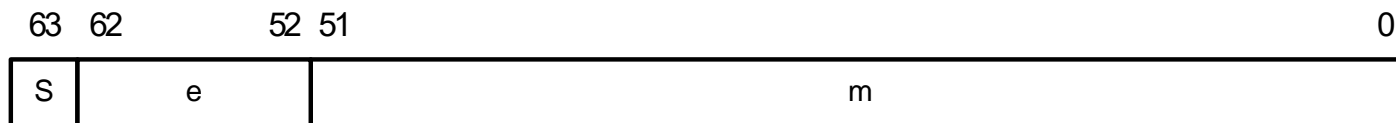
- Được tiến hành trên cơ sở các giá trị của phần định trị và phần mũ
- Ví dụ, cho  $R_1$  và  $R_2$  là số dấu phẩy động:
  - $R_1 = M_1 \times R^{E_1}$  và  $R_2 = M_2 \times R^{E_2}$
- Các phép toán số học thực hiện như sau:
  - $R_1 \pm R_2 = (M_1 \times R^{E_1-E_2} \pm M_2) \times R^{E_2}$ ,
  - Giả thiết  $E_1 \geq E_2$
  - $R_1 \times R_2 = (M_1 \times M_2) \times R^{E_1+E_2}$
  - $R_1 / R_2 = (M_1 / M_2) \times R^{E_1-E_2}$

# Chuẩn IEEE 754/85

- Là chuẩn mã hóa số dấu phẩy động
- Cơ số  $R = 2$
- Các dạng cơ bản:
  - Dạng có độ chính xác đơn, 32-bit
  - Dạng có độ chính xác kép, 64-bit
  - Dạng có độ chính xác kép mở rộng, 80-bit



Khuôn dạng mã hóa



# Chuẩn IEEE 754/85

- **S** là bit dấu,
  - $S = 0$  đó là số dương/  $S = 1$  đó là số âm.
- **e** là mã lệch (excess) của phần mũ E

$$E = e - b$$

Trong đó b là độ lệch (bias):

- Dạng 32-bit :  $b = 127$ , hay  $E = e - 127$
- Dạng 64-bit :  $b = 1023$ , hay  $E = e - 1023$
- Dạng 80-bit :  $b = 16383$ , hay  $E = e - 16383$
- **m** là các bit phần lẻ của phần định trị M
  - Phần định trị được ngầm định:  $M = 1.m$

# Chuẩn IEEE 754/85



$$R = (-1)^s \times 1.m \times 2^{e-b}$$

# Ví dụ

- Tìm giá trị số thực R biết biểu diễn của R theo chuẩn IEEE 754 dạng 32 bit như sau

1100 0001 0101 0110 0000 0000 0000 0000

$S = 1 \rightarrow R$  là số âm

- $e = 1000\ 0010 = 130$ 
  - Số 32 bit:  $b = 127 \Rightarrow E = 130 - 127$
- $m = 10101100\dots00 \Rightarrow M = 1.m$
- Vậy  $R = (-1)^1 \times 1.10101100\dots00 \times 2^{130-127}$   
 $= -1.101011 \times 2^3 = -1101.011 = -13.375$

# Ví dụ

- Xác định giá trị thập phân của số thực X có dạng biểu diễn theo chuẩn IEEE 754 dạng 32 bit như sau:

0011 1111 1000 0000 0000 0000 0000 0000

- Giải:

- $S = 0 \rightarrow X$  là số dương
- $e = 0111\ 1111 = 127$
- $m = 000000...00$
- Vậy  $X = (-1)^0 \times 1.0000...00 \times 2^{127-127}$   
 $= 1.0 \times 2^0 = 1$



# Ví dụ

- Biểu diễn số thực  $X = 9.6875$  về dạng số dấu chấm động theo chuẩn IEEE 754 dạng 32 bit

- Giải:

$$X = 9.6875_{(10)} = 1001.1011_{(2)} = 1.0011011 \times 2^3$$

Theo chuẩn IEEE754:

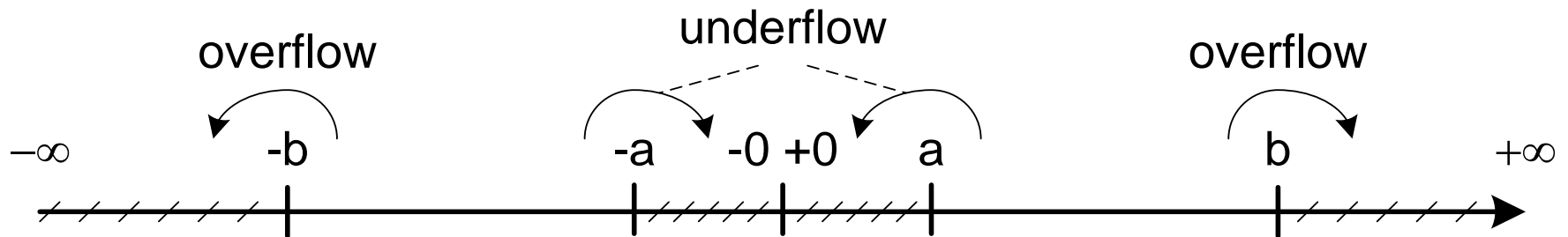
- $S = 0$  vì đây là số dương
- $E = e - 127$  nên  $e = 127 + 3 = 130_{(10)}$   
 $= 1000\ 0010_{(2)}$
- $m = 001101100\dots00$  (23 bit)

$$X = 0\mathbf{100}\ \mathbf{0001}\ 0001\ 1011\ 0000\ 0000\ 0000\ 0000$$
$$= 411B0000_{(16)}$$

# Một số quy ước đặc biệt

- Nếu tất cả các bit của  $e$  đều bằng 0, các bit của  $m$  đều bằng 0, thì  $X = \pm 0$
- Nếu tất cả các bit của  $e$  đều bằng 1, các bit của  $m$  đều bằng 0, thì  $X = \pm \infty$
- Nếu tất cả các bit của  $e$  đều bằng 1,  $m$  có ít nhất một bit bằng 1, thì  $X$  không phải là số (not a number - NaN)

# Phạm vi biểu diễn



- Dạng 32 bit:  $a = 2^{-127} \approx 10^{-38}$        $b = 2^{+127} \approx 10^{+38}$
- Dạng 64 bit:  $a = 2^{-1023} \approx 10^{-308}$        $b = 2^{+1023} \approx 10^{+308}$
- Dạng 80 bit:  $a = 2^{-16383} \approx 10^{-4932}$        $b = 2^{+16383} \approx 10^{+4932}$

$$X = (-1)^s \times 1.m \times 2^{e-127}$$

# Chú ý

- Khoảng cách giữa 2 số thực liên tiếp ?
  - Sai số máy (Machine epsilon)
- Sai số làm tròn
  - $0.7 = ?$ 
    - $0.1011001100 = 0.69921875$ 
      - $0.101100110011001100 = 0.69999980926..$
    - $0.1011001101 = 0.700195312$

# Biểu diễn ký tự

- Các ký tự cần được chuyển đổi thành chuỗi bit nhị phân gọi là **mã ký tự**.
- Số bit dùng cho mỗi ký tự theo các mã khác nhau là khác nhau.

Ví dụ: Bộ mã ASCII dùng 8 bit cho mỗi ký tự.

**ASCII: American Standard Codes for  
Information Interchangeable**

# Bộ mã ASCII

- Do ANSI (**A**merican **N**ational **S**tandard **I**nstitute) thiết kế
- Là bộ mã được dùng để *trao đổi thông tin chuẩn của Mỹ*.
  - Lúc đầu chỉ dùng 7 bit (128 ký tự) sau đó mở rộng cho 8 bit và có thể biểu diễn 256 ký tự khác nhau trong máy tính
- Bộ mã 8 bit  $\rightarrow$  mã hóa được cho  $2^8 = 256$  kí tự, có mã từ  $00_{16} \div FF_{16}$ , bao gồm:
  - 128 kí tự chuẩn có mã từ  $00_{16} \div 7F_{16}$
  - 128 kí tự mở rộng có mã từ  $80_{16} \div FF_{16}$

# ASCII Chart

	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	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SPC	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
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	{		}	~	DEL

Physical Device Controls: Format Effectors

# Bộ mã ASCII

- **95** kí tự hiển thị được: có mã từ  $20_h \div 7E_h$ 
  - 26 chữ cái Latin hoa 'A'  $\div$  'Z', mã từ  $41_h \div 5A_h$
  - 26 chữ cái Latin thường 'a'  $\div$  'z', mã từ  $61_h \div 7A_h$
  - 10 chữ số thập phân '0'  $\div$  '9', mã từ  $30_h \div 39_h$
  - Các dấu câu: . , ? ! : ; ...
  - Các dấu phép toán: + - \* / ...
  - Một số kí tự thông dụng: #, \$, &, @, ...
  - Dấu cách (mã là  $20_h$ )
- **33** mã điều khiển:  $00_h \div 1F_h$  và  $7F_h$ 
  - Điều khiển định dạng, phân cách thông tin..



# Các kí tự mở rộng của bảng mã ASCII

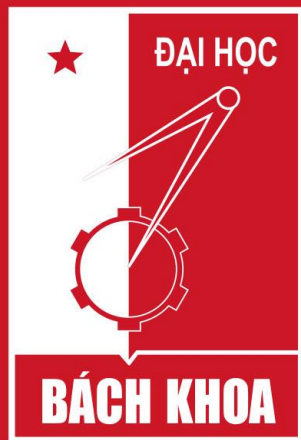
- Được định nghĩa bởi:
  - Nhà chế tạo máy tính
  - Người phát triển phần mềm

## Ví dụ:

- Bộ mã ký tự mở rộng của IBM: được dùng trên máy tính IBM-PC.
- Bộ mã ký tự mở rộng của Apple: được dùng trên máy tính Macintosh.
- Các nhà phát triển phần mềm tiếng Việt cũng đã thay đổi phần này để mã hoá cho các ký tự riêng của chữ Việt, ví dụ như bộ mã TCVN 5712.

# Bộ mã Unicode

- Unicode: Mã “thống nhất”
  - Quản lý bởi Unicode Consortium gồm các hãng máy tính và phần mềm lớn (và Univ. of California, Berkeley)
  - Hiện có hơn 100000 ký tự
- Quy định nhiều chế độ mã hóa, cho phép sử dụng 8/16/24/32 bit để biểu diễn một ký tự
- Các chế độ mã hóa như UTF16, UTF32 cho phép mã hóa số lượng ký tự lớn → hỗ trợ nhiều ngôn ngữ trong cùng bảng mã



25 YEARS ANNIVERSARY  
SOICT

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY