

# Tin học đại cương và ứng dụng

*Máy tính và biểu diễn thông tin trong máy tính*

---

Đặng Xuân Hà

Department of Computer Science

Hanoi Agricultural University

Office location: 3<sup>rd</sup> floor, Administrative building

Office phone: 8276346, Ext: 132

Email: [dxha@hau1.edu.vn](mailto:dxha@hau1.edu.vn)

Website: <http://www.hau1.edu.vn/cs/dxha>

# Ch1. Máy tính và biểu diễn thông tin trong máy tính

1. Thông tin và tin học
2. Lịch sử máy tính
3. Phân loại máy tính
4. Các hệ đếm
5. Biểu diễn thông tin trong máy tính
6. Đại số logic
7. Bài tập

# 1. Thông tin và tin học

- Dữ liệu: chưa mang lại hiểu biết về đối tượng
- Thông tin: dữ liệu sau khi được xử lý, cho ta hiểu biết về đối tượng
- Ví dụ
  - Ảnh mây vệ tinh: Dữ liệu
  - Bản tin dự báo thời tiết: Thông tin
- Tin học
  - Ngành khoa học nghiên cứu các vấn đề thu thập và xử lý dữ liệu để có được thông tin mong muốn, sử dụng máy tính như một công cụ hỗ trợ chính.

## 2. Lịch sử máy tính {1}

- 1937, Turing, khái niệm về các con số tính toán và máy Turing.
- 1943-1946, **ENIAC**
  - Máy tính điện tử đa chức năng đầu tiên.
  - J.Mauchly & J.Presper Eckert.
- 1945, John Von Neumann đưa ra khái niệm về chương trình được lưu trữ.
- 1952, Neumann **IAS parallel-bit machine.**



ENIAC



Newman & IAS

## 2. Lịch sử máy tính {2}

- 1945 – 1954, thế hệ 1 (first generation)
  - Bóng đèn chân không (vacuum tube)
  - Bìa đục lỗ
  - ENIAC: 30 tấn, 18.000 bóng đèn, 100.000 phép tính/giây.
- 1955-1964, thế hệ 2
  - Transistor
  - Intel transistor processor
- 1965-1974, thế hệ 3
  - Mạch tích hợp (Integrated Circuit – IC)
- 1975, Thế hệ 4
  - LSI (Large Scale Integration), VLSI (Very LSI), ULSI (Ultra LSI).

# 3. Phân loại máy tính

- Personal Computer (PC)/Microcomputer
- Minicomputer
  - Nhanh hơn PC 3-10 lần
- Mainframe
  - Nhanh hơn PC 10-40 lần
- Supercomputer
  - Nhanh hơn PC 50-1.500 lần
  - Phục vụ nghiên cứu là chính
  - *VD:Earth Simulator (NEC, 5104 CPUs, 35.600 GF).*
- Laptop Computer
- Handheld Computer: Pocket PC, Palm, Mobile devices.



PC



Mini



Super



Mainframe



Laptop



Handheld

## 4. Hệ đếm

- Khái niệm
- Hệ đếm cơ số 10
- Hệ đếm cơ số bất kỳ
- Hệ đếm cơ số 2
- Hệ đếm cơ số 16

# 4.1. Khái niệm

## ■ Hệ đếm

- Sử dụng để đếm (biểu diễn thông tin số)
- Cơ số: Số lượng ký hiệu

## ■ Ví dụ: hệ đếm cơ số 10

- 10 ký hiệu (cơ số 10) : 0..9.
- 123789 là một số trong hệ 10.

## ■ Hệ đếm cơ số $a$

- Có  $a$  ký hiệu.



## 4.2. Hệ đếm cơ số 10

- Cơ số 10

- 10 ký hiệu: 0,1,2,...,9

- $a_n a_{n-1} \dots a_0 = a_n \cdot 10^n + a_{n-1} \cdot 10^{n-1} + \dots + a_0 \cdot 10^0$

- $123 = 1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0$

- Viết: 2004 hoặc  $2004_{10}$

## 4.3. Hệ đếm cơ số a bất kỳ

- Sử dụng a ký hiệu để biểu diễn
- Ký hiệu có giá trị nhỏ nhất là '0'
- Ký hiệu có giá trị lớn nhất là  $a-1$
- Giá trị của chữ số thứ  $n$  bằng số đó nhân với giá trị của vị trí
  - Giá trị của vị trí  $= a^n$
  - $n =$  vị trí, chữ số đầu tiên có vị trí là  $n-1$
  - Phần thập phân được đánh số âm

## 4.4. Hệ đếm cơ số 2

- Sử dụng 2 ký hiệu 0 và 1
- Binary (nhị phân)
- Biểu diễn thông tin trong máy tính
  - Các linh kiện điện tử chỉ có hai trạng thái:
    - Đóng hoặc mở (công tắc).
    - Có điện hoặc không có điện.
- Số nhị phân = BIT (Binary digiT).
- Viết:  $1001_2$  hoặc  $1001_B$

## 4.4.1. Chuyển từ hệ 2 sang hệ 10

- $(a_n a_{n-1} \dots a_0)_B = a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \dots + a_0 \cdot 2^0$

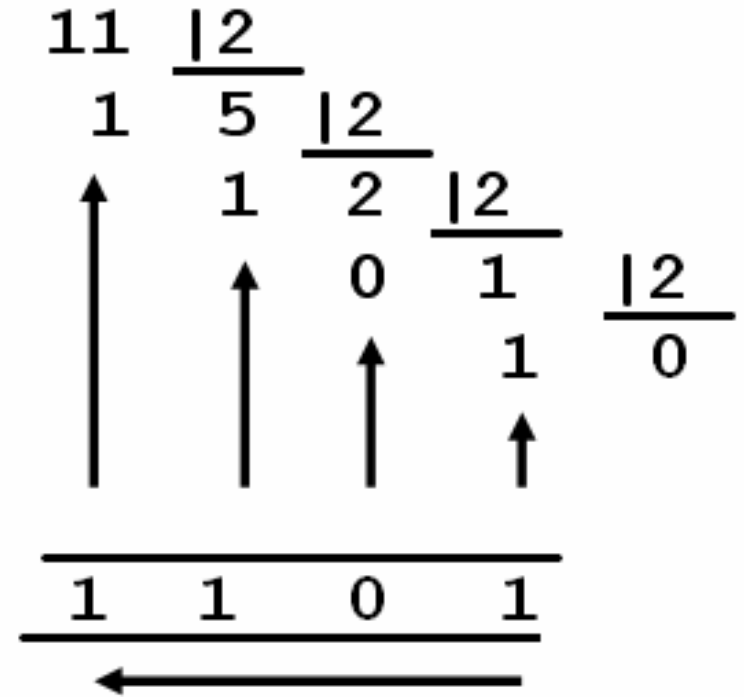
- Ví dụ:

- $0_B = 0; 10_B = 2$

- $1001_B = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 9$

## 4.4.2. Chuyển từ hệ 10 sang hệ 2

- $D$  = số cần chuyển
- Chia  $D$  (chia nguyên) liên tục cho 2 cho tới khi kết quả phép chia = 0
- Lấy phần dư các lần chia viết theo thứ tự ngược lại



## 4.4.3. Chuyển đổi số lẻ từ hệ 10 sang hệ 2

### ■ Phần nguyên

- Chia liên tiếp cho 2.
- Viết phần dư theo chiều ngược lại.

### ■ Phần phân

- $X$  = phần phân.
- Nhân  $X$  với 2  $\rightarrow$  kết quả:
  - Phần nguyên (0,1)
  - Phần phân
- Lặp lại từ bước đầu, đến khi muốn dừng hoặc kết quả=0.
- Viết các phần nguyên theo đúng thứ tự được kết quả.

$$12.73 = ?_B$$

	0.73
1	0.46
0	0.92
1	0.84
1	0.68



0.1011

$$12 = 1100_B$$

$$0.73 = 0.1011_B$$

$$\rightarrow 12.73 = 1100.1011_B$$

## 4.4.4. Các phép toán trên hệ 2

- Phép cộng
- Số âm (số bù hai)
- Phép trừ
- Phép nhân

# Cộng hai số nhị phân

- Cộng có nhớ các cặp số cùng vị trí từ phải sang trái
- Bảng cộng
- Ví dụ
  - $1010 + 1111 = 11001$

<u>A</u>	<u>B</u>	<u>Tổng</u>	<u>Nhớ</u>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\begin{array}{r} 1010 \\ 1111 \\ \hline 11001 \end{array}$$



# Số bù hai (số âm)

## ■ Số bù một

- Đảo tất cả các bit của một số nhị phân ta được số bù một của nó.

## ■ Lấy số bù một cộng 1 ta được số bù hai của số nhị phân ban đầu.

## ■ Ví dụ:

- $B = 1001$
- Bù một của B: 0110
- Bù hai của B: 0111

# Trừ hai số nhị phân $B1 - B2$

- $B2 + \text{bù hai của } B2 = 0$  (lấy số chữ số = số chữ số của  $B2$ ).
- Có thể coi bù hai của  $B2$  là số đối của  $B2$ .
- $B1 - B2 = B1 + \text{bù hai của } B2$ .

# 1010 - 0101

- Bù một của 0101: 1010
- Bù hai của 0101 =  $1010 + 1 = 1011$
- $1010 - 0101 = 1010 + 1011 = 0101$   
(chỉ lấy 4 bit kết quả !!!)

# Nhân hai số nhị phân

- Nhân từ trái phải qua trái theo cách thông thường
- Bảng nhân
- Ví dụ
  - $1011 \times 101 = 110111$

$$\begin{array}{r} 1011 \\ 101 \\ \hline 1011 \\ 0000 \\ 1011 \\ \hline 110111 \end{array}$$

<u>A</u>	<u>B</u>	<u>KQ</u>
0	0	0
0	1	0
1	0	0
1	1	1

# Chia hai số nhị phân

- Sau khi đã biết cách nhân, cộng, trừ các số nhị phân, hãy thử tưởng tượng ra cách chia số nhị phân giống như số hệ 10.
- Ví dụ:  $11101/101=101$ , dư 100.

$$\begin{array}{r} \overset{'''}{11101} \quad \overline{)101} \\ - \underline{101} \phantom{000} \\ 0100 \\ - \underline{000} \\ 1001 \\ - \underline{101} \\ \phantom{0}100 \end{array}$$

$$\rightarrow 11101/101 = 101 \text{ dư } 100$$

## 4.5. Hệ đếm cơ số 16 (Hexa)

- Sử dụng 16 ký hiệu:
  - 0..9
  - A,B,C,D,E,F
- Viết  $1A_{16}$  hoặc  $1AF_{16}$  hoặc 1AFH
- $A_{16} = 10$
- $F_{16} = 15$
- $10_{16} = 16$

## 4.5.1. Chuyển đổi hệ 16 và hệ 10

### ■ Từ hệ 10 $\rightarrow$ hệ 16

- Thực hiện chia liên tiếp cho 16
- Lấy phần dư viết ngược lại

### ■ Từ hệ 16 $\rightarrow$ hệ 10

- $(a_n a_{n-1} \dots a_0)_H = a_n \cdot 16^n + a_{n-1} \cdot 16^{n-1} + \dots + a_0 \cdot 16^0$

## 4.5.2. Chuyển đổi hệ 16 và hệ 2

- Một chữ số hệ 16 tương đương 4 BIT của hệ hai
  - $1_H = 0001_B$
  - $F_H = 1111_B$
- Xem bảng chuyển đổi các hệ

Hệ 10	Hệ 16	Hệ 2
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111



# Hệ 16 $\rightarrow$ hệ 2

- Căn cứ vào bảng chuyển đổi, thay thế 1 chữ số của số hệ 16 bằng 4 bit nhị phân.
- Ví dụ:

$$\square A_H = 1100_B$$

$$\square 7_H = 0111_B$$

$$\rightarrow A7_H = 1100\ 0111_B$$

$$A7_H = ?_B$$

Hexa	A				7			
Binary	1	1	0	0	0	1	1	1

# Hệ 2 $\rightarrow$ hệ 16

- Nhóm 4 bit một từ phải sang trái rồi căn cứ vào bảng chuyển đổi, thay thế bằng chữ số tương ứng trong hệ 16.

- Ví dụ:

$$\begin{aligned} & 1111100_B \\ = & 0111 \ 1100_B \\ = & 7A_H \end{aligned}$$

$$1111100_B = ?_H$$

0111 . 1100    Nhóm 4 bit  
←

Binary	0	1	1	1	1	1	0	0
Hexa	7				A			

# 5. Biểu diễn thông tin trong máy tính

- Cách biểu diễn
- Đơn vị thông tin
- Mã hoá

# 5.1. Cách biểu diễn

- Thông tin trong máy tính được biểu diễn dạng nhị phân
- Ví dụ:
  - 5 bit biểu diễn được 32 trạng thái.
  - 5 bit có thể dùng để biểu diễn 26 chữ cái A..Z.

1	1	0	0	0	0	0	1	1	0
1	0	0	0	0	0	1	1	1	1
0	1	0	1	1	1	0	0	0	0
0	0	1	1	0	1	1	0	0	1
1	1	1	0	1	1	0	1	0	0
0	1	0	1	1	0	0	0	0	0
1	1	1	0	0	1	0	1	0	0
1	1	1	0	1	1	0	1	0	0
0	1	1	0	1	0	1	1	0	1
1	1	0	1	0	1	1	0	0	1

## 5.2. Đơn vị thông tin

- BIT

- Chỉ nhận giá trị 0 hoặc 1

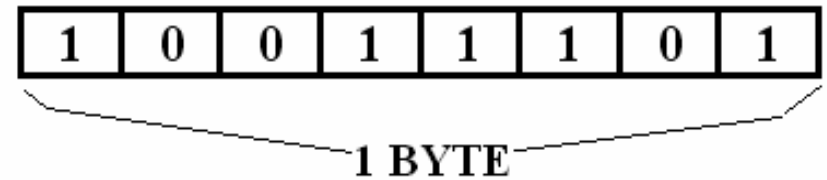
- 1Byte = 8 BIT

- 1KB =  $2^{10}$  Bytes  
= 1024 Bytes

- 1MB = 1024 KB

- 1GB = 1024 MB

- ...



## 5.3. Mã hoá

- Dù thông tin lưu trữ ở đâu cũng cần có quy luật để hiểu nó → **mã hoá**.
- Ví dụ
  - Mã SV: 20041021234
    - 2004: Vào trường năm 2004
    - 102: Mã ngành
    - 1234: Số hiệu sinh viên
  - Phòng: B209 (Nhà B - Tầng 2 - Phòng 09)
  - Biển số xe,...
- Mã hoá phải “rõ ràng” và “đầy đủ”

# Mã hoá trong máy tính

- Sử dụng số nhị phân
- Độ lớn của mã = số bit sử dụng để mã hoá
- Quy luật hiểu được mã nhị phân
- Ví dụ: Sử dụng 5 bit để mã hoá chữ cái hoa A..Z (26 chữ cái)
  - 00000 ← A
  - 00001 ← B
  - ...
  - 11001 ← Z
  - 11001 – 11111: chưa sử dụng

# ASCII

- ASCII (American Standard Code for Information Interchange)
- Dùng 8 bit để mã hoá các chữ cái.
- Mỗi chữ cái được gọi là một ký tự.
- Mã hoá được  $2^8 = 256$  ký tự.
  - $0 \rightarrow 31, 127$ : Các ký tự điều khiển
  - $32 \rightarrow 126$ : Các ký tự thông thường
  - $128 \rightarrow 255$ : Các ký tự đặc biệt



# Unicode

- Sử dụng nhiều hơn 8 bit (2,3,4,... Bytes) để mã hoá ký tự.
- 2 Bytes mã hoá được  $2^{16} = 65536$  ký tự.
- Hầu hết các chữ cái của các nước trên thế giới
  - Việt Nam
  - Trung Quốc
  - Nga,...

# 6. Đại số logic

- Mệnh đề logic
- Biến logic
- Hằng, biểu thức, hàm logic
- Các toán tử logic
- Mạch logic

# 6.1. Mệnh đề logic

- Khẳng định hay phủ định một sự kiện hay vấn đề
- Chỉ đúng hoặc sai
  - Đúng – TRUE (1)
  - Sai – FALSE (0)
- Ví dụ
  - “Con voi to nặng hơn con kiến bé” là mệnh đề đúng.
  - “Rửa bát đi!” không phải mệnh đề.
  - “Hổ là động vật ăn cỏ” là mệnh đề sai.

## 6.2. Biến logic

- Là biến đại diện cho đại lượng logic.
- Chỉ có thể nhận một trong hai giá trị:
  - Đúng (TRUE), hoặc
  - Sai (FALSE)
- VD 1:  $X = \text{“M là số âm.”}$ 
  - Khi M là số âm:  $X = \text{TRUE}$
  - Ngược lại,  $X = \text{FALSE}$
- VD2:  $Y = \text{“Hôm nay trời đẹp.”}$ 
  - Giá trị của Y thay đổi theo ngày.

## 6.3. Hằng, biểu thức và hàm logic

### ■ Hằng logic

- Có giá trị xác định
- Giá trị đó là TRUE hoặc FALSE
- Ví dụ: “ $2 > 3$ ” là một hằng logic nhận giá trị FALSE

### ■ Biểu thức, hàm logic

- Sự kết hợp của hằng, biến và toán tử
- Toán tử: và, hoặc, ...
- Ví dụ: “ $m \geq 3$ ” và “ $m \leq 5$ ”

## 6.4. Toán tử logic

- Là các phép toán với các mệnh đề, hằng, biến logic.
- Các toán tử cơ bản:
  - NOT
  - AND
  - OR
  - XOR

## 6.4.1. Toán tử “PHỦ ĐỊNH”

- Ký hiệu: **NOT**
  - NOT X
- Gọi tên
  - PHỦ ĐỊNH
- Ví dụ
  - NOT (“2>3”) = TRUE

X	NOT X
TRUE	FALSE
FALSE	TRUE

## 6.4.2. Toán tử “VÀ”

- Ký hiệu: **AND**
  - $X \text{ AND } Y$
- Gọi tên
  - VÀ
  - HỘI
- $X \text{ AND } Y$  chỉ đúng khi cả  $X$  và  $Y$  cùng đúng.
- Ví dụ
  - “ $2 > 3$ ” AND “ $3 = 4 - 1$ ” nhận giá trị FALSE

X	Y	X AND Y
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE



## 6.4.3. Toán tử “HOẶC”

- Ký hiệu: **OR**
  - $X \text{ OR } Y$
- Gọi tên
  - HOẶC
  - TUYỂN
- $X \text{ OR } Y$  chỉ sai khi cả  $X$  và  $Y$  cùng sai.
- Ví dụ
  - “ $2 > 3$ ” OR TRUE nhận giá trị TRUE

X	Y	$X \text{ OR } Y$
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE

## 6.4.4. Toán tử “HOẶC LOẠI TRỪ”

- Ký hiệu: **XOR**

- $X \text{ XOR } Y$

- **$X \text{ OR } Y$  sai khi  $X = Y$**

- Ví dụ

- “ $2 > 3$ ” XOR TRUE nhận giá trị TRUE
  - FALSE XOR “ $2 > 3$ ” nhận giá trị FALSE

X	Y	$X \text{ XOR } Y$
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	FALSE

# Thứ tự ưu tiên các phép toán

1. Dấu ngoặc ( ).
2. NOT, dấu trừ (-).
3. \*, /, DIV, MOD, AND.
4. +, -, OR, XOR.
5. =, <>, >, >=, <, <=.

# 6.5. Mạch logic

- Mạch điện tử
  - Tín hiệu điện
  - TRUE: hiệu điện thế  $\geq$  mức
- Đầu vào
  - Hằng, biến logic
- Đầu ra
  - Kết quả phép toán, hàm
- Các mạch cơ bản
  - NOT
  - AND
  - OR
  - XOR