

# Nhập môn điện toán

Đối tượng: Sinh viên năm thứ nhất khoa Khoa Học & Kỹ thuật Máy Tính, trường Đại Học Bách Khoa TpHCM.

Thời lượng học: 16 buổi LT + 2 buổi tiếp xúc doanh nghiệp,  
7 bài tập + 5 bài thực hành

Phân lý thuyết:

- C1. Khái niệm cơ bản.
- C2. Phần cứng máy tính.
- C3. Hệ điều hành
- C4. Mạng máy tính.
- C5. Cơ sở dữ liệu.
- C6. Lập trình với máy tính.
- C7. Các ứng dụng.

Tài liệu tham khảo :

- [1] Computing, 3<sup>rd</sup> ed., Geoffrey Knott & Nick Waites, 2000.
- [2] Tập Slide bài giảng & thực hành của môn học này.



# *Chương 1. KHÁI NIỆM CƠ BẢN*

---

1.1 Định nghĩa sơ khởi về máy tính số

1.2 Lịch sử phát triển máy tính số

1.3 Hệ thống số đếm

1.4 Biểu diễn dữ liệu

1.5 Luận lý máy tính

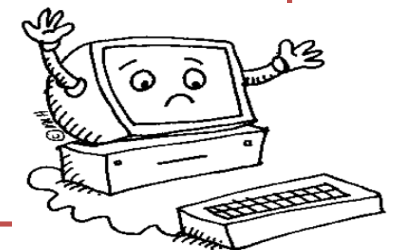
# 1.1. Định nghĩa sơ khởi về máy tính số

- ❖ Trong cuộc sống, con người đã biết chế tạo ngày càng nhiều *công cụ, thiết bị* để hỗ trợ mình trong hoạt động.
- ❖ Các công cụ, thiết bị chế tạo ngày càng tinh vi, phức tạp
  - ❖ mỗi công cụ, thiết bị thường chỉ thực hiện được 1 vài công việc cụ thể nào đó. (ví dụ, cây chổi để quét, radio để bắt và nghe đài audio...)
- ❖ **Máy tính số (digital computer)** là 1 thiết bị đặc biệt,
  - ❖ có thể thực hiện 1 số hữu hạn các chức năng cơ bản (*tập lệnh*).
  - ❖ cơ chế thực hiện các lệnh là *tự động* và *tuần tự*.
  - ❖ danh sách các lệnh được thực hiện này được gọi là *chương trình*.



# Định nghĩa sơ khởi về máy tính số (tt)

- ❖ Các lệnh mà máy hiểu và thực hiện được gọi là *lệnh máy*.
- ❖ *Ngôn ngữ lập trình* dùng để miêu tả các lệnh, gồm 2 yếu tố chính:
  - ❖ *cú pháp* qui định trật tự kết hợp các phần tử để cấu thành 1 lệnh (câu),
  - ❖ *ngữ nghĩa* cho biết ý nghĩa của lệnh đó.
- ❖ Để giải quyết một công việc (*bài toán*), ta có thể chia vấn đề thành trình tự nhiều công việc nhỏ được gọi là *giải thuật*.
- ❖ Mỗi công việc nhỏ hơn cũng có thể được chia nhỏ hơn nữa nếu nó còn phức tạp,...  $\Rightarrow$  giải pháp có thể được miêu tả bằng 1 trình tự các lệnh máy (*chương trình ngôn ngữ máy*).



# Định nghĩa sơ khởi về máy tính số (tt)



- ❖ Quá trình máy tính giải quyết công việc ngoài đời gọi là *lập trình* (qui trình xác định trình tự đúng các lệnh).
  - ❖ *Cho đến nay, lập trình là công việc của con người (với sự trợ giúp ngày càng nhiều của máy tính).*
- ❖ Máy tính hiện nay bao gồm tập lệnh máy rất sơ khai, mỗi lệnh máy chỉ có thể thực hiện 1 công việc rất nhỏ và đơn giản  
⇒ các giải pháp thực tế gồm một trình tự rất lớn các lệnh máy
  - ❖ *Lập trình bằng ngôn ngữ máy rất phức tạp, tốn nhiều thời gian, công sức, kết quả rất khó bảo trì, phát triển.*
- ❖ Nhu cầu về máy luận lý với tập lệnh (được đặc tả bởi ngôn ngữ lập trình) cao cấp và gần gũi hơn với con người. Ta thường hiện thực máy này bằng 1 *máy vật lý* + 1 *chương trình dịch*.



# Định nghĩa sơ khởi về máy tính số (tt)



❖ Có 2 loại *chương trình dịch* :

❖ *trình biên dịch (compiler)*

❖ *trình thông dịch (interpreter)*

❖ Gọi ngôn ngữ máy vật lý là  $N_0$ ; trình biên dịch ngôn ngữ  $N_1$  sang ngôn ngữ  $N_0$  sẽ nhận đầu vào là chương trình được viết bằng ngôn ngữ  $N_1$ , phân tích từng lệnh  $N_1$  rồi chuyển thành danh sách các lệnh ngôn ngữ  $N_0$  có chức năng tương đương.

❖ Để viết chương trình dịch từ ngôn ngữ  $N_1$  sang  $N_0$  dễ dàng, độ phức tạp của từng lệnh ngôn ngữ  $N_1$  không quá cao so với từng lệnh ngôn ngữ  $N_0$ .

➤ sau khi có máy luận lý hiểu được ngôn ngữ luận lý  $N_1$ ,

➤ có thể định nghĩa và hiện thực máy luận lý  $N_2$  theo cách trên

➤ ...và tiếp tục đến khi có 1 máy luận lý hiểu được ngôn ngữ  $N_m$  rất gần gũi với con người, dễ dàng miêu tả giải thuật của bài toán cần giải quyết...

➤ *Tuy nhiên, qui trình trên chưa có điểm dừng !*

# Định nghĩa sơ khởi về máy tính số (tt)

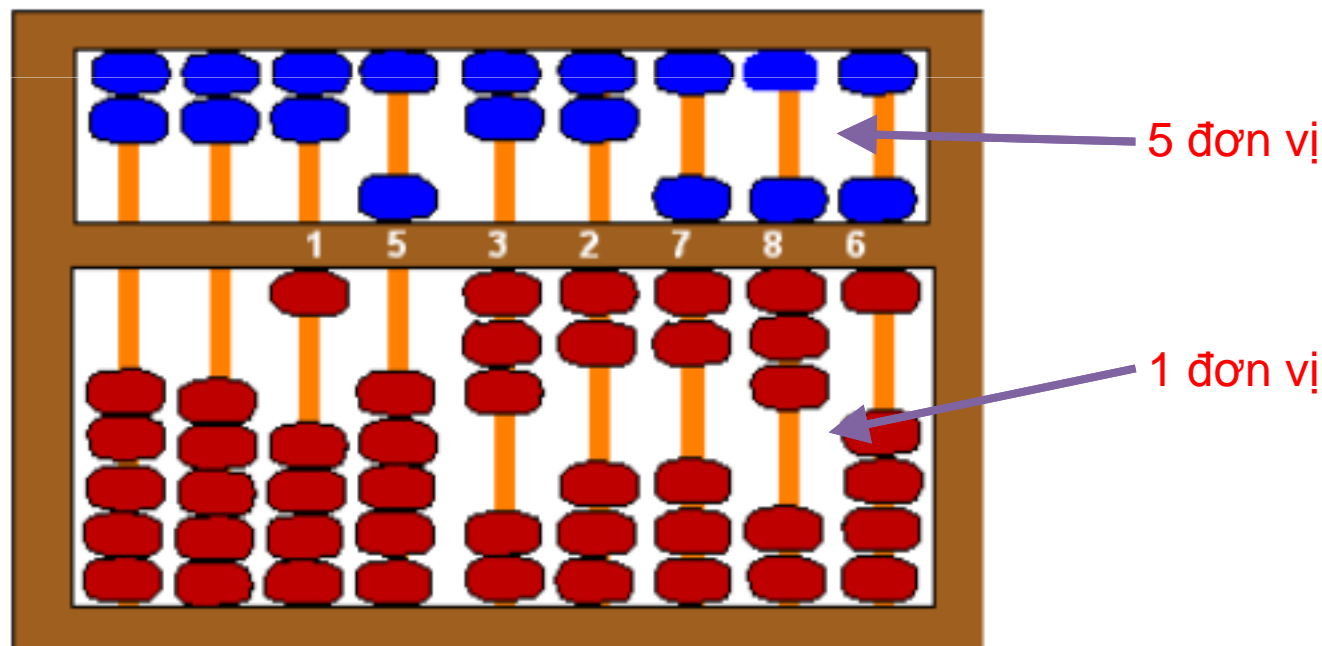
- ❑ **Ngôn ngữ máy vật lý** : ngôn ngữ cấp thấp nhất mà có thể lập trình
  - là loại ngôn ngữ mà máy vật lý có thể hiểu trực tiếp, nhưng con người thì gặp nhiều khó khăn trong việc viết và bảo trì chương trình ở cấp này.
- ❑ **Ngôn ngữ assembly** gần với ngôn ngữ máy + "lệnh macro" để nâng sức mạnh miêu tả giải thuật.
  - những lệnh cơ bản nhất tương ứng với lệnh máy, nhưng dưới dạng gọi nhớ.
- ❑ **Ngôn ngữ cấp cao** theo trường phái lập trình cấu trúc (Pascal, C,...)
  - tập lệnh của ngôn ngữ khá mạnh và gần với tư duy con người.
- ❑ **Ngôn ngữ hướng đối tượng** (C++, Visual Basic, Java, C#,,...)
  - Cải tiến phương pháp lập trình cấu trúc sao cho trong sáng, ổn định, dễ phát triển và thay thế linh kiện.



## 1.2. Vài dòng lịch sử . . .



- Máy tính xuất hiện từ rất lâu theo nhu cầu buôn bán và trao đổi tiền tệ.
- Bàn tính tay abacus là dạng sơ khai của máy tính.



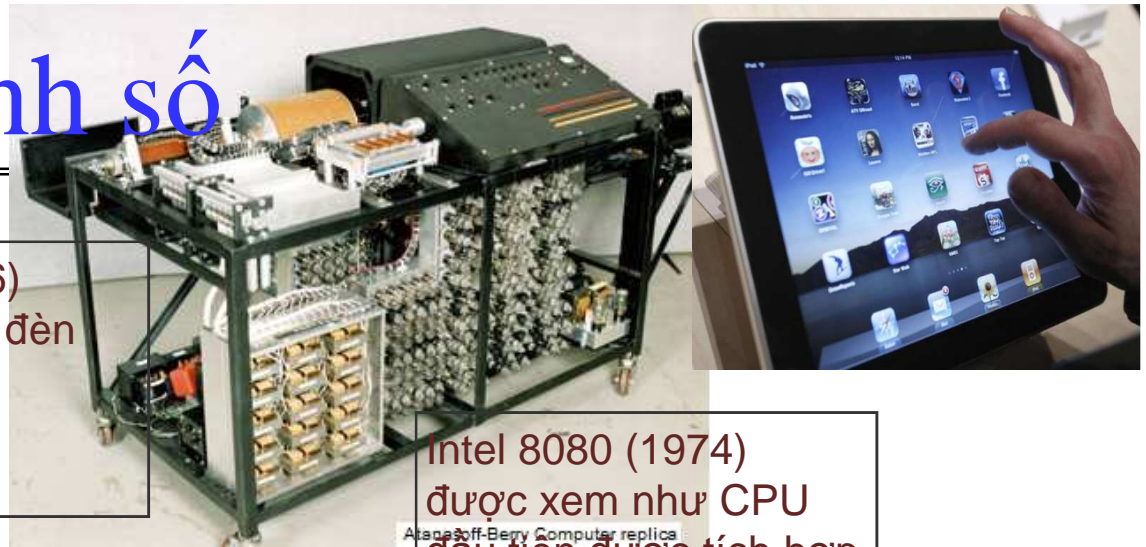


# Các thế hệ máy tính số

Blaise Pascal (Pháp-1642)

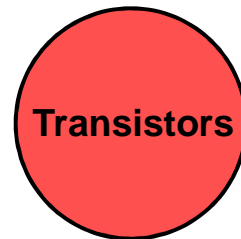
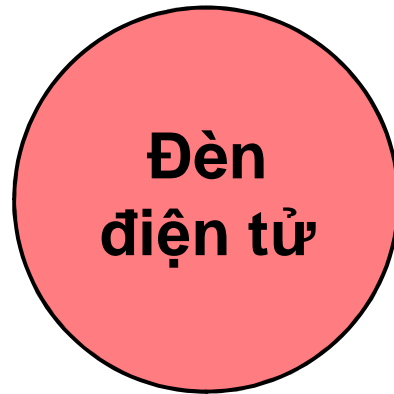
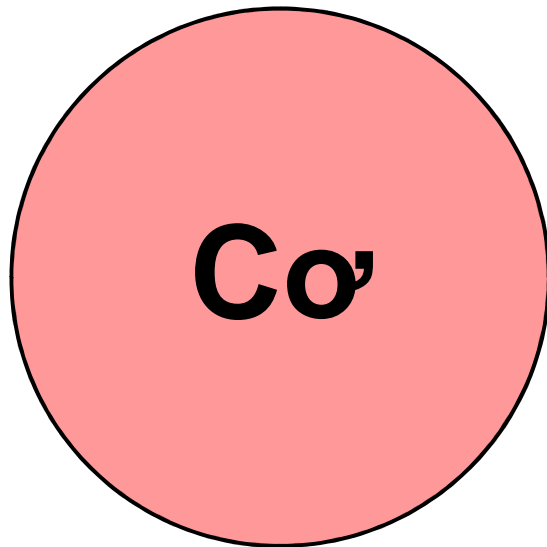
Charles Babbage (Anh-1830)

ENIAC (1946)  
18.000 bóng đèn  
1500 rờ le  
30 tấn  
140 KW



Intel 8080 (1974)  
được xem như CPU  
đầu tiên được tích hợp  
trên 1 chip

Von Neumann (1945)



PDP-1 (1961)

IBM 360 (1965)

80x86 (1978)

(1642 - 1945)

(1945 - 1955)

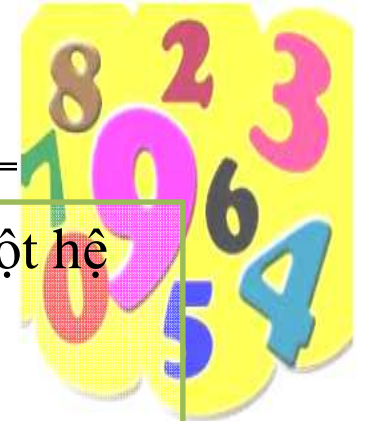
(1955 - 1965) (1965 - 1980) (1980 - ????)

Herman Hollerith lập IBM (International Business Machine) ở Mỹ - 1890

Bộ nhớ dây trể, tĩnh điện. Giấy, phiếu đục lỗ. Băng từ

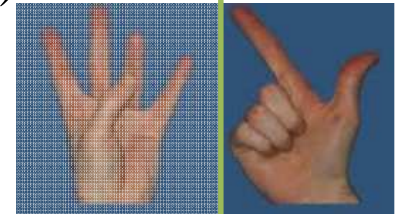
Bộ nhớ xuyên từ. Băng từ, trống từ, đĩa từ.

# 1.3. Hệ thống số đếm



❖ **Hệ thống số** (number system) là công cụ để biểu thị đại lượng. Một hệ thống số gồm 3 thành phần chính :

- 1) **cơ số** : số lượng ký số (ký hiệu để nhận dạng các số cơ bản).
- 2) **qui luật kết hợp** các ký số để miêu tả 1 đại lượng nào đó.
- 3) **các phép tính cơ bản** trên các số.



❖ chỉ có thành phần 1 là khác nhau giữa các hệ thống số, còn 2 thành phần 2 và 3 thì giống nhau giữa các hệ thống số.

## Ví dụ

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

# Hệ thống số đếm – Cơ số

- Trước khi có máy tính, con người dùng hệ số đếm thập phân (10).

## Thập phân (decimal)

Ký số

0	1	2	3	4
5	6	7	8	9

### Quy tắc đếm

0 → 1 → 2 → ... → 9 →

10 → 11 → 12 → ... → 19 →

20 → 21 → 22 → ... → 29 → ... → 90 → 91 → 92 → ... → 99 →

100 → 101 → ... → 109 → ... → 990 → 991 → ... → 999 →

1000 → 1001 → 1002 → ... → 1009 →

→ ...

# Hệ thống số đếm – Cơ số (tt.1)

- Sau khi máy tính số ra đời, các hệ số mới hình thành.

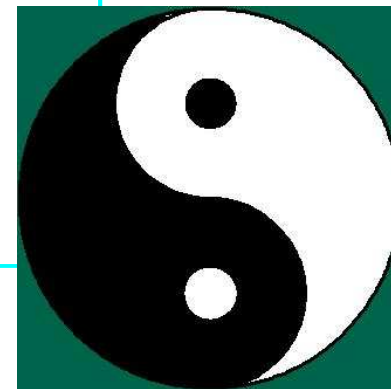
## Hệ nhị phân (Binary)

Ký số

0 1

### Quy tắc đếm

0 → 1 →  
10 → 11 →  
100 → 101 → 110 → 111 →  
1000 → 1001 → ... → 1110 → 1111 →  
10000 → 10001 →  
→ ...



# Hệ thống số đếm – Cơ số (tt.2)

- Số ở hệ nhị phân dài, khó nhớ chỉ dùng cho máy.
- Con người dùng số hệ bát phân (8) và thập lục phân (16) thay cho hệ nhị phân.

## Hệ bát phân (Octal)

Ký số	0	1	2	3
	4	5	6	7

### Quy tắc đếm

$0 \rightarrow 1 \rightarrow 2 \rightarrow \dots \rightarrow 7 \rightarrow$   
 $10 \rightarrow 11 \rightarrow 12 \rightarrow \dots \rightarrow 17 \rightarrow$   
 $20 \rightarrow 21 \rightarrow 22 \rightarrow \dots \rightarrow 77 \rightarrow$   
 $100 \rightarrow 101 \rightarrow 102 \rightarrow \dots \rightarrow 107 \rightarrow \dots \rightarrow 777 \rightarrow$   
 $1000 \rightarrow 1001 \rightarrow 1002 \rightarrow \dots \rightarrow 1007 \rightarrow$   
 $\rightarrow \dots$



# Hệ thống số đếm – Cơ số (tt.3)

- Một ký số hệ 8 bằng 3 ký số hệ 2.
- Một ký số hệ 16 bằng 4 ký số hệ 2.

## Hệ thập lục phân (hexadecimal)

Ký số

0	1	2	3	4	5	6	7
8	9	A	B	C	D	E	F

Quy tắc đếm

0 → 1 → 2 → ... → 9 → A → B → ... → F →  
10 → 11 → 12 → ... → 19 → 1A → ... → 1F → 20 → ... → 9F →  
A0 → A1 → A2 → ... → AF → ... → F0 → F1 → F2 → ... → FF →  
100 → 101 → 102 → ... → 10F → ... → FFF →  
1000 → 1001 → 1002 → ... → 100F →  
→ ...

# Công thức tính trị số

- Nếu  $B$  là cơ số,  $v_i$  là ký số ở hàng  $i$  (0 là hàng đơn vị, 1 là hàng "chục", 2 là hàng "trăm", ...) thì giá trị  $Q$  của số tính trong hệ 10 theo công thức sau :

$v_n v_{n-1} \dots v_0 \cdot v_{-1} \dots v_{-m}$

Chấm B phân

$$Q = v_n \times B^n + v_{n-1} \times B^{n-1} + \dots + v_0 \times B^0 + v_{-1} \times B^{-1} + \dots + v_{-m} \times B^{-m}$$

hay

$$\sum_{i=-m}^n v_i \times B^i$$

# Ví dụ số nguyên

$$\begin{array}{c} 1011_B \\ \swarrow \quad \downarrow \quad \searrow \quad \swarrow \\ 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 0 + 2 + 1 = 11_D \end{array}$$
$$\begin{array}{c} 173_O \\ \swarrow \quad \downarrow \quad \searrow \\ 1 \times 8^2 + 7 \times 8^1 + 3 \times 8^0 = 64 + 56 + 3 = 123_D \end{array}$$

$$\begin{array}{c} A4B5_H \\ \swarrow \quad \downarrow \quad \searrow \quad \swarrow \\ A \times 16^3 + 4 \times 16^2 + B \times 16^1 + 5 \times 16^0 \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ 10 \times 4096 + 4 \times 256 + 11 \times 16 + 5 \times 1 = 40960 + 1024 + 176 + 5 \\ = 42165_D \end{array}$$

# Ví dụ số lẻ

---

**1011.01**<sub>2</sub>

$$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

$$1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 + 0 \times 0.5 + 1 \times 0.25 = 11.25_{10}$$

**10.4**<sub>8</sub>

$$1 \times 8^1 + 0 \times 8^0 + 4 \times 8^{-1}$$

$$1 \times 8 + 0 \times 1 + 4 \times 0.125 = 8.5_{10}$$

# Bảng chuyển miêu tả số

Số hệ 10	Số hệ 16	Số 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



(0+4+2+0)

(8+0+2+1)

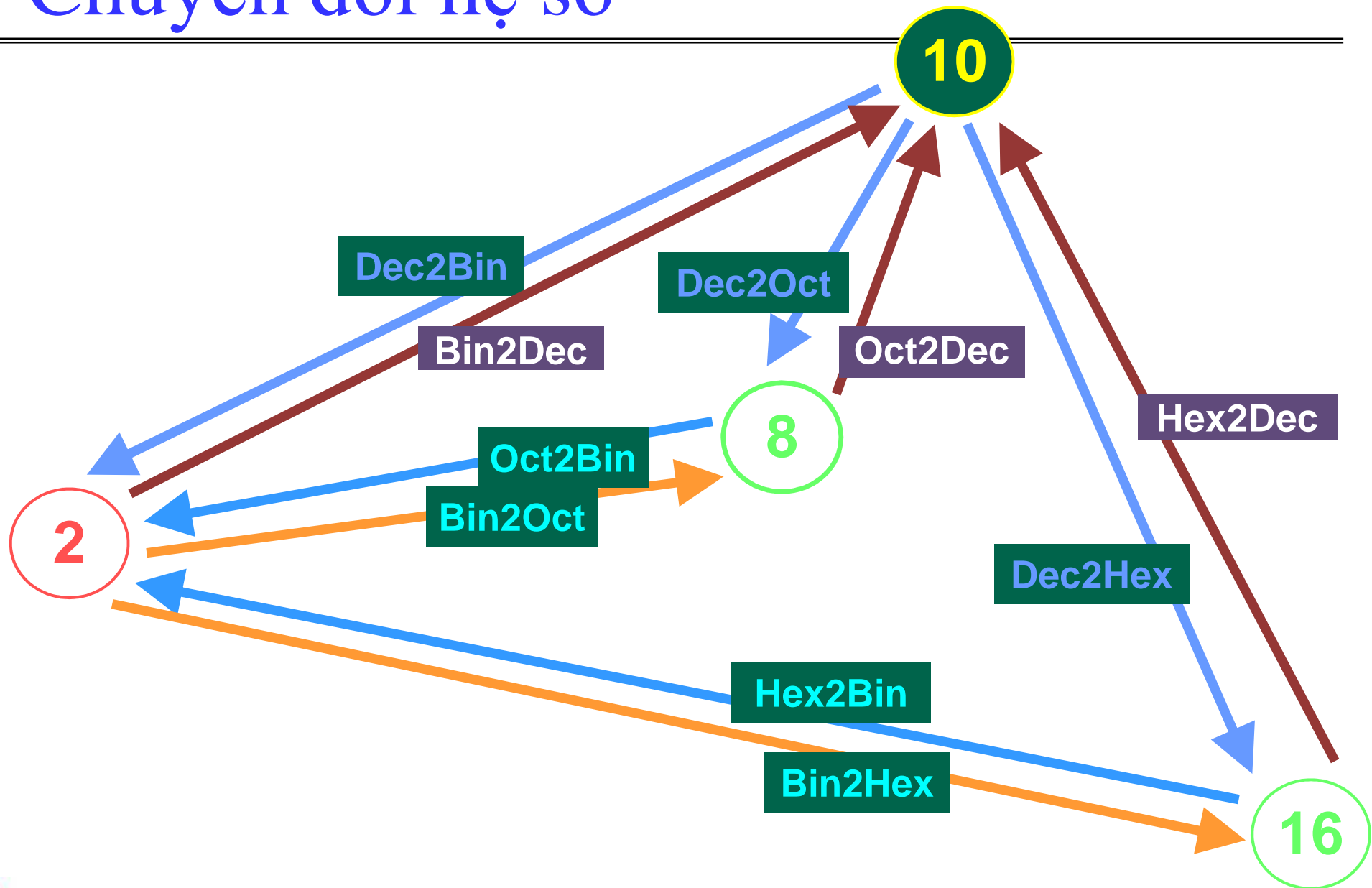
(8+4+0+1)



# Các phương pháp chuyển miêu tả số

- Để chuyển 1 miêu tả số từ hệ thống số này sang hệ thống số khác, ta cần dùng 1 phương pháp chuyển thích hợp. Có 4 phương pháp sau tương ứng với từng yêu cầu chuyển tương ứng :
- chuyển từ hệ thống số khác về thập phân.
  - chuyển từ hệ thống số thập phân về hệ thống số khác.
  - chuyển từ nhị phân về thập lục phân (hay bát phân).
  - chuyển từ thập lục phân (hay bát phân) về nhị phân.

# Chuyển đổi hệ số



# Từ hệ thống số khác về thập phân

## Xxx2Dec

- Để chuyển 1 miêu tả số từ hệ thống số khác (nhị phân, thập lục phân hay bát phân) sang hệ thập phân, ta dùng công thức tính Q.

### Ví dụ

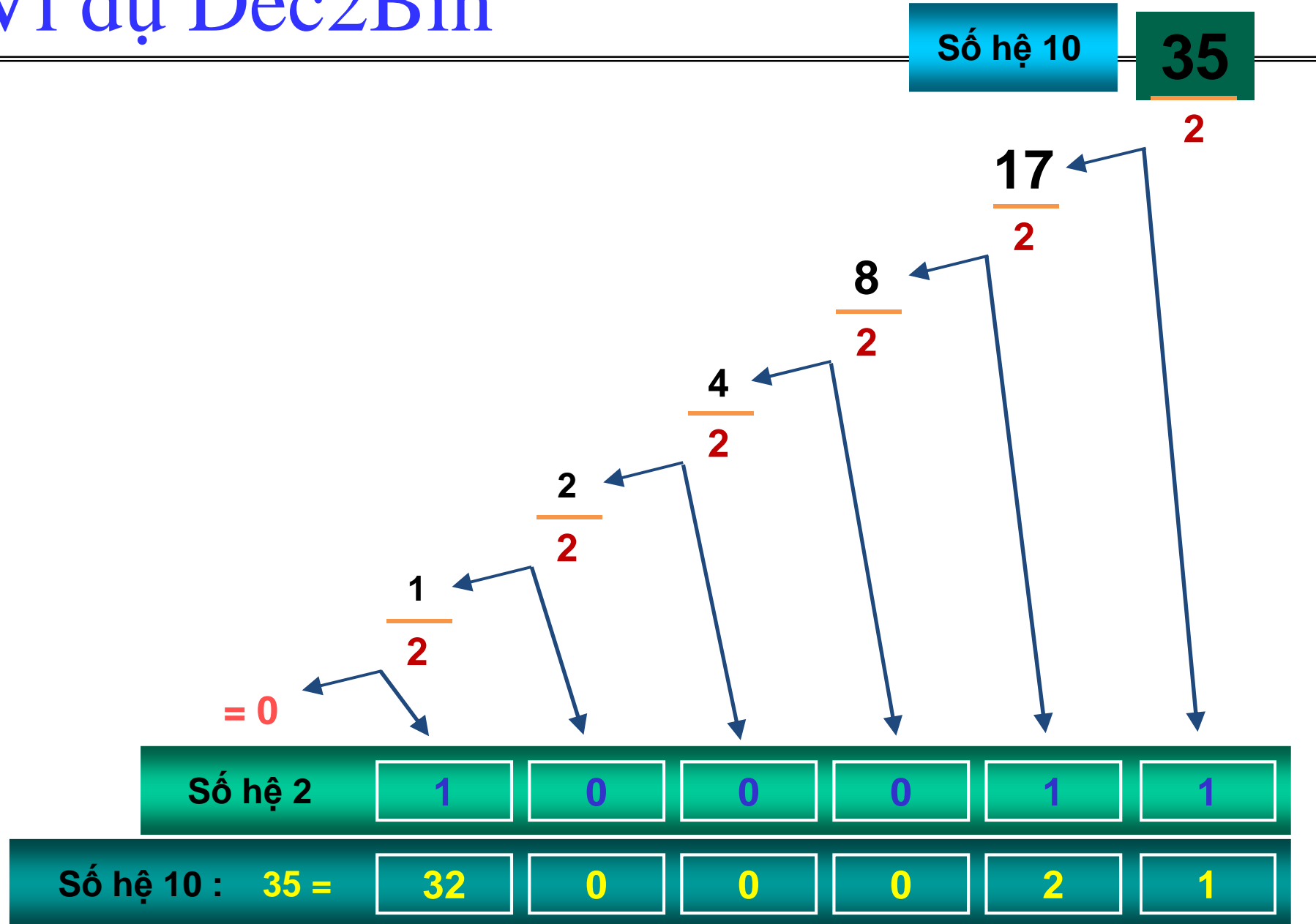
$$1A2_H = 1 \cdot 16^2 + XXXX10 \cdot 16^1 + 2 \cdot 16^0 = 256 + 160 + 2 = 418_D$$

$$642_O = 6 \cdot 8^2 + 4 \cdot 8^1 + 2 \cdot 8^0 = 384 + 32 + 2 = 418_D$$

$$110100010_B = 2^8 + 2^7 + 2^5 + 2^1 = 256 + 128 + 32 + 2 = 418_D$$

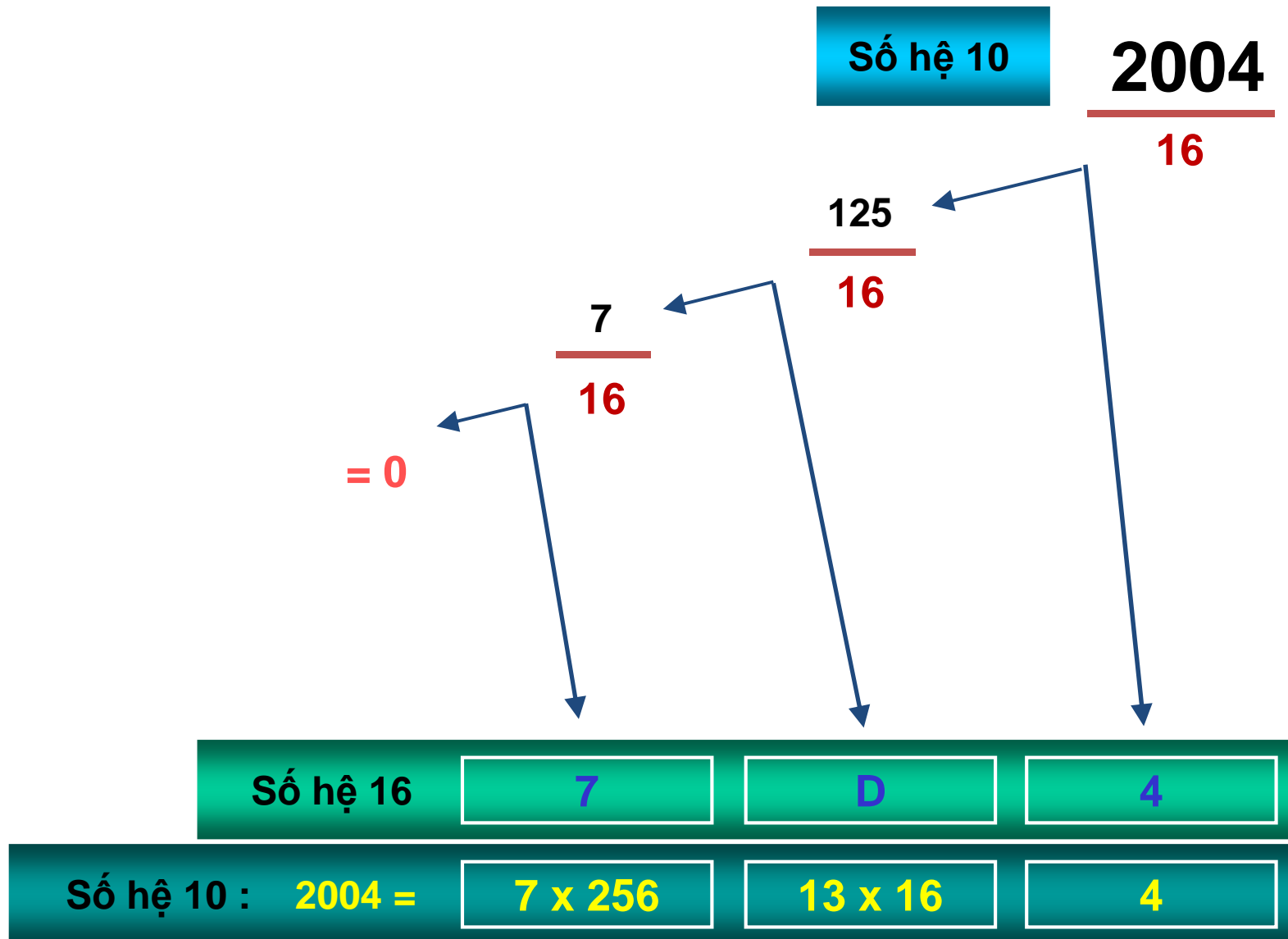


# Ví dụ Dec2Bin





# Ví dụ Dec2Hex



# Đổi hệ 2 ra hệ 8, 16

**Bin2Oct**

Ghép nhóm + bảng chuyển  
miêu tả số

Số hệ 2    1 0 1 0 0 0 1 1

3 bit

3 bit

3 bit

Mã 8421 → (010)

(100)

(011)

Số hệ 8    2    4    3

**Bin2Hex**

Số hệ 2    1 0 1 0 0 0 1 1

4 bit

4 bit

Mã 8421 → (1010)

(0011)

Số hệ 16    A    3

# Bảng chuyển miêu tả số

Số hệ 10	Số hệ 16	Số hệ 8	Số hệ 2
0	0	0	0000
1	1	1	0001
2	2	2	0010
3	3	3	0011
4	4	4	0100
5	5	5	0101
6	6	6	0110
7	7	7	0111
8	8	10	1000
9	9	11	1001
10	A	12	1010
11	B	13	1011
12	C	14	1100
13	D	15	1101
14	E	16	1110
15	F	17	1111

# Chuyển đổi số

---

Ví dụ

$$1A2_{\text{H}} = 1 * 16^2 + 10 * 16^1 + 2 * 16^0 = 256 + 160 + 2 = 418_{\text{D}}$$

$$642_{\text{O}} = 6 * 8^2 + 4 * 8^1 + 2 * 8^0 = 384 + 32 + 2 = 418_{\text{D}}$$

$$110100010_{\text{B}} = 2^8 + 2^7 + 2^5 + 2^1 = 256 + 128 + 32 + 2 = 418_{\text{D}}$$

# Hệ thống số đếm và các phép tính

- Các phép tính cơ bản trong 1 hệ thống số là :
  1. phép cộng (+).
  2. phép trừ (-).
  3. phép chia (/).
  4. phép nhân (\*).
  5. phép dịch trái n ký số ( $\ll n$ ).
  6. phép dịch phải n ký số ( $\gg n$ ).
- Ngoài ra do đặc điểm của hệ nhị phân, hệ này còn cung cấp 1 số phép tính sau (các phép tính luận lý) :
  1. phép OR bit ( $\mid$ ).
  2. phép AND bit ( $\&$ ).
  3. phép XOR bit ( $\wedge$ ).
  4. ....

# Hệ thống số đếm và các phép tính

Ví dụ

$$\begin{array}{r} 0110 \\ + 0011 \\ \hline 1001 \end{array}$$

$$\begin{array}{r} 1001 \\ - 0011 \\ \hline 0110 \end{array}$$

$$\begin{array}{r} 1011 \\ - 10 \\ \hline 01 \\ - 00 \\ \hline 11 \\ - 10 \\ \hline 01 \end{array} \quad \begin{array}{r} 10 \\ \hline 101 \end{array}$$

$$\begin{array}{r} \phantom{x} 1001 \\ x 0101 \\ \hline 1001 \\ 0000 \\ 1001 \\ 0000 \\ \hline 0101101 \end{array}$$

# Các phép tính của đại số Boole

Biểu thức Boole là 1 biểu thức toán học cấu thành từ các phép toán Boole trên các toán hạng là các biến chỉ chứa 2 trị 0 và 1.

X	Y	NOT X	X AND Y	X NAND Y	X OR Y	X NOR Y	X XOR Y
0	0	1	0	1	0	1	0
0	1	1	0	1	1	0	1
1	0	0	0	1	1	0	1
1	1	0	1	0	1	0	0

# Biểu diễn thông tin bằng hệ nhị phân

BIT ( **B**inary digi**T** ) : 0 1  
BYTE = tổ hợp 8 bit : 01001101 11111111  
(BYTE được chọn làm đơn vị tổ chức thông tin trong máy tính)  
WORD = tổ hợp 2 byte : 10110 1011100101  
DWORD = tổ hợp 4 byte

1 KiloByte (KB) = 1024 byte =  $2^{10}$  byte  
1 MegaByte (MB) = 1024 KB =  $2^{20}$  byte  
1 GigaByte (GB) = 1024 MB =  $2^{30}$  byte  
1 TetraByte (TB) = 1024 GB =  $2^{40}$  byte

1 số dài n bit thì biểu diễn được  $2^n$  giá trị

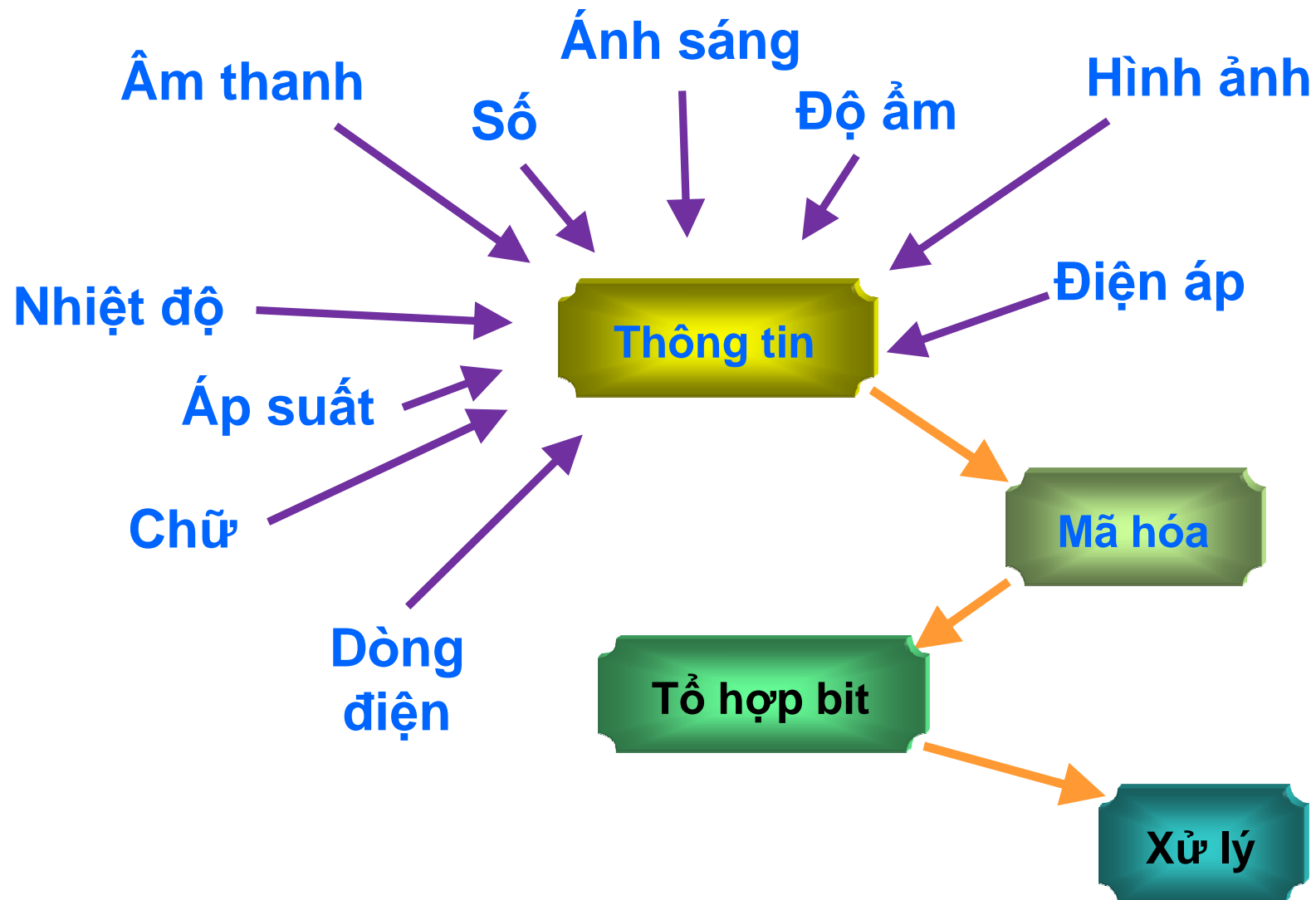


# 1.4. Biểu diễn dữ liệu

---

- ◎ Máy tính làm việc trên số nhị phân.
- ◎ Con người không thể làm việc với số nhị phân vì dài, khó nhớ.
- ◎ Dữ liệu cần biểu diễn, xử lý, lưu trữ bằng máy tính gồm có đại lượng số và phi số.
- ◎ Dữ liệu đưa vào máy tính phải được mã hóa thành số nhị phân (code) rồi mới xử lý.

# Mã hóa thông tin đầu vào



# Biểu diễn số

## Số không dấu

Số n bit có giá trị :  $0 \div (2^n - 1)$

Số 8 bit có giá trị :  $0 \div 255$

Số 16 bit có giá trị :  $0 \div 65\,535$

Số 32 bit có giá trị :  $0 \div 4\,294\,967\,295$

## Số có dấu

Quy ước: chọn bit có trọng số cao nhất (MSB) làm bit dấu

MSB  
(Most Significant Bit)



(Least Significant Bit)

Số 8 bit có dấu có giá trị :  $-128 \div +127$

Số 16 bit có dấu có giá trị :  $-32768 \div +32767$

# Biểu diễn số nguyên có dấu

- ❑ **Phần dương** có 32768 số từ số 0 tới 32767, được miêu tả theo công thức  $Q$ .
- ❑ **Phần âm** có 32768 số từ -32768 tới -1, được miêu tả ở dạng số bù 2 như sau :
  - ✓ **Số bù 1** của 1 số  $n$  bit là  $n$  bit mà mỗi bit là ngược với bit gốc ( $0 \rightarrow 1$  và  $1 \rightarrow 0$ )
  - ✓ **Số bù 2** của 1 số  $n$  bit là số bù 1 của số đó rồi tăng lên 1 đơn vị.

Sự biểu diễn	Giá trị
0000 0000 0000 0000	0
0000 0000 0000 0001	1
...	
0111 1111 1111 1111	32767
1000 0000 0000 0000	-32768
1000 0000 0000 0001	-32767
...	
1111 1111 1111 1111	-1

# Biểu diễn số nguyên có dấu (tt)

- ❑ Vì mỗi ô nhớ máy tính chỉ chứa được 1 byte, do đó ta phải dùng nhiều ô liên tiếp (2 hay 4) để chứa số nguyên. Có 2 cách chứa các byte của số nguyên (hay dữ liệu khác) vào các ô nhớ : **BE** & **LE**.
- Cách **BE (Big Endian)** chứa byte trọng số cao nhất vào ô nhớ địa chỉ thấp trước, sau đó lần lượt đến các byte còn lại.
- Cách **LE (Little Endian)** chứa byte trong số nhỏ nhất vào ô nhớ địa chỉ thấp trước, sau đó lần lượt đến các byte còn lại.
- ✓ CPU Intel & HĐH Windows sử dụng cách **LE** để chứa số nguyên vào bộ nhớ (Integer và Long).

# Biểu diễn số nguyên có dấu (tt)

❖ Số 15 được miêu tả dưới dạng nhị phân 16 bit như sau :

0000 0000 0000 1111

- Nếu dùng 2 byte để lưu trữ, có thể dùng dạng 16 bit viết ngắn gọn là  $000F_H$ . Nếu lưu vào bộ nhớ dưới dạng LE thì ô nhớ có địa chỉ thấp (i) chứa byte  $0F_H$ , và ô nhớ kế (i+1) chứa byte  $00_H$ .
- Trường hợp dùng 4 byte:  $0000000F_H$  và lưu vào bộ nhớ dạng LE tốn 4 ô nhớ với giá trị lần lượt từ địa chỉ thấp đến cao là  $0F_H$ ,  $00_H$ ,  $00_H$ ,  $00_H$ .
- ✓ Số bù 1 của 15 là 1111 1111 1111 0000,
- ✓ Số bù 2 của 15 là 1111 1111 1111 0001
- Như vậy -15 được lưu vào máy dạng Integer là 2 byte có giá trị  $FFF1_H$ . Nếu lưu vào ô nhớ dạng LE thì ô nhớ có địa chỉ thấp (i) chứa byte  $F1_H$ , và ô nhớ kế (i+1) chứa byte  $FF_H$ .

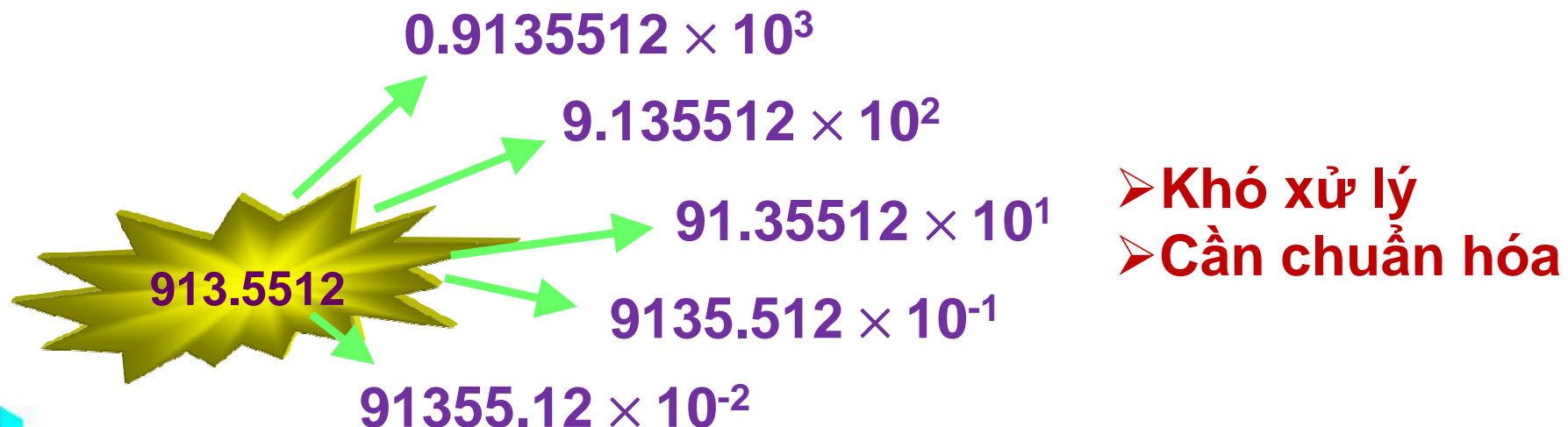
# Số thực - số chấm động

Số chấm động (floating point) dùng để tính toán trên số thực.

$$\pm m \times B^{\pm e}$$

m (mantissa) quyết định độ chính xác  
B (base)  
e (exponent) quyết định độ lớn/nhỏ

Một giá trị có thể biểu diễn dưới nhiều dạng



# Số chấm động theo chuẩn IEEE 754

## Chuẩn IEEE 754 qui định

Có 2 dạng  $\longrightarrow$  Chính xác đơn

$$(-1)^S 2^{E-127} (1.M)$$

Mũ thừa 127  
Cơ số 2  
 $1 \leq (1.M) < 2$

Significand

Chính  
xác  
kép

$$(-1)^S 2^{E-1023} (1.M)$$

Mũ thừa 1023  
Cơ số 2  
 $1 \leq (1.M) < 2$

dài 32 bit



dài 64 bit





# Ví dụ số chấm động độ chính xác đơn

Số  $N = -1.5$   $\begin{cases} \rightarrow S = 1 \\ \rightarrow E = 127 \\ \rightarrow M = 1.1 \end{cases}$

$$(-1)^1 2^{127-127} (1.1)$$

$N = 1011 \ 1111 \ 1100 \ 0000 \ 0000 \ 0000 \ 0000 \ 0000$   
          B      F      C      0      0      0      0      0

Giới hạn

$$(-1)^0 2^{-126} (1.0) \div 2^{+127} (2-2^{-23})$$

nghĩa là  $1.18 \times 10^{-38} \div 3.40 \times 10^{+38}$

# Các bảng mã dùng cho ký tự

---

**Mã BCD (Binary Coded Decimal)**

6 bit = 2 bit vùng + 4 bit số = 64 ký tự

**Mã EBCDIC (Extended Binary Coded Decimal Interchange Code)**

8 bit = 256 ký tự

**Mã ASCII American Standard Code for Information Interchange**

7 bit = 128 ký tự

Máy tính ngày nay

**Mã extended ASCII, ISO8859-1**

8 bit = 256 ký tự    lưu trữ 1 byte/ký tự

**Mã Unicode**

16 bit = 65536 ký tự    lưu trữ 2 byte/ký tự

**Yếu tố  
chữ viết  
các dân  
tộc**

(Tiếng Việt trên máy tính chưa thống nhất)



# Số BCD (*Binary Coded Decimal*)

- được dùng để tính toán trên số thập phân trong hệ nhị phân.
- được viết theo hệ 16 nhưng giá trị tính theo hệ 10.
- Số  $12_{\text{BCD}}$  được viết trong hệ 2 là 0001 0010 nhưng có giá trị là 12 thay vì 18 ( $12_{16} = 18_{10}$ ).

## Chỉnh dạng số BCD sau khi cộng

kết quả **đúng**  
không chỉnh

$$\begin{array}{r} + 18 \\ + 01 \\ \hline 19 \end{array}$$

Chỉnh dạng vì A không phải là số BCD

$$\begin{array}{r} + 18 \\ + 02 \\ \hline 1A \\ + 6 \\ \hline 20 \end{array}$$

kết quả **không đúng**  
chỉnh dạng

$$\begin{array}{r} + 18 \\ + 09 \\ \hline 21 \\ + 6 \\ \hline 27 \end{array}$$

Chỉnh dạng vì có nhớ ở hàng đơn vị

# Mã ASCII

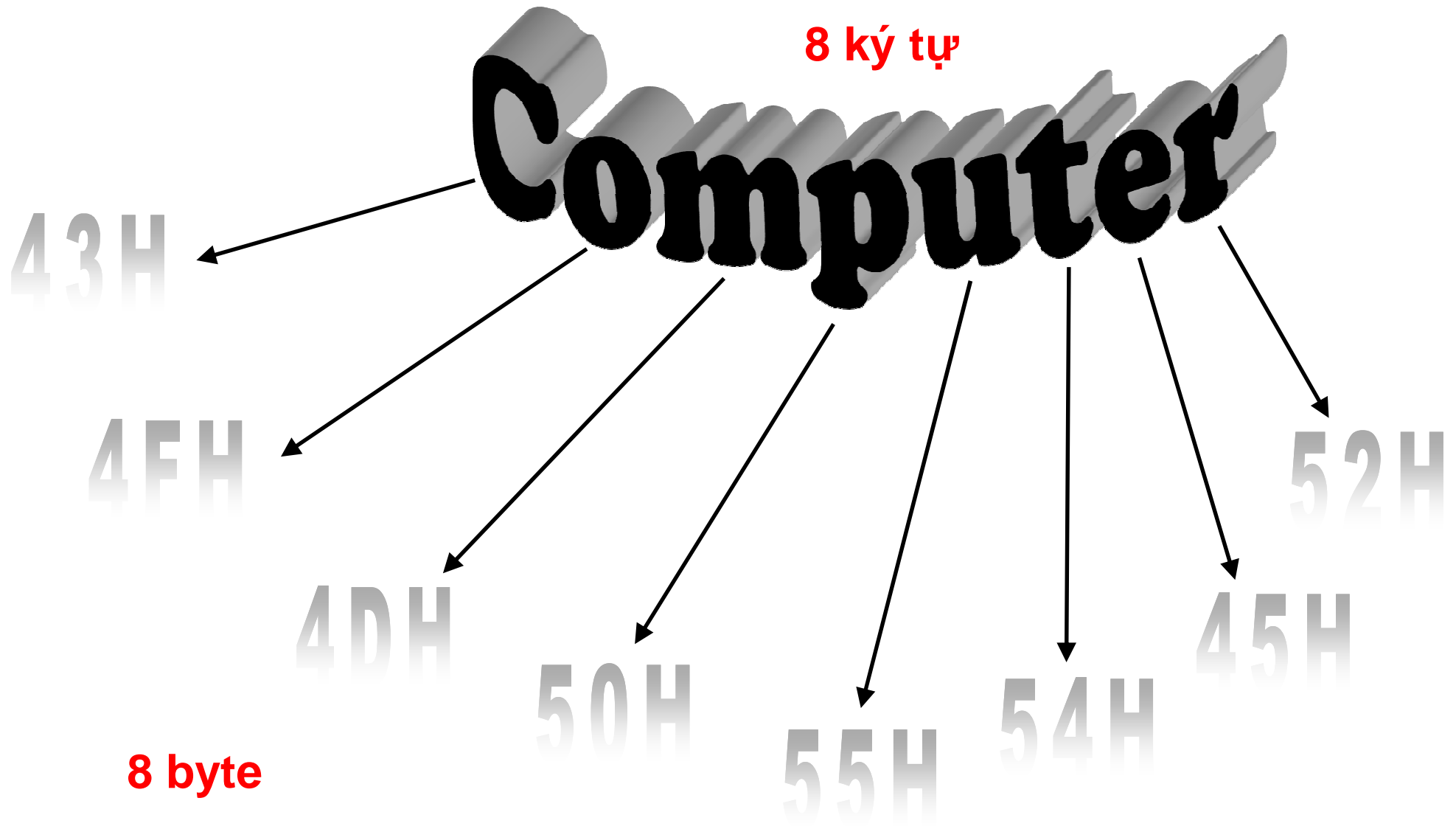
\$00	NUL	\$20	SP	\$40	@	\$60	`
\$01	SOH	\$21	!	\$41	A	\$61	a
\$02	STX	\$22	"	\$42	B	\$62	b
\$03	ETX	\$23	#	\$43	C	\$63	c
\$04	EOT	\$24	\$	\$44	D	\$64	d
\$05	ENQ	\$25	%	\$45	E	\$65	e
\$06	ACK	\$26	&	\$46	F	\$66	f
\$07	BEL	\$27	'	\$47	G	\$67	g
\$08	BS	\$28	(	\$48	H	\$68	h
\$09	HT	\$29	)	\$49	I	\$69	i
\$0A	LF	\$2A	*	\$4A	J	\$6A	j
\$0B	VT	\$2B	+	\$4B	K	\$6B	k
\$0C	FF	\$2C	,	\$4C	L	\$6C	l
\$0D	CR	\$2D	-	\$4D	M	\$6D	m
\$0E	SO	\$2E	.	\$4E	N	\$6E	n
\$0F	SI	\$2F	/	\$4F	O	\$6F	o
\$10	DLE	\$30	0	\$50	P	\$70	p
\$11	DC1	\$31	1	\$51	Q	\$71	q
\$12	DC2	\$32	2	\$52	R	\$72	r
\$13	DC3	\$33	3	\$53	S	\$73	s
\$14	DC4	\$34	4	\$54	T	\$74	t
\$15	NAK	\$35	5	\$55	U	\$75	u
\$16	SYN	\$36	6	\$56	V	\$76	v
\$17	ETB	\$37	7	\$57	W	\$77	w
\$18	CAN	\$38	8	\$58	X	\$78	x
\$19	EM	\$39	9	\$59	Y	\$79	y
\$1A	SUB	\$3A	:	\$5A	Z	\$7A	z
\$1B	ESC	\$3B	;	\$5B	[	\$7B	{
\$1C	FS	\$3C	<	\$5C	\	\$7C	
\$1D	GS	\$3D	=	\$5D	]	\$7D	}
\$1E	RS	\$3E	>	\$5E	^	\$7E	~
\$1F	US	\$3F	?	\$5F	_	\$7F	DEL

# Mã ISO8859-1

ISO/CEI 8859-1																
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	<i>positions inutilisées</i>															
1x																
2x	SP	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8x	<i>positions inutilisées</i>															
9x																
Ax	NBSP	ı	¢	£	¤	¥	¦	§	¨	©	*	«	¬	¯	®	—
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ø	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

# Lưu trữ bằng mã ASCII

---

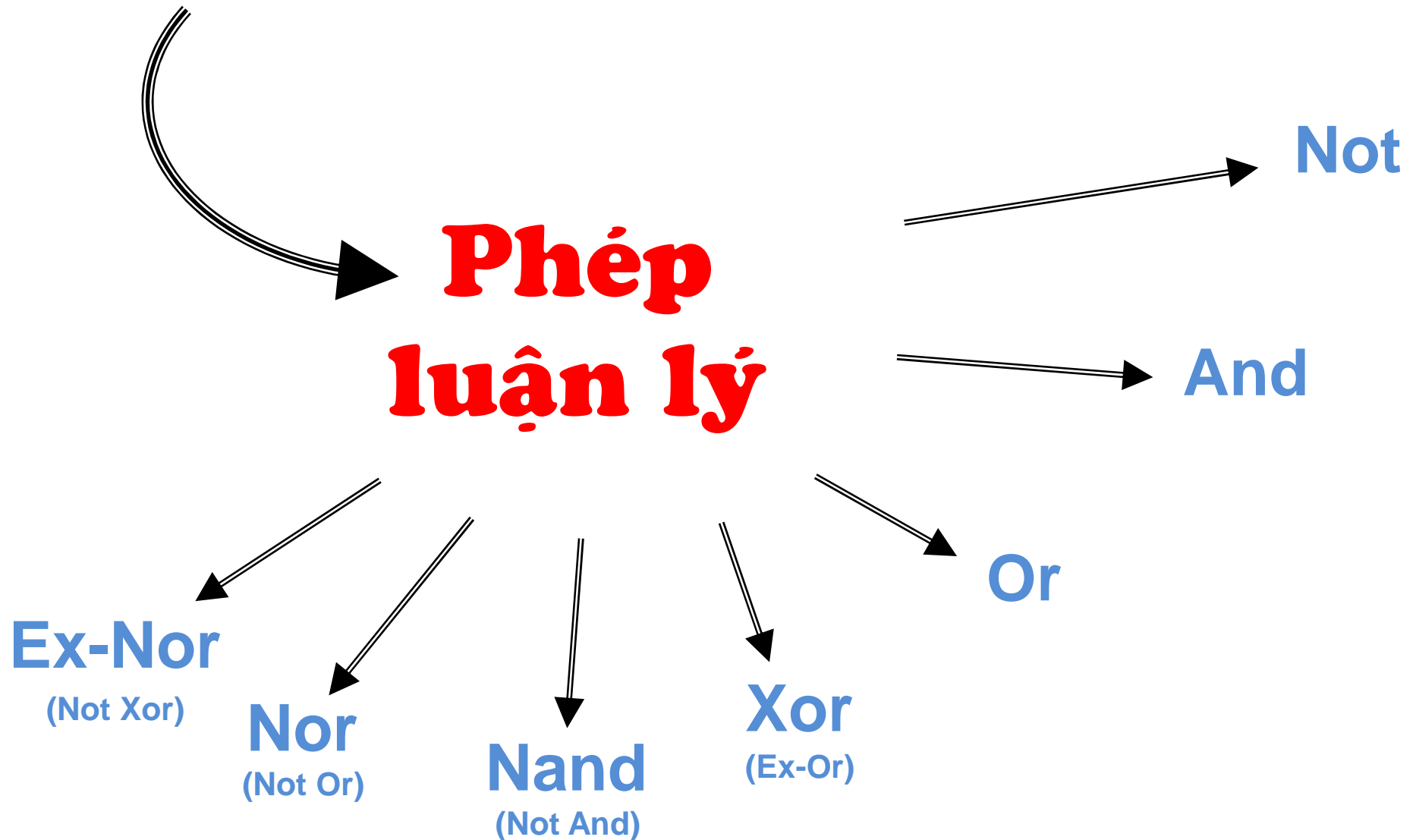


## 1.5. Luận lý máy tính

- Luận lý máy tính dựa trên nền tảng một nhánh của luận lý toán học được gọi là **đại số Boole** (George Boole).
- **Biến luận lý (boolean variable)** có hai giá trị, thường được biểu diễn bằng 1 và 0 (bit).
- Về mặt hiện thực, biến luận lý thể hiện trạng thái điện áp trên dây dẫn tín hiệu ( $1 = 5V$ ;  $0 = 0V$ ).

# Các phép toán trên đại số Boole

---





# Phép Not

---

Ký hiệu dấu gạch ngang trên đầu

Bảng sự thật

x	$\bar{x}$
0	1
1	0

$$x = 1011 \Rightarrow \bar{x} = 0100$$

$$\Rightarrow \overline{\bar{x}} = 1011 = x$$

# Phép And

Ký hiệu dấu chấm  
như phép nhân

Bảng sự thật

x	y	x . y
0	0	0
0	1	0
1	0	0
1	1	1



$$y . 0 = 0$$

$$y . 1 = y$$

# Phép Or

Bảng sự thật

x	y	x + y
0	0	0
0	1	1
1	0	1
1	1	1

Ký hiệu dấu cộng  
như phép cộng

Nhận xét

$$y + 0 = y$$

$$y + 1 = 1$$

# Ví dụ phép luận lý

---

$$\text{Tính hàm } f(x,y) = x \cdot \bar{y} + \bar{x} \cdot y$$

x	y	$\bar{x}$	$\bar{y}$	$x \cdot \bar{y}$	$\bar{x} \cdot y$	f(x,y)
0	0	1	1	0	0	0
0	1	1	0	0	1	1
1	0	0	1	1	0	1
1	1	0	0	0	0	0

# Phép Xor (Ex-Or)

Ký hiệu dấu cộng trong vòng tròn như phép modulo

Bảng sự thật

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0



$$y \oplus 0 = y$$

$$y \oplus 1 = \bar{y}$$

# Bảng tóm tắt

## Bảng sự thật

		NOT	AND	OR	XOR
x	y	not y	x and y	x or y	x xor y
0	0	1	0	0	0
0	1	0	0	1	1
1	0	1	0	1	1
1	1	0	1	1	0

$$y \text{ and } 0 = 0$$

$$y \text{ and } 1 = y$$

$$y \text{ or } 0 = y$$

$$y \text{ or } 1 = 1$$

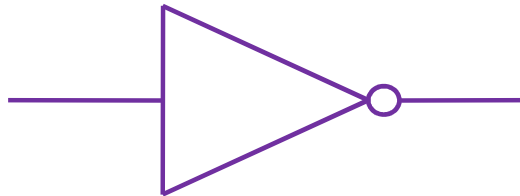
$$y \text{ xor } 0 = y$$

$$y \text{ xor } 1 = \text{not } y$$

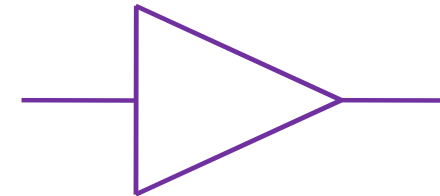
# Công luận lý

---

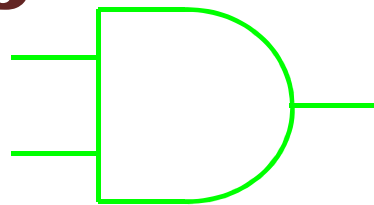
**NOT**



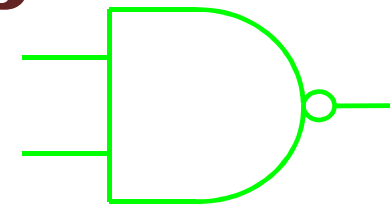
**BUFFER**



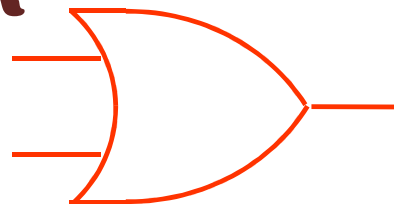
**AND**



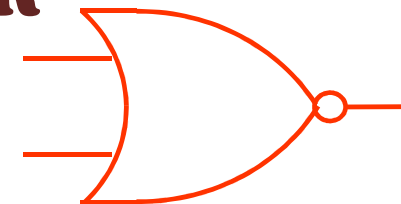
**NAND**



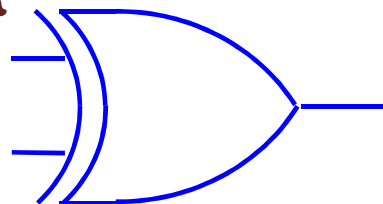
**OR**



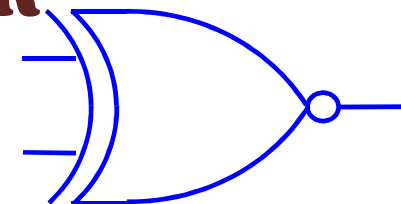
**NOR**



**XOR**

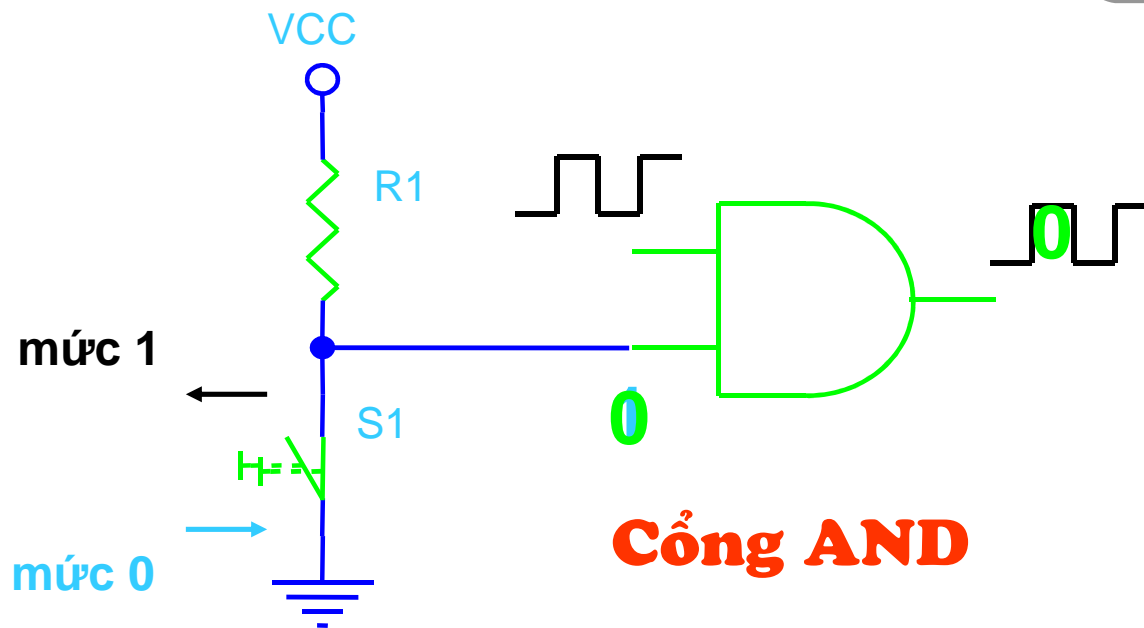


**EX-NOR**



# Chức năng đóng mở

mức luận lý 1 = 5V  
mức luận lý 0 = 0V



$y \text{ and } 1 = y$

$1 = \text{mở}$

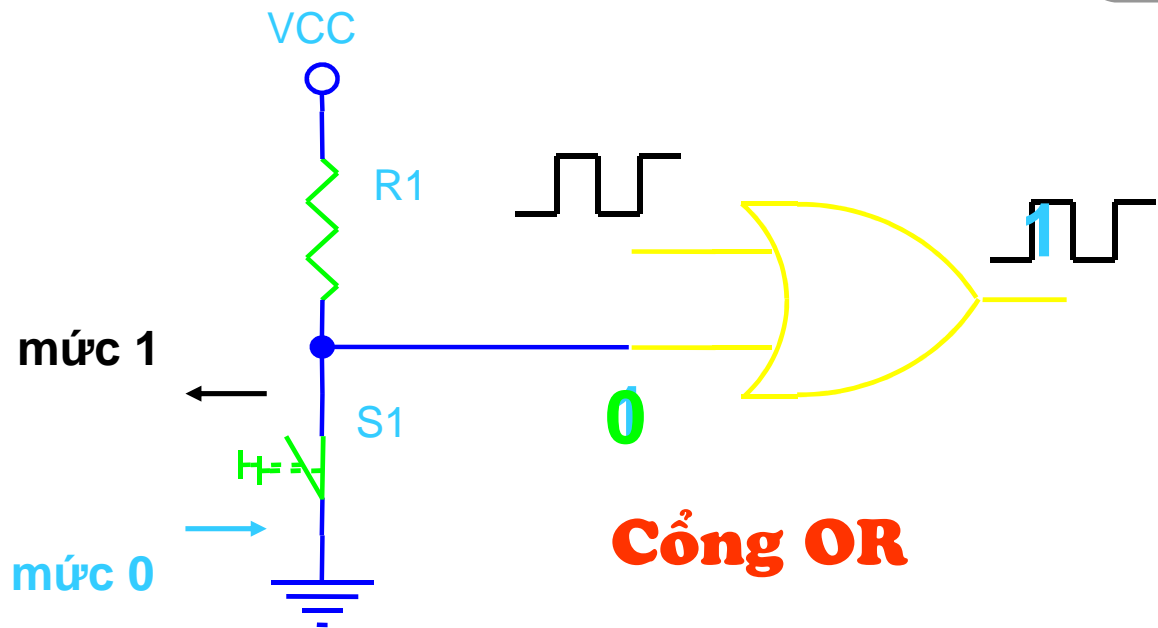
$y \text{ and } 0 = 0$

$0 = \text{đóng}$



# Chức năng đóng mở (tt.)

mức luận lý 1 = 5V  
mức luận lý 0 = 0V



$y \text{ or } 1 = 1$

$1 = \text{đóng}$

$y \text{ or } 0 = y$

$0 = \text{mở}$

# Ứng dụng đơn giản của công luận lý

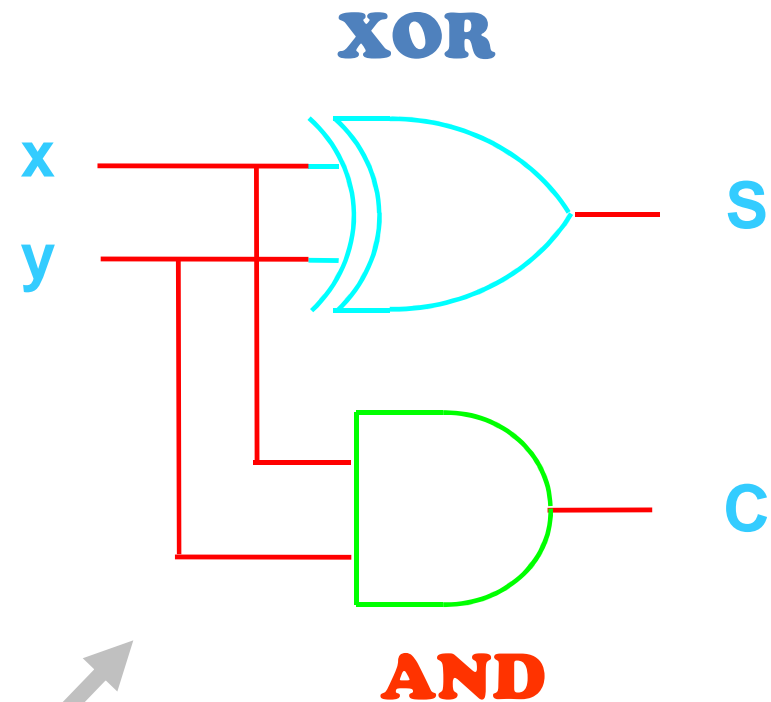
---

- Mạch cộng bán phần thực hiện phép cộng trên hai bit, cho ra kết quả là bit tổng  $S$  và bit nhớ  $C$ .
- Mạch cộng toàn phần cũng tương tự mạch cộng bán phần nhưng đầu vào có cộng thêm bit nhớ  $C_0$ .
- Mạch cộng toàn phần có thể được thiết kế dựa vào mạch cộng bán phần.

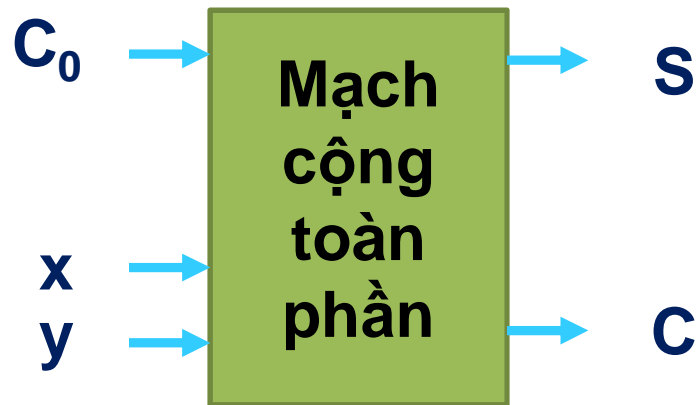
# Mạch cộng bán phần



x	y	XOR	AND
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



# Mạch cộng toàn phần



$$S = x + y + C_0$$
$$S = (x + y) + C_0$$

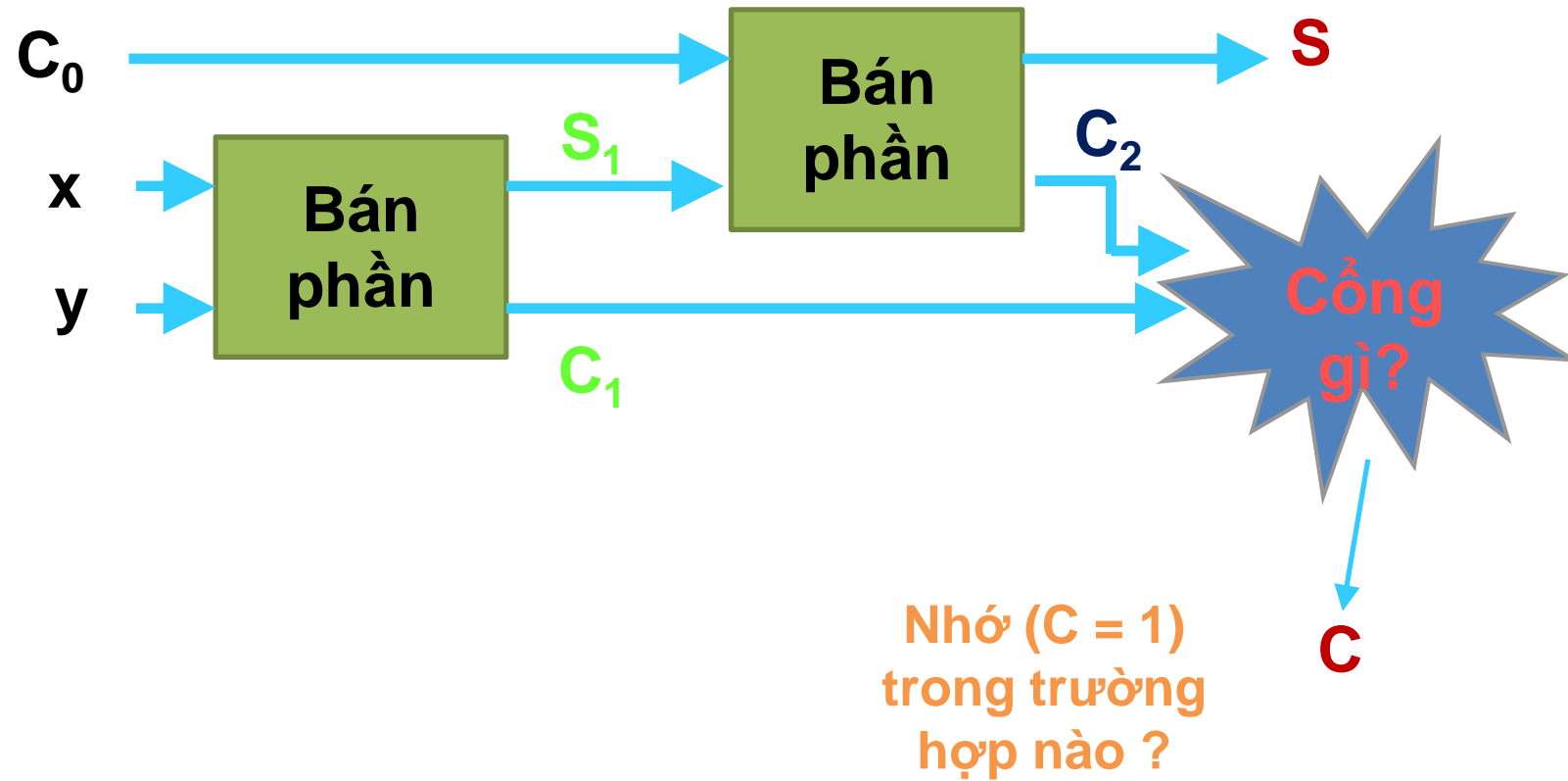
Tính:  $S_1 = x + y$

Tính:  $S_2 = S_1 + C_0$

Cần bộ cộng bán phần 1

Cần bộ cộng bán phần 2

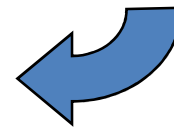
# Mạch cộng toàn phần (tt.)

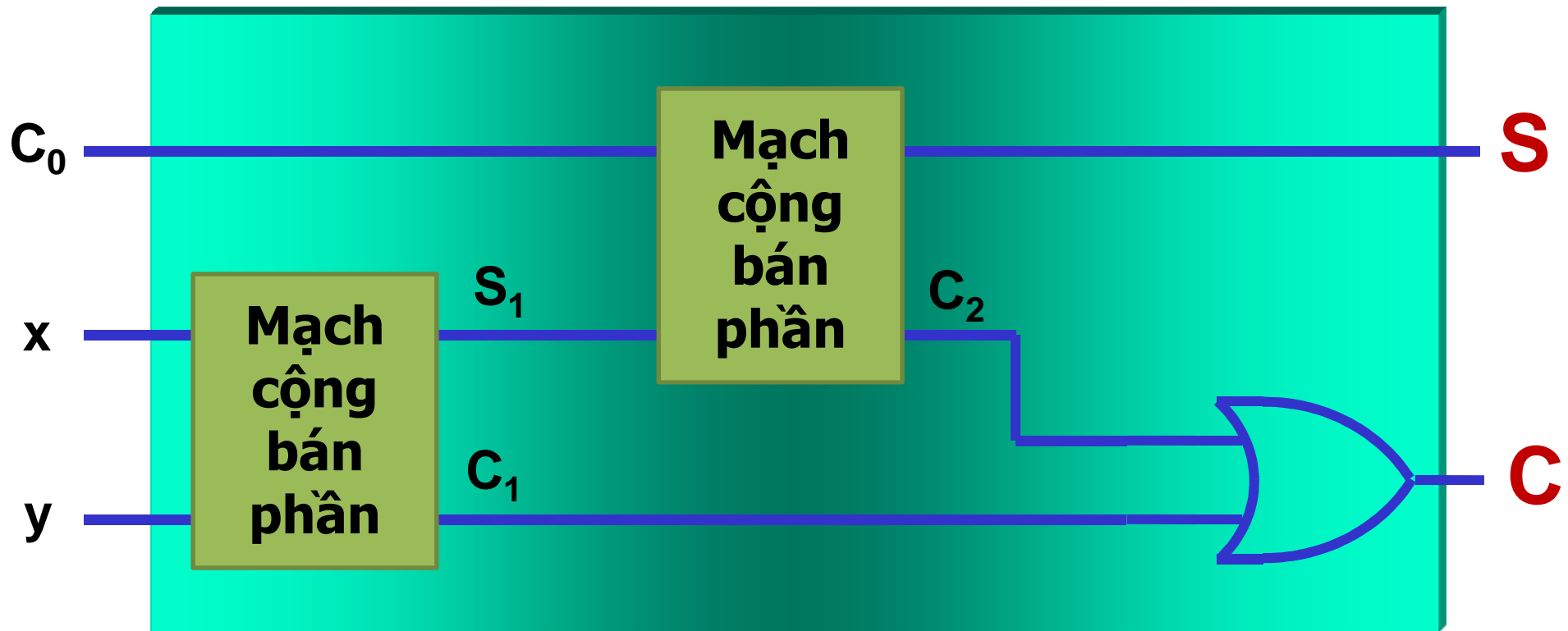


# Mạch cộng toàn phần (tt.)

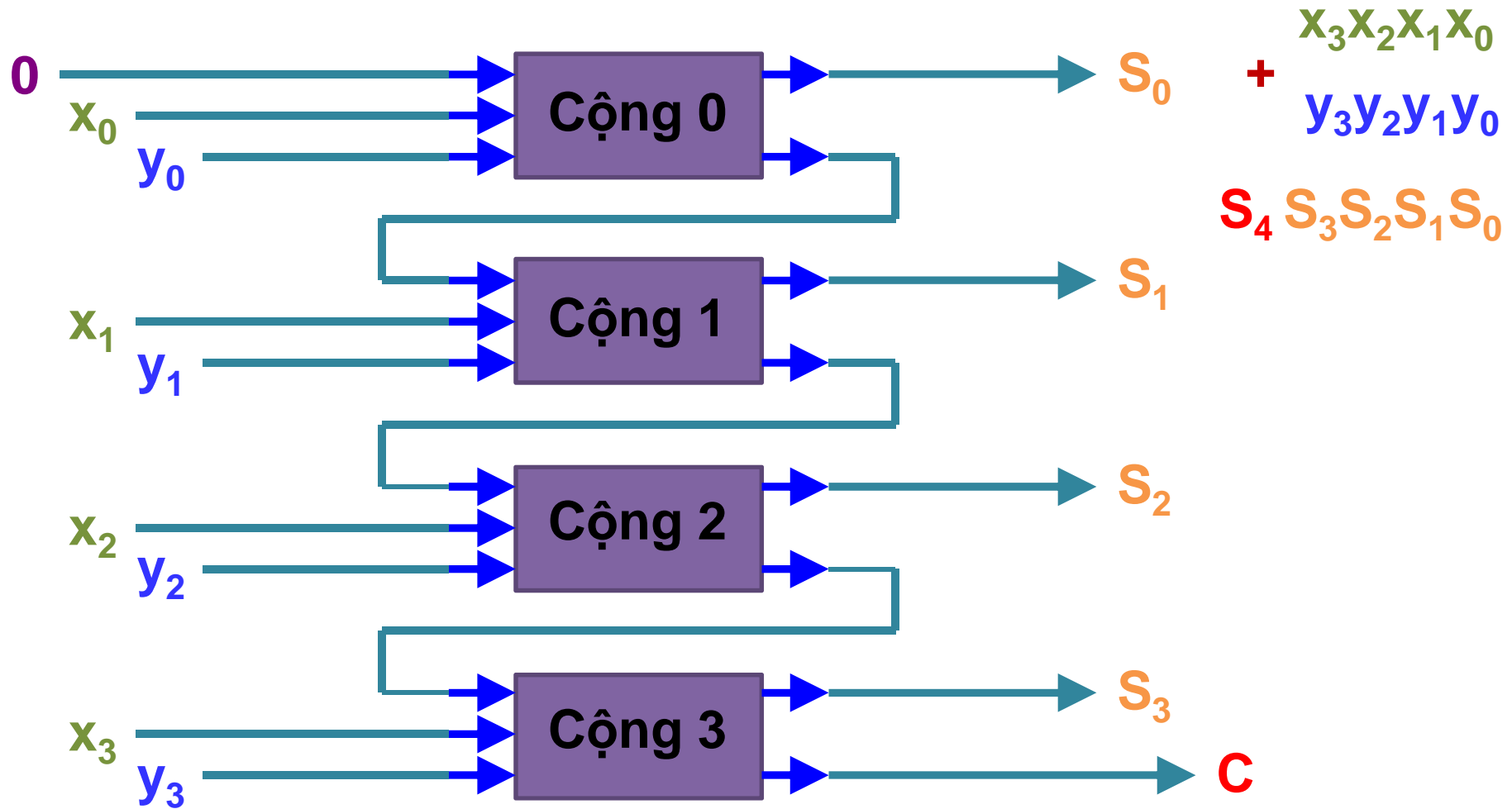
$C_0$	x	y	S	C	$C_0$	$S_1$	$C_1$	$C_2$	C
0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	0	0	0
0	1	0	1	0	0	1	0	0	0
0	1	1	0	1	0	0	1	0	1
1	0	0	1	0	1	0	0	0	0
1	0	1	0	1	1	1	0	1	1
1	1	0	0	1	1	1	0	1	1
1	1	1	1	1	1	0	1	0	1

$C = 1$  khi  $C_1 = 1$  hoặc  $C_2 = 1$





# Mạch cộng nhiều bit





# *Chương 2. PHẦN CỨNG*

---

2.1 Hệ thống máy tính

2.2 Kiến trúc máy tính

2.3 Thiết bị xuất nhập

# 1. Hệ thống máy tính

---

- Hệ thống máy tính có các khối chức năng sau :
  - Khối nhập (input).
  - Bộ nhớ chính (memory).
  - Đơn vị xử lý trung tâm CPU (Central processing unit).
  - Khối xuất (output).
  - Bộ nhớ phụ (storage).
  - Thiết bị ngoại vi (peripherals).

# Khối nhập - Input



- Giữ vai trò nhận dữ liệu cho máy tính.
- Có nhiệm vụ chuyển đổi các thông tin từ thế giới ngoài thành dữ liệu mà máy tính có thể xử lý.
- Có rất nhiều thiết bị có thể làm việc này nhưng bàn phím (keyboard) là thiết bị được dùng phổ biến nhất.



# Bộ nhớ chính - Main memory

- ❑ Còn gọi là bộ nhớ RAM hay bộ nhớ bán dẫn.
- ❑ Có 2 chức năng chính :
  - ❖ Chứa tạm chương trình đang được sử dụng để xử lý thông tin.
  - ❖ Chứa tạm dữ liệu.
- ❑ Dữ liệu dùng trong máy tính có 3 loại :
  - ❖ Dữ liệu ban đầu nhận từ khối nhập.
  - ❖ Dữ liệu trung gian đang được xử lý.
  - ❖ Kết quả cuối cùng chờ đưa ra khối xuất.



# Đơn vị xử lý trung tâm - CPU

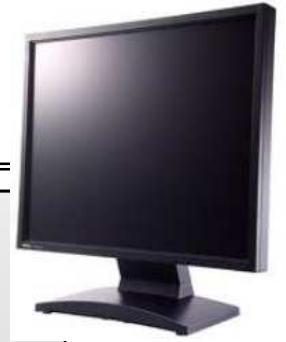


- ❑ Thường còn gọi là bộ xử lý (processor), vi xử lý (micro-processor).
- ❑ CPU có nhiệm vụ thi hành lệnh của chương trình và xử lý các dữ liệu trong chương trình.
- ❑ Trong CPU có 2 phần chính :
  - ❖ Đơn vị số học luận lý ALU (Arithmetic / logic unit).
  - ❖ Đơn vị điều khiển (control unit).
- ❑ ALU dùng để tính toán các phép số học (cộng, trừ, nhân, chia) và các phép luận lý (not, and, or, xor).
- ❑ Đơn vị điều khiển chi phối toàn bộ hoạt động của máy tính bằng cách **lấy lệnh** từ bộ nhớ, **giải mã lệnh** và **thực hiện lệnh** đó.



# Khối xuất - Output

- ❑ Ngược lại với khối nhập, khối xuất chuyển dữ liệu mà máy xử lý (số nhị phân) ra thành dạng thông tin mà con người có thể chấp nhận.
- ❑ Hai thiết bị thông dụng dùng trong khối này là màn hình và máy in.
- ❑ Đôi khi các thông tin mà máy tính đưa ra cần được xử lý tiếp sau này nên còn phải được lưu trên bộ nhớ phụ (chủ yếu là trên đĩa từ).





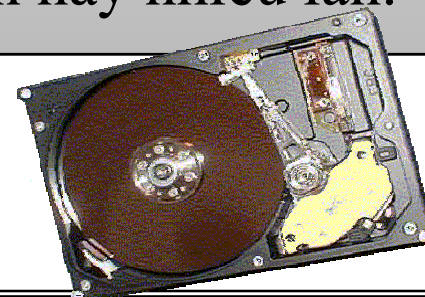
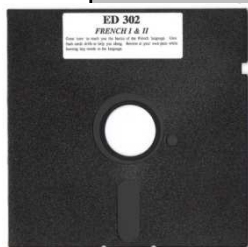
# Bộ nhớ phụ - Storage



- ❑ Cung cấp cho máy tính chức năng lưu trữ, sắp xếp, phân loại thông tin theo dạng tập tin (file).
- ❑ Cần phân biệt hai khái niệm sau :

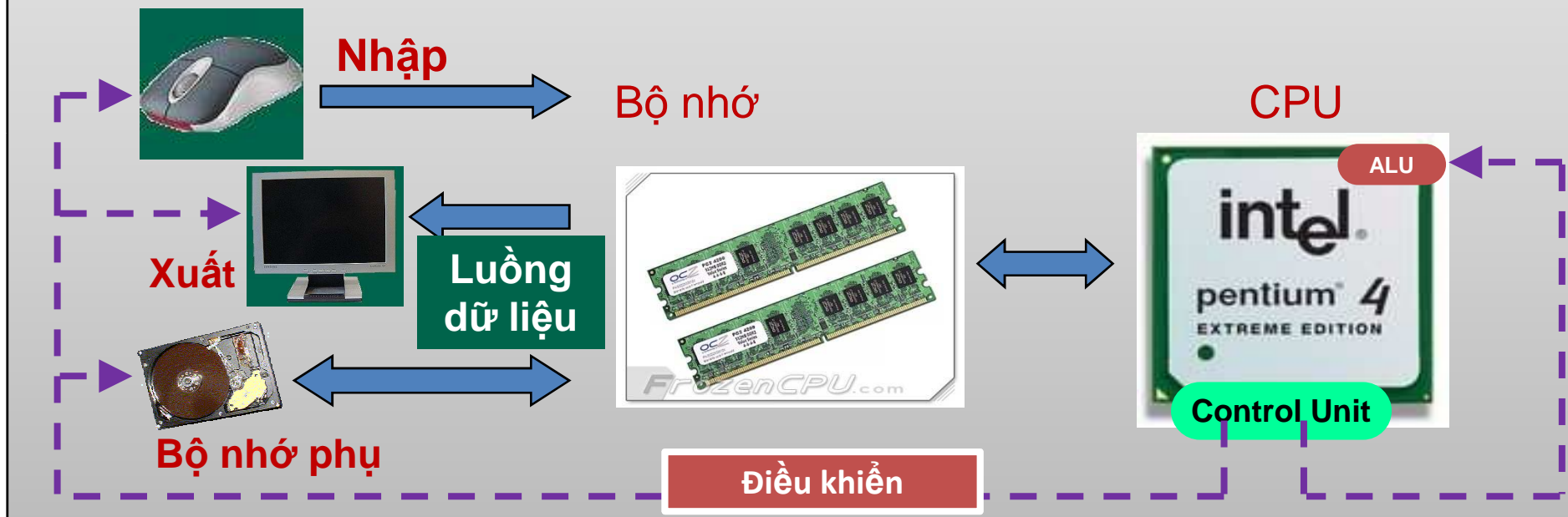
❖ **Bộ nhớ bốc hơi (memory volatility)** : là bộ nhớ mà thông tin lưu giữ trong nó sẽ bị mất đi, hoặc là do tắt máy, hoặc là do thông tin khác ghi chồng lên. Chính vì vậy nên loại bộ nhớ này còn được gọi là **RAM (Random Access Memory)**. Bộ nhớ chính của máy tính là bộ nhớ bay hơi.

❖ **Dữ liệu có thể dùng lại (retrievable data)** : bộ nhớ phụ có thể giữ chương trình hay dữ liệu lâu dài mà không bị bốc hơi. Điều đó cho phép ta có thể sử dụng lại các thông tin này nhiều lần.



# Thiết bị ngoại vi - Peripherals

- ❑ Thiết bị ngoại vi là các thiết bị phụ trợ xung quanh CPU và bộ nhớ chính.
- ❑ Các thiết bị đáp ứng chức năng của các khối **nhập**, **xuất** và **bộ nhớ phụ** đều là thiết bị ngoại vi.



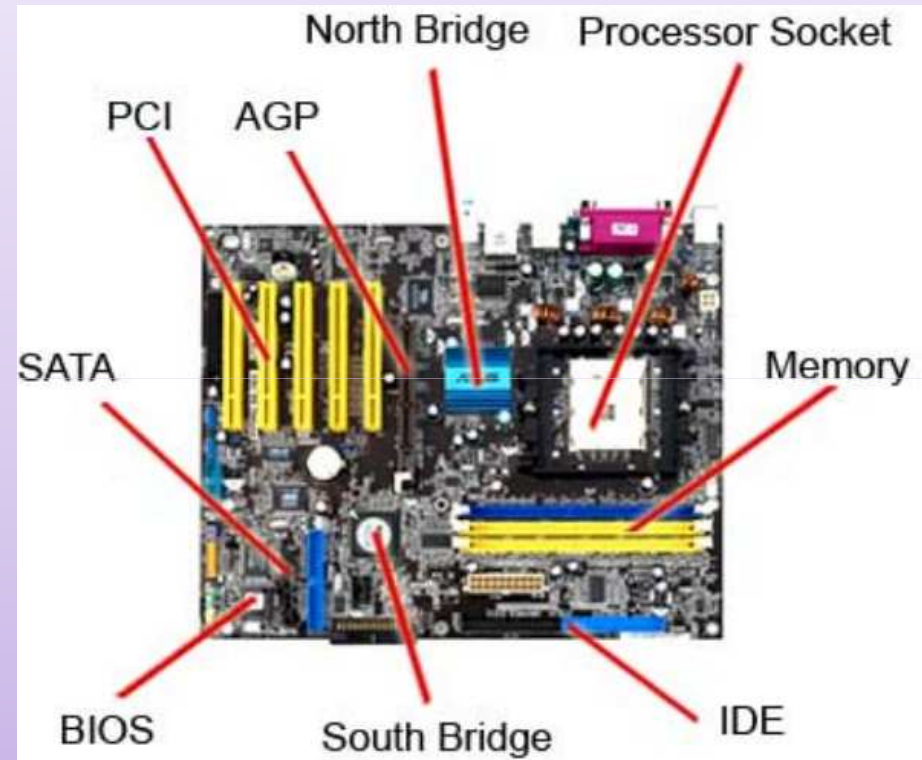
Cấu trúc luận lý của một máy tính



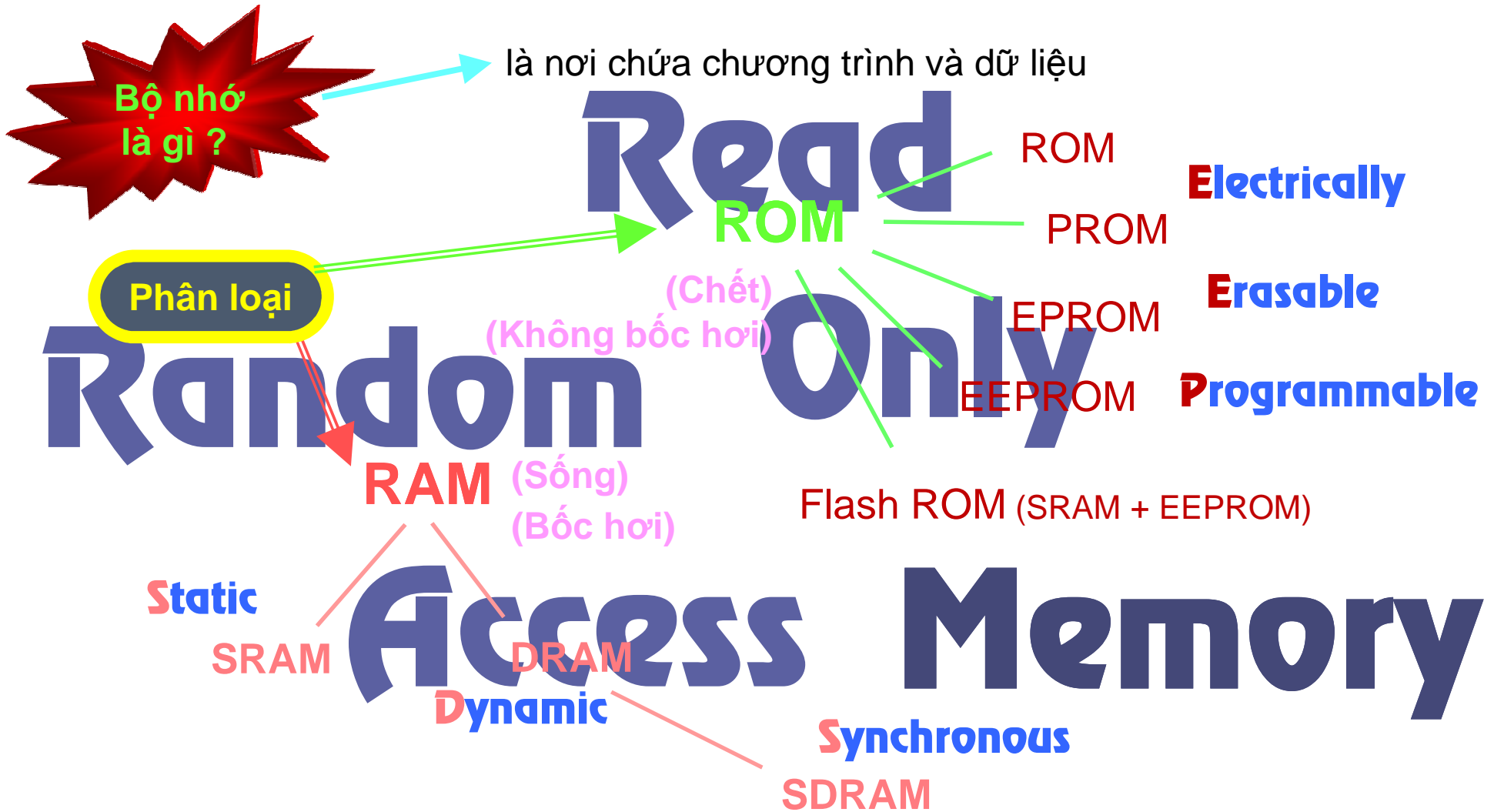
## 2. Kiến trúc máy tính

- Kiến trúc máy tính ngày nay được biết đến như là một hệ thống gồm có :

- Bộ nhớ (memory).
- Bộ xử lý (processor).
- Các tuyến (buses).

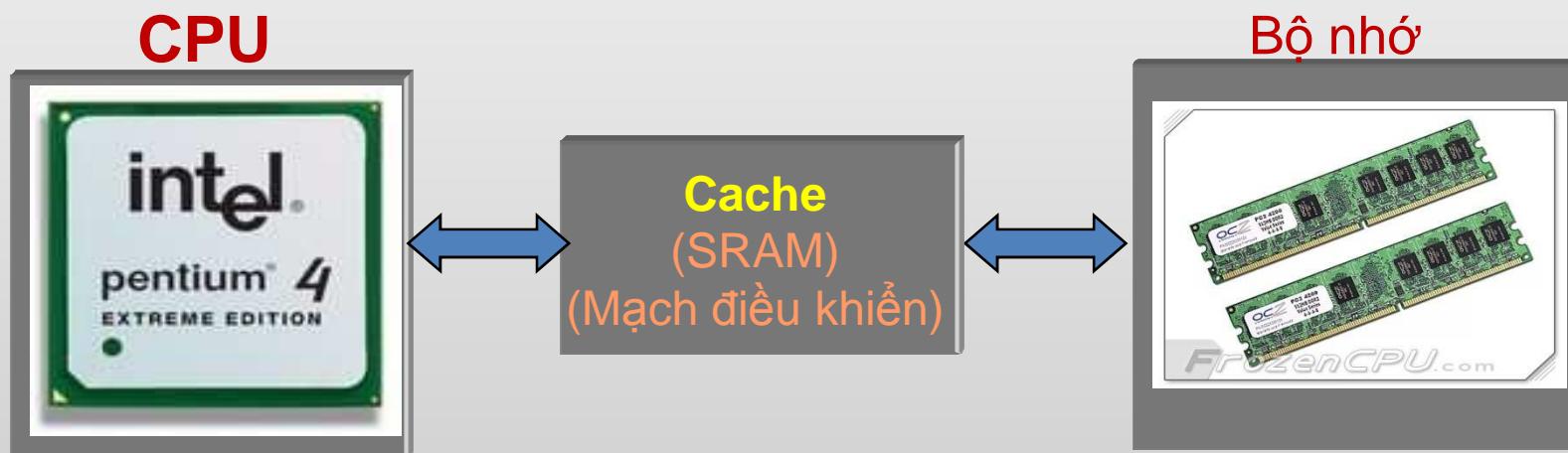


# Bộ nhớ



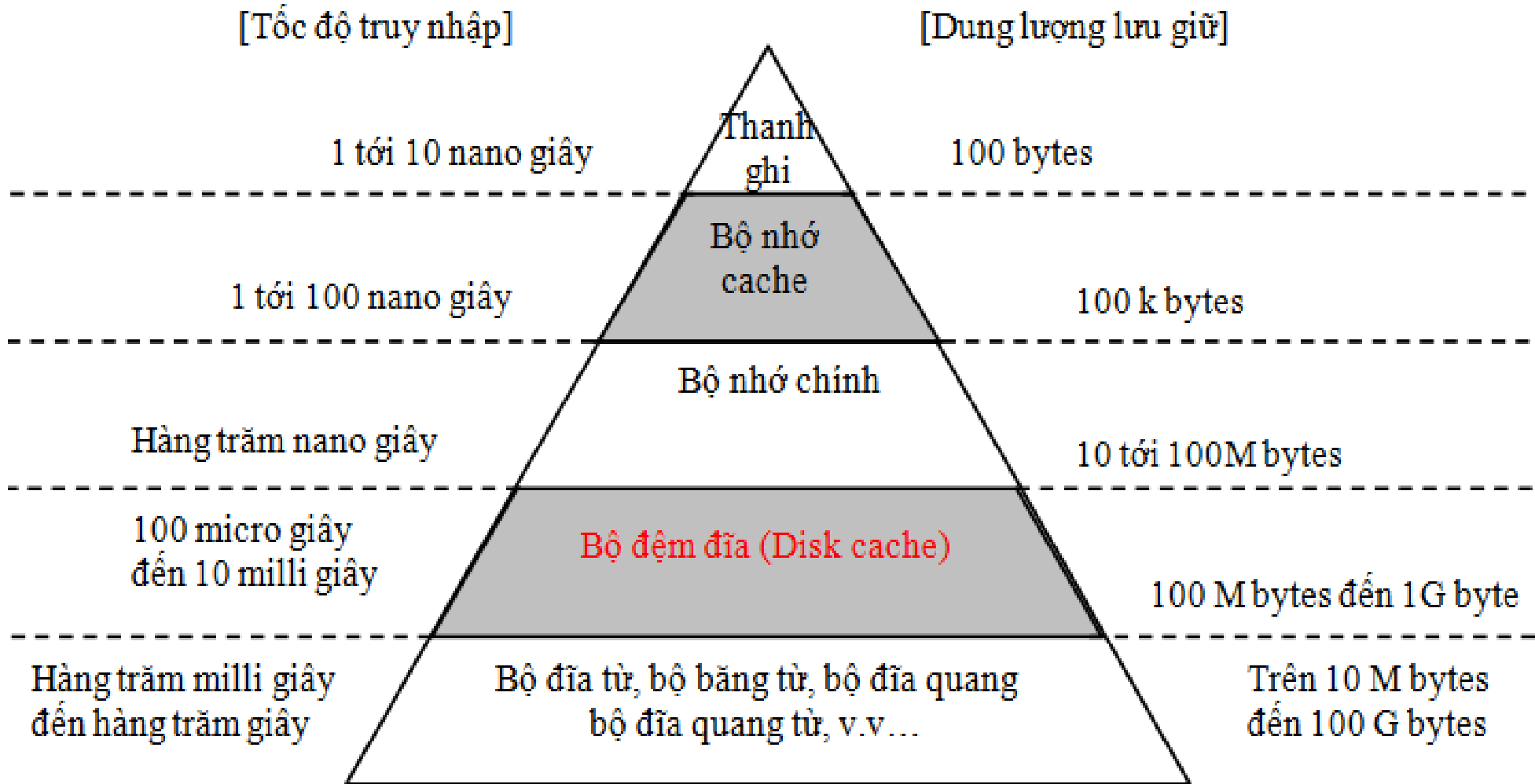
# Bộ nhớ đệm - Cache

- Cache là bộ nhớ đệm giữa CPU và bộ nhớ chính



- Cache được chế tạo từ SRAM có tốc độ làm việc rất cao và có dung lượng nhỏ.
- Nhiệm vụ của cache là làm giảm thời gian đợi (wait-state) của CPU khi truy xuất bộ nhớ chính bằng cơ chế đọc trước các ô nhớ kế tiếp.
- Các bộ xử lý hiện đại đều có cache bên trong.

# Cấu trúc phân cấp của bộ nhớ



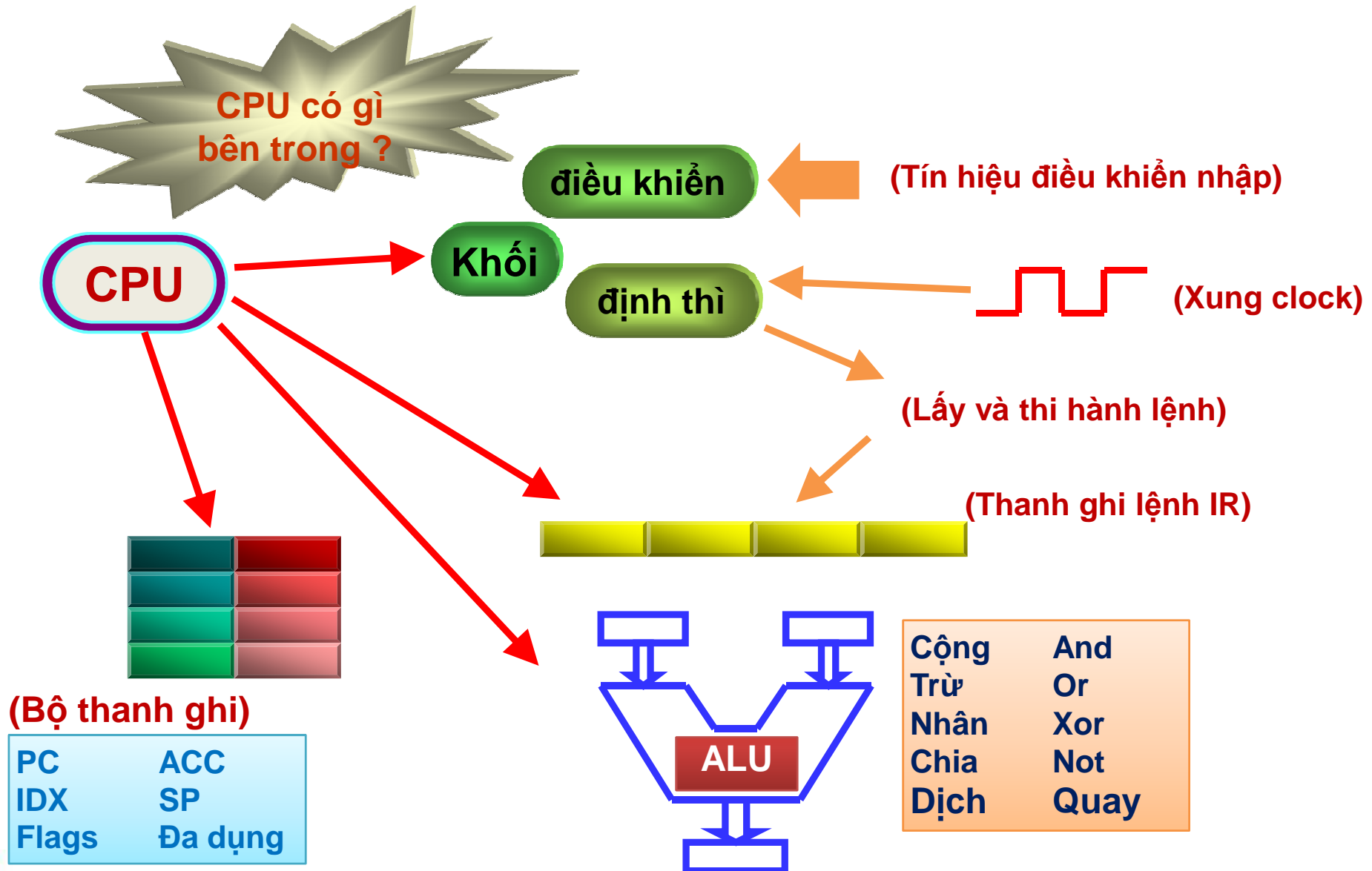
Nguồn: [www.vitec.org](http://www.vitec.org)

# Bộ xử lý - Processor

- ❑ Bộ xử lý hay còn gọi là **CPU** (*Central Processing Unit*) là nguồn phát sinh mọi hoạt động của máy tính.
- ❑ Bộ xử lý điều khiển hoạt động của máy tính thông qua việc lấy và thi hành lệnh nằm trong bộ nhớ.



# CPU



# Kiến trúc bộ xử lý

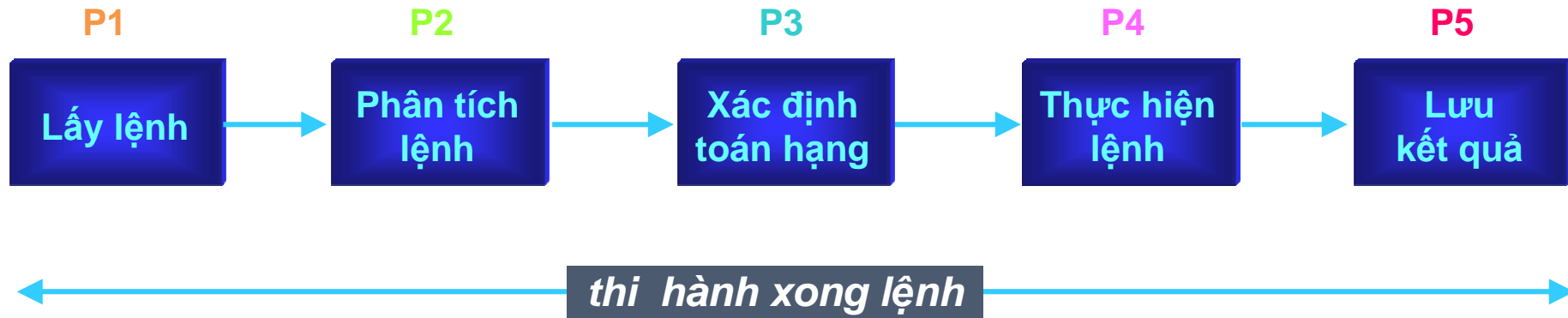
## ❑ Kiến trúc **CISC** (Complex Instruction Set Computer)

- ❖ Các lệnh của CPU có chiều dài khác nhau.
- ❖ Thời gian thi hành lệnh cũng khác nhau.

## ❑ Kiến trúc **RISC** (Reduced Instruction Set Computer)

- ❖ Các lệnh dài bằng nhau.
- ❖ Thời gian thi hành các lệnh chỉ bằng 1 chu kỳ xung clock.
- ❖ Cung cấp khả năng thi hành nhiều hoạt động cùng lúc (Super scalar execution).
- ❖ Dùng cơ chế đường ống (Pipelining) để giảm thời gian thi hành.
- ❖ Vấn đề đoán trước rẽ nhánh (Branch prediction).

# Cơ chế đường ống - Pipelining





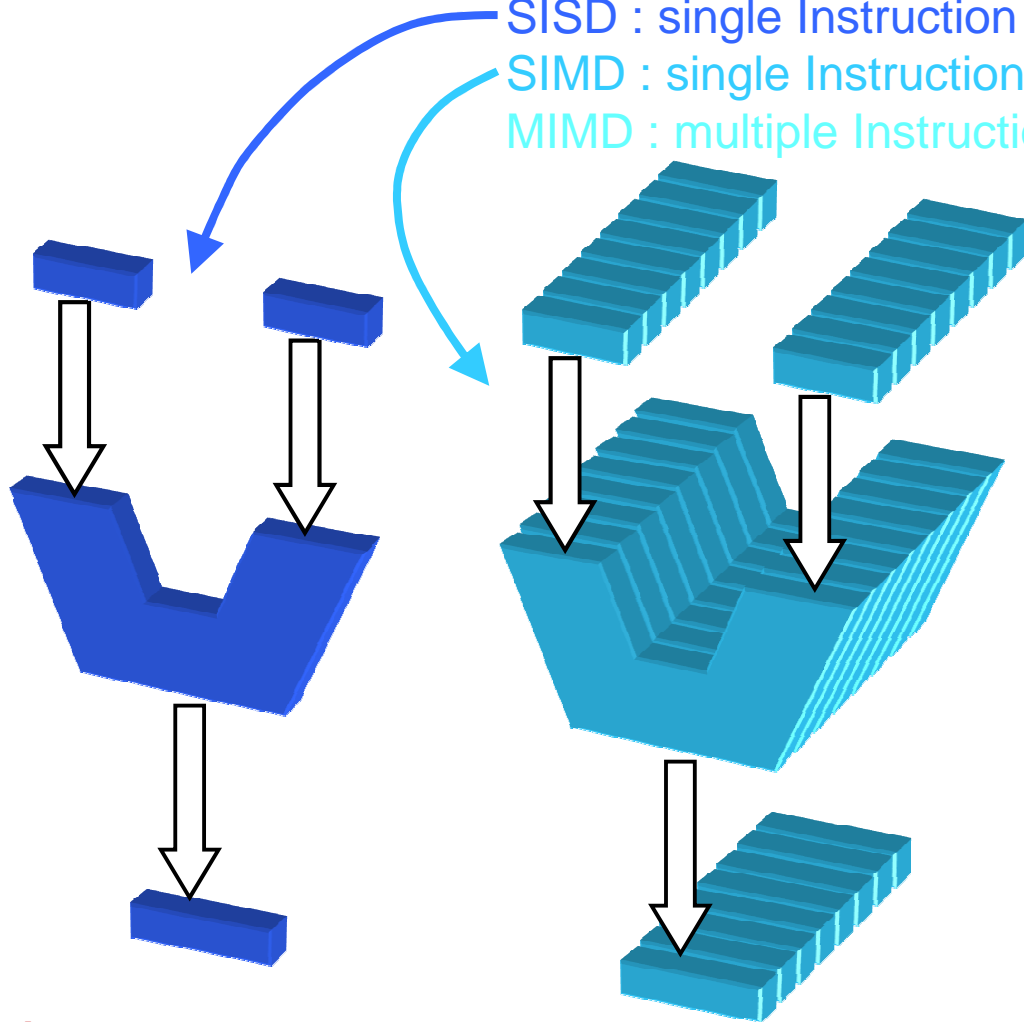
# Máy tính song song

## 3 loại máy song song

SISD : single Instruction stream, single data stream

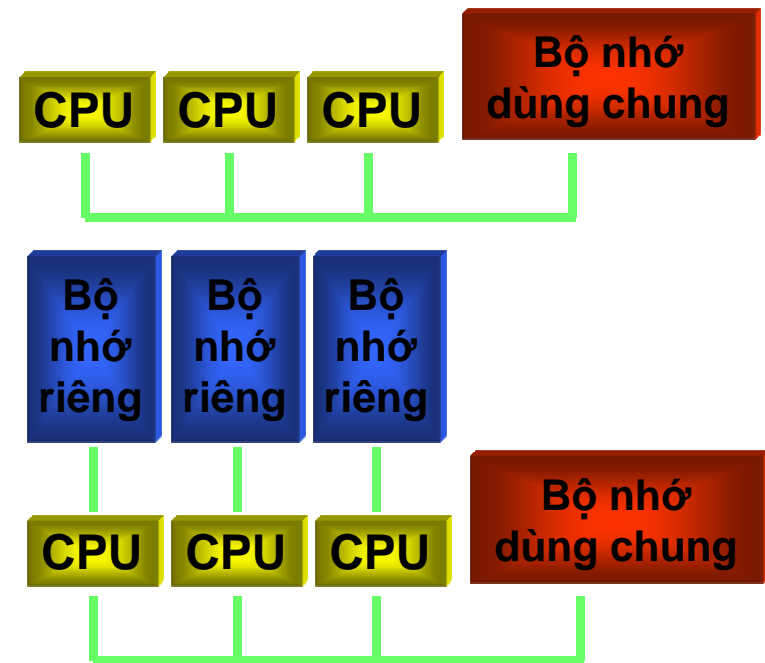
SIMD : single Instruction stream, multiple data stream

MIMD : multiple Instruction stream, multiple data stream



Máy Von Neumann

Máy Vector 8 ALU

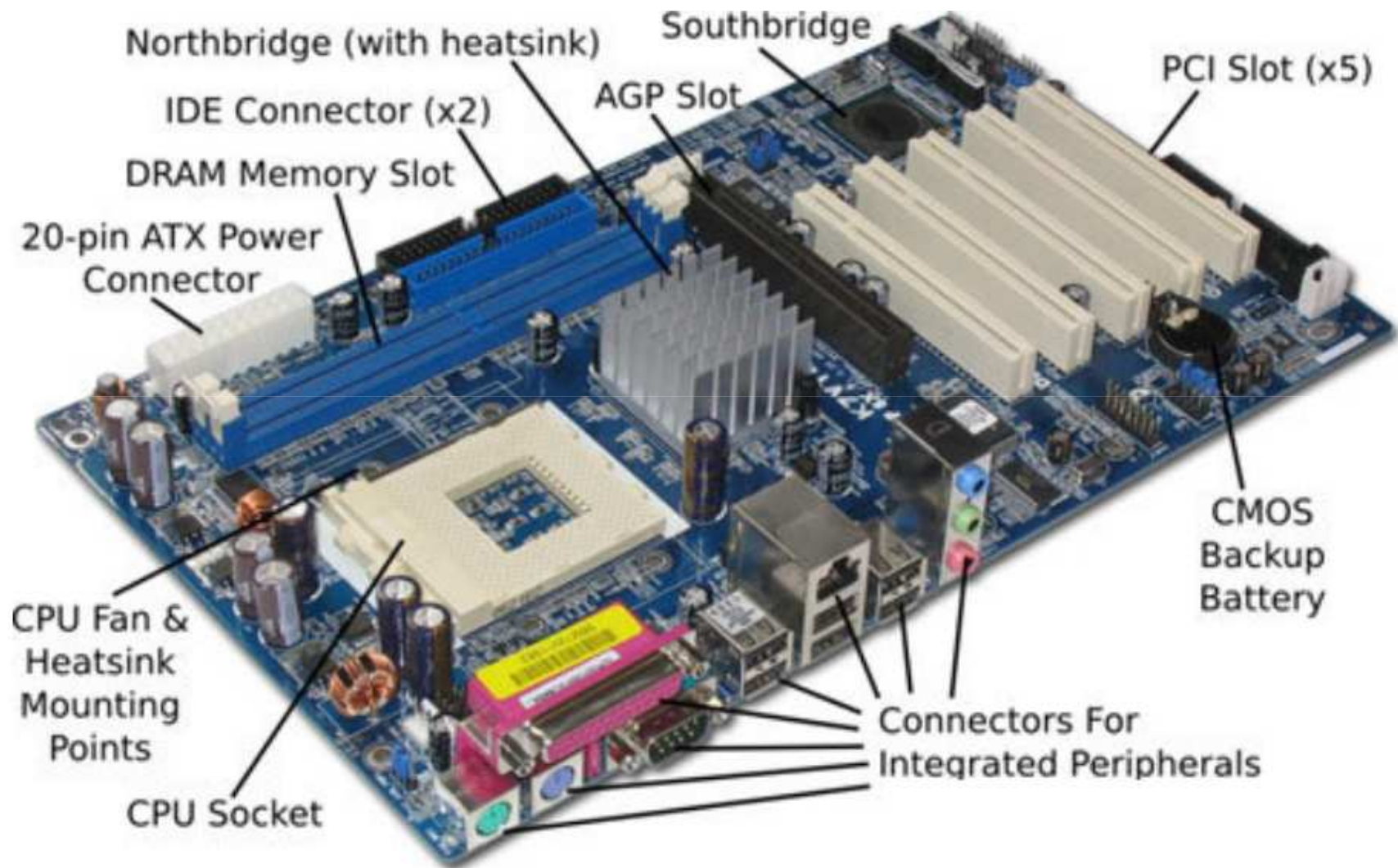


# Tuyến - Bus

- ❑ Tuyến là một nhóm các dây dẫn song song mà mỗi đường có nhiệm vụ truyền tải 1 bit thông tin.
- ❑ Tuyến hệ thống là tuyến kết nối giữa CPU với các bộ phận mà nó muốn trao đổi thông tin mà cụ thể là bộ nhớ và khối xuất nhập (I/O).
- ❑ Trên một tuyến có thể truyền tải nhiều loại thông tin khác nhau.
- ❑ Một số tuyến có khả năng truyền thông tin theo cả 2 chiều. Tuy nhiên, trong từng thời điểm, luồng dữ liệu chỉ đi một chiều.
- ❑ Độ rộng của tuyến (số đường) xác định chiều dài của một từ (word) thông tin mà CPU trao đổi mỗi lần.

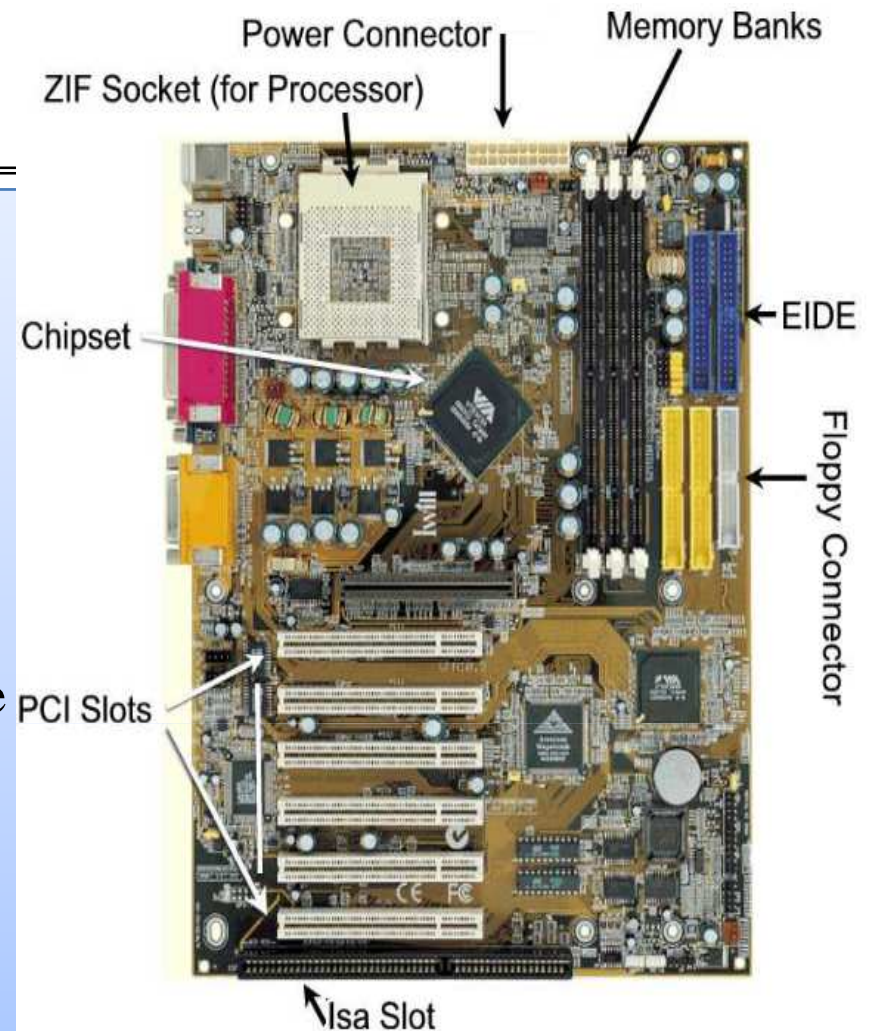
**Ví dụ : CPU dùng bus 16 bit để truyền dữ liệu 32 bit thì phải thực hiện 2 lần.**

# Tuyến - Bus



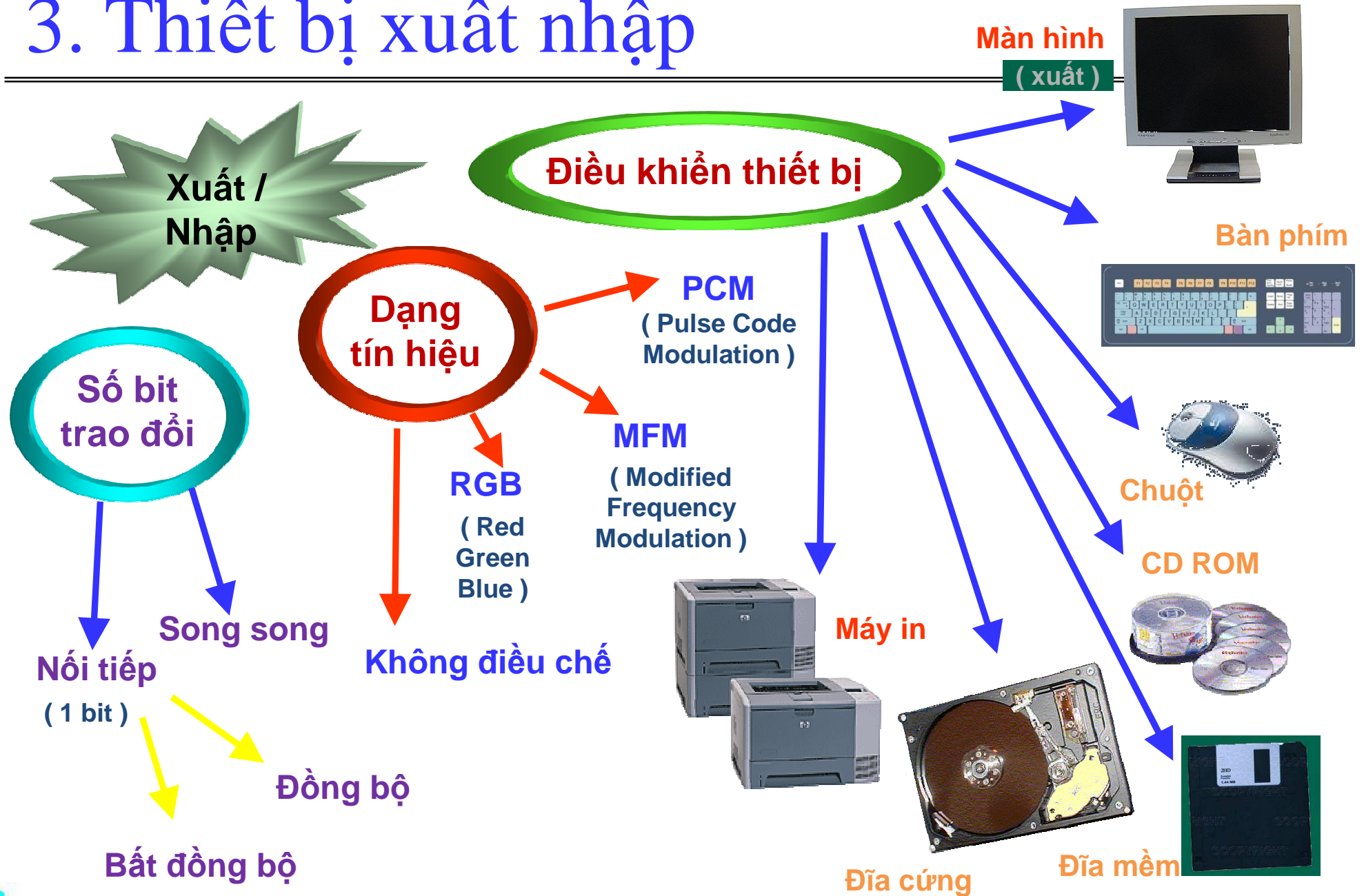
# Kiến trúc tuyến

- Tuyến chuẩn (standard bus) :
  - MCA : micro channel architecture
  - ISA : industry standard architecture
  - IBM AT : advanced technology
  - PS/2 : personal system 2
  - EISA : extended industry standard architecture
- Tuyến cục bộ (local bus) :
  - VESA : video electronics standard association
  - PCI : Peripheral Component Interface
  - AGP : Accelerated Graphics Port



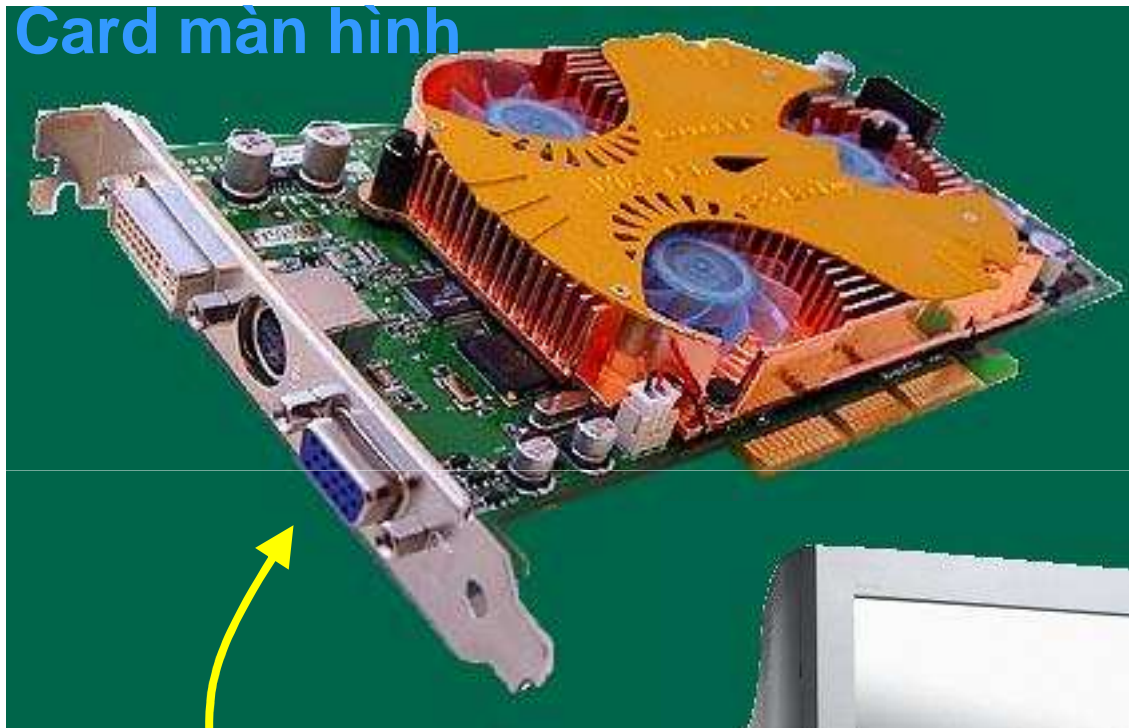


# 3. Thiết bị xuất nhập



# Màn hình và card màn hình

Card màn hình



Màn hình LCD

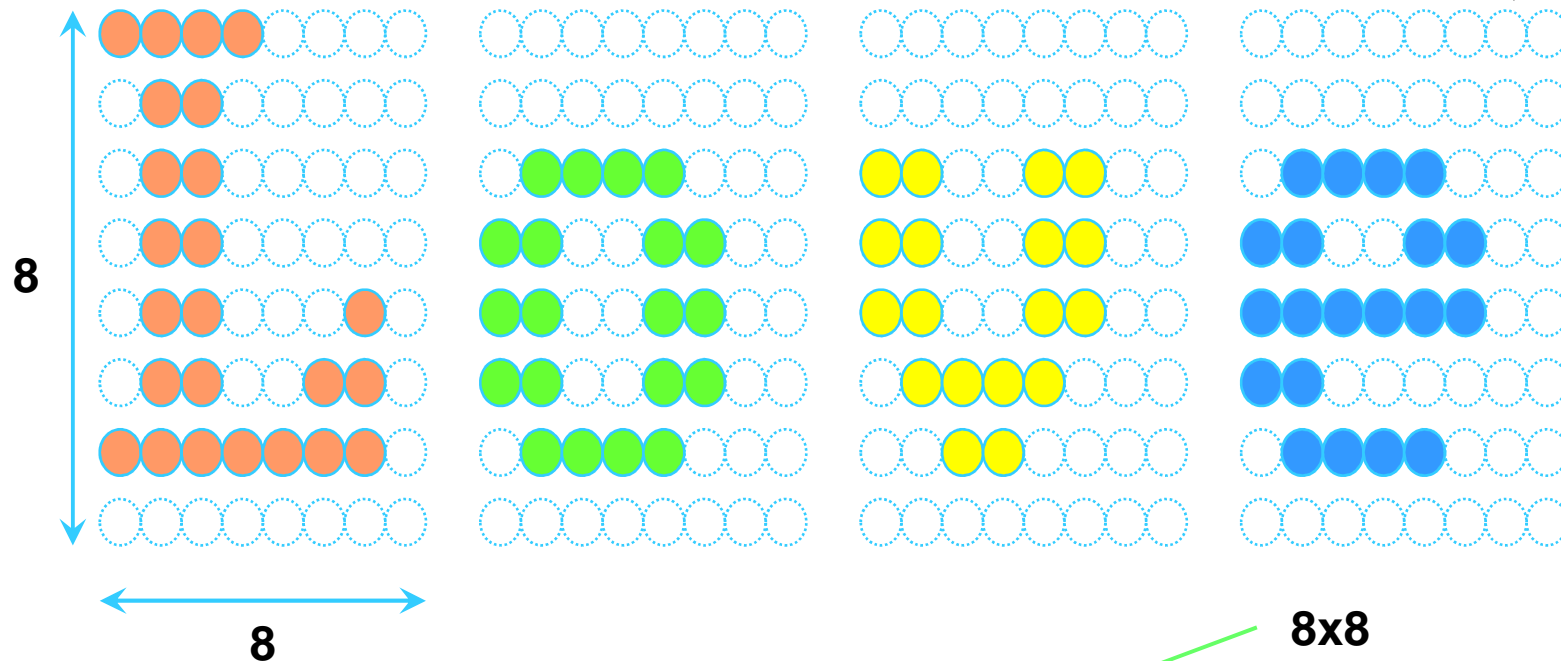


Màn hình CRT

# Hiện thị trong chế độ văn bản (text)

## Ma trận điểm

Chế độ  
văn bản



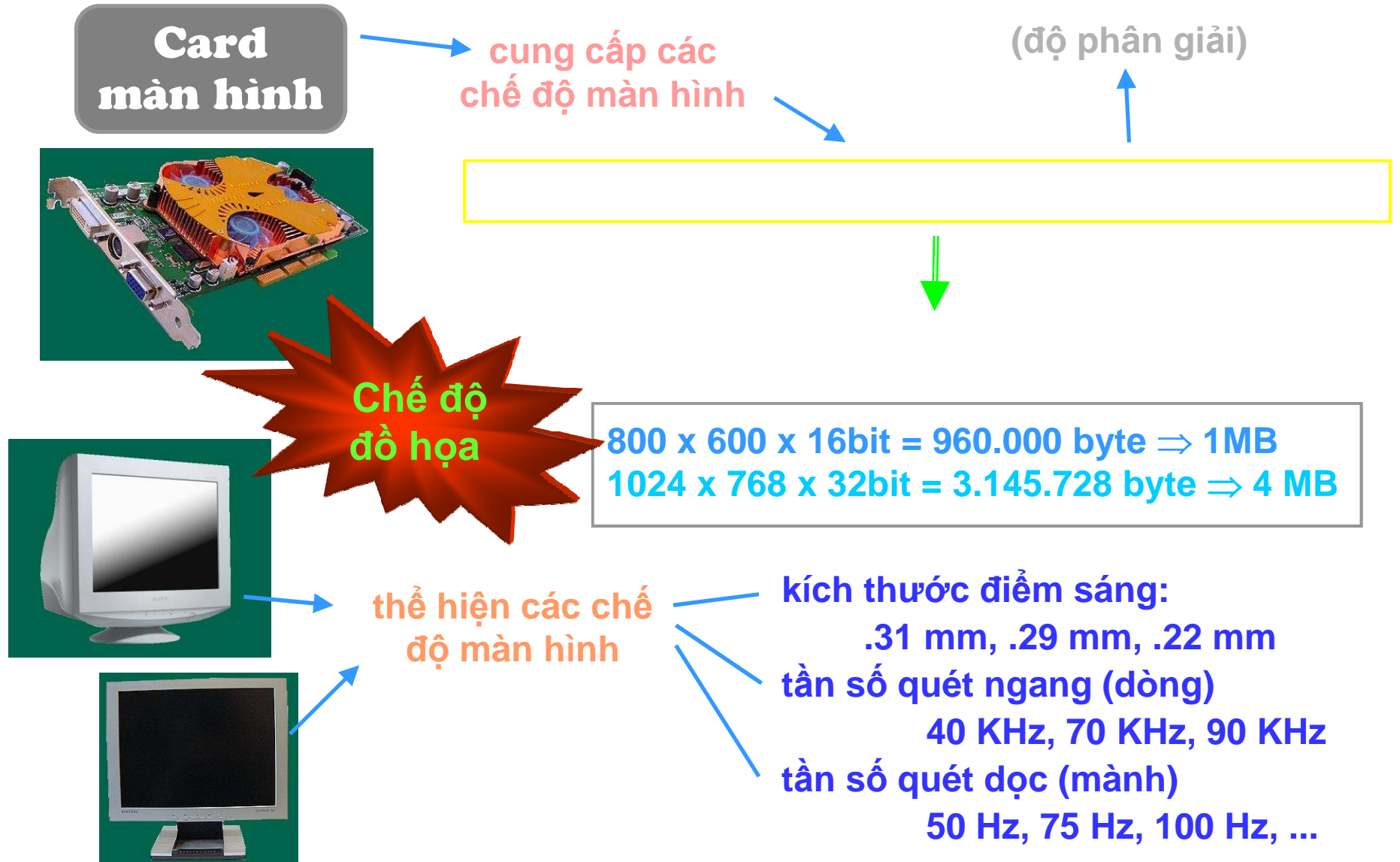
Kích thước

8x8

14x8

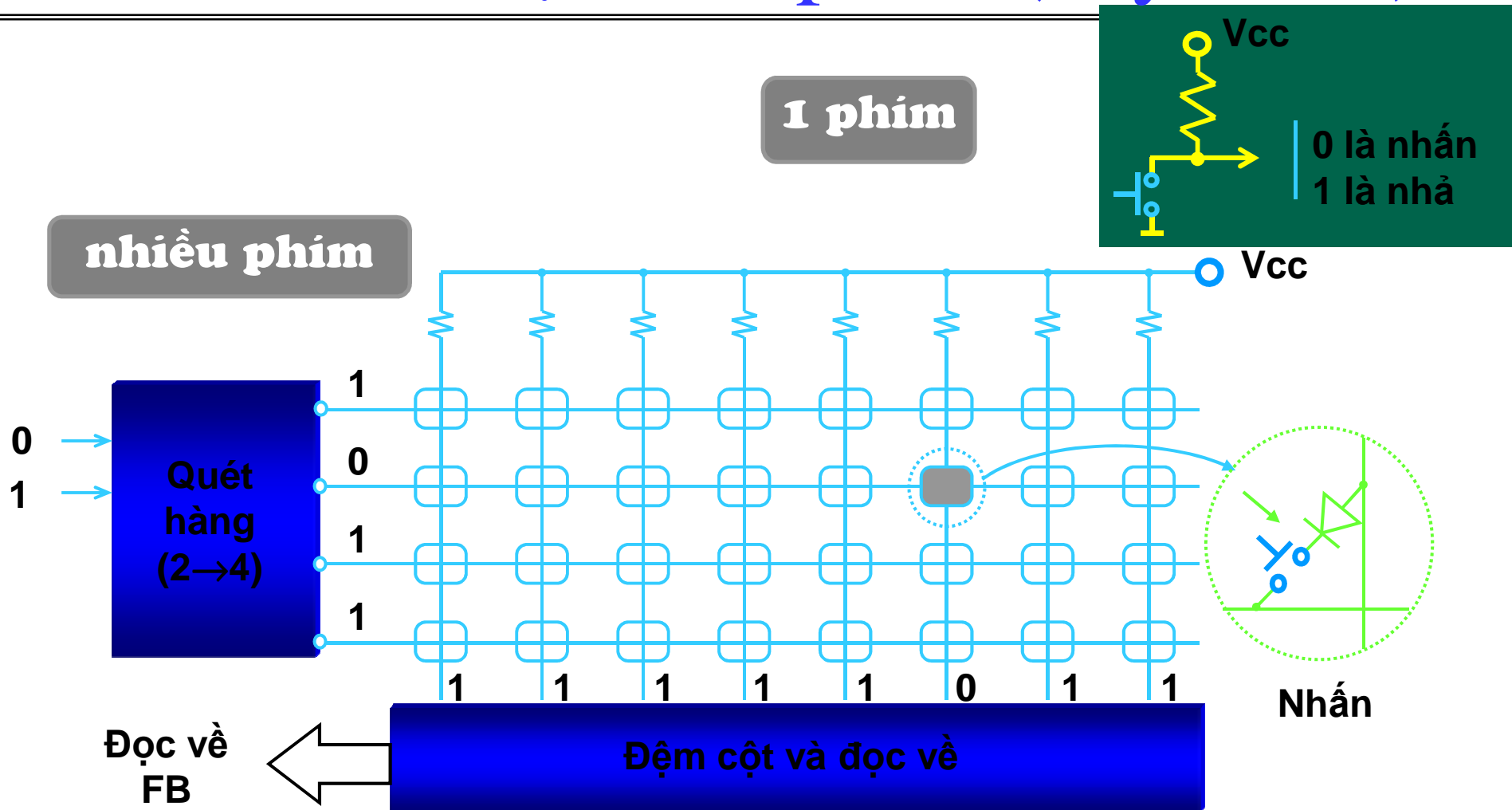
16x8

# Hiển thị trong chế độ đồ họa (graphics)





# Tổ chức ma trận bàn phím (keyboard)



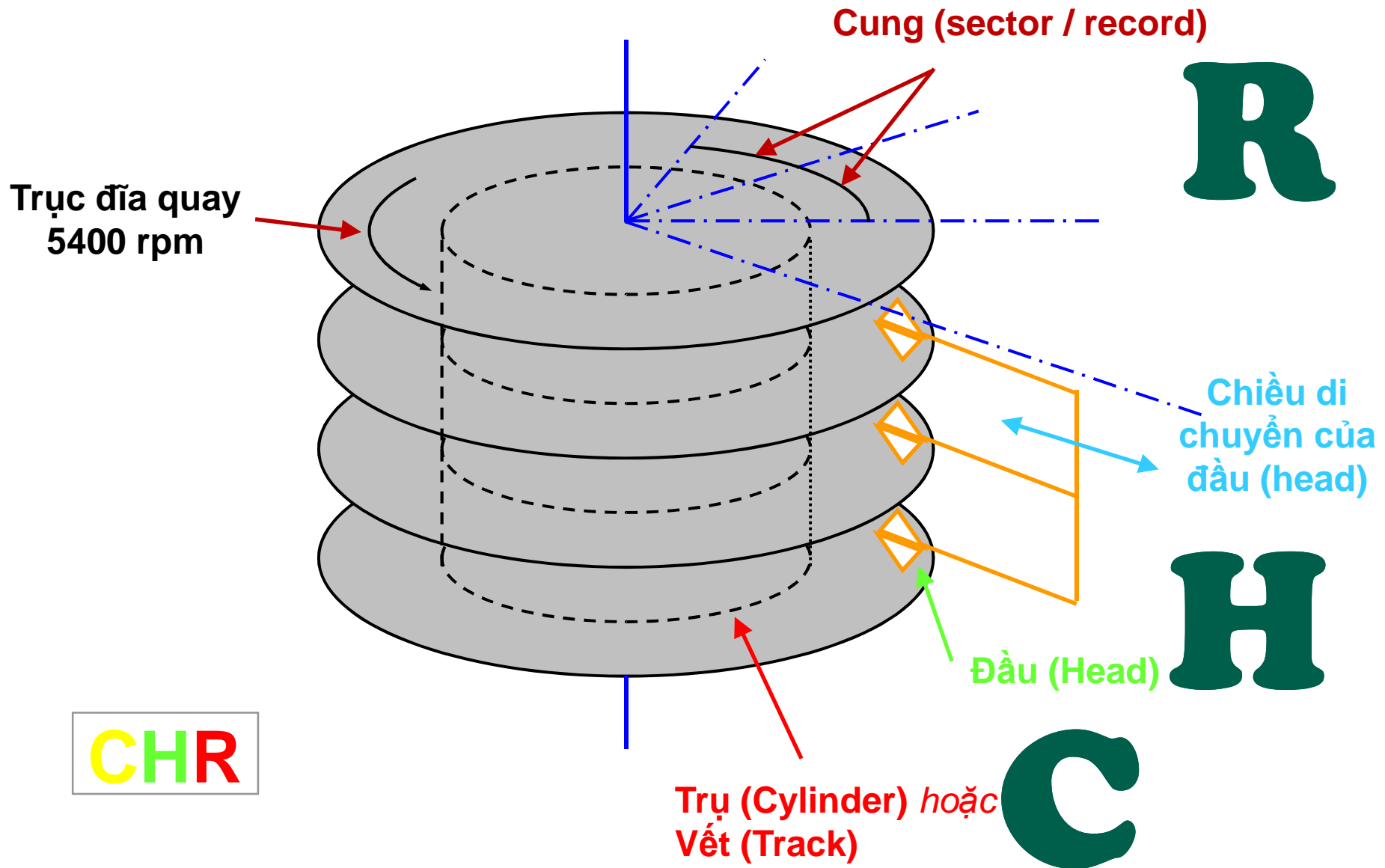
Hiện tượng rung phím  
(5 - 15 ms)

→ Chống rung

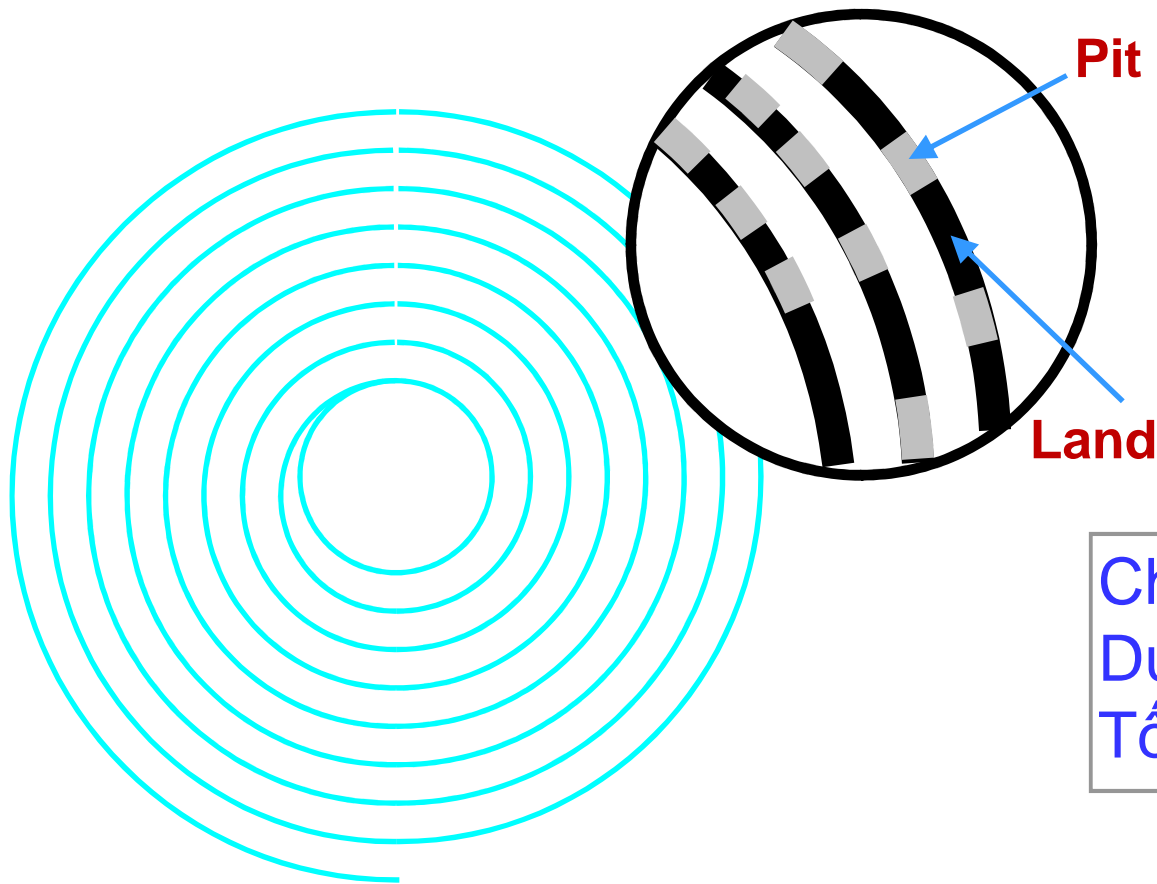
Cứng

Mềm

# Tổ chức thông tin trên đĩa cứng



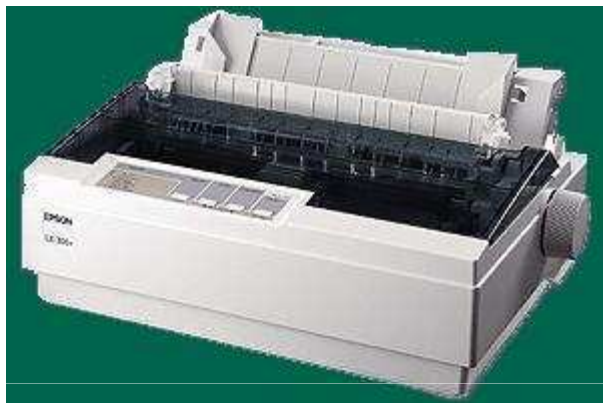
# CDROM



Chứa 330.000 khối dữ liệu.  
Dung lượng 650 MB / 74 min  
Tốc độ x1 = 153.60 KByte/s

Thông tin ghi theo rãnh (track) hình xoắn ốc.  
Dùng tia laser đục lỗ  $1 \mu\text{m}$  trên rãnh gọi là Pit.  
Phần không bị đục lỗ trên rãnh gọi là Land.

# Máy in



## Máy in kim

- + Máy rẻ tiền
- + Băng mực rẻ tiền
- + Lâu hết mực
- + In chậm



## Máy in phun

- + Máy rẻ tiền
- + Mực lỏng, đắt tiền
- + Mau hết mực
- + In chậm



## Máy in laser

- + Máy đắt tiền
- + Mực bột, đắt tiền
- + Lâu hết mực
- + In nhanh

# Ma trận điểm trên máy in kim

