

GIỚI THIỆU KHOA HỌC MÁY TÍNH



NGUYỄN THANH TRUNG



Giới thiệu về môn học

- Tên học phần: Giới thiệu về Khoa học Máy tính
- Mã số: CS100
- Số tín chỉ : 1 (lý thuyết)
- Thời gian: 18 giờ
- Điều kiện tiên quyết: Không



Mục tiêu

- Cung cấp kiến thức tổng quan về máy tính, sự phát triển của công nghệ chế tạo và các sản phẩm CNTT.
- Tổng quan về hệ thống máy tính, xử lý thông tin trên máy tính.
- Các khái niệm cơ bản về Hệ điều hành
- Tổng quan về mạng máy tính và Internet.
- Cơ bản về lập trình
- Các vấn đề cơ bản về công nghệ phần mềm
- Các hệ Quản trị Cơ sở dữ liệu, trí tuệ nhân tạo



Nhiệm vụ Sinh viên

- Tham gia đầy đủ các buổi học lý thuyết – chuyên cần
- Tham gia Thảo luận, trả lời câu hỏi tại lớp
- Làm bài tập nhóm - (Seminar tại lớp)
- Bài kiểm tra giữa kỳ
- Thi hết môn



Các hình thức kiểm tra đánh giá

- Chuyên cần : 5%
- Kiểm tra thường xuyên: 15%
- Thái độ: 10%
- Kiểm tra giữa kỳ: 15%
- Bài tập nhóm: 15%
- Thi hết môn: 40%
- Thang điểm 10/10 , sau đó quy sang điểm A,B,C,D



Tài liệu tham khảo

- Computer Science, 9th Edition, J. Glenn Brookshear (thư viện)
- Các tài liệu liên quan đến:
 1. Cấu trúc máy tính
 2. Hệ điều hành
 3. Mạng máy tính
 4. Thuật toán
 5. Công nghệ phần mềm
 6. Các hệ quản trị Cơ sở dữ liệu



Tài liệu cung cấp

- [2] Giáo trình Kiến trúc máy tính, Đại học Cần thơ, 2003 (ebook)
- [3] Bài giảng Hệ điều hành, Đại học Khoa học Tự nhiên (ebook)
- [4] Mạng Máy Tính, Phạm Thế Quế, 2006 (ebook)
- [5] Giáo trình cơ sở dữ liệu (ebook)
- [6] Kỹ thuật lập trình (ebook)
- [7] Công nghệ phần mềm (ebook)
- [8] Và một số tài liệu trên mạng tại các trang <http://ebook.edu.net.vn>, <http://quantrimang.com>, ...



NỘI DUNG

Chương 1 – GIỚI THIỆU TỔNG QUAN

- 1.1. Sơ lược về máy tính
- 1.2. Biểu diễn thông tin trong máy tính
- 1.3. Các cổng logic cơ bản



Chương 2 - HỆ THỐNG MÁY TÍNH

- 2.1. Hệ thống máy tính
- 2.2. Bộ xử lý trung tâm
- 2.3. Bộ nhớ máy tính
- 2.4. Thiết bị ngoại vi
- 2.5. Xử lý thông tin bên trong máy tính



Chương 3 – HỆ ĐIỀU HÀNH

- 3.1. Lịch sử các hệ điều hành
- 3.2. Kiến trúc hệ điều hành
- 3.3. Nguyên lý hoạt động của tiến trình
- 3.4. Bảo mật



Chương 4 - MẠNG MÁY TÍNH

- 4.1. Cơ bản về mạng máy tính
- 4.2. Internet
- 4.3. Các hình thức tấn công và các giải pháp bảo vệ mạng



Chương 5 - THUẬT TOÁN & CHƯƠNG TRÌNH

- 5.1. Cơ bản về thuật toán
- 5.2. Trình bày thuật toán
- 5.3. Đánh giá hiệu quả thuật toán
- 5.4. Chương trình & cài đặt thuật toán

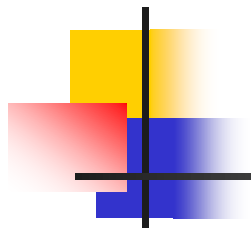


Chương 6 – CÔNG NGHỆ PHẦN MỀM

- 6.1. Giới thiệu sơ lược
- 6.2. Các mô hình
- 6.3. Các công cụ phần mềm
- 6.4. Các vấn đề khác

Chương 7 – CÁC LĨNH VỰC NGHIÊN CỨU KHÁC CỦA KHMT

- 7.1. Cơ bản về CSDL, hệ quản trị CSDL
- 7.2. Trí tuệ nhân tạo





CÁC CHỦ ĐỀ CHO BTNHÓM



CÁC NỘI DUNG SEMINAR



Yêu cầu

- 4-5 người/ nhóm
- Báo cáo: từ 5-8 trang A4
- Trình bày bằng slide: khoảng 15-20 slide
- Yêu cầu:
 1. Trình bày ngắn gọn, súc tích, ít chữ nhưng đầy đủ hình ảnh, hình vẽ minh họa...
 2. Ghi rõ nguồn tham khảo trong các trích dẫn
 3. Font size tối thiểu 26
 4. Bắt đầu trình bày từ tuần 2.



Tuần 2

ĐT1: Tìm hiểu về các máy tính nổi tiếng qua các giai đoạn phát triển

ĐT2: Tìm hiểu Các bảng mã ký tự



Tuần 3

ĐT 3. Các kỹ thuật làm tăng tốc bộ VXL

VD: đường ống (pipeline), Hyper threading (siêu phân luồng),

ĐT 4. Các VXL đa nhân,...

ĐT 5. Cấu tạo và tổ chức dữ liệu trên đĩa cứng

ĐT6. Cấu tạo và nguyên lý hoạt động máy in laser, màn hình



Tuần 4

ĐT7. Quản lý bộ nhớ (Hệ điều hành)

- Bộ nhớ phân đoạn, phân trang
- Bộ nhớ ảo.
- Cache...

ĐT8. Quản lý nhập xuất (HĐH)

- Cơ chế quản lý vào ra của HĐH



Tuần 4

ĐT9. Các kiểu tấn công trên mạng, tấn công các trang Web; 1 số giải pháp phòng chống

ĐT10. Tìm hiểu Internet (giao thức, quản lý), các ứng dụng trên Internet, khai thác Internet...



Tuần 5

ĐT11. Virus máy tính và các biến thể...

- (trojan, worm,....)

ĐT 12. Mạng máy tính (các vấn đề cơ bản: lịch sử, ứng dụng, các nguy cơ tiềm ẩn, các biện pháp phòng tránh)

ĐT13. Tìm hiểu chung về Công nghệ phần mềm, ngành CNPM ở VN và TG



Tuần 6

ĐT14. Giới thiệu chung về các hệ quản trị cơ sở dữ liệu

ĐT.15. ứng dụng CNTT vào đời sống

GIỚI THIỆU KHOA HỌC MÁY TÍNH



NGUYỄN THANH TRUNG

Chương 1 – GIỚI THIỆU TỔNG QUAN

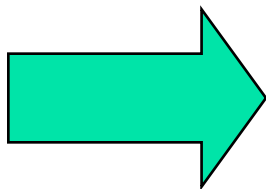
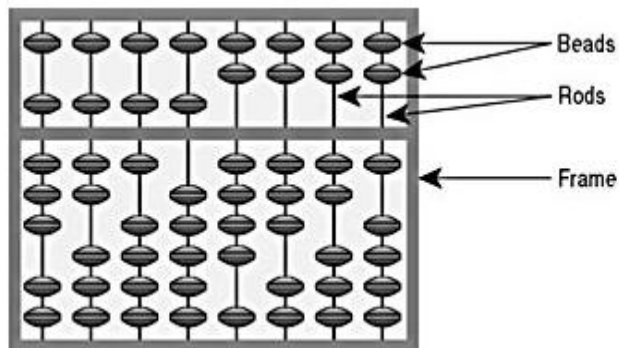
- 1.1. Sơ lược về máy tính và ngành KHMT
- 1.2. Biểu diễn thông tin trong máy tính
- 1.3. Các cổng logic cơ bản



1.1. Sơ lược về máy tính

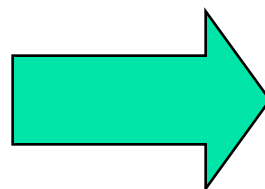
- Máy tính (Computer) là một thiết bị điện tử dùng để lưu trữ và xử lý thông tin theo các chương trình định trước.
- Máy tính, máy tính tương tự (Analog), máy tính số (Digital)... Sơ lược về lịch sử và phân loại máy tính

1.1.1. LỊCH SỬ PHÁT TRIỂN



Siêu máy tính BlueGene L của IBM (2005)

- 131702 processor
- 32768 GB main memory
- 280600-367000 GFlops



Các thế hệ máy tính ...

Blaise Pascal (Pháp-
1642)

Charles Babbage (Anh-
1830)

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

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

IBM 360 (1965)

Von Neumann (1945)

Cơ'

(1642 - 1945)

+, -, X, :

**Đèn
điện tử**

(1945 - 1955)

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

PDP-1 (1961)

Transistors

(1955 -
1965)

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

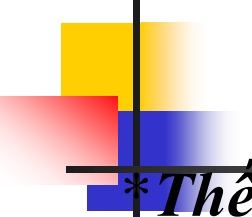
80x86 (1978)

IC

(1965 -
1980)

?

(1980 - ????)



**Thế hệ thứ nhất (1945-1955) máy tính dùng đèn điện tử:*

Trong những năm 40- 50 các thiết bị đầu tiên của máy tính điện tử được xây dựng và phát triển với.

+ **Phần cứng:** Chủ yếu là dùng đèn điện tử, độ tin cậy thấp, tốc độ chậm tiêu hao năng lượng rất lớn.

Ví dụ: Chiếc máy tính điện tử đầu tiên là chiếc ENIAC(Electronic Numerical Intergrator And Calculator) do John Mauchley và J.Presper Eckert thiết kế. Nó bao gồm 18.000 đèn điện tử, 1.500 role, nặng 30 tấn tiêu thụ 140 KW điện.

Dùng hệ thập phân

+ **Phần mềm:** Chủ yếu dùng ngôn ngữ máy và đặt công tắc bật tắt trực tiếp.

Ví dụ: Với chiếc ENIAC người ta phải đặt 6000 switch.

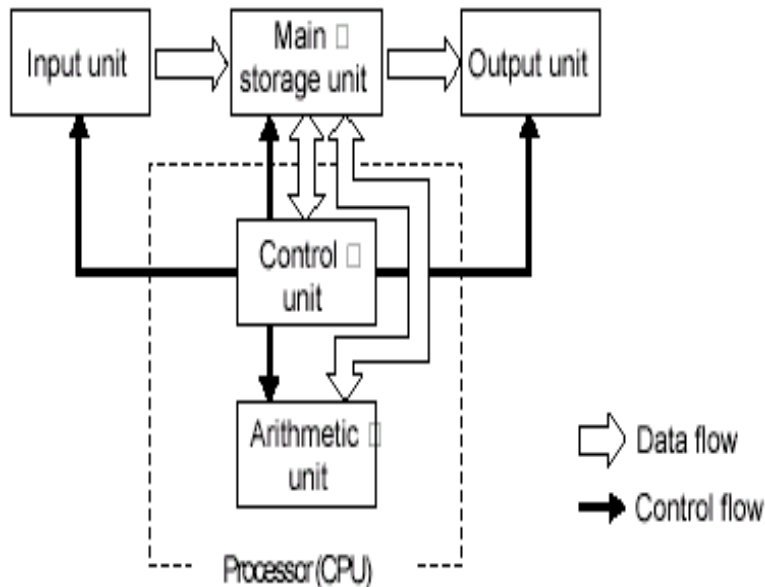
Máy tính ENIAC



Máy IAS (Institute of Advanced Study)

Do Von Neumann thiết kế, gồm các thành phần cơ bản sau (1947-1952)

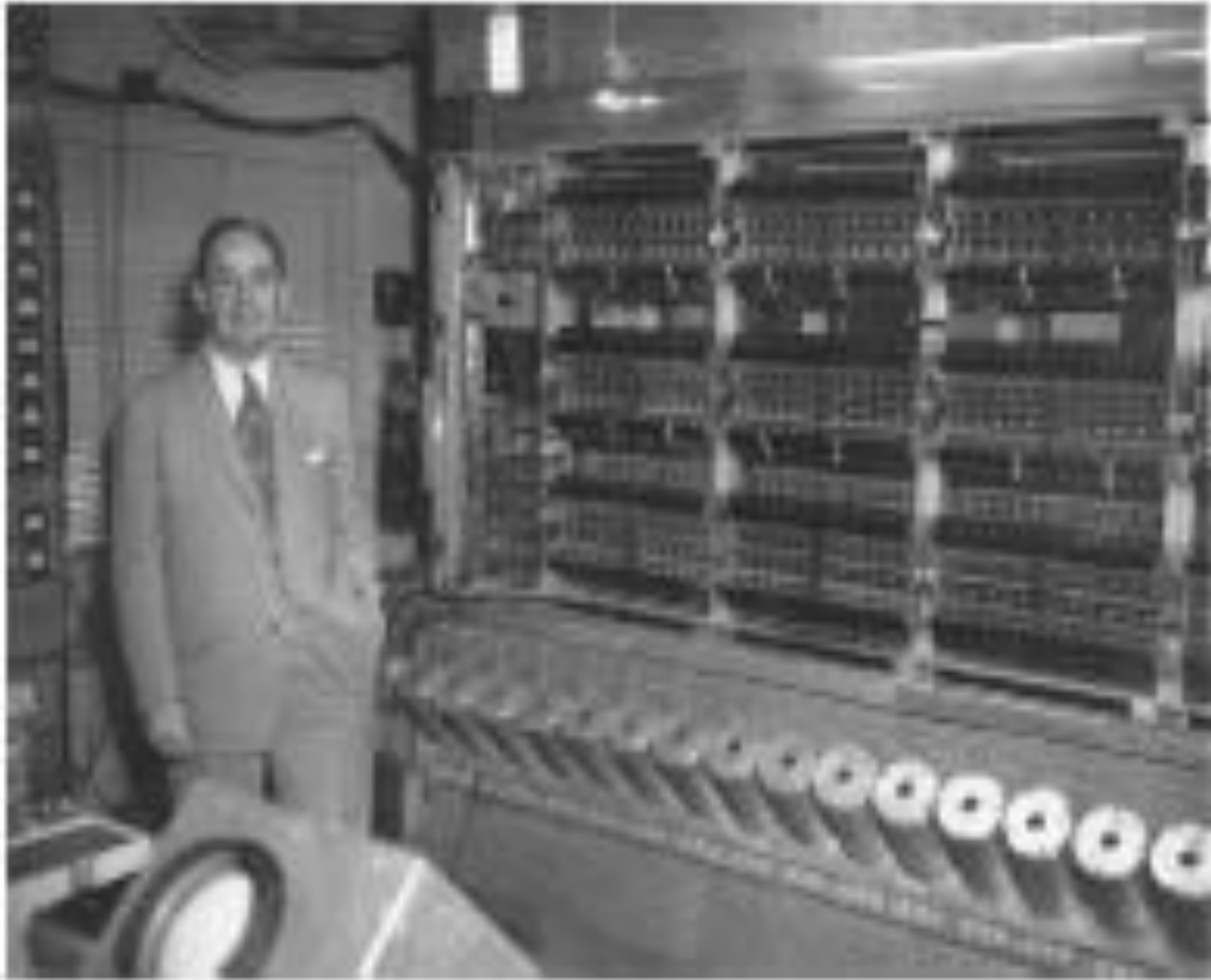
Máy tính hệ 2 đầu tiên

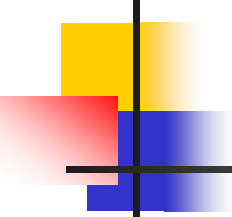


Đặc điểm chính của máy tính IAS

- Bao gồm các thành phần: đơn vị điều khiển, đơn vị số học và logic (ALU), bộ nhớ chính và các thiết bị vào ra.
- Bộ nhớ chính chứa chương trình và dữ liệu.
- Bộ nhớ chính được đánh địa chỉ theo từng ngăn nhớ, không phụ thuộc vào nội dung của nó.
- ALU thực hiện các phép toán với số nhị phân.
- Đơn vị điều khiển nhận lệnh từ bộ nhớ, giải mã và thực hiện lệnh một cách tuần tự.
- Đơn vị điều khiển điều khiển hoạt động của các thiết bị vào-ra.
- Trở thành mô hình cơ bản của máy tính.

John von Neumann và IAS





**Thế hệ thứ hai (1955-1965) máy tính đúng thiết bị bán dẫn:*

+ **Phần cứng:** Dùng linh kiện mới là Transistor (được phòng thí nghiệm Bell phát triển năm 1948). Bộ nhớ máy tính được tăng lên đáng kể và trở nên nhỏ gọn hơn. Chiếc máy đầu tiên của thế hệ này là chiếc TX-0.

+ **Phần mềm:** Đã bắt đầu sử dụng các ngôn ngữ lập trình bậc cao như Cobol, PL1,...

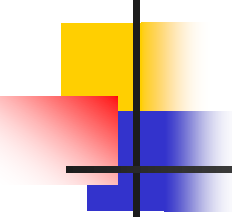
Các máy tính tiêu biểu

DEC's PDP-1 (1960)



IBM 7030 (1961)

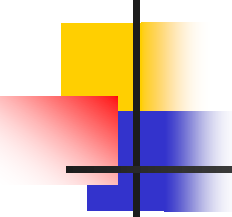




**Thế hệ thứ ba (1965-1980) dùng mạch hợp tích hợp IC:*

+**Phần cứng:** Công nghệ điện tử giờ đã phát triển rất nhanh cho phép đặt hàng chục Transitor vào một vỏ chung gọi là con chip. Linh kiện chủ yếu là các mạch tích hợp IC, đã bắt đầu xuất hiện đĩa từ để lưu trữ dữ liệu. Cho phép tốc độ tính toán đạt vài triệu phép tính/giây, có dung lượng bộ nhớ trong lên tới nhiều Mega bytes (MB).

+**Phần mềm:** Đã xuất hiện các hệ điều hành thế hệ đầu tiên. Các phần mềm ứng dụng ngày càng phát triển.



**Thế hệ thứ tư (1980-199x) sử dụng công nghệ (VLSI):*

+ **Phần cứng:** Vào những năm 80 công nghệ (VLSI - Very Large Scale Integrator) ra đời cho phép tích hợp trong một con chip hàng triệu Transitor khiến cho máy tính trở nên nhỏ hơn, nhanh hơn với tốc độ hàng triệu phép tính một giây là nền tảng cho chiếc máy tính PC (Personal Computer) ngày nay.

+ **Phần mềm:** Cùng với sự phát triển của máy tính các phần mềm ứng dụng đã phát triển như vũ bão làm cho tin học len lỏi vào mọi ứng dụng trong cuộc sống.



Thế hệ thứ 5: Người máy ?

- Hiện nay đang được nghiên cứu và phát triển dưới dạng các máy tính thông minh, robot,...
- Ví dụ: Deep Blue, Asimo,...



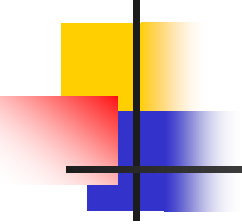
1.1.2. Các loại máy tính

- + **Máy tính cá nhân** (Personal Computer): Là máy tính để bàn, chỉ có một chip xử lý và thường dùng cho một người.
- + **Máy tính Mini** (Minicomputer): Thường được dùng trong các lĩnh vực ứng dụng thời gian thực và cho các ứng dụng vừa và nhỏ trong các dây chuyền sản xuất hay trong hàng không.
- + **Máy tính lớn** (Main Frame): Thường dùng trong các trung tâm tính toán đòi hỏi phải tốc độ xử lý tốt.
- + **Siêu máy tính** (Super Computer): Là một hệ thống gồm nhiều máy lớn ghép song song có tốc độ tính toán cực kỳ lớn và thường dùng trong các lĩnh vực đặc biệt, chủ yếu trong quân sự và vũ trụ. Siêu máy tính Deep Blue là một trong những chiếc thuộc loại này.



1.1.3. CÁC KHÁI NIỆM CƠ BẢN

- **Thông tin:** là một khái niệm trừu tượng mô tả tất cả những gì đem lại *hiểu biết, nhận thức* của con người và các sinh vật sống khác.
- **Dữ liệu**
 - Dữ liệu (dữ kiện) có thể hiểu nôm na là vật liệu thô mang thông tin.
 - Dữ liệu được tập hợp lại và được xử lí sẽ cho ta thông tin.
 - => Dữ liệu là nguồn gốc, là vật mang thông tin, là vật liệu sản xuất ra thông tin.

- 
- ❑ **Dữ liệu trong thực tế có thể là**
 - Tín hiệu vật lí (Physical signal)
 - Các số liệu (number)
 - Kí hiệu (symbol)

Thí dụ 1: Nhiệt độ cháu bé 39°C ...

Thí dụ 2: 28, 27, 30, 32, 27 ... ?

Thí dụ 3: Tính qui ước biểu diễn thông tin
I là chữ cái i hay là số I La mã ?



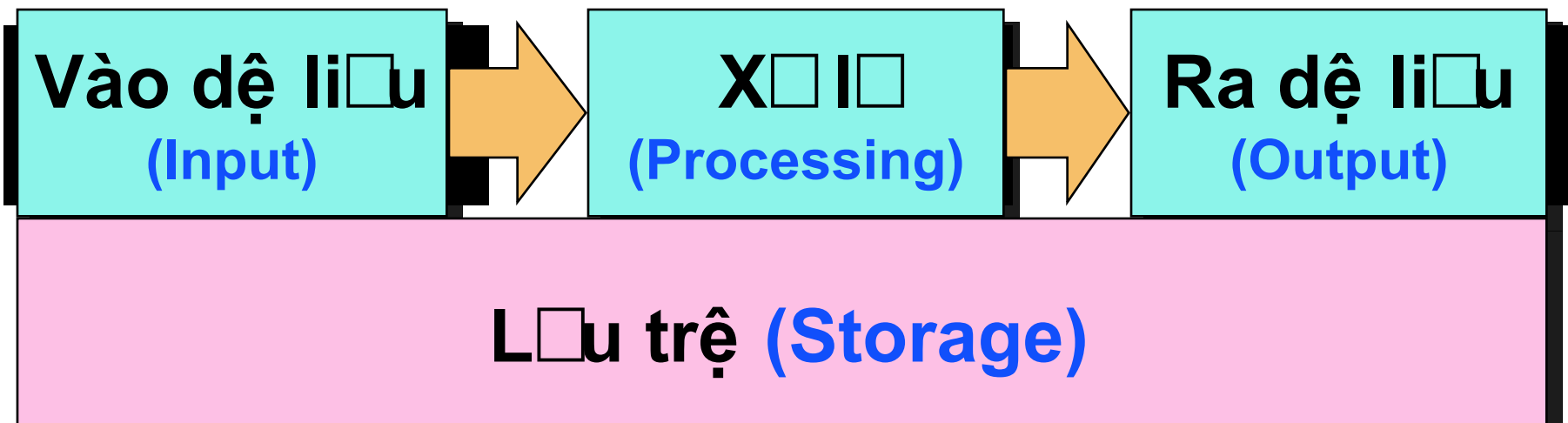
- **Xử lí thông tin**

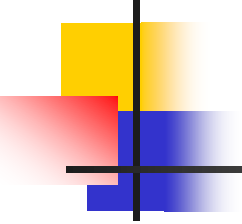
- lọc lấy thông tin cần thiết
- truyền tin: nhanh, chính xác ...
- lọc nhiễu
- lưu trữ
- tìm kiếm, lấy ra
- sao chép
- mã hoá bảo mật
- ...



- **Xử lý thông tin bằng máy tính**

- Khi thông tin ít, có thể làm thủ công.
- Khi thông tin nhiều lên, đòi hỏi máy móc tự động làm thay, đặc biệt là máy tính điện tử.
- Ưu điểm của máy tính: Làm nhanh, không biết chán, chính xác ...

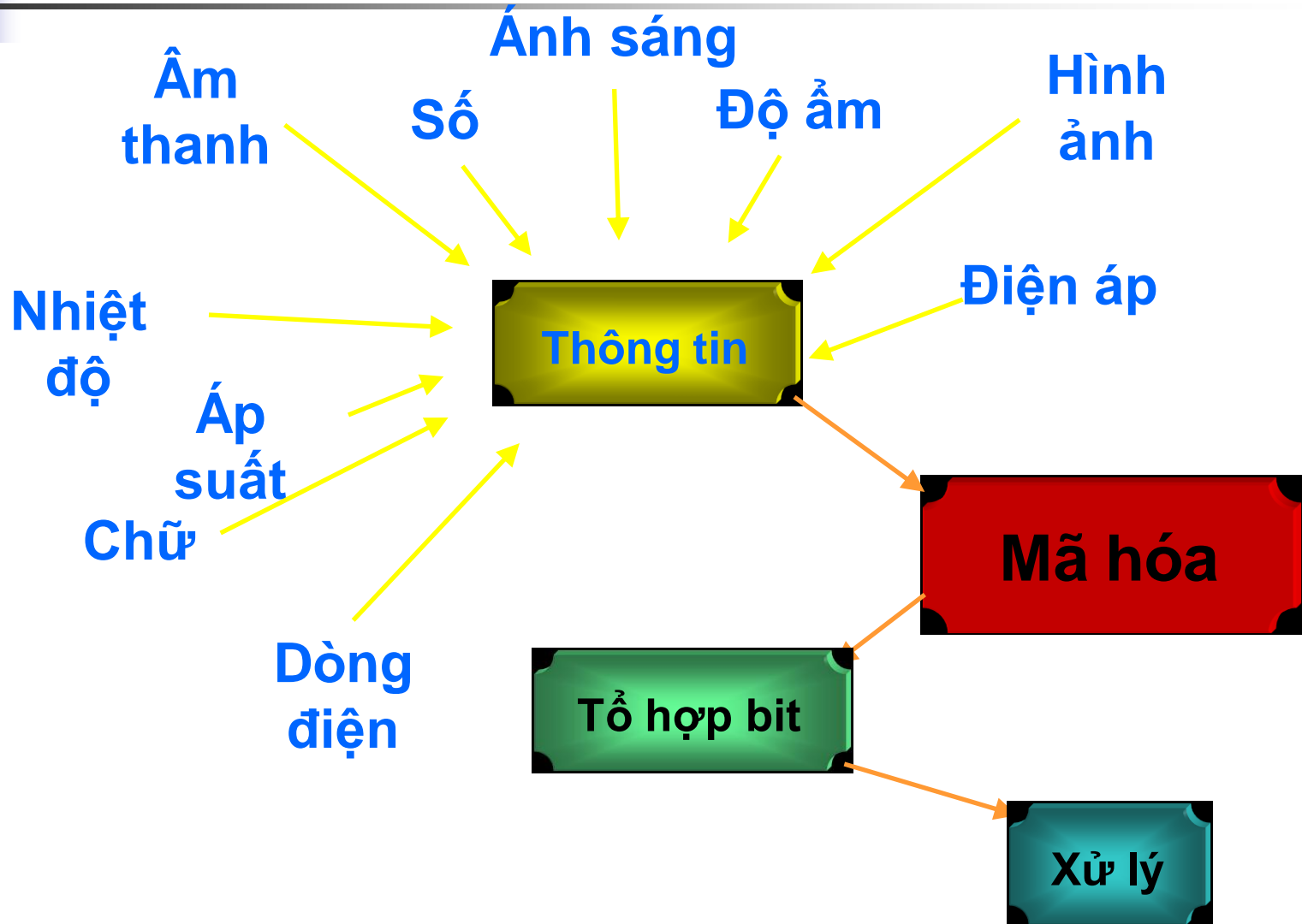


- 
-
- **Phần cứng (Hardware)**
 - **Phần mềm (Software)**
 - **Chương trình, lập trình**
 - **...**

1.2. Biểu diễn thông tin trong máy tính

- Làm thế nào để biểu diễn thông tin trong máy tính ?
- → Ta phải dùng **mã nhị phân** !
- Mã nhị phân là gì ?
- → 00111010 → bit (Binary Digit)
- Làm cách nào ?

Các dạng Thông tin





Mã hóa thông tin ?

- Thông tin để máy tính xử lí được thì cần phải biến đổi thành một dãy bit. Cách biến đổi như vậy gọi là mã hoá thông tin. **Biến đổi ngược lại ?**
- Người ta sử dụng mã ASCII để mã hoá kí tự. Bảng mã ASCII gồm 256 kí tự
- Ví dụ: Kí tự “A” có mã ASCII thập phân là 65 và mã ASCII nhị phân là 01000001.
- Để chuyển tín hiệu âm thanh → máy tính cần có bộ chuyển đổi: Card âm thanh ...
- ...



Đơn vị đo thông tin

- Đơn vị cơ bản để đo thông tin được gọi là bit.
- 1 bit có 2 trạng thái : 0/1
- Dãy số 0101... gọi là số nhị phân - số hệ 2
- Dãy 8 bit = 1 byte.
- Ngoài ra ta có các đơn vị lớn hơn để đo thông tin.

Các bội số của Byte

Kilobyte (KB) 1 nghìn Byte

Megabyte (MB) 1 triệu Byte

Gigabyte (GB) 1 tỉ Byte


Terabyte (TB) 1 Nghìn tỉ Byte

Petabyte (PB) 1 nghìn triệu triệu

Exabyte (EB) 10^{18} Byte

Zettabyte (ZB) 10^{21} Byte

Yottabyte (YB) 10^{42} Byte



Biểu diễn các dạng thông tin cơ bản trong máy tính

- Dạng Số
- Dạng văn bản: Dạng quen thuộc gồm các chữ cái, chữ số
- Dạng hình ảnh: Bức tranh, bản đồ, băng hình
- Dạng âm thanh: Tiếng nói con người, tiếng sóng, bản nhạc..
- Nhiệt độ...



1.2.1. Biểu diễn số

Hệ đếm: Tập các ký hiệu, quy tắc dùng để biểu diễn, tính toán các giá trị. Gồm:

- Hệ đếm không theo vị trí: Số La mã
- Hệ đếm theo vị trí: Hệ thập phân, nhị phân, Hexa, ...

Ví dụ: $2356 = 2 \times 10^3 + 3 \times 10^2 + 5 \times 10^1 + 6 \times 10^0$

→ Một số X bất kỳ có thể biểu diễn như sau:

$$X = x_{n-1}a^{n-1} + x_{n-2}a^{n-2} + x_{n-3}a^{n-3} + \dots + x_0a^0 + x_{-1}a^{-1} + x_{-2}a^{-2} + \dots \quad (1)$$

Trong đó : a gọi là cơ số



Một số hệ đếm theo vị trí thông dụng:

Hệ thập phân – Decimal (cơ số 10): Dùng tập số $\{0,1, \dots, 9\}$

Ví dụ: 123d; 25,4 ; 580, ...

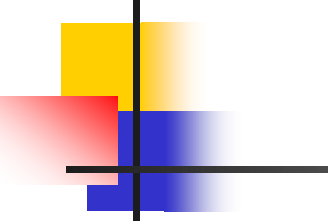
Nhi phân – Binary (cơ số 2): Dùng tập số $\{0,1\}$,

Ví dụ: 10001110b; 11000010b

Hệ bát phân (cơ số 8): Dùng tập số $\{0,1,\dots,7\}$

Thập lục phân - HexaDecimal (cơ số 16): Dùng tập số $\{0,1,\dots,9,A,B,C,D,E,F\}$, ...

Ví dụ: AC0Dh, 123h,..



| 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 |

Chuyển đổi giữa các hệ đếm

- Thập phân (nguyên) \rightarrow nhị phân: Chia liên tiếp cho 2, đến khi kết quả bằng 0. Đảo ngược trật tự các số dư \rightarrow số nhị phân tương ứng.

Vd: $12d = 1100b$

- Thập phân \rightarrow Hexa: tương tự ta chia liên tiếp cho 16

...

- Nhị phân \rightarrow Thập phân : Dùng công thức (1)

- Hexa \rightarrow thập phân : Dùng công thức (1).

Chuyển đổi giữa các hệ đếm (tt)

• Nhị phân \rightarrow Hexa: Nhóm 4 bit (Nibble) của số nhị phân tương ứng 1 số Hexa (Nhóm từ phải sang).

Ví dụ: $100\ 1100\ 0110\ 1010b = 4C6Ah$

• Hexa \rightarrow nhị phân: Cứ 1 số Hexa tương ứng 4 số nhị phân.

Ví dụ: $1ADFh = 0001\ 1010\ 1101\ 1111b$



Chuyển đổi giữa các hệ đếm (tt)

- Trường hợp thập phân không nguyên \rightarrow nhị phân thì :
- phần nguyên : Chia liên tiếp cho 2, đến khi kết quả bằng 0. Đảo ngược trật tự các số dư \rightarrow số nhị phân tương ứng.
- Phần lẻ: Nhân kết quả với 2 liên tục cho đến khi hết phần lẻ

$$\underline{\underline{\text{Vd: } 12,125_{10} = 1100,0010_2}}$$

vì: Phần nguyên đổi sang được 1100

$$\text{Phần lẻ : } 0,125 \times 2 = \underline{0},25 \times 2 = \underline{0},5 \times 2 = \underline{1},0 \rightarrow 001$$

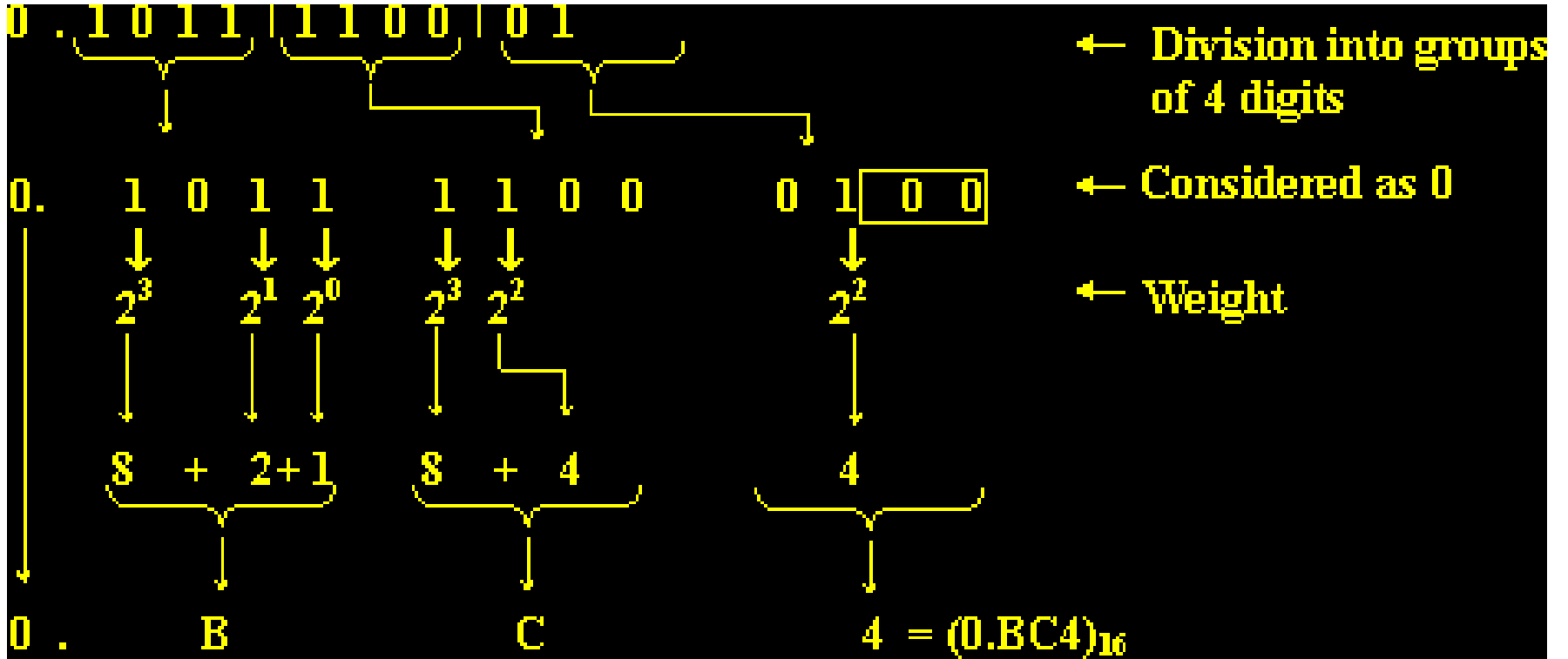


Chuyển đổi giữa các hệ đếm (tt)

Trường hợp nhị phân không nguyên \rightarrow thập phân thì áp dụng công thức (1)

$$\begin{aligned} \text{Ví dụ: } 1100,0010b &= 1x 2^3 + 1x 2^2 + 0x 2^1 + \\ &0x 2^0 + 0x 2^{-1} + 0x 2^{-2} + 1x 2^{-3} \\ &= 8 + 4 + 0 + 0 + 0 + 0 + 0,125 = 12,125d \end{aligned}$$

Trường hợp số nhị phân không nguyên \rightarrow Hexa ?



Biểu diễn số trong máy tính

Biểu diễn số nguyên

Gồm số nguyên không dấu và số nguyên có dấu.

Biểu diễn Số nguyên không dấu

Dùng để biểu diễn các đại lượng nguyên dương như địa chỉ ô nhớ, mã ASCII, ... tùy vào số byte biểu diễn mà phạm vi biểu diễn của số nguyên không dấu là khác nhau, cụ thể:

Số 8 bit (1 Byte): 256 số khác nhau từ 0, 1, 2, .. , 2^8-1

Số 16 bit (2 Byte): 65536 số khác nhau từ 0, 1, 2, .. , $2^{16}-1$



BIỂU DIỄN SỐ NGUYÊN CÓ DẤU

- Biểu diễn bằng giá trị tuyệt đối
- Biểu diễn bằng số bù 2



BIỂU DIỄN BẰNG GIÁ TRỊ TUYỆT ĐỐI

- Bit cao nhất được sử dụng làm bit dấu
 - Số dương biểu diễn bởi 0
 - Số âm biểu diễn bởi 1
- Các bit còn lại biểu diễn giá trị (tuyệt đối)
- Khoảng biểu diễn $-(2^{n-1}-1), 2^{n-1}-1$
- Hệ quả: có 2 số 0 là số 0 dương (+0) và số không âm (-0)



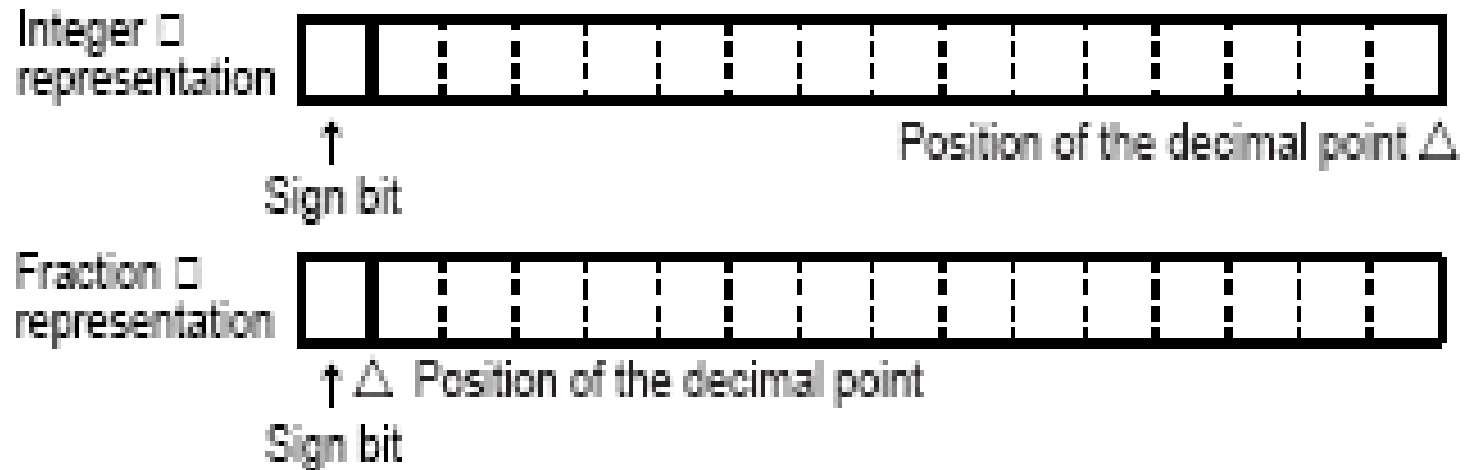
BIỂU DIỄN BẰNG SỐ BÙ 2

- Số bù 1 của một số: là số thu được khi lấy số bao gồm toàn bit 1 trừ đi số đó. (Thực chất chính là đảo từng bit của số đó)
- Số bù 2 của một số: là số bù 1 của số đó cộng 1
- Số bù 2 của một số: được sử dụng để biểu diễn giá trị đảo của số đó
- Khoảng biểu diễn $(-2^{n-1}, 2^{n-1} - 1)$

Biểu diễn số trong máy tính (tt)

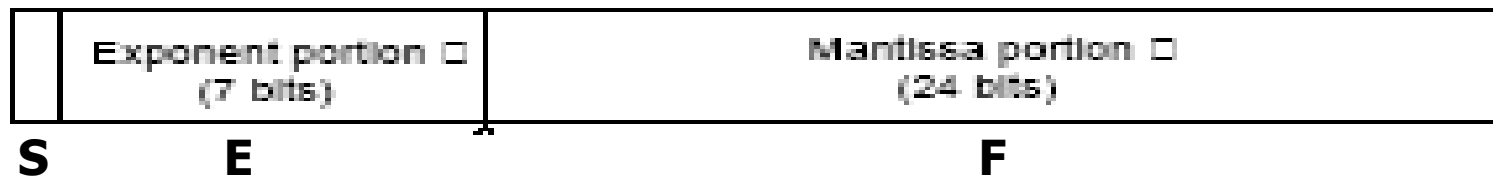
Biểu diễn Số thực: Bảng dấu chấm tĩnh và dấu chấm động

Bảng dấu chấm tĩnh



Bảng dấu chấm động (chuẩn IEEE/754)

ĐỘ chính xác đơn (32 bit)



- S - phần dấu (1 bit) là dấu của phần định trị (0 là dương; 1 là âm)
- E - phần mũ (8 bit) là số mũ với cơ số 2
- F - phần định trị (23 bit) là phần phân
- Giá trị của số được biểu diễn $(-1)^s \times 2^{E-127} \times (1.F)$

Bảng dấu chấm động (chuẩn IEEE/754)

ĐỘ chính xác kép (64 bit)



- S - phần dấu (1 bit) là dấu của phần định trị (0 là dương; 1 là âm)
- E - phần mũ (11 bit) là số mũ với cơ số 2
- F - phần định trị (52 bit) là phần phân số
- Giá trị của số được biểu diễn $(-1)^s \times 2^E - 1023 \times (1.F)$



Biểu diễn ký tự

Các loại mã khác biểu diễn ký tự

- ASCII (American Standard Code for Information Interchange)
- ISO (International Organization for Standardization)
- JIS (Japanese Industrial Standards)
- EBCDIC (Extended Binary Coded Decimal Interchange Code)
- Unicode



ASCII

Bảng mã ASCII (American Standard Code for Information Interchange)

Dùng 1 byte để mã hoá nên có 256 ô kí tự.

- 26 chữ cái latin ‘a’..’z’, 26 chữ cái hoa
- 10 chữ số thập phân ‘0’..’9’
- Các dấu chấm câu ...
- Các kí tự điều khiển
- 128 kí tự đầu tiên là chuẩn quốc tế
- 128 kí tự sau (128..255) thay đổi theo quốc gia

■ Mã từ 0 .. 127 → chuẩn, người dùng không thể thay đổi.

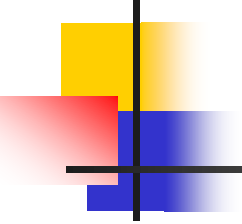
✓ Văn bản dùng mã này gọi là văn bản thô (plain text) khác với văn bản Rich text với nhiều kiểu định dạng hơn.

✓ 32 ký tự đầu là mã điều khiển, không nhìn thấy được (32 ký tự này tương ứng ký tự A..Z – 64 → dùng phím Ctrl trên bàn phím).

✓ Ngoài ra 1 mã ASCII bất kỳ có thể được nhận qua bàn phím bằng cách nhấn ALT + mã thập phân của ký tự đó.

✓ Mã ASCII mở rộng tính từ 128 ..255 dùng cho các ký hiệu khác và người dùng có thể thay đổi.

✓ Với 256 ký tự có thể biểu diễn các từ trong tiếng Anh, Đức, ... Và tạm đủ cho tiếng Việt nhưng không thể đủ cho các chữ tượng hình trong tiếng Trung Quốc, Nhật.



- **ISO (International Organization for Standardization):**

Các mã ISO 10646 được đưa ra vào năm 1983 với 2 Byte, nhằm đáp ứng cho các chữ tượng hình nói trên nhưng cũng không được thống nhất.

- **JIS (Japanese Industrial Standards)**

- **Unicode: Do hãng Xerox đưa ra dùng 2 Byte để mã hoá 1 ký tự, hiện nay mã này đang được khuyến khích dùng như bản mã chuẩn. Là mã 16 bit nên khi dùng trong môi trường 8 bit (truyền dữ liệu hoặc lập trình API) → chuỗi byte theo quy định UTF-8 (Unicode Transformation Format - 8).**



Biểu diễn âm thanh: Audio

Người ta mã hoá âm thanh theo hệ nhị phân bằng cách ngắt âm thanh ra thành nhiều giá trị có khoảng thời gian nhất định. Sau đó máy tính gán cho mỗi giá trị tương ứng 1 mã nhị phân và lưu tuần tự trong tập tin.

Với thính giác con người chỉ phân biệt được tần số âm thanh tối đa là 20 KHz nên khi lấy mẫu người ta ngắt âm thanh theo tần số lớn hơn 2 lần tần số tối đa đó. Chuẩn tần số lấy mẫu hiện nay là 44KHz.

Để biểu diễn 1 giá trị âm thanh có thể dùng 8 bit (256 giá trị khác nhau). Hiện nay người ta dùng 16 bit.

Các định dạng file âm thanh như Wave, Midi...



Biểu diễn hình ảnh:

Để biểu diễn ảnh đồ họa người ta có thể dùng 2 phương pháp là đồ họa điểm và đồ họa vectơ.

Đồ họa điểm: Đồ họa điểm biểu diễn ảnh bằng ma trận điểm ảnh. Có thể dùng 1/2/3/4 Byte để biểu diễn 1 điểm ảnh. Người ta ghi thông tin của điểm ảnh lên tập tin dưới các định dạng BMP, GIF, JPG,...

Ảnh biểu diễn theo cách này tốn rất nhiều bộ nhớ.



Biểu diễn hình ảnh (tt)

Đồ hoạ vectơ: Chia hình ảnh ra thành nhiều đối tượng như Điểm, đường thẳng, đa giác, mặt, khối. Máy tính dùng công thức tính toán vectơ để dựng nên các hình ảnh từ các đối tượng cơ bản trên.

Có thể xử lý các đối tượng và đặc tính đối tượng bằng mã lệnh nên lưu trữ ít tốn bộ nhớ.

Đồ hoạ vectơ có thể hiển thị được hình ảnh 3-D.



Biểu diễn hình ảnh động

- Để biểu diễn ảnh động theo điểm ảnh người ta có thể cho hiển thị 30 ảnh / giây để tạo nên ảnh động → 1 đoạn phim 1 giây phải lưu đến 30 ảnh mới tạo nên cảm giác liên tục → tốn bộ nhớ → Nén dữ liệu.
- Các định dạng: MPEG, MOV, ... dùng để nén từng ảnh tĩnh và nén những phần ảnh không đổi theo thời gian.
- Đối với việc biểu diễn ảnh động theo vectơ, người ta thêm 1 hàm để mô tả vị trí của đối tượng trong không gian theo thời gian. → Cần máy có công suất lớn.



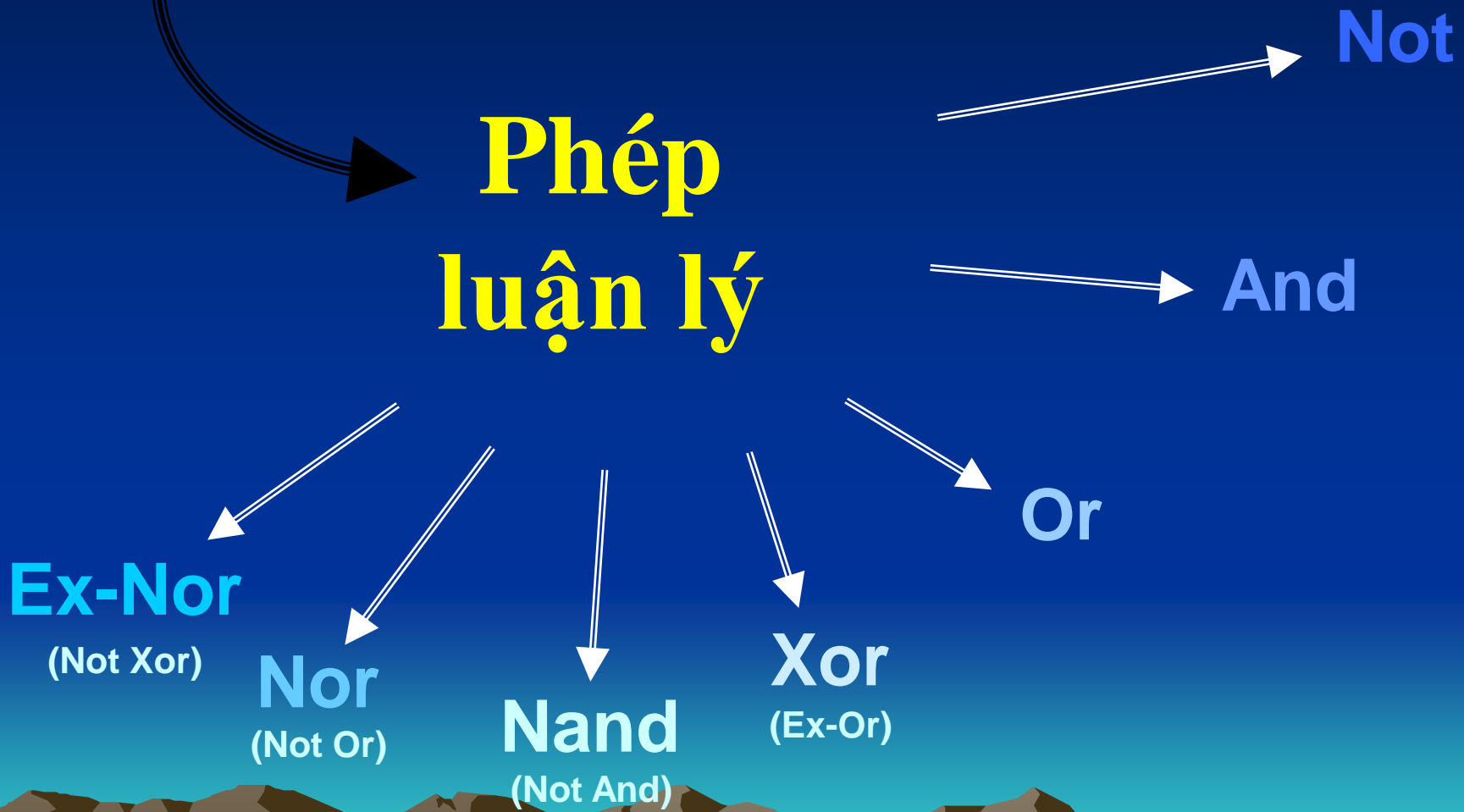
Biểu diễn các thông tin khác

- Người ta cũng có thể số hoá các đại lượng dạng tương tự trong tự nhiên thành rời rạc trong máy tính bằng các bộ cảm biến để chuyển các tín hiệu vật lý → điện.
- Các thiết bị chuyển đổi như vậy gọi là ADC (Analog to Digital Convert) hay DAC (Digital to Analog Convert).

1.3. Các mạch cơ bản

- 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 giây dẫn tín hiệu.

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

Ký hiệu dấu cộng
như phép cộng

Bảng sự
thật

| x | y | x + y |
|----------|----------|--------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |



Nhận xét

$$y + 0 = y$$

$$y + 1 = 1$$

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

NOT

AND

OR

XOR

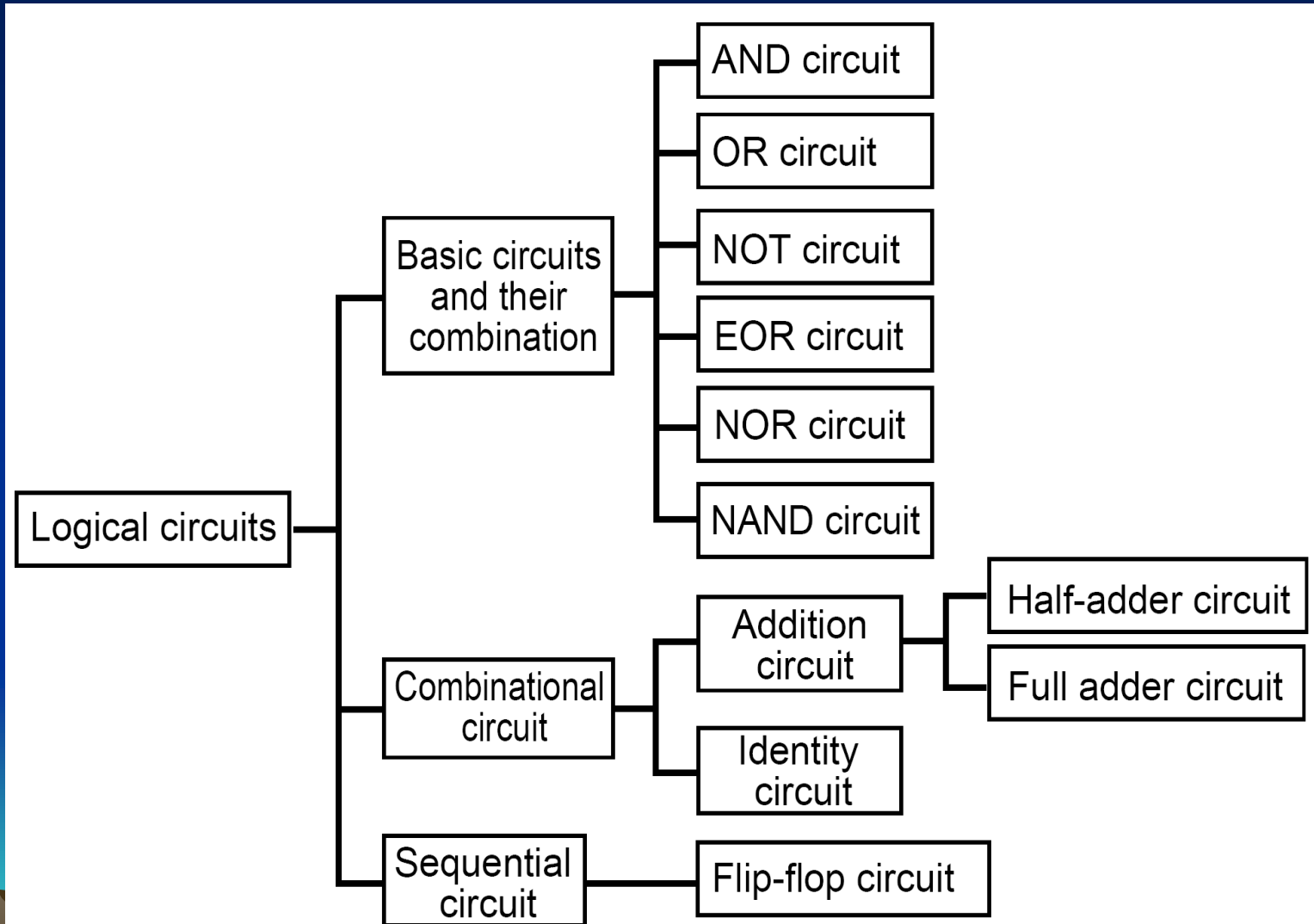
| thật x | y | not y | x and y | x or y | x xor y |
|-----------|---|-------|---------|--------|---------|
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 |

$y \text{ and } 0 = 0$
 $y \text{ and } 1 = y$

$y \text{ or } 0 = y$
 $y \text{ or } 1 = 1$

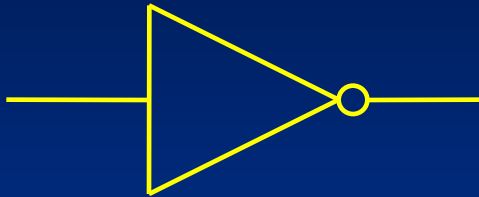
$y \text{ xor } 0 = y$
 $y \text{ xor } 1 = \text{not } y$

Các mạch cơ bản dùng trong máy tính

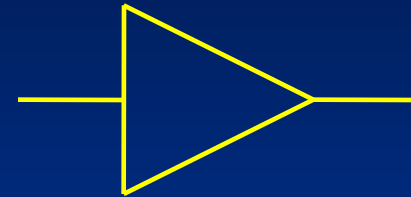


Cổng luận lý

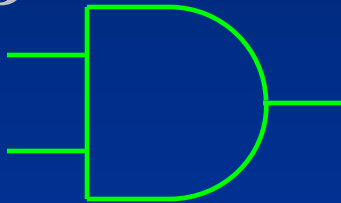
NOT



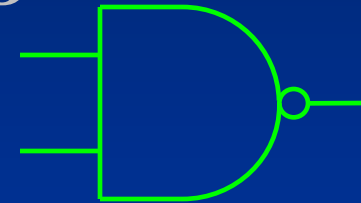
BUFFER



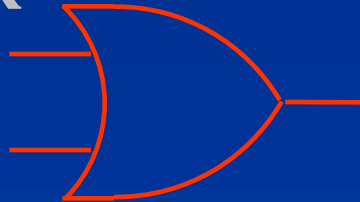
AND



NAND



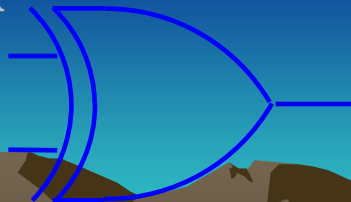
OR



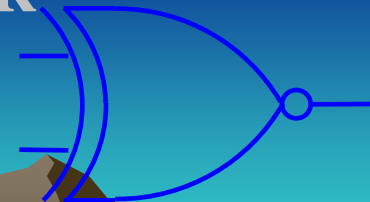
NOR



XOR

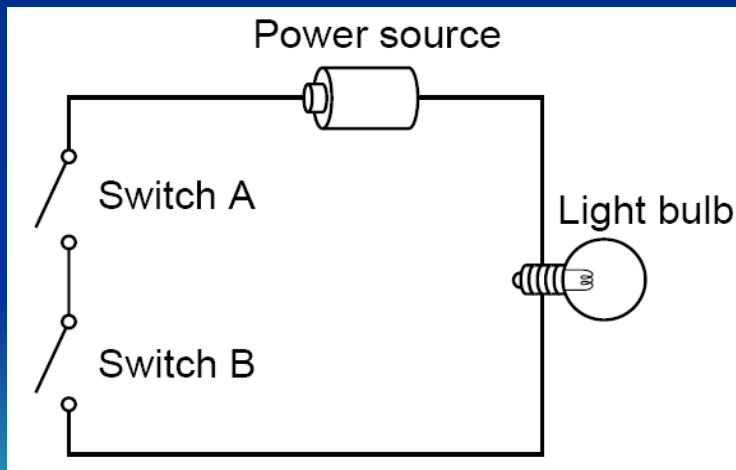
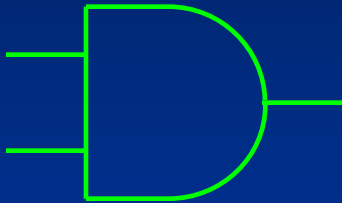


EX-NOR



Chức năng đóng mở

mức luận lý 1 = 5V
mức luận lý 0 = 0V



y and 1 = y

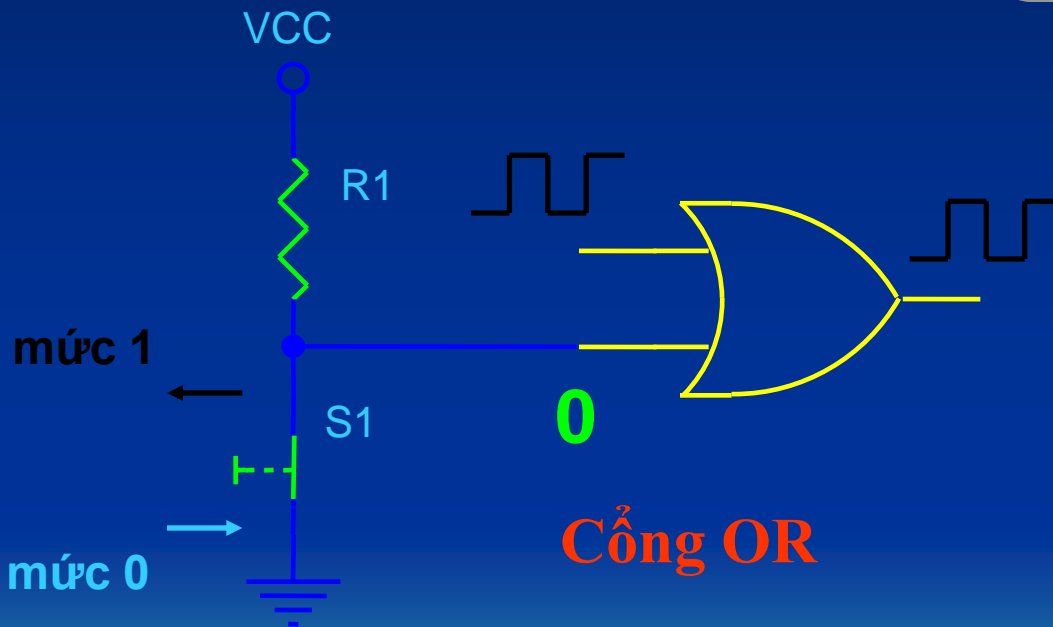
1 = mở

y and 0 = 0

0 = đóng

Chức năng đóng mở (tt.)

mức luận lý 1 = 5V
mức luận lý 0 = 0V



y or 1 = 1

1 = đóng

y or 0 = y

0 = mở

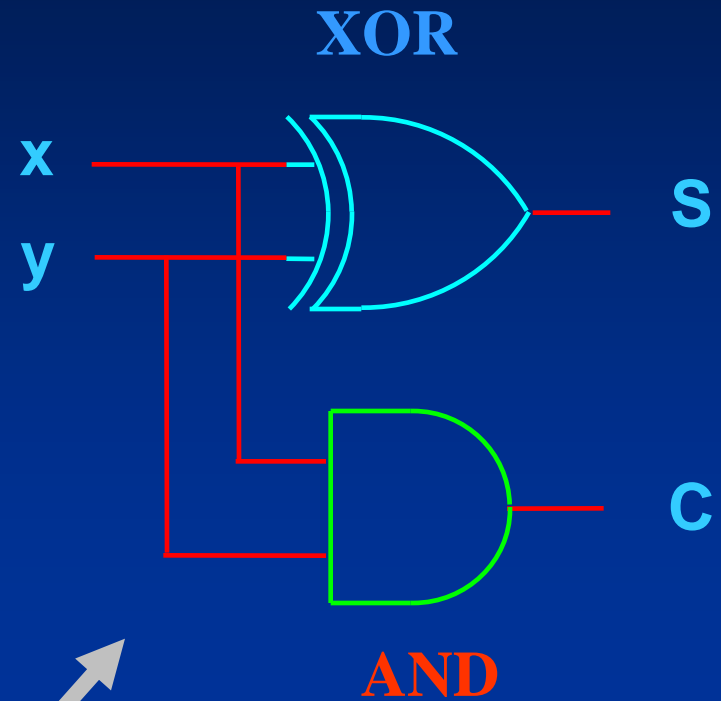
Các mạch tích hợp

- 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 .
- Các mạch dãy (Flip/flop)...

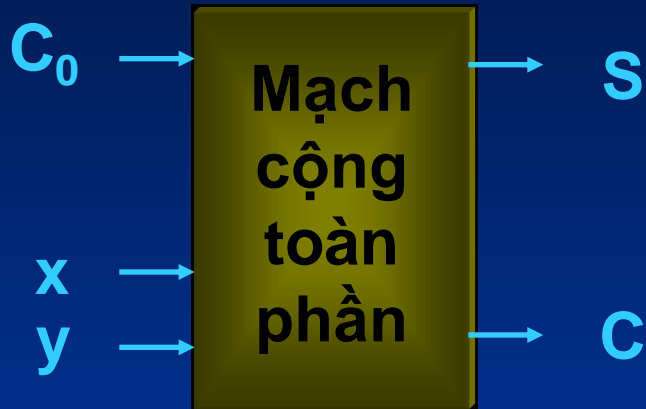
Mạch cộng bán phần



| x | y | S | C |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |



Mạch cộng toàn phần



$$S = x + y + C_0$$

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

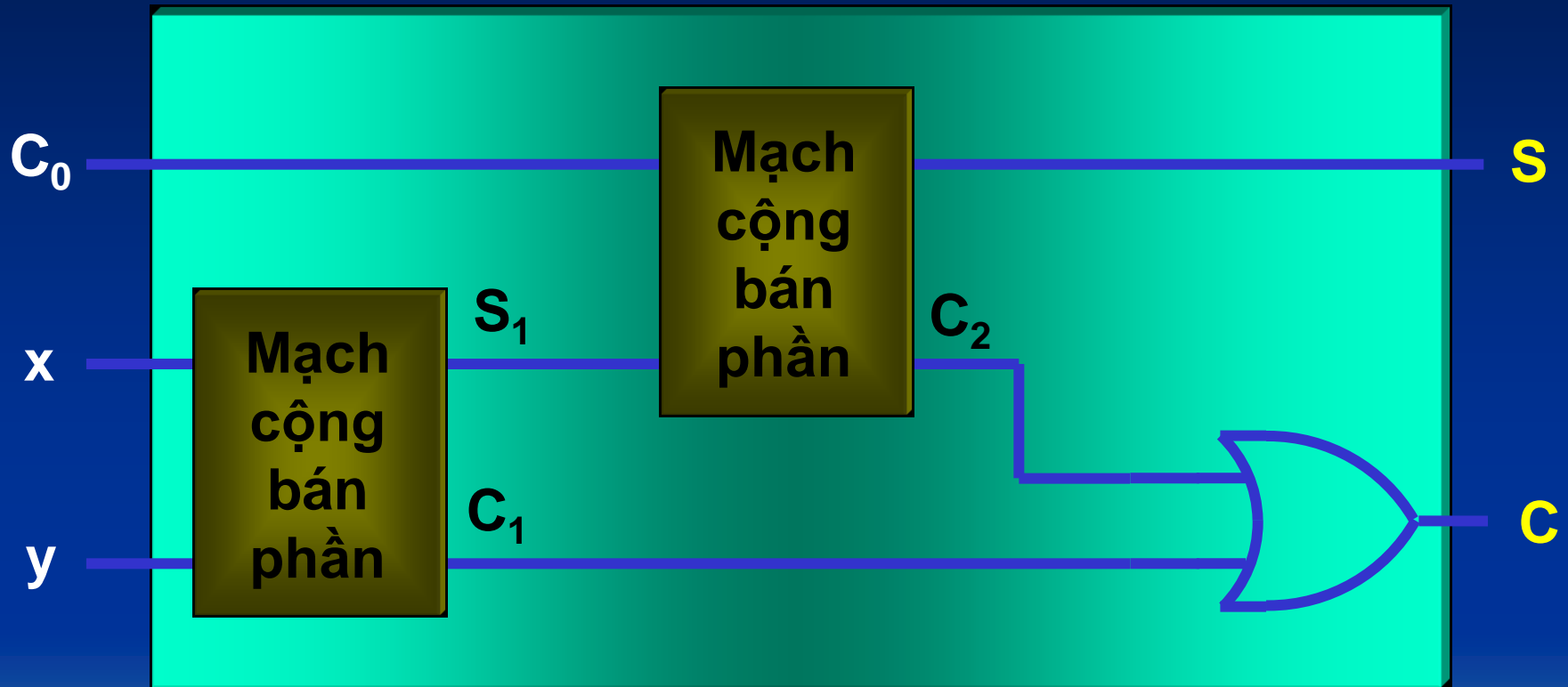
$$\text{Tính: } S_1 = x + y$$

$$\text{Tính: } S_2 = S_1 + C_0$$

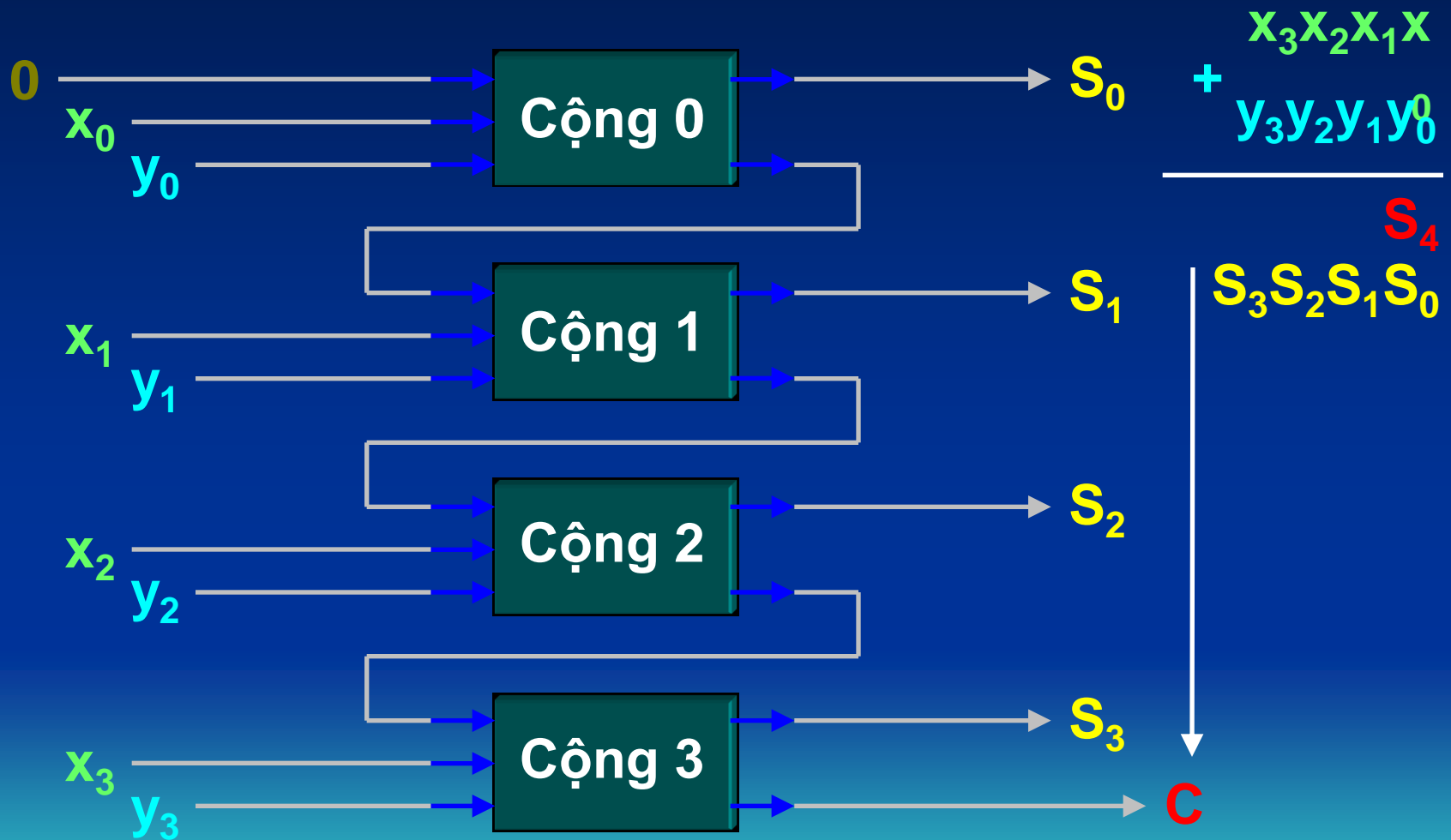
Cần bộ cộng bán phần 1

Cần bộ cộng bán phần 2

Mạch cộng toàn phần (tt.)



Mạch cộng nhiều bit



1.4. Khoa học máy tính

- *Computer science* hay *computing science*
- Là ngành nghiên cứu các cơ sở lý thuyết về thông tin và tính toán cùng với việc thực hiện và ứng dụng của chúng trong các hệ thống máy tính.

Các lĩnh vực của khoa học máy tính

- Cơ sở toán học
- Lý thuyết tính toán
- Cấu trúc dữ liệu và giải thuật
- Ngôn ngữ lập trình và trình biên dịch
- Hệ thống phân tán, song song, tương tranh
- Kỹ nghệ phần mềm
- Kiến trúc máy tính
- Truyền thông
- Cơ sở dữ liệu
- Trí tuệ nhân tạo
- ...

Cơ sở toán học

- Lôgic toán (*Mathematical logic*)
 - Lôgic Bool và các phương pháp tương ứng dùng để mô hình hóa các truy vấn lôgic; Sự sử dụng các phương pháp chứng minh hình thức (*formal proof*).
- Lý thuyết số (*Number theory*)
 - Lý thuyết về chứng minh và các khảo nghiệm trong việc tìm những chứng minh trong giới hạn các số nguyên. Lý thuyết số được sử dụng trong mật mã học và đồng thời được dùng như một phương thức kiểm thử trong trí tuệ nhân tạo.
- Lý thuyết đồ thị (*Graph theory*)
 - Cơ sở cho cấu trúc dữ liệu và các thuật toán tìm kiếm

Lý thuyết tính toán

- Lý thuyết Ôtômat (*Automata theory*)
 - Các cấu trúc logic khác nhau có thể sử dụng để giải quyết các bài toán.
- Lý thuyết khả năng tính toán (*Computability theory*)
 - Những gì có thể tính toán được bằng các mô hình máy tính hiện tại. Các chứng minh của Alan Turing và những người khác trình bày cho chúng ta biết được khả năng những gì có thể tính toán được và những gì không thể.
- Lý thuyết độ phức tạp tính toán (*Computational complexity theory*)

Cấu trúc dữ liệu và giải thuật

- Phân tích thuật toán (*Analysis of algorithms*)
 - Độ phức tạp về thời gian và không gian của các thuật toán.
- Thuật toán (*Algorithms*)
 - Các quá trình logic trên nguyên tắc được sử dụng cho việc tính toán và tính hiệu quả của các quá trình này.
- Cấu trúc dữ liệu (*Data structures*)
 - Tổ chức của dữ liệu và các quy tắc thao tác dữ liệu.

Ngôn ngữ lập trình và trình biên dịch

- Trình biên dịch (*Compilers*)
 - Những phương thức khác nhau dùng để dịch các chương trình máy tính, thường là từ các ngôn ngữ lập trình bậc cao sang các ngôn ngữ lập trình bậc thấp.
- Trình thông dịch (*Interpreter*)
 - Một chương trình sử dụng để biên dịch và thi hành trực tiếp một chương trình phần mềm khác dùng trong máy tính mà không phải thông qua quá trình biên dịch.
- Ngôn ngữ lập trình (*Programming languages*)
 - Các khuôn mẫu ngôn ngữ dùng để biểu diễn các thuật toán. ...

Hệ thống phân tán, song song, tương tranh

- Tương tranh (*Concurrency*)
 - Lý thuyết và thực tiễn của tính toán đồng thời; an toàn dữ liệu trong môi trường đa nhiệm hay đa luồng bất kỳ.
- Tính toán phân tán (*Distributed computing*)
 - Tính toán sử dụng nhiều thiết bị tính toán trên một mạng để thực hiện một nhiệm vụ hoặc một mục tiêu chung.
- Tính toán song song (*Parallel computing*)
 - Tính toán sử dụng nhiều luồng thực thi đồng thời.

Kỹ nghệ phần mềm

- Thiết kế thuật toán (*Algorithm design*)
- Lập trình máy tính (*Computer programming*)
- Các phương pháp hình thức (*Formal methods*)
 - Sử dụng toán học để miêu tả và lập luận đối với các thiết kế phần mềm.
- Kỹ nghệ phần mềm (*Software development*)
 - Những nguyên lý và thực hành trong việc thiết kế, phát triển và kiểm thử các chương trình, cùng những phương pháp thực hành kỹ nghệ đúng đắn.

Kiến trúc máy tính

- Kiến trúc máy tính (*Computer architecture*)
 - Việc thiết kế, tổ chức, tối ưu hóa và kiểm định một hệ thống máy tính, chủ yếu về CPU và tiểu hệ bộ nhớ máy tính (và hệ thống bus nối giữa chúng).
- Tổ chức máy tính (*Computer organization*)
 - Nghiên cứu các kiến trúc máy tính trên cơ sở các mô tả mạch điện, bộ xử lý trung tâm, bộ xử lý tín hiệu số của máy tính.
- Hệ điều hành
 - Những hệ thống dùng để quản lý các tài nguyên máy tính và cung cấp nền tảng cơ bản cho một hệ thống khả dụng. ..

Truyền thông

- Xử lý âm thanh trong máy tính (*Computer audio*)
 - Những thuật toán và cấu trúc dữ liệu dùng để kiến tạo, thao tác, lưu trữ, và truyền thanh các bản ghi âm thanh kỹ thuật số (*digital audio*), nhận dạng giọng nói (*voice recognition*).
- Mạng máy tính (*Computer networking*)
 - Các giao thức dành cho việc truyền thông dữ liệu một cách đáng tin cậy qua các môi trường truyền thông chuyên dụng hoặc chia sẻ khác nhau. Thường khi bao gồm cả việc sửa lỗi (*error correction*) trong truyền thông.
- Mật mã học (*Cryptography*)
 - Áp dụng kết quả của các lý thuyết độ phức tạp tính toán, lý thuyết xác suất, và lý thuyết số để kiến tạo và phá mật

Cơ sở dữ liệu

- Khai phá dữ liệu (*Data mining*)
 - Nghiên cứu các phương pháp sàng lọc, rút ra những thông tin cần thiết từ các nguồn dữ liệu khác nhau.
- Cơ sở dữ liệu quan hệ (*Relational databases*)
 - Nghiên cứu các thuật toán tìm kiếm và xử lý thông tin trong các tài liệu và cơ sở dữ liệu; có quan hệ gần gũi với ngành thu thập thông tin (*information retrieval*).
- CSDL hướng đối tượng,...

Trí tuệ nhân tạo

- Trí tuệ nhân tạo (*Artificial intelligence*)
 - Sự nghiên cứu và thực thi các hệ thống có khả năng tự thể hiện trí thông minh hoặc tự biểu đạt những hành vi của chính bản thân mình.
- Lập luận tự động (*Automated reasoning*)
 - Nghiên cứu các động cơ giải quyết bài toán, chẳng hạn như được sử dụng trong Prolog, các động cơ này tạo ra các bước dẫn đến một kết quả nếu cho trước một truy vấn về một sự kiện và một cơ sở dữ liệu gồm các luật (*rule database*).
- Máy học (*Machine learning*)
 - Nghiên cứu việc tự động tạo nhóm các luật và tiên đề dựa trên những dữ liệu cho trước.
- Xử lý ngôn ngữ tự nhiên
 - Tự động hóa việc tiếp thu và kiến tạo ngôn ngữ loài người.
- Rô-bô học (*Robotics*): Các thuật toán điều khiển hành vi của rô-bô.

Bài tập

- Tìm hiểu lỗi truyền thông và các cơ chế kiểm tra, sửa lỗi:
 - Parity bit
 - Mã sửa lỗi
- Nộp báo cáo bằng bản viết tay khoảng 1→2 trang A4 (tính vào điểm BTCN)
- Tuần sau nộp

GIỚI THIỆU KHOA HỌC MÁY TÍNH



NGUYỄN THANH TRUNG

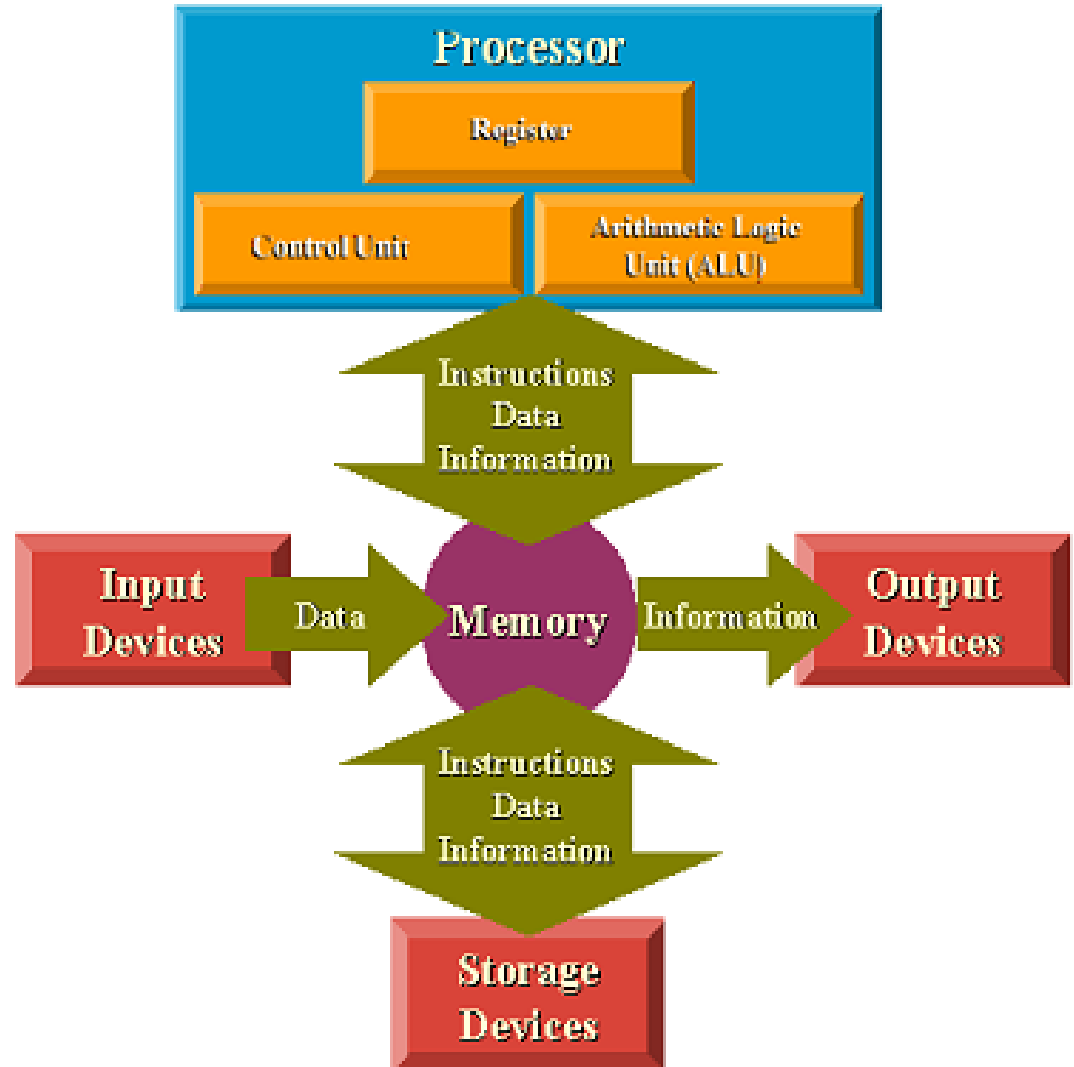


Chương 2 - CẤU TRÚC MÁY TÍNH

- 2.1. Hệ thống máy tính
- 2.2. Bộ xử lý trung tâm
- 2.3. Bộ nhớ máy tính
- 2.4. Thiết bị ngoại vi

2.1. Hệ thống máy tính

Máy tính ngày nay vẫn dựa trên kiến trúc của máy tính Von Neumann

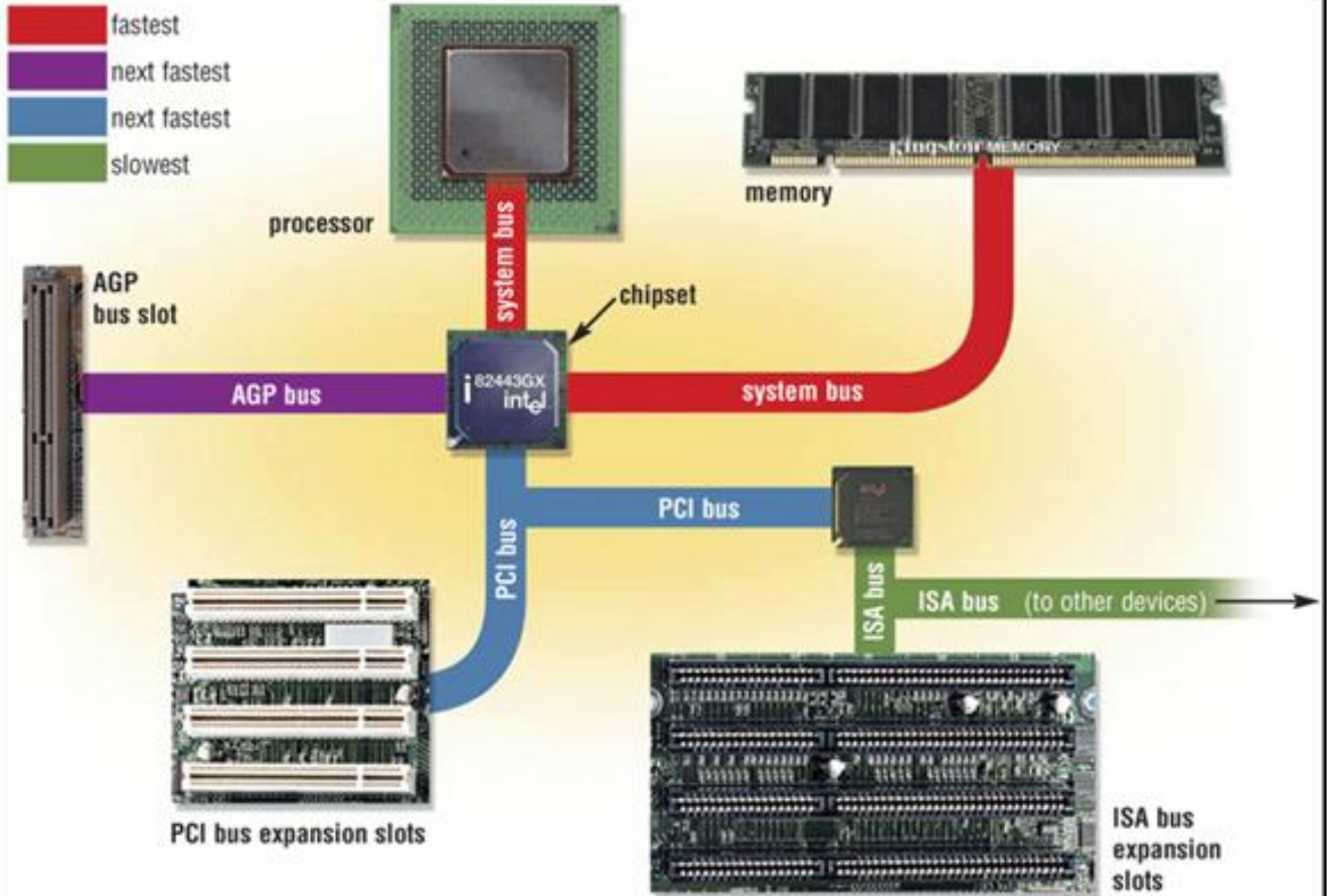




2.1.1. CÁC THÀNH PHẦN

- BỘ XỬ LÝ
- BỘ NHỚ
- THIẾT BỊ NGOẠI VI
- ĐƯỜNG TRUYỀN

TỔ CHỨC VẬT LÝ

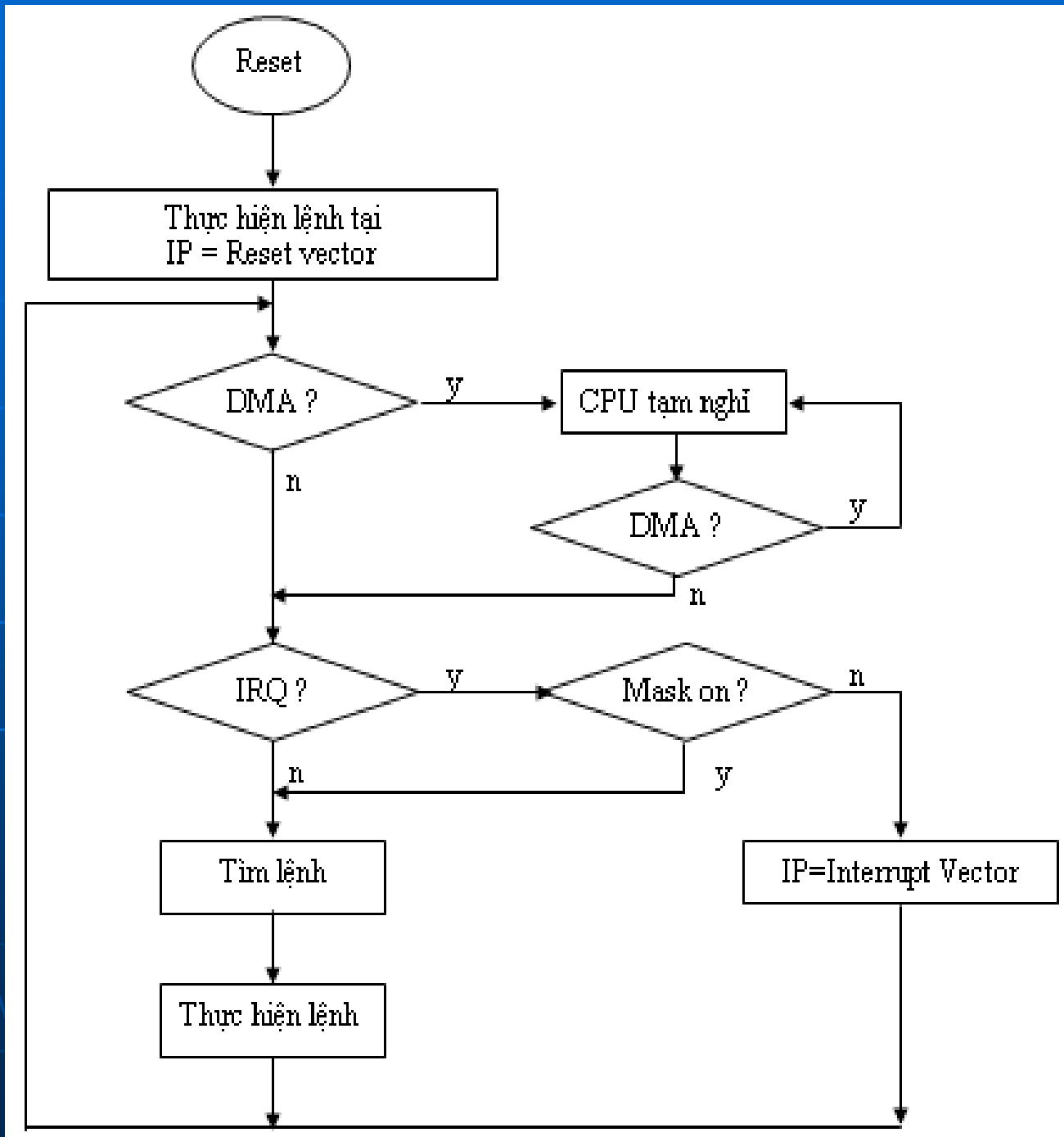




2.1.2. Hoạt động của Hệ thống

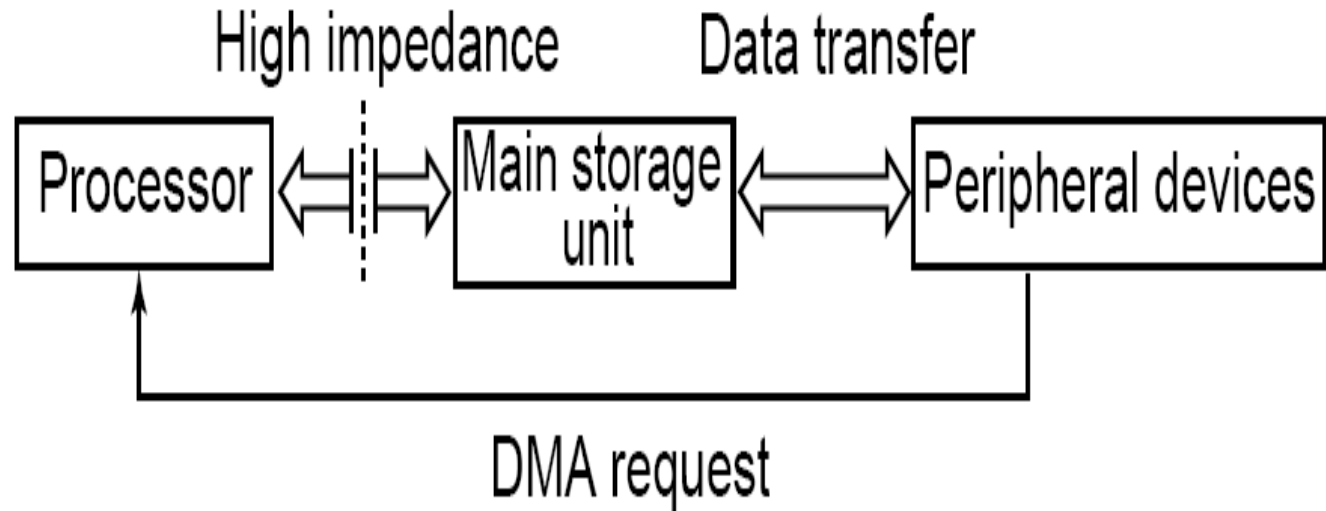
- Khởi động hệ thống
 - POST
 - Nạp hệ điều hành → bộ nhớ
 - Trao quyền điều khiển hệ thống cho HĐH.
- Các hoạt động chính

Hoạt động chính của Hệ thống



a. Cơ chế DMA

Khi cần di chuyển lượng dữ liệu lớn → DMAC (Direct Memory Access Controller)



⇒ Data flow

→ Control flow



Cơ chế hoạt động

Truy cập dữ liệu trực tiếp qua DMAC như sau:

- Thiết bị/Bộ nhớ cần chuyển dữ liệu, phát tín hiệu Data Request → DMAC.
- DMAC đưa tín hiệu đến CPU.
- Nếu chấp nhận, CPU trả lời tín hiệu cho DMAC và tự treo để nhường quyền điều khiển hệ thống Bus cho DMAC.
- DMAC dành quyền điều khiển Bus và tiến hành truy cập dữ liệu theo từng block dữ liệu.
- Sau khi kết thúc việc truy cập dữ liệu, DMAC phát tín hiệu kết thúc CPU và trả quyền điều khiển bus lại cho CPU.

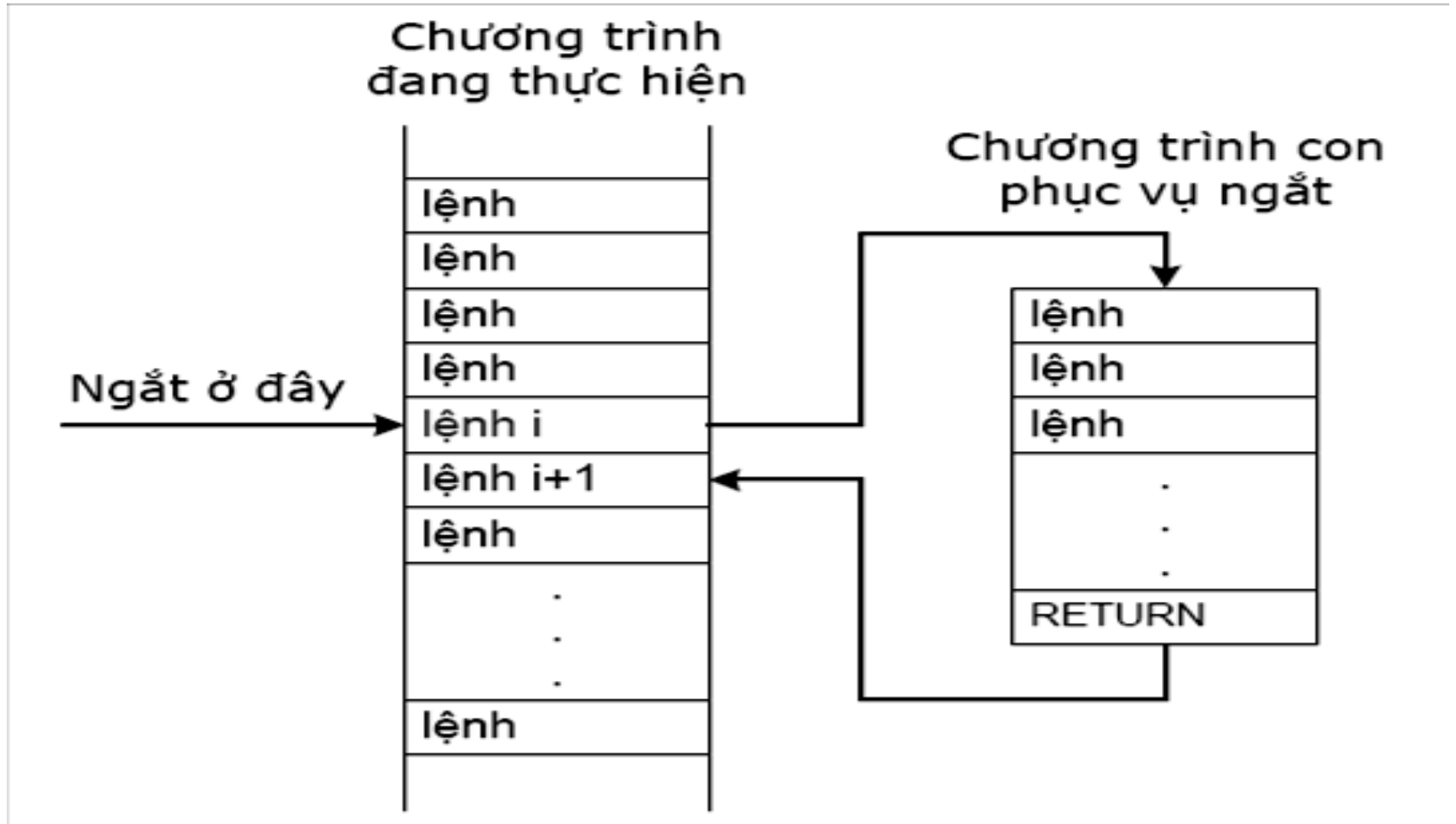


b. Ngắt và đáp ứng ngắt

Có 2 loại ngắt cơ bản:

- Ngắt cứng (hardware Interrupt): Ngắt sinh ra do các thiết bị phần cứng. Có 2 loại là ngắt che được (Maskable Interrupt) và ngắt không che được (Non Maskable Interrupt).
- Ngắt mềm: Ngắt sinh ra do chương trình, ví dụ dùng lệnh `Int` trong hợp ngữ để gọi 1 chương trình con phục vụ ngắt nào đó.

Hoạt động ngắt



Cơ chế hoạt động ngắt

Khi có yêu cầu phục vụ ngắt, CPU sẽ kiểm tra ngắt này có thể phục vụ được thì:

- Cất nội dung thanh ghi cờ \rightarrow Stack.
- Thiết lập cờ $IF = 0$ (Cấm ngắt khác) và $TF=0$ (vi xử lý chạy bình thường).
- Cất $CS:IP \rightarrow$ Stack.

Lấy địa chỉ chương trình con phục vụ ngắt từ bảng vector ngắt $\rightarrow CS:IP$, và thực hiện chương trình này cho đến khi gặp lệnh kết thúc (IRET).

- Lần lượt lấy lại $CS:IP$ và nội dung thanh ghi cờ từ Stack để thực hiện công việc đang tiến hành.

Các mức ưu tiên phục vụ ngắt:

Để tránh tranh chấp xảy ra khi có đồng thời nhiều yêu cầu ngắt thì người ta chia ra các mức ưu tiên để phục vụ:

- Cao nhất là ngắt nội bộ (lỗi chia 0, tràn số, lỗi phần cứng ...)
- → Ngắt cứng không che được
- → Ngắt cứng che được
- → Ngắt chạy từng lệnh.

Trật tự phục vụ ngắt

Bảng yêu cầu ngắt trong máy PC

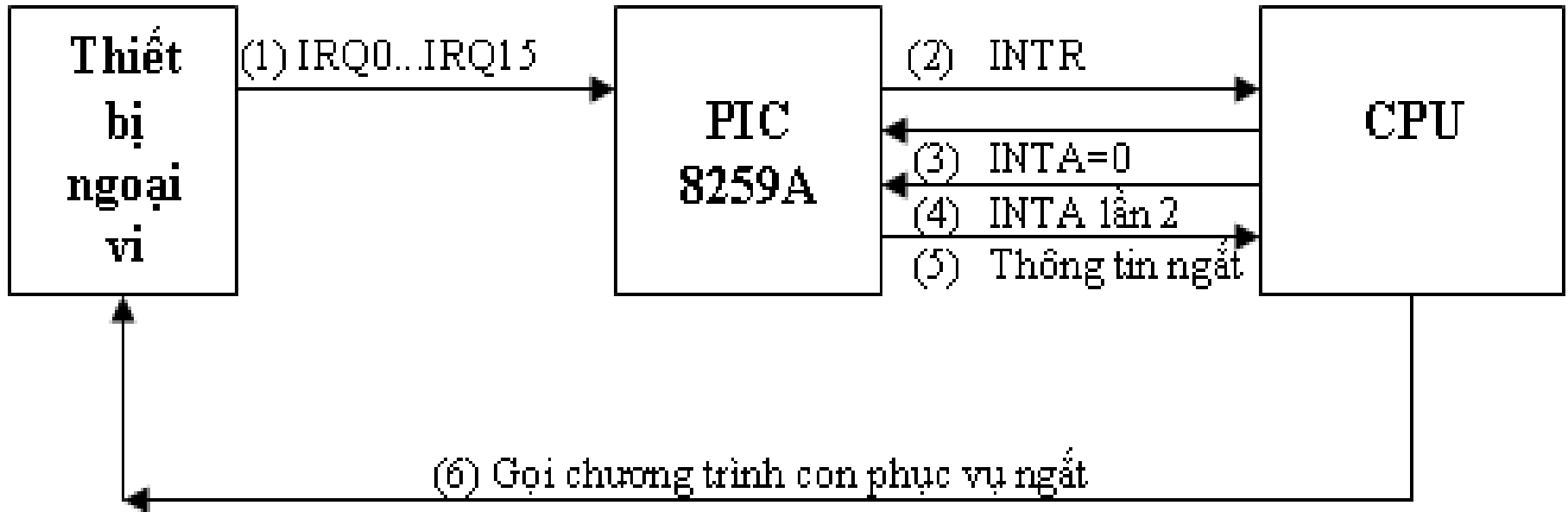
| Interrupt Request | Software Interrupt | Nội dung |
|-------------------|--------------------|---|
| IRQ0 | INT 08h | Ngắt nhịp đồng hồ hệ thống của bộ đếm 0 |
| IRQ1 | INT 09h | Ngắt bàn phím |
| IRQ2 | INT 0Ah | Ngắt của 8259A slaver |
| IRQ3 | INT 0Bh | Ngắt COM2 |
| IRQ4 | INT 0Ch | Ngắt COM1 |
| IRQ5 | INT 0Dh | Ngắt LPT2 |
| IRQ6 | INT 0Eh | Ngắt ổ mềm |
| IRQ7 | INT 0Fh | Ngắt LPT1 / card âm thanh |
| IRQ8 | INT 70h | Ngắt CMOS-RAM/ đồng hồ thời gian thực |
| IRQ9 | INT 71h | Dự trữ |
| IRQ10 | INT 72h | Dành cho bus ISA/PCI |
| IRQ11 | INT 73h | Dành cho bus ISA/PCI |
| IRQ12 | INT 74h | Ngắt PS2 |
| IRQ13 | INT 75h | Ngắt đồng xử lý toán học |
| IRQ14 | INT 76h | Ngắt ổ cứng 1 (IDE1) |
| IRQ15 | INT 77h | Ngắt ổ cứng 2 (IDE 2) |



Mạch điều khiển ngắt PIC

- Mạch PIC 8259A (Programmable Interrupt Controller)
- Mạch này có chức năng phân chia các mức ưu tiên phục vụ ngắt, chọn ngắt phù hợp và gửi thông tin ngắt (INT n) đến CPU.
- Mạch 8259A có 2 loại Master và Slaver

Hoạt động phục vụ ngắt của PIC





Các bước thực hiện lệnh

- Tìm lệnh
- Giải mã
- Thực hiện

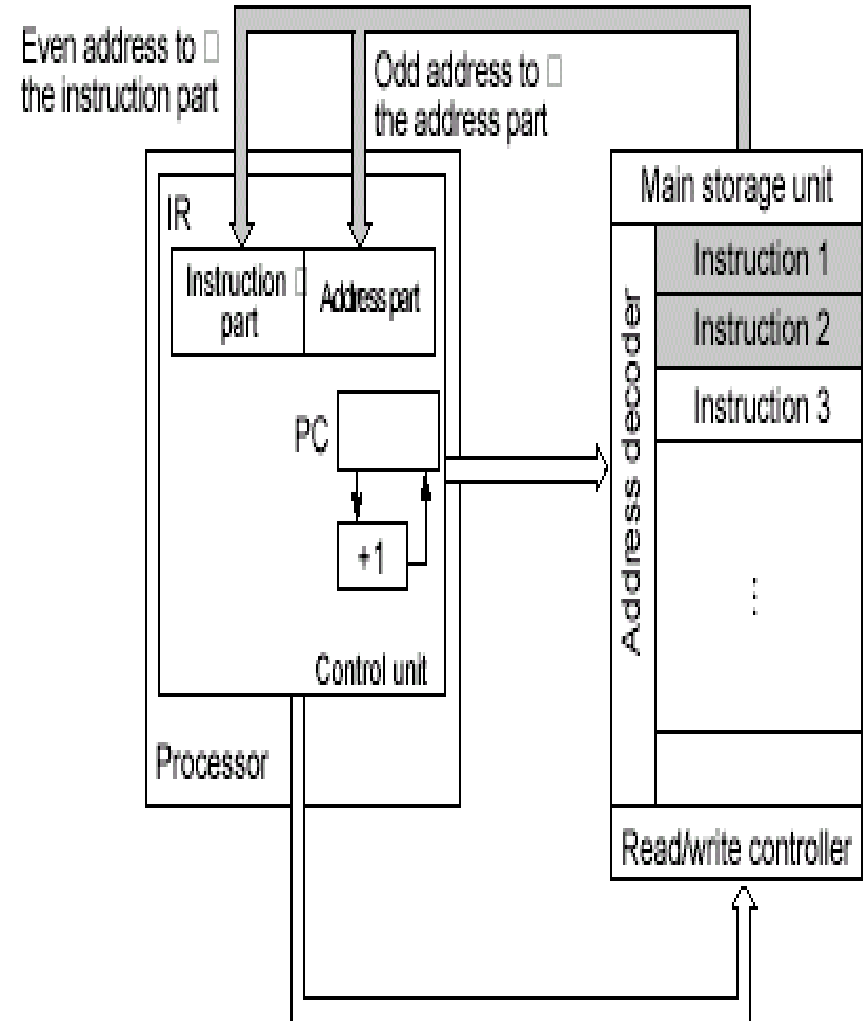
Tìm lệnh

Chu kỳ tìm lệnh

Lệnh được đọc từ bộ nhớ → thanh ghi IR

Phần lệnh được giải mã bằng bộ giải mã

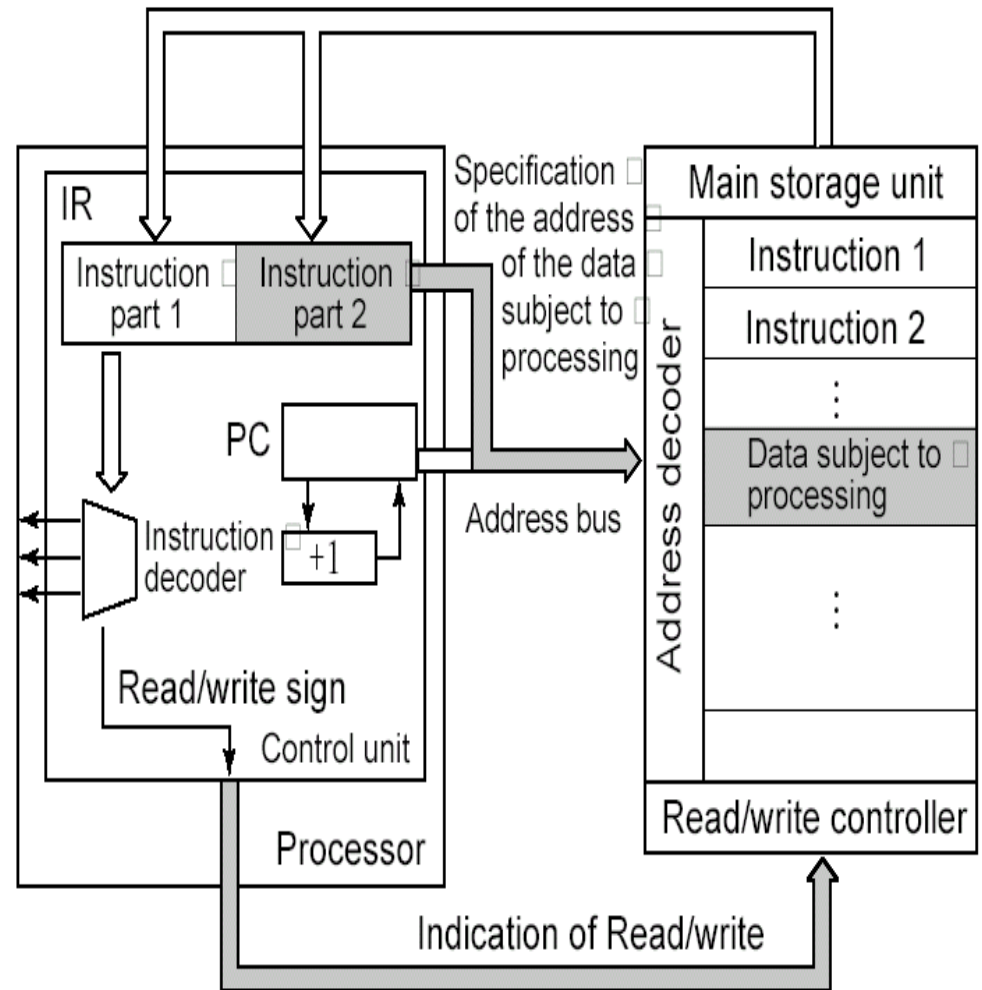
Phần địa chỉ được tính toán để xác định địa chỉ toán hạng



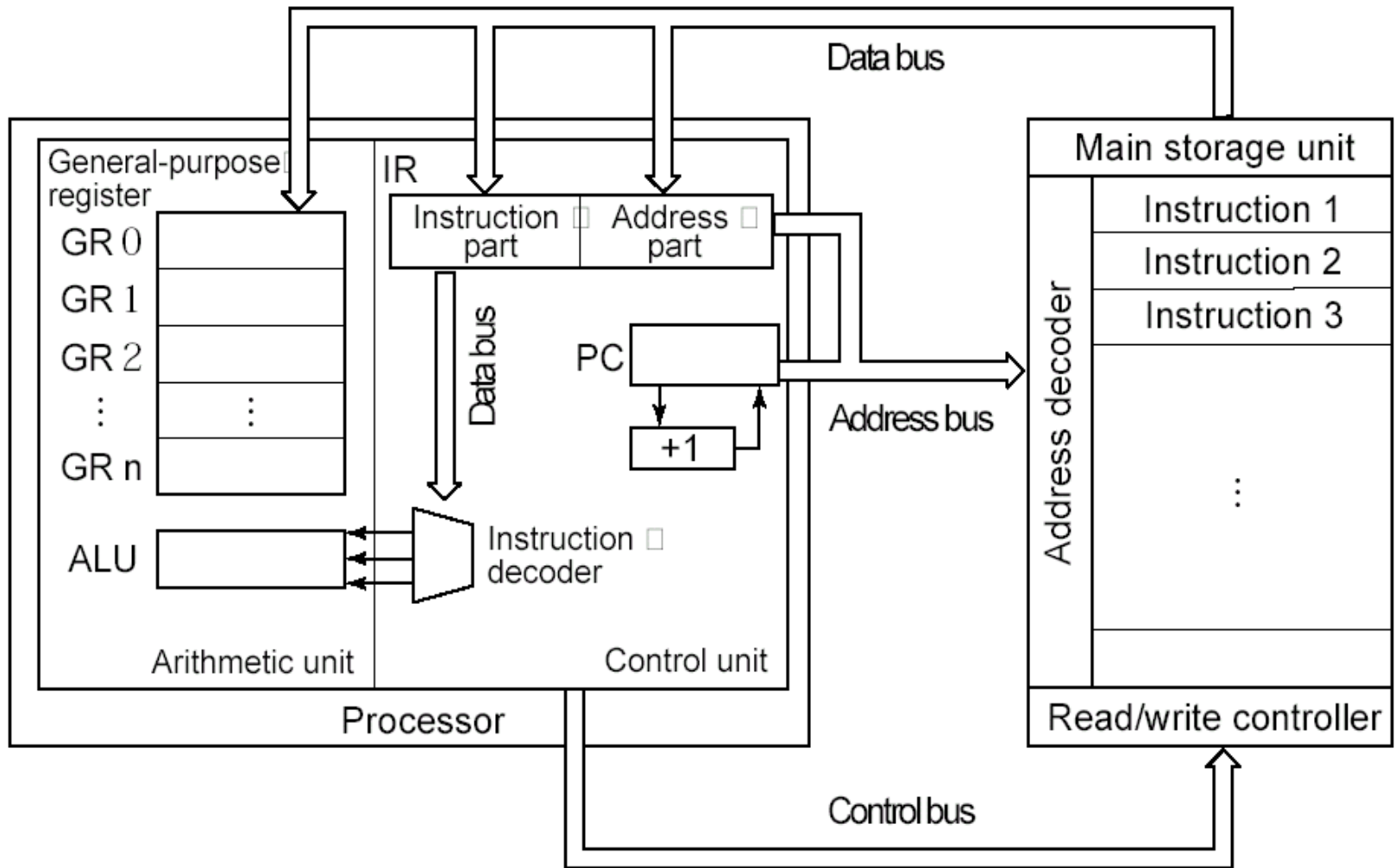
Giải mã

Giải mã lệnh

Phần lệnh →
bộ giải mã lệnh
→ tín hiệu sau
khi giải mã →
các khối thực
hiện lệnh ...



Thực hiện lệnh





2.1.3. CÁC ĐƯỜNG TRUYỀN (BUS)

- BUS NỘI BỘ
- BUS HỆ THỐNG
- BUS MỞ RỘNG



a. BUS hệ thống (System Bus)

- Bus điều khiển (Control Bus),
- Bus dữ liệu (Data Bus),
- Bus địa chỉ dùng để xác định địa chỉ các ô nhớ hay địa chỉ các thiết bị (Address Bus).



Data Bus

- Cung cấp một đường truyền cho việc di chuyển dữ liệu giữa các module hệ thống.
- Thường có 8, 16, 32 hay 64 đường riêng biệt, số các đường được đề cập đến với tên gọi độ rộng của đường truyền dữ liệu.
- Chẳng hạn, nếu đường truyền dữ liệu có độ rộng 8 bit và mỗi chỉ thị có độ dài 16 bit, CPU phải truy cập module bộ nhớ hai lần trong mỗi chu kỳ chỉ thị (lệnh).



Address Bus

- Sử dụng để chỉ định nguồn hay đích của dữ liệu có trên đường truyền dữ liệu. Lấy ví dụ, nếu CPU muốn đọc dữ liệu từ bộ nhớ, nó sẽ đặt địa chỉ ô nhớ cần đọc trên các đường địa chỉ.
- Độ rộng của đường truyền địa chỉ sẽ xác định dung lượng bộ nhớ tối đa có thể có của hệ thống.
- 2^{20}
1MB. c cho



Control Bus

- Sử dụng để điều khiển việc truy cập đến và sử dụng các đường dữ liệu cũng như địa chỉ. Vì các đường này được dùng chung bởi tất cả các thành phần, phải có một phương tiện điều khiển việc sử dụng chúng.
- Các tín hiệu điều khiển truyền cả lệnh lẫn thông tin định thời giữa những module hệ thống.
 - Tín hiệu định thời chỉ ra sự đúng đắn của dữ liệu và thông tin địa chỉ.
 - Tín hiệu lệnh đặc tả thao tác cần được thực hiện.



Các đường điều khiển chính gồm

- Ghi bộ nhớ
- Đọc bộ nhớ
- Ghi nhập/xuất
- Đọc nhập/xuất
- Truyền ACK: Chỉ ra rằng dữ liệu được chấp nhận hay đặt trên đường truyền.
- ...



b. Đường truyền mở rộng (Expanded Bus)

c nhau.

i.

VXL.



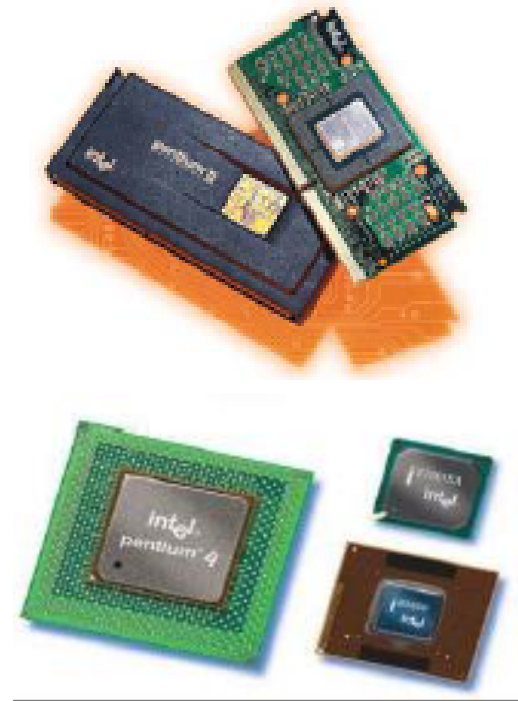
Các loại Bus mở rộng

- ISA, EISA
- AGP bus (Accelerated Graphic Port) - Bus đồ họa tăng tốc: nối ghép card màn hình tăng tốc.
- PCI bus (Peripheral Component Interconnection): nối ghép với các TBNV có tốc độ trao đổi dữ liệu nhanh.
- USB (Universal Serial Bus): Bus nối tiếp đa năng
- IDE (Integrated Driver Electronics): Bus kết nối với ổ đĩa cứng hoặc ổ đĩa CD, DVD.

2.2. BỘ XỬ LÝ TRUNG TÂM

2.2.1. Tổng quan về bộ xử lý trung tâm

- CPU = Central Processing Unit
- Là trái tim của hệ thống, điều khiển mọi hoạt động của hệ thống máy tính.





Đặc điểm về cấu trúc

Công suất: (Độ dài từ máy, khả năng đánh địa chỉ, tốc độ thực hiện lệnh) → Nói đến khả năng xử lý dữ liệu, gồm:

1. Độ dài từ (Word length): Phụ thuộc vào loại Vi xử lý và công nghệ chế tạo: 4/8/16/32/64 bit, độ rộng của thanh ghi, ALU, và Bus dữ liệu bên trong cũng tương ứng với độ dài từ.
2. Khả năng đánh địa chỉ: Tức là khả năng định vị địa chỉ bộ nhớ, điều này do số bit của Bus địa chỉ.



3. Tốc độ xử lý lệnh.

Thường đo bằng tốc độ thực hiện các lệnh dấu phẩy động FLOPS (Floating Point Operations per Second) hoặc tính bằng triệu lệnh / giây – MIPS (Millions Instructions per Second)

$$\text{MIPS} = f * N / (M + T)$$

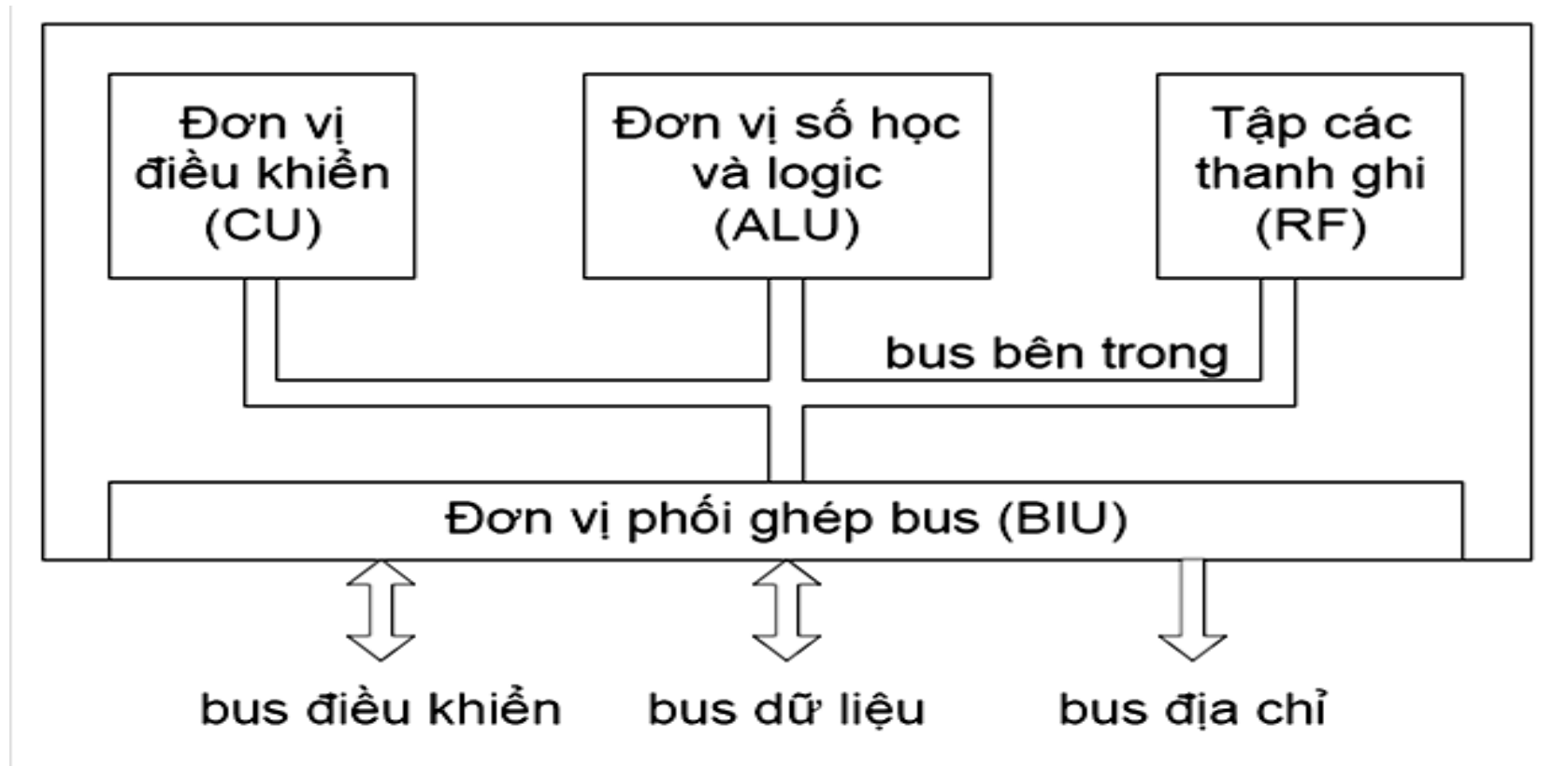
+f: Tần số Vi xử lý (MHz, GHz)

+N: Số lượng ALU

+M: Số lượng vi lệnh / 1 lệnh (khoảng 4 - 7).

+T: Thời gian truy cập bộ nhớ.

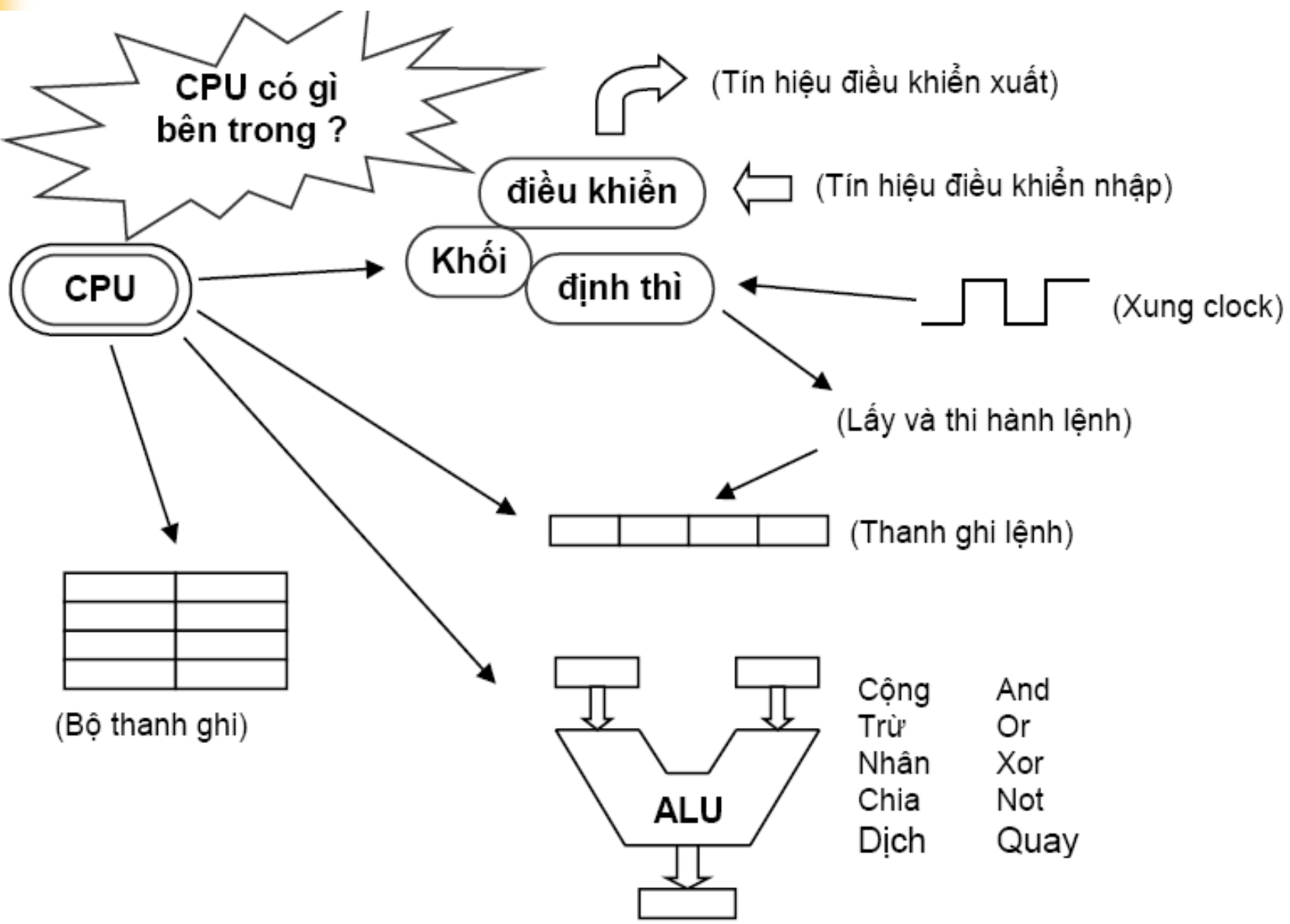
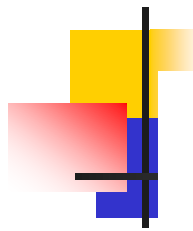
2.2.2. Các thành phần cơ bản của CPU





Các thành phần cơ bản của CPU

- **Đơn vị điều khiển (Control Unit – CU):**
- **Đơn vị số học và logic (Arithmetic and Logic Unit – ALU):**
- **Tập thanh ghi (Register File - RF):**
- **Đơn vị nối ghép bus (Bus interface Unit - BIU):**

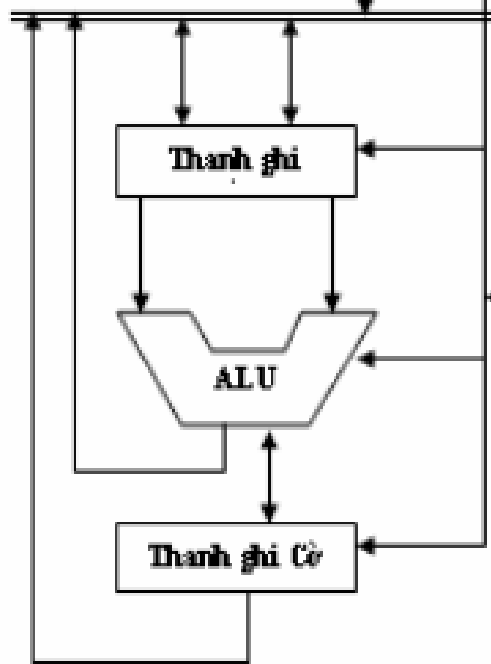


2.2.3. Vi xử lý Intel 8086/8088

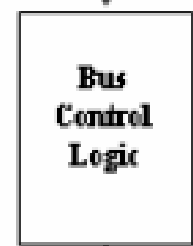
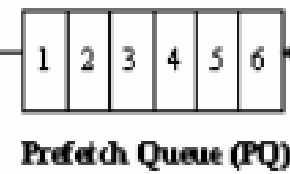
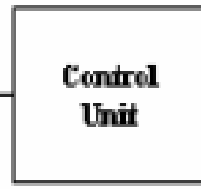
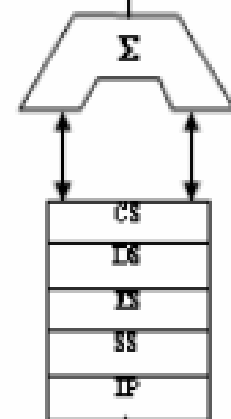
- Vi xử lý 16 bit đầu tiên của Intel sản xuất 1978.
- Tần số 4.7 → 10 MHz.
- 16 bit dữ liệu, 20 bit địa chỉ → quản lý 1MB bộ nhớ.
- Đồng xử lý toán học 8087
- Độ dài từ máy 16 bit.

Execution Unit (EU)

| | |
|----|----|
| AH | AL |
| BH | BL |
| CH | CL |
| DH | DL |
| SP | |
| BP | |
| DI | |
| SI | |



Bus Interface Unit (BIU)





BIU (Bus Interface Unit)

- gồm các thanh ghi đoạn (CS, DS, ES, SS),
 - con trỏ lệnh IP,
 - hàng đợi lệnh trước PQ
 - và đơn vị điều khiển Bus (Bus Control Logic).
- BIU có nhiệm vụ nhận lệnh và đưa về chứa vào hàng đợi lệnh trước theo cơ chế FIFO.



EU (Execution Unit)

- gồm đơn vị Số học – Logic (ALU),
 - đơn vị điều khiển (CU),
 - các thanh ghi đa năng (AX, BX, CX, DX),
 - các thanh ghi con trỏ và chỉ mục (SP, BP, SI, DI),
 - thanh ghi cờ (FR).
- Nhiệm vụ của EU là giải mã lệnh và thực hiện lệnh.



b. Hoạt động

- Hoạt động của 8086 có thể chi làm 3 giai đoạn chính:
 - **Nhận lệnh,**
 - **Giải mã lệnh,**
 - **Thực hiện lệnh.**
- BIU điều khiển các Bus để lấy lệnh từ bộ nhớ, hay thiết bị ngoại vi về chứa tại PQ (Prefetch Queue) gồm 6 byte
- EU nhận lệnh lần lượt từ PQ, giải mã lệnh và tiến hành thực hiện lệnh.
- ...



c. Bộ thanh ghi 8086

Thanh ghi (Register) là một phần bộ nhớ có tốc độ truy cập nhanh nằm ngay bên trong CPU, khác với **bộ nhớ được máy xác định thông qua địa chỉ** thì thanh ghi được **xác định thông qua tên**.

8086 có cả thảy 13 thanh ghi, mỗi thanh ghi dài 16 bit, từng thanh ghi có nhiệm vụ cụ thể và được chia vào các nhóm sau:

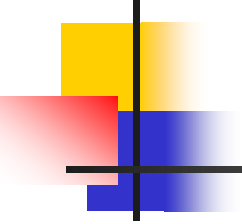
- Các thanh ghi đa năng
- Các thanh ghi đoạn
- Các thanh ghi con trỏ và chỉ mục
- Thanh ghi cờ

Thanh ghi cờ (Flag)

- Là thanh ghi 16 bit, các bit trong thanh ghi này phản ánh trạng thái hiện tại của bộ xử lý, với 8086 thì có 9 bit được sử dụng.

| | | | | | | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|----|---|----|---|----|---|----|
| bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Flag | | | | | OF | DF | IF | TF | SF | ZF | | AF | | PF | | CF |

- + **CF (Carry Flag)**: Là cờ nhớ, khi có nhớ hay có mượn tại bit cao nhất thì $CF=1$.
- + **PF (Parity Flag)**: Cờ chẵn lẻ, $PF=1$ khi tổng số bit 1 là chẵn.
- + **AF (Auxiliary Flag)**: Cờ nhớ phụ, dùng cho các phép tính với mã BCD.

- 
- + **ZF (Zero Flag)**: Cờ Zero, $ZF=1$ khi kết quả tính toán = 0.
 - + **SF (Sign Flag)**: Cờ dấu, $SF=1$ khi kết quả âm.
 - + **TF (Trap Flag)**: Cờ bẫy lỗi, $TF=1$ vi xử lý sẽ chạy từng lệnh.
 - + **IF (Interrupt Flag)**: Cờ ngắt, cho phép gọi ngắt (dùng cho các ngắt che được – Maskable Interrupt).
 - + **DF (Direction Flag)**: Cờ hướng, $DF=1$ thì xử lý xâu ký tự từ phải sang trái.
 - + **OF (Overflow)**: Cờ tràn, $OF=1$ khi kết quả phép tính bị tràn số (vượt giới hạn), dùng cho các phép tính có dấu.
→ Ta có thể kiểm tra sự thay đổi giá trị trong các thanh ghi kể cả thanh ghi cờ bằng chương trình Debug.



2.2.4. Lệnh của Vi xử lý

- Là những chỉ thị cơ bản nhất mà CPU có thể hiểu được
- Kiến trúc tập lệnh
 - CISC
 - RISC
- Một số nhóm lệnh
- Việc thực hiện 1 lệnh



Bộ lệnh (tập lệnh)

- Bộ lệnh là khác nhau với các máy tính khác nhau. Chương trình là một nhóm lệnh lấy từ tập lệnh và được sắp xếp theo thứ tự nào đó, nhằm thực hiện một công việc xác định.
- Các lệnh vi xử lý được viết dưới dạng mã nhị phân, 1 lệnh nhị phân dài từ 1 đến 6 byte tùy mỗi lệnh.
- VD: VXL 8086 có khoảng <200 lệnh

Cấu trúc chung của lệnh 8086/8088

| | | | | | | | | | | | | | | | | | | |
|---------|--|--|--|--------|---|------|--|--------|--|-----|--|--------|--|--|--------------|--|-------------|--|
| | | | | | | | | | | | | | | | Địa chỉ thấp | | Địa chỉ cao | |
| Mã lệnh | | | | D | W | Mode | | REG | | M/R | | | | | | | | |
| Byte 1 | | | | Byte 2 | | | | Byte 3 | | | | Byte 4 | | | | | | |

Mỗi lệnh chiếm khoảng 1 → 6 Byte, trong đó:

- Prefix trước mã lệnh
- Mã lệnh dùng để xác định đó là lệnh gì, ví dụ: Mov có mã lệnh 100010.
- Toán hạng cho biết cái gì sẽ được xử lý (thanh ghi hay ô nhớ).
- Địa chỉ trực tiếp (2 Byte).

Cấu trúc chung của mỗi lệnh 8086/8088

| | | | | | | | | | | | | | | | | | | |
|---------|--|--|--|--------|---|------|-----|--------|--|--|--|--------|--|--|--|--|--------------|-------------|
| | | | | | | | | | | | | | | | | | Địa chỉ thấp | Địa chỉ cao |
| Mã lệnh | | | | D | W | Mode | REG | M/R | | | | | | | | | | |
| Byte 1 | | | | Byte 2 | | | | Byte 3 | | | | Byte 4 | | | | | | |

| D=1 | D=0 |
|-----------------------|-------------------------|
| Dữ liệu đến thanh ghi | Dữ liệu đi từ thanh ghi |

- **D (Direction)**
- **W (Word):** Thanh ghi đang dùng là 8 bit hay 16 bit
- **Mode và R/M dùng để xác định chế độ địa chỉ của lệnh (Addressing Mode).** Đây là cách giúp cho vi xử lý tìm ra toán hạng cho lệnh nào đó, vi xử lý Intel có 7 chế độ địa chỉ

| Thanh ghi | | Mã Thanh ghi - REG |
|-----------|-----|--------------------|
| W=1 | W=0 | |
| AX | AL | 000 |
| BX | BL | 011 |
| CX | CL | 001 |
| DX | DL | 010 |
| SP | AH | 100 |
| DI | BH | 111 |
| BP | CH | 101 |
| SI | DH | 110 |



Các kiểu thực hiện lệnh

- Bộ xử lý giải mã lệnh đã được nhận và phát tín hiệu điều khiển thực hiện thao tác mà lệnh yêu cầu.
- Các kiểu thao tác của lệnh:
 - Trao đổi dữ liệu giữa CPU và bộ nhớ chính
 - Trao đổi dữ liệu giữa CPU và môđun vào-ra
 - Xử lý dữ liệu: thực hiện các phép toán số học hoặc phép toán logic với các dữ liệu
 - Điều khiển rẽ nhánh
 - Kết hợp các thao tác trên



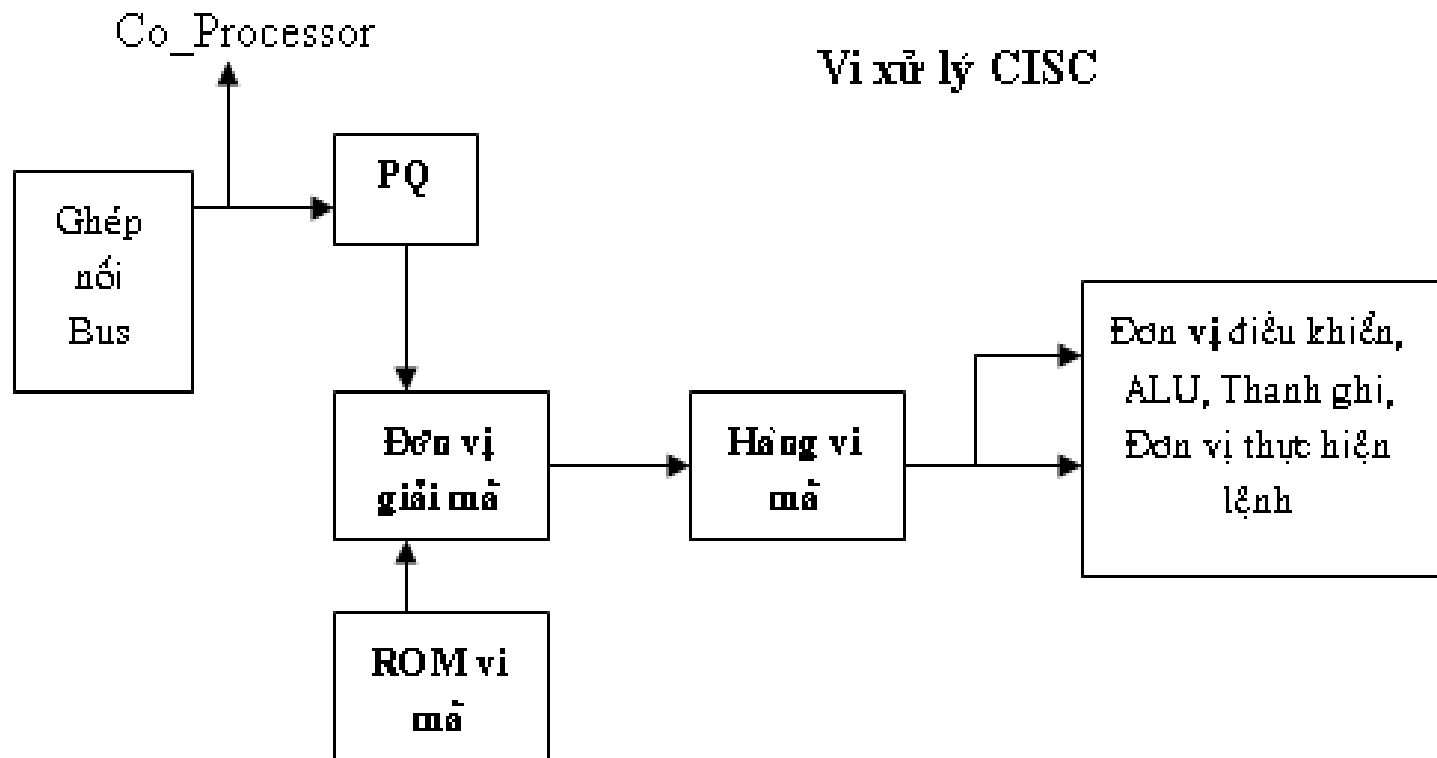
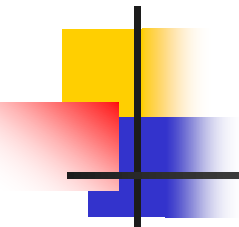
Kiến trúc tập lệnh

- RISC
- CISC



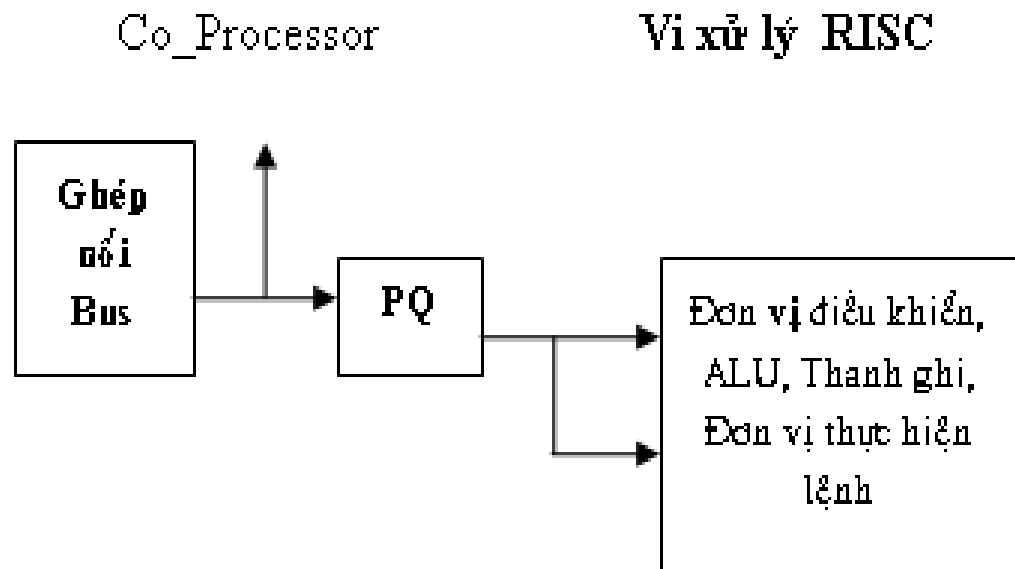
Kiến trúc tập lệnh CISC

- CISC (Complex Instruction Set Computer)
- Các lệnh có độ phức tạp cao
- Các lệnh thực hiện bằng các vi chương trình
- Có sự khác nhau lớn giữa thời gian và kích thước của mỗi lệnh



Tập lệnh RISC

- RISC (Reduced Instruction Set Computer)
- Gồm các lệnh đơn giản
- Thực hiện bằng phần cứng
- Kích thước và thời gian thực hiện các lệnh là như nhau





Các nhóm lệnh cơ bản

- Chuyển dữ liệu
- Xử lý số học với số nguyên
- Xử lý logic
- Điều khiển vào-ra
- Chuyển điều khiển (rẽ nhánh)
- Điều khiển hệ thống
- Xử lý số dấu chấm động
- Xử lý các dữ liệu chuyên dụng



Các lệnh chuyển dữ liệu

| | |
|-------------------------|---|
| MOVE | Copy dữ liệu từ nguồn đến đích |
| LOAD | Nạp dữ liệu từ bộ nhớ đến bộ xử lý |
| STORE | Cất dữ liệu từ bộ xử lý đến bộ nhớ |
| EXCHANGE | Trao đổi nội dung của nguồn và đích |
| CLEAR | Chuyển các bit 0 vào toán hạng đích |
| SET | Chuyển các bit 1 vào toán hạng đích |
| PUSH ngăn xếp | Cất nội dung toán hạng nguồn vào |
| POP | Lấy nội dung đỉnh ngăn xếp đưa đến toán hạng đích |



Các lệnh số học

| | |
|-----------|------------------------------|
| ADD | Cộng hai toán hạng |
| SUBTRACT | Trừ hai toán hạng |
| MULTIPLY | Nhân hai toán hạng |
| DIVIDE | Chia hai toán hạng |
| ABSOLUTE | Lấy trị tuyệt đối toán hạng |
| NEGATE | Đổi dấu toán hạng (lấy bù 2) |
| INCREMENT | Tăng toán hạng thêm 1 |
| DECREMENT | Giảm toán hạng đi 1 |
| COMPARE | Trừ 2 toán hạng để lập cờ |



Các lệnh logic

AND Thực hiện phép AND hai toán hạng

OR Thực hiện phép OR hai toán hạng

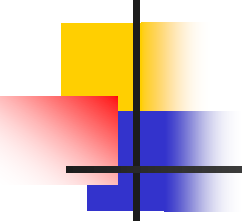
XOR Thực hiện phép XOR hai toán hạng

NOT Đảo bit của toán hạng (lấy bù 1)

TEST Thực hiện phép AND 2 toán hạng
để lập cờ

SHIFT Dịch trái (phải) toán hạng

ROTATE Quay trái (phải) toán hạng



Minh hoạ các lệnh AND, OR, XOR

- Giả sử có hai thanh ghi chứa dữ liệu như sau:
 - $(R1) = 1010\ 1010$
 - $(R2) = 0000\ 1111$
- $R1 \leftarrow (R1) \text{ AND } (R2) = 0000\ 1010$
 - Phép toán AND dùng để xoá một số bit và giữ nguyên một số bit còn lại của toán hạng.
- $R1 \leftarrow (R1) \text{ OR } (R2) = 1010\ 1111$
 - Phép toán OR dùng để thiết lập một số bit và giữ nguyên một số bit còn lại của toán hạng.
- $R1 \leftarrow (R1) \text{ XOR } (R2) = 1010\ 0101$
 - Phép toán XOR dùng để đảo một số bit và giữ nguyên một số bit còn lại của toán hạng.



Các lệnh vào ra chuyên dụng

- INPUT Copy dữ liệu từ một cổng xác định đến đích
- OUTPUT Copy dữ liệu từ nguồn đến một cổng xác định



Các lệnh chuyển điều khiển

- JUMP (BRANCH) - Lệnh nhảy không điều kiện:
 - Nạp vào PC một địa chỉ xác định
- JUMP CONDITIONAL - Lệnh nhảy có điều kiện:
 - Điều kiện đúng nạp PC một địa chỉ xác định
 - Điều kiện sai không làm gì cả
 - Điều kiện thường được kiểm tra thông qua các cờ
- CALL - Lệnh gọi chương trình con:
 - Cất nội dung của PC (địa chỉ trở về) ra một vị trí xác định (thường ở Stack)
 - Nạp vào PC địa chỉ của lệnh đầu tiên của chương trình con
- RETURN - Lệnh trở về từ chương trình con:
 - Khôi phục địa chỉ trở về trả lại cho PC để trở về chương trình chính



Các lệnh điều khiển hệ thống

- NO OPERATION Không thực hiện gì cả
- HALT Dừng thực hiện chương trình
- WAIT Tạm dừng thực hiện chương trình, lặp kiểm tra điều kiện cho đến khi thoả mãn thì tiếp tục thực hiện
- LOCK Cấm không cho xin chuyển nhượng bus
- UNLOCK Cho phép xin chuyển nhượng bus

Ví dụ minh họa (lệnh 16 bit)



**Ví dụ 1 lệnh 16 bit, 4 bit mã lệnh
12 bit đc vùng nhớ (dữ liệu)**

Program Counter (PC) = Address of instruction

Instruction Register (IR) = Instruction being executed

Accumulator (AC) = Temporary storage

Chức năng các TG

0001 = Load AC from Memory

0010 = Store AC to Memory

0101 = Add to AC from Memory

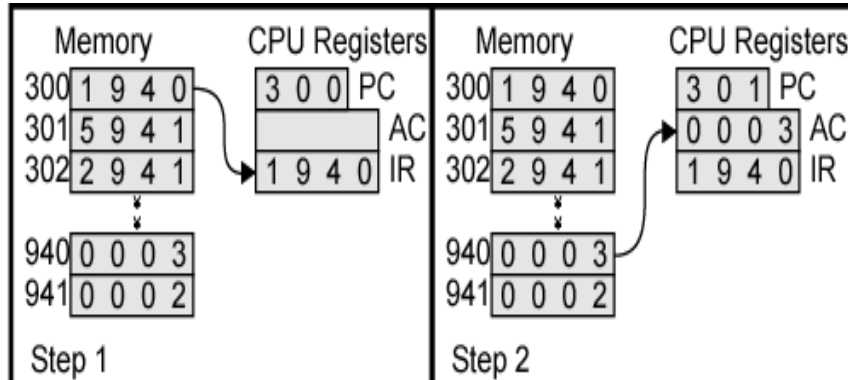
Ví dụ mã của các lệnh...

Ví dụ về lấy lệnh và thực hiện lệnh

Fetch

Execute

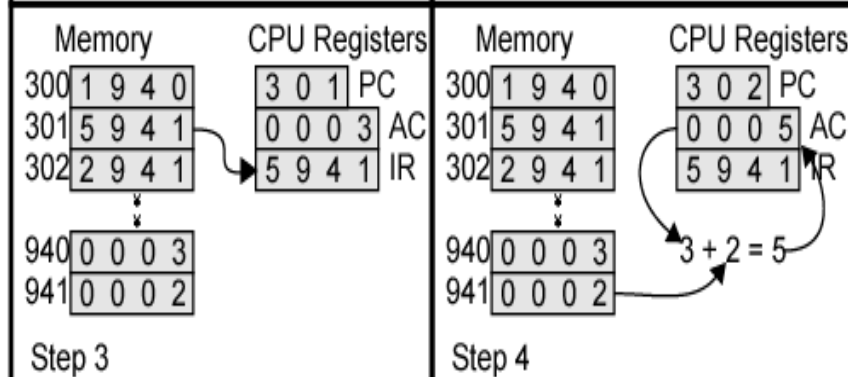
Lệnh 1



Opcode = 1h

Nạp nội dung ô nhớ có đc 940h → AC

Lệnh 2

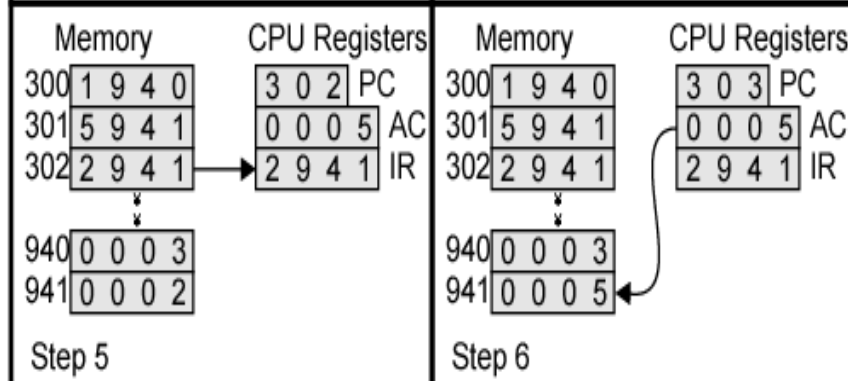


Opcode = 5h

Cộng nội dung ô nhớ có đc 941h với AC.

Kết quả chứa ở AC

Lệnh 3



Opcode = 2h

Lưu nội dung AC → ô nhớ có đc 941h



2.3. Bộ nhớ máy tính (Memory)

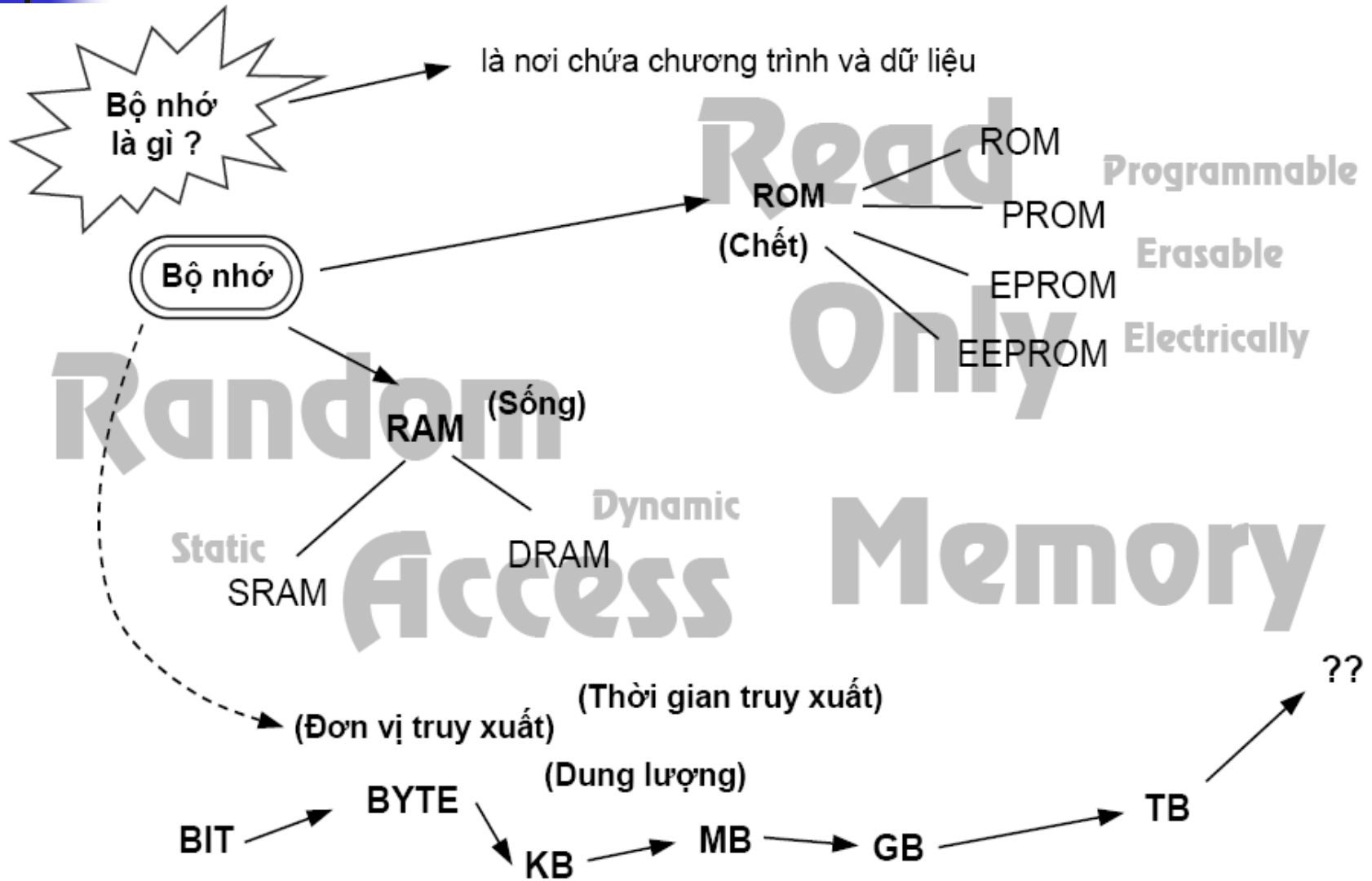
- Chức năng: lưu trữ chương trình và dữ liệu.
- Các thao tác cơ bản với bộ nhớ:
 - Đọc (Read)
 - Ghi (Write)
- Các thành phần chính:
 - Bộ nhớ trong (Internal Memory)
 - Bộ nhớ ngoài (External Memory)



Bộ nhớ trong (Internal memory)

- Chức năng và đặc điểm:
 - Chứa các thông tin mà CPU có thể trao đổi trực tiếp
 - Tốc độ rất nhanh
 - Dung lượng không lớn
 - Sử dụng bộ nhớ bán dẫn: ROM, RAM
- Các loại bộ nhớ trong:
 - Bộ nhớ chính
 - Bộ nhớ cache (bộ nhớ đệm nhanh)

Các loại RAM & ROM



Bộ nhớ chính (Main memory)

- Chứa các chương trình và dữ liệu đang được CPU sử dụng.
- Tổ chức thành các ngăn nhớ được đánh địa chỉ.
- Ngăn nhớ thường được tổ chức theo byte.
- Nội dung của ngăn nhớ có thể thay đổi, song địa chỉ vật lý của ngăn nhớ luôn cố định

| Nội dung | Địa chỉ |
|----------|---------|
| 00101011 | 0000 |
| 11010101 | 0001 |
| 00001010 | 0010 |
| 01011000 | 0011 |
| 11111011 | 0100 |
| 00001000 | 0101 |
| 11101010 | 0110 |
| 00000000 | 0111 |
| 10011101 | 1000 |
| 00101011 | 1001 |
| 11101011 | 1010 |
| 00101000 | 1011 |
| 11111111 | 1100 |
| 10101010 | 1101 |
| 00101011 | 1110 |
| 01010101 | 1111 |



Bộ nhớ đệm nhanh (Cache memory)

- Bộ nhớ có tốc độ nhanh được đặt đệm giữa CPU và bộ nhớ chính nhằm tăng tốc độ CPU truy nhập bộ nhớ
- Dung lượng nhỏ hơn bộ nhớ chính
- Tốc độ nhanh hơn
- Cache thường được chia thành một số mức
- Cache có thể được tích hợp trên chip vi xử lý.
- Cache có thể có hoặc không.



Bộ nhớ ngoài (External memory)

- Chức năng và đặc điểm:
 - Lưu giữ tài nguyên phần mềm của máy tính
 - Được kết nối với hệ thống dưới dạng các thiết bị vào-ra
 - Dung lượng lớn
 - Tốc độ chậm
- Các loại bộ nhớ ngoài:
 - Bộ nhớ từ: đĩa cứng, đĩa mềm
 - Bộ nhớ quang: đĩa CD, DVD
 - Bộ nhớ bán dẫn: Flash disk, memory card



2.4 Hệ thống vào ra và thiết bị ngoại vi

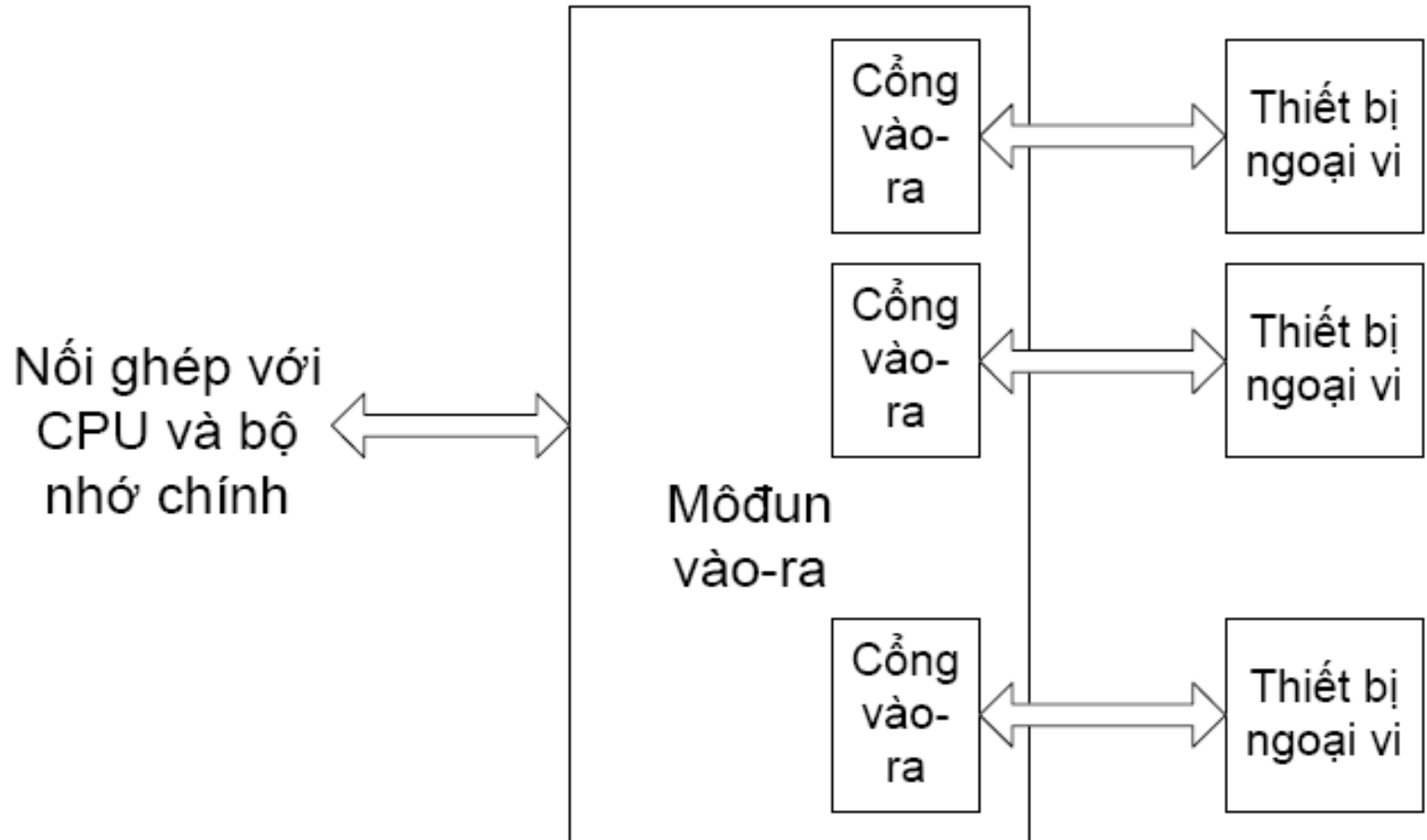
- Hệ thống vào ra
- Thiết bị ngoại vi



Hệ thống vào-ra (Input/Output System)

- Chức năng: trao đổi thông tin giữa máy tính với thế giới bên ngoài.
- Các thao tác cơ bản:
 - Vào dữ liệu (Input)
 - Ra dữ liệu (Output)
- Các thành phần chính:
 - Các thiết bị ngoại vi (Peripheral Devices)
 - Các môđun vào-ra (IO Modules)

Cấu trúc cơ bản của hệ thống vào-ra





Các thiết bị ngoại vi

- Chức năng: chuyển đổi dữ liệu giữa bên trong và bên ngoài máy tính.
- Các loại thiết bị ngoại vi (TBNV) cơ bản:
 - Thiết bị vào: bàn phím, chuột, máy quét ...
 - Thiết bị ra: màn hình, máy in ...
 - Thiết bị nhớ: các ổ đĩa ...
 - Thiết bị truyền thông: MODEM ...



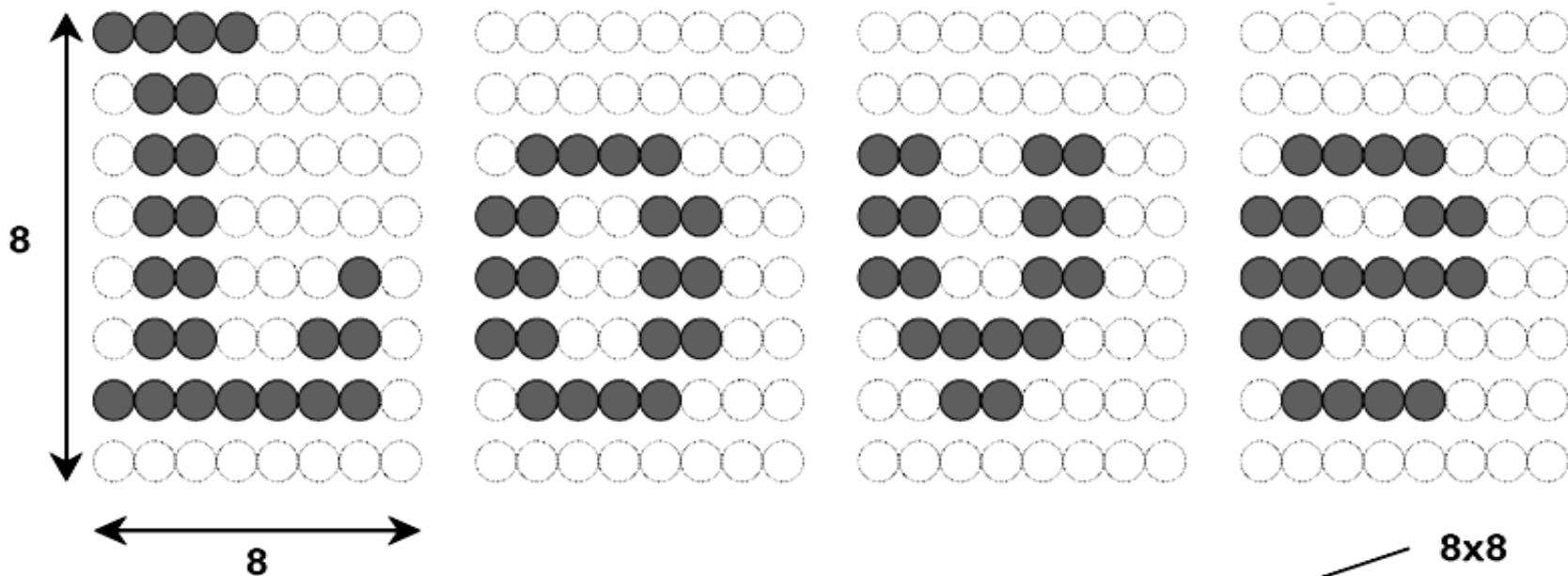
Môđun vào-ra

- Chức năng: nối ghép các TBNV với máy tính.
- Mỗi môđun vào-ra có một hoặc một vài cổng vào-ra (I/O Port).
- Mỗi cổng vào-ra được đánh một địa chỉ xác định.
- Các TBNV được kết nối và trao đổi dữ liệu với máy tính thông qua các cổng vào-ra.

Ma trận ký tự trên màn hình

Thiết bị
ngoại vi

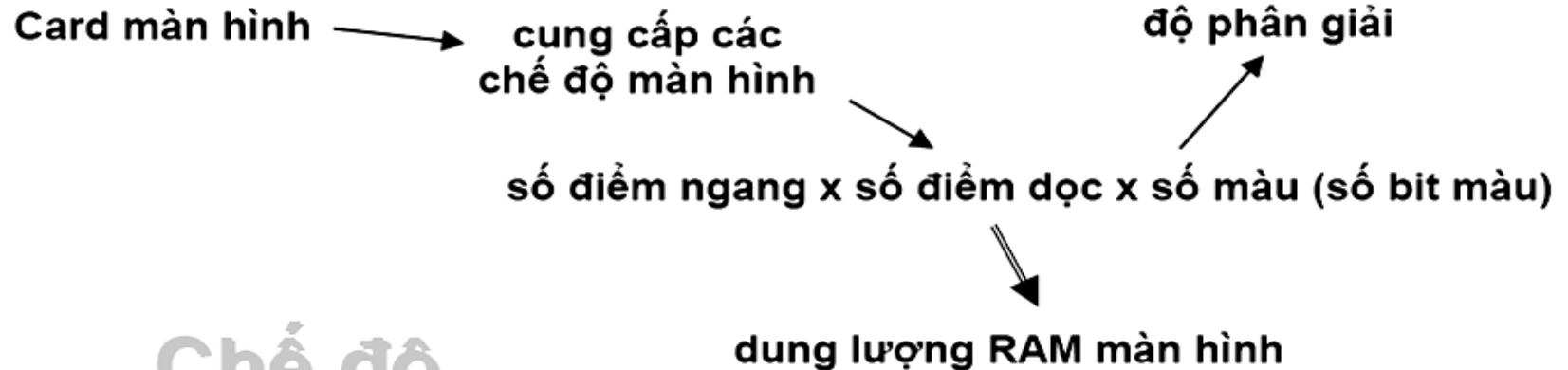
Chế độ văn bản



Kích thước

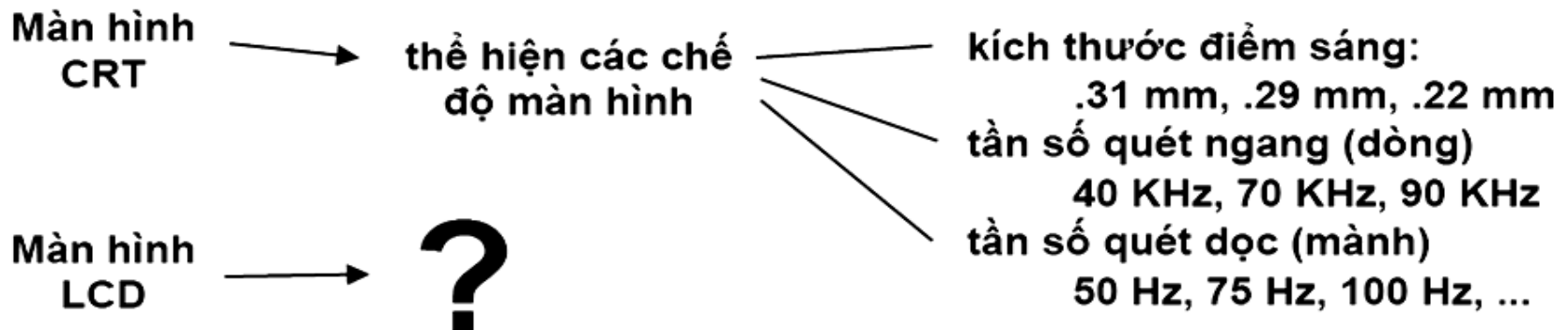
- 8x8
- 14x8
- 16x8

Màn hình



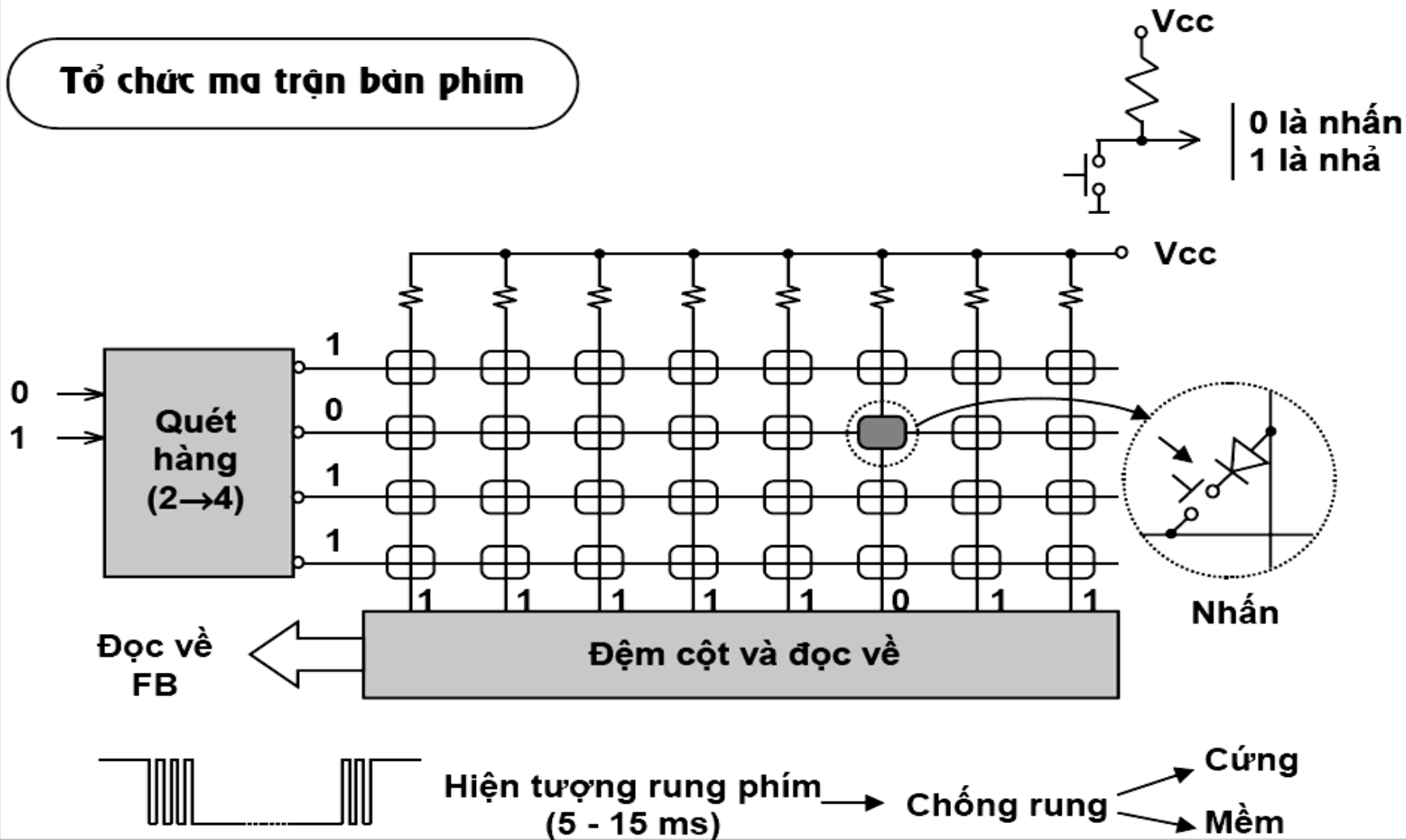
Chế độ đồ họa

800 x 600 x 16bit = 960.000 byte ⇒ 1MB
1024 x 768 x 32bit = 3.145.728 byte ⇒ 4 MB



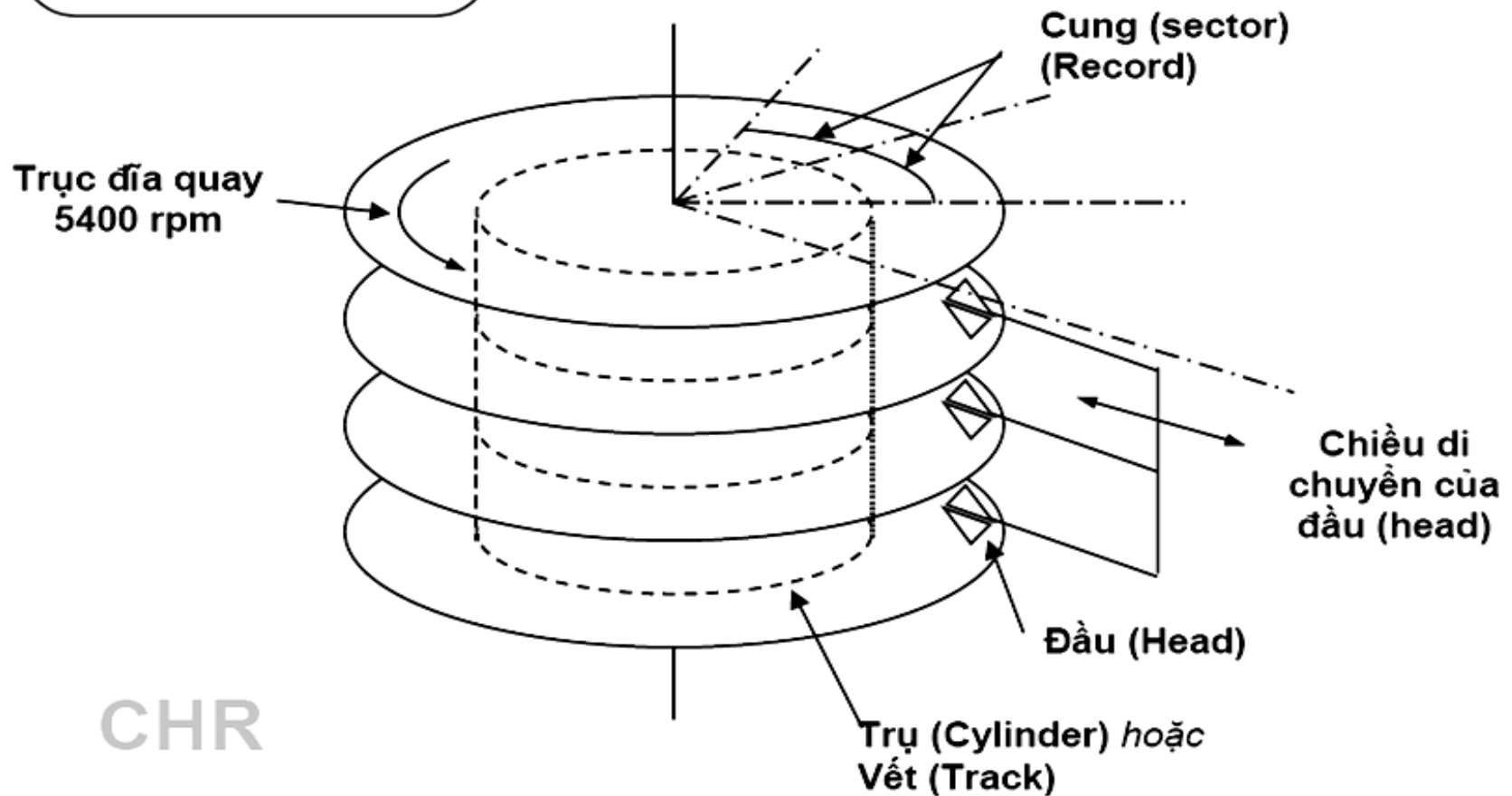
Bàn phím

Tổ chức ma trận bàn phím



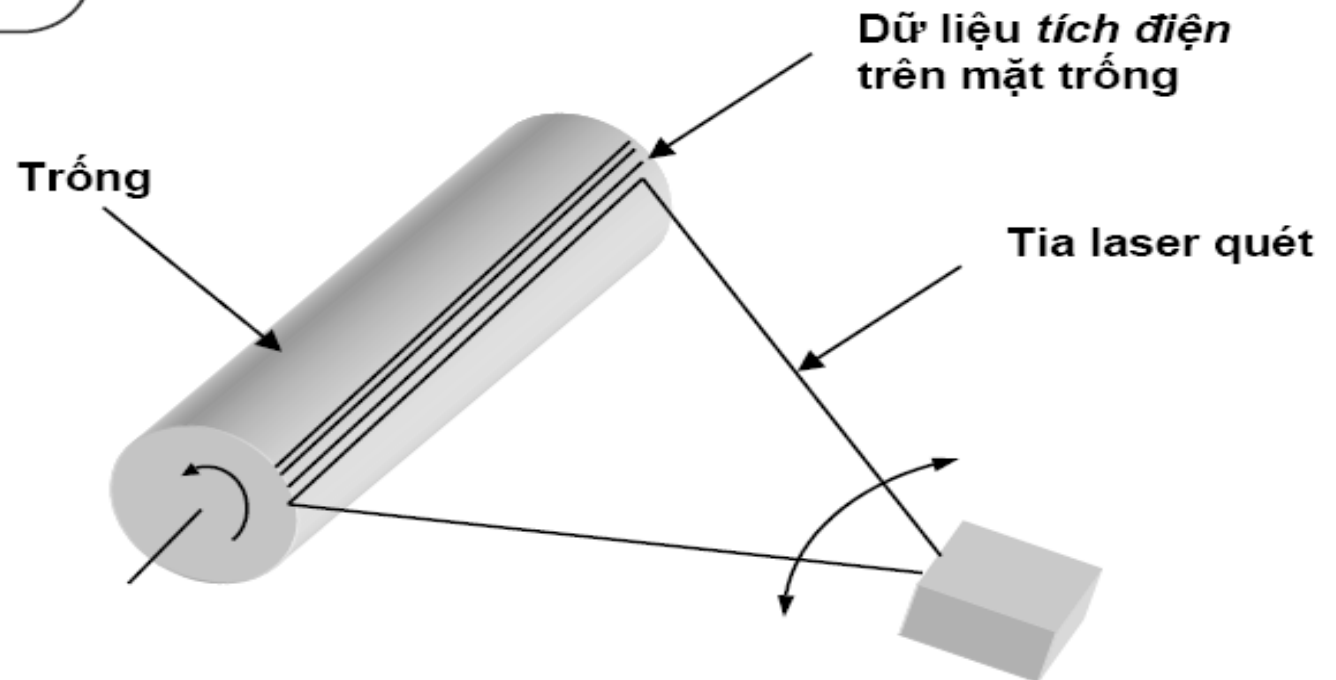
Đĩa cứng vật lý

Phân chia đĩa vật lý



Máy in

Máy in laser





Máy tính PC ngày nay

- Máy tính cá nhân hay PC được giới thiệu lần đầu vào năm 1981, ngày nay có rất nhiều loại khác nhau, lúc đầu là các PC/AT → PC/XT, từ năm 1995 Intel và Microsoft đưa ra đề nghị về cấu hình máy tính cá nhân PC97, PC98, PC99, PC2001...



PC99

- Cấu hình PC99 quy định loại bỏ bus ISA ra khỏi cấu trúc máy PC, thay thế các giao diện tuần tự và song song cổ điển bằng các giao diện nhanh hơn và thông minh hơn (USB, IEEE 1394) card màn hình phải dùng giao diện PCI hoặc AGP.



PC2001

- Cấu hình PC2001 quy định cho cấu hình các máy tính cá nhân từ năm 2001 về sau. Loại bỏ cấu trúc IBM-PC (gồm bus ISA, giao diện tuần tự và song song cổ điển chỉ trừ máy in, ...), yêu cầu PC tối thiểu có vi xử lý 400MHz, bus PCI, giao diện USB và IEEE 1394.

Các yêu cầu hệ thống cho PC 2001

| Tham số | PC | Trạm làm việc | Xách tay |
|--------------------|--|--------------------------------------|--------------------------------|
| Vi xử lý | >667MHz | >700MHz | >400MHz |
| Cache | 128KB | 256KB | 128KB |
| RAM | 64, 128 MB | 128MB → 2GB | 64MB |
| Quản lý nguồn | ACPI 1.0b | ACPI 1.0b | ACPI 1.0b |
| Bus mở rộng | USB, PCI, SCSI | USB, PCI, PCI 64 bit, SCSI | USB, PCI |
| Giao diện ngoại vi | 2USB | 2USB | 1USB |
| Thiết bị hiển thị | Video playback, đầu vào cho tín hiệu video tuần tự | Màn hình lớn, AGP 1.1 | Màn hình cấy trong hệ thống |
| Thiết bị lưu trữ | Ổ đĩa cứng, ổ CD hay DVD | Nhiều Ổ đĩa cứng, ổ CD hay DVD | Ổ đĩa cứng, ổ CD hay DVD |



PC2001

- Các thiết bị thu nhận dữ liệu không nằm trong vỏ máy đều phải có giao diện tuần tự USB và theo chuẩn USB HID 1.1 (Universal Serial Bus Human Interface Device). Các thiết bị đều hỗ trợ chuẩn Plug - n - Play, các thiết bị mạng tuân theo chuẩn NDIS 5.0, ...



PC2001

- Các thiết bị hiển thị: Nếu máy có giao diện hiển thị số thì ổ cắm và màn hình cần tuân thủ đặc tả DIV (Digital Visual Interface Specification). Thiết bị hiển thị cần có độ phân giải tối thiểu 1024x768, 32 bit màu, khả năng 3D, Plug n Play, quản lý nguồn,...máy in dù vẫn hỗ trợ giao diện tuần tự cổ điển hoặc giao diện song song IEEE 1284 nhưng phải trang bị giao diện USB và IEEE 1394.



PC2001

- Thiết bị truyền dữ liệu: Modem phải tuân theo bộ lệnh V.250 AT, Modem ISDN cũng phải tuân theo bộ lệnh AT cơ bản và các lệnh mới đặc trưng cho ISDN. Mọi modem ngoại vi đều phải dùng giao diện USB để nối với máy tính.



Một số ví dụ

Aspire SA85

Là phiên bản mới nhất trong dòng Aspire, vừa được Acer giới thiệu tại thị trường Việt Nam trong tháng 10.

Mainboard: mini-ATX 661FX-M7 (v1.2A). card đồ họa, card âm thanh và card mạng được tích hợp trên Mainboard.

các cổng thông dụng như PS/2, serial, parallel, VGA, LAN, 3 ngõ Audio và 4 cổng USB; mặt trước có thêm 2 cổng USB và 2 ngõ Audio.

BXL Intel Celeron D 336 (2,8GHz, 256KB Cache L2), hoặc Pentium 4 HT 524 (3,06GHz, 1MB Cache L2).

Bộ nguồn P4 300W (20 chân) của Liteon



VIA PC1-1500

- bo mạch chủ (BMC) mini-ITX
- 1 khe cắm RAM (tối đa 1GB),
- 2 giao tiếp IDE cho ổ cứng, CD-ROM
- 1 khe PCI.
- Các cổng cơ bản như PS/2, VGA (tích hợp), serial, parallel, 3 ngõ audio, 2 cổng USB và card mạng tích hợp; mặt trước có 2 cổng USB và 2 ngõ audio.
- BXL VIA C3 1,0GHz (133,6MHzx7,5),
- DDR-SDRAM 256MB
- ổ cứng ATA 40GB.



VIA PC1-2500

- BMC dạng mini-ATX kích thước (19x22,5cm),
- có 2 khe gắn RAM (tối đa 2GB),
- 2 giao tiếp IDE, 2 giao tiếp SATA
- 2 khe PCI
- các cổng cơ bản như PS/2, VGA (tích hợp), serial parallel, 3 ngõ audio, 4 cổng USB và card mạng tích hợp mặt trước có 2 cổng USB và 2 ngõ audio.
- BXL VIA C7 1,5GHz, đồ họa tích hợp VIA/S3C UniChrome 64 MB,
- DDR2-SDRAM 256MB
- ổ cứng SATA 40GB





CMS PowerCom 7020

- Ngoài các cổng cơ bản ở mặt sau như PS/2, Serial, Parallel, 3 ngõ Audio, 4 cổng USB và card mạng (tích hợp). Mặt trước được tăng cường thêm 2 cổng USB, 2 ngõ Audio ...
- BXL Core 2 Duo 6600 (2,4GHz socket 775, FSB 1066MHz, cache L2 4MB), 512 MB PC2 4300 (DDR2 533MHz), card đồ họa ATI Radeon X300/TD 128MB; ổ cứng SATA 80GB trên nền chipset 965 của Intel.

Những tính năng mới nhất

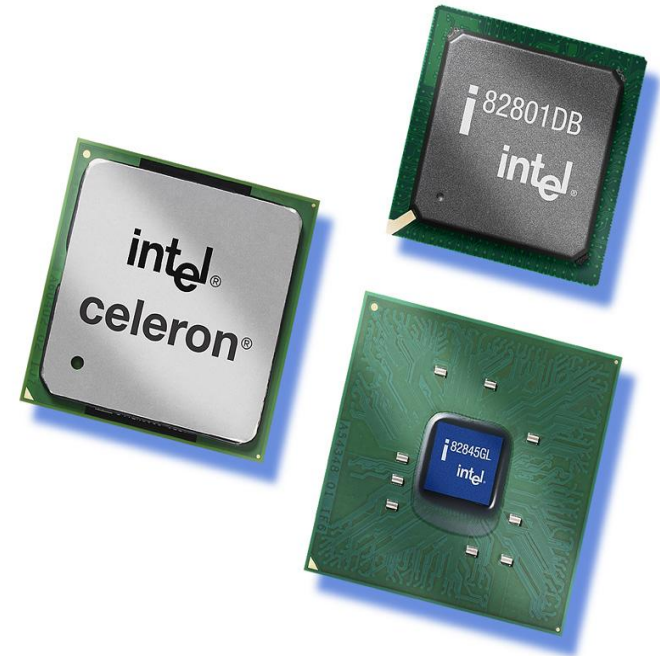
- Bộ mạch chủ: bo mạch chủ 2 socket cho phép chạy 2 BXL lõi kép.
- 2 khe PCI-Express x16 với băng thông đúng mức x16 cho hệ đồ họa kép.
- CPU lõi kép tốc độ cao → >3GHz...
- RAM sử dụng DDR2 có dung lượng cao đến 2GB.
- ổ cứng khá rẻ và tốc độ lại rất cao dung lượng lớn khoảng trợ SATA0-2, có thể tận dụng được ưu thế tốc độ hoặc tính liệu của RAID-0 hoặc RAID-1 .
- ổ quang với khả năng ghi đĩa giá thành đang giảm,
- HĐH 64-bit ...



Các thành phần cơ bản

Bộ xử lý

- Bộ xử lý trung tâm,
- Các bộ đồng xử lý,
- Các vi mạch xử lý khác...



Chức năng

Điều khiển mọi hoạt động của máy tính

Các thành phần cơ bản

Bộ nhớ

- Bộ nhớ trong
- Bộ nhớ ngoài (Storage Devices).

Memory



Hard Disk



CD-ROM / DVD ROM





Các thành phần cơ bản

Các thiết bị nhập

- keyboards,
- mouse
- voice recognition devices,
- sound cards,
- modems,
- scanners,
- tape drives,
- CD/DVD drives,
- video camera ...

Các thành phần cơ bản

Các thiết bị xuất

- monitors,
- printers,
- fax,
- modems,
- plotters,
- CD-recordable discs,
- Speaker ...

PC Speakers



Monitor





ng ISA (Industry Standard Architecture)

ISA .

-

- .

chai (bottleneck)

Bus Micro Channel Architecture (MCA)

-

nh hơn bus ISA. (-

c cao.)

n 100 MHz.

C. Bus EISA

m 9 công ty (

Zenith Data System

ng.

nh.

i giây.

ng PCI

-

nh.

-

y ra.



Plug and Play

- y.
- Với người dùng thì chuẩn này giúp họ có thể tự gắn thêm các thiết bị một cách dễ dàng mà không cần phải là chuyên gia máy tính.



Bus PCMCIA

- Bus PCMCIA (personal Computer Manufacturer's Computer Interface Adapter).



ng.



nh notebook.

Bus nội bộ



.



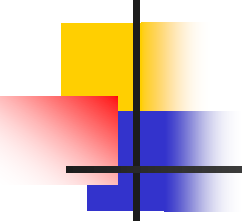
i

BXL

Các thanh ghi đa năng (General)

- Có nhiệm vụ chứa tham số cho mã lệnh, chứa dữ liệu cho một số phép toán, địa chỉ công khi làm việc với ngoại vi, ...
- Là các thanh ghi 16 bit (AX, BX, CX, DX) nhưng có khi lại dùng như các thanh ghi 8 bit (AH, AL, BH, BL, CH, CL, DH, DL) một cách độc lập.

| Thanh ghi | 15 | 7 | 0 |
|-----------|----|----|---|
| AX | AH | AL | |
| BX | BH | BL | |
| CX | CH | CL | |
| DX | DH | DL | |

- 
-
- + AX (Accumulator): Chuyên dùng để chứa kết quả các thao tác lệnh, chứa số hiệu của hàm khi gọi ngắt (ah), ...
 - + BX (Base): Thường dùng cho địa chỉ cơ sở, đôi khi cũng dùng chứa kết quả tạm trong các thao tác lập trình.
 - + CX (Count): Chứa số lần lặp trong các lệnh lặp như Loop, hoặc số lần quay bit hay dịch bit trong các lệnh Rotate, Shift.
 - + DX (Data): Là thanh ghi dữ liệu, chứa dữ liệu trong các phép nhân chia 16 bit, chứa địa chỉ cổng ngoại vi.

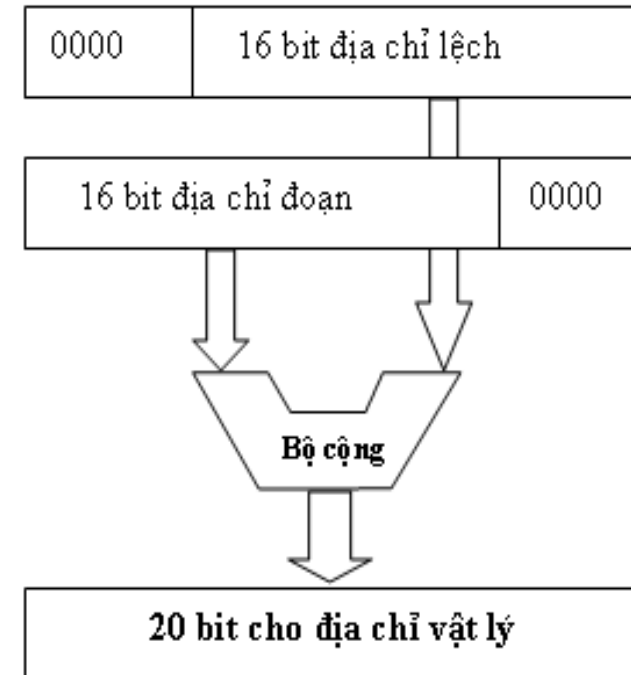
Các thanh ghi đoạn (Segment)

- Khi dùng các thanh ghi 16 bit để quản lý địa chỉ các ô nhớ trong 1M bộ nhớ thì không đủ.
- → dùng 2 thanh ghi 16 bit quản lý địa chỉ các ô nhớ trong vùng nhớ 1M này theo cách sau:
 - Người ta chia bộ nhớ thành các đoạn với độ rộng tối đa là 64K ($=2^{16}$ ô nhớ),
 - trong mỗi đoạn dùng 1 thanh ghi khác để chứa các ô nhớ trong mỗi đoạn gọi là địa chỉ lệch.
- Thanh ghi đoạn dùng để chứa các địa chỉ đoạn, 8086 có các đoạn sau: Mã lệnh, Dữ liệu, Stack

■ Để quản lý các ô nhớ 8086 dùng cặp địa chỉ Segment:offset (đoạn: lệch). Địa chỉ này gọi là địa chỉ logic, còn địa chỉ vật lý của bộ nhớ được tính như sau:

Địa chỉ vật lý = địa chỉ đoạn * 16 + địa chỉ lệch

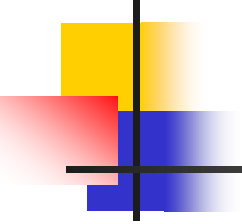
■ Trong thực tế để tính địa chỉ vật lý người ta dịch trái địa chỉ đoạn 4 bit sau đó cộng với địa chỉ lệch.



Các thanh ghi đoạn

- + CS (Code Segment): Thanh ghi đoạn mã lệnh, kết hợp với IP tạo thành cặp CS:IP xác định địa chỉ mã lệnh trong bộ nhớ.
- Ví dụ: CS:IP = 1000 : 0100 là địa chỉ của đoạn CS=1000h, IP là địa chỉ lệch (Offset) có giá tại ô thứ 0100h trong đoạn CS đó. Như vậy địa chỉ vật lý của ô nhớ ở trên là 10100h.

| |
|--------------|
| 1000: ffff |
| . |
| . |
| . |
| → 1000: 0100 |
| . |
| . |
| . |
| 1000: 0001 |
| 1000: 0000 |
| CS:IP |

- 
-
- + DS (Data): Là thanh ghi đoạn dữ liệu kết hợp với SI, DI để xác định địa chỉ cho dữ liệu.
 - + ES (Extra): Là thanh ghi đoạn dữ liệu mở rộng, tương tự DS.
 - + SS (Stack): Thanh ghi đoạn ngăn xếp, thường kết hợp với SP để xác định địa chỉ đỉnh của Stack.



Các thanh ghi con trỏ, chỉ mục

- + IP (Instruction Pointer): Thanh ghi con trỏ lệnh, luôn trỏ vào lệnh kế tiếp sẽ được thực hiện, kết hợp với CS.
- + BP (Base Pointer): Thanh ghi con trỏ cơ sở, thường kết hợp với SS.
- + SI (Source Index): Thanh ghi chỉ mục nguồn, kết hợp với DS.
- + DI (Destination Index): Thanh ghi chỉ mục đích, kết hợp với DS.

Các chế độ địa chỉ của 8086

| R\M | Mod | Mod = 00 | Mod = 01 | Mod = 10 | Mod = 11 | |
|-----|-----|-----------|--------------|---------------|----------|-----|
| | | | | | W=0 | W=1 |
| 000 | | [BX]+[SI] | [BX]+[SI]+D8 | [BX]+[SI]+D16 | AL | AX |
| 001 | | [BX]+[DI] | [BX]+[DI]+D8 | [BX]+[DI]+D16 | CL | CX |
| 010 | | [BP]+[SI] | [BP]+[SI]+D8 | [BP]+[SI]+D16 | DL | DX |
| 011 | | [BP]+[DI] | [BP]+[DI]+D8 | [BP]+[DI]+D16 | BL | BX |
| 100 | | [SI] | [SI]+D8 | [SI]+D16 | AH | SP |
| 101 | | [DI] | [DI]+D8 | [DI]+D16 | CH | BP |
| 110 | | D16 | [BP]+D8 | [BP]+D16 | DH | SI |
| 111 | | [BX] | [BX]+D8 | [BX]+D16 | BH | DI |

Định vị thanh ghi

- Toán tử toàn là thanh ghi, các lệnh này thực hiện nhanh vì không cần truy cập bộ nhớ.
- Ví dụ: `Mov ax,bx`



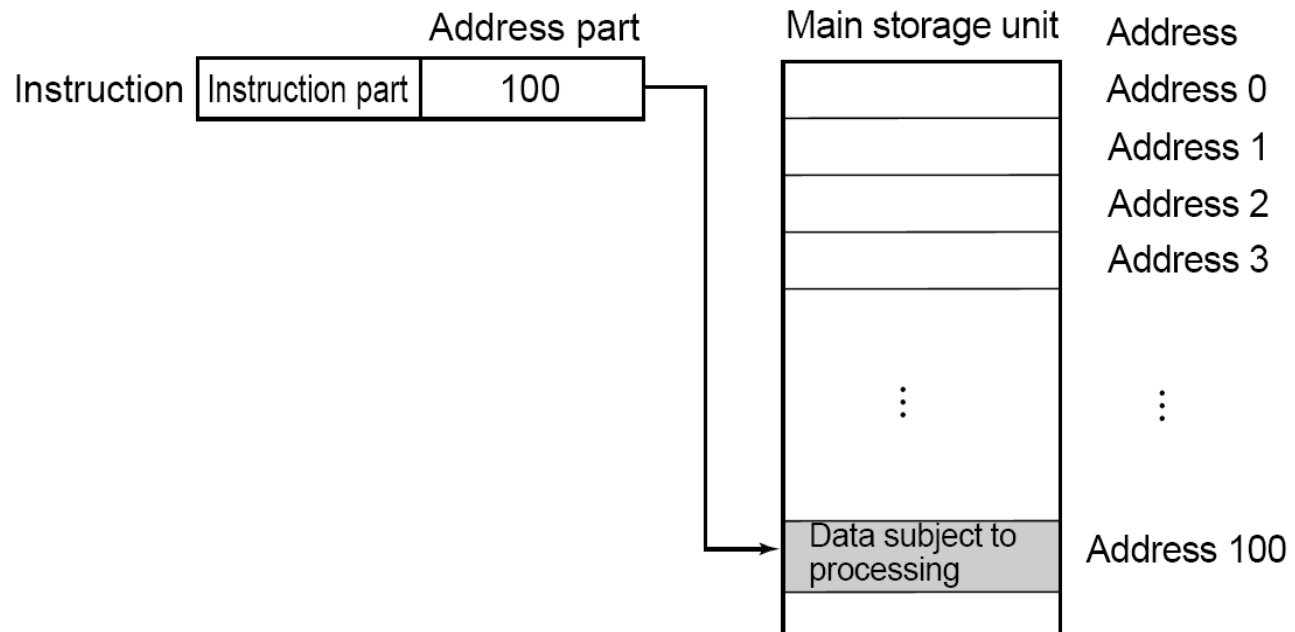
Định vị tức thì

- Dữ liệu được lưu ngay trong phần địa chỉ, không cần thiết phải thêm nhập vào bộ nhớ
- Ví dụ: `Mov ax,50h`



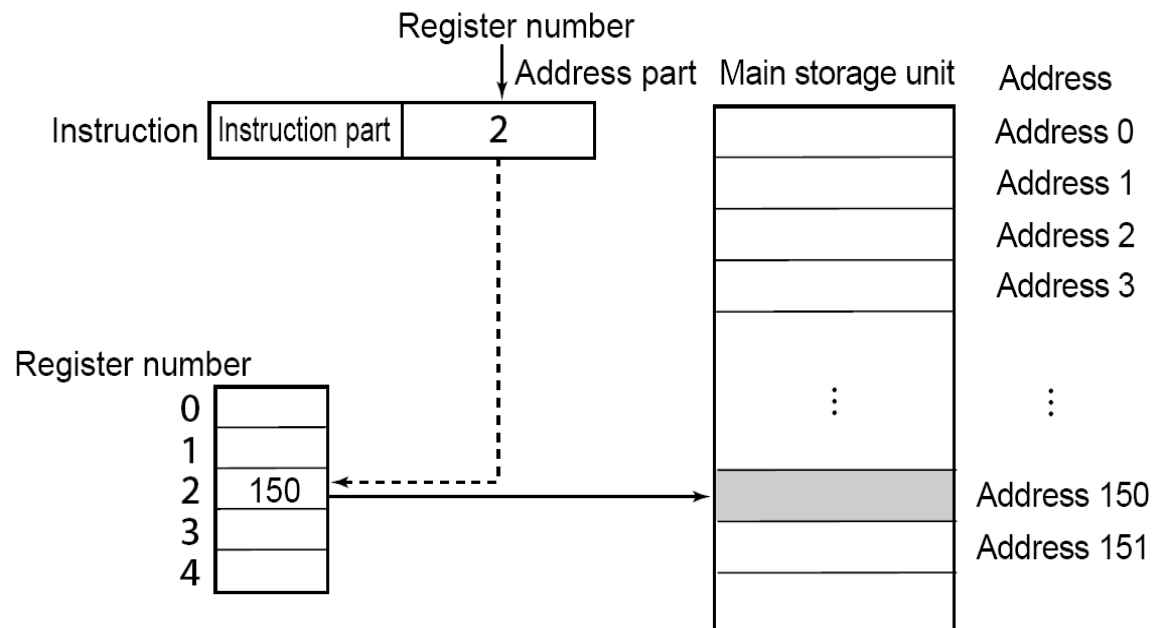
Định vị trực tiếp

- Chuyển trực tiếp nội dung giữa thanh ghi và ô nhớ
- Ví dụ: `Mov ah,[1234h]`; chuyển nội dung ô nhớ có địa chỉ `DS:[1234h]` vào thanh ghi `ah`.



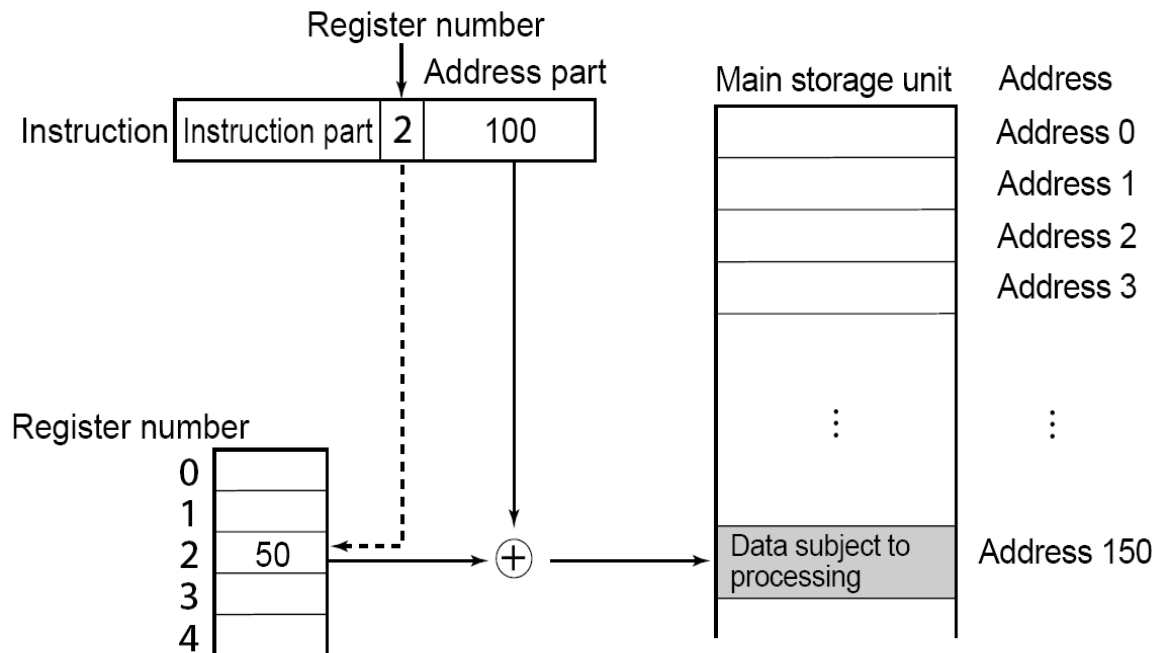
Định vị gián tiếp thanh ghi

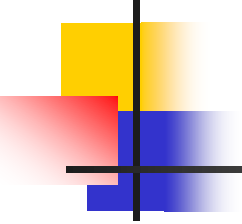
- Cũng chuyển trực tiếp nội dung giữa thanh ghi và ô nhớ, nhưng địa chỉ ô nhớ được chứa trong 1 thanh ghi nào đó.
- Ví dụ: `mov ah,[bx]`; chuyển nội dung ô nhớ có địa chỉ `DS:[bx]` vào thanh ghi `ah`.



Định vị tương đối cơ sở

- Dùng những thanh ghi cơ sở BX và BP và giá trị dịch chuyển.
- Ví dụ: `Mov ah, [bx] + 2`; Chuyển nội dung `DS:[bx + 2]` vào ah
- `Mov ah, [bx + 2]` ; cùng kết quả.



- 
- Định vị tương đối chỉ số: Giống định vị tương đối cơ sở những dùng những thanh ghi chỉ số DI, SI thay cho thanh ghi cơ sở.
 - Ví dụ: `Mov ah, [DI] + 2`; Chuyển nội dung `DS:[DI + 2]` vào `ah`
 - `Mov ah, [DI + 2]` ; cùng kết quả.
 - Định vị tương đối chỉ số cơ sở: Kết hợp 2 kiểu định vị tương đối cơ sở và định vị tương đối chỉ số.
 - Ví dụ: `Mov ah, [bx][DI] + 2`; Chuyển nội dung `DS:[bx+ DI + 2]` vào `ah`
 - `Mov ah, [bx+di+ 2]` ; cùng kết quả.
 - Khi dùng thanh ghi chỉ số, thanh ghi cơ sở, thanh ghi con trỏ thì các cặp địa chỉ `Segment:offset` sau là mặc định: `CS:IP`, `DS:SI`, `DS:DI`, `DX:BX`, `ES:DI`, `SS:SP`, `SS:BP`.

GIỚI THIỆU KHOA HỌC MÁY TÍNH



CHƯƠNG 3 – HỆ ĐIỀU HÀNH

NGUYỄN THANH TRUNG



Mục tiêu

- Giới thiệu tổng quan về hệ điều hành gồm: HĐH là gì, chức năng của HĐH, phân loại và những HĐH phổ biến,...
- Trình bày những kiến thức cơ bản về nguyên lý hoạt động cũng cấu trúc bên trong của HĐH.
- Giúp sinh viên biết được tầm quan trọng của HĐH cũng như việc lựa chọn HĐH cho phù hợp với mục tiêu sử dụng.



BỒ CỤC

- 3.1. Lịch sử các hệ điều hành
- 3.2. Tổ chức và hoạt động
- 3.3. Cơ chế bảo vệ thông tin của HĐH



Tài liệu tham khảo

- Chương 3, Computer Science
- Chương 3, bài giảng Giới thiệu Khoa học Máy tính.
- Tham khảo Bài giảng Hệ điều hành, Đại học Khoa học Tự nhiên (ebook)



3.1. Tổng quan hệ điều hành

- **Khái niệm**
- **Lịch sử phát triển**
- **Chức năng**
- **Phân loại**



3.1.1 Khái niệm

- **Hệ điều hành** là một chương trình chạy trên máy tính, dùng để điều hành, quản lý các thiết bị phần cứng và các tài nguyên phần mềm trên máy tính.
- Đóng vai trò trung gian trong việc giao tiếp giữa người sử dụng và phần cứng máy tính, cung cấp một môi trường cho phép người sử dụng phát triển và thực hiện các ứng dụng của họ một cách dễ dàng.



3.1.2. Lịch sử Hệ điều hành Theo các giai đoạn phát triển

- Thế hệ I: Chưa có HĐH, thao tác bằng tay trên bảng điều khiển.
- Thế hệ II: Hệ thống xử lý theo lô, gồm thực hiện các yêu cầu trên băng từ 1 cách tuần tự.
- Thế hệ III: Hệ điều hành đầu tiên gồm nhiều dòng lệnh hợp ngữ; HĐH đa chương (bộ nhớ chia thành nhiều phần chứa các công việc khác nhau); HĐH chia sẻ thời gian... trên máy mainframe, mini,...
- Thế hệ IV: Với sự ra đời máy tính cá nhân, nhiều HĐH: HĐH đa nhiệm, HĐH Mạng, HĐH phân tán ... ra đời



3.1.3.Chức năng

Quản lý chia sẻ tài nguyên

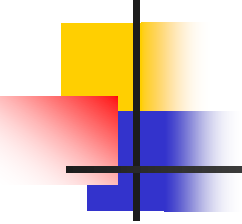
- Tài nguyên của hệ thống (CPU, bộ nhớ, thiết bị ngoại vi, ...) ... hệ điều hành cần phải có cơ chế và chiến lược thích hợp để quản lý việc phân phối tài nguyên.
- Ngoài yêu cầu dùng chung tài nguyên để tiết kiệm chi phí, người sử dụng còn cần phải chia sẻ thông tin (tài nguyên phần mềm) lẫn nhau, khi đó hệ điều hành cần đảm bảo việc truy xuất đến các tài nguyên này là hợp lệ, không xảy ra tranh chấp, thiếu nhất quán ...

Giả lập máy tính mở rộng

- Hệ điều hành làm ẩn đi các chi tiết phần cứng, người sử dụng được cung cấp 1 giao diện đơn giản, dễ hiểu và không phụ thuộc vào thiết bị cụ thể.
- Thực tế, ta có thể xem Hệ điều hành như là 1 hệ thống bao gồm nhiều máy tính trừu tượng xếp thành nhiều lớp chồng lên nhau, máy tính mức dưới phục vụ cho máy tính mức trên. Lớp trên cùng là giao diện trực quan nhất để chúng ta điều khiển.
- Ngoài ra có thể chia theo 4 chức năng sau : Quản lý tiến trình (process management), Quản lý bộ nhớ (memory management) Quản lý hệ thống lưu trữ (storage management) Giao tiếp với người dùng (user interaction)

Một số nhiệm vụ cụ thể

- Điều khiển và quản lý trực tiếp các phần cứng như bo mạch chủ, bo mạch đồ họa và bo mạch âm thanh,...
- Thực hiện một số thao tác cơ bản trong máy tính như các thao tác đọc, viết tập tin, quản lý hệ thống tập tin (*file system*) và các kho dữ liệu.
- Cung ứng một hệ thống giao diện sơ khai cho các ứng dụng thường là thông qua một hệ thống thư viện các hàm chuẩn để điều hành các phần cứng mà từ đó các ứng dụng có thể gọi tới.
- Cung ứng một hệ thống lệnh cơ bản để điều hành máy. Các lệnh này gọi là lệnh hệ thống (*system command*).
- Ngoài ra cũng cung cấp các dịch vụ cơ bản cho các phần mềm ứng dụng ...



3.1.4. Phân loại các HĐH Theo loại máy tính

- Hệ điều hành dành cho máy MainFrame
- Hệ điều hành dành cho máy Server
- Hệ điều hành dành cho máy nhiều CPU
- Hệ điều hành dành cho máy tính cá nhân (PC)
- Hệ điều hành dành cho máy PDA (Embedded OS - hệ điều hành nhúng)
- Hệ điều hành dành cho máy chuyên biệt
- Hệ điều hành dành cho thẻ chip (SmartCard)



Dưới góc độ số chương trình được sử dụng cùng lúc

- Hệ điều hành đơn nhiệm
- Hệ điều hành đa nhiệm

Dưới góc độ người dùng (truy xuất tài nguyên cùng lúc)

- Một người dùng
- Nhiều người dùng



Dưới góc độ hình thức xử lý

- Hệ thống xử lý theo lô
- Hệ thống chia sẻ
- Hệ thống song song
- Hệ thống phân tán
- Hệ thống xử lý thời gian thực

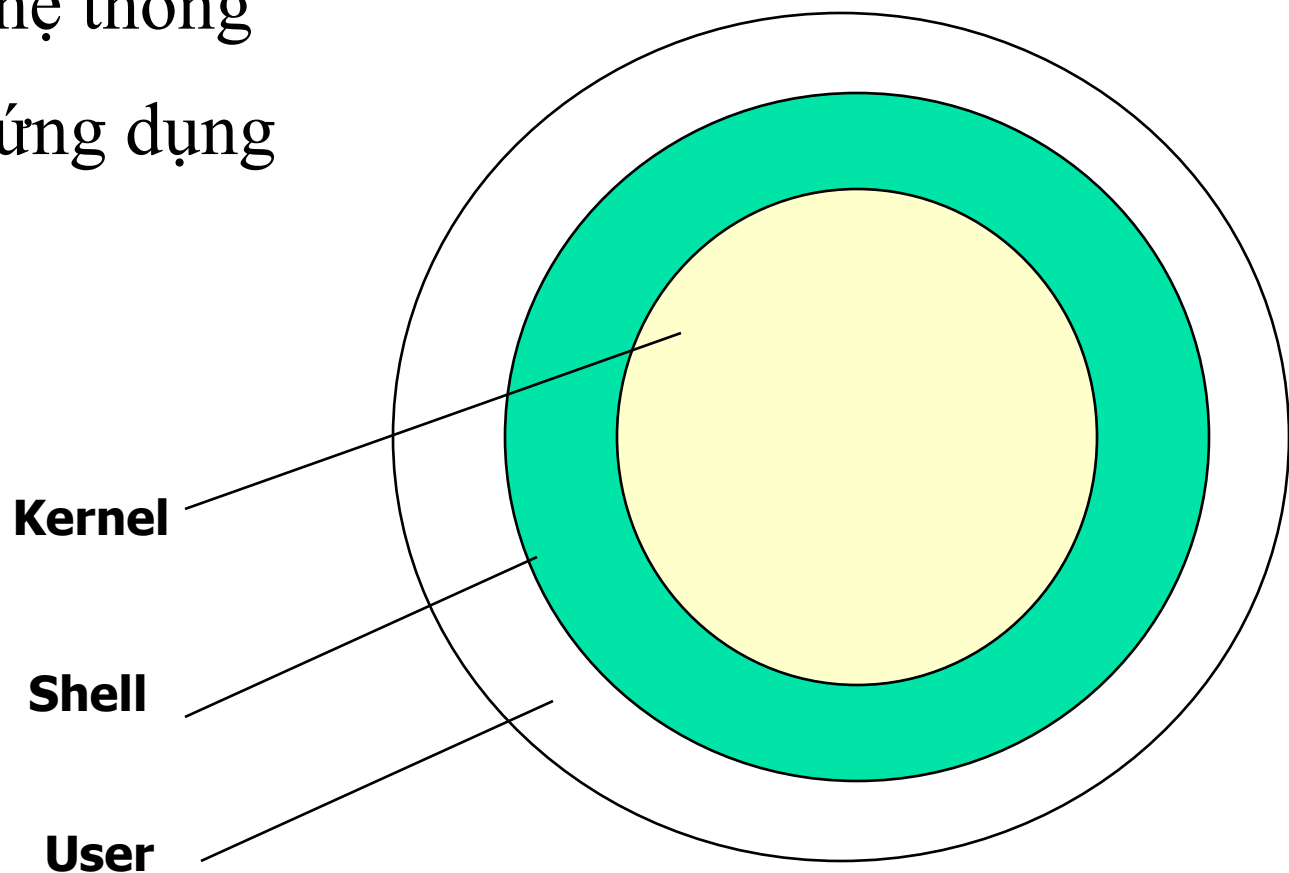


3.2.TỔ CHỨC VÀ HOẠT ĐỘNG

- Cấu trúc HĐH
- Phân chia thời gian
- Phân phối tài nguyên
- Các kiến trúc HĐH

Kiến trúc phân lớp của HĐH

- Phân loại phần mềm
 - Phần mềm hệ thống
 - Phần mềm ứng dụng
- Hệ điều hành
 - Shell
 - Kernel





shell

- GUI (Graphical User Interface)

EX: Windows Manager

- Thông dịch dòng lệnh



Kernel

- File Manager
- Device Driver
- Memory Manager
- Schedule
- Dispatcher



3.2.1. Cấu trúc HĐH

- Đơn vị xử lý câu lệnh (Command Processor)
- Bộ lập lịch (Scheduler)
- Đơn vị quản lý tập tin, ql tài nguyên
- Đơn vị ql bộ nhớ
- Bộ điều phối (Dispatcher)



Đơn vị xử lý câu lệnh

- Giúp HĐH giao tiếp với người dùng
- Thông qua các thiết bị nhập/xuất
- Khi nhận lệnh hợp lệ và tìm thấy yêu cầu xử lý của chương trình → yêu cầu bộ lập lịch sắp xếp các câu lệnh.



Đơn vị quản lý tập tin và đơn vị quản lý bộ nhớ

- Trước khi chương trình được bộ lập lịch chia lịch thực hiện thì bộ lập lịch cũng liên lạc với 2 dịch vụ khác đó là đơn vị quản lý tập tin và đơn vị quản lý bộ nhớ.
- đơn vị quản lý tập tin: Cung cấp thông tin liên quan đến dữ liệu trong khối lưu trữ; bảo vệ tập tin trong khối lưu trữ tránh việc truy cập bất hợp pháp.
- đơn vị quản lý bộ nhớ: Phân chia việc sử dụng bộ nhớ chính, cấp phát/thu hồi vùng nhớ.



Đơn vị quản lý tài nguyên

- Quản lý, phân phối các tài nguyên.
- Phối hợp với Bộ lập lịch phân chia tài nguyên cho các tiến trình để tránh việc tranh chấp tài nguyên...
- VD: Nếu 1 tiến trình xử lý cần 1 tài nguyên mà không thể cung cấp được thì đơn vị quản lý tài nguyên sẽ thông báo cho Bộ lập lịch biết. Còn ngược lại thì nó sẽ cấp tài nguyên cho tiến trình này → tiến trình được lập lịch và thực hiện.

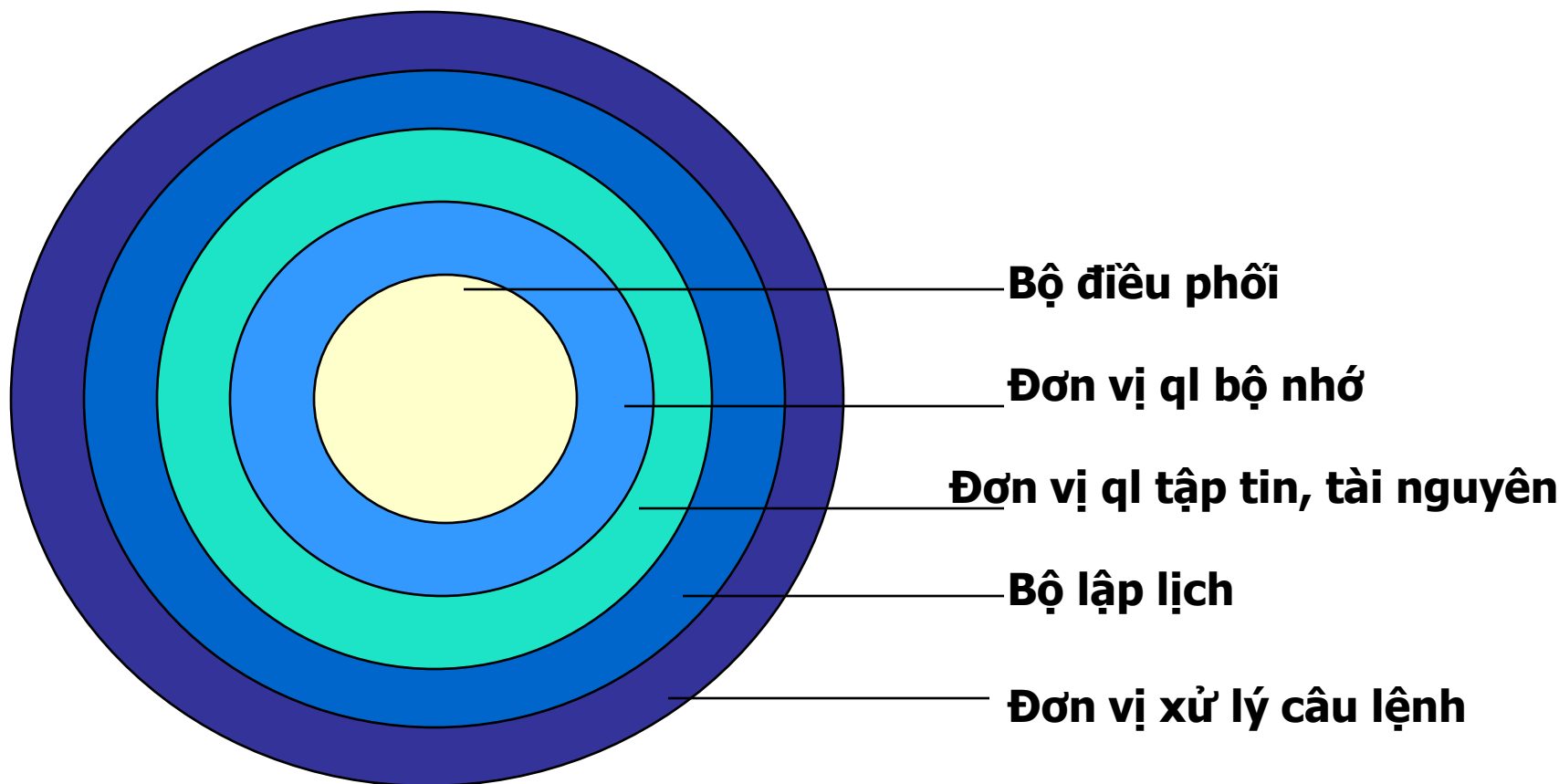


Bộ lập lịch

- Sắp xếp việc thực hiện các chương trình (tiến trình).
- Vd: đối với hệ thống xử lý theo lô thì chương trình sẽ được đưa vào hàng đợi theo độ ưu tiên của chương trình.
- Đối với hệ thống đa nhiệm thì nó yêu cầu chương trình phối hợp với các hoạt động khác cùng phân chia thời gian.
- Các hoạt động khác cùng phân chia thời gian này thực chất là các tiến trình.

bộ điều phối (dispatcher)

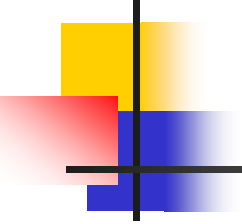
- Điều phối các tiến trình: Lựa chọn các tiến trình đang quan tâm trong số các tiến trình đã được lập lịch...



3.2.2. Các nguyên lý cơ bản của phân chia thời gian

- Điều khiển ngắt
- Cấp phát thời gian: Cho 2 tiến trình được lập lịch là A và B, giả sử A thực hiện được 1 khoảng thời gian thì bị ngắt bởi bộ điều phối \rightarrow thực hiện B, sau 1 khoảng thời gian B bị ngắt \rightarrow thực hiện A,....

Như vậy, bằng cách điều khiển ngắt mà bộ điều phối có thể cho phép thực hiện xoay vòng nhiều tiến trình trong 1 khoảng thời gian gọi là kỹ thuật xoay vòng.

- 
- VD1: trong 1 máy tính thực hiện xoay vòng khoảng 50 tiến trình, khoảng thời gian thực hiện mỗi tiến trình khoảng 10 – 100 miligiây... → cảm giác nhiều tiến trình được thực hiện đồng thời.
 - VD2: Một máy tính đa người dùng, thì HĐH tạo ra mỗi người dùng 1 máy ảo được tạo ra theo cách phân chia thời gian như vậy.

Ngoài ra các tiến trình có thể có sự ưu tiên khác nhau.



3.2.3. Phân phối tài nguyên

a. Điều phối tài nguyên

- Tài nguyên có 2 loại: phân chia được và không phân chia được. (VD)
- Nguyên tắc:
 - Tạo Bộ phân phối và thu hồi các tài nguyên
 - Các tiến trình trong môi trường đa xử lý không được phép liên lạc trực tiếp với các ngoại vi mà phải thông qua HĐH
 - Gán cờ hiệu cho tài nguyên (0/1)



b. Deadlock

- Deadlock là gì? Đợi 1 sự kiện không bao giờ xảy ra.
- Ví dụ:
- Phân tích: Deadlock xuất hiện khi thoả cả 3 đk sau:
 1. Có sự cạnh tranh ở các tài nguyên không thể chia.
 2. Những tài nguyên bị yêu cầu trên từng phần cơ bản; để kết thúc và nhận lại 1 số tài nguyên từ 1 tiến trình thì phải cung cấp cho nó 1 số tài nguyên khác.
 3. Một lần 1 tài nguyên bị phân chia, nó không thể yêu cầu sử dụng lại.



Xử lý Deadlock

- Bỏ đi 1 tiến trình nào đó gây ra vấn đề trên.
- Chuyển các tài nguyên không phân chia → có thể phân chia. VD: kỹ thuật Spooling khi in dữ liệu.



3.3. Quản lý tiến trình

Các máy tính ngày nay đều có khả năng xử lý đồng thời nhiều chương trình, nhưng thiết bị phần cứng chỉ có 1 cho nên HĐH thiết kế mô hình song song giả lập để xử lý 1 lúc nhiều chương trình → mỗi chương trình này → Tiến trình.



Tiến trình (Process)

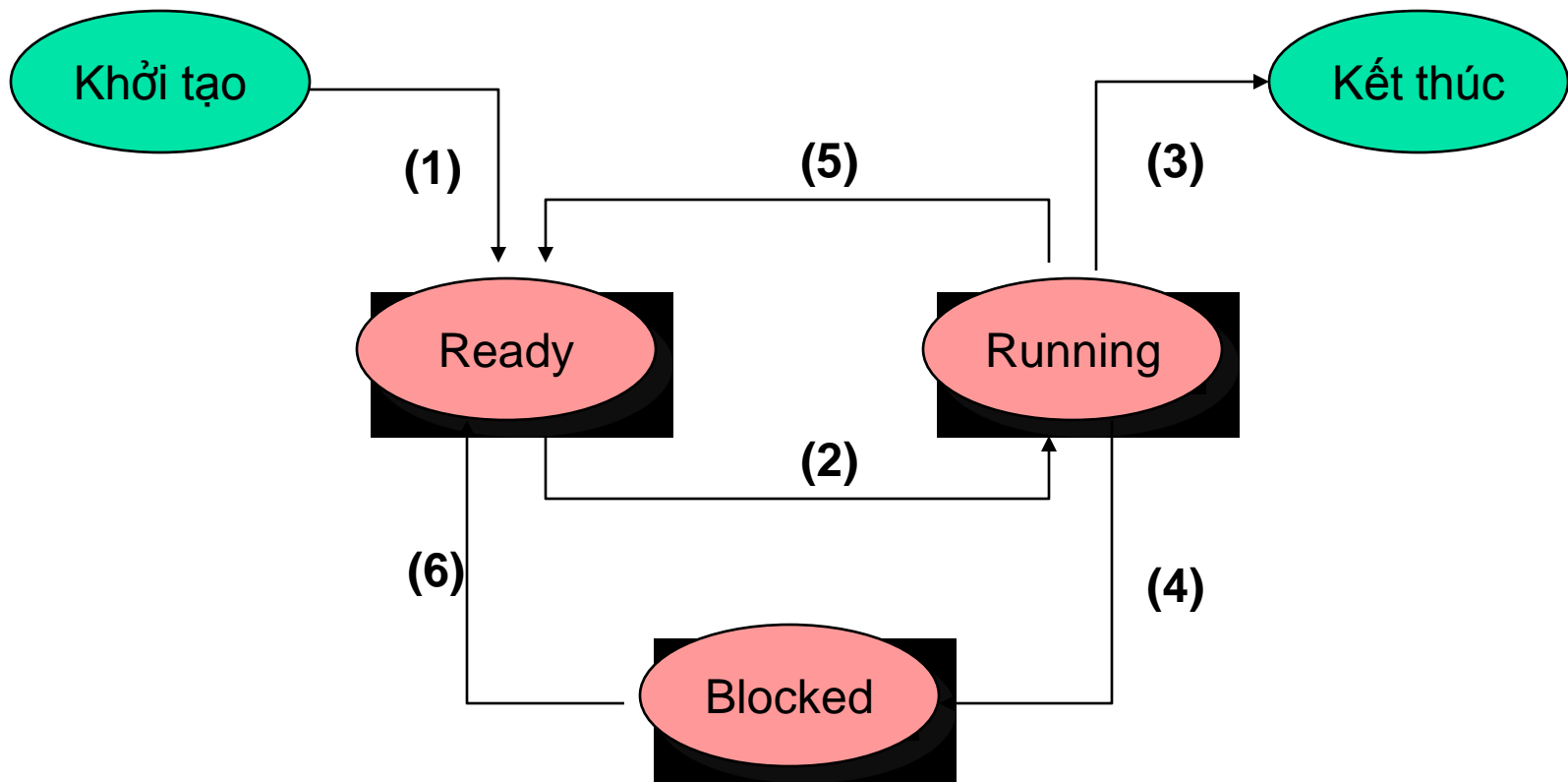
- Là ctrình đang xử lý
- Sở hữu 1 con trỏ lệnh, tập các thanh ghi, và các biến.
- Cần 1 số tài nguyên: CPU, memory, các tập tin, và thiết bị xuất/nhập.
- Khác với chương trình-xem như thực thể thụ động thì tiến trình xem như thực thể tích cực, có thể tương tác, thay đổi trạng thái...



Các trạng thái của ttrình

- **Khởi tạo:** ttrình được tạo lập
- **Ready:** ttrình chờ được cấp phát tài nguyên để xử lý
- **Running:** các lệnh của ttrình đang được xử lý.
- **Blocked:** ttrình chờ cấp phát tài nguyên, hay chờ 1 sự kiện xảy ra.
- **Kết thúc:** ttrình hoàn tất xử lý.

Chuyển trạng thái của ttrình





Chuyển đổi trạng thái của trình

- (1): trình mới tạo được đưa vào hệ thống.
- (2): Bộ điều phối cấp phát cho trình 1 khoảng thời gian sử dụng CPU.
- (3): trình kết thúc.
- (4): trình yêu tài nguyên nhưng chưa được đáp ứng; hoặc phải chờ 1 sự kiện hay thao tác xuất/nhập.
- (5): Bộ điều phối chọn trình khác để cho xử lý.
- (6): tài nguyên yêu cầu đã sẵn sàng; hoặc sự kiện hay thao tác xuất/nhập đã hoàn tất.



Điều phối tiến trình

- Chia sẻ thời gian – để chuyển đổi CPU qua lại giữa các ttrình.
- Bộ phân phối (dispatcher)
 - Lựa chọn ttrình để xử lý tiếp theo.
 - Chuyển ngữ cảnh (context).
- Mục tiêu điều phối (scheduling)
 - Sự công bằng (fairness)
 - Hiệu quả (efficiency)
 - Thời gian đáp ứng hợp lý (response time)
 - Thời gian lưu lại hệ thống (turnaround time)
 - Thông lượng tối đa (throughput)



Cơ chế điều phối của HĐH

- Điều phối độc quyền (preemptive)
 - Trình nhận được CPU sẽ có quyền độc chiếm CPU đến khi hoàn tất xử lý hoặc tự nguyện giải phóng CPU.
 - Quyết định điều phối xảy ra khi:
 - Trình chuyển từ tthái running sang blocked
 - Trình kết thúc

Cơ chế điều phối của HĐH (tt)

- Điều phối không độc quyền (non-preemptive)
 - Trình nhận được CPU vẫn được sử dụng CPU đến khi hoàn tất xử lý hoặc tự nguyện giải phóng CPU.
 - Nhưng trình khác có độ ưu tiên có thể dành quyền sử dụng CPU.
 - Quyết định điều phối xảy ra khi:
 - Trình chuyển từ tthái running sang blocked
 - Trình chuyển từ running sang ready
 - Trình chuyển từ blocked sang ready
 - Trình kết thúc

Interval timer/ interrupting clock

- HĐH sử dụng một bộ đếm thời gian / đồng hồ ngắt giờ
- Khoảng thời gian t thích hợp ứng với 1 lượt cấp phát CPU cho 1 trình.
- Sau khoảng thời gian t xảy ra ngắt báo hiệu hết thời gian xử lý của trình hiện hành. Khi đó, HĐH được kích hoạt, và bộ điều phối sẽ quyết định trình nào sẽ được cấp phát CPU trong lượt kế tiếp.



Độ ưu tiên của tiến trình

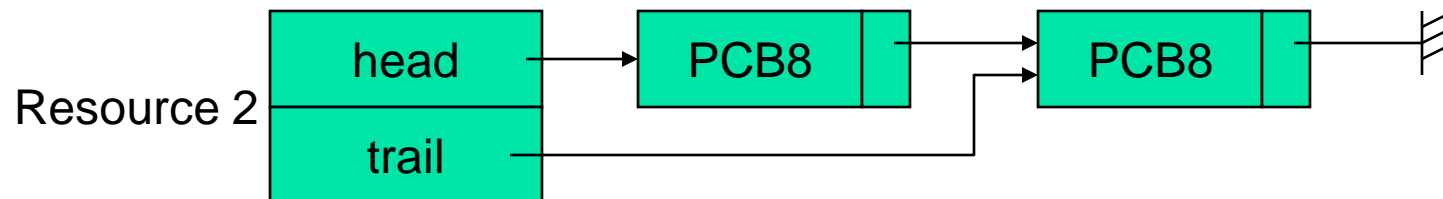
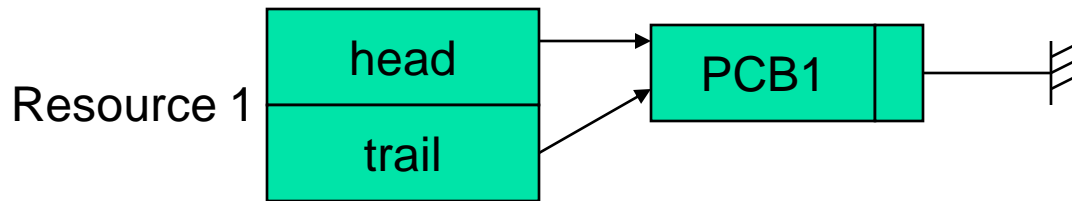
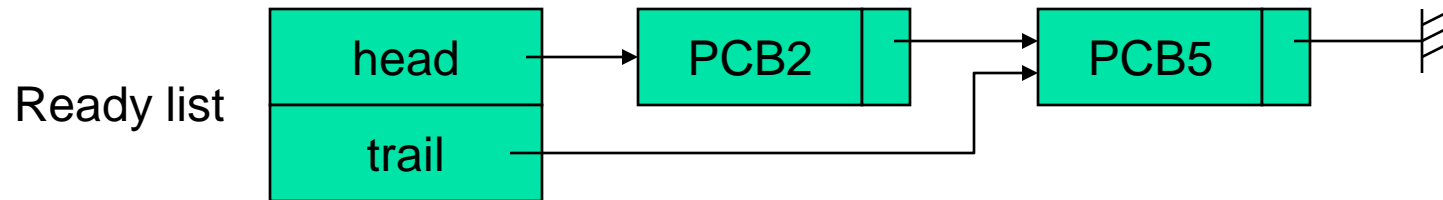
- Được gán bởi hệ thống hay gán tường minh bởi user.
- Độ ưu tiên tĩnh: không thay đổi bất kể sự biến động của môi trường.
- Độ ưu tiên động: thay đổi theo thời gian và môi trường xử lý



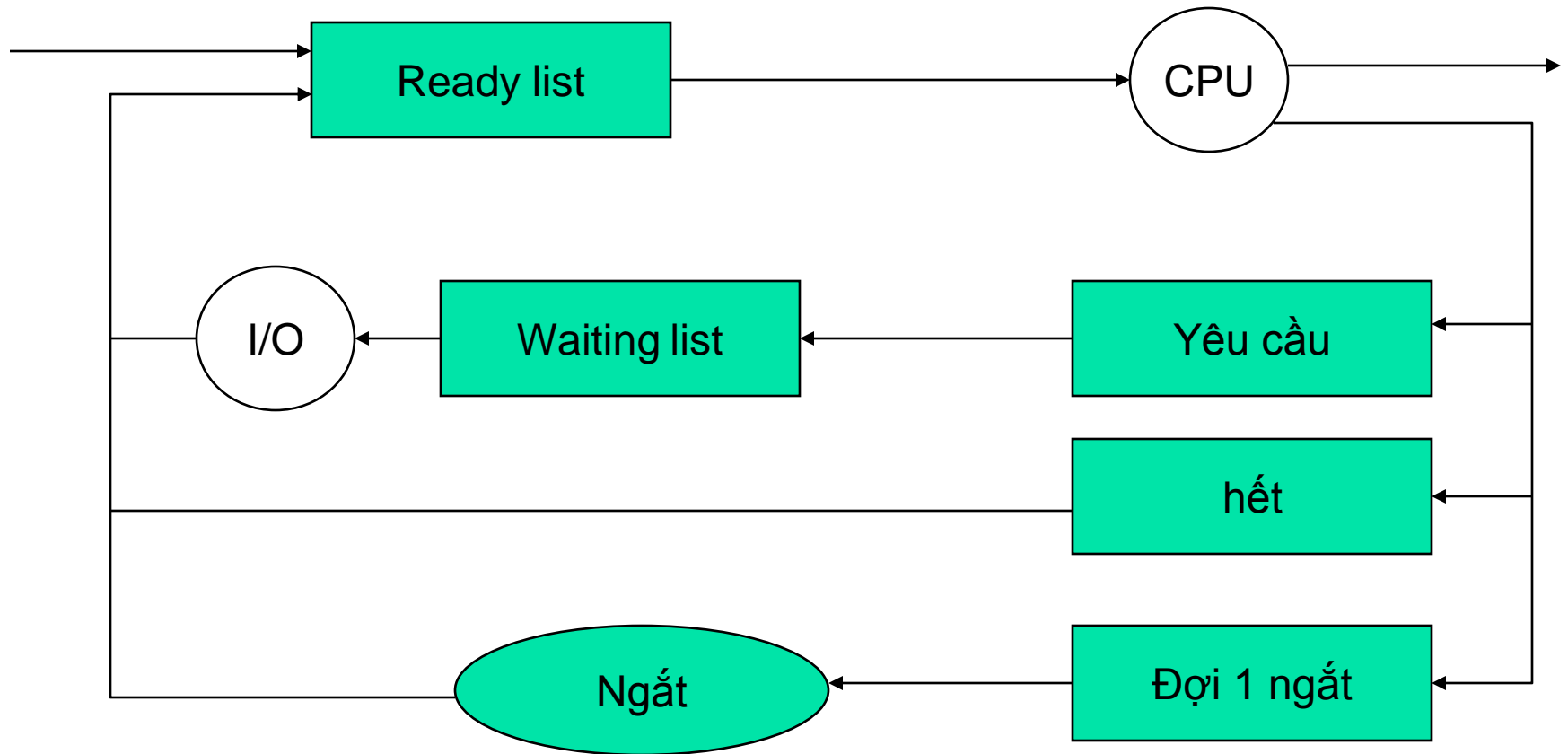
Tổ chức điều phối

- Danh sách sẵn sàng (**ready list**) chứa các trình đã được nạp vào bộ nhớ chính và ở trạng thái sẵn sàng (ready) tiếp nhận CPU.
- Danh sách chờ đợi (**waiting list**): chứa các trình chờ đợi 1 thao tác xuất nhập hoàn tất, yêu cầu tài nguyên chưa được thoả mãn, yêu cầu tạm dừng
- Mỗi tài nguyên (tạm ngoại vi) có danh sách chờ đợi riêng bao gồm các trình đang chờ được cấp tài nguyên đó.

Các danh sách điều phối



Sơ đồ chuyển đổi giữa các danh sách điều phối





Các cấp độ điều phối

- Điều phối theo tác vụ (job scheduling)
 - Quyết định lựa chọn tác vụ nào đưa vào hệ thống và nạp những trình của tác vụ đó vào bộ nhớ chính.
 - Tạo lập 1 trình hay có 1 trình kết thúc: kích hoạt chức năng điều phối tác vụ mới.
 - Chức năng điều phối tác vụ có tần xuất hoạt động thấp do tính đa chương tương đối ổn định.
 - Cần phân biệt trình theo hướng xuất/nhập hay hướng xử lý.
 - Cân bằng hoạt động CPU và tị ngoại vi: pha trộn hợp lý giữa các trình hướng xuất/nhập và hướng xử lý.



Các cấp độ điều phối (tt)

- Điều phối theo tiến trình (process scheduling)
 - Quyết định chọn 1 trình ở trạng thái sẵn sàng.
 - Có tần xuất hoạt động cao do bị ngắt (đồng hồ, bị ngoại vi...)
- Cấp độ điều phối trung gian: Kết hợp 2 cấp độ điều phối theo tác vụ và theo tiến trình



Các chiến lược điều phối

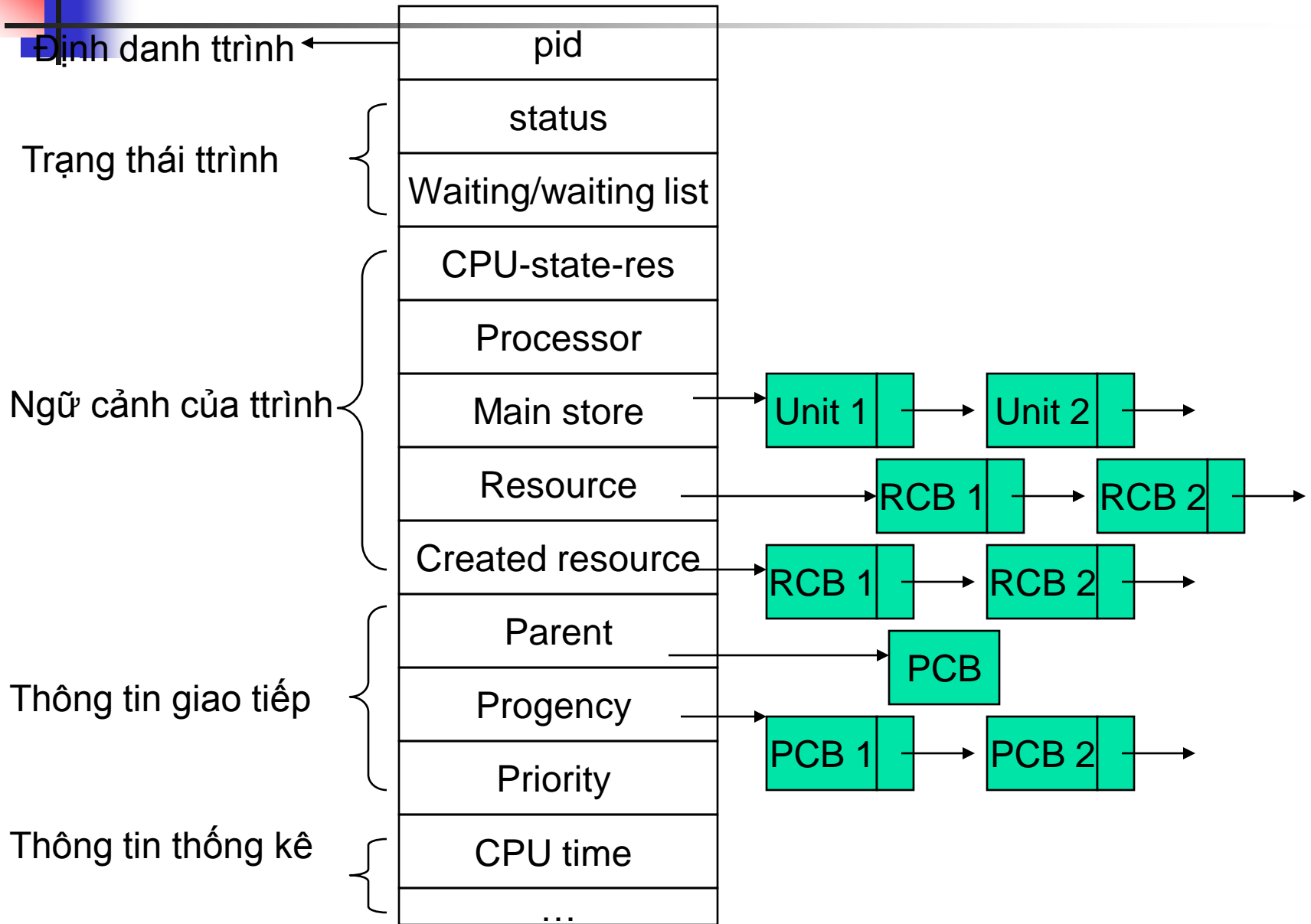
- Chiến lược FIFO
- Chiến lược phân phối xoay vòng (round robin)
- Chiến lược điều phối theo độ ưu tiên
- Chiến lược theo công việc ngắn nhất (shortest-Job-First) SJF
- Chiến lược điều phối theo nhiều mức độ ưu tiên ...



Khởi điều khiển tiến trình

- Process Control Block –PCB
- HĐH quản lý tiến trình thông qua PCB
- PCB là 1 cấu trúc gồm nhiều trường:
 - Định danh tiến trình: Pid
 - Trạng thái tiến trình:
 - Ngưỡng cảnh
 - Thông tin giao tiếp
 - Thông tin thống kê

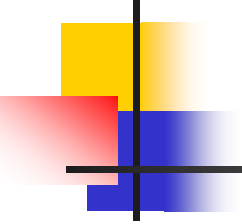
Khối điều khiển ttrình





Giải thích

- Định danh tiến trình giúp phân biệt các tiến trình khác nhau.
- Trạng thái tiến trình là tình trạng hiện tại của tiến trình.
- Ngữ cảnh tiến trình chứa các thông tin về tài nguyên của tiến trình gồm:
 - Trạng thái CPU: Registers, IP
 - CPU trong trường hợp máy có nhiều CPU
 - Memory
 - Tài nguyên sử dụng: danh sách các tài nguyên hệ thống mà tiến trình sử dụng
 - Tài nguyên tạo lập: danh sách các tài nguyên được tiến trình tạo lập

- 
- Thông tin giao tiếp: Thông tin về quan hệ giữa tiến trình này với các tiến trình khác
 - Tiến trình cha: Tạo ra tiến trình này
 - Tiến trình con: Do nó tạo ra
 - Độ ưu tiên: Để bộ điều phối xác định chế độ ưu tiên
 - Thông tin thống kê: Thông tin thống kê hoạt động của tiến trình (thời gian sử dụng CPU, thời gian chờ, ...)

Tiểu trình (thread) & Tiến trình (Process)

- Một tiến trình: Có Không gian địa chỉ riêng, chỉ 1 dòng xử lý
- Các tiến trình độc lập liên lạc với nhau thông qua cơ chế do HĐH cung cấp.

Mong muốn nhiều dòng xử lý chia sẻ 1 không gian địa chỉ và các dòng xử lý hoạt động song song → Tiểu trình:

- 1 đơn vị xử lý cơ bản
- Sở hữu 1 con trỏ lệnh, tập các thanh ghi, 1 vùng nhớ stack riêng
- Có các trạng thái như tiến trình thật.
- Các tiểu trình trong cùng tiến trình chia sẻ không gian địa chỉ chung...



Phân bố thông tin lưu trữ

- Tiến trình
 - Không gian địa chỉ
 - Tài nguyên toàn cục
 - Các thông tin thống kê
- Tiêu trình
 - Con trỏ lệnh + các thanh ghi, stack
 - Tài nguyên cục bộ



3.4. Một số Kiến trúc Hệ điều hành

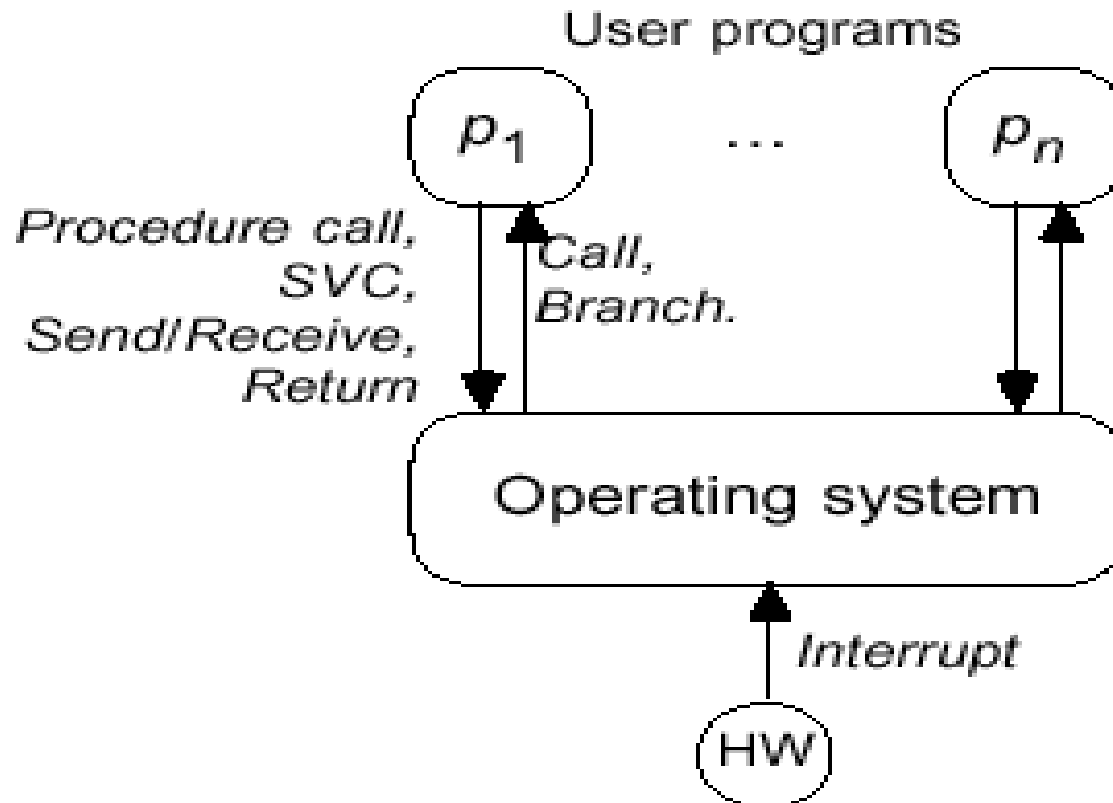
- Đơn giản (Monolithic)
- Hạt nhân (Kernel)
- Phân lớp (Layered)
- Máy ảo (Virtual Machine)
- Hướng đối tượng (OOOS)
- Exokernel



Monolithic

- OS = Thư viện tiện ích
- Có thể tổ chức thành nhiều module : CPU scheduling, Mem Management, Device management ... nhưng chỉ có 1 trong những module này hoạt động tại một thời điểm
- Đơn nhiệm
- Quyền điều khiển được chuyển đổi thông qua lời gọi hàm
- Khi tầm vóc phát triển hệ thống trở nên thiếu tin cậy.
- Ví dụ : MS-DOS, Ultrix (mature Unix)

Monolithic

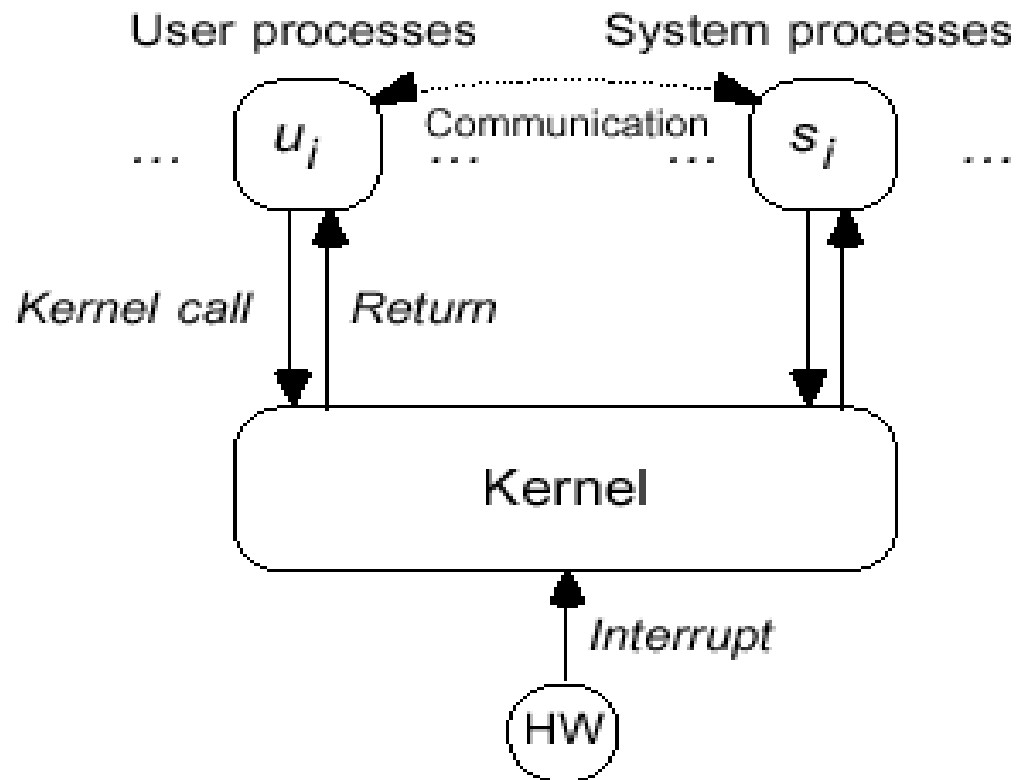




Kernel

- OS = Kernel + System processes
- Kernel được bảo vệ
- Đa nhiệm
- Kernel chịu trách nhiệm phân chia thời gian sử dụng CPU, Giao tiếp giữa các tiến trình
- Chỉ có 2 mức kernel/non-kernel =>kernel lớn, thiếu tin cậy như trước
- Định nghĩa cứng các giao tiếp với ứng dụng trong kernel
- Ví dụ : Windows NT

Kernel





Layered

- OS = các lớp trừu tượng hoá một tác vụ quản lý
- Lớp trên được sử dụng các hàm xử lý tài nguyên thuộc tác vụ do lớp dưới cung cấp
- Khó xác định được các lớp xử lý rạch ròi, thứ tự lớp ?
- Xếp lớp theo hàm xử lý , thay vì tác vụ
- Ví dụ : THE , MULTICS

Layered

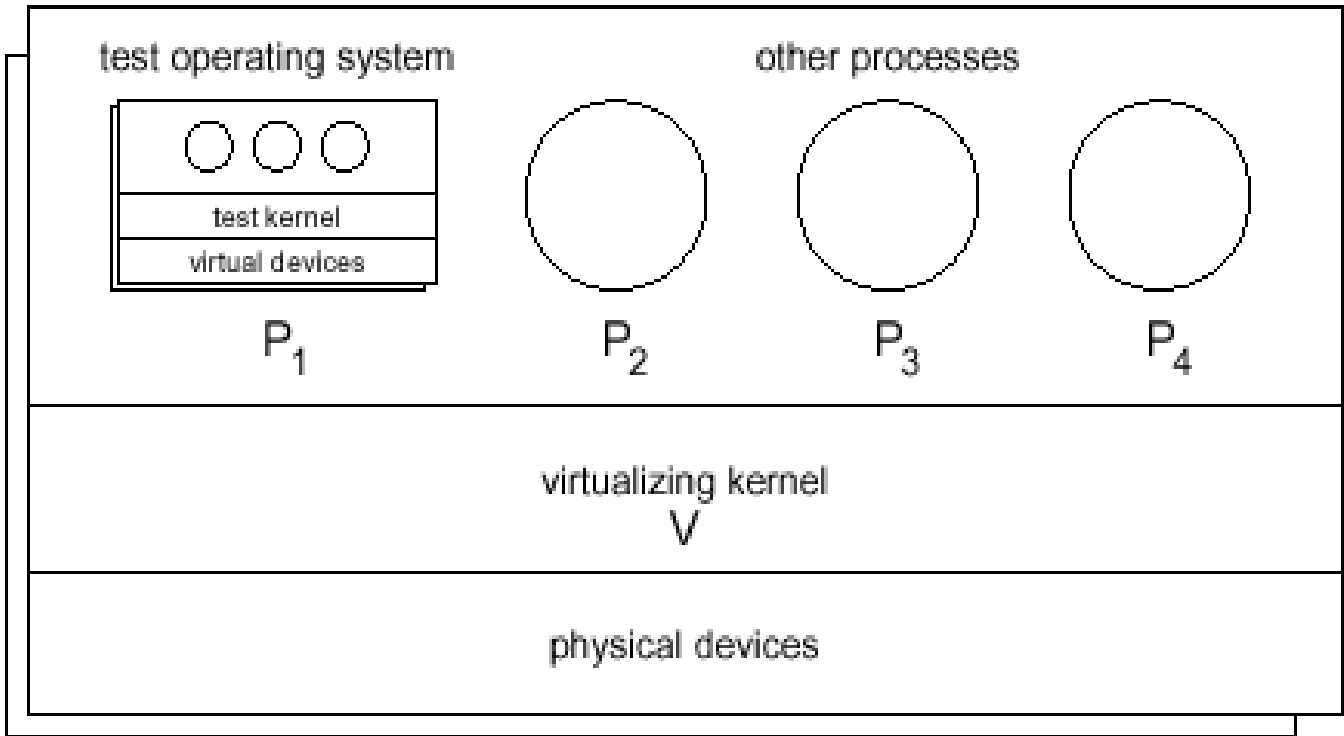
| | | | | |
|---|--|---|--|-------------------------------|
| <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; border-radius: 50%; padding: 5px; text-align: center;">User₁</div> <div style="border: 1px solid black; border-radius: 50%; padding: 5px; text-align: center;">User₂</div> <div style="text-align: center;">...</div> <div style="border: 1px solid black; border-radius: 50%; padding: 5px; text-align: center;">User_n</div> </div> | | | | L4: Indep. user processes |
| | | | <div style="border: 1px solid black; border-radius: 50%; padding: 5px; text-align: center;">I/O device processes</div> | L3: Virtual I/O devices |
| | | <div style="border: 1px solid black; border-radius: 50%; padding: 5px; text-align: center;">Command Interpreter</div> | | L2: Virtual Operator Consoles |
| | <div style="border: 1px solid black; border-radius: 50%; padding: 5px; text-align: center;">Segment Controller</div> | | | L1: Virtual Segmented Memory |
| CPU alloc., synchroniz'tn. | | | | L0: Virtual CPUs |
| CPU | Main mem., secondary storage | Operator's console | I/O devices | Actual hardware |

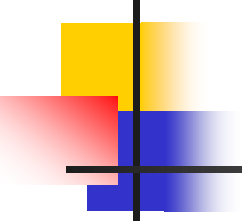


Virtual Machine

- OS = Virtualizing kernel + virtual machines
- Virtual machine = physical hardware
- Virtualizing kernel tạo ra nhiều VM trên 1 máy tính.
- Process interface = hardware interface
- Ưu điểm :
 - Môi trường thuận lợi cho sự tương thích (compatibility)
 - Tăng tính an toàn hệ thống do cung cấp các VM độc lập.
 - Dễ phát triển các HĐH đơn nhiệm cho mỗi VM
- Khuyết điểm:
 - Phức tạp cho việc giả lập (transput, add translation...)
- Ví dụ : CMS(conversational Monitor System) trên VM/370 (hỗ trợ hardware)

Virtual Machine

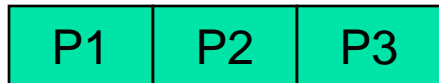


- 
-
- **Nói thêm về các chiến lược điều phối tiến trình (đọc thêm)**

Chiến lược FIFO

- CPU cấp cho tiến trình đầu tiên trong ready list.

| Tiến trình | Thời điểm vào | Thời gian xử lý |
|------------|---------------|-----------------|
| P1 | 0 | 24 |
| P2 | 1 | 3 |
| P3 | 2 | 3 |



0 24 27 30

Thời gian chờ để được xử lý:

$$P1 \rightarrow 0; P2 \rightarrow 24 - 1 = 23; P3 = 24 + 3 - 2$$

Thời gian chờ trung bình: $(0+23+25)/3 = 16$ ms



Chiến lược FIFO (tt)

- Thảo luận:
 - Thời gian chờ trung bình không đạt cực tiểu
 - Có thể xảy ra hiện tượng tích lũy thời gian chờ
 - Không phù hợp với HĐH phân chia theo thời gian



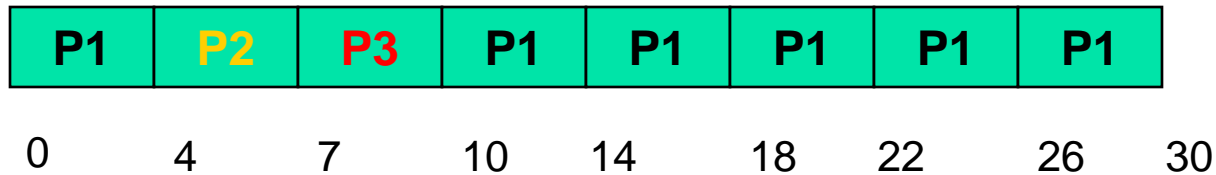
Chiến lược round robin

- Bộ điều phối lần lượt cấp phát cho từng tiến trình trong ready list 1 khoảng thời gian sử dụng CPU là *quantum*.
- Hết thời gian quantum mà trình chưa hoàn tất, trình đưa trở lại vào cuối ready list.

Chiến lược round robin (tt)

| Tiến trình | Thời điểm vào | Thời gian xử lý |
|------------|---------------|-----------------|
| P1 | 0 | 24 |
| P2 | 1 | 3 |
| P3 | 2 | 3 |

Quantum= 4ms



- Thời gian chờ trung bình:

$$(0+6+3+5)/3=4,66\text{ms}$$



Chiến lược round robin (tt)

- n trình trong ready list; Quantum q
- Mỗi trình không đợi quá $(n-1)q$ đvị thời gian để đến lượt nhận CPU.
- Thao luận:
 - Độ dài quantum ?
 - bé: phát sinh nhiều chuyển đổi giữa các trình
 - Lớn: tăng thời gian phản hồi, giảm khả năng tương tác.



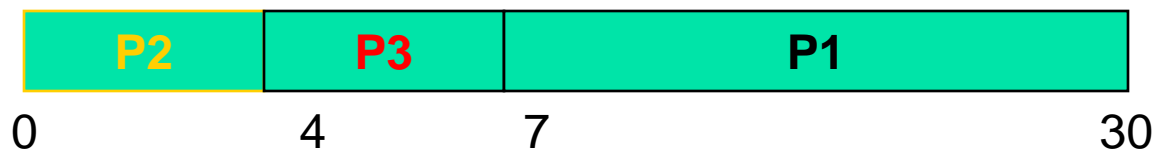
Chiến lược điều phối theo độ ưu tiên

- Gán độ ưu tiên cho mỗi ttrình
- Ttrình có độ ưu tiên cao nhất sẽ được chọn.
- Độ ưu tiên xác định
 - đ/nghĩa nội tại (e.g. giới hạn thời gian, nhu cầu bộ nhớ,...)
 - nhờ yếu tố bên ngoài (e.g. tầm quan trọng của ttrình, loại user sở hữu ttrình)
- Có thể hoạt động theo nguyên tắc độc quyền hay không độc quyền.

Chiến lược điều phối theo độ ưu tiên (tt)

- Khi ttrình được đưa vào ready list
 - Trong chế độ không độc quyền: so sánh độ ưu tiên với ttrình đang được xử lý hiện hành nếu độ ưu tiên cao hơn -> cấp phát CPU cho ttrình mới.
 - Trong chế độ độc quyền: chèn ttrình mới vào ready list.

| Tiến trình | độ ưu tiên | Thời gian xử lý |
|------------|------------|-----------------|
| P1 | 3 | 24 |
| P2 | 1 | 3 |
| P3 | 2 | 3 |



Giải thuật độ quyền

Chiến lược điều phối theo độ ưu tiên (tt)

- Thảo luận:
 - Trình trạng đói CPU (starvation): trình có độ ưu tiên thấp chờ đợi vô thời hạn.
 - Khắc phục giảm độ ưu tiên của các trình có độ ưu tiên cao sau mỗi ngắt đồng hồ.

độ ưu tiên của trình giảm xuống thấp hơn trình có độ ưu tiên cao thứ nhì -> chuyển đổi quyền sử dụng CPU (“lão hoá” (aging) trình).

Chiến lược theo công việc ngắn nhất (shortest-Job-First) SJF

- Giải thuật đặc biệt của giải thuật điều phối theo độ ưu tiên
- Độ ưu tiên $p = 1/t$; t : thời gian xử lý yêu cầu
- Giải thuật này có thể độc quyền hay không độc quyền.
 - Không độc quyền: dừng trình hiện hành, khi có 1 trình mới có độ ưu tiên cao hơn vào ready list.
 - Độc quyền: trình hiện hành tiếp tục xử lý.

Chiến lược theo công việc ngắn nhất (shortest-Job-First) SJF (tt)

| Tiến trình | Thời gian xử lý |
|------------|-----------------|
| P1 | 6 |
| P2 | 8 |
| P3 | 7 |
| P4 | 3 |

SJF độc quyền: P4 ->P1->P3->P2

Thời gian chờ tbình: $(3+16+9+0)/3 = 7ms$

■ Thảo luận

- Thời gian chờ trung bình đạt cực tiểu
- Làm sao biết thời gian yêu cầu còn lại xử lý ?

Chiến lược theo công việc ngắn nhất (shortest-Job-First) SJF (tt)

- Dự đoán thời gian xử lý còn lại:



0

Chiến lược điều phối theo nhiều mức độ ưu tiên

- Phân lớp các trình tùy theo độ ưu tiên của chúng. -> cách thức điều phối thích hợp cho từng nhóm.
- Ready list được phân thành các list riêng biệt theo cấp độ ưu tiên.
- Trình trong list ở cấp độ ưu tiên i chỉ được cấp phát CPU khi các list ở cấp ưu tiên lớn hơn i đã trống.

Chiến lược điều phối theo nhiều mức độ ưu tiên (tt)

- 1 trình gán vĩnh viễn cho 1 list ở cấp ưu tiên i ; trình không chuyển giữa các lists
 - => Giảm chi điều phối, thiếu linh động và có thể dẫn đến “đói CPU”.
- Xây dựng giải thuật điều phối nhiều cấp ưu tiên và xoay vòng.
 - 1 trình từ list có độ ưu tiên cao xuống list có độ ưu tiên thấp hơn sau mỗi lần sử dụng CPU. 1 trình từ list có độ ưu tiên thấp -> cao hơn.
 - Các tham số liên quan.



Tóm tắt chương & câu hỏi

- Chức năng ?
- (Cấu trúc) Các thành phần chính ?
- Việc điều phối tiến trình ? Các chiến lược ?



Các chủ đề thảo luận thêm

- - Cơ chế quản lý bộ nhớ của Hệ điều hành.
- - Cơ chế quản lý nhập xuất của HĐH.



GIỚI THIỆU KHOA HỌC MÁY TÍNH

CHƯƠNG 4- MẠNG MÁY TÍNH

NGUYỄN THANH TRUNG



MỤC TIÊU

- Cung cấp những kiến thức nền tảng về mạng máy tính, Internet gồm: các loại mạng, mô hình OSI, giao thức TCP/IP, cách thức truyền thông tin trên mạng...
- Giới thiệu một số hình thức tấn công mạng và các giải pháp bảo vệ mạng.
- Giúp sinh viên thấy được vai trò của mạng, internet trong xu thế hội nhập, những ảnh hưởng của mạng, internet vào cuộc sống.



BỒ CỤC

- 4.1. Cơ bản về mạng máy tính
- 4.2. Internet
- 4.3. Các hình thức tấn công và các giải pháp bảo vệ mạng



Tài liệu tham khảo

- Chương 4, Computer Science
- -Chương 4 bài giảng Giới thiệu Khoa học Máy tính.
- - Tham khảo tài liệu Mạng Máy Tính, Phạm Thế Quế, 2006 (ebook)



4.1. Cơ bản về mạng máy tính

- 1.1 - Mạng máy tính và ứng dụng trong đời sống.
- 1.2 - Phân loại mạng (network taxonomy).
- 1.3 - Giao thức mạng (software).
- 1.4 - Các mô hình tham chiếu (reference models).
- 1.5 - Chuẩn mạng máy tính (network standards).
- 1.6 - Hệ điều hành trong môi trường mạng.

1.1 Mạng máy tính và ứng dụng trong đời sống

- **Mạng máy tính (computer network)** là hệ thống bao gồm nhiều hệ máy tính đơn lẻ (nút mạng) được *kết nối* với nhau theo *kiến trúc* nào đó và có khả năng trao đổi thông tin.
 - **Kết nối (interconnected):** *dây (wire), sóng (wave)...*
 - **Kiến trúc (architecture):** cách thức kết nối và trao đổi thông tin.
 - **Nút mạng (node):** host, workstation, network component...
- **Lợi ích của mạng:**
 - Chia sẻ, trao đổi thông tin.
 - Tăng cường sức mạnh của hệ thống (distributed system, parallel system).

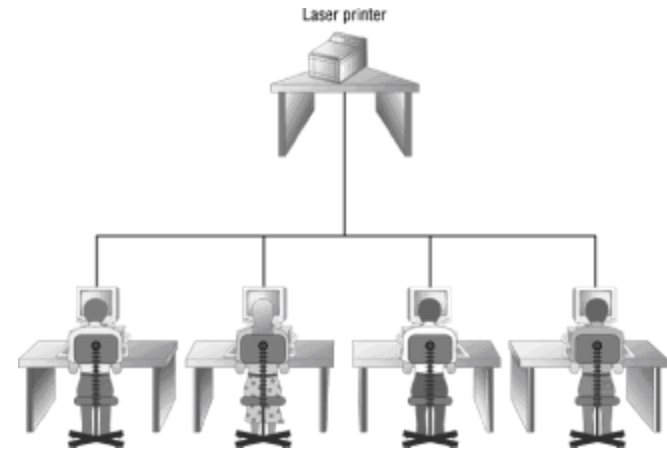
Ứng dụng của mạng máy tính trong đời sống

- Mạng nội bộ (cơ quan, toà nhà)
 - Chia sẻ tài nguyên
 - Liên lạc trong mạng nội bộ cơ quan
- Cung cấp dịch vụ (mô hình client/server).
 - Web, Email, search engine, tin tức.
 - Thương mại điện tử
- People online communication.
 - Chatting, conference
 - Chính phủ, Bộ GD hợp qua mạng về tuyển sinh
 - Điện thoại (PSTN, Mobile).
- Chính phủ điện tử (egovernment)...

Chia sẻ tài nguyên máy in



before 2003



since 2003



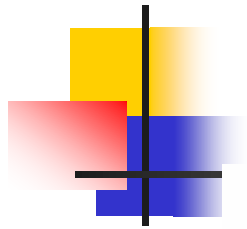
Các yếu tố của mạng máy tính

- Đường truyền vật lý (physical media)
 - Truyền *tín hiệu* giữa các hệ thống.
 - Hữu tuyến (cable) và vô tuyến (wireless).
 - Dải thông (bandwidth):
 - Dải tần số cho phép truyền.
 - Đôi khi được sử dụng để ám chỉ lượng dữ liệu cho phép truyền
 - Tốc độ (speed) hay thông lượng (throughput):
 - Số lượng bit truyền được trong một giây (bps).
 - Số lượng thay đổi tín hiệu trong một giây (baud).
- Kiến trúc mạng (network architecture)
 - Hình trạng mạng (topology).
 - Giao thức (protocol).

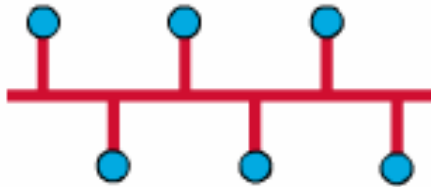
1.2. Phân loại (network classification, taxonomy)

- Theo kỹ thuật truyền (transmission technique)
 - Circuit-switched
 - Packet-switched
 - Message-switched
- Theo quy mô (scale)
 - LAN
 - MAN
 - WAN, ...

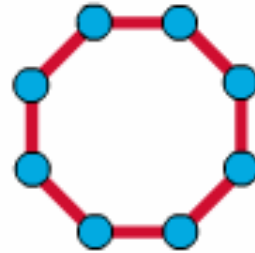
| Interprocessor distance | Processors located in same | Example |
|-------------------------|----------------------------|---------------------------|
| 1 m | Square meter | Personal area network |
| 10 m | Room | |
| 100 m | Building | |
| 1 km | Campus | Local area network |
| 10 km | City | |
| 100 km | Country | Metropolitan area network |
| 1000 km | Continent | |
| 10,000 km | Planet | Wide area network |
| | | |
| | | The Internet |



Topology



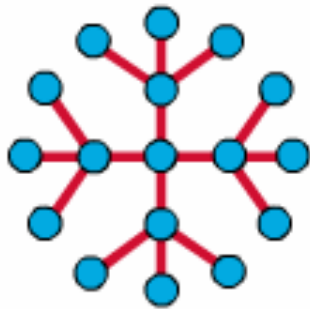
Bus Topology



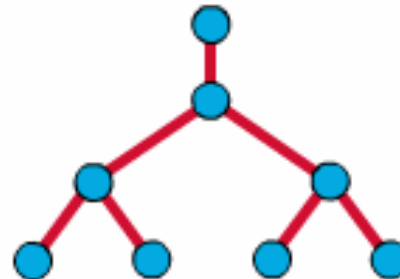
Ring Topology



Star Topology



Extended Star Topology



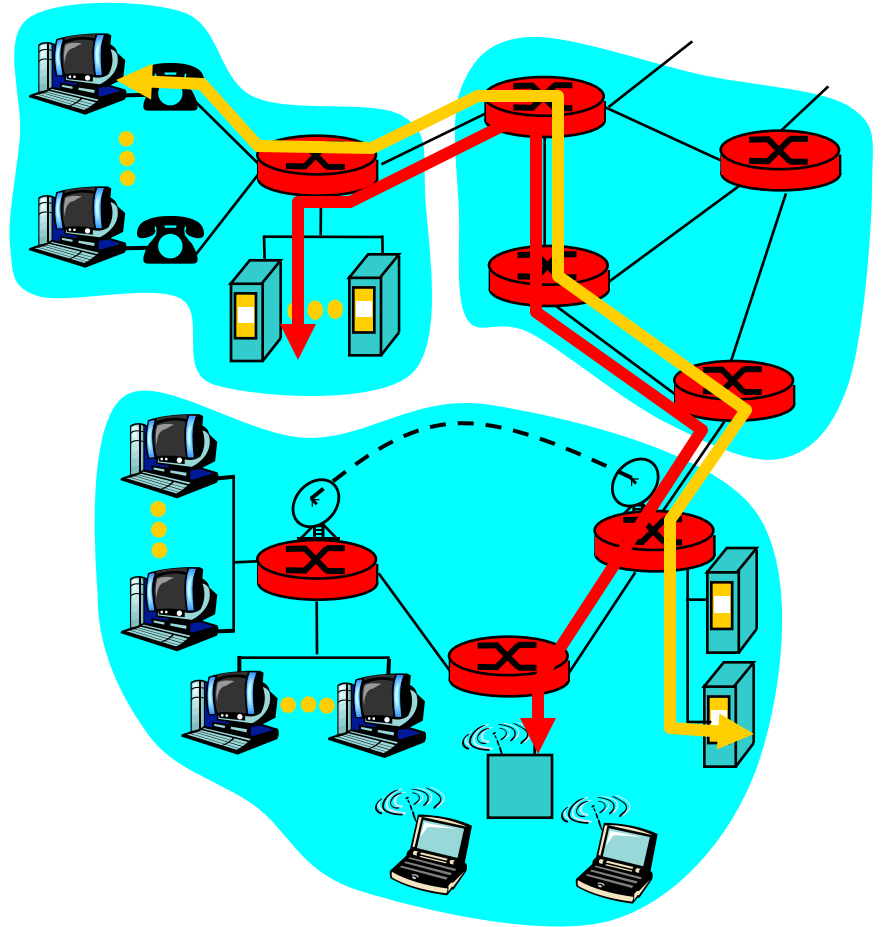
Hierarchical Topology



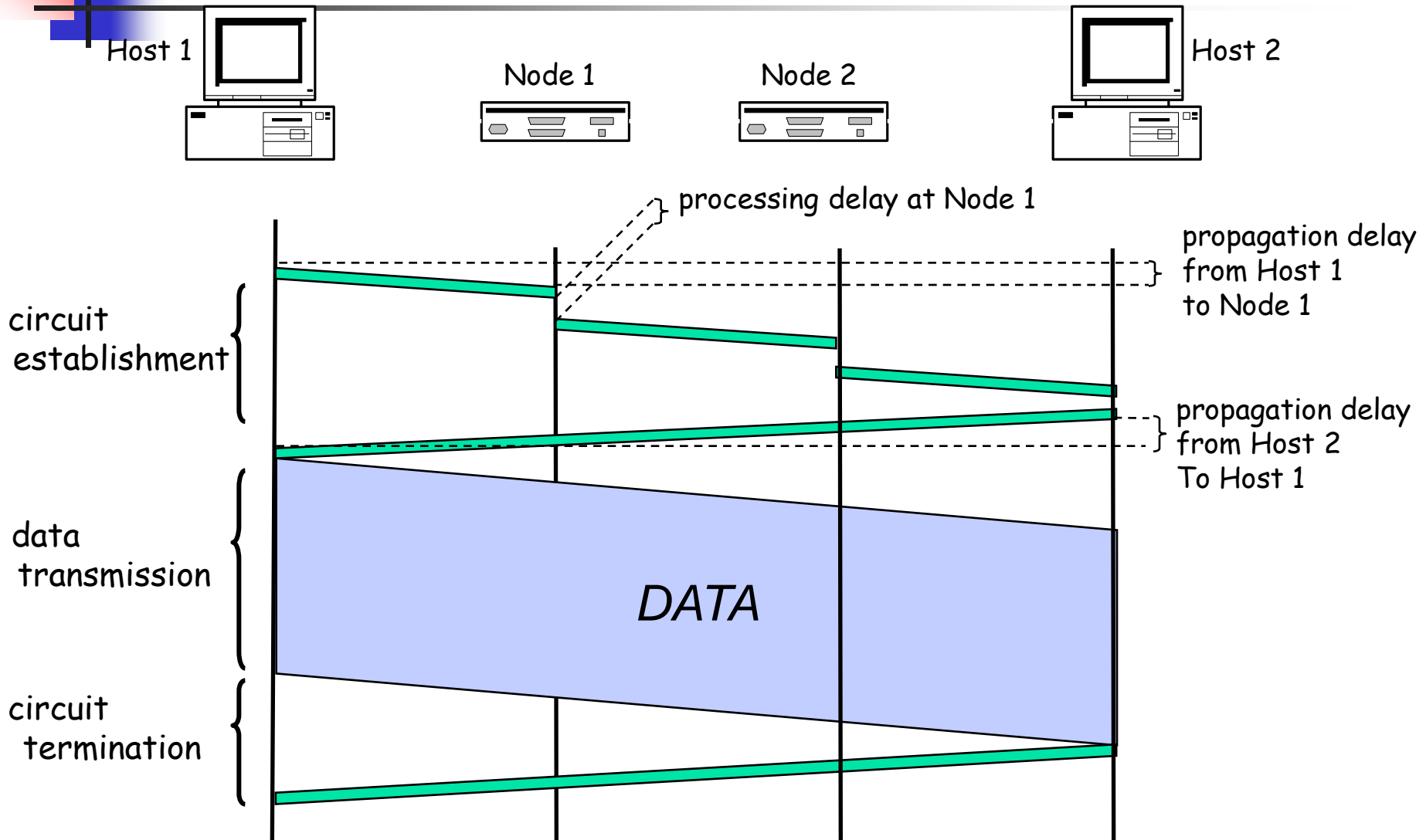
Mesh Topology

Circuit switching

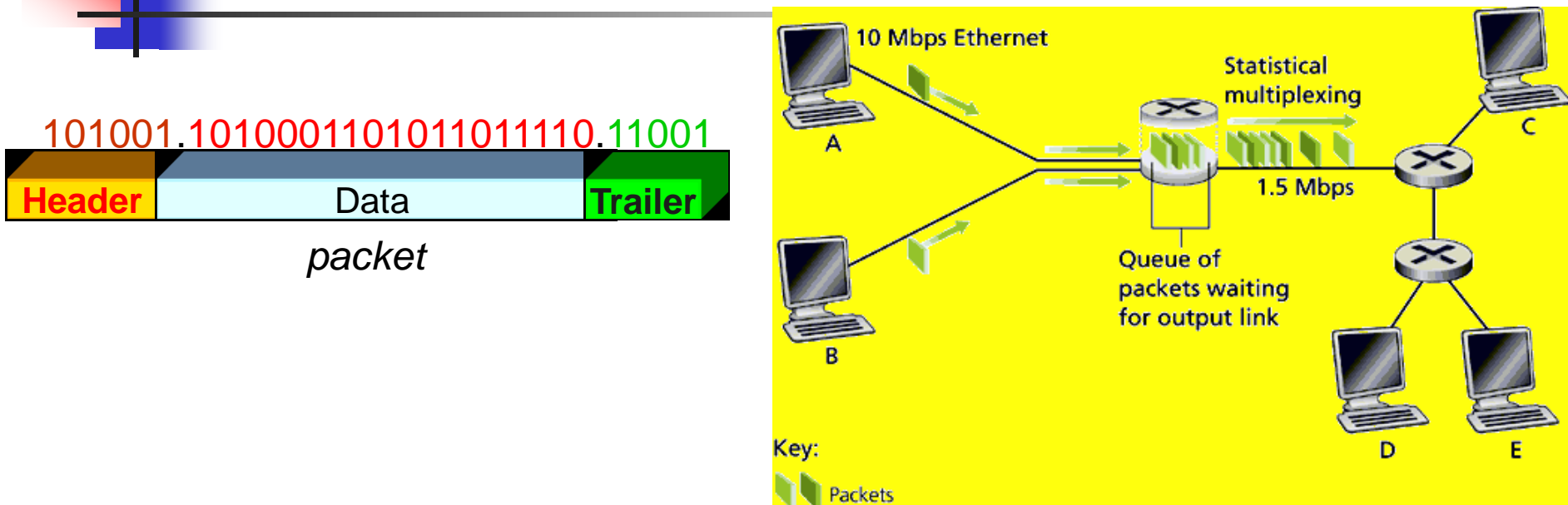
- Khi hai nút muốn trao đổi thông tin → thiết lập kênh (circuit).
- Kênh được giữ riêng cho hai nút cho tới khi kết thúc phiên trao đổi.
- VD: Mạng điện thoại.



Các giai đoạn của circuit switching



Packet switching



- Dữ liệu được chia thành các gói tin (packet). Mỗi gói đều có phần thông tin điều khiển (header, trailer) cho biết nguồn gửi, đích nhận...
- Các gói tin có thể đến và đi theo những đường khác nhau → dồn kênh (multiplexing), được lưu trữ rồi chuyển tiếp khi đi qua nút trung gian (store & forward).

So sánh circuit switching và packet switching

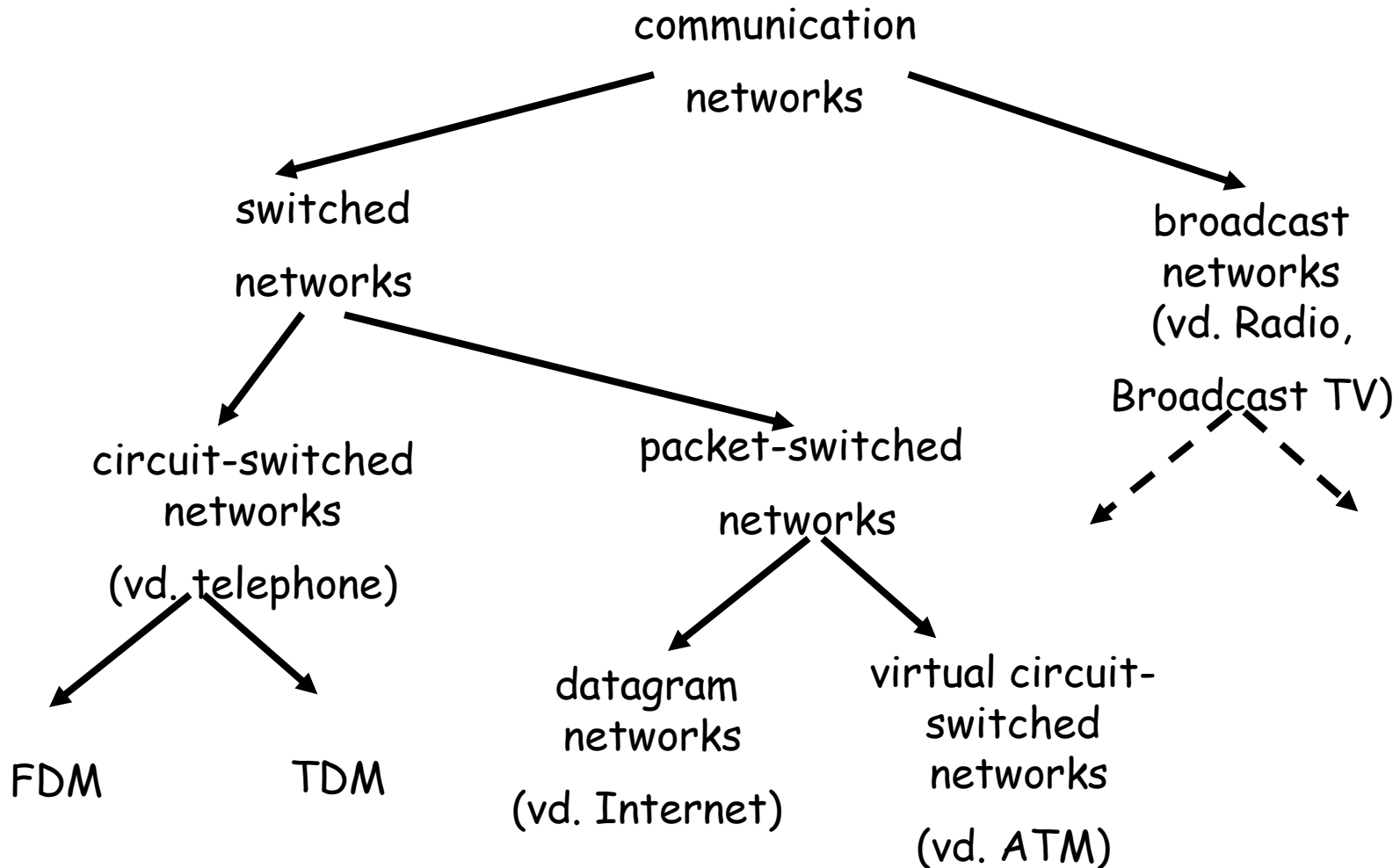
■ Packet switching

- Không chiếm dụng đường truyền → cho phép nhiều người dùng hơn, hiệu suất sử dụng đường truyền cao.
- Không cần thiết lập kênh truyền (call setup).
- Có độ trễ gói tin.
- Cần phải có cơ chế khắc phục lỗi.

■ Circuit switching

- Call setup
- Thích hợp với truyền tin chất lượng cao, tức thì.

Sơ đồ phân loại các Mạng





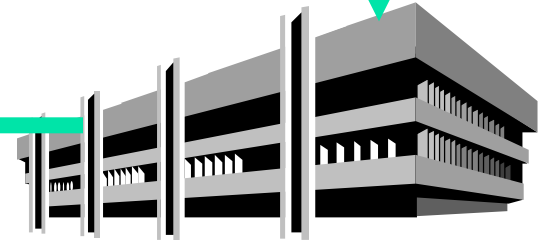
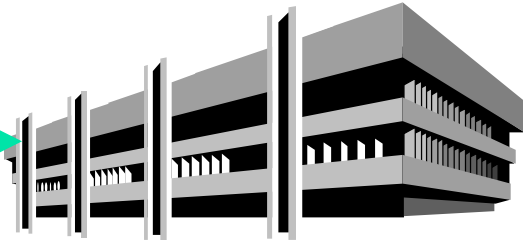
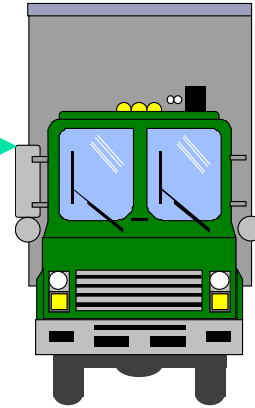
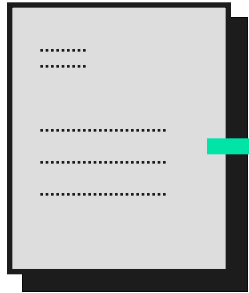
1.3. Giao thức mạng (Protocol)

- Giao thức (protocol): Tập hợp các quy tắc giao tiếp giữa các hệ máy tính.
- Mô hình giao thức mạng hiện nay tuân theo kiến trúc phân tầng (layer architecture).
 - Mỗi tầng đảm nhận những chức năng nhất định.
 - Chỉ có tầng dưới cùng là giao tiếp trực tiếp với nhau.
 - Một tầng từ tầng 2 trở lên chỉ giao tiếp với nhiều nhất hai tầng (kề trên, kề dưới).
 - Thông tin truyền từ tầng N của hệ thống 1 sang tầng N của hệ thống 2 phải truyền qua các tầng $N-1 \rightarrow N-2 \rightarrow \dots \rightarrow 1$ của hệ thống 1 và các tầng $1 \rightarrow 2 \rightarrow \dots \rightarrow N-1$ của hệ thống 2.

Mailing system

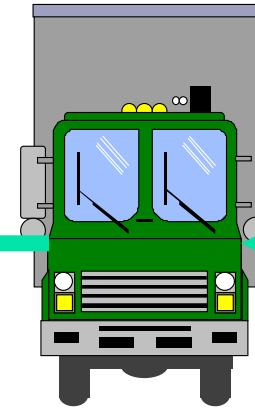
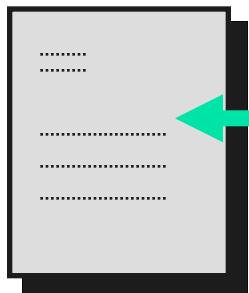
Letter

Addressed
Envelope



Letter

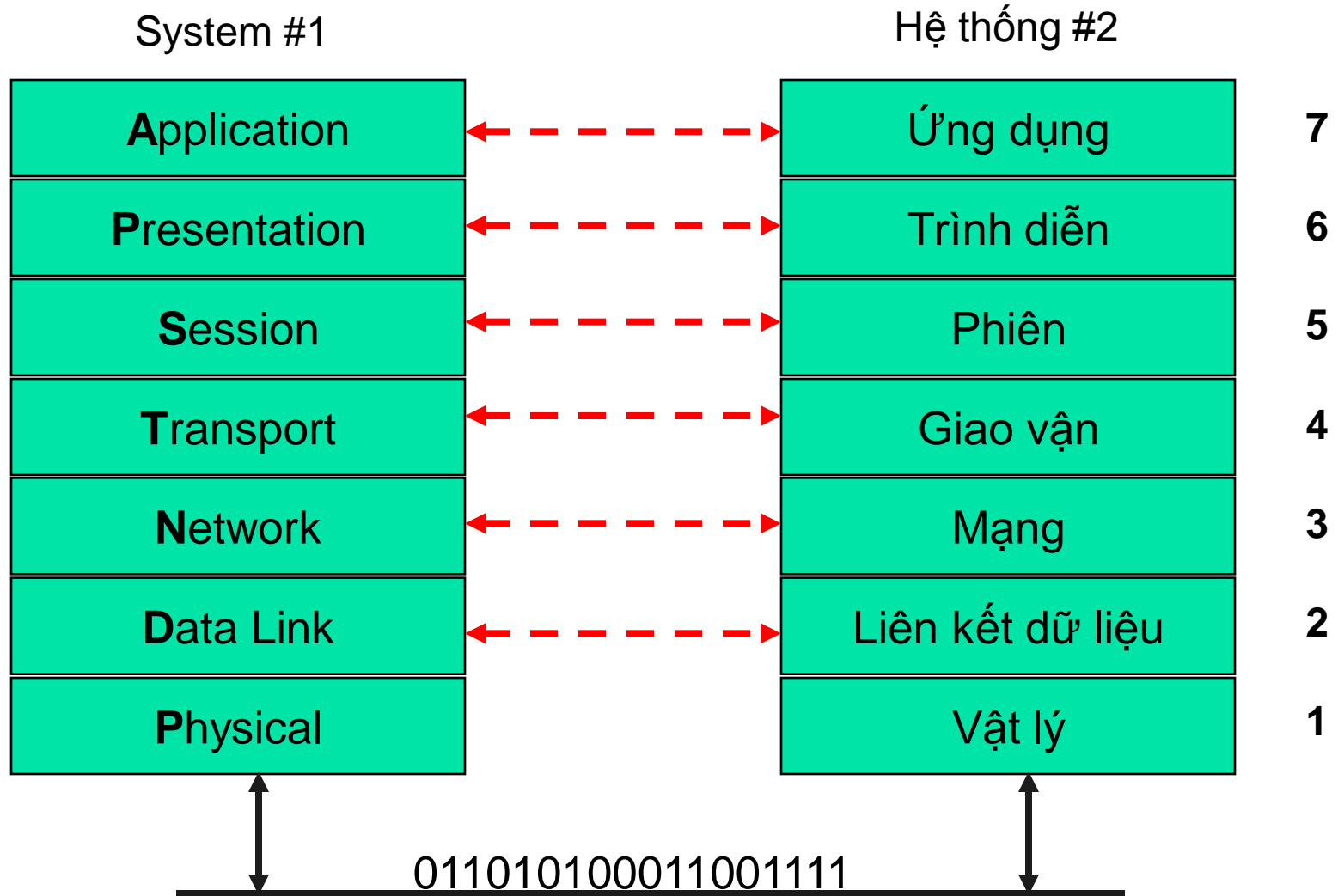
Addressed
Envelope



1.4. Các mô hình tham chiếu (Reference Models)

- Open System Interconnection Reference Model (OSI Reference Model)
 - Đưa ra bởi ISO (International Organization for Standardization) năm 1984.
 - Mô hình tham chiếu lý thuyết cho các hệ thống mở nói chung.
 - 7 tầng: Physical, Data Link, Network, Transport, Session, Presentation, Application.
- TCP/IP Reference Model
 - Sử dụng cho mạng Internet.
 - 4 tầng: Host-to-network, Internet, Transport, Application.

Mô hình OSI





Tầng 1: The Physical Layer

- Chỉ có tầng vật lý của hai hệ thống được ***kết nối*** và truyền thông ***trực tiếp*** với nhau (wire/wireless).
- Các đặc tả vật lý (điện, điện từ...) nhằm đảm bảo sự kết nối và truyền ***tín hiệu*** giữa hai hệ thống.
- Một số yếu tố:
 - Cáp truyền (Cable).
 - Mức điện thế (voltage levels).
 - Thời gian biến thiên hiệu điện thế.
 - Chu kỳ tín hiệu, khoảng cách...



Tầng 2: The Data Link Layer

- Biên dữ liệu thô nhận được từ tầng vật lý thành dữ liệu có cấu trúc logic cụ thể hơn.
 - Framing.
 - 001101010 → Khung (frame) có cấu trúc.
 - Physical Addressing.
 - Dữ liệu đến từ đâu? Máy tính nào gửi đến?
 - Dữ liệu cần phải gửi tiếp đi đâu?
- Đảm bảo sự *tin cậy* của tín hiệu truyền giữa hai tầng vật lý.
 - Kiểm soát lỗi (error control).
 - Kiểm soát luồng (flow control).
- Bao gồm hai tầng con (LLC và MAC).



Tầng 3: The Network Layer

- Chọn đường đi giữa các nút mạng (path-selection).
- Điều khiển luồng mạng con (subnet flow control).
- Cắt hợp dữ liệu (fragmentation & reassembly).
- Kết nối các mạng có kiến trúc khác nhau.



Tầng 4: The Transport Layer

- Tầng trên cùng của quá trình truyền dữ liệu.
- Đảm bảo dữ liệu được truyền thông suốt và tin cậy giữa hai hệ thống (2 end-systems).
 - Cắt/hợp dữ liệu (fragmentation/reassembly).
 - Kiểm soát lỗi, kiểm soát luồng (error detection and recovery, information flow control).
 - Thiết lập, quản lý các kênh liên lạc (virtual circuits).
 - Dồn kênh (multiplexing).



Tầng 5: The Session Layer

- Thiết lập và quản lý các phiên truyền thông giữa hai hệ thống.
 - Chứng thực (security authentication).
 - Thiết lập liên kết (connection establishment).
 - Huỷ bỏ liên kết (connection release).
 - Phản hồi (acknowledgement).
 - Truyền lại (data retransmission).



Tầng 6: The Presentation Layer

- Đảm bảo thông tin truyền từ ứng dụng của hệ thống truyền có thể đọc được bởi ứng dụng của hệ thống nhận.
 - Cú pháp và ngữ nghĩa của dữ liệu (syntax & semantic).
 - Định dạng dữ liệu (data formatting).
 - Chuyển đổi định dạng (format exchange).
 - Nén dữ liệu (data compression).

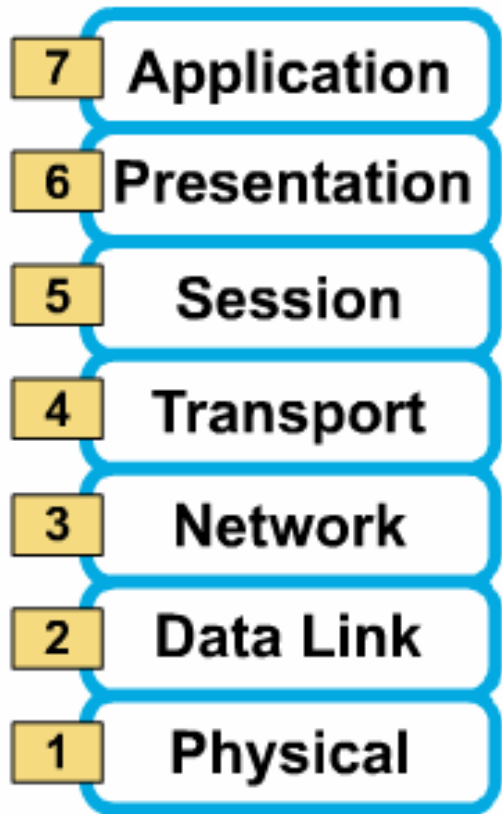


Tầng 7: The Application Layer

- Cung cấp các phương tiện để người dùng có thể truy cập vào mô hình OSI.
 - Các giao thức truyền thông điệp giữa các chương trình ứng dụng (web, mail...).
 - Các dịch vụ cho các ứng dụng nằm ngoài mô hình OSI (Word, Access, SQL Server...)

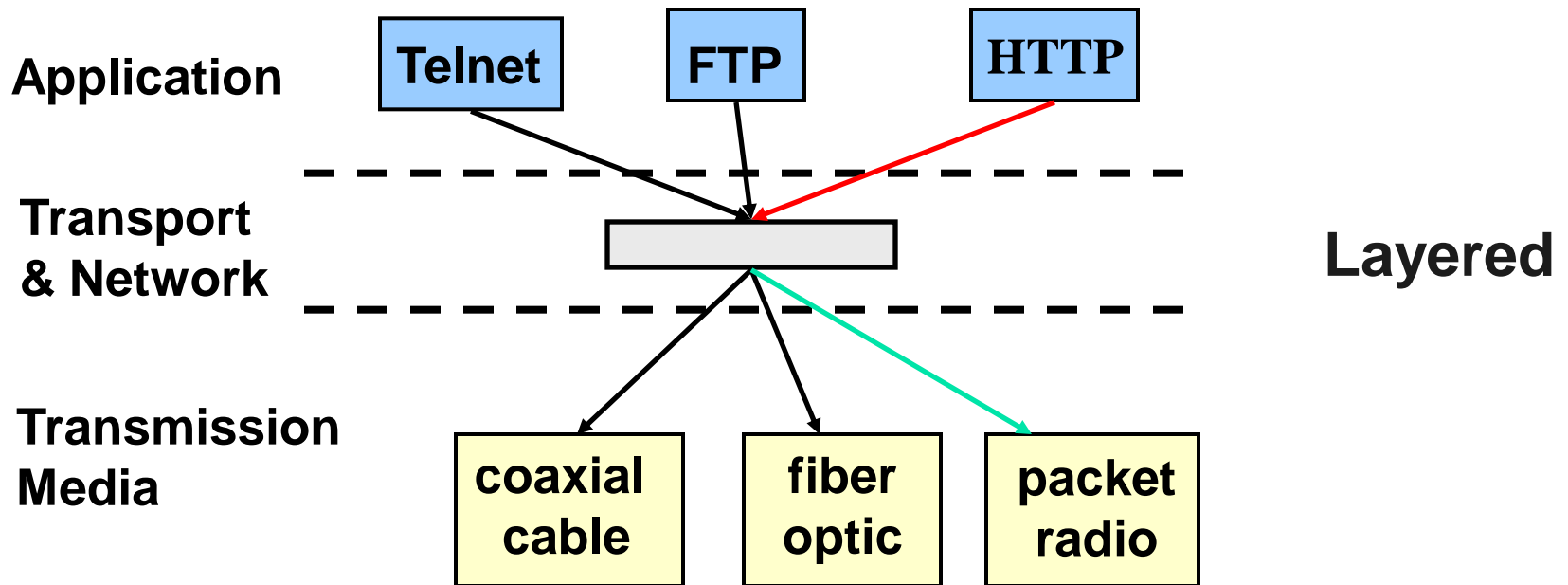
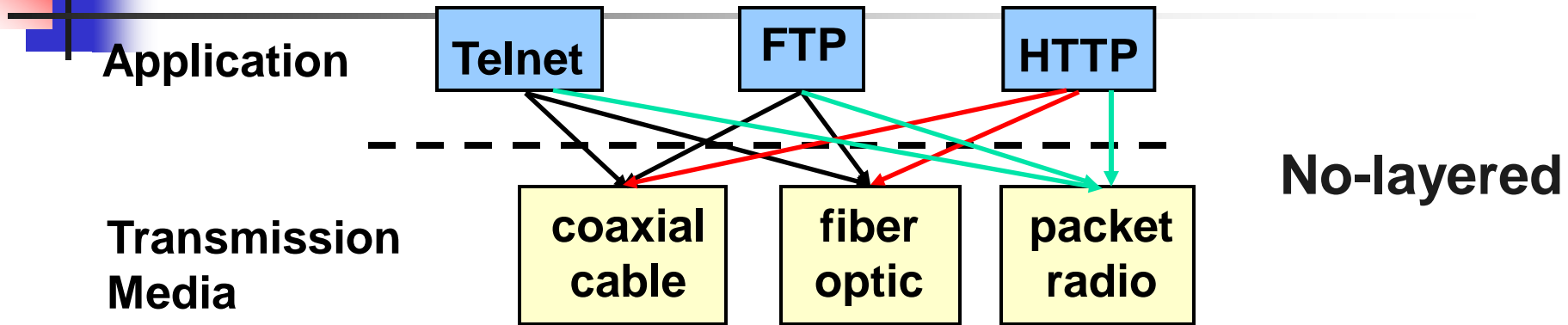
Tại sao phải phân tầng?

Why a Layered Network Model?



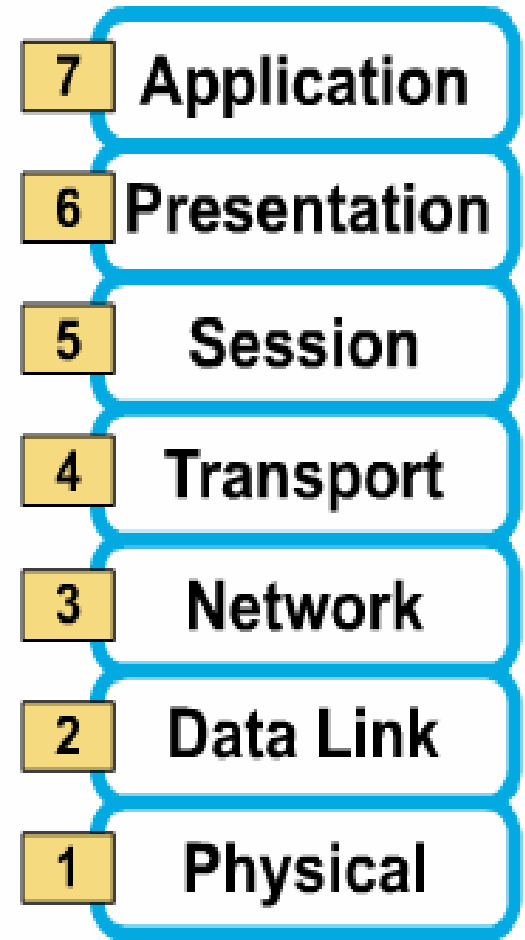
- ◆ Reduces complexity
- ◆ Standardizes interfaces
- ◆ Facilitates modular engineering
- ◆ Ensures interoperable technology
- ◆ Accelerates evolution
- ◆ Simplifies teaching and learning

No-layered & Layered



Tóm lược về Mô hình OSI

- Physical: binary transmission
 - signals, media, connectors, voltages...
- Data Link: access to media
 - bits error control, flow control.
 - physical addressing, net topology.
- Network: address and best path
 - path selection, routing, addressing, intern
- Transport: end-to-end transmission
 - data transportation, virtual circuit
 - error detection and recovery, information
- Session: interhost communication
 - session management
- Presentation: data representation
 - data format, data syntax
- Application: network services to applications

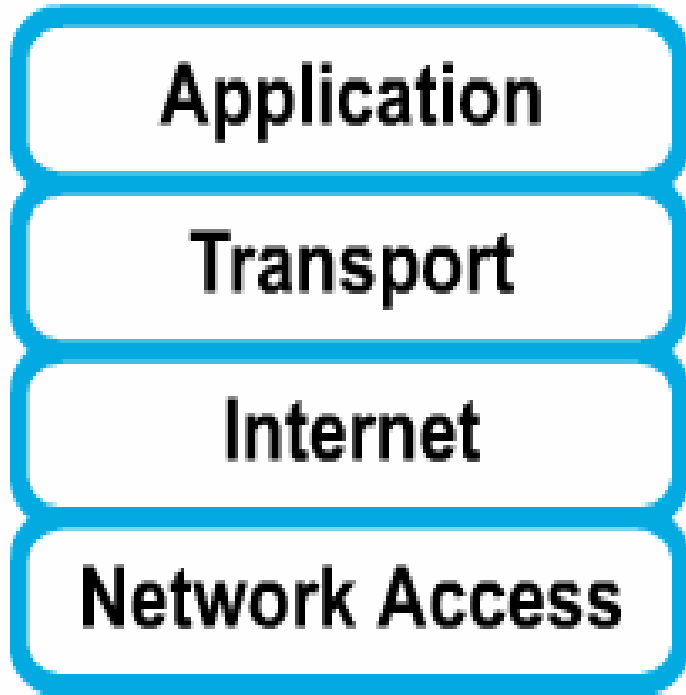




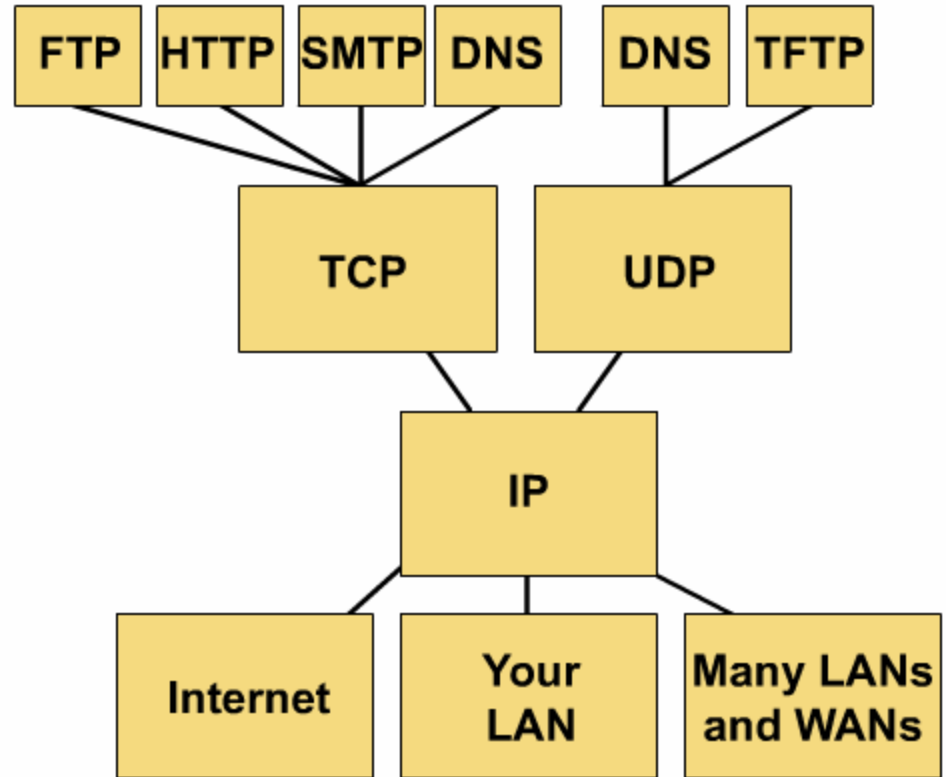
Mô hình/Giao thức TCP/IP

- Mô hình OSI chỉ mang tính chất lý thuyết, phục vụ nghiên cứu và học tập.
- TCP/IP là mô hình áp dụng cho mạng Internet.
 - TCP = Transmission Control Protocol.
 - IP = Internet Protocol.
 - TCP, IP là hai giao thức phổ biến trong họ giao thức TCP/IP.

TCP/IP Layers & Protocols



Layers

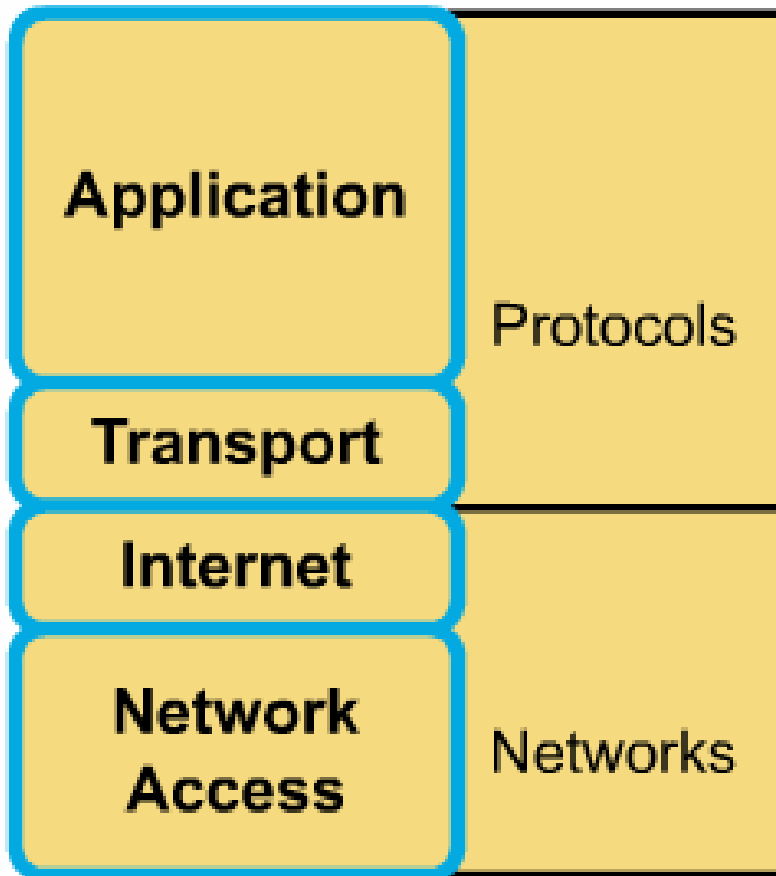


Protocols

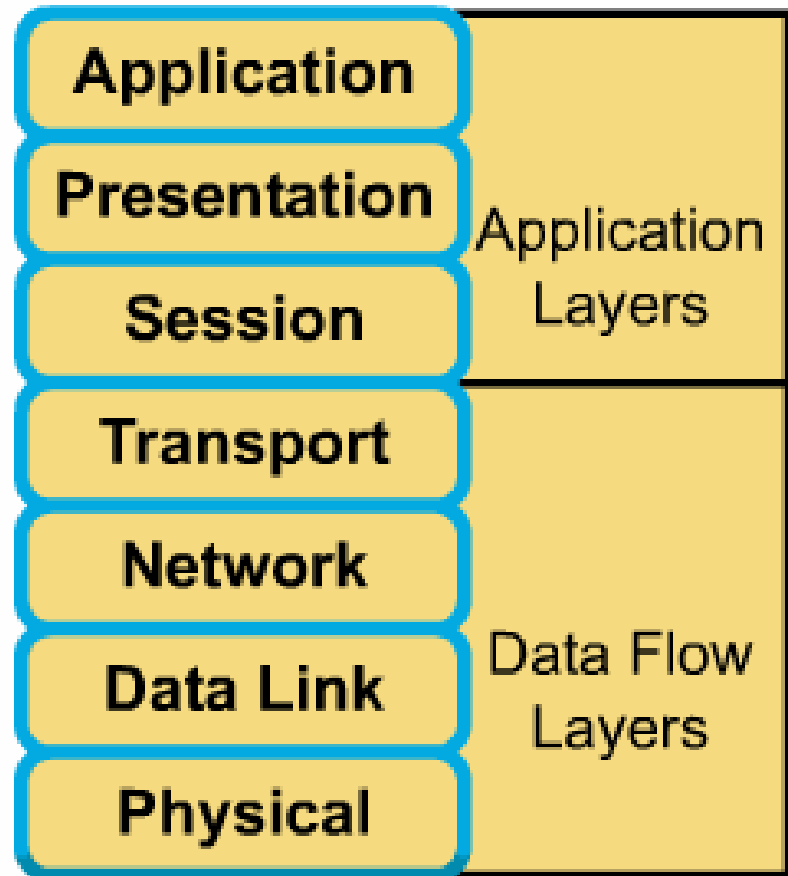
***Network Access = Host-to-network = Data link + Physical
Network = Internet***

Đổi sánh OSI và TCP/IP

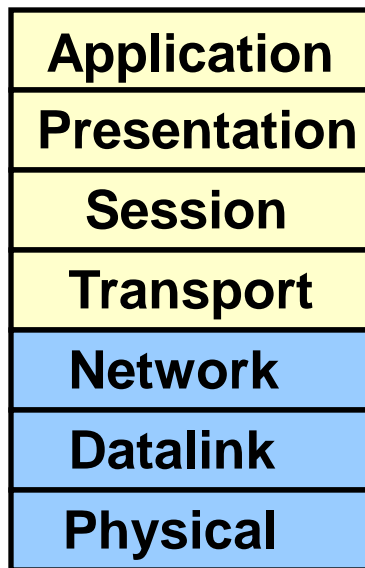
TCP/IP Model



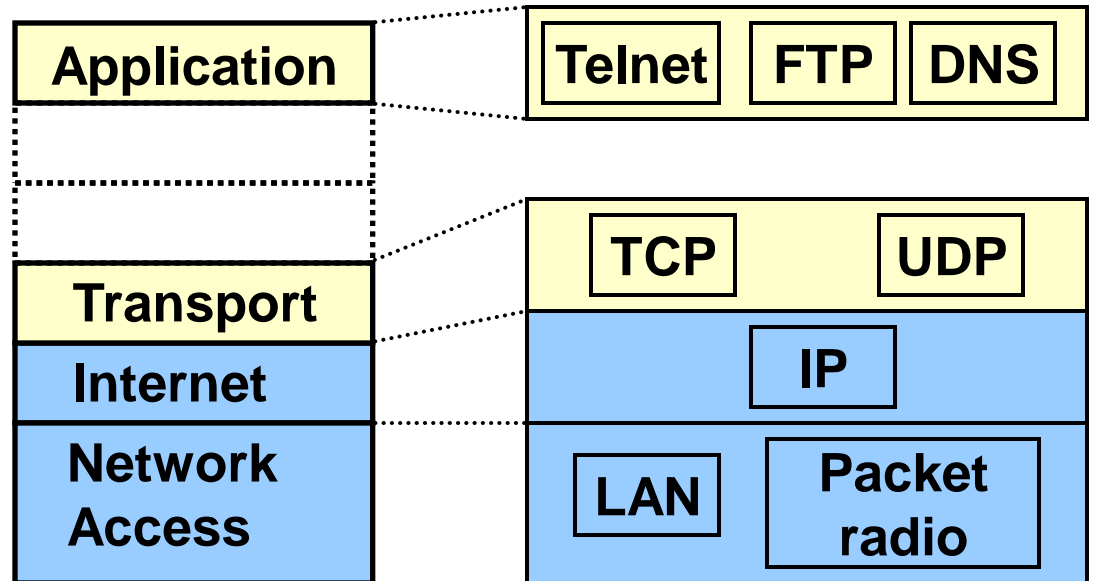
OSI Model



TCP/IP to OSI



OSI



TCP/IP

Protocols

1.5. Chuẩn hóa mạng và các tổ chức quản lý về Mạng

- OSI:
 - Chỉ nêu lên chức năng của từng tầng.
 - Không chỉ ra cài đặt các tầng này như thế nào.
 - Để hai hệ thống giao tiếp được với nhau thì chúng cần:
 - Được cài đặt các chức năng truyền thông chung.
 - Tổ chức thành cùng một tập tầng.
 - Hai tầng đồng mức phải có chung giao thức.
- Từ đó, cần phải chuẩn hoá (standardization)

Các tổ chức và chuẩn thông dụng

- ISO (International Organization for Standardization)
 - <http://www.iso.org>
- CCITT (Comité Consultatif International pour Télégraphe et Téléphone)
 - Không ban hành các chuẩn mà ban hành các khuyến nghị (X.25...).
- ANSI (American National Standard Institute)
- NIST (National Institute of Standards and Technology)
- IEEE (Institute of Electrical and Electronics Engineers)
- IAB (Internet Architecture Board).
- RFCs (Request For Comments).
- IRTF (Internet Research Task Force).
- IETF (Internet Engineering Task Force).
- ISOC (Internet Society)...



Nhóm chuẩn IEEE 802 (LAN)

| Number | Topic |
|----------|--|
| 802.1 | Overview and architecture of LANs |
| 802.2 ↓ | Logical link control |
| 802.3 * | Ethernet |
| 802.4 ↓ | Token bus (was briefly used in manufacturing plants) |
| 802.5 | Token ring (IBM's entry into the LAN world) |
| 802.6 ↓ | Dual queue dual bus (early metropolitan area network) |
| 802.7 ↓ | Technical advisory group on broadband technologies |
| 802.8 † | Technical advisory group on fiber optic technologies |
| 802.9 ↓ | Isochronous LANs (for real-time applications) |
| 802.10 ↓ | Virtual LANs and security |
| 802.11 * | Wireless LANs |
| 802.12 ↓ | Demand priority (Hewlett-Packard's AnyLAN) |
| 802.13 | Unlucky number. Nobody wanted it |
| 802.14 ↓ | Cable modems (defunct: an industry consortium got there first) |
| 802.15 * | Personal area networks (Bluetooth) |
| 802.16 * | Broadband wireless |
| 802.17 | Resilient packet ring |

1.6. Hệ điều hành trong môi trường mạng

- Không hỗ trợ mạng
 - DOS
- Trạm làm việc
 - Windows 9x/NT Workstation/2000/XP
 - MacOS
 - Linux...
- Máy chủ quản lý mạng, cung cấp dịch vụ mạng.
 - Windows NT/2000/2003 Server
 - Novel NetWare
 - Unix, Linux, Sun Solaris...



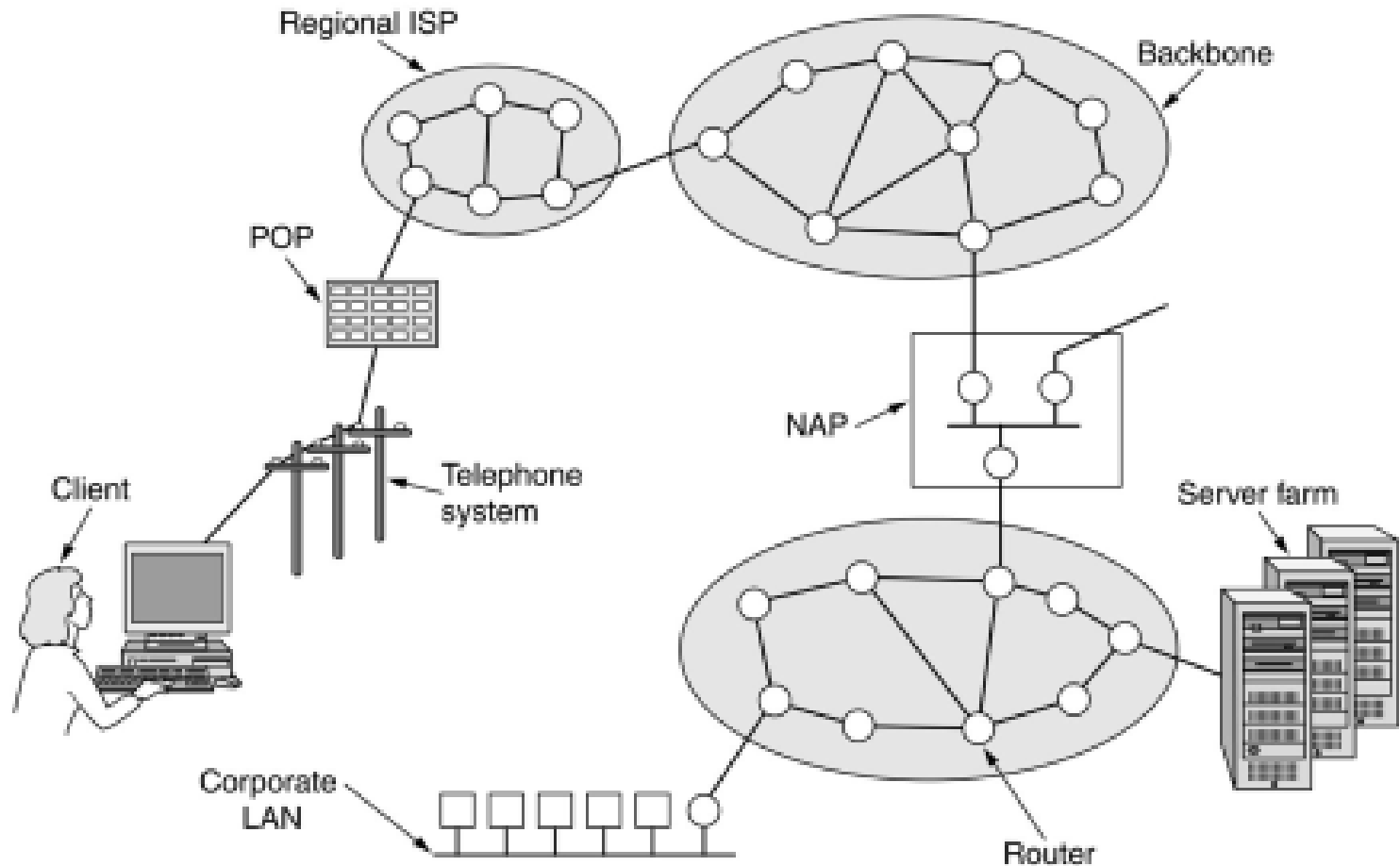
4.2. Internet

- Internet là gì?
- Làm thế nào để kết nối vào Internet.
- World wide web.
- Search engine.
- Email.
- Instant messenger, Internet Phone,...
- Thương mại điện tử.
- Sử dụng Internet phục vụ học tập.



Internet

- WAN toàn cầu (Global Wide Area Network).
- Mạng toàn cầu của các mạng máy tính (Global Network of networks).
- Internet khởi đầu từ ARPANET (Advanced Research Projects Agency Network), 1969, USA.
- Không có một cơ quan nào quản lý chính thức Internet.
- Không phải toàn bộ máy tính trên thế giới đều được kết nối với Internet.



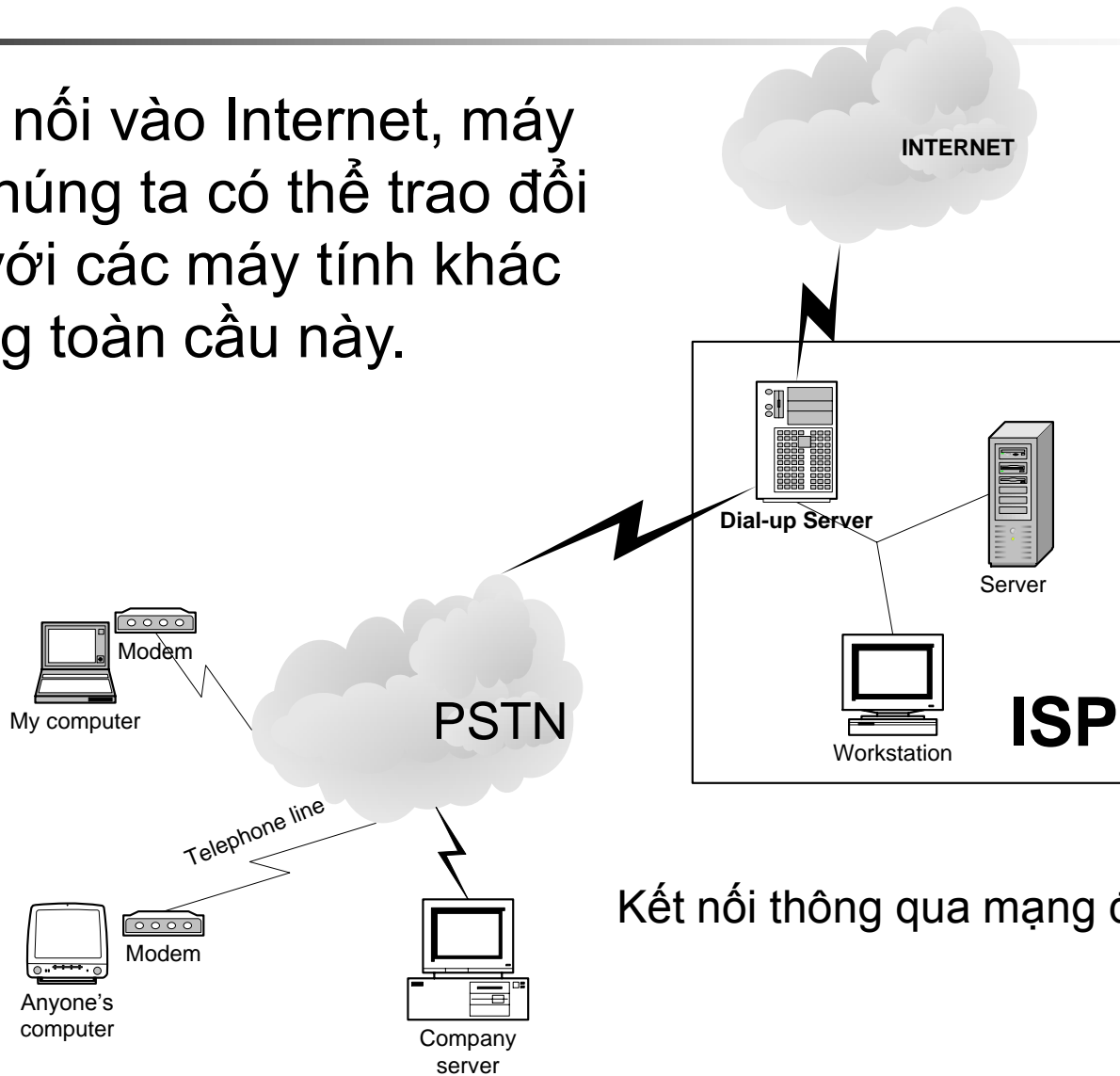
- Hàng trăm triệu nút mạng được kết nối khắp hành tinh.
- Hàng triệu dịch vụ được cung cấp (web, mail...).



Kết nối với Internet

- Nhà cung cấp dịch vụ Internet (ISP - Internet Service Provider):
 - America Online.
 - VDC (VNPT).
 - FPT, Viettel, Netnam,...
- Các máy tính gia đình, cơ quan,... phải kết nối thông qua (ISP).
 - Dial-up.
 - ADSL.
 - Leased line.
 - Wireless.
 - ...

Khi đã kết nối vào Internet, máy tính của chúng ta có thể trao đổi thông tin với các máy tính khác trong mạng toàn cầu này.



Kết nối thông qua mạng điện thoại

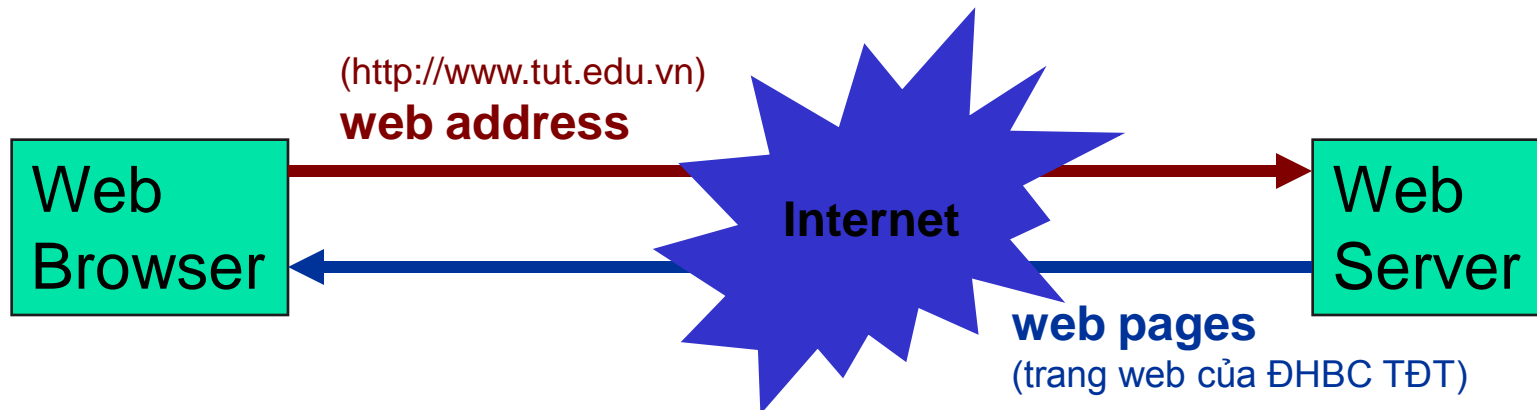


World wide web (WWW)

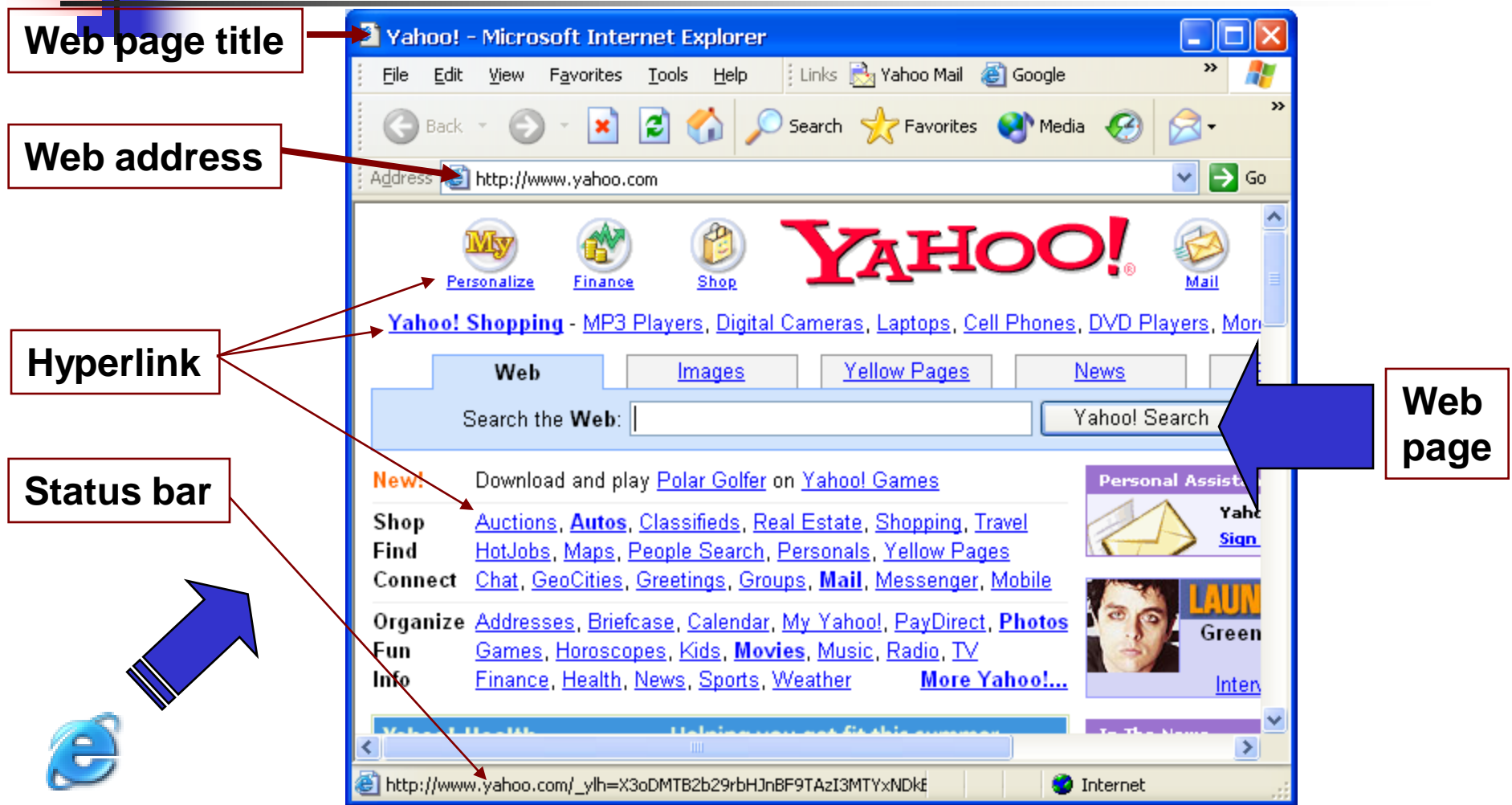
- Trang web (web page):
 - Một dạng tài liệu được sử dụng phổ biến trên Internet.
 - Loại file: HTML (Hyper Text Markup Language).
- Máy tính cung cấp tài liệu HTML → web server.
- WWW = Tất cả web servers+ web pages.
- Web pages của một tổ chức, trường ĐH,... → website.
- Tim Berner Lee là tác giả của những khái niệm HTML, WWW, 1989.

Khai thác WWW.

- Để khai thác WWW, cần:
 - Kết nối với Internet.
 - Trình duyệt web (web browser): Internet Explorer, Mozilla, Netscape Navigator,...



Internet Explorer



Internet Explorer



Web Address

- Web address:
 - Cho biết trang web đặt tại đâu.
 - Bắt đầu bằng http:// (Hyper Text Transfer Protocol)
 - <http://www.yahoo.com>, <http://www.dtu.edu.vn>.



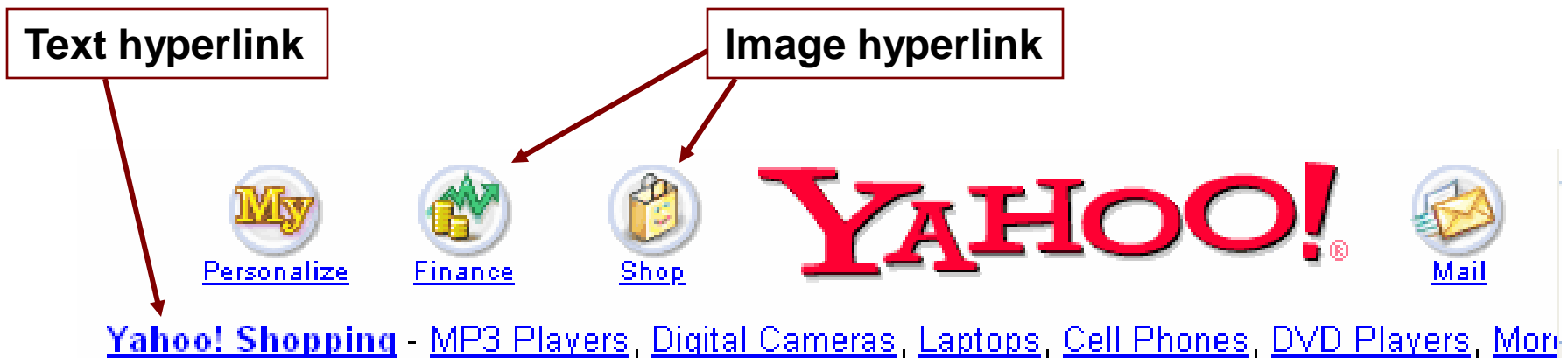
Domain types and countries

- **edu** (education), **ac** (academic) : các trang web giáo dục.
- **com** (commercial): các trang web thương mại.
- **org** (organization): các trang web phi lợi nhuận.
- **gov** (government): các trang web của chính phủ.
- **net, biz, info,...**

- **vn**: Vietnam.
- **uk**: United Kingdom.
- **au**: Australia.
- **ca**: Canada.
- **de**: Germany.
- **jp**: Japan.
- **cn**: Chinese.
- **hk**: Hong Kong.
- **th**: Thailand.
- **sg**: Singapore.

Hyperlink

- Liên kết từ trang web này tới trang web khác.
- Kích đơn vào link → chuyển sang trang web khác (được chỉ ra bởi link đó).
- **Màu xanh** → chưa thăm, **Màu tím** → đã thăm.



Một vài dịch vụ WWW đáng quan tâm

Giáo dục:

- Harvard: <http://www.harvard.edu>
- Stanford: <http://www.stanford.edu>
- MIT:
 - <http://www.mit.edu>
 - <http://ocw.mit.edu> → MIT Open CourseWare.
- Cambridge: <http://www.cam.ac.uk>
- Queensland: <http://www.uq.edu.au>
- Bách Khoa: <http://www.hut.edu.vn>
- ĐH Cần Thơ: <http://www.ctu.edu.vn>
- Mạng giáo dục: <http://www.edu.net.vn>
- Thư viện:
 - Internet Public Library: <http://www.ipl.org>
 - Thư viện Quốc Gia: <http://www.nlv.gov.vn/>



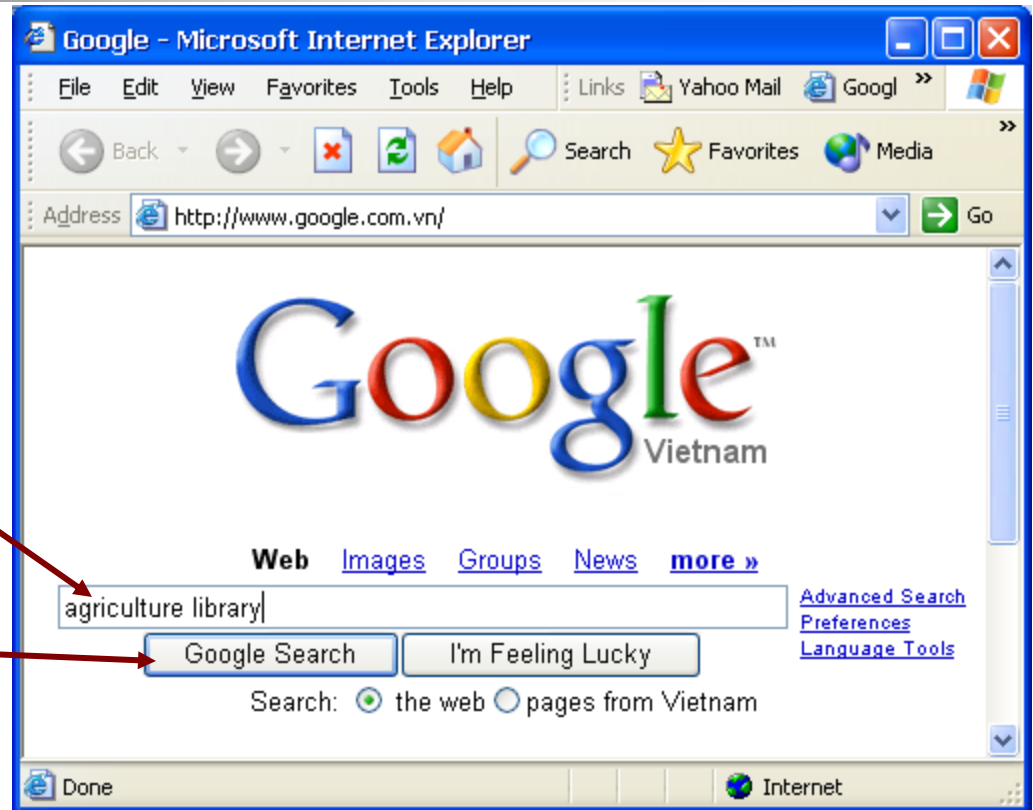
Search engine (SE)

- SE (công cụ tìm kiếm) là dịch vụ tìm kiếm địa chỉ web theo “yêu cầu” trên Internet.
- Một vài SE:
 - <http://www.google.com.vn>
 - <http://www.yahoo.com>
 - <http://www.altavista.com>
 - <http://www.lycos.com>
 - <http://www.metacrawler.com>
 - <http://www.alltheweb.com>
 - <http://www.vinaseek.com>
 - <http://www.panvietnam.com>
 - ...


Google

Truy vấn tìm kiếm

Kích chuột vào đây
để bắt đầu tìm.



Kết quả tìm kiếm với Google

Address  http://www.google.com.vn/search?q=agriculture+library&ie=UTF-8&hl=vi&meta=

Google™ [Internet](#) [Hình Ảnh](#) [Nhóm](#) [Thư Mục](#)

agriculture library [Nâng cao](#)
[Tùy chọn](#)

Tim kiếm: mạng lưới những trang từ nước Việt Nam

Internet Các kết quả **1 - 10** của khoảng **3,060,0**

[National Agricultural Library](#)
Search Our Web Site. AGRICOLA The NAL Catalog and Article Citation Database.
Services and Programs Reference, Info Centers, Special Collections, & more. ...
www.nal.usda.gov/ - 13k - 5 Tháng tám 2004 - [Sẵn có \(trong máy chủ Google\)](#) - [Tim trang tương tự](#)

[Agricola](#)
Bibliographic database of citations to the agricultural literature created by the National Agricultural.
www.nal.usda.gov/ag98/ - [Tim trang tương tự](#)
[[Thêm kết quả từ www.nal.usda.gov](#)]

[The World Wide Web Virtual Library: Sustainable Agriculture](#)
WWW VLWelcome to the World Wide Web Virtual **Library** SUSTAINABLE **AGRICULTURE**
homepage| navigate the **library** with these links: Abstracts ...
www.floridaplants.com/sustainable.htm - 14k - [Sẵn có \(trong máy chủ Google\)](#) - [Tim trang tương tự](#)

[Canadian Agriculture Library / Bibliothèque canadienne de l' ...](#)
Information services offered by the Canadian **Agriculture Library**. / Services
d'information offert par la Bibliothèque canadienne de l'**agriculture**. ...
www.agr.gc.ca/cal/ - 5k - [Sẵn có \(trong máy chủ Google\)](#) - [Tim trang tương tự](#)



Electronic mail (Email, E-mail)

- Thư điện tử (email) là một phương thức trao đổi thông tin người-người trong thời đại Internet.
- Muốn sử dụng thư điện tử phải đăng ký với nhà cung cấp:
 - Miễn phí:
 - Yahoo Mail: <http://mail.yahoo.com>
 - Hotmail: <http://hotmail.com>
 - Trả tiền:
 - FPT, VNN,...
 - Các công ty, trường đại học chỉ cung cấp email cho cán bộ hoặc sinh viên.



Email box và Email address

■ Email box:

- Hòm thư điện tử.
- Là một không gian chứa email tại máy chủ của nhà cung cấp email.
- Mỗi hòm thư có một địa chỉ (email address).



News Group & Mailing list

- **News group services:**

- Dịch vụ cho phép xây dựng các nhóm tin.
- Thảo luận thông qua việc gửi thông điệp tới nhóm tin.
- Một thông điệp được gửi đi sẽ truyền tới mọi thành viên khác.
- Vd: <http://groups.google.com>; <http://groups.yahoo.com>.

- **Mailing list:**

- Là một địa chỉ email đặc biệt mà khi gửi email tới địa chỉ này, mọi thành viên của list sẽ nhận được email đó.
- Vd: <http://groups.yahoo.com>; <http://www.topica.com>; <http://www.coollist.com>.



Các dịch vụ khác

- Instant Messenger:
 - Cho phép chúng ta “nói chuyện” với nhau thông qua việc gửi các messages.
 - Vd: America Online Messenger, Yahoo Messenger.
- Internet Phone:
 - Cho phép gọi điện thoại qua mạng Internet.
 - Nhà cung cấp: FPT, OneConnection, Viettel, VDC,...
- Internet Games, Greeting Card,...

Thương mại điện tử (e-commerce)

- Là “thương mại” trong môi trường “điện tử”, các giao dịch được thực hiện với sự hỗ trợ của công nghệ “điện tử”, đặc biệt là Internet. Vd:
 - Giới thiệu sản phẩm trên Internet.
 - Mua hàng và thanh toán qua Internet (Credit, Master card,...).
 - ...
- Một vài địa chỉ mua bán qua mạng:
 - Amazon: <http://www.amazon.com>.
 - Ebay: <http://www.ebay.com>.
 - Mua máy tính tại IBM: <http://www.ibm.com>.
 - Tìm kiếm hàng hóa, khảo giá: <http://www.shopping.com>.
 - VDC Tiền Phong: <http://vdcsieuthi.vnn.vn>.
 - ...
- Thương mại điện tử tại Việt Nam vẫn chưa phát triển mạnh, hiện tại vẫn chưa phát triển mạnh. Hãy chờ xem!!!



Sử dụng Internet phục vụ học tập

- Hãy quan tâm tới những websites có nội dung liên quan tới những bài học của mình trên lớp:
 - Website có chứa bài giảng:
 - Tìm thông tin liên quan đến bài giảng thông qua các trang web dò tìm
 - Các trang web của các trường ĐH khác,...
 - Website có các tài liệu tham khảo, các website thư viện.
- Sử dụng Internet ở đâu:
 - Sử dụng tại nhà
 - Café Internet.



4.3. An toàn mạng

- Các nguy cơ
- Các biện pháp an toàn mạng



Virus máy tính

- Virus là chương trình máy tính được viết ra với mục đích phá hoại hoặc trục lợi, có khả năng lây lan từ máy tính này sang máy tính khác.
- Các virus nguy hiểm hiện nay lây lan chủ yếu qua môi trường mạng với tốc độ khủng khiếp.
- Ai viết ra virus?
 - Thanh thiếu niên.
 - Sinh viên.
 - Một số thành phần khác.
- Mục đích của người viết virus:
 - Muốn chứng tỏ mình.
 - Phá hoại, đánh cắp thông tin (tài khoản ngân hàng, tài khoản Internet, các thông tin bí mật,...).



Nguy cơ bị nhiễm virus

Máy tính có thể bị nhiễm virus khi:

- Khởi động máy tính từ một đĩa mềm nhiễm virus.
- Chạy một chương trình bị nhiễm virus.

Virus ẩn náu ở đâu:

- Các máy tính đã bị nhiễm.
- Các trang web “nguy hiểm”. Đa phần các trang web có nội dung không lành mạnh đều có virus. Chỉ cần mở trang web là bị nhiễm virus.
- Các tệp đính kèm trong thư điện tử (rất phổ biến).

Tấn công trực tiếp qua mạng:

- Một hacker có thể phát tán virus tới các máy tính trên mạng thông qua các “lỗ hổng” của hệ thống.
- Một máy tính bị nhiễm virus có thể lây sang các máy khác trong mạng.



Phòng và chống Virus

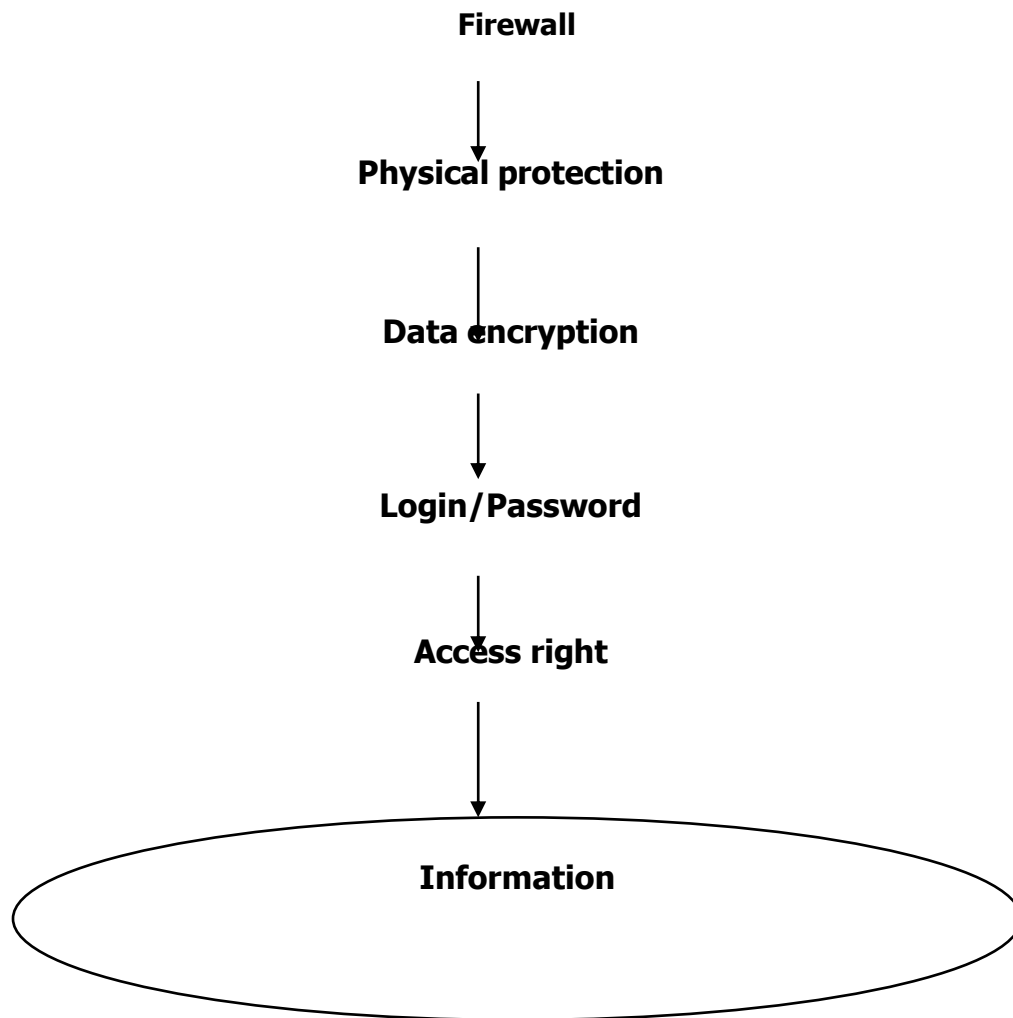
- Phòng ngừa:
 - Cài đặt một chương trình bảo vệ: Norton Antivirus, McAfee Virus Scan, BKAU,... (<http://www.symantec.com>,...)
 - Sử dụng máy tính một cách “an toàn”: cẩn thận khi duyệt web, nhận thư, hãy quét virus các đĩa mềm, các tệp lạ trước khi sử dụng chúng.
 - Thường xuyên cập nhật thông tin về virus , cập nhật chương trình quét Virus, định kỳ quét virus trên máy tính.
- Khi máy tính đã bị nhiễm virus:
 - Quét virus bằng chương trình đã được cập nhật.
 - Truy cập các websites bảo mật để có được hướng dẫn chi tiết cách diệt loại Virus bị nhiễm (<http://www.symantec.com>), thậm chí có thể tải về các Removal Tools.

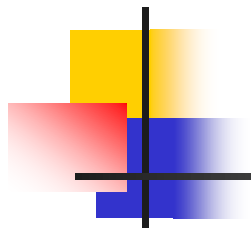


Các nguy cơ phổ biến từ mạng

- Ăn cắp thông tin, thâm nhập hệ thống trái phép
- Tê liệt hệ thống
- Phát tán spam, thư rác, tin quảng cáo...

Các biện pháp







DUY TAN UNIVERSITY

THUẬT TOÁN & CHƯƠNG TRÌNH

Nguyễn Thanh Trung

MỤC TIÊU

- Giới thiệu tổng quan về thuật toán và cách chuyển từ 1 thuật toán thành 1 chương trình bằng một ngôn ngữ lập trình cụ thể (C).
- Những yêu cầu khi xây dựng thuật toán: tính đúng đắn, khả thi,... cũng như xác định độ phức tạp của thuật toán.

Bố cục

- Giới thiệu tổng quan
- Trình bày và triển khai thuật toán
- Đánh giá thuật toán
- Cài đặt Chương trình

Tài liệu tham khảo

- -Chương 5,6 Computer Science
- -Chương 5 bài giảng Giới thiệu Khoa học Máy tính.

5.1. Tổng quan

- Máy tính?
 - ❑ Làm theo “lệnh” của con người.
 - ❑ Điểm mạnh là tính toán với tốc độ cao (hàng tỷ phép tính trên giây).
- Làm thế nào để “ra lệnh” cho máy tính?
 - ❑ Lập chương trình cho máy tính
- Chương trình?
 - ❑ Nói cho máy tính biết phải làm gì, như thế nào,...



- Muốn “ra lệnh” cho máy tính: Sử dụng một “ngôn ngữ” chung → ngôn ngữ lập trình (programming language)
- Lập trình (computer programming)
 - Dùng ngôn ngữ lập trình lập nên chương trình hoạt động cho máy tính.
- Các thể hệ của ngôn ngữ lập trình
 - Thế hệ 1 (bậc thấp): ngôn ngữ máy, assembly.
 - Thế hệ 2: Gần với ngôn ngữ tự nhiên hơn, phục vụ những nhu cầu lập trình nhất định (FORTRAN, COBOL, ALGOL,...)
 - Thế hệ 3: Gần gũi, vạn năng (PASCAL, C, C++,...)
 - Thế hệ 4: Truy vấn, hỗ trợ quyết định, lập trình trí tuệ nhân tạo
 - (SQL, LISP, PROLOG,...)

Thuật toán

- Giải thuật, thuật giải, thuật toán đều dùng để ám chỉ một thuật ngữ tiếng Anh có tên là **ALGORITHM**.
- **Chúng ta sẽ tìm hiểu:**
 - Giải thuật theo cách hiểu thông thường
 - Các thao tác trong giải thuật
 - Định nghĩa giải thuật

- Theo nghĩa rộng, khái niệm “giải thuật” (algorithm) được sử dụng ở mọi nơi, không riêng gì trong lĩnh vực tin học.
- Giải thuật là một loạt các thao tác (operation) có thứ tự (order) nhằm giải quyết một bài toán nào đó.
- Ví dụ: “Thuật toán nấu cơm”
 - Bước 0: Ước lượng gạo cần thiết
 - Bước 1: Vo gạo
 - Bước 2: Cho gạo và nước thích hợp vào nồi cơm điện(NCĐ)
 - Bước 3: Cắm điện, chuyển chế độ “cook”
 - Bước 4: Chờ đến khi NCĐ chuyển sang chế độ “warm”
 - Bước 5: Chờ thêm 10 phút nữa
 - Bước 6: Cơm chín, kết thúc.

Các thao tác trong giải thuật

- Thao tác tuần tự (sequential operation): Một công việc đã được xác định rõ ràng, thực hiện xong thì chuyển sang công việc khác.
- Thao tác kiểm tra điều kiện (conditional operation): Kiểm tra điều kiện đưa ra có thoả mãn hay không để quyết định thao tác tiếp theo.
- Thao tác lặp (iterative operation): Quay trở lại bước nào đó trong dãy thao tác. Một thao tác có thể được lặp đi lặp lại nhiều lần tới khi một điều kiện nào đó được thoả mãn

Định nghĩa về giải thuật

- ***Giải thuật là một dãy các câu lệnh chặt chẽ và rõ ràng xác định một trình tự thao tác trên một đối tượng nào đó sao cho sau một số bước hữu hạn thực hiện, ta thu được kết quả mong muốn.***
- □ Câu lệnh (statement/instruction): đơn vị thao tác, tính toán, xử lý
- □ Trình tự rõ ràng (well-ordered): thực hiện xong bước này mới chuyển sang bước khác, không nhập nhằng.
- □ Đối tượng (object): các dữ kiện của bài toán, dữ liệu trung gian, kết quả,...
- □ Kết quả (result): Thông tin, lời giải cho bài toán,...

Yêu cầu đối với giải thuật

- Tính dừng: Một giải thuật bất kỳ phải đảm bảo dừng sau một số hữu hạn bước.
- Tính đúng đắn: Giải thuật phải đảm bảo giải quyết bài toán một cách đúng đắn, cho kết quả “chính xác” và “đầy đủ” theo yêu cầu.
- Tính đơn giản và hiệu quả
 - Đơn giản: Dễ hiểu, dễ lập trình.
 - Hiệu quả: Tiêu tốn ít thời gian và tài nguyên máy tính

Mọi bài toán đều có giải thuật ?

- Có những bài toán không có giải thuật tổng quát để giải quyết.
- Có những bài toán chưa có giải thuật hữu hiệu để giải quyết.
- Có những bài toán chưa có giải thuật tìm lời giải.

5.2. Trình bày và triển khai thuật toán

- Liệt kê từng bước
- Sơ đồ khối
- Sử dụng giả ngôn ngữ

5.2.1. Liệt kê từng bước

- Các thao tác của giải thuật được liệt kê từng bước
- Tại mỗi bước, sử dụng ngôn ngữ tự nhiên để diễn tả công việc phải làm.
- Bước đứng trước (có số thứ tự nhỏ hơn) được thực hiện trước.
- Ưu nhược điểm
 - Dễ hiểu, dễ làm
 - Phụ thuộc vào “cách hành văn” của người diễn đạt
 - Với những giải thuật phức tạp, cách diễn đạt này trở nên rườm rà
 - ...

Ví dụ 1: Giải phương trình bậc nhất $ax+b=0$

- 1. Nhập a, b .
- 2. Nếu $a=0$ và $b=0$ thì viết “Phương trình nghiệm đúng với mọi x ” rồi sang bước 5.
- 3. Nếu $a=0$ và $b \neq 0$ thì viết “Phương trình vô nghiệm” rồi sang bước 5.
- 4. Nếu $a \neq 0$ thì tính $x=-b/a$ rồi viết x là nghiệm.
- 5. Kết thúc

VD2: Nhập n phần tử $a_1..a_n$, sau đó tính tổng n số: $S=a_1+a_2+a_3+.....+a_n$

- Bước 1: Nhập n.
- Bước 2: Cho $S=0$ (lưu trữ số 0 trong S)
- Bước 3: Cho $i=1$ (lưu trữ số 1 trong i)
- Bước 4: Kiểm tra nếu $i \leq n$ thì thực hiện bước 5, ngược lại thực hiện bước 8.
- Bước 5: Nhập a_i
- Bước 6: Cho $S=S+a_i$ (lưu trữ giá trị $S + a_i$ trong S)
- Bước 7: Tăng i lên 1 đơn vị và quay lại bước 4.
- Bước 8: In S và kết thúc chương trình.

VD3: Viết GT nhập vào 2 giá trị a, b tìm số lớn hơn

- Bước 1: Nhập 2 số a và b
- Bước 2: Nếu $a > b$ thì viết a lớn hơn b, sang thực hiện bước 5, (còn không sang bước 3)
- Bước 3: Nếu $a = b$ thì viết $a = b$, sang thực hiện bước 5, (còn không sang bước 4).
- Bước 4: viết b lớn hơn a
- Bước 5: kết thúc.

VD4: Xây dựng TT nhập n giá trị a_1, a_2, \dots, a_n . in ra giá trị lớn nhất

- Bước 1: Nhập số n
- Bước 2: Nhập số thứ nhất a_1
- Bước 3: Gán $\text{max}=a_1$
- Bước 4: Gán $i=2$
- Bước 5: Nếu $i \leq n$ thì thực hiện bước 6, ngược lại thực hiện bước 9
- Bước 6: Nhập a_i
- Bước 7: Nếu $\text{max} < a_i$ thì gán $\text{max}=a_i$.
- Bước 8: Tăng i lên một đơn vị và quay lại bước 5
- Bước 9: In max - kết thúc

VD5: Xây dựng TT nhập n giá trị a_1, a_2, \dots, a_n . Sắp theo thứ tự tăng dần

- Bước 1: Gán $i=1$
- Bước 2: Gán $j=i+1$
- Bước 3: Nếu $i \leq n-1$ thì thực hiện bước 4, ngược lại thực hiện bước 8
- Bước 4: Nếu $j \leq n$ thì thực hiện bước 5, ngược lại thì thực hiện bước 7.
- Bước 5: Nếu $a_i > a_j$ thì hoán đổi a_i và a_j cho nhau (nếu không thì thôi).
- Bước 6: Tăng j lên một đơn vị và quay lại bước 4
- Bước 7: Tăng i lên một đơn vị và quay lại bước 3
- Bước 8: In dãy số a_1, a_2, \dots, a_n - Kết thúc.

5.2.2. Diễn đạt giải thuật bằng sơ đồ khối (block diagram)

- Sử dụng các hình khối để minh họa cho các lệnh hay thao tác.
- Sử dụng mũi tên để diễn đạt thứ tự thực hiện
- Đây là cách diễn đạt khoa học, có tính nhất quán cao

■ Sơ đồ khối (Lưu đồ)

□ Các ký hiệu được sử dụng trong lưu đồ:

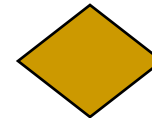
■ Nhập – Input



■ Xử lý – Process



■ Quyết định – Decision



■ Thủ tục – Procedure



■ Đường kết nối - Flow line



■ Giải thuật và Lưu đồ (tt)

□ Các ký hiệu được sử dụng trong lưu đồ (tt):

■ Bắt đầu và Kết thúc – Start and Stop



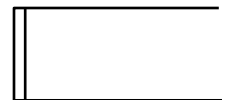
■ Kết nối trên cùng một trang – On page connector



■ Kết nối ở trang khác – Off page connector



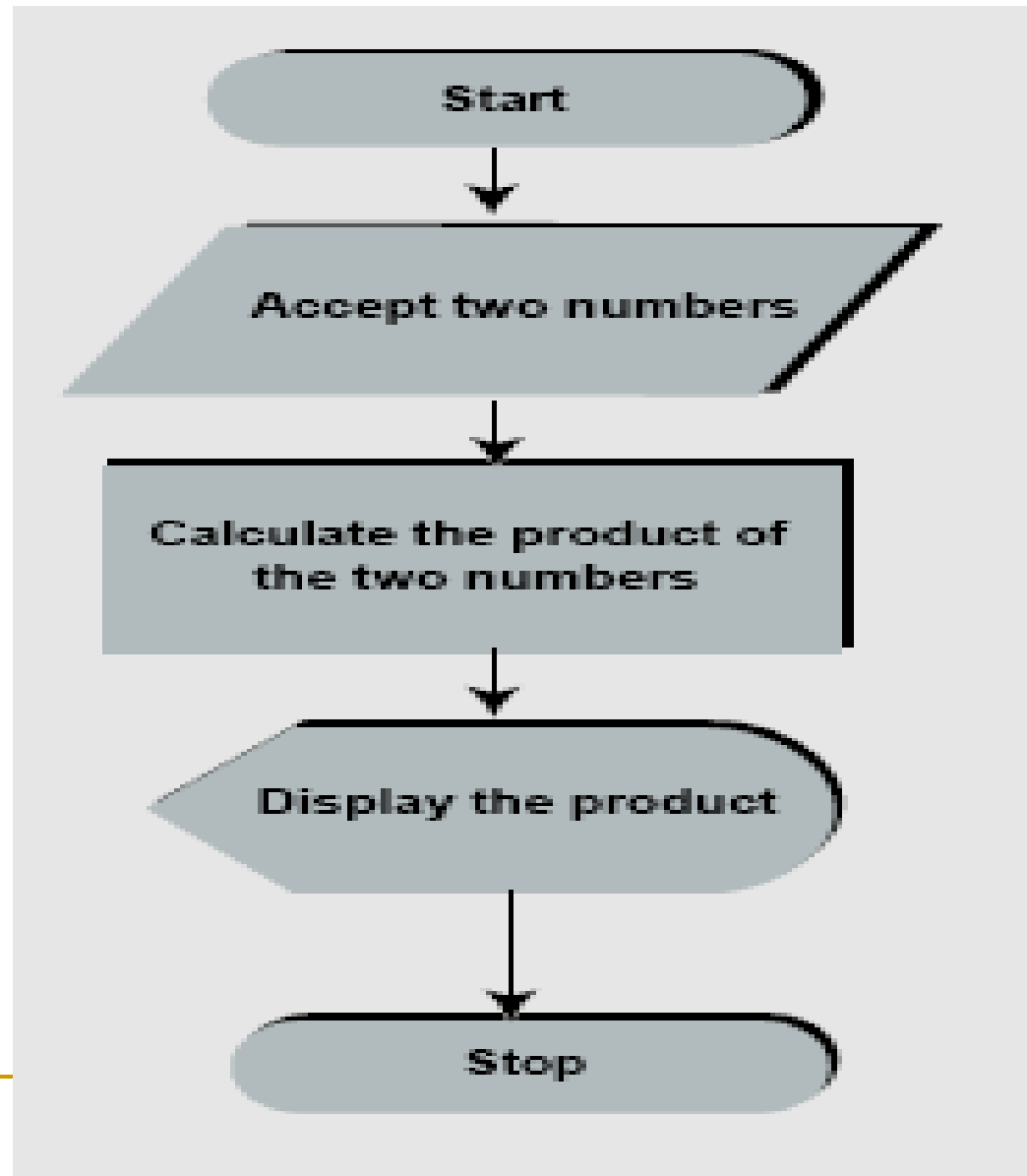
■ Chú thích – Annotation



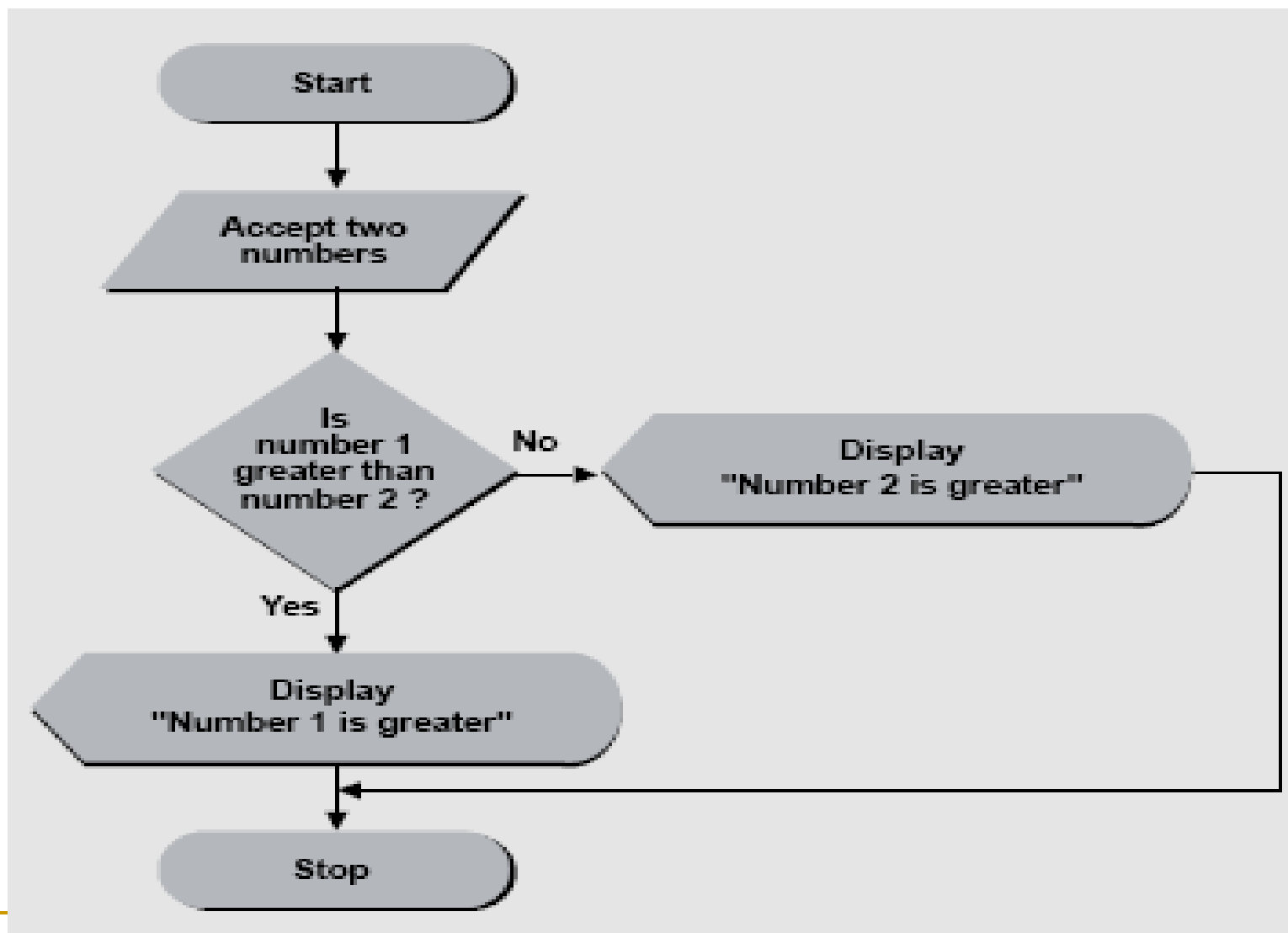
■ Hiển thị - Display/Output



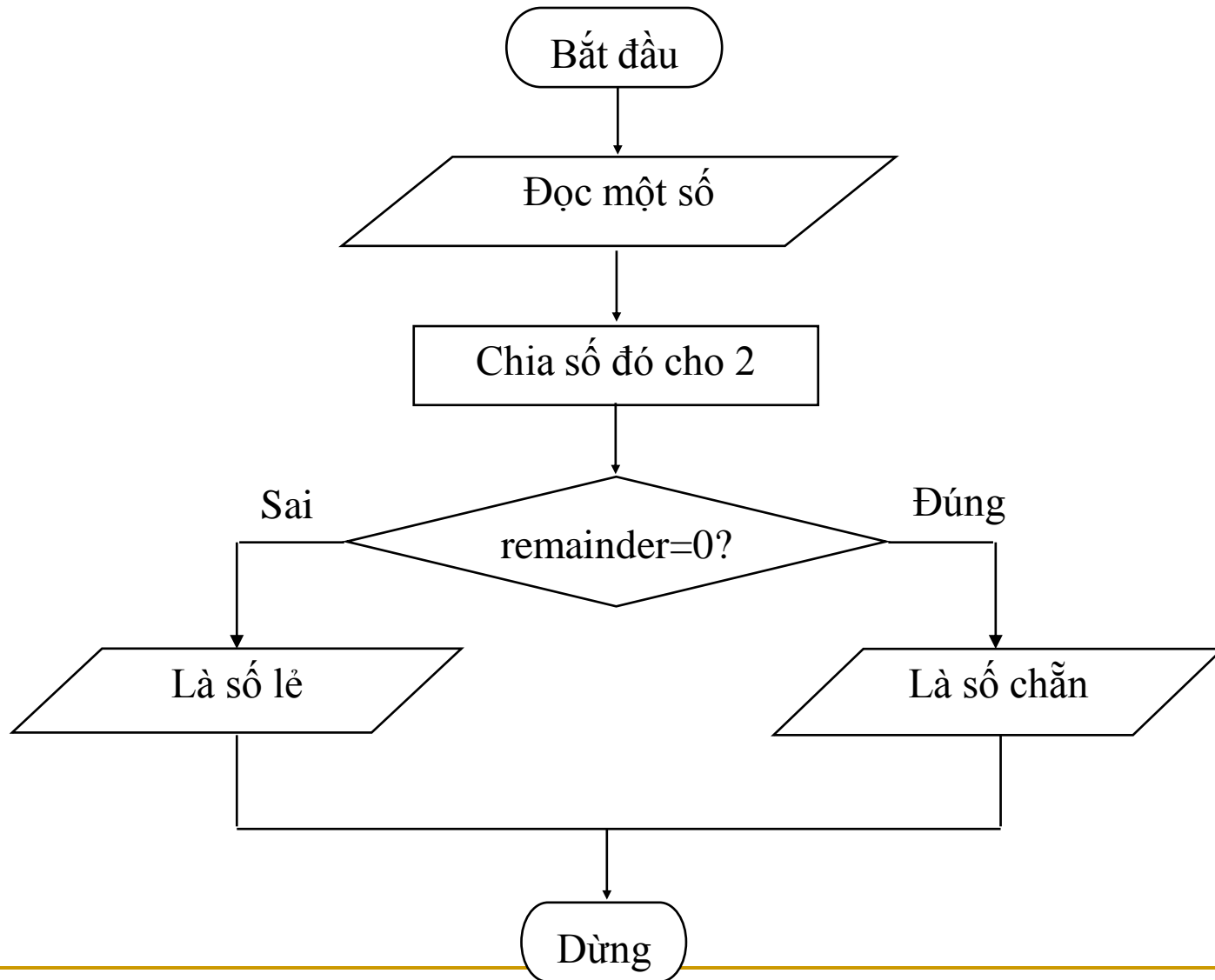
- Lưu đồ sau đây nhập hai số, tính tích và hiển thị tích của hai số.



Lưu đồ sau đây hiển thị số lớn hơn trong hai số đã nhập.



?



Bắt đầu

Hỏi giá trị a, b, c ?

$$\Delta = b^2 - 4ac$$

$\Delta > 0$

S

$\Delta = 0$

S

Đ

Có hai nghiệm
phân biệt x_1, x_2

Có nghiệm kép x_0

Vô nghiệm

$$x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a}$$

$$x_0 = \frac{-b}{2a}$$

Kết thúc

Các qui tắc vẽ lưu đồ

American National Standards Institute (ANSI) đề nghị một số qui luật cần phải tuân theo khi vẽ lưu đồ như sau:

- Toàn bộ lưu đồ nên được biểu diễn bằng ký hiệu chuẩn.
- Lưu đồ nên rõ ràng, chính xác và dễ theo dõi.
- Lưu đồ chỉ nên có một điểm bắt đầu và một điểm kết thúc.
- Các bước trong lưu đồ nên đi từ trên xuống và đi từ trái sang phải.
- Tất cả dữ liệu nhập phải được liệt kê hết theo một thứ tự hợp lý.

Các qui tắc vẽ lưu đồ

- Ký hiệu bắt đầu và kết thúc chỉ nên có một đường kết nối duy nhất.
- Ký hiệu nhập dữ liệu, xử lý, xuất dữ liệu, hiển thị nên có hai đường kết nối để nối ký hiệu đứng trước và đứng sau chúng.
- Ký hiệu quyết định nên có một đường kết nối với ký hiệu trước nó và có hai đường kết nối đến hai ký hiệu đứng sau nó.
- Nếu có quá nhiều đường kết nối làm cho lưu đồ trở nên phức tạp thì nên dùng ký hiệu kết nối trang để giảm số đường kết nối.

■ Điểm mạnh của lưu đồ

- ❑ Lưu đồ là phương pháp tốt để truyền đạt lập luận của giải thuật.
- ❑ Lưu đồ giúp phân tích vấn đề một cách hiệu quả.
- ❑ Lưu đồ đóng vai trò như người hướng dẫn trong giai đoạn phát triển chương trình.
- ❑ Dễ tìm và sửa lỗi bằng lưu đồ.
- ❑ Lưu đồ trợ giúp đắc lực trong việc bảo trì chương trình.

■ Điểm yếu của lưu đồ

- ❑ Lưu đồ dài có thể trải ra trên nhiều trang, làm giảm tính dễ đọc.
- ❑ Việc vẽ lưu đồ bằng các công cụ đồ họa là việc làm tốn nhiều thời gian.
- ❑ Thay đổi chỉ một bước nào đó có thể dẫn đến việc vẽ lại toàn bộ lưu đồ.
- ❑ Lưu đồ biểu diễn giải thuật phức tạp có thể chứa rất nhiều đường kết nối. Điều này làm giảm tính dễ đọc, tốn thời gian để vẽ và hiểu lập luận của giải thuật.

5.3.3. Diễn đạt giải thuật bằng ngôn ngữ giả

- Dựa trên ngôn ngữ lập trình bậc cao
- Gần với ngôn ngữ tự nhiên của con người
- Ví dụ: Ngôn ngữ giả Pascal (tựa Pascal) có các ký pháp khá giống với ngôn ngữ lập trình Pascal, được rút gọn sao cho dễ diễn đạt
- Giả ngôn ngữ được đưa ra với mục đích diễn đạt các giải thuật sao cho gần với ngôn ngữ lập trình và ngôn ngữ tự nhiên.
- Sử dụng giả ngôn ngữ khiến việc chuyển từ giải thuật sang chương trình dễ dàng hơn.

Ví dụ: Giải thuật tính tổng N số tự nhiên đầu tiên

- **Nhập N**
- **$i:=0$**
- **$S:=0$**
- **REPEAT**
- **$S:=S+i$**
- **$i:=i+1$**
- **UNTIL ($i>N$)**
- **In S**

5.3. Đánh giá thuật toán

- Phân tích thuật toán: tìm một cách đánh giá về thời gian và “không gian” cần thiết để thực hiện thuật toán
- Đánh giá về thời gian của thuật toán không phải là xác định thời gian tuyệt đối mà là mối liên quan giữa dữ liệu đầu vào và chi phí để thực hiện thuật toán $T=f(\text{input})$
- Thông thường chỉ quan tâm đến mối liên quan giữa *độ lớn* của dữ liệu đầu vào và chi phí $T=f(n)$ trong trường hợp *tốt nhất* và *xấu nhất*

Độ phức tạp của thuật toán

Độ phức tạp của thuật toán

- Chi phí chung = chi phí so sánh + chi phí gán
- Trong mọi trường hợp, phép so sánh $a_i > a_{\max}$ luôn được thực hiện và số lần thực hiện là $n-1$: chi phí cố định (bất biến) của bài toán
- Trường hợp tốt nhất: số lớn nhất ở đầu dãy. Do đó phép gán $a_{\max} = a_i$ không cần thực thi, như vậy chi phí chung cho trường hợp này chính là chi phí cố định $T=f(n)=n-1$
- Trường hợp xấu nhất: số lớn nhất ở cuối dãy. Do đó phép gán $a_{\max} = a_i$ luôn cần thực thi, số lần thực thi là $n-1$, như vậy chi phí chung cho trường hợp này là $T=f(n)=2(n-1)=2n-2$
- Độ phức tạp của thuật toán là $O(n)$, tuyến tính

5.4. Xây dựng chương trình

1. Xác định vấn đề - bài toán
2. Lựa chọn phương pháp giải
3. Xây dựng thuật toán hoặc thuật giải
4. Cài đặt chương trình
5. Hiệu chỉnh chương trình
6. Thực thi chương trình

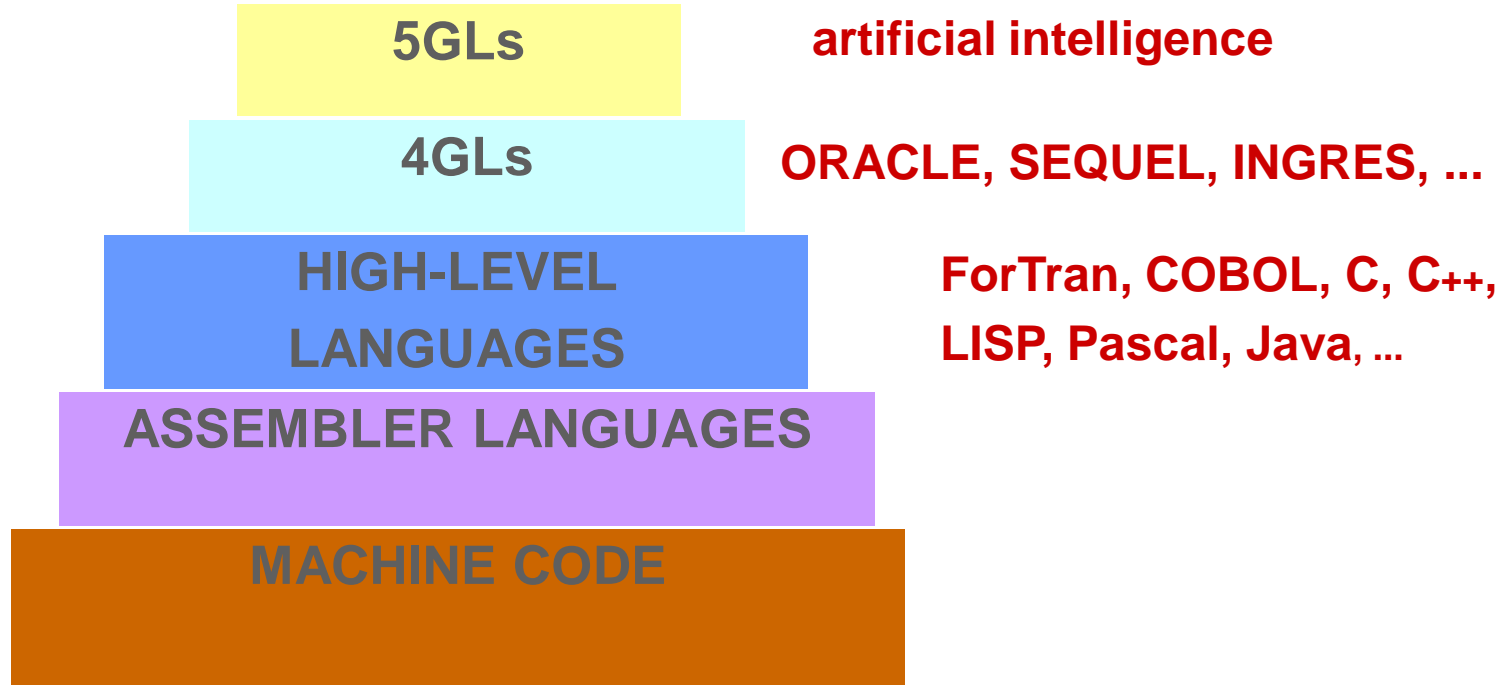
Phương pháp lập trình

- Lập trình cấu trúc có các đặc trưng
 - Tính đơn thể
 - Cấu trúc điều khiển: phối hợp cấu trúc tuần tự, cấu trúc chọn và cấu trúc lặp
 - Tính vào/ra đơn
- Lập trình hướng đối tượng: một tiếp cận mới trong lập trình sử dụng khái niệm đối tượng, kết hợp cả dữ liệu và các câu lệnh của chương trình

Ngôn ngữ lập trình

- Tập từ ngữ và cú pháp cho phép lập trình viên hoặc người dùng có thể nói chuyện với máy tính
- Chương trình máy tính là tập các chỉ thị nhằm thực hiện nhiệm vụ cụ thể được viết bằng ngôn ngữ lập trình
- Các loại ngôn ngữ lập trình
 - Ngôn ngữ máy
 - Hợp ngữ
 - Ngôn ngữ cấp cao

Ngôn ngữ lập trình



- Ngôn ngữ lập trình

- Ngôn ngữ lập trình là phương tiện truyền thông giữa Lập trình viên và máy tính.
- Lập trình viên sử dụng ngôn ngữ lập trình để viết các lệnh yêu cầu máy tính thực hiện một công việc nào đó. Tập hợp các lệnh này được gọi là chương trình.
- Các loại ngôn ngữ lập trình:

Ngôn ngữ máy:

- Từ vựng được tạo nên từ mã nhị phân 0-1 để tạo ra các câu lệnh, các chỉ dẫn nhằm yêu cầu máy tính thực hiện một công việc nào đó.
- Các lệnh được máy tính thực hiện trực tiếp (không qua mã trung gian).
- Ngoài ngôn ngữ máy, các NNLT khác đều cần được dịch ra mã trung gian trước khi thực thi.

Ngôn ngữ Assembly:

- Còn gọi là ngôn ngữ máy tính thế hệ thứ 2 là ngôn ngữ sử dụng bảng mã chữ cái và các ký hiệu để đưa ra các chỉ dẫn

Ngôn ngữ cấp cao:

- Đây là ngôn ngữ thân thiện với người sử dụng vì dùng những từ ngữ thông dụng tiếng Anh để đưa ra các chỉ dẫn cho máy tính ví dụ như PRINT, GOTO

-
- Ví dụ về các ngôn ngữ lập trình cấp cao
 - Một số ngôn ngữ lập trình cao cấp là:
 - BASIC
 - Pascal
 - C
 - Java
 - C#

■ Lựa chọn ngôn ngữ lập trình

Một số các yếu tố khi lựa chọn ngôn ngữ lập trình:

- Tiêu chí 1: lựa chọn đầu tiên là tùy thuộc vào loại ứng dụng cần phát triển.

Ví dụ:

- Tạo ra chương trình thực hiện các phép toán đơn giản (+ - * /) ta có thể sử dụng Pascal.
- Để phát triển ứng dụng dùng trong gia đình hoặc doanh nghiệp nhỏ có thể dùng Visual BASIC
- Tiêu chí 2: Nếu có nhiều ngôn ngữ phù hợp để phát triển ứng dụng thì bạn nên chọn ngôn ngữ mà bạn thông thạo nhất.
- Tiêu chí 3: Nếu bạn không thông thạo ngôn ngữ nào cả thì nên chọn ngôn ngữ lập trình dễ học và dễ sử dụng.

Cài đặt chương trình từ 1 thuật toán

- Ví dụ: Chuyển một số giải thuật cơ bản
 - Lặp
 - Đệ quy
 - Sắp xếp
 - Tìm kiếm
 - ...

Tóm tắt và một số câu hỏi

- Tại sao cần có thuật toán ?
- Một số phương pháp biểu diễn thuật toán?
- Những yêu cầu đối với 1 thuật toán ?
- Biểu diễn thuật toán tìm số lớn nhất giữa 2 số a và b ?
- Biểu diễn thuật toán giải PT bậc 2?
- Tìm giá trị lớn nhất của 1 dãy số ?
- Đánh giá độ phức tạp và cài đặt 3 thuật toán trên bằng C hoặc Pascal ?

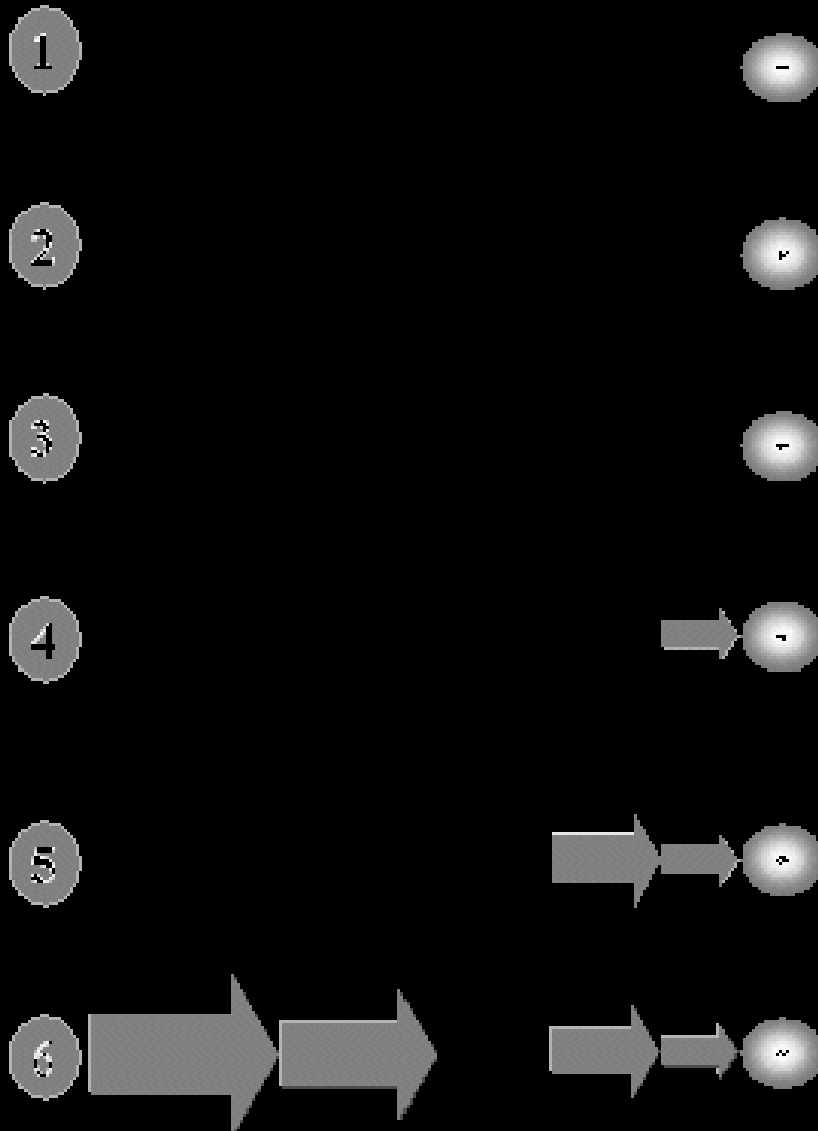
Thuật toán đệ quy

- Đưa bài toán hiện tại về một bài toán cùng loại, cùng tính chất (đồng dạng) nhưng ở cấp độ thấp hơn và quá trình này tiếp tục cho đến lúc bài toán được đưa về một cấp độ mà tại đó có thể giải được. Từ kết quả ở cấp độ này, lần ngược để giải được bài toán ở cấp độ cao đến lúc ban đầu
 - Định nghĩa giai thừa: $0! = 1$, $n! = (n-1)!n$ với mọi $n > 0$
 - Định nghĩa dãy số Fibonacci: $f_0 = 1$, $f_1 = 1$, $f_n = f_{n-1} + f_{n-2}$ với mọi $n > 1$
- Định nghĩa theo kiểu quy nạp

Thuật toán đệ quy

- Mọi thuật toán đệ quy gồm hai phần:
 - Phần cơ sở: các trường hợp không cần thực hiện lại thuật toán (không gọi đệ quy), chính là các điểm dừng
 - Phần đệ quy: có yêu cầu gọi đệ quy, tức yêu cầu thực hiện lại thuật toán với cấp độ dữ liệu thấp hơn

Thuật toán đệ quy



Trình thông dịch và biên dịch

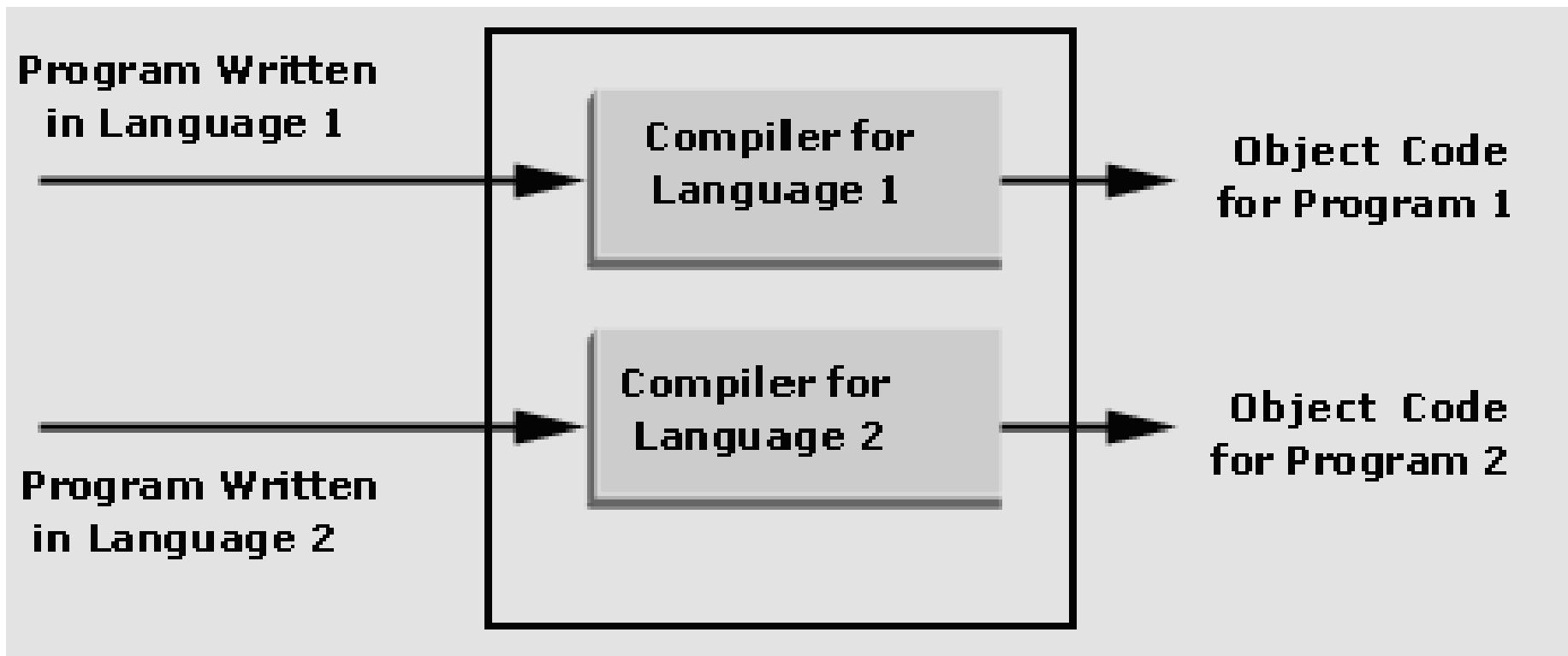
- Mọi chương trình đều phải được chuyển đổi qua ngôn ngữ máy trước khi thực thi
- Trình hợp dịch (assembler)
- Trình biên dịch (compiler): chuyển đổi toàn bộ chương trình cấp cao nguồn sang chương trình đối tượng để thực thi
- Trình thông dịch (interpreter): chuyển đổi từng mệnh đề và thực thi đoạn mã kết quả ngay, không tạo ra chương trình đối tượng

■ Trình biên dịch

- Là chương trình (phần mềm) được sử dụng để chuyển các câu lệnh được viết bằng ngôn ngữ cấp cao sang ngôn ngữ máy tính có thể hiểu được.
- Mỗi ngôn ngữ lập trình cấp cao có một trình biên dịch riêng tương ứng.

■ Trình biên dịch (tiếp theo)

- Hình vẽ sau thể hiện sự hoạt động của trình biên dịch:



■ Trình biên dịch (tiếp theo)

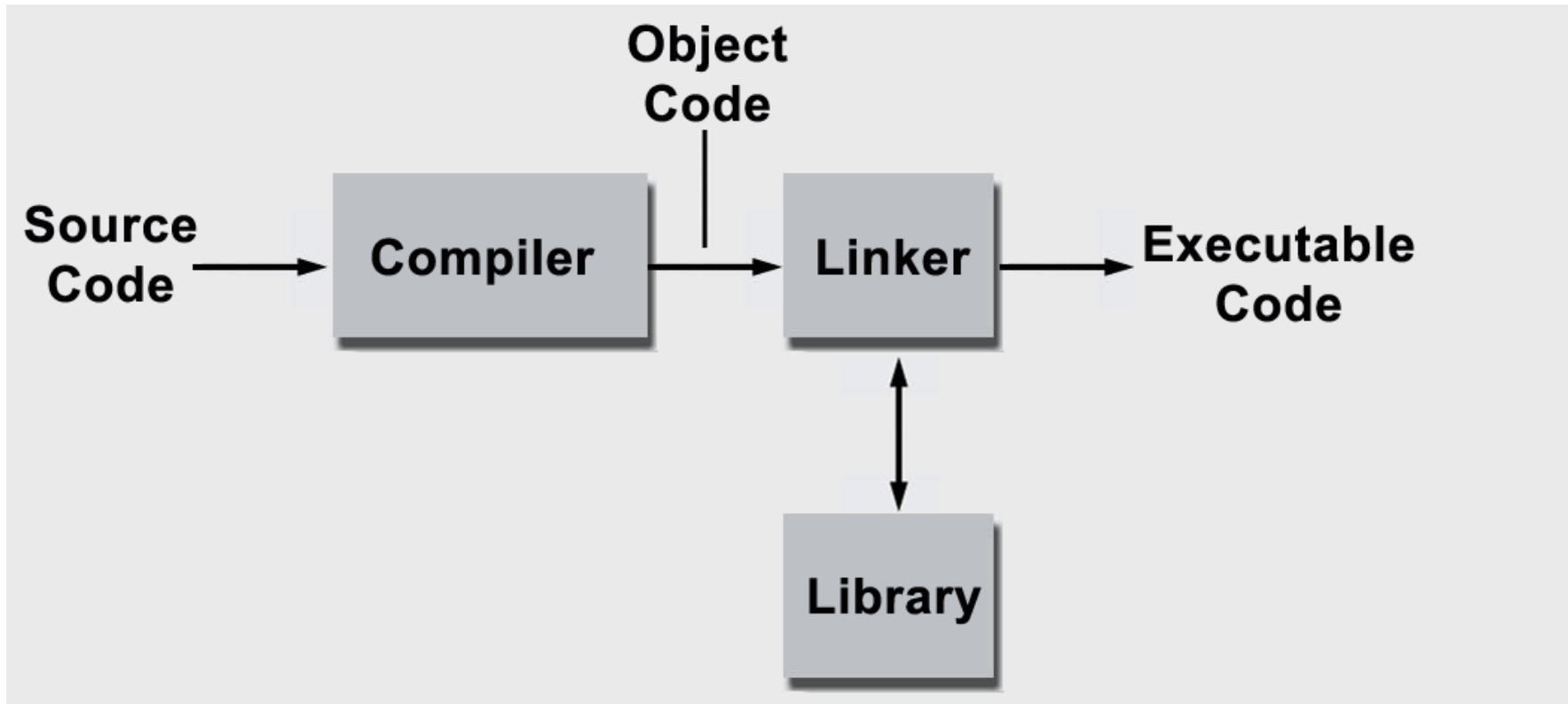
- ❑ Khi trình biên dịch dịch một chương trình, nó sẽ kiểm tra Từ vựng và Cú pháp các câu lệnh.
- ❑ Nếu trình biên dịch tìm thấy lỗi trong chương trình nguồn thì nó sẽ hiển thị danh sách các lỗi.
- ❑ Trình biên dịch sẽ không sinh ra mã trung gian nếu như tất cả các lỗi của chương trình nguồn chưa được sửa hết.

■ Bộ liên kết

- ❑ Mỗi NNLT sẽ có các thư viện chứa các từ khóa với các chức năng được định nghĩa trước. Người lập trình sẽ sử dụng những từ khóa này kết hợp với các cú pháp để viết chương trình.
- ❑ Khi dịch, trình biên dịch sẽ sinh ra mã trung gian.
- ❑ Để sinh mã thực thi từ mã trung gian thì chức năng của các từ khóa (lưu trong các thư viện) và các thư viện cũng cần được đưa vào mã trung gian.
- ❑ Bộ liên kết thực hiện việc gắn chức năng của các từ khóa với mã trung gian để tạo ra mã thực thi.

■ Bộ liên kết (tiếp theo)

- Quá trình liên kết này được thể hiện bằng hình vẽ sau đây:



■ Trình thông dịch

Một số ngôn ngữ lập trình cấp cao sử dụng cơ chế dịch khác gọi là thông dịch. Chương trình dịch theo cơ chế này được gọi là trình thông dịch.

- ❑ Trình thông dịch sẽ đọc một lệnh được viết bằng ngôn ngữ lập trình cấp cao, chuyển sang mã máy, thực thi câu lệnh này và không lưu mã trung gian.
- ❑ Chương trình thực thi dùng trình thông dịch sẽ tiết kiệm được thời gian hơn trình biên dịch.
- ❑ Trình thông dịch cũng giúp cho việc dò tìm lỗi dễ dàng hơn vì nó chỉ ra chính xác những dòng nào phát sinh lỗi.
- ❑ ~~Ngôn ngữ lập trình sử dụng trình thông dịch như PERL, BASIC, Visual Basic, v.v..~~

GIỚI THIỆU KHOA HỌC MÁY TÍNH



NGUYỄN THANH TRUNG



Chương 6 – GIỚI THIỆU CNPM

- 6.1. Giới thiệu chung
- 6.2. Các mô hình phát triển
- 6.3. Các giai đoạn phát triển
- 6.4. Ngành công nghiệp phần mềm ở Việt nam



6.1. Giới thiệu chung

- Phần mềm ?

Là chương trình máy tính & sản phẩm liên quan

- Sản phẩm phần mềm ?

Có thể xây dựng theo yêu cầu riêng của khách hàng →
Sp theo đơn đặt hàng; hay yêu cầu chung của thị trường → SP phổ biến

- Công nghệ Phần mềm ?

Là ngành công nghệ có liên quan đến tất cả các khía cạnh của SP phần mềm



- Phát triển PM ?

- 1 nhóm người

- làm việc theo kỷ luật

- làm ra một sản phẩm hay hệ thống mô tả nghiệp vụ

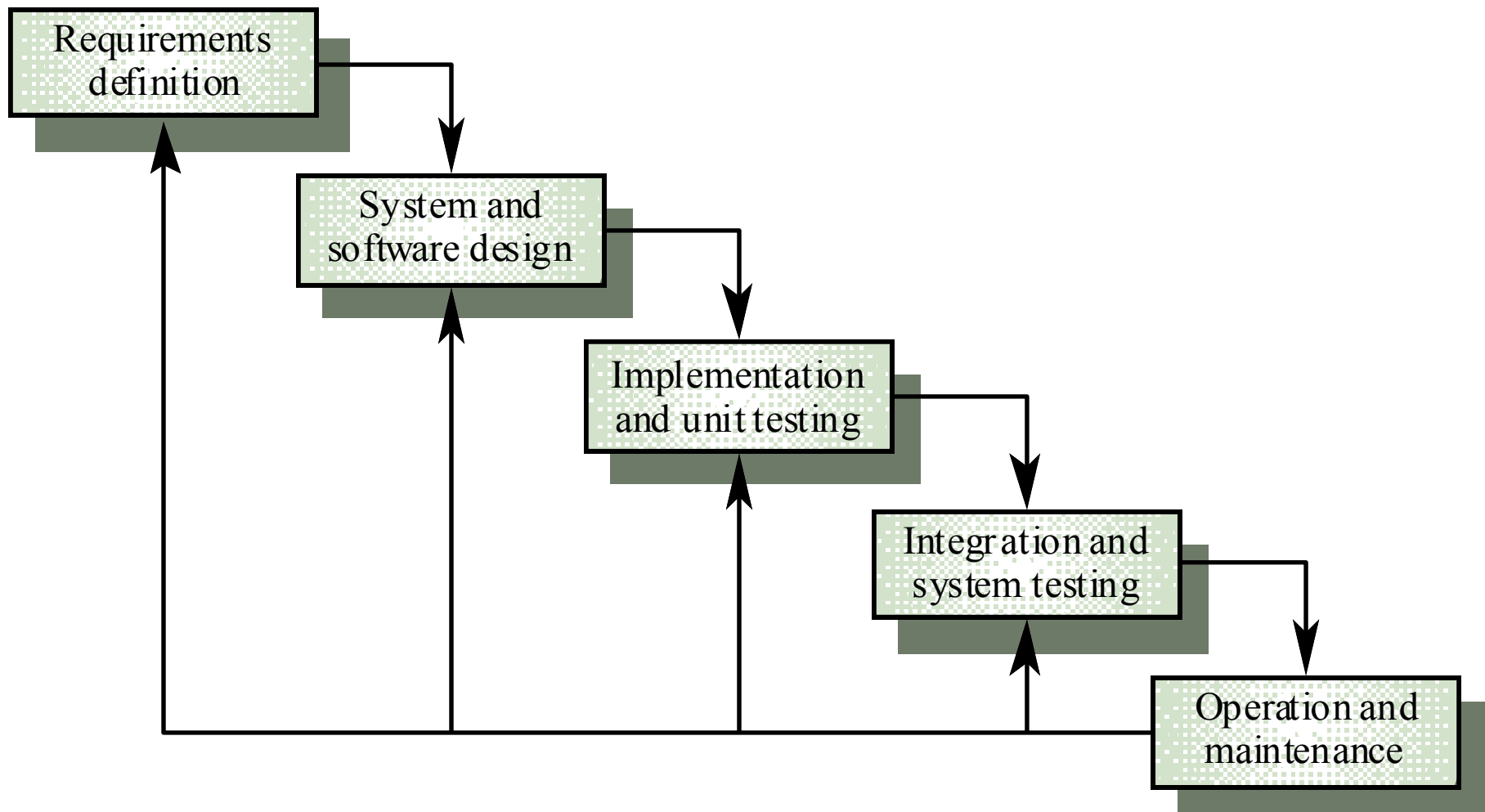
➔ (học nghiệp vụ, học cách làm mô hình, học cách tìm giải pháp tổng thể, học cách biểu diễn giải pháp trong phần mềm máy tính).



6.2. Các mô hình phát triển

- Giúp cho mỗi người biết mình, các thành viên và nhóm khác, đang ở đâu và phải làm gì trong tiến trình phát triển.
- Giúp việc quản lý dự án chặt chẽ và bảo đảm chất lượng PM
- Các mô hình cơ bản: Thác nước, phát triển tăng dần (tiến hóa), bản mẫu, xoắn ốc

6.2.1. Mô hình thác nước





Các giai đoạn của mô hình thác nước

- Phân tích và định nghĩa các yêu cầu
- Thiết kế hệ thống và phần mềm
- Cài đặt và test các đơn vị chương trình
- Tích hợp và test hệ thống
- Hoạt động và bảo trì
- Mặt hạn chế của mô hình thác nước là hiệu chỉnh (thay đổi) khó khăn khi quá trình được phát triển



Hạn chế

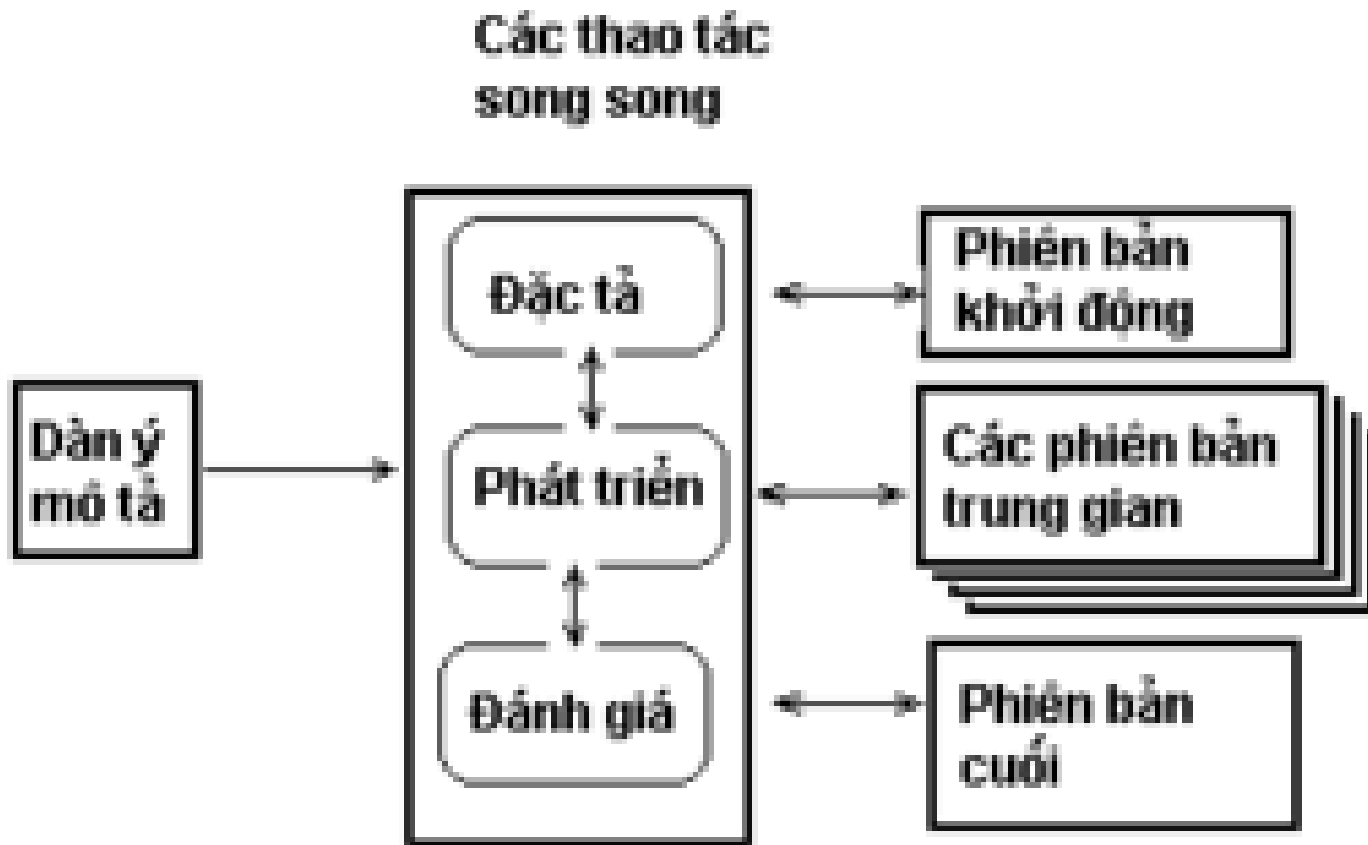
- Không có tính uyển chuyển trong các giai đoạn của mô hình
- Khó khăn khi khách hàng có sự thay đổi yêu cầu
- Chính vì vậy, mô hình này chỉ phù hợp với những yêu cầu đã được nắm vững một cách trọn vẹn



6.2.2. Mô hình tiến hóa

- Triển khai thăm dò: Mục đích là làm việc với khách hàng và đưa ra hệ thống cuối cùng từ một đặc tả phác thảo ban đầu. Khởi đầu từ những yêu cầu đã được cách giải quyết một cách thấu đáo
- Tinh chỉnh:
Mục đích: Hiểu chính xác những yêu cầu của hệ thống. Bắt đầu từ những yêu cầu mơ hồ chưa có cách giải quyết rõ ràng

Mô hình tiến hóa



Mô hình phát triển phần mềm theo kiểu tiến hoá



Mô hình tiến hóa

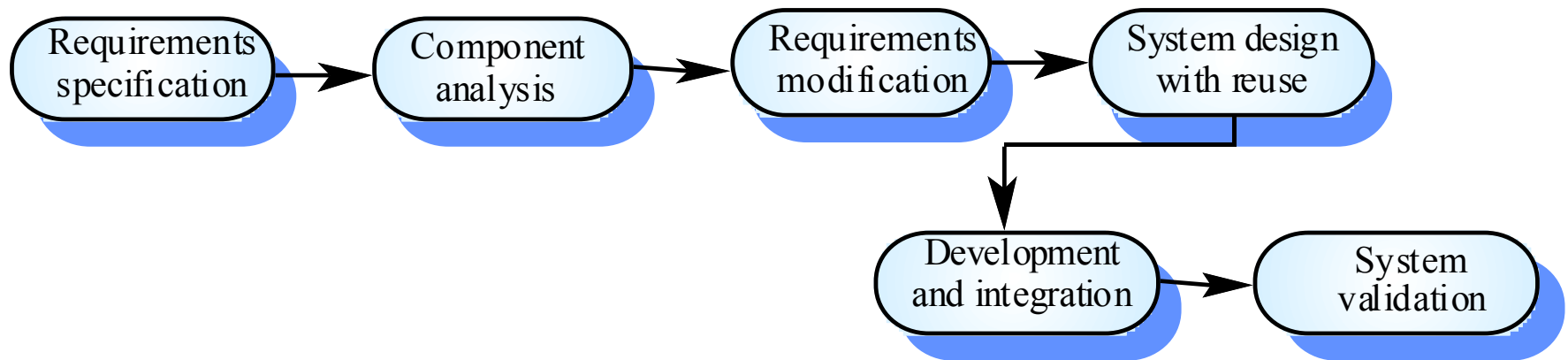
- Hạn chế
 - Không hình dung được tiến trình
 - Hệ thống không có tính mở
 - Những kỹ năng đặc biệt có thể được yêu cầu
- Áp dụng được:
 - Những hệ thống nhỏ và vừa
 - Những phần nhỏ của hệ thống lớn (VD: Giao diện)
 - Những hệ thống có tuổi thọ ngắn



6.2.3. Triển khai theo hướng sử dụng lại

- Dựa vào việc sử dụng lại một cách có hệ thống và hệ thống sẽ được tích hợp từ những thành phần sẵn có
- Những giai đoạn triển khai
 - Phân tích các thành phần
 - Hiệu chỉnh phù hợp yêu cầu
 - Thiết kế hệ thống với tính sử dụng lại
 - Triển khai và tích hợp
- Áp dụng được với những kinh nghiệm và hệ thống component phong phú

Triển khai theo hướng sử dụng lại

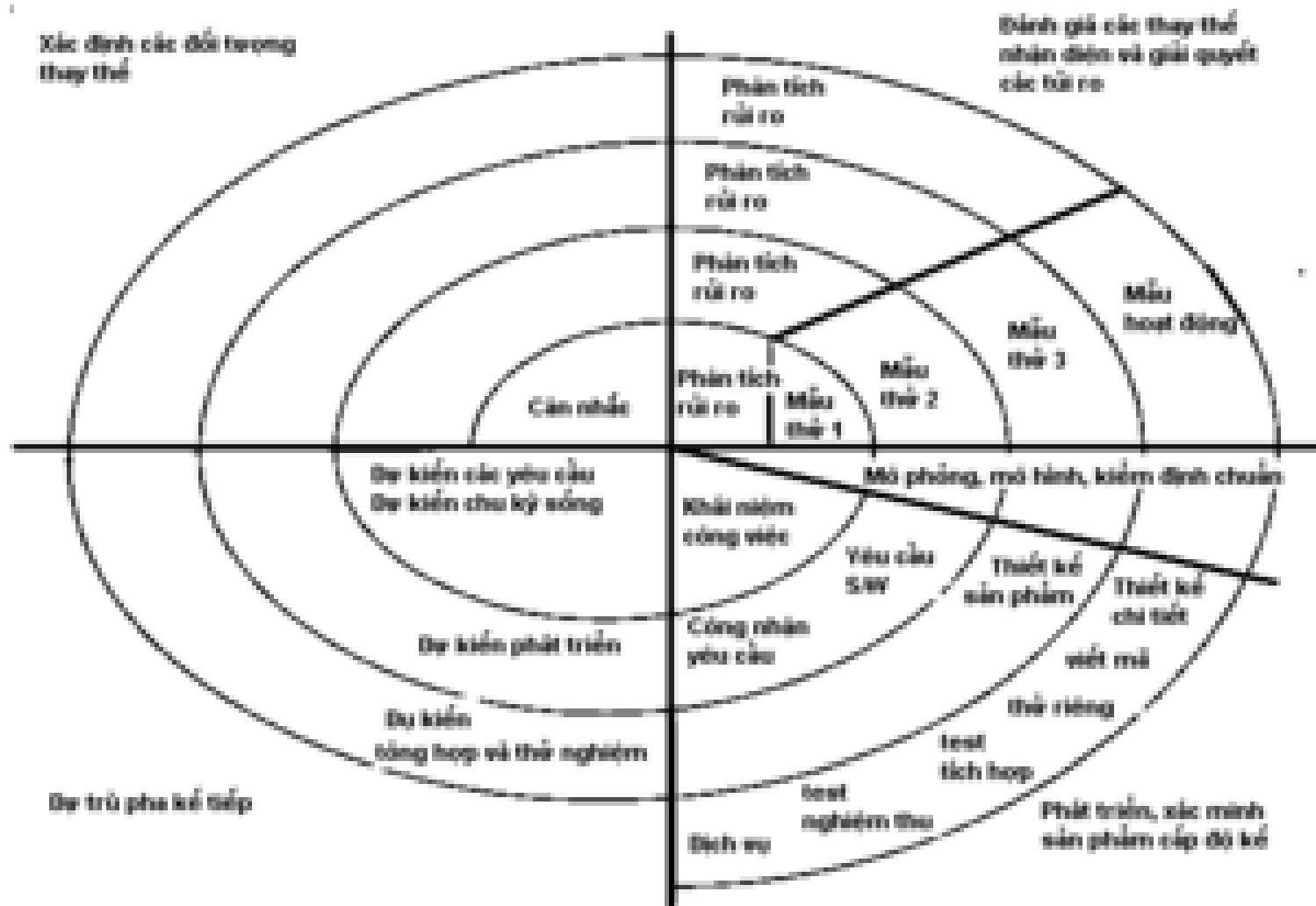


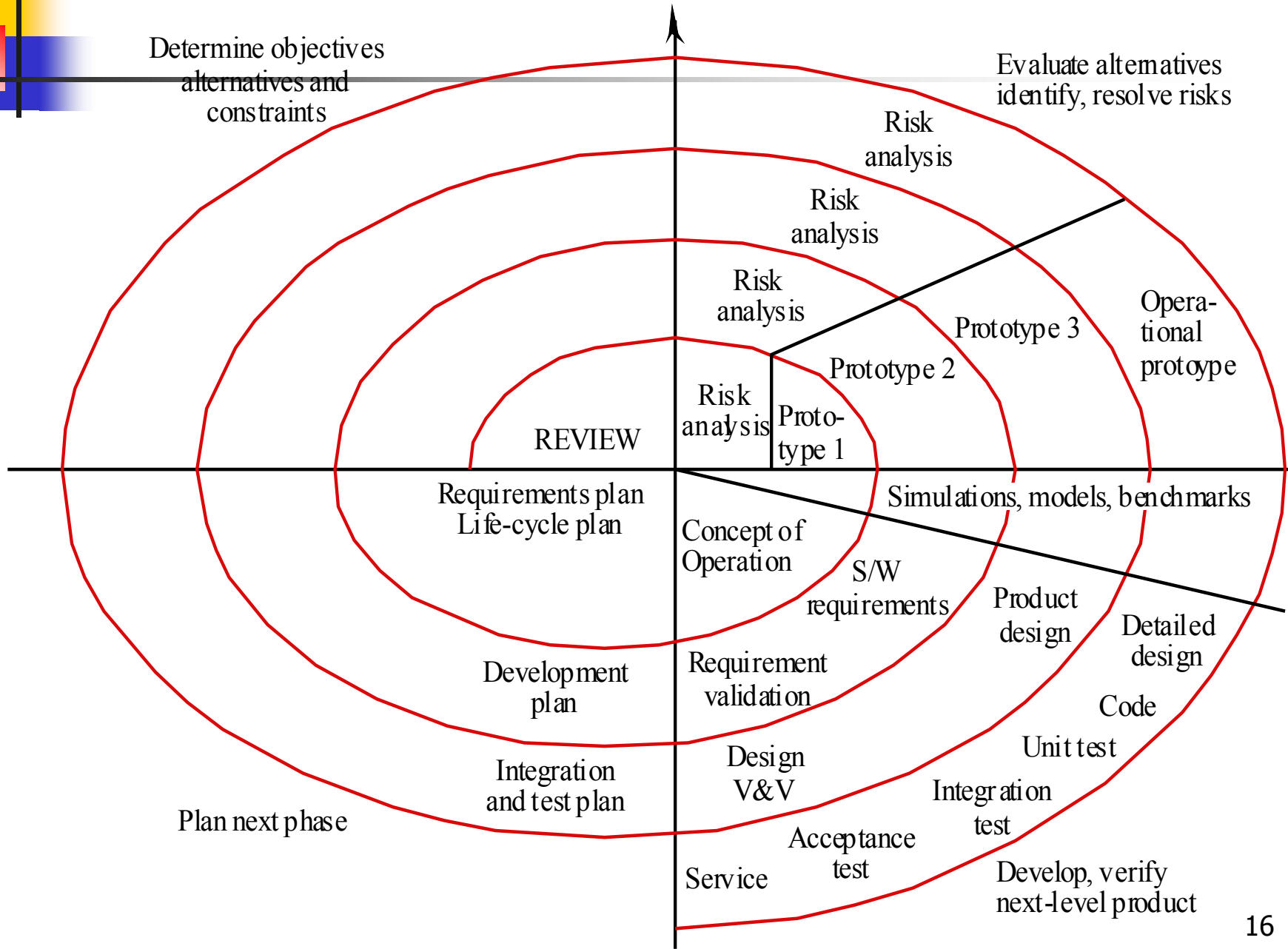
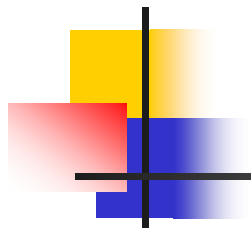


6.2.4. Mô hình xoắn ốc

- Quá trình thể hiện như một hình xoắn ốc hơn là một dãy tuần tự các hành động
- Mỗi vòng lặp trong hình xoắn ốc thể hiện là một chu kỳ phần mềm trong tiến trình phát triển phần mềm.
- Không xác định trước số lần lặp lại trong mô hình
- Rủi ro được ước lượng và xử lý ngay trong quá trình thực hiện

Mô hình xoắn ốc







Các cung trong Xoắn ốc

- Xác định mục tiêu
 - Những mục tiêu cụ thể của hệ thống
- Xác định và giảm thiểu rủi ro
 - Các rủi ro được xác định và khắc phục ngay trong quá trình triển khai
- Phát triển và chứng thực
 - Phát triển hệ thống được chọn lựa theo một mô hình nào đó
- Lập kế hoạch
 - Khái quát về những công việc thực hiện trong giai đoạn tới trong hình xoắn ốc



6.3. Các giai đoạn phát triển

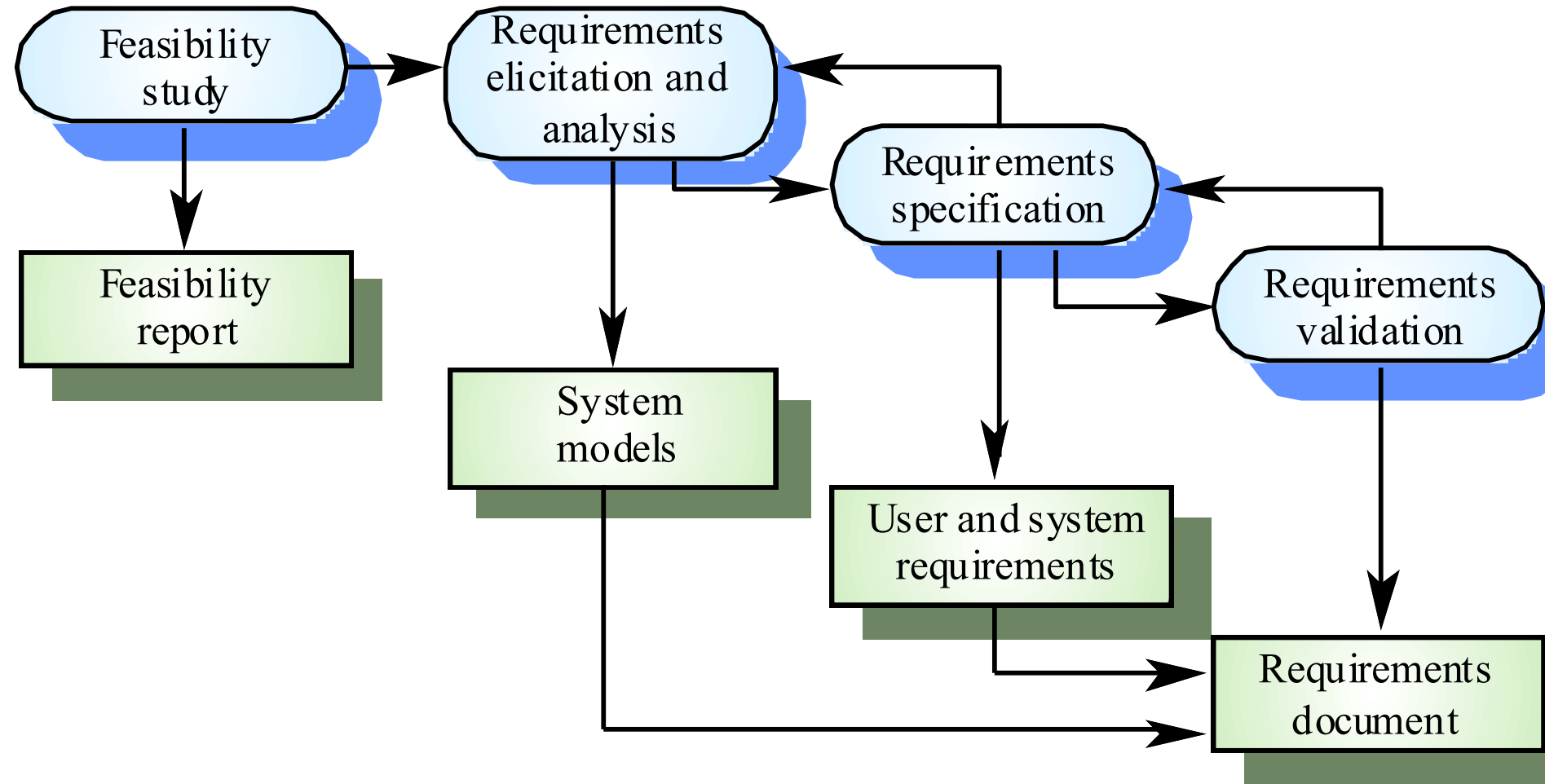
- Đặc tả
- Phân tích
- Thiết kế
- Lập trình
- Kiểm thử
- Vận hành và bảo trì



Đặc tả hệ thống phần mềm

- Thiết lập những yêu cầu và ràng buộc triển khai của hệ thống
- Những yêu cầu kỹ thuật
 - Nghiên cứu khả thi
 - Nêu lên và phân tích các yêu cầu
 - Đặc tả yêu cầu
 - Kiểm chứng các yêu cầu

Phân tích yêu cầu





Software design and implementation

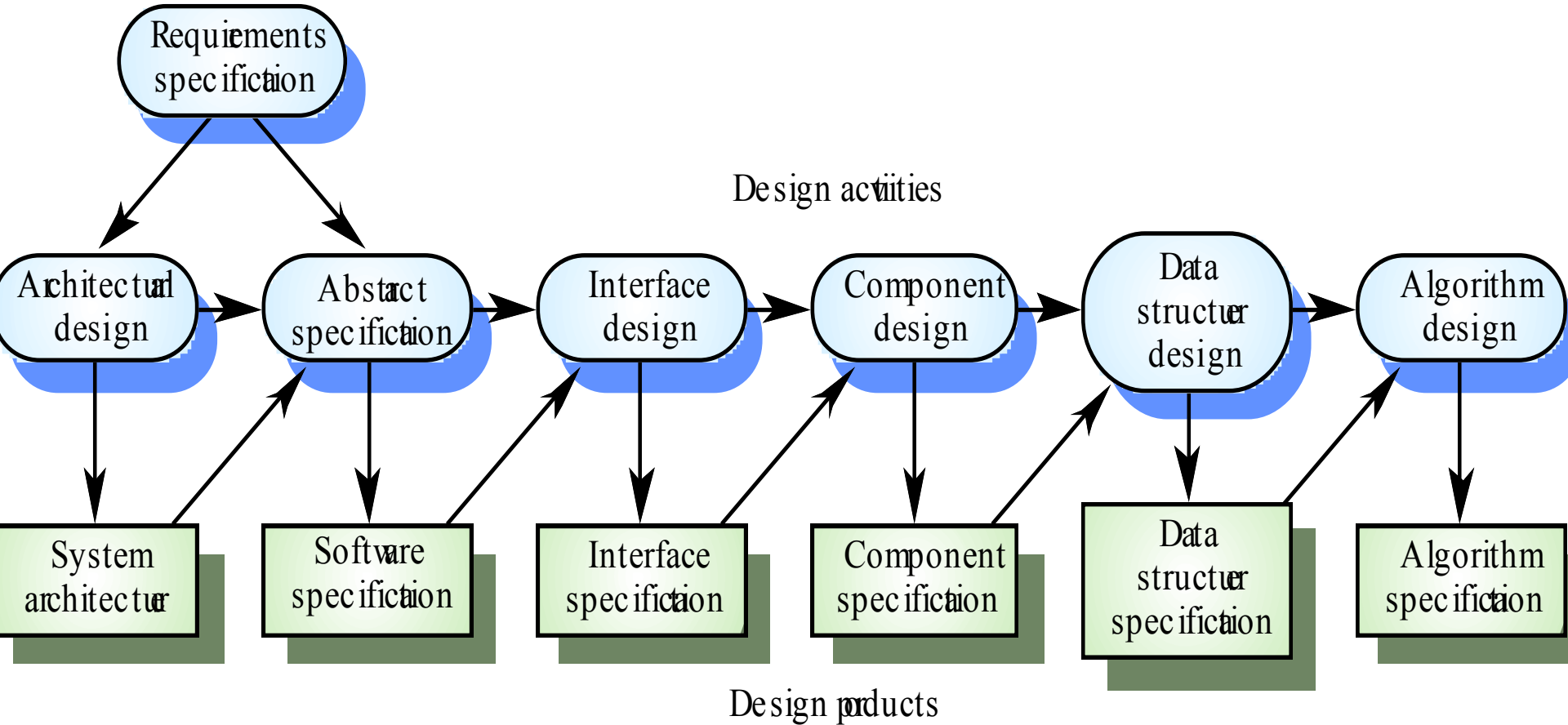
- Tiến trình chuyển đổi đặc tả hệ thống thành hệ thống thi hành được
 - Thiết kế cấu trúc phần mềm: Những đặc tả thi hành được
 - Hiện thực: Chuyển đổi cấu trúc phần mềm thành chương trình
- Hoạt động thiết kế và hiện thực có mối quan hệ chặt chẽ với nhau



Thiết kế

- Thiết kế kiến trúc
- Đặc tả trừu tượng
- Thiết kế giao diện
- Thiết kế thành phần
- Thiết kế cấu trúc dữ liệu
- Thiết kế thuật giải

Tiến trình thiết kế phần mềm





Phương pháp thiết kế

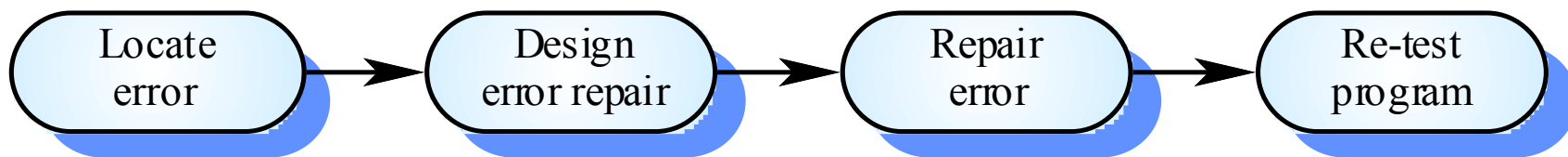
- Tiếp cận một cách có hệ thống để triển khai thiết kế hệ thống phần mềm
- Thiết kế thường được mô tả như một bộ sưu tập các mô hình đồ họa
- Các mô hình
 - Mô hình dòng dữ liệu
 - Mô hình thuộc tính thực thể kết hợp
 - Mô hình cấu trúc
 - Mô hình đối tượng



Lập trình và tìm lỗi

- Chuyển đổi từ thiết kế thành chương trình và khắc phục lỗi
- Lập trình là một quá trình mang tính con người, không có công cụ phát sinh tự động
- Người lập trình kiểm tra chương trình và khắc phục lỗi (debug)

Tìm lỗi và xử lý

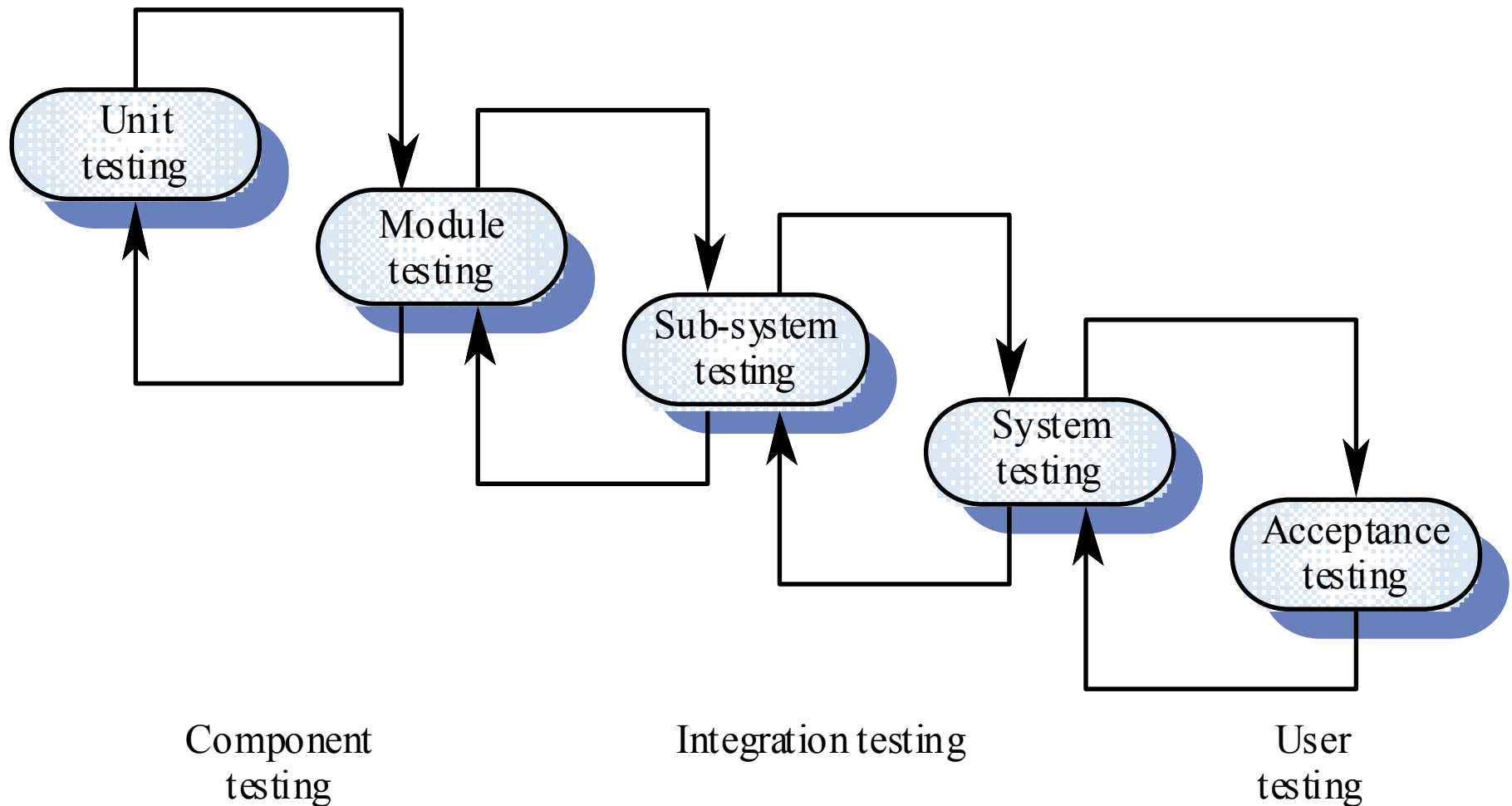




Chứng thực

- Chứng thực và xác nhận các yêu cầu của khách hàng được bảo đảm trong hệ thống.
- Cần phải kiểm tra và xem xét lại tất cả các tiến trình và toàn hệ thống
- Kiểm tra toàn bộ hệ thống với các tình huống được đưa ra từ đặc tả với những bộ số liệu thực

Giai đoạn kiểm thử

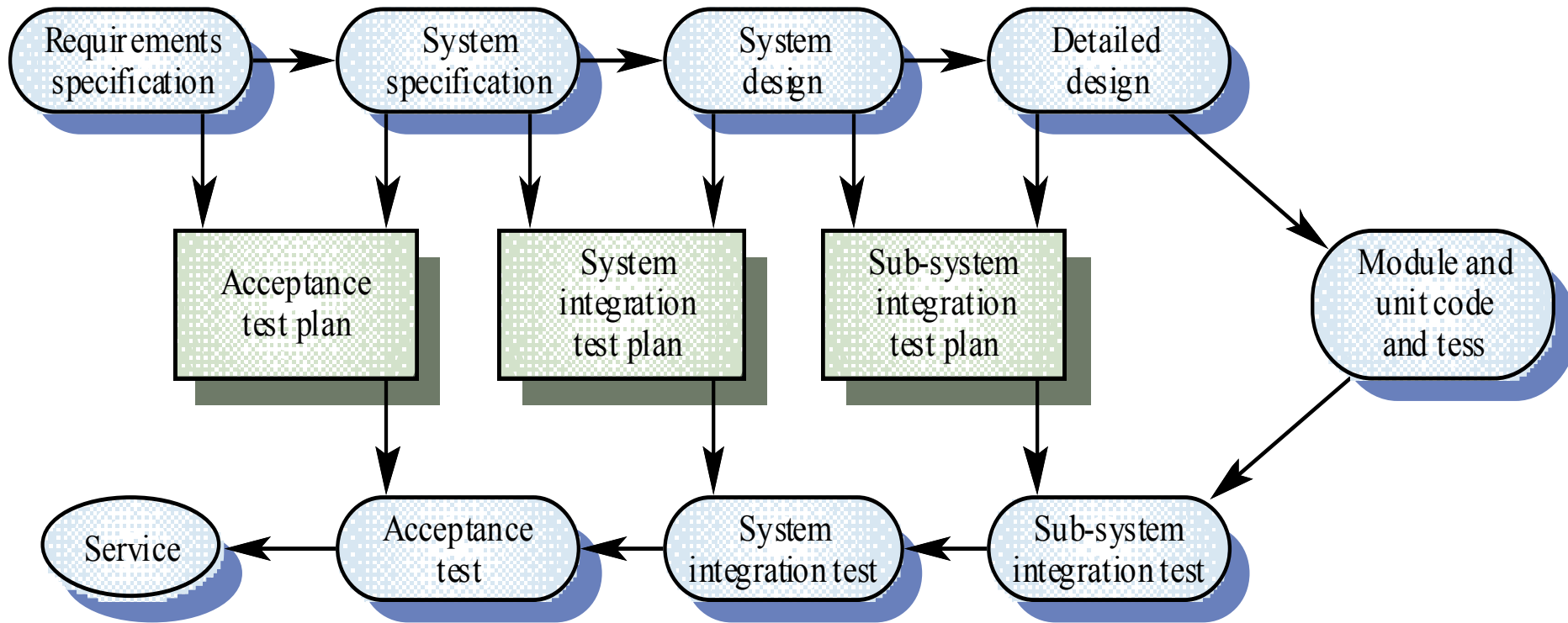




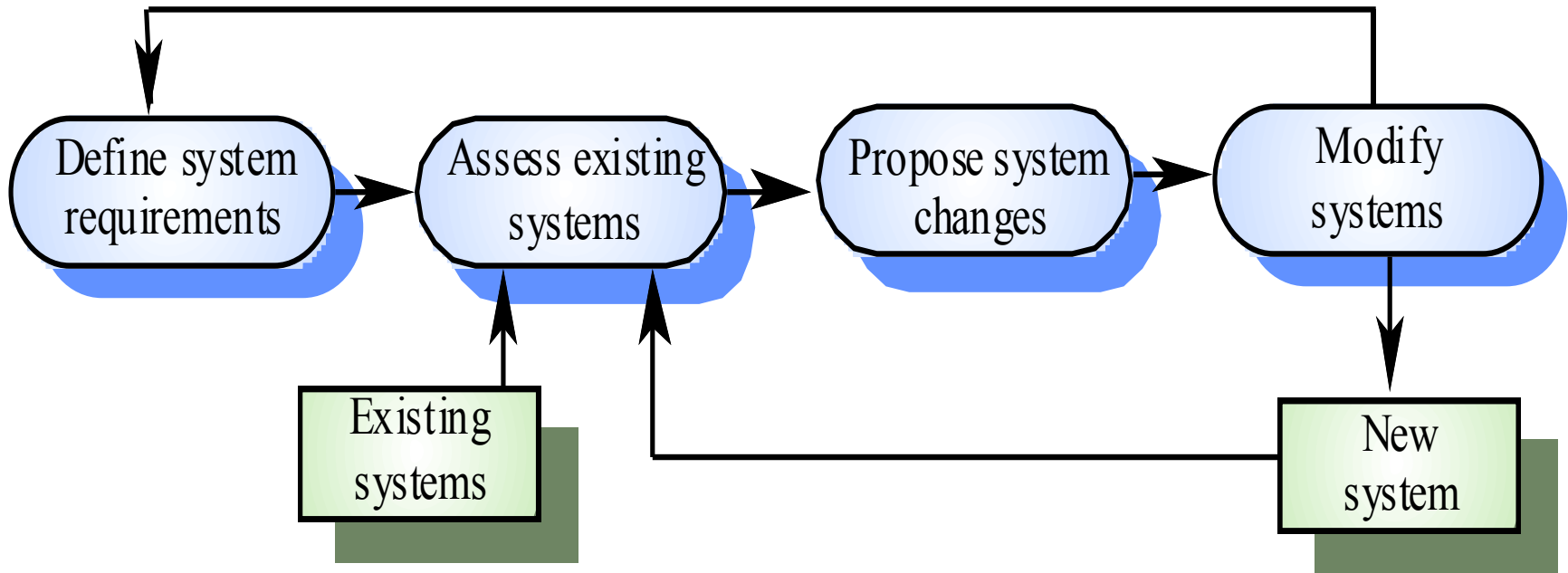
Giai đoạn kiểm thử

- Kiểm thử đơn vị: Từng thành phần sẽ được kiểm thử
- Kiểm thử module: Những thành phần có liên quan với nhau sẽ được tích hợp và kiểm thử
- Kiểm thử phân hệ: Các module được tích hợp và kiểm thử, thông thường là kiểm thử giao diện
- Kiểm thử hệ thống: Kiểm thử toàn bộ hệ thống dựa trên các chức năng
- Kiểm thử chấp nhận: Kiểm thử bởi người sử dụng

Giai đoạn kiểm thử



System evolution





6.4. Ngành công nghiệp phần mềm ở Việt nam

- Còn non trẻ
- Được Chính phủ hỗ trợ
- Có nhiều triển vọng

Chương 7+ một số lĩnh vực
khác của khoa học máy tính

Tổng quan về CSDEI HOT CSDEI

Giới thiệu về trí tuệ nhân tạo và ứng dụng

Mục tiêu

- Giới thiệu chung về CSDL, các mô hình dữ liệu, HQT CSDL, các tính năng của HQT CSDL
- Các vấn đề cơ bản về Trí tuệ nhân tạo, ứng dụng trí tuệ nhân tạo vào cuộc sống

Bố cục

- 7.1. Cơ sở dữ liệu, hệ quản trị cơ sở dữ liệu
- 7.2. Giới thiệu về trí tuệ nhân tạo và ứng dụng

Tài liệu tham khảo

- Chương 9,10 - Computer Science
- -Chương 7- bài giảng Giới thiệu Khoa học Máy tính.
- - tham khảo Giáo trình cơ sở dữ liệu (ebook)

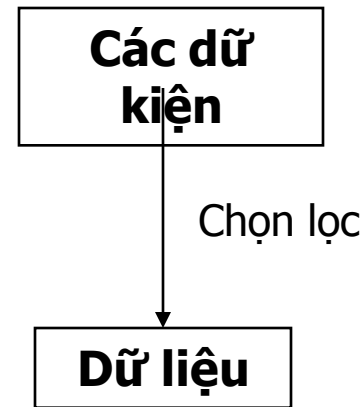
7.1. Cơ bản về CSDL, HQT CSDL

- Giới thiệu
- Quá trình phát triển
- Một số đặc tính của CSDL
- Người sử dụng CSDL
- Kiến trúc của HQT CSDL
- Các tính năng của HQT CSDL
- Các khái niệm
- Ngôn ngữ CSDL

Giới thiệu

■ Ví dụ

- Kinh doanh
- Ngân hàng và tài chính
- Giáo dục
- Hành chính
- Giải trí
- ...



■ Dữ liệu (Data)

- Một mô tả hình thức về thông tin và hoạt động
 - Tên, địa chỉ, số điện thoại của khách hàng
 - Báo cáo doanh thu
 - Đăng ký học phần

Giới thiệu (tt)

■ Cơ sở dữ liệu (Database)

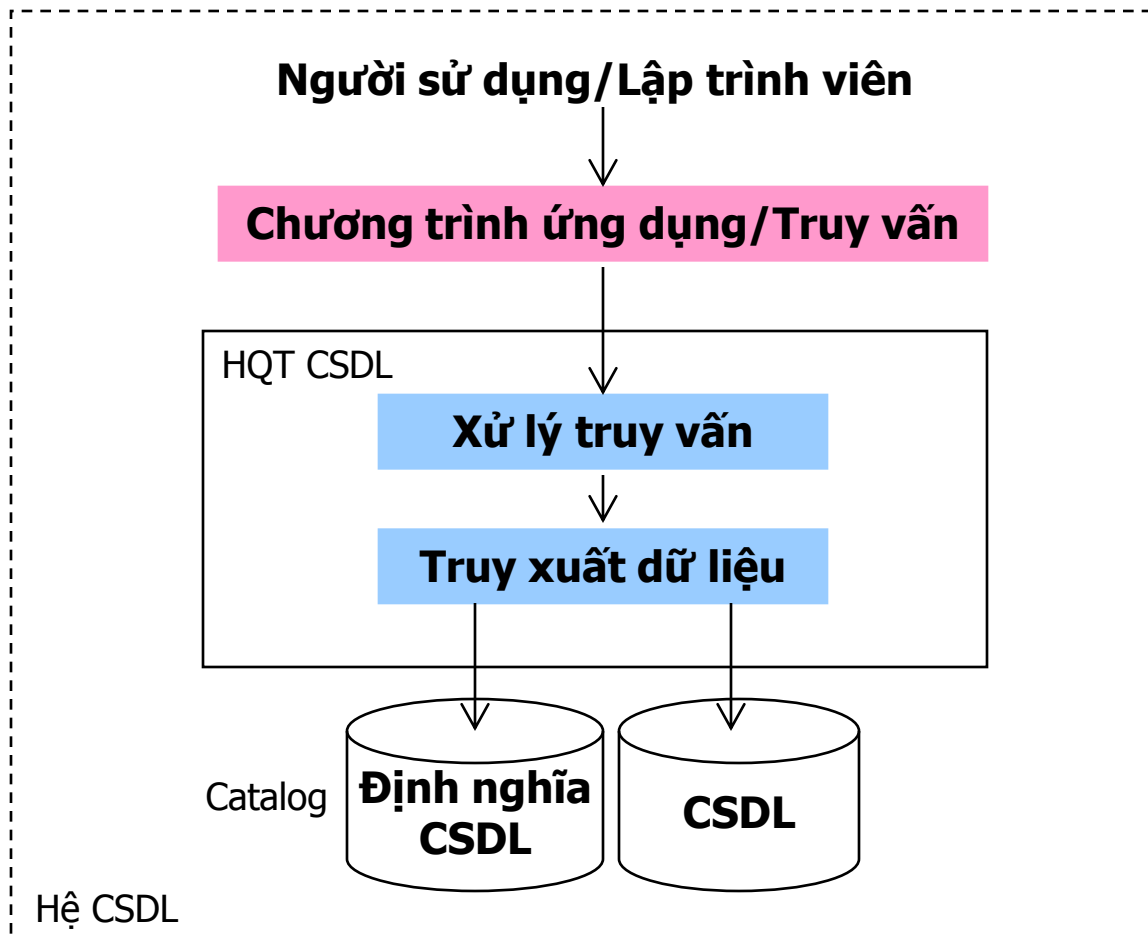
- Một tập hợp có cấu trúc của những dữ liệu có liên quan với nhau được lưu trữ trong máy tính
 - Danh sách sinh viên
 - Niên giám điện thoại
 - Danh mục các đề án
- Một CSDL biểu diễn một phần của thế giới thực (thế giới thu nhỏ)
- CSDL được thiết kế, xây dựng, và lưu trữ với một mục đích xác định, phục vụ cho một số ứng dụng và người dùng
- Tập ngẫu nhiên của các dữ liệu không thể xem là một CSDL

Giới thiệu (tt)

- Hệ quản trị CSDL (Database Management System)
 - Tập hợp các chương trình cho phép người sử dụng tạo ra và duy trì CSDL
 - Một phần mềm hệ thống cho phép định nghĩa, xây dựng và xử lý dữ liệu
 - Định nghĩa – khai báo bộ khung dữ liệu cùng với các mô tả chi tiết về dữ liệu
 - Xây dựng – lưu trữ dữ liệu lên bộ nhớ phụ
 - Xử lý – truy vấn, cập nhật và phát sinh báo cáo

Giới thiệu (tt)

- Hệ CSDL (Database System)



Một ví dụ về CSDL

| NHANVIEN | HONV | TENLOT | TENNV | MANV | NGSINH | MA_NQL | PHG |
|-----------------|--------|--------|-------|-----------|------------|-----------|-----|
| | Tran | Hong | Quang | 987987987 | 03/09/1969 | 987654321 | 4 |
| | Nguyen | Thanh | Tung | 333445555 | 12/08/1955 | 888665555 | 5 |
| | Nguyen | Manh | Hung | 666884444 | 09/15/1962 | 333445555 | 5 |
| | Tran | Thanh | Tam | 453453453 | 07/31/1972 | 333445555 | 5 |

| DEAN | TENDA | MADA | DDIEM_DA | PHONG |
|-------------|-------------|------|-----------|-------|
| | San pham X | 1 | VUNG TAU | 5 |
| | San pham Y | 2 | NHA TRANG | 5 |
| | San pham Z | 3 | TP HCM | 5 |
| | Tin hoc hoa | 10 | HA NOI | 4 |

| PHANCONG | MA_NVIAN | SODA | THOIGIAN |
|-----------------|-----------|------|----------|
| | 123456789 | 1 | 32.5 |
| | 123456789 | 2 | 7.5 |
| | 666884444 | 3 | 40.0 |
| | 453453453 | 1 | 20.0 |

Một ví dụ về CSDL (tt)

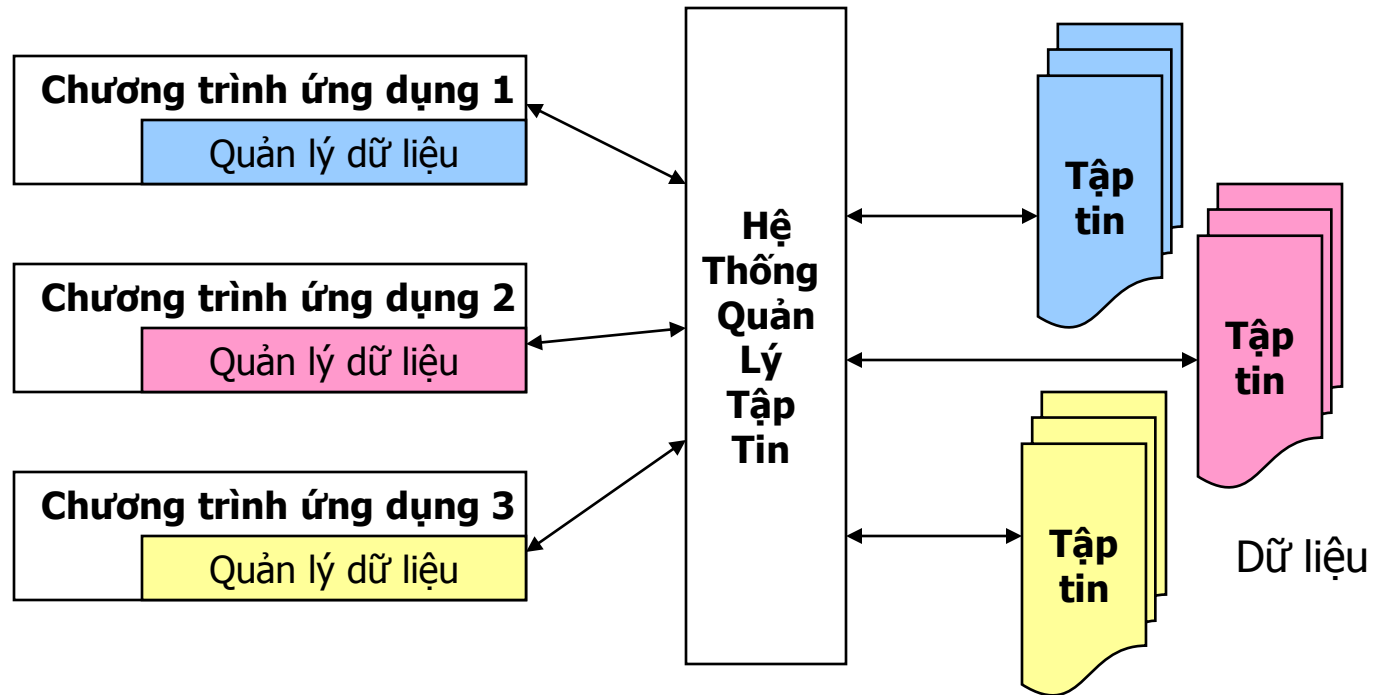
- Quản lý đề án của một công ty
 - Định nghĩa CSDL
 - Cấu trúc bảng, bao gồm các thành phần dữ liệu và kiểu dữ liệu tương ứng
 - Xây dựng CSDL
 - Đưa dữ liệu vào các bảng
 - Xử lý CSDL
 - Thực hiện các truy vấn: “Cho biết những nhân viên thuộc phòng 5”
 - Thực hiện các phép cập nhật: “Chuyển nhân viên Nguyễn Thanh Tùng sang phòng số 1”

Nội dung chi tiết

- Giới thiệu
- **Quá trình phát triển**
- Một số đặc tính của CSDL
- Người sử dụng CSDL
- Kiến trúc của HQT CSDL
- Các tính năng của HQT CSDL
- Các khái niệm
- Ngôn ngữ CSDL

Quá trình phát triển

■ Tập tin (File)



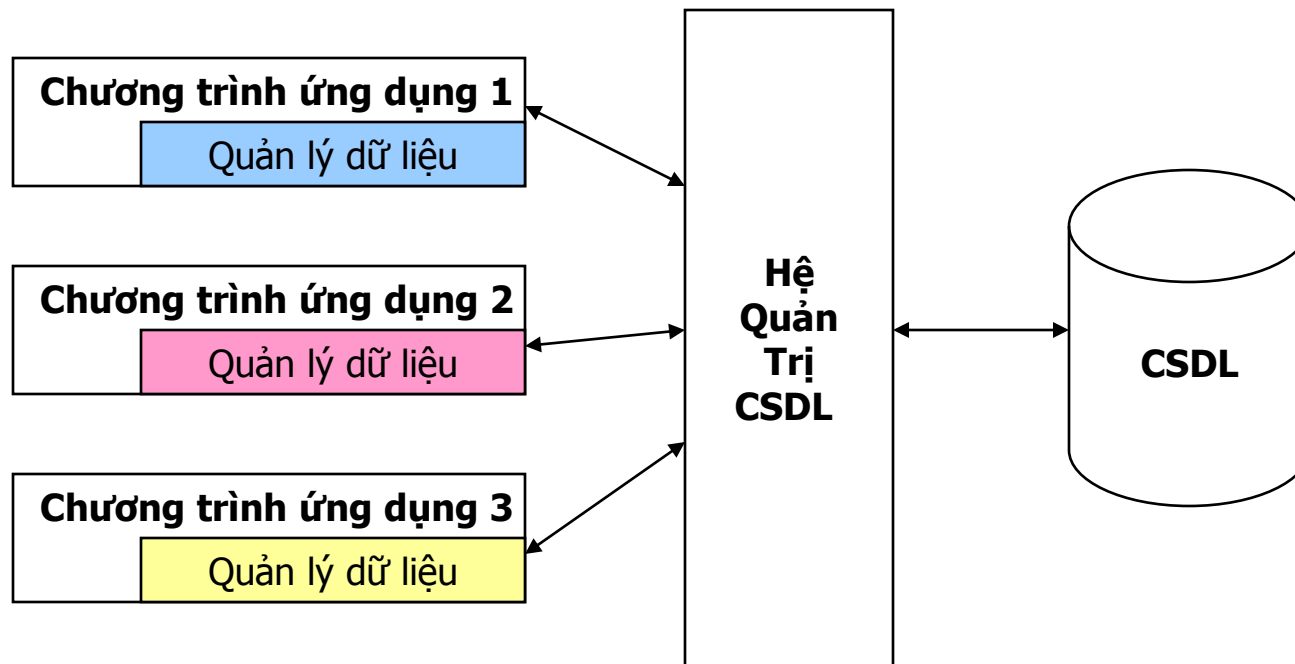
Quá trình phát triển (tt)

■ Hạn chế

- Dữ liệu bị trùng lặp và dư thừa
- Thiếu tính nhất quán giữa các dữ liệu
- Khó khăn trong việc truy xuất
- Việc chia sẻ dữ liệu bị hạn chế
- Khó khôi phục

Quá trình phát triển (tt)

- Cơ sở dữ liệu (Database)



Nội dung chi tiết

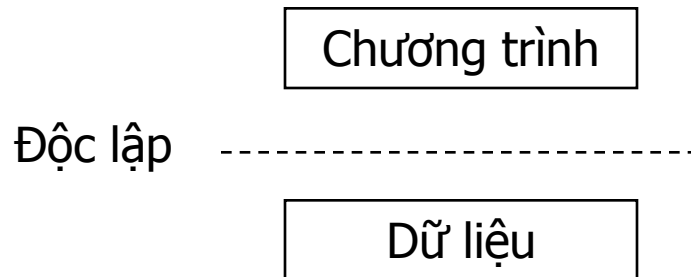
- Giới thiệu
- Quá trình phát triển
- **Một số đặc tính của CSDL**
 - Tính tự mô tả
 - Tính độc lập giữa chương trình và dữ liệu
 - Tính trừu tượng dữ liệu
 - Tính nhất quán
 - Các cách nhìn dữ liệu
- Người sử dụng CSDL
- Kiến trúc của HQT CSDL
- Các tính năng của HQT CSDL
- Các khái niệm
- Ngôn ngữ CSDL

Tính tự mô tả

- Hệ CSDL không chỉ chứa bản thân CSDL mà còn chứa định nghĩa đầy đủ (mô tả) của CSDL
- Các định nghĩa được lưu trữ trong catalog
 - Chứa các thông tin về cấu trúc tập tin, kiểu và dạng thức lưu trữ của mỗi thành phần dữ liệu và những ràng buộc dữ liệu
- Dữ liệu trong catalog gọi là meta-data (data of data)
- Các CTƯD có thể truy xuất đến nhiều CSDL nhờ thông tin cấu trúc được lưu trữ trong catalog

Tính độc lập

- Vì định nghĩa về cấu trúc CSDL được lưu trữ trong catalog nên khi có thay đổi nhỏ về cấu trúc ta ít phải sửa lại chương trình



Tính trừu tượng

- Hệ CSDL cho phép trình bày dữ liệu ở một mức trừu tượng cho phép, nhằm che bớt những chi tiết lưu trữ thật của dữ liệu
- Trừu tượng hóa dữ liệu
 - Mô hình dữ liệu
 - Đối tượng
 - Thuộc tính của đối tượng
 - Mối liên hệ

Tính nhất quán

- Lưu trữ dữ liệu thống nhất
 - Tránh được tình trạng trùng lặp thông tin
- Có cơ chế điều khiển truy xuất dữ liệu hợp lý
 - Tránh được việc tranh chấp dữ liệu
 - Bảo đảm dữ liệu luôn đúng tại mọi thời điểm

Các cách nhìn dữ liệu

- Hệ CSDL cho phép nhiều người dùng thao tác lên cùng một CSDL
- Mỗi người đòi hỏi một cách nhìn (view) khác nhau về CSDL
- Một view là
 - Một phần của CSDL hoặc
 - Dữ liệu tổng hợp từ CSDL

Nội dung chi tiết

- Giới thiệu
- Quá trình phát triển
- Một số đặc tính của CSDL
- **Người sử dụng CSDL**
 - Quản trị viên (Database Administrator - DBA)
 - Thiết kế viên (Database Designer)
 - Người dùng cuối (End User)
- Kiến trúc của HQT CSDL
- Các tính năng của HQT CSDL
- Các khái niệm
- Ngôn ngữ CSDL

Quản trị viên

- Có trách nhiệm quản lý hệ CSDL
 - Cấp quyền truy cập CSDL
 - Điều phối và giám sát việc sử dụng CSDL

Thiết kế viên

- Chịu trách nhiệm về
 - Lựa chọn cấu trúc phù hợp để lưu trữ dữ liệu
 - Quyết định những dữ liệu nào cần được lưu trữ
- Liên hệ với người dùng để nắm bắt được những yêu cầu và đưa ra một thiết kế CSDL thỏa yêu cầu này
- Có thể là 1 nhóm các DBA quản lý các CSDL sau khi việc thiết kế hoàn tất

Người dùng cuối

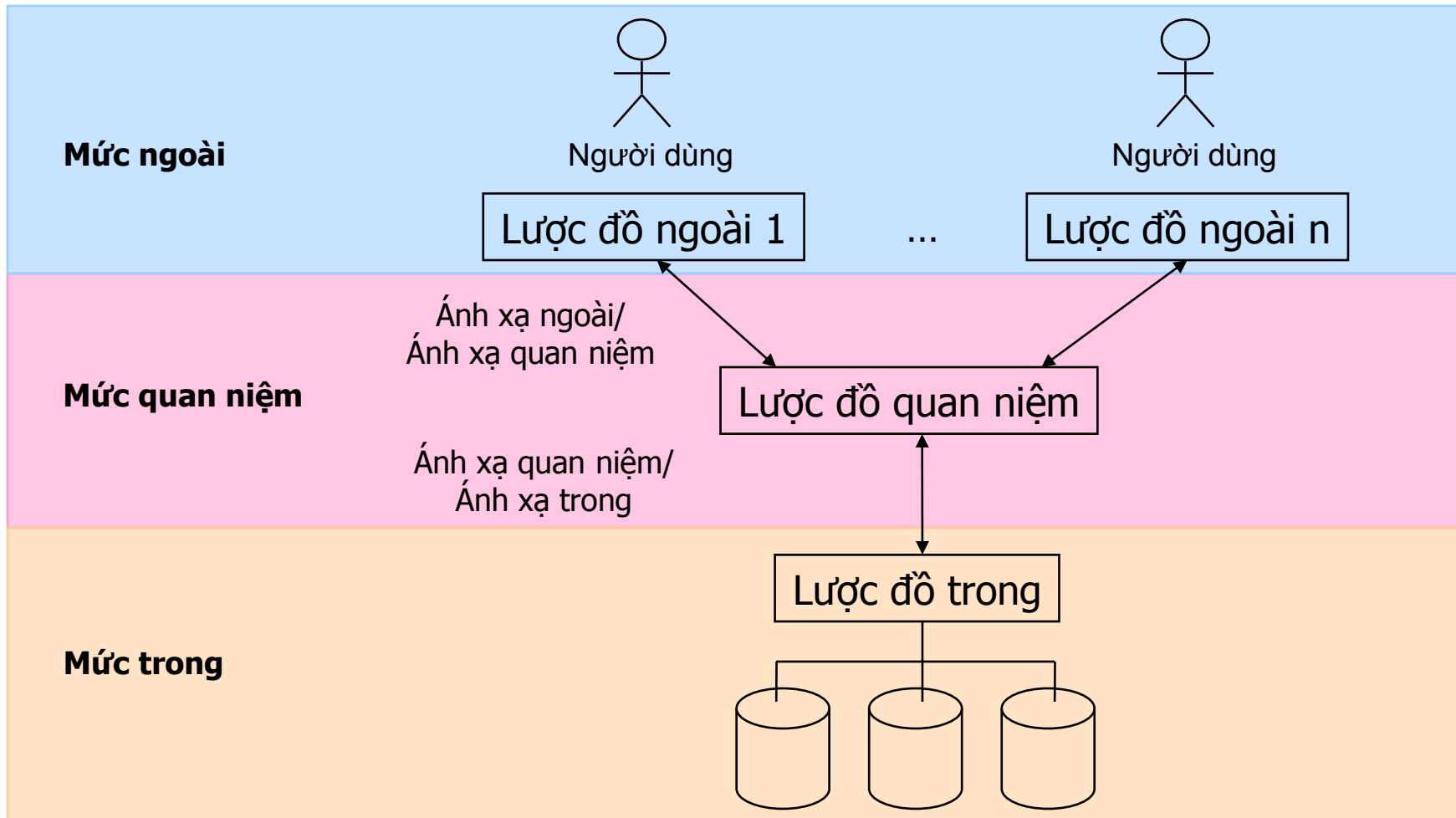
- Người ít sử dụng
 - Ít khi truy cập CSDL, nhưng cần những thông tin khác nhau trong mỗi lần truy cập và dùng những câu truy vấn phức tạp
 - Người quản lý
- Người sử dụng thường xuyên
 - Thường xuyên truy vấn và cập nhật CSDL nhờ vào một số các chức năng đã được xây dựng sẵn
 - Nhân viên
- Người sử dụng đặc biệt
 - Thông thạo về HQT CSDL, tự xây dựng những truy vấn phức tạp cho công việc
 - Kỹ sư, nhà khoa học, người phân tích,...

Nội dung chi tiết

- Giới thiệu
- Quá trình phát triển
- Một số đặc tính của CSDL
- Người sử dụng CSDL
- **Kiến trúc của HQT CSDL**
- Các tính năng của HQT CSDL
- Các khái niệm
- Ngôn ngữ CSDL

Kiến trúc của HQT CSDL

■ Kiến trúc 3 lược đồ



Kiến trúc của HQT CSDL (tt)

- Mức trong (lược đồ trong)
 - Mô tả cấu trúc lưu trữ vật lý CSDL
- Mức quan niệm (lược đồ quan niệm)
 - Mô tả cấu trúc của toàn thể CSDL cho 1 cộng đồng người sử dụng, gồm thực thể, kiểu dữ liệu, mối liên hệ và ràng buộc
 - Che bớt các chi tiết của cấu trúc lưu trữ vật lý
- Mức ngoài (lược đồ ngoài)
 - Còn gọi là mức khung nhìn (view)
 - Mô tả một phần của CSDL mà 1 nhóm người dùng quan tâm đến và che dấu phần còn lại của CSDL đối với nhóm người dùng đó

Kiến trúc của HQT CSDL (tt)

- Độc lập dữ liệu

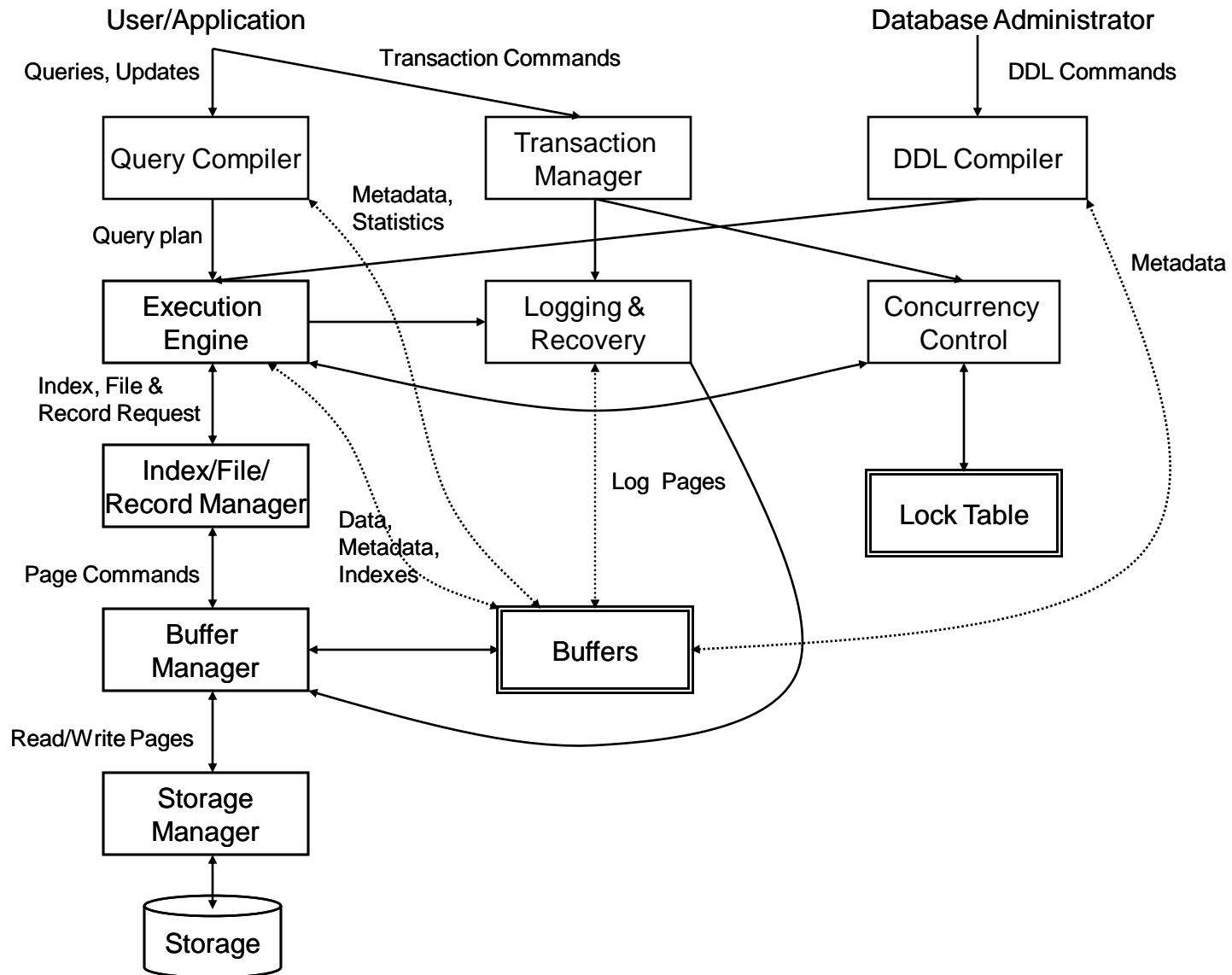
- Độc lập logic

Khả năng thay đổi lược đồ quan niệm mà không thay đổi lược đồ ngoài hoặc các CTƯD

- Độc lập vật lý

Khả năng thay đổi lược đồ trong mà không làm thay đổi lược đồ quan niệm cũng như lược đồ ngoài

Kiến trúc của HQT CSDL (tt)



Nội dung chi tiết

- Giới thiệu
- Quá trình phát triển
- Một số đặc tính của CSDL
- Người sử dụng CSDL
- Kiến trúc của HQT CSDL
- **Các tính năng của HQT CSDL**
- Các khái niệm
- Ngôn ngữ CSDL

Các tính năng của HQT CSDL

- Kiểm soát được tính dư thừa của dữ liệu
 - Tích hợp các nhu cầu dữ liệu của người dùng để xây dựng một CSDL thống nhất
- Chia sẻ dữ liệu
 - Trong môi trường đa người dùng, các HQT phải cho phép truy xuất dữ liệu đồng thời
- Hạn chế những truy cập không cho phép
 - Từng người dùng và nhóm người dùng có một tài khoản và mật mã để truy xuất dữ liệu
- Cung cấp nhiều giao diện
 - HQT cung cấp ngôn ngữ giữa CSDL và người dùng

Các tính năng của HQT CSDL (tt)

- Đảm bảo các ràng buộc toàn vẹn
 - RBTV (Integrity Constraints) là những qui định cần được thỏa mãn để đảm bảo dữ liệu luôn phản ánh đúng ngữ nghĩa của thế giới thực
 - Một số RB có thể được khai báo với HQT và HQT sẽ tự động kiểm tra. Một số RB khác được kiểm tra nhờ CTƯD
- Khả năng sao lưu dự phòng khi gặp sự cố
 - Có khả năng khôi phục dữ liệu khi có sự hư hỏng về phần cứng hoặc phần mềm

Các tính năng của HQT CSDL (tt)

■ Các tính năng khác

- Chuẩn hóa

Cho phép DBA định nghĩa và bắt buộc áp dụng một chuẩn thống nhất cho mọi người dùng

- Uyển chuyển

Khi nhu cầu công việc thay đổi, cấu trúc CSDL rất có thể thay đổi, HQT cho phép thêm hoặc mở rộng cấu trúc mà không làm ảnh hưởng đến CTƯD

- Giảm thời gian phát triển ứng dụng

- Tính khả dụng

Khi có một sự thay đổi lên CSDL, tất cả người dùng đều thấy được

Nội dung chi tiết

- Giới thiệu
- Quá trình phát triển
- Một số đặc tính của CSDL
- Người sử dụng CSDL
- Các tính năng của HQT CSDL
- Kiến trúc của HQT CSDL
- **Các khái niệm**
 - Mô hình dữ liệu
 - Lược đồ
 - Thể hiện
- Ngôn ngữ CSDL

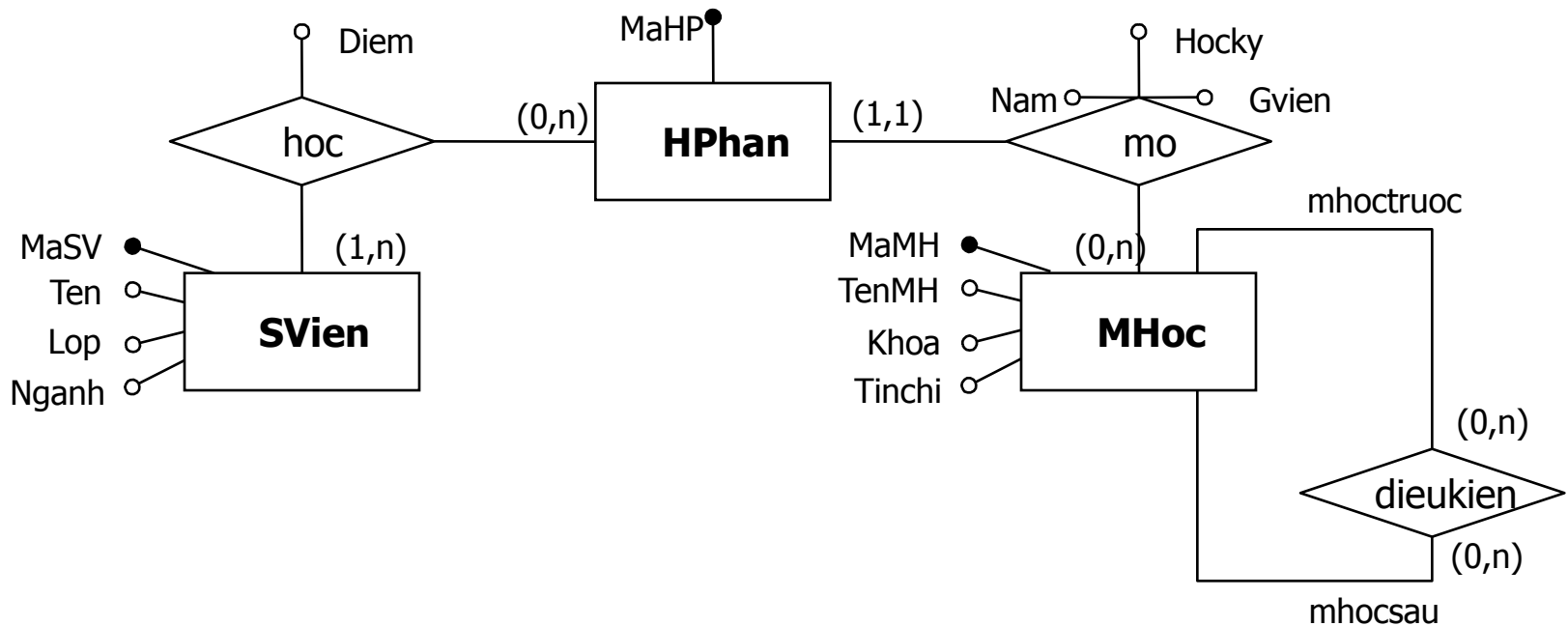
Mô hình dữ liệu

- Mô hình dữ liệu (Data Model) bao gồm
 - Các khái niệm biểu diễn dữ liệu
 - Các phép toán xử lý dữ liệu

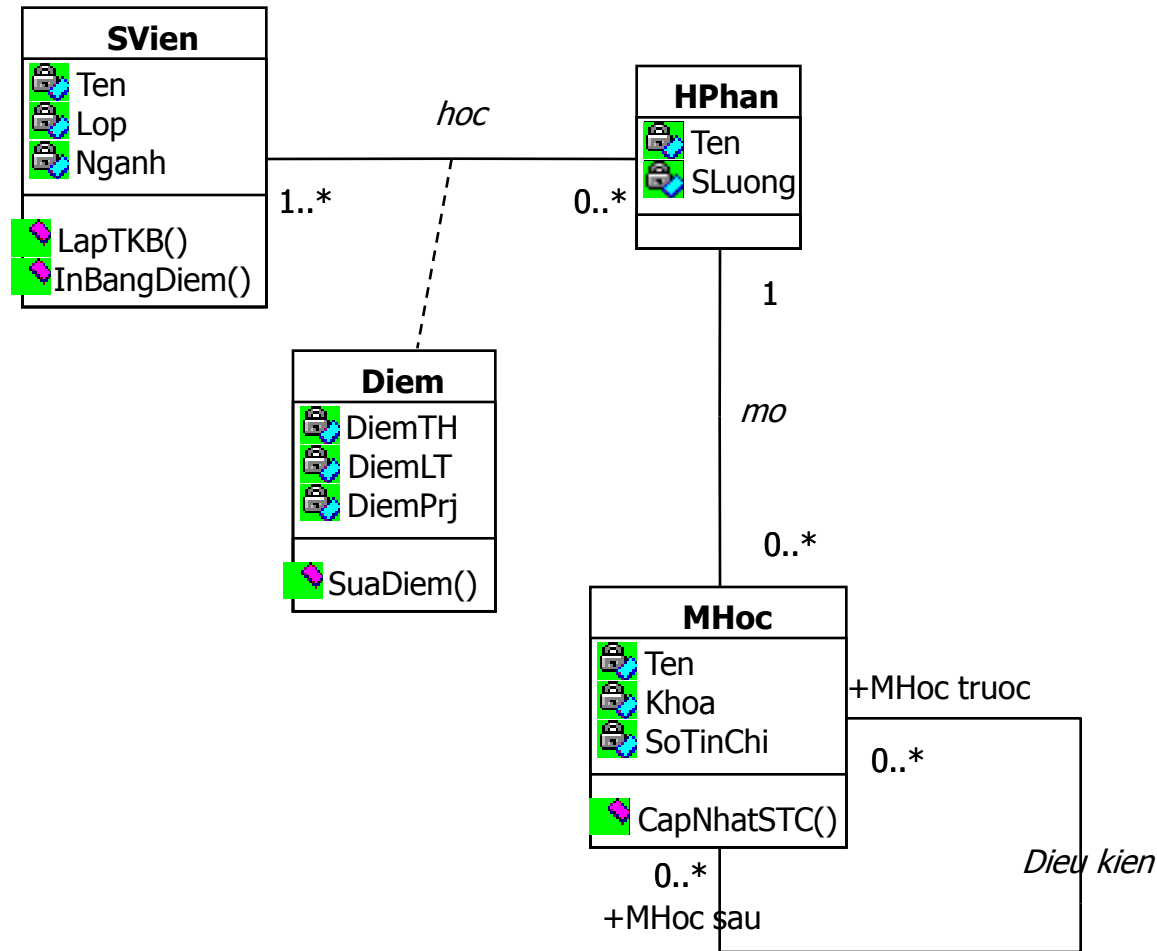
Mô hình dữ liệu (tt)

- Mô hình mức cao
 - Cung cấp các khái niệm gần gũi với người dùng
 - Mô hình phải tự nhiên và giàu ngữ nghĩa
 - VD: mô hình thực thể kết hợp (ER), mô hình đối tượng...
- Mô hình cài đặt
 - Đưa ra các khái niệm người dùng có thể hiểu được nhưng không quá xa với cách dữ liệu được tổ chức thật sự trên máy tính
 - VD: mô hình quan hệ, mô hình mạng, mô hình phân cấp
- Mô hình mức thấp (mô hình vật lý)
 - Đưa ra các khái niệm mô tả chi tiết về cách thức dữ liệu được lưu trữ trong máy tính

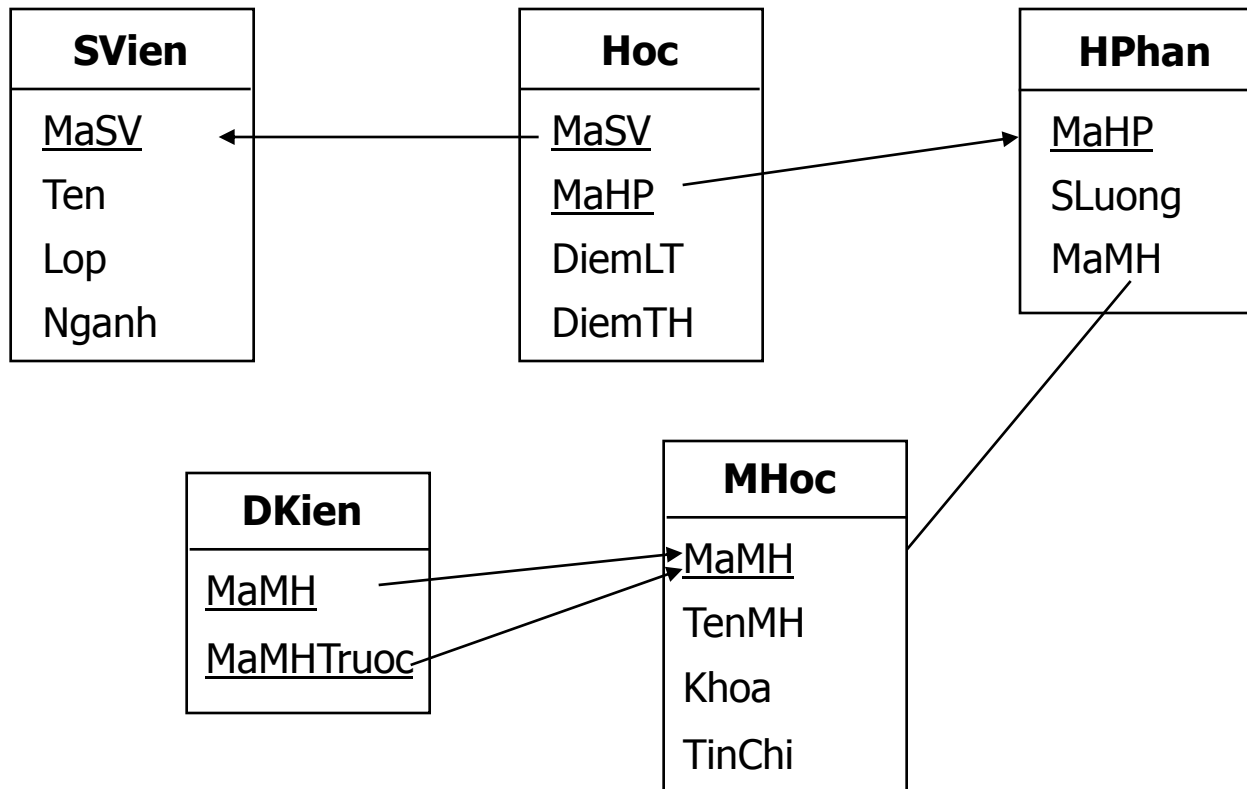
Ví dụ mô hình ER



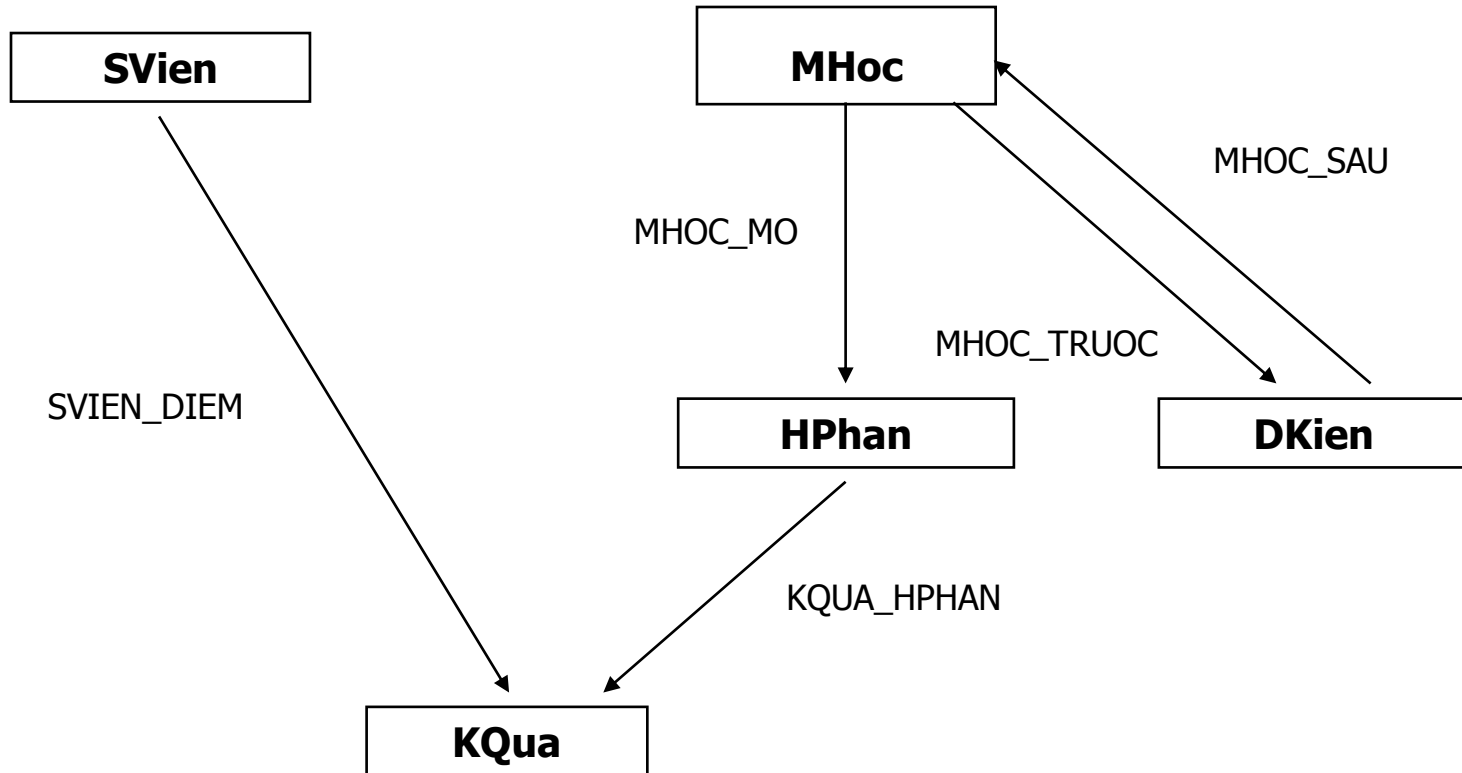
Ví dụ mô hình đối tượng



Ví dụ mô hình quan hệ

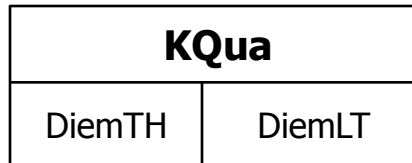


Ví dụ mô hình mạng

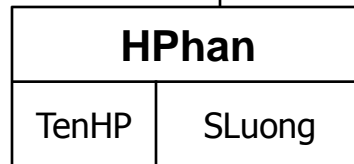


Ví dụ mô hình phân cấp

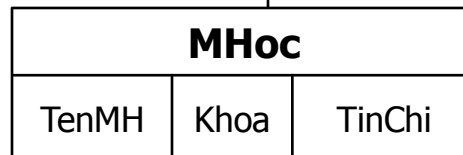
Mức 1:



Mức 2:

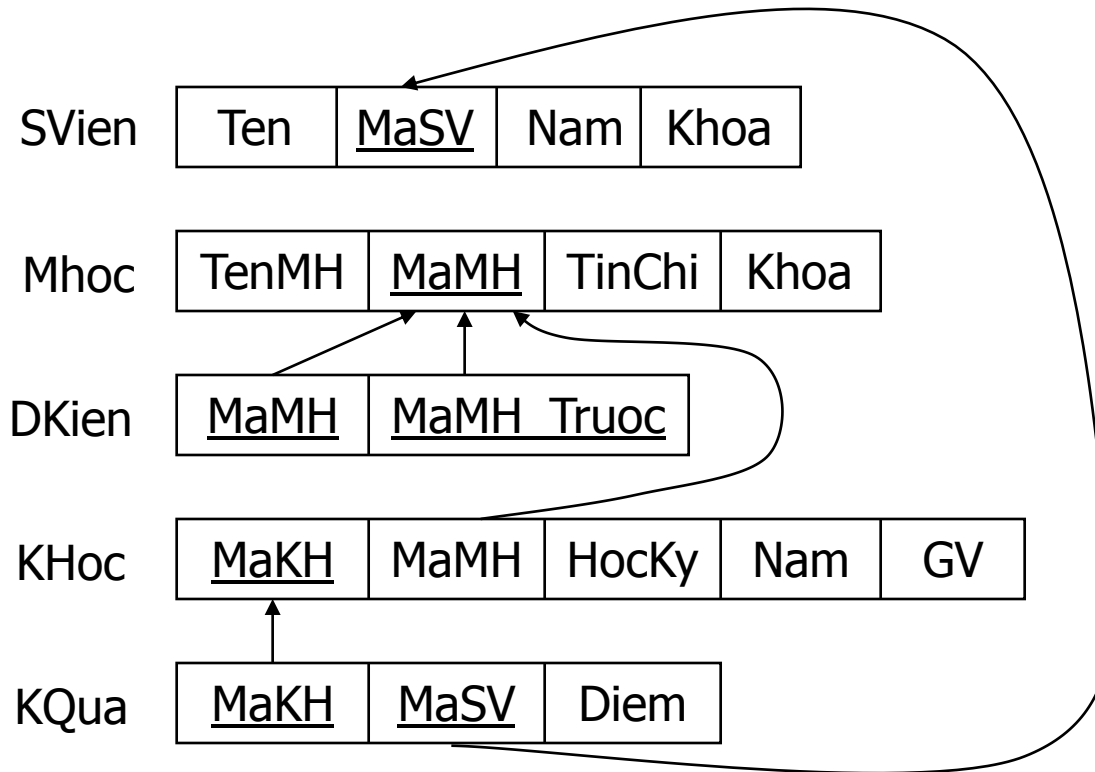


Mức 3:



Lược đồ

- Lược đồ CSDL (Database Schema)
 - Là các mô tả về cấu trúc và ràng buộc trên CSDL



Ngôn ngữ CSDL

- Ngôn ngữ định nghĩa dữ liệu
(DDL – Data Definition Language)
Xác định ra lược đồ quan niệm
- Ngôn ngữ lưu trữ dữ liệu
(SDL – Storage Definition Language)
Ngôn ngữ định nghĩa lược đồ trong
- Ngôn ngữ định nghĩa khung nhìn
(VDL – View Definition Language)
Ngôn ngữ định nghĩa lược đồ ngoài

Ngôn ngữ CSDL (tt)

- Ngôn ngữ thao tác dữ liệu
(DML – Data Manipulation Language)
Cho phép truy xuất, thêm, xóa, sửa dữ liệu
Mức cao (phi thủ tục)
Mức thấp (thủ tục)



7.2. GIỚI THIỆU VỀ TRÍ TuỆ NHÂN TẠO – (AI)

AI là gì ?

- Trí tuệ nhân tạo hay trí thông minh nhân tạo (tiếng Anh: *artificial intelligence* hay *machine intelligence*, thường được viết tắt là **AI**) là trí tuệ được biểu diễn bởi bất cứ một hệ thống nhân tạo nào.
-

AI làm việc như thế nào ?

- Chương trình AI cơ bản nhất là một lệnh Nếu-Thì đơn, chẳng hạn "Nếu A, thì B". Nếu bạn gõ một chữ 'A', máy tính sẽ in cho bạn một chữ 'B'. Về cơ bản, bạn đang dạy máy tính làm một nhiệm vụ. Bạn cho dữ liệu vào là một thứ, máy tính sẽ đáp lại bằng một cái gì đó bạn bảo nó làm hoặc nói. Mọi chương trình đều có các điều kiện Nếu-Thì.
 - Một ví dụ phức tạp hơn là: nếu bạn gõ từ "Xin chào", máy tính đáp lại "Bạn có khỏe không?" Phản ứng này không phải ý nghĩ của máy tính mà chẳng qua chỉ là một dòng lệnh bạn đã viết từ trước trong chương trình. Mỗi khi bạn gõ "Xin chào", máy tính sẽ luôn luôn trả lời "Bạn có khỏe không?". Nhìn qua thì có vẻ như máy tính đang sống và suy nghĩ, nhưng thực ra đó chỉ là một phản hồi tự động. Trí tuệ nhân tạo thường là một chuỗi dài các lệnh Nếu-Thì.
-

-
- Có thể dùng thêm một chút ngẫu nhiên để tạo hai hoặc nhiều hướng trả lời. Ví dụ, nếu bạn gõ từ "Xin chào", máy tính có thể đáp "Bạn có khỏe không?" hoặc "Hôm nay trời đẹp nhỉ?" hay "Công việc của bạn dạo này tốt chứ?". Bây giờ có thể có ba phản ứng (hay ba kiểu 'thì') thay vì chỉ có một. Một chương trình trí tuệ nhân tạo có thể có 1000 cách trả lời cùng một dữ liệu vào. Điều đó làm cho nó đỡ đoán trước được và gần với kiểu trả lời của một người bình thường hơn, vì một người bình thường sẽ trả lời một cách không đoán trước được.
-

-
- Nhiều trò chơi, chẳng hạn cờ và các trò chơi chiến lược, sử dụng các phản ứng bằng hành động, do đó, người có thể chơi với máy tính. Các robot với các bộ não trí tuệ nhân tạo cũng sẽ sử dụng các lệnh Nếu-Thì và các lựa chọn ngẫu nhiên để đưa ra các quyết định và nói. Tuy nhiên, dữ liệu vào có thể là các đối tượng trước mặt robot thay vì một dòng chữ "Xin chào", và phản ứng có thể là nhặt đối tượng lên thay vì in một dòng chữ ra màn hình.
-

-
- Những ý tưởng đã có từ trước
 - Trí tuệ nhân tạo đã được bắt đầu nghiên cứu từ thập niên 1950 bởi : Allen Newell và Herbert Simon, người sáng lập phòng thí nghiệm trí tuệ nhân tạo đầu tiên ở Đại học Carnegie Mellon, và John McCarthy và Marvin Minsky, người sáng lập phòng thí nghiệm trí tuệ nhân tạo tại MIT, MIT AI Lab, năm 1959.
-

Phương pháp tiếp cận

- Hai trường phái chính "ngăn nắp" và "bê bối".
 - "ngăn nắp" *cổ điển* thường sử dụng các biểu tượng và khái niệm phức tạp.
 - "bê bối" (*scruffy* hay *connectionist*). Ví dụ mạng thần kinh nhân tạo, nó "tiến hóa" sự thông minh qua việc xây dựng và hoàn thiện hệ thống bằng một quá trình tự động thay cho việc thiết kế cách thi hành nhiệm vụ một cách có hệ thống

AI hiện đại

- chú trọng vào các nhiệm vụ thiên về kỹ thuật thực tế...
 - Một vài ví ngành phát triển từ TTNT là: nhân dạng mẫu, xử lý ảnh, mạng neuron, xử lý ngôn ngữ tự nhiên, robot học và lý thuyết trò chơi.
-

Một số ứng dụng

- Một vài ví dụ bao gồm:
- Chinook được xem là Nhà vô địch thế giới Máy-Người trong thu tiền vào năm 1994.
- Deep Blue, máy tính chơi cờ vua, đã đánh bại Garry Kasparov trong một trận đấu nổi tiếng vào năm 1997.
- InfoTame, một động cơ tìm kiếm và phân tích văn bản được phát triển bởi KGB để tự động sắp xếp hàng triệu giấy tờ để có thể ngăn chặn giao tiếp khi cần thiết.
- Logic mờ, một kĩ thuật đưa ra suy luận khi không chắc chắn, được dùng rộng rãi trong các hệ thống điều khiển công nghiệp.

-
- Hệ chuyên gia vẫn đang được sử dụng ở một mức độ nào đó trong ngành công nghiệp. Một ví dụ về hệ chuyên gia là Clippy, cái kẹp giấy trong Microsoft Office. Nó cố gắng dự đoán những lời khuyên mà người dùng mong muốn.
 - Machine translation: Google translator, ...
 - Mạng neuron được dùng cho nhiều nhiệm vụ khác nhau, từ hệ thống công xâm nhập đến trò chơi máy tính.
 - Các hệ thống nhận dạng kí tự quang học có thể chuyển các tài liệu viết tay bằng tiếng châu Âu bất kì sang văn bản lưu trong máy.
-

-
- Nhận dạng chữ viết tay được sử dụng trong hàng triệu thiết bị PDA.
 - Nhận dạng tiếng nói có các sản phẩm thương mại và được sử dụng rộng rãi.
 - Computer algebra system, như Mathematica và Macsyma, là những điển hình.
-

-
- Các hệ thống Machine vision đã được sử dụng trong nhiều ứng dụng công nghiệp từ kiểm định phần cứng (*hardware verification*) tới các hệ thống bảo vệ.
 - Các phương pháp lập kế hoạch TTNT đã được sử dụng để tự động lập phương án chuyển quân Mỹ trong Chiến tranh vùng Vịnh lần thứ nhất. Nhiệm vụ này có thể tiêu tốn thời gian tới nhiều tháng và tiền tới hàng triệu đô-la nếu làm bằng tay. Cơ quan nghiên cứu phòng vệ tiên tiến Mỹ đã tuyên bố rằng riêng lượng tiền tiết kiệm được nhờ ứng dụng này đã nhiều hơn tổng chi phí của họ cho các nghiên cứu TTNT trong vòng 30 năm qua.
-

AI trong thương mại

- **Hệ chuyên gia** áp dụng khả năng suy luận để đạt tới một kết luận. Một hệ chuyên gia có thể xử lý một lượng lớn thông tin đã biết và cung cấp các kết luận dựa trên các thông tin đó.
 - **Mạng nơ-ron** là TTNT có khả năng tìm kiếm và phân biệt các mẫu. Các Phòng Cảnh sát sử dụng mạng nơ-ron để nhận dạng tham nhũng.
-

-
- Các **Giải thuật di truyền** được thiết kế để áp dụng quá trình đấu tranh sinh tồn để tạo các lời giải ngày càng tốt hơn cho bài toán. Các chuyên gia đầu tư sử dụng Giải thuật di truyền để tạo tổ hợp tốt nhất của các cơ hội đầu tư cho các khách hàng của họ.
 - Một **Agent thông minh** là phần mềm hỗ trợ ta hoặc thay mặt ta thực hiện các công việc có liên quan tới máy tính. Ví dụ, chúng được sử dụng dưới dạng các chương trình khai phá dữ liệu và các agent kiểm soát và theo dõi.
-

Ngôn ngữ, Phong cách lập trình và Văn hoá phần mềm

- các ngôn ngữ Lisp hoặc Prolog. Tính toán Bayes thường sử dụng Matlab hoặc ngôn ngữ lập trình Lush (một biến thể của Lisp). Các ngôn ngữ này kèm theo nhiều thư viện xác suất chuyên dụng. Các hệ thống thực, đặc biệt các hệ thống thời gian thực, thường sử dụng C++.

Tóm tắt và câu hỏi ôn tập

- Cơ sở dữ liệu là gì? HQT CSDL ? Kể tên các HQT CSDL thông dụng?
 - Các mô hình CSDL
 - Kiến trúc HQT CSDL
 - Ứng dụng của Trí tuệ nhân tạo vào đời sống ? Kể tên một số ứng dụng cụ thể ?
-