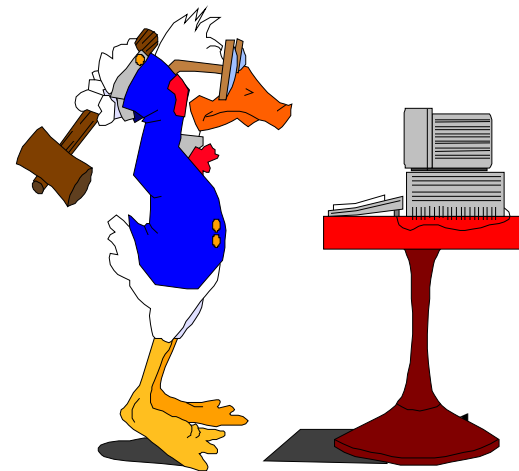


BÀI GIẢNG
CẤU TRÚC MÁY TÍNH
(Computer Structure)





Giới thiệu

Cấu trúc Máy tính

(Computer Structure)

Trình bày: Đinh Đồng Lương.

ĐT: 058.832078

Mobile: 0914147520

Email: luongdd10@yahoo.com



Mục đích và yêu cầu

Mục đích:

- Tìm hiểu *cấu trúc và tổ chức* các máy tính.
- Tìm hiểu nguyên lý hoạt động cơ bản Máy tính.
- Giới thiệu cấu trúc máy tính tiên tiến của Intel.

Yêu cầu:

- Có kiến thức lập trình cơ bản.
- Sinh viên đọc tài liệu và làm việc theo nhóm để thực hiện báo cáo trên lớp.



Tài liệu tham khảo (sách)

1. Willian Stallings - Computer Organization and Architecture.
2. Andrew Stamenbaum – Structure Computer Organization.
3. Cẩm nang sửa chữa và nâng cấp máy tính cá nhân
Nguyễn Đăng Khoa
4. Giáo trình bảo trì và nâng cấp máy tính
(Trường KHTN - TPHCM)
Lê Công Bảo
5. Cấu tạo và nguyên lý hoạt động của hệ thống máy
RON WHITE - Nguyễn Trọng Tuấn (Dịch)



Tài liệu tham khảo (trang web)

[@www.williamstallings.com](http://www.williamstallings.com)

[@ocw.mit.edu](http://ocw.mit.edu)

[@www.intel.com](http://www.intel.com)

[@www.asus.com](http://www.asus.com)

[@www.gigabyte.com](http://www.gigabyte.com)

[@www1.guidePC.com](http://www1.guidePC.com)



Nội dung môn học

- 1. Giới thiệu chung.**
- 2. Hệ thống máy tính.**
- 3. Biểu diễn dữ liệu và số học máy tính.**
- 4. Bộ xử lý trung tâm.**
- 5. Bộ nhớ Máy tính.**
- 6. Hệ thống vào ra.**



Chương 1

Giới thiệu chung

1.1 Khái niệm chung máy tính

1.2 Phân loại máy tính

1.3 Sự tiến hóa của máy tính

1.1 Khái niệm chung

Máy tính (computer) là thiết bị điện tử thực hiện công việc sau:

- ✓ nhận thông tin vào.
- ✓ xử lý thông tin theo chương trình nhớ sẵn bên trong bộ nhớ máy tính.
- ✓ đưa thông tin ra.

Chương trình (Program): chương trình là dãy các câu lệnh nằm trong bộ nhớ, nhằm mục đích hướng dẫn máy tính thực hiện một công việc cụ thể nào đấy. Máy tính thực hiện theo chương trình.

1.1 Khái niệm chung

Phần mềm (Software): Bao gồm chương trình và dữ liệu.

Phần cứng (Hardware): Bao gồm tất cả các thành phần vật lý cấu thành lên hệ thống Máy tính.

Phần dẻo (Firmware): Là thành phần chứa cả hai thành phần trên.

Kiến trúc máy tính (Computer Architecture) đề cập đến các thuộc tính của hệ thống máy tính dưới cái nhìn của người lập trình. Hay nói cách khác, là những thuộc tính ảnh hưởng trực tiếp đến quá trình thực hiện logic của chương trình. Bao gồm: tập lệnh, biểu diễn dữ liệu, các cơ chế vào ra, kỹ thuật đánh địa chỉ,...

1.1 Khái niệm chung

Tổ chức máy tính(Computer Organization): đề cập đến các khối chức năng và liên hệ giữa chúng để thực hiện những đặc trưng của kiến trúc.

Ví dụ: trong kiến trúc bộ nhân: đây là thuộc tính của hệ thống xử lý. Bộ nhân này sẽ được tổ chức riêng bên trong máy tính hoặc nó được tính toán nhiều lần trên bộ cộng để cũng được một kết quả nhân tương ứng.

Cấu trúc máy tính(Computer Structure): là những thành phần của máy tính và những liên kết giữa các thành phần.

Ở mức cao nhất máy tính bao gồm 4 thành phần:



1.1 Khái niệm chung

- ✓ Bộ xử lý : điều khiển và xử lý số liệu.
- ✓ Bộ nhớ : chứa chương trình và dữ liệu.
- ✓ Hệ thống vào ra : trao đổi thông tin giữa máy tính với bên ngoài.
- ✓ Liên kết giữa các hệ thống : liên kết các thành phần của máy tính lại với nhau.

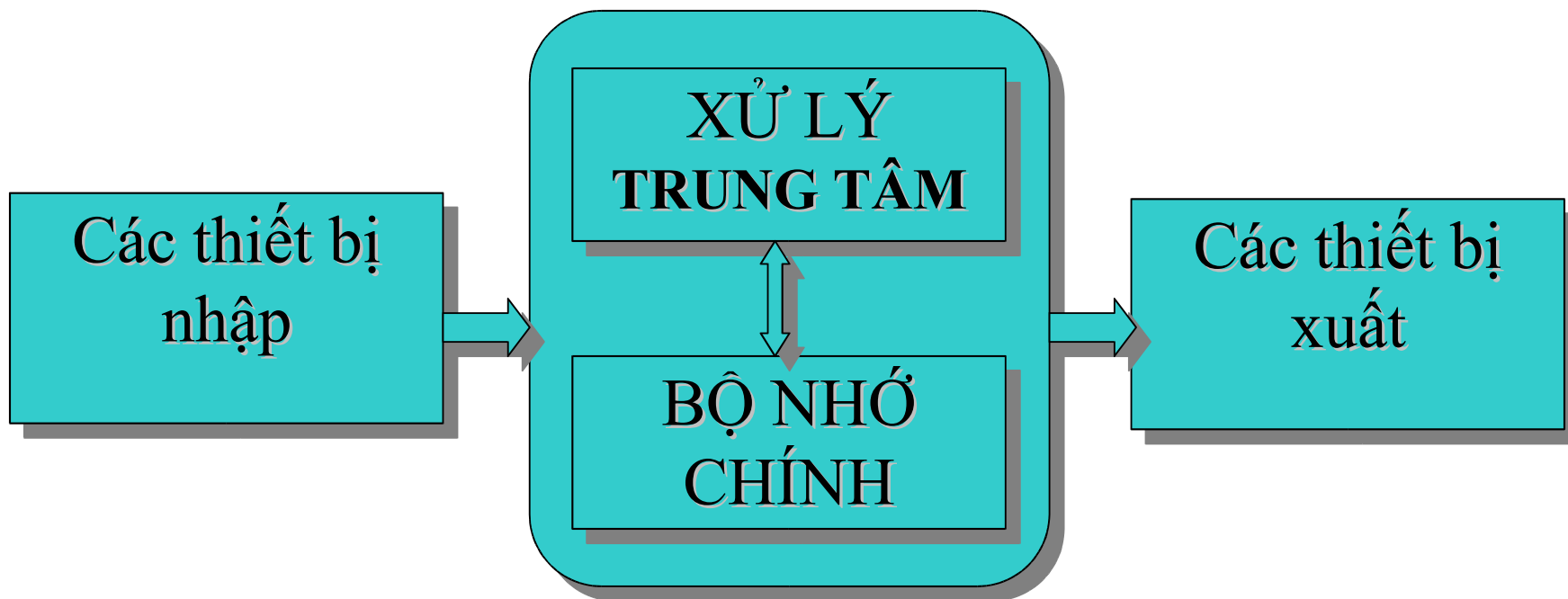
1.1 Khái niệm chung

- Mô hình phân lớp của hệ thống



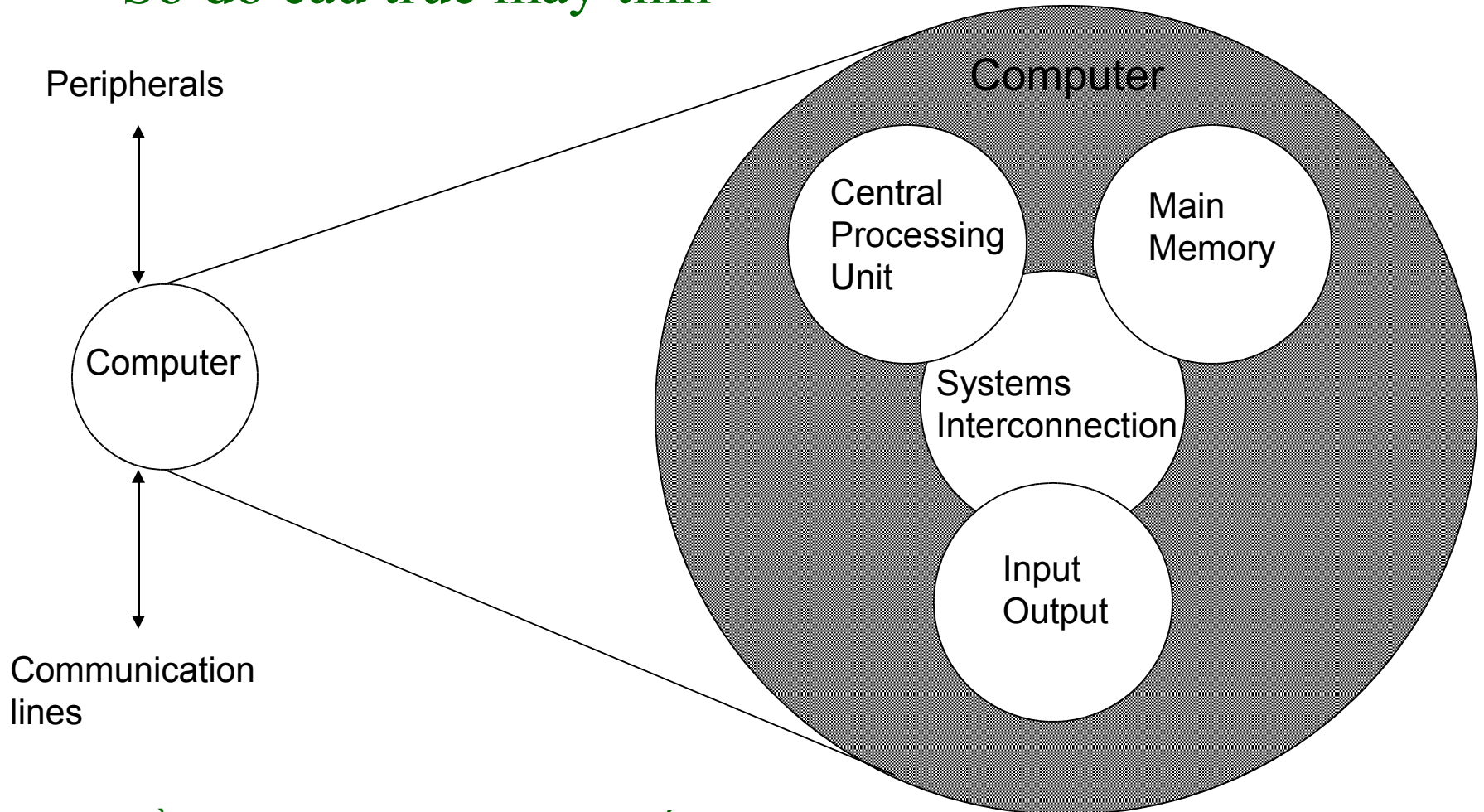
1.1 Khái niệm chung

- Mô hình cơ bản



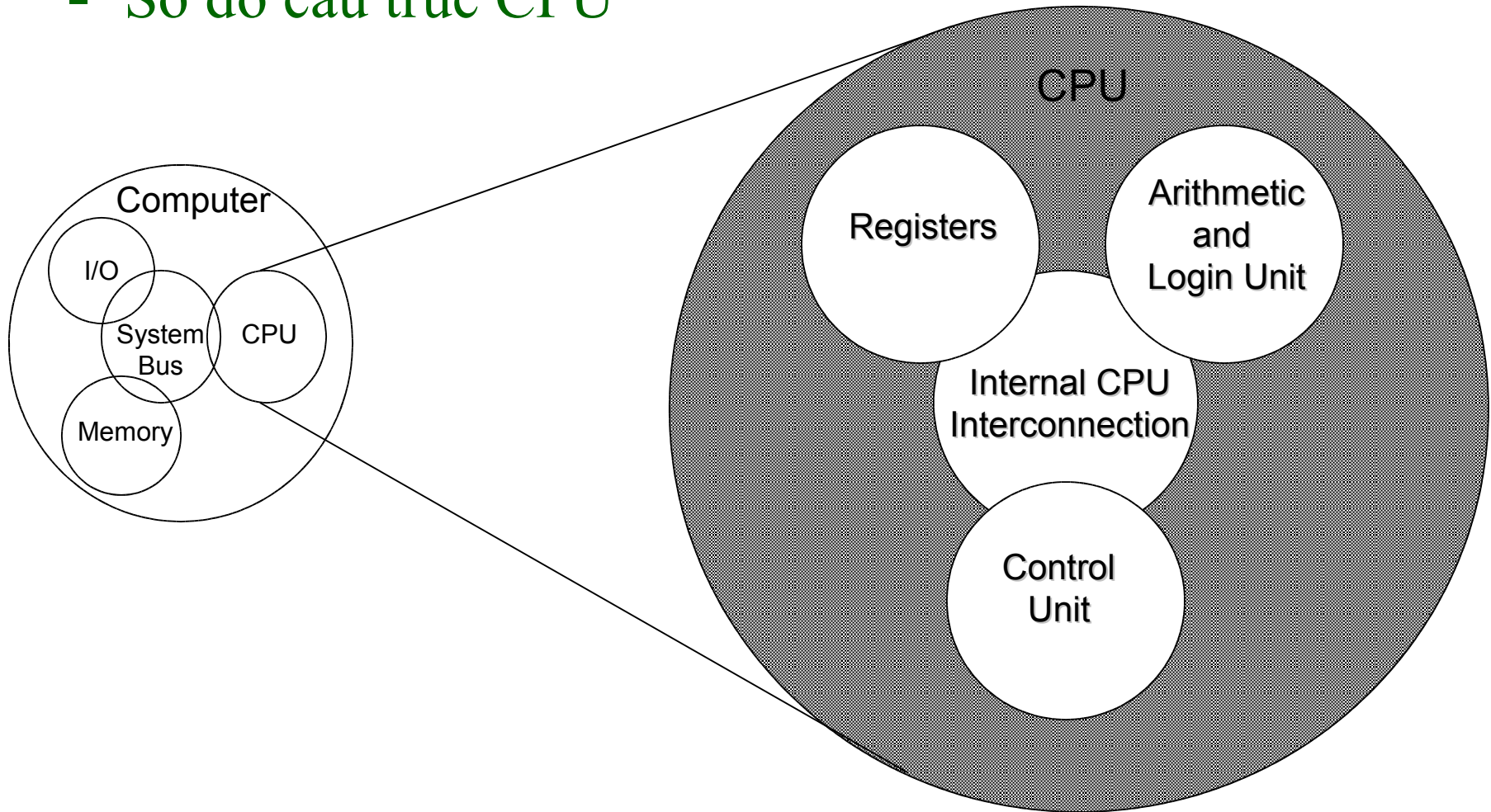
1.1 Khái niệm chung

■ Sơ đồ cấu trúc máy tính



1.1 Khái niệm chung

■ Sơ đồ cấu trúc CPU



1.1 Khái niệm chung

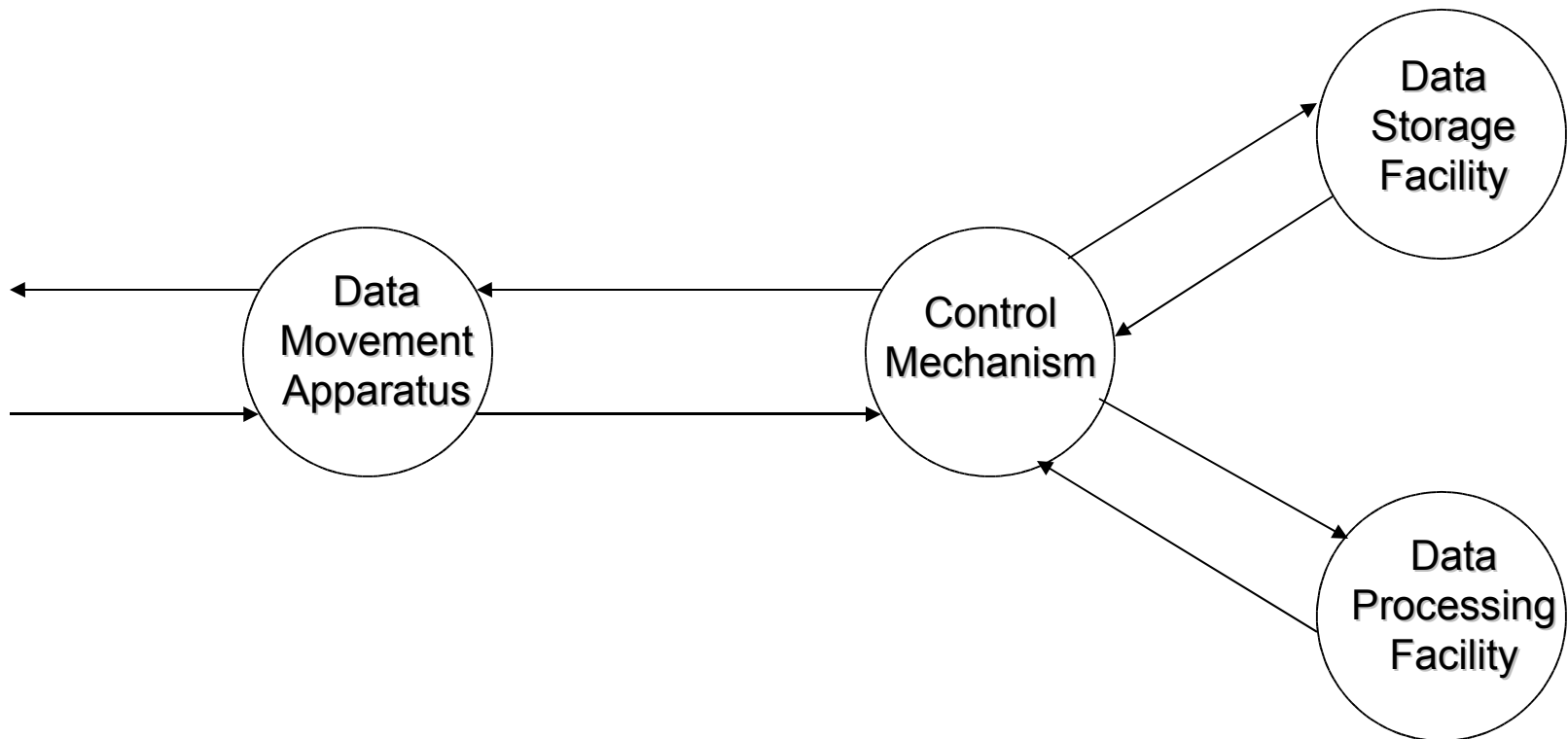
Chức năng (Computer Function): là mô tả hoạt động của hệ thống hay từng thành phần của hệ thống.

Chức năng chung của một hệ thống bao gồm:

- ✓ Xử lý dữ liệu.
- ✓ Lưu trữ dữ liệu.
- ✓ Vận chuyển dữ liệu.
- ✓ Điều khiển

1.1 Khái niệm chung

- Sơ đồ chức năng máy tính





1.2 Phân loại máy tính

Phân loại theo phương pháp truyền thống

- ✓ Máy vi tính (*Microcomputer*)
- ✓ Máy tính nhỏ (*Minicomputer*)
- ✓ Máy tính lớn (*Mainframe Computer*)
- ✓ Siêu máy tính (*Super Computer*)

Phân loại theo phương pháp hiện đại

- ✓ Máy tính để bàn (*Desktop Computer*)
- ✓ Máy chủ (*Servers*)
- ✓ Máy tính nhúng (*Embedded Computer*)

1.2 Phân loại máy tính

Máy để bàn:

- là loại máy thông dụng nhất hiện nay.
- bao gồm máy tính cá nhân (PC: Personal Computer) và trạm (Workstation Computer).
- giá mua 100\$ đến 10.000\$

Máy chủ

- là máy phục vụ(*server*)
- dùng trong mạng theo mô hình Client/Server
- có tốc độ, hiệu năng, bộ nhớ và độ tin cậy cao
- giá vài chục nghìn đến vài chục triệu đô



1.2 Phân loại máy tính

Máy tính nhúng

- được đặt trong nhiều thiết bị khác nhau để điều khiển thiết bị làm việc
- được thiết kế chuyên dụng
- ví dụ: điện thoại di động, bộ điều khiển các thiết gia đình, Router định tuyến,...

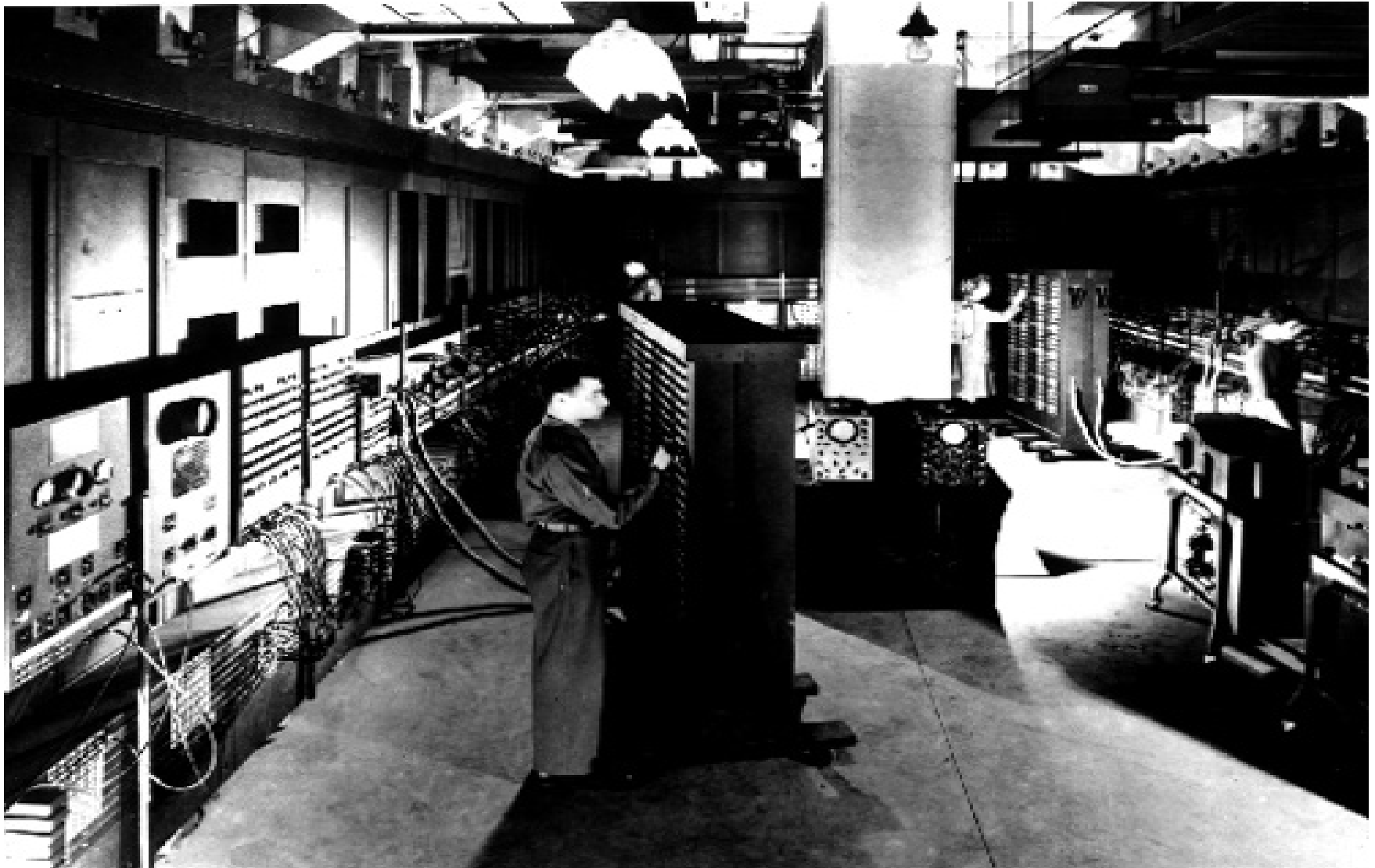
1.3 Sự tiến hóa của máy tính

Sự phát triển của máy tính chia ra 4 thế hệ:

- ✓ Thế hệ 1: Máy tính dùng đèn chân không (Vacuum Tube) 1946-1955
- ✓ Thế hệ 2: Máy tính dùng Transistor (1955-1965)
- ✓ Thế hệ 3: Máy tính dùng mạch tích hợp IC (Integrated Circuit) 1966 – 1980
- ✓ Thế hệ 4: Máy tính dùng mạch tích hợp cực lớn VLSI (Very Large Scale Integrated)1980 đến nay

Máy tính ENIAC

Electronic Numerical Integrator And Computer



1.3 Sự tiến hóa của máy tính

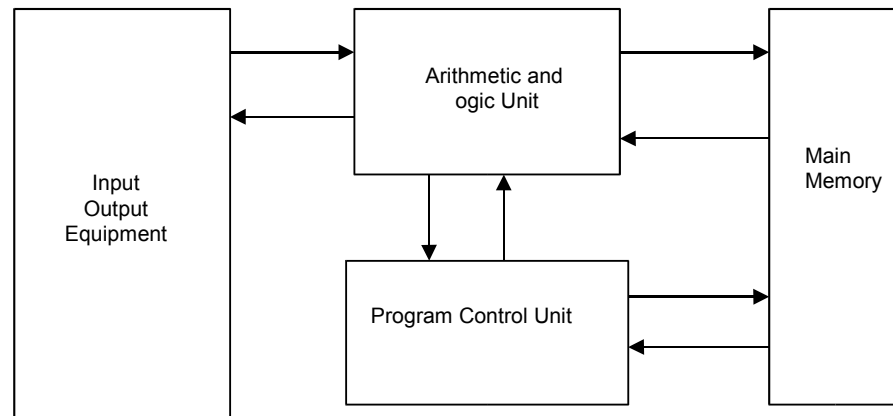
Đề xuất năm 1943 và hoàn thành 1946 được sử dụng đến 1955 do thầy trò Eckert và Mauchly Trường đại học Pennsylvania của Mỹ

Đặc điểm chính:

- ✓ Nặng 30 tấn, chiếm diện tích 150m² và sử dụng 140KW.
- ✓ 5000 nghìn phép cộng trên giây.
- ✓ Sử dụng hệ thập phân.
- ✓ Lập trình bằng công tắc.
- ✓ Sử dụng 18000 bóng đèn điện tử (vacuum tubes)

Máy tính Von Neumann

- Máy tính IAS(Institute for Advanced Studies)
- Máy có mô hình cơ bản là máy tính này nay
- Thế kể 1947 hoàn thành 1952
- Xây dựng dựa trên ý tưởng của Turring (Mỹ) và Von Neumann(Anh)



1.3 Sự tiến hóa của máy tính

- Các sản phẩm của công nghệ VLSI (Very Large Scale Integrated)
 - ✓ Bộ vi xử lý được chế tạo trên một con chip
 - ✓ Vi mạch điều khiển tổng hợp (Chipset)
 - ✓ Bộ nhớ bán dẫn độc lập (ROM, RAM) thiết kế thành Module
 - ✓ Các bộ vi điều khiển chuyên dụng.
- Bộ vi xử lý đầu tiên của Intel 4004 năm 1971
- Bộ xử lý được coi hoàn thiện nhất là 8088/8086 năm 1978, 1979 đây được coi là ngày sinh nhật của các máy tính sau này

Máy Micral, André Trương Trọng Thi sáng chế

- Micral Pháp, máy vi tính lắp ráp hoàn toàn đầu tiên



1.3 Sự tiến hóa của máy tính

Lịch sử phát triển máy tính thế hệ thứ 4

1978	8086 (Intel)	
1979	8088 (Intel)	
1980	80286 (Intel)	
1993	Pentium (Intel)	
1997	Pentium II (Intel)	Celeron
1999	Pentium III (Intel)	Celeron
2003	Pentium 4 (Intel)	Celeron

Chương 2

Hệ thống máy tính

- 2.1 Các thành phần cơ bản của máy tính
- 2.2 Hoạt động cơ bản của máy tính
- 2.3 Liên kết hệ thống

2.1 Các thành phần cơ bản của máy tính

- **Mô hình cơ bản của máy tính.**

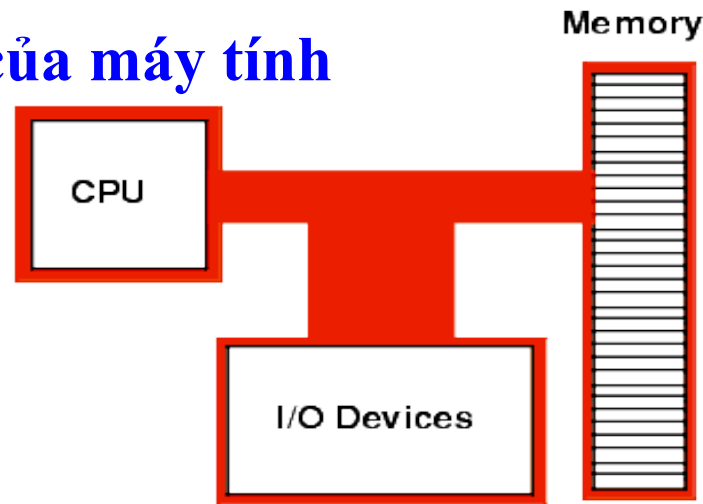
Các mô hình máy tính hiện nay được thiết kế dựa trên kiến trúc Von Neumann.

- **Các đặc điểm kiến trúc của Von Neumann:**

- ✓ Dữ liệu và chương trình chứa trong bộ nhớ đọc ghi.
- ✓ Bộ nhớ được đánh địa chỉ cho các ngăn nhớ không phụ thuộc vào nội dung của chúng.
- ✓ Máy tính thực hiện lệnh một cách tuần tự.

2.1 Các thành phần cơ bản của máy tính

▪ Sơ đồ cấu trúc cơ bản của máy tính



- ✓ **Bộ xử lý trung tâm (CPU):** Điều khiển hoạt động của máy tính và xử lý số liệu
- ✓ **Hệ thống nhớ:** chứa chương trình và dữ liệu đang được xử lý.
- ✓ **Hệ thống vào/ra (I/O: Input/Output) :** trao đổi thông tin giữa bên ngoài và bên trong máy tính
- ✓ **Liên kết hệ thống (Interconnection):** kết nối và vận chuyển thông tin giữa các thành phần với nhau

1. Bộ xử lý trung tâm

(CPU: Central Processing Unit)

Chức năng: Điều khiển toàn bộ hoạt động của máy tính.

Xử lý dữ liệu (vd: các phép toán số học và logic)

Nguyên tắc hoạt động: CPU hoạt động theo chương trình nằm trong bộ nhớ chính.

Cấu trúc cơ bản CPU

- ✓ Đơn vị điều khiển (CU: Control Unit): Điều khiển hoạt động của máy tính theo chương trình đã định sẵn.
- ✓ Đơn vị số học và logic (ALU: Arithmetic And Logic Unit): thực hiện các phép toán số học và logic trên các dữ liệu cụ thể.
- ✓ Tập thanh ghi (RF: Register File): Lưu trữ các thông tin tạm thời phục vụ cho hoạt động của CPU.
- ✓ Đơn vị nối ghép BUS (BIU: Bus Interface Unit): kết nối và trao đổi thông tin giữa Bus bên trong và Bus bên ngoài CPU.

1. Bộ xử lý trung tâm

(CPU: Central Processing Unit)

- ✓ Bộ vi xử lý hoạt động theo xung nhịp(clock) có tần số xác định.
- ✓ Tốc độ vi xử lý được đánh giá gián tiếp thông qua tần số xung nhịp.
- ✓ Gọi T_0 : chu kỳ xung nhịp, $f_0 = 1/T_0$ tần số xung nhịp.
- ✓ Mỗi thao tác của bộ xử lý cần kT_0 . T_0 càng nhỏ thì bộ xử lý chạy càng nhanh
- ✓ Ví dụ: Một máy tính Pentium 4 tốc độ 2GHz
Ta có $f_0 = 2\text{GHz} = 2 \cdot 10^9 \text{Hz}$
 $T_0 = 1/f_0 = 1/2 \cdot 10^9 = 0.5\text{ns}$

2. Bộ nhớ máy tính

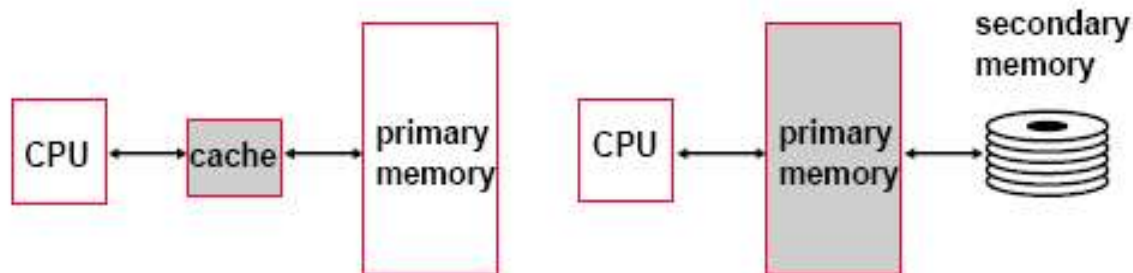
- Chức năng: Lưu trữ chương trình và dữ liệu.

Các thao tác cơ bản:

- ✓ Thao tác đọc dữ liệu (Read)
- ✓ Thao tác ghi dữ liệu (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 thông tin mà CPU có thể trao đổi trực tiếp
 - ✓ Tốc độ rất nhanh
 - ✓ Dung lượng không lớn
 - ✓ Sử dụng bộ nhớ bán dẫn RAM, ROM
- Các loại bộ nhớ
 - ✓ Bộ nhớ chính (Main memory)
 - ✓ Bộ nhớ Cache (Cache Memory) hay gọi bộ nhớ đệm



Bộ nhớ chính (main memory)

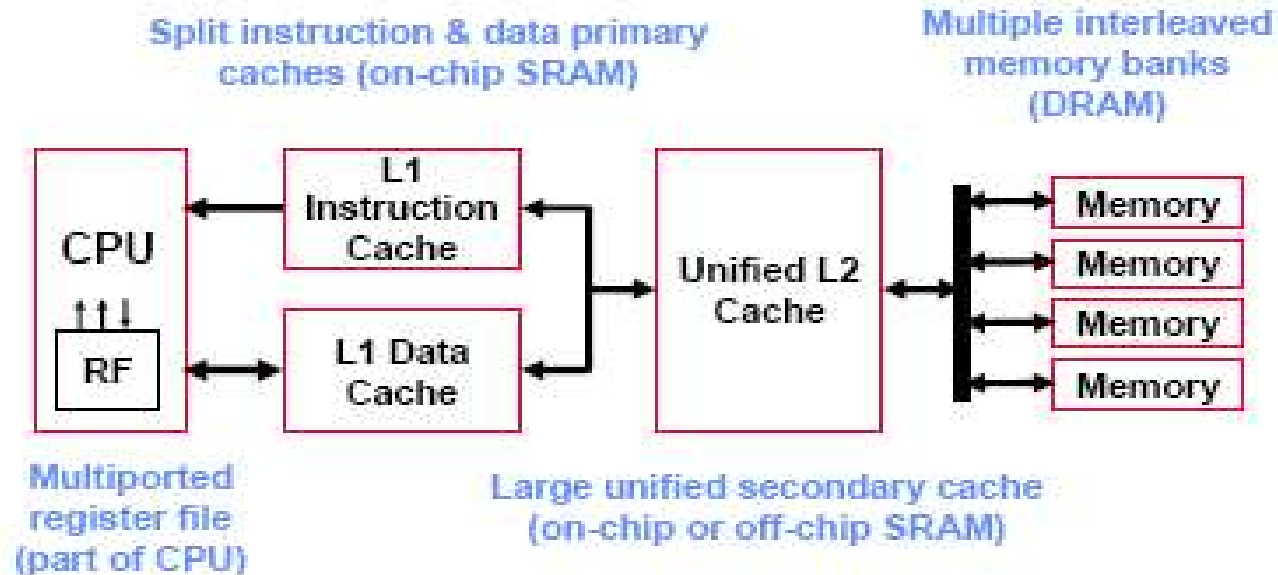
- Chứa chương trình và dữ liệu đang được sử dụng bởi CPU
- Bộ nhớ chính được tổ chức thành các ngăn nhớ và được đánh địa chỉ
- Ngăn nhớ thường được tổ chức theo byte
- Nội dung của một ngăn nhớ có thể thay đổi nhưng địa chỉ vật lý của nó đã được đánh là không thay đổi

Bộ nhớ đệm nhanh(cache memory)

- ✓ Đây là bộ nhớ bán dẫn có tốc độ nhanh và chúng được đặt đệm giữa CPU và bộ nhớ chính nhằm tăng tốc truy xuất của CPU tới bộ nhớ chính.
- ✓ Dung lượng nhỏ hơn rất nhiều bộ nhớ chính
- ✓ Tốc độ nhanh hơn rất nhiều lần
- ✓ Ngay nay Cache được tích hợp vào trong bộ vi xử lý và nó trong suốt với người sử dụng.
- ✓ Bộ nhớ Cache thông thường được chia ra thành 2 mức.
- ✓ Cache có thể có hoặc không

2. Bộ nhớ máy tính

Chi tiết cấu trúc bộ nhớ Cache



Bộ nhớ ngoài(External memory)

Chức năng và đặc điểm

- Lưu trữ tài nguyên phần mềm Máy tính.
- Được kết nối với hệ thống như thiết bị vào ra.
- Dung lượng rất lớn (vài trăm GB)
- Tốc độ chậm

Các loại bộ nhớ ngoài

- Bộ nhớ từ: Đĩa cứng, đĩa mềm,...
- Bộ nhớ quang: CD, VCD, DVD,...
- Bộ nhớ bán dẫn: flash Disk, memory Card, pen Disk,...

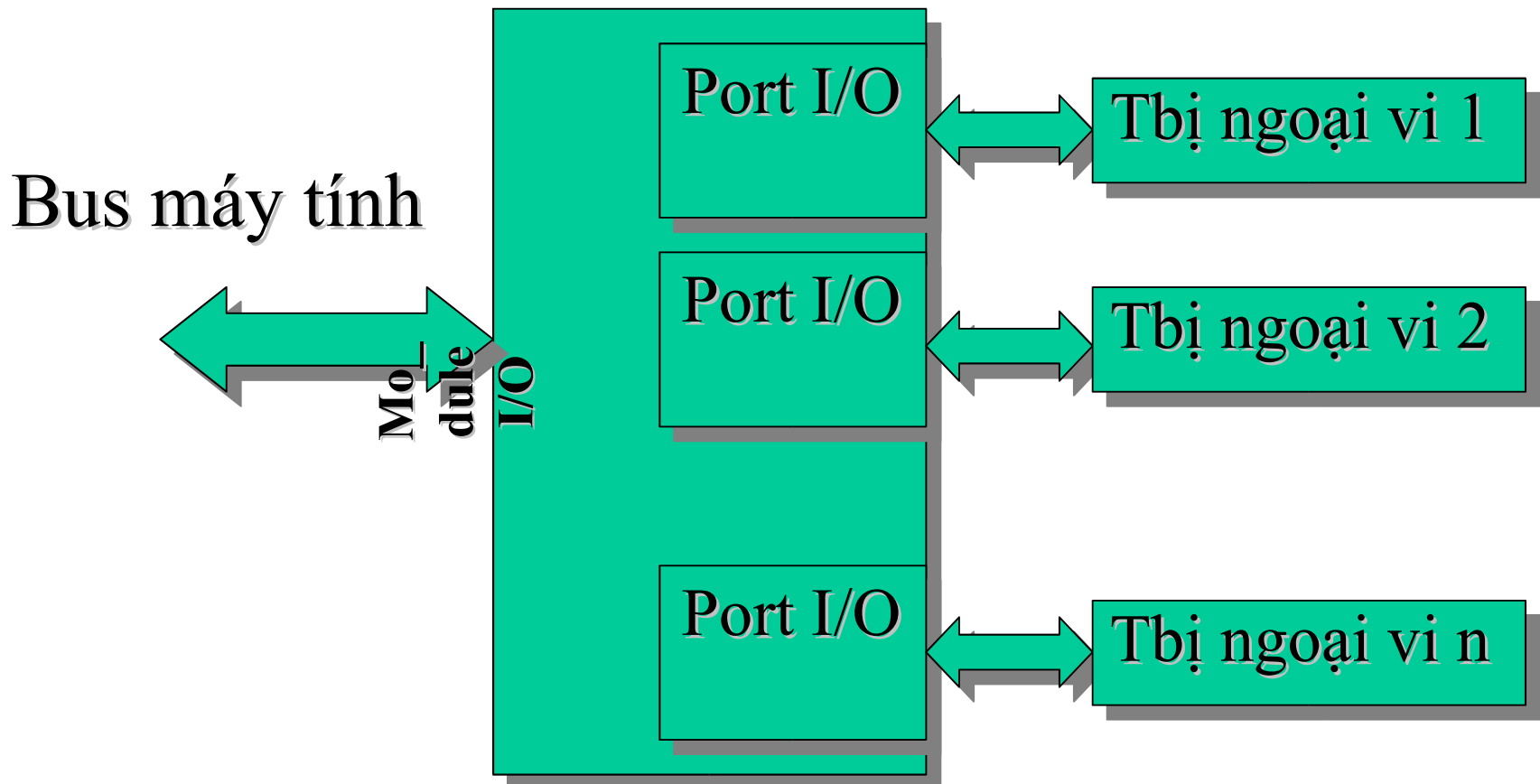


Hệ thống vào ra (Input/Output System)

- Chức năng: trao đổi thông tin giữa máy tính với thế giới bên ngoài.
- Thao tác cơ bản
 - ✓ Vào dữ liệu (In)
 - ✓ Ra dữ liệu (Out)
- Các thành phần chính
 - ✓ Thiết bị ngoại vi (Peripheral Devices)
 - ✓ Các Module I/O (IO Module)

Hệ thống vào ra (Input/Output System)

- Cấu trúc vào ra cơ bản





Thiết bị ngoại vi (Peripherals)

Các thiết bị ngoại vi (Peripherals)

- *Chức năng*: chuyển đổi thông tin từ bên ngoài thành dữ liệu máy tính và ngược lại.
- *Các thiết bị ngoại vi cơ bản*:
 - ✓ Thiết bị vào: bàn phím, chuột, ...
 - ✓ Thiết bị ra: máy in, màn hình, ...
 - ✓ Thiết bị nhớ: đĩa từ, quang, ...
 - ✓ Thiết bị truyền thông: Modem, ...



Module vào ra

Chức năng: nối ghép thiết bị ngoại vi với máy tính

- ☞ Mỗi Module có 1 hay nhiều cổng vào ra
- ☞ Mỗi cổng được đánh địa chỉ xác định

Các thiết bị ngoại vi được kết nối với máy tính thông qua cổng vào ra (ví dụ: COM, LPT, USB, VGA,...)

2.2 Hoạt động của máy tính

1. Thực hiện chương trình

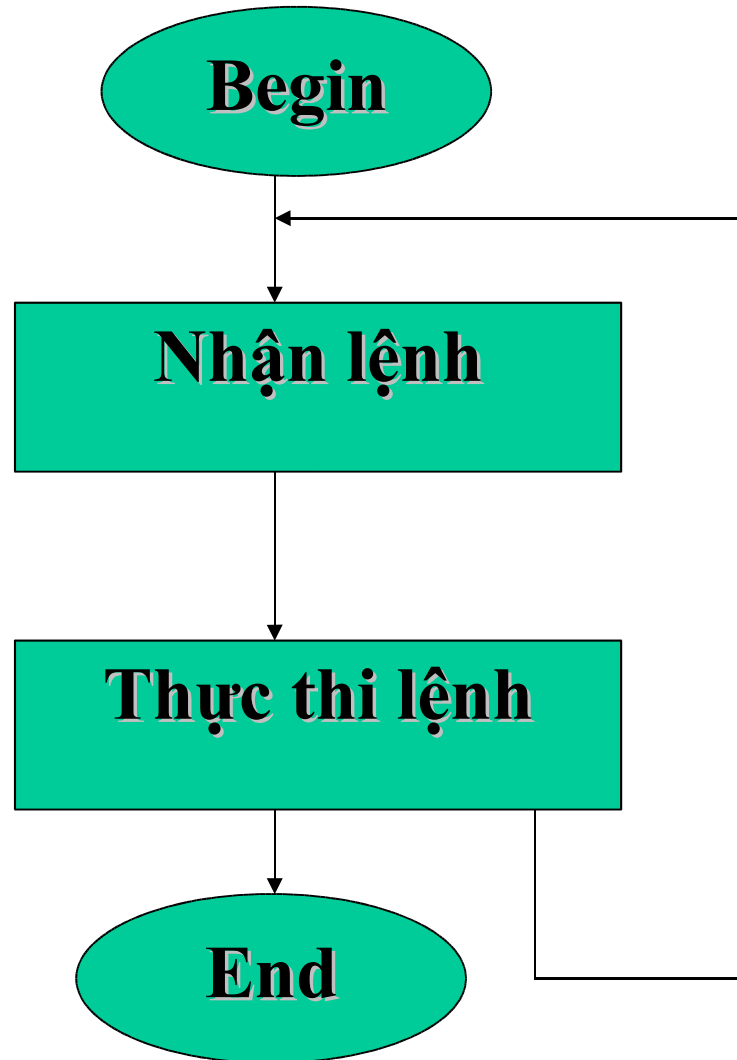
Là hoạt động cơ bản của Máy tính. Máy tính lặp đi lặp lại quá trình thực hiện lệnh gồm hai bước cơ bản:

- ✓ Nhận lệnh (Fetch)
- ✓ Thực hiện lệnh (Execute)

Thực hiện chương trình dừng khi:

- ✓ Mất nguồn
- ✓ Gặp lệnh dừng
- ✓ Gặp tình huống không giải quyết được (lỗi)

Chu kỳ thực hiện lệnh



1. Thực hiện chương trình

Nhận lệnh (Fetch)

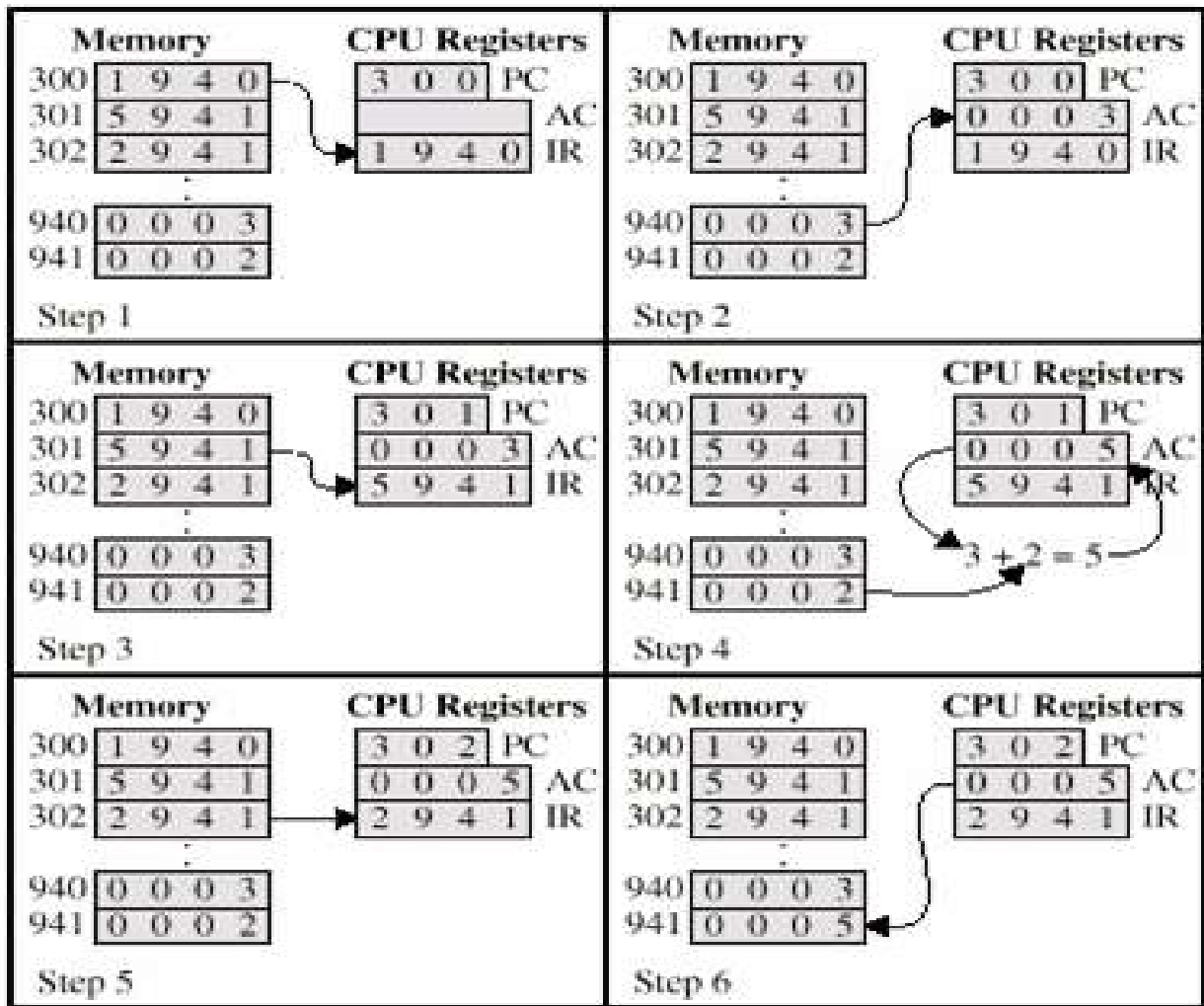
- Bắt đầu mỗi chu kỳ lệnh là CPU tiến hành lấy lệnh từ bộ nhớ chính. Trong quá trình lấy và thực hiện lệnh có 2 thanh ghi CPU mà ta quan tâm đó PC (Program Counter) và thanh ghi IR (Instruction Register)
- Bộ đếm chương trình thanh ghi PC giữ địa chỉ của lệnh *sẽ được nhận*.
- CPU lấy lệnh từ ngăn nhớ được *trở bởi PC* đưa vào thanh ghi lệnh IR lưu giữ
- Sau mỗi lệnh được nhận thì nội dung của thanh ghi PC tự động tăng để trở tới lệnh kế tiếp sẽ được thực hiện.

1. Thực hiện chương trình

Thực hiện (Execute)

- Bộ xử lý giải mã lệnh đã được nhận và phát tín hiệu điều khiển thực hiện thao tác mà lệnh yêu cầu.
- Thực hiện trao đổi giữa CPU và bộ nhớ chính
- Thực hiện trao đổi giữa CPU và Module I/O.
- Xử lý dữ liệu thực hiện các phép toán số học và logic.
- Điều khiển rẽ nhánh.
- Kết hợp các thao tác trên.

Ví dụ: Thực hiện chương trình



0001: loader
 0010: store
 0101: add

Ví dụ: Thực hiện chương trình

Evaluated $d = a + b \times c$, when $a = 5$, $b = 3$ and $c = 11$

Program Pseudocode:	Assembly Code:	Meaning:
1. Multiply b with c	MOV R1, [1058H] MOV R2, [1059H] MUL R1, R2	$R1 \leftarrow [1058H]$ $R2 \leftarrow [1059H]$ $R1 \leftarrow R1 * R2$
2. Add the result of multiplication to a	MOV R3, [1057H] ADD R1, R3	$R3 \leftarrow [1057H]$ $R1 \leftarrow R1 + R3$
3. Save the result as d	STR R1, [105AH]	$R1 \rightarrow [105AH]$

Ví dụ: Thực hiện chương trình

Following arithmetic expression to be evaluated for **d**,

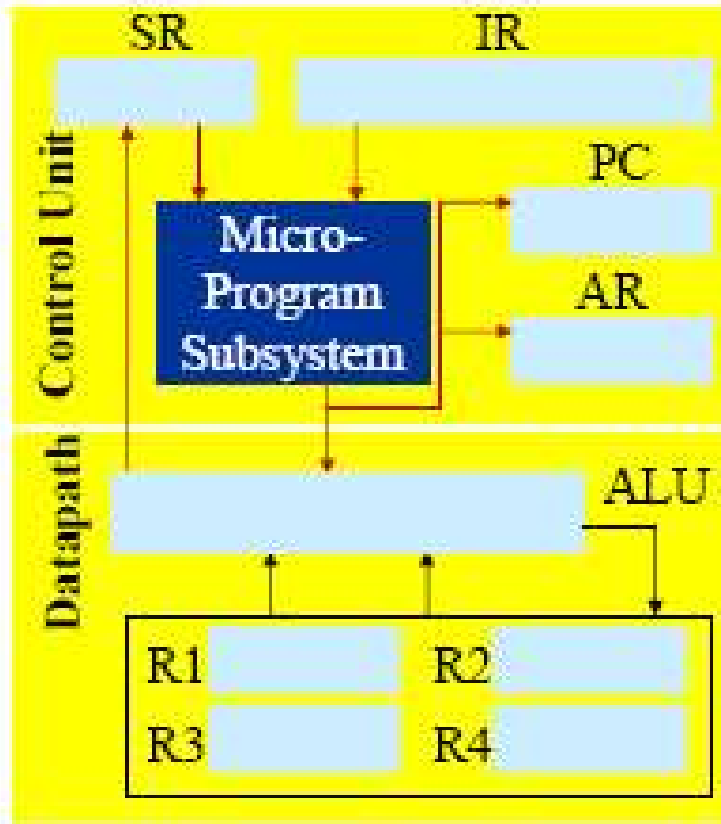
$$d = a + b * c$$

when **a** = 5, **b** = 3 and **c** = 11

Memory map shows where the program and input data are loaded

Address	Word Contents
0020H	MOV R1, [1058H]
0021H	MOV R2, [1059H]
0022H	MUL R1, R2
0023H	MOV R3, [1057H]
0024H	ADD R1, R3
0025H	STR R1, [105AH]
	⋮
1057H	a = 5
1058H	b = 3
1059H	c = 11
105AH	

Ví dụ: Thực hiện chương trình



Instruction	0020H	MOV R1, [1058H]
Address	0021H	MOV R2, [1059H]
Address	0022H	MUL R1, R2
Address	0023H	MOV R3, [1059H]
Address	0024H	ADD R1, R3
Address	0025H	STR R1, [105AH]
		⋮
Data	1057H	a = 5
	1058H	b = 3
	1059H	c = 11
	105AH	d = 38

ALU Arithmetic Logic Unit
 R1 .. R4 Data Registers
 PC Program counter

ALU Arithmetic Logic Unit
 IR Instruction Register
 SR Status Register

Animated Illustration

2. Ngắt (Interrupt)

Khái niệm chung về ngắt: Ngắt là cơ chế cho phép CPU tạm dừng chương trình đang thực hiện chuyển sang thực hiện một chương trình khác, gọi là chương trình con phục vụ ngắt.

Các loại ngắt

- ✓ Ngắt do lỗi thực hiện chương trình: chia cho 0
- ✓ Ngắt do lỗi phần cứng: lỗi RAM
- ✓ Ngắt do module I/O phát ra tín hiệu ngắt đến CPU yêu cầu trao đổi dữ liệu

Hoạt động của ngắt

2. Ngắt (Interrupt)

- ✓ Sau khi hoàn thành một lệnh, bộ xử lý kiểm tra tín hiệu ngắt.
- ✓ Nếu không có ngắt thì bộ xử lý tiếp tục nhận lệnh tiếp theo.
- ✓ Nếu có tín hiệu ngắt:

Tạm dừng chương trình đang thực hiện. cất ngữ cảnh (thông tin có liên quan đến chương trình đang thực hiện).

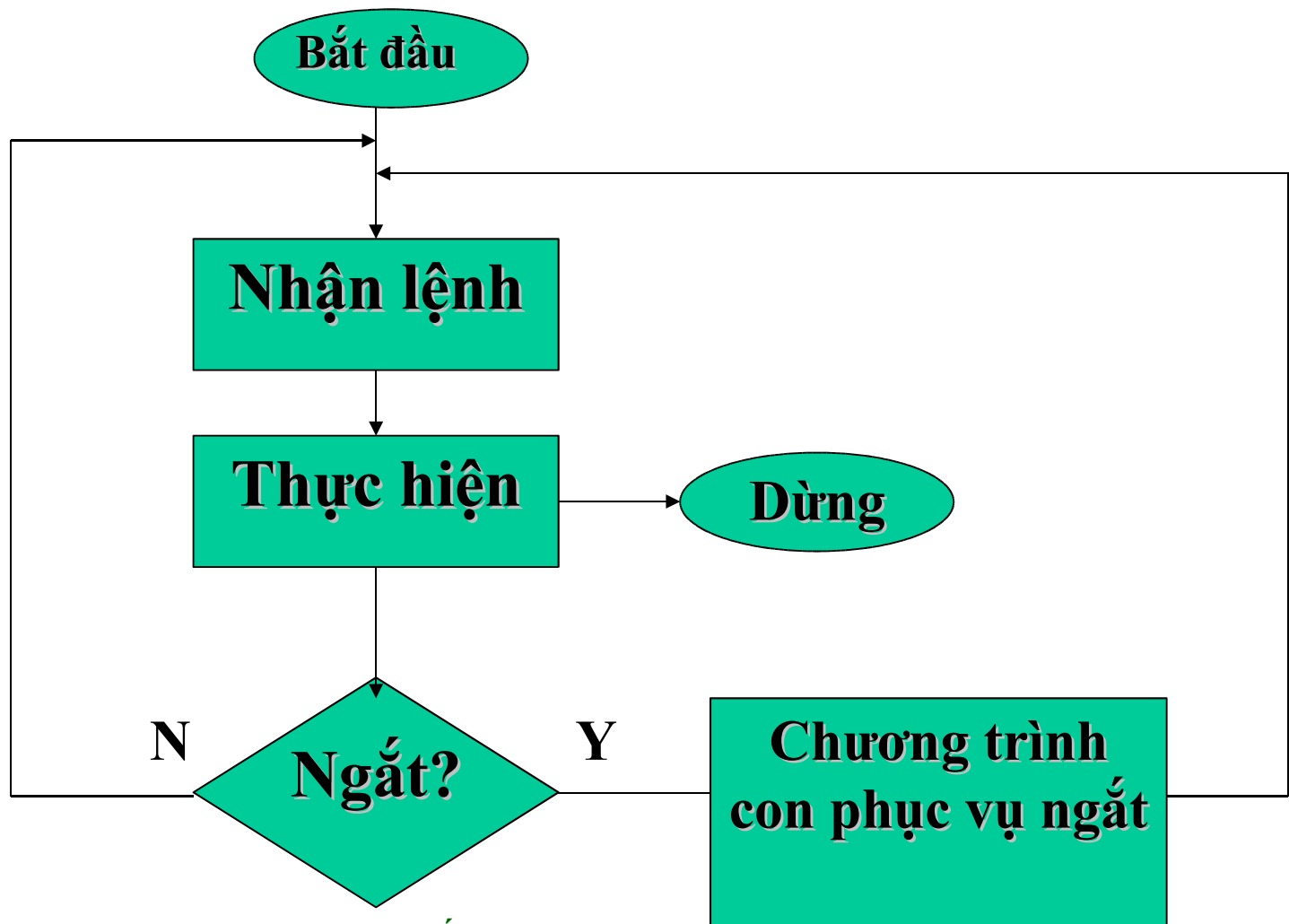
Thiết lập bộ đếm chương trình PC trở đến chương trình con phục vụ ngắt

Thực hiện chương trình con phục vụ ngắt.

Cuối chương trình con phục vụ ngắt. khôi phục lại ngữ cảnh và tiếp tục chương trình đang bị tạm dừng.

2. Ngắt (Interrupt)

- Chu kỳ lệnh với ngắt



2. Ngắt (Interrupt)

Xử lý tín hiệu ngắt

- Cấm ngắt: Bộ xử lý bỏ qua các ngắt tiếp theo trong khi đang xử lý ngắt.
- Các ngắt vẫn đang đợi và được kiểm tra sau khi ngắt đầu tiên được thực hiện xong
- Các ngắt được thực hiện tuần tự nếu cùng thứ tự ưu tiên.
- Các ngắt trong máy tính máy tính được định nghĩa mức độ ưu tiên khác nhau.
- Ngắt có mức ưu tiên thấp có thể bị ngắt bởi ngắt có ưu tiên cao hơn. Vì vậy có thể xảy ra tình trạng ngắt lồng nhau



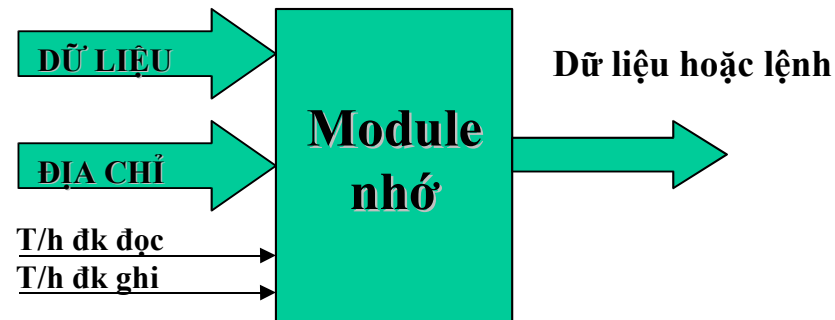
3. Hoạt động vào ra

- Là hoạt động trao đổi dữ liệu giữa thiết bị ngoại vi với bên trong máy tính
- Các kiểu hoạt động I/O: CPU trao đổi dữ liệu với module vào ra. Module vào ra trao đổi dữ liệu trực tiếp với bộ nhớ chính

2.3 Liên kết hệ thống

1. Thông tin các thành phần trong máy tính

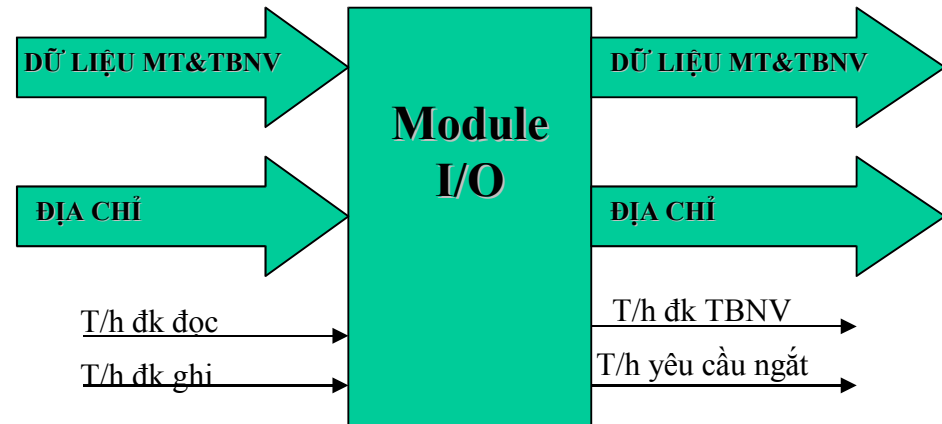
- *Kết nối Module nhớ bao gồm*



- ✓ Địa chỉ: nhận địa chỉ để xác định ngăn nhớ
- ✓ Dữ liệu: truyền nhận dữ liệu và lệnh từ bộ nhớ
- ✓ Tín hiệu điều khiển: Bao gồm tín hiệu điều khiển đọc và tín hiệu điều khiển ghi

2.3 Liên kết hệ thống

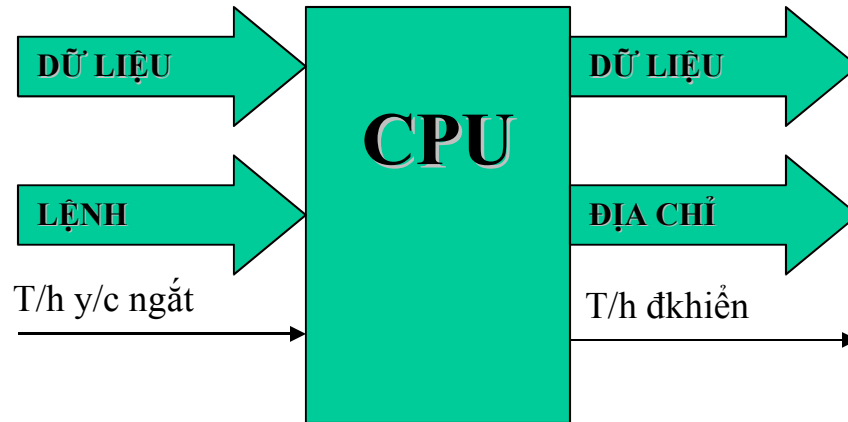
■ Kết nối Module I/O



- ✓ Địa chỉ: nhận địa chỉ để xác định cổng vào ra
- ✓ Dữ liệu: nhận dữ liệu từ thiết bị ngoại vi, CPU hay bộ nhớ chính, đưa ra dữ liệu tới thiết bị ngoại vi, CPU hay bộ nhớ chính.
- ✓ Nhận các tín hiệu điều khiển từ CPU
- ✓ Phát tín hiệu điều khiển đến TBNV
- ✓ Phát tín hiệu yêu cầu của TBNV tới CPU

2.3 Liên kết hệ thống

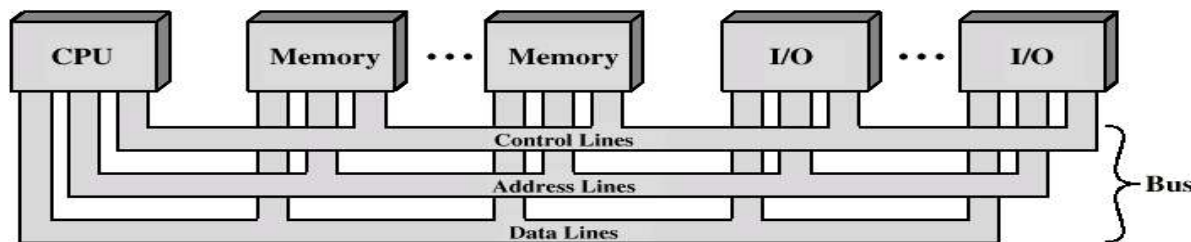
▪ Kết nối CPU



- ✓ CPU phát địa chỉ đến bộ nhớ hay Module vào ra.
- ✓ Đọc lệnh và dữ liệu
- ✓ Đưa dữ liệu ra sau khi xử lý
- ✓ Phát tín hiệu điều khiển đến Module nhớ hay Module vào ra
- ✓ Nhận các tín hiệu ngắt.

2. Cấu trúc BUS

- **Khái niệm BUS:** Bus là tập hợp các đường dây dùng để vận chuyển thông tin từ thành phần này tới thành phần khác bên trong máy tính.
- **Độ rộng của BUS :** là số đường dây có khả năng vận chuyển các bit thông tin đồng thời.
- **Phân loại BUS:** theo chức năng ta chia bus ra làm 3 loại: BUS địa chỉ, BUS dữ liệu và BUS điều khiển



2.3 Liên kết hệ thống

▪ BUS địa chỉ :





✓ *Chức năng*: dùng để vận chuyển địa chỉ từ CPU đến các Module nhớ hay các Module vào ra, nhằm để xác định ngăn nhớ hay cổng vào ra nào cần truy xuất trao đổi thông tin. (đây là BUS một chiều).

Độ rộng của BUS địa chỉ (A_0, A_1, \dots, A_{n-1})

Cho biết khả năng quản lý cực đại số các ngăn nhớ. Nếu sử dụng độ rộng bus địa chỉ n đường thì dung lượng cực đại của bộ nhớ có thể quản lý là 2^n ngăn nhớ hay tương đương với 2^n byte nhớ (nếu mỗi ngăn nhớ 1 byte)

2.3 Liên kết hệ thống

Ví dụ: Bus địa chỉ của một số bộ VXL là

 8088/8086	n=20	2^{20} (1MB)
 80286	n=24	2^{24} (16MB)
 80386...Pentium	n=32	2^{32} (4GB)
 Pentium II, III,IV	n=36	2^{36} (64GB)

2.3 Liên kết hệ thống

▪ BUS dữ liệu:

Chức năng: vận chuyển lệnh từ bộ nhớ -> CPU, vận chuyển dữ liệu giữa CPU, bộ nhớ và cổng vào ra.

Độ rộng của Bus dữ liệu (D_0, D_1, \dots, D_{m-1})

Cho biết số byte có khả năng trao đổi đồng thời
 $m=8, 16, 32, 64, 128$ bit.

Ví dụ:

8088	->	$m=8$
8086	->	$m=16$
80386	->	$m=32$
Pentium	->	$m=64$

2.3 Liên kết hệ thống

▪ **BUS điều khiển:**

Tập hợp các tín hiệu điều khiển gồm có

- ✓ Các tín hiệu phát ra từ CPU để điều khiển Module nhớ và Module vào ra.
- ✓ Các tín hiệu từ Module nhớ, Module vào ra gửi đến CPU yêu cầu.
- ✓ Ngoài ra còn là BUS cung cấp nguồn tín hiệu xung nhịp (clock) với các BUS đồng bộ.
- ✓ Một số tín hiệu điển hình

2.3 Liên kết hệ thống

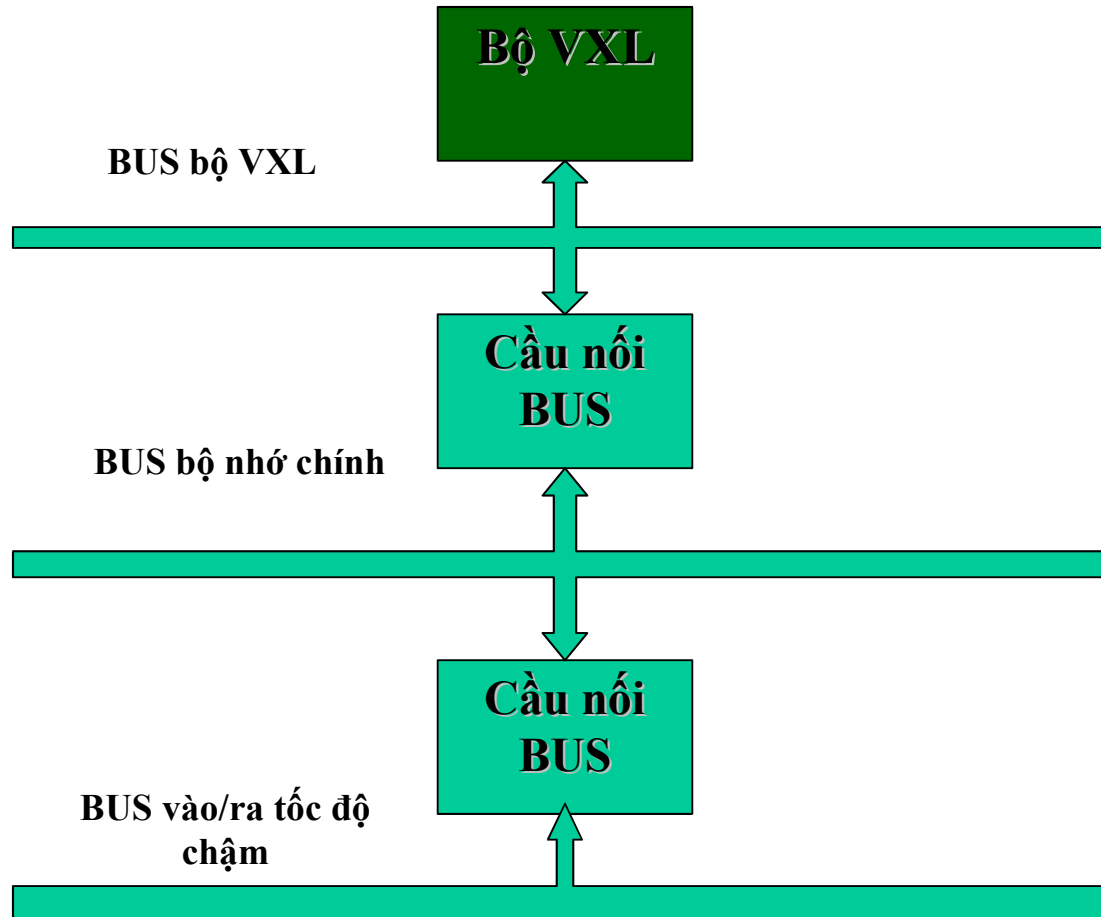
- ✓ Tín hiệu (MemR) điều khiển đọc dữ liệu từ bộ nhớ từ ngăn nhớ xác định. (IOR) Tín hiệu đọc dữ liệu từ một cổng vào ra.
- ✓ Tín hiệu (MemW) điều khiển ghi dữ liệu có sẵn trên BUS dữ liệu đến một ngăn nhớ xác định. Tín hiệu điều khiển (IOW) ghi dữ liệu có sẵn ra cổng.
- ✓ Interrupt Request(INTR) tín hiệu yêu cầu ngắt từ các thiết bị ngoại vi
- ✓ Interrupt Acknowledge(INTA) tín hiệu chấp nhận ngắt phát ra từ CPU
- ✓ Ngoài ra còn có các tín hiệu khác như: t/h yêu cầu và chấp nhận CPU chuyển nhượng BUS (BRQ,BGT),...

2.3 Liên kết hệ thống

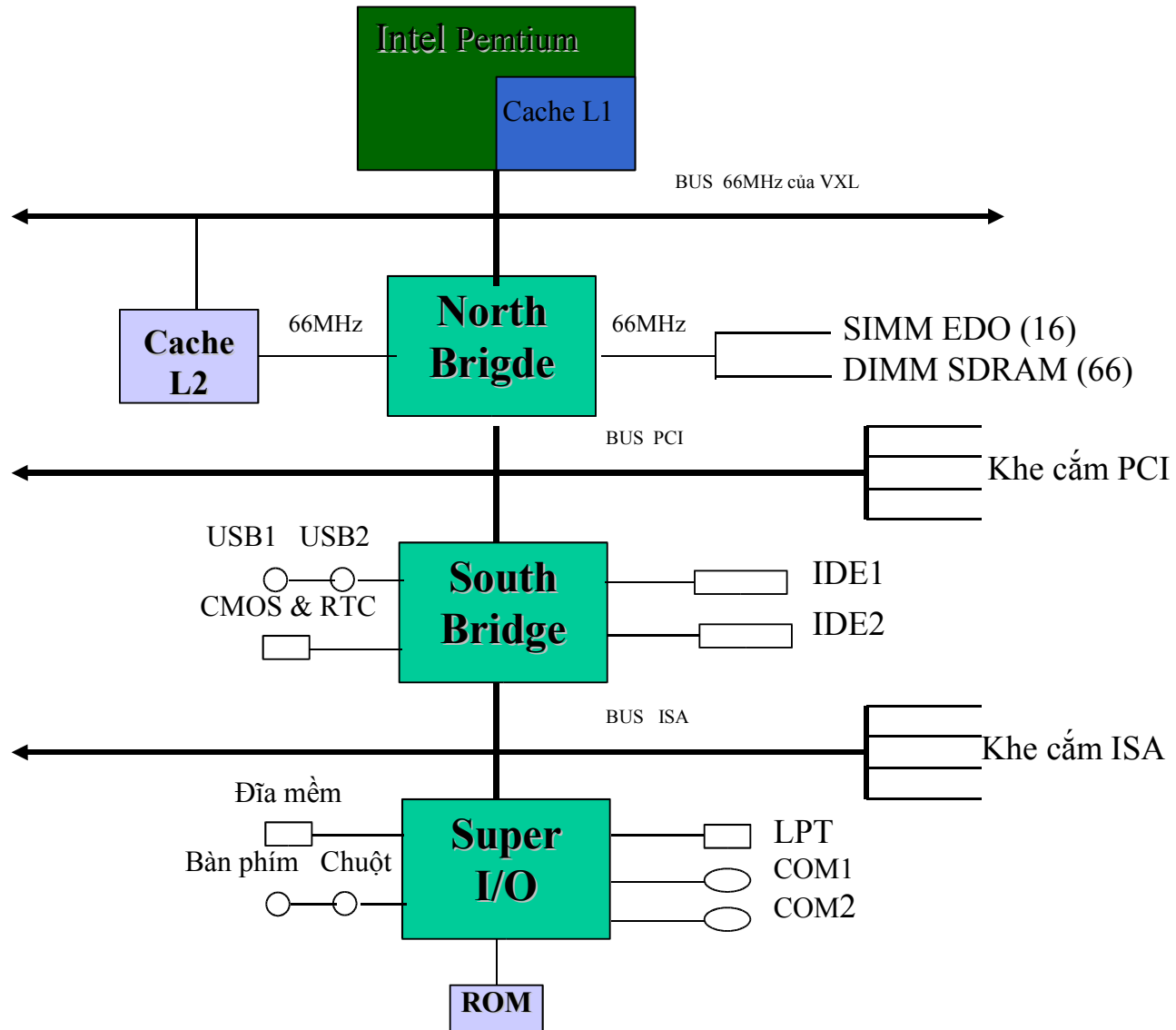
Đặc điểm của cấu trúc đơn BUS.

- Có nhiều thành phần nối vào một BUS chung.
- Tại một thời điểm chỉ phục vụ được một yêu cầu trao đổi dữ liệu.
- Các thành phần nối vào BUS có thể có tốc độ khác nhau.
- Các module nhớ và module vào ra phụ thuộc vào cấu trúc của CPU.
- ☛ Khắc phục:
 - ✓ Xây dựng cấu trúc đa BUS bao gồm các hệ thống BUS khác nhau về tốc độ.
 - ✓ Trong hầu hết các máy PC bus được phân 3 cấp và các bus nối với nhau thông qua cầu nối BUS

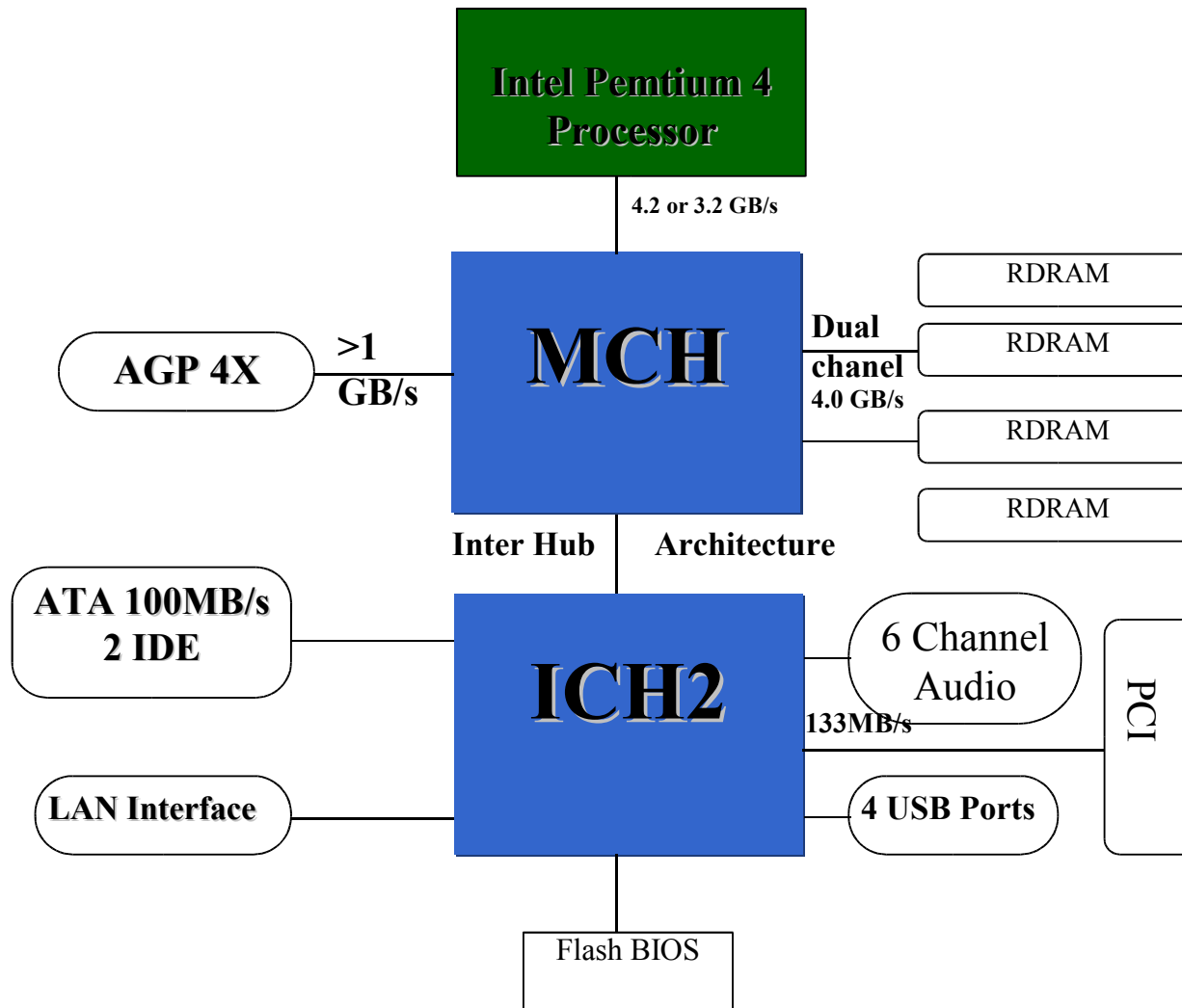
2.3 Liên kết hệ thống



Cấu trúc Pentium II điển hình



Cấu trúc Pentium 4





Ví dụ: Cấu hình một máy tính

- Intel MotherBoard D915PBL, Socket T ATX 800FSB, DDR2 533, PCI-E 16x, SATA, 8ch Audio & LAN
- 3.2GHz Pentium IV processor.
- 512 MB DDRAM.
- 80 GB hard disk.
- keyboard and a mouse,
- floppy disk drive,
- 24x speed DVD drive,
- 19" monitor with 1280 x 1024 pixels resolution,
- 56 Kbit Modem,
- 100 Mbit Ethernet card.



Phần trao đổi và giải đáp



Tóm tắt chương 2

- Đặc điểm kiến trúc Von Neumann.
- Cấu trúc và chức năng cơ bản của hệ thống máy tính.
- Quy trình thực hiện chương trình trong máy tính.
- Ngắt là gì? Tại sao phải sử dụng ngắt trong hệ thống máy tính.
- BUS máy tính? Phân loại và chức năng BUS máy tính.
- Cấu trúc đa bus trong máy tính.
- Nhận diện được tất cả các thành phần phần cứng trong máy tính của bạn.

Chương 3

Biểu diễn dữ liệu và số học máy tính

3.1 Các hệ đếm cơ bản

3.2 Mã hoá và lưu trữ trong máy tính

3.3 Biểu diễn số nguyên

3.4 Số học nhị phân

3.5 Biểu diễn số dấu chấm động

3.6 Biểu diễn ký tự



3.1 Các hệ đếm cơ bản

- ❖ Hệ thập phân (Decimal System): con người sử dụng
- ❖ Hệ nhị phân (Binary System): máy tính sử dụng
- ❖ Hệ thập lục phân (Hexadecimal System): dùng biểu diễn rút ngắn số học nhị phân
- ❖ Cách chuyển đổi giữa các hệ đếm.

Hệ thập phân (decimal)

Bộ ký tự cơ sở gồm 10 số: 0...9

Dạng tổng quát: $a_{n-1} a_{n-2} a_{n-3} \dots a_1 a_0, a_{-1} a_{-2} \dots a_{-m}$

$$A = \sum_{i=-m}^{n-1} a_i * 10^i \quad \text{Trong đó } (a_i = 0 \dots 9).$$

Ví dụ: 123,45

Phần nguyên : $123 : 10 = 12 \text{ dư } 3$

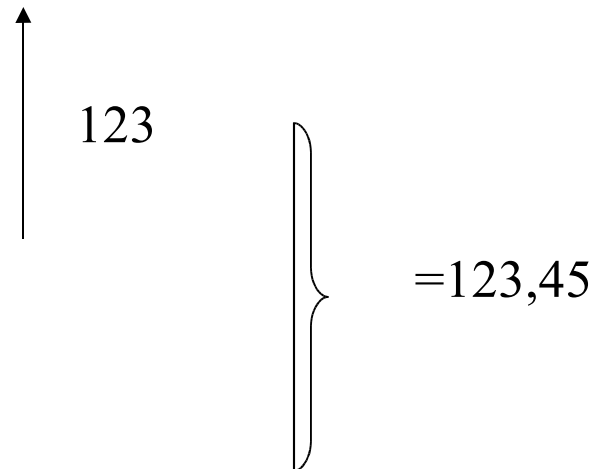
$12 : 10 = 1 \text{ dư } 2$

$1 : 10 = 0 \text{ dư } 1$

Phần phân :

$0,45 * 10 = 4,5$ ↓ 45

$0,5 * 10 = 5$



Hệ nhị phân (Binary)

Hệ thập lục phân (Hexadecimal)

Hệ nhị phân(Binary)

Bộ ký tự cơ sở gồm 2 số: 0,1

Dạng tổng quát: $a_{n-1} a_{n-2} a_{n-3} \dots a_1 a_0, a_{-1} a_{-2} \dots a_{-m}$

$$A = \sum_{i=-m}^{n-1} a_i * 2^i \quad (a_i = 0,1)$$

Ví dụ: $11011,011_2 = 2^4 + 2^3 + 2^1 + 2^0 + 2^{-2} + 2^{-3} = 27,375$

Thập lục phân (hexadecimal)

Bộ ký tự cơ sở: 0...9, A...F

Dạng tổng quát: $a_{n-1} a_{n-2} a_{n-3} \dots a_1 a_0, a_{-1} a_{-2} \dots a_{-m}$

$$A = \sum_{i=-m}^{n-1} a_i * 16^i \quad (a_i = 0..9, A..F)$$

Ví dụ: $89AB_{16} = 1000\ 1001\ 1010\ 1011B$.

3.2 Mã hoá và lưu trữ trong máy tính

Nguyên tắc chung về mã hoá dữ liệu

Mọi dữ liệu được đưa vào máy tính được mã hoá thành số nhị phân.

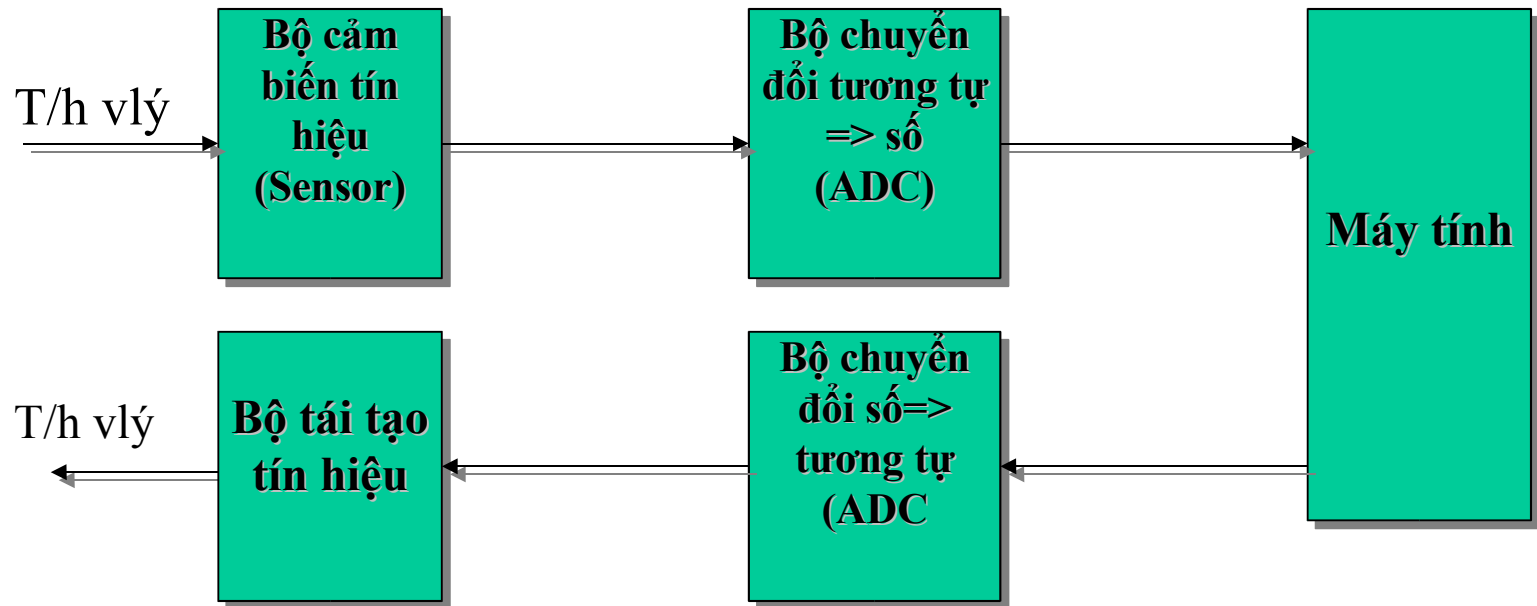
Các loại dữ liệu:

- Dữ liệu nhân tạo: do con người quy ước
- Dữ liệu tự nhiên: tồn tại khách quan với con người

Mã hoá dữ liệu nhân tạo

- Dữ liệu số nguyên: mã hoá theo một số chuẩn đã qui ước
- Dữ liệu số thực: mã hoá bằng số dấu chấm động
- Dữ liệu phi số (ký tự): mã hoá theo các bộ mã ký tự hiện hành như : ASCII, Unicode,...

Mô hình mã hoá và tái tạo tín hiệu vật lý



Các dữ liệu vật lý thông dụng

- ✓ Âm thanh
- ✓ Hình ảnh

Thứ tự lưu trữ các byte dữ liệu MT

- ❖ Bộ nhớ chính tổ chức lưu trữ dữ liệu theo đơn vị byte
- ❖ Độ dài từ dữ liệu có thể chiếm từ 1 đến 4 byte. Vì vậy cần phải biết thứ tự chúng lưu trữ trong bộ nhớ chính đối các dữ liệu nhiều byte.
- ❖ Có hai cách lưu trữ được đưa ra
 - ✓ **Little Endian** (đầu nhỏ): Byte có ý nghĩa thấp hơn được lưu trữ trong bộ nhớ ở vị trí có địa chỉ nhỏ hơn.
 - ✓ **Big Endian** (đầu to): Byte có ý nghĩa thấp hơn được lưu trữ trong bộ nhớ ở vị trí có địa chỉ lớn hơn.

Thứ tự lưu trữ các byte dữ liệu MT

☞ Ví dụ: lưu trữ một từ 32bit

0001 1010 0010 1011 0011 1100 0100 1101_B

1 A 2 B 3 C 4 D_H

Biểu diễn trong ngăn nhớ theo 2 cách

300	4D
301	3C
302	2B
303	1A

Little Endian

300	1A
301	2B
302	3C
303	4D

Big Endian

Thứ tự lưu trữ các byte dữ liệu MT

- ☞ Lưu trữ của các bộ vi xử lý điển hình
- ❖ Loại máy Intel: 80x86, Pentium -> little endian
- ❖ Motorola 680x0 và các bộ xử lý RISC -> big endian
- ❖ Power PC & Itanium: tích hợp cả hai cách trên

3.3. Biểu diễn số nguyên

Máy tính biểu diễn số nguyên chia thành 2 loại

- ❑ Biểu diễn số nguyên không dấu (unsigned integer)
- ❑ Biểu diễn số nguyên có dấu (signed integer)

Số nguyên không dấu:

Giả sử dùng n bit để biểu diễn số nguyên không dấu \rightarrow dải mà n bit biểu diễn được từ 0 $\rightarrow 2^n - 1$. Giá trị của số nguyên đó được tính:

$$\sum_{i=0}^{n-1} a_i * 2^i$$

- Dải miền trị của số nguyên không dấu được biểu bằng hình tròn
- Giá trị nhỏ nhất bằng 0
- Giá trị lớn nhất bằng $2^n - 1$



Ví dụ: Số nguyên không dấu

- Ví dụ: $n=8$ $0 \dots 2^8 - 1$ (255)
 $n=16$ $0 \dots 2^{16} - 1$ (65535).
 $n=32$ $0 \dots 2^{32} - 1$

Số nguyên có dấu

▪ Số bù một và số bù hai

ĐN: Cho một số nhị phân N được biểu diễn bởi n bit. Ta có

✓ Số bù một của N bằng $(2^n - 1) - N$

✓ Số bù hai của N bằng $2^n - N$

Ví dụ: Cho số $N = 0001\ 0001_2$ được biểu diễn bởi $n=8$ bit.

Xác định số bù 1 và bù 2 của N .

Ap dụng công thức

$$\begin{array}{r} \underline{\quad 1111\ 1111} \quad (2^n - 1) \\ \underline{\quad 0001\ 0001} \quad N \\ \hline \end{array}$$

$$\text{số bù một của } N \quad 1110\ 1110$$

Số nguyên có dấu

→ Nhận xét: số bù một của một số N được xác định bằng cách đảo các bit trong N

$$\begin{array}{r} \text{Ap dụng công thức} \\ 1\ 0000\ 0000\ (2^n) \\ - \quad 0001\ 0001\ N \\ \hline \text{số bù hai của } N \quad 1110\ 1111 \end{array}$$

→ Nhận xét: số bù hai của một số N được xác định bằng cách lấy số bù một của N cộng thêm 1

$$\text{Số bù 2 của } N = (\text{số bù 1 của } N) + 1$$

Số nguyên có dấu

Giả sử dùng n bit để biểu diễn số nguyên có dấu \rightarrow dải mà n bit biểu diễn được từ $(-2^{n-1} \dots -1, 0 \dots 2^{n-1}-1)$. Giá trị của số nguyên đó được tính theo 2 phần riêng biệt:

- Phần giá trị dương $(0 \rightarrow 2^{n-1}-1)$.
- Phần giá trị âm $(-2^{n-1} \dots -1)$.
- Dải miền trị của số nguyên có dấu được biểu bằng hình tròn
- Giá trị nhỏ nhất bằng -2^{n-1}
- Giá trị lớn nhất bằng $+2^{n-1}-1$

Số nguyên có dấu

Trong đó: Bit có trọng số cao nhất (hay bit ngoài cùng bên trái của dãy nhị được máy tính sử dụng để biểu diễn dấu của giá trị) nếu:

= 0 : thì số nhị phân cần tính giá trị là số dương.

Dạng tổng quát là: $0a_{n-2}a_{n-3}\dots a_0$

= 1 : thì số nhị phân cần tính giá trị là số âm.

Dạng tổng quát là: $1a_{n-2}a_{n-3}\dots a_0$

Ví dụ 1

Ví dụ 1: Cho số nguyên có dấu biểu diễn $n=8\text{bit}$ sau:

$$A=B5_H \text{ và } B=6A_H$$

Hãy xác định giá trị của hai số nguyên có dấu A và B dưới dạng hệ số người sử dụng

Bài giải

- Biểu diễn số nguyên A dưới dạng nhị phân

$$A=B5_H = 1011\ 0101_2$$

$$\Rightarrow A = -128 + 53 = -75$$

- Biểu diễn số nguyên B dưới dạng nhị phân

$$B=6A_H = 0110\ 1010_2$$

$$\Rightarrow B = 64+32+8+2 = 106$$

Ví dụ 2

Ví dụ 2: Biểu diễn số nguyên có dấu sau đây $A=+97$ và $B=-101$ theo hai dạng kiểu $n=8\text{bit}$ và $n=16\text{bit}$ trong máy tính.

Lời giải

- ✓ Biểu diễn số A dạng số nguyên có dấu trong máy tính

$$A = 0110\ 0001_2 \quad (n=8\text{bit})$$

- ✓ Biểu diễn số B dạng số nguyên có dấu trong máy tính

$$\text{Biểu diễn số } +101 = 0110\ 0101_2$$

$$\text{Lấy bù 2} \quad 1001\ 1011_2$$

$$\Rightarrow B = -101 = 1001\ 1011_2$$

Ví dụ 2

- ✓ Biểu diễn số A dạng số nguyên có dấu trong máy tính

$$A = 0000\ 0000\ 0110\ 0001_2 \quad (n=16\text{bit})$$

- ✓ Biểu diễn số B dạng số nguyên có dấu trong máy tính

$$\text{Biểu diễn số } +101 = 0000\ 0000\ 0110\ 0101_2$$

$$\text{Lấy bù 2} \quad 1111\ 1111\ 1001\ 1011_2$$

$$\Rightarrow B = -101 = 1111\ 1111\ 1001\ 1011_2$$

3.4 Biểu diễn số dấu chấm động

➤ **Dạng tổng quát**

M.R^E Trong đó: M (Mantissa) phần định trị
R (Radix) cơ số
E (Exponent) số mũ

$$X = (-1)^s 1.M 2^{E-B}$$

Trong đó: s: là bit dấu (s=0 phần định trị là dương; s=1 phần định trị là âm)

M : là phần định trị.

E: là số mũ được dịch chuyển đi B đơn vị.

R đã được biết (R=2) máy tính lưu số dấu chấm động bao gồm hai thành phần chính

3.4 Biểu diễn số dấu chấm động

❖ Chuẩn IEEE 754-1985 phân định 3 dạng số dấu chấm động cơ bản

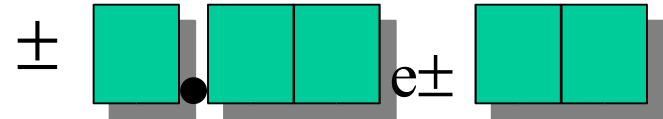
(IEEE: Institute of Electrical and Electronics Engineers)

- Số có độ chính xác đơn dài 32 bit (single)
- Số có độ chính xác kép dài 64 bit (double)
- Số có độ chính xác mở rộng dài 128bit (quadruple)

3.4 Biểu diễn số dấu chấm động

Loại	Single	Double	Quadruple
Bề rộng của trường (bit)			
S	1	1	1
E	8	11	15
M	23	52	111
Tổng cộng	32	64	128
E cực đại	255	2047	32767
E cực tiểu	0	0	0
Độ dịch	127	1023	16383

Biểu diễn số dấu chấm động chuẩn IEEE 32bit



S=1 phần định trị là âm

S=0 phần định trị là dương

E: giá trị E nằm trong 8 bit, là số mũ được dịch chuyển đi 127

M: phần định trị, giá trị nằm trong 23 bit

Ta có số $-2345,125$ trong hệ thập phân. Hãy biểu diễn chúng dưới dạng chuẩn IEEE 32bit trong máy tính

3.4 Biểu diễn số dấu chấm động

B1: Chuyển đổi số trên ra hệ hai

-2345,125d = -1001 0010 1001.001b (dãy số nhị phân được biểu diễn bình thường)

B2: Chuẩn hoá theo IEEE 32bit

-1.001 0010 1001 001 x 2¹¹

B3: Xác định các thông số biểu diễn s,M,E

S: phần định trị là số âm, nên s là 1

E : phần mũ được xác định e = E-127

=> E = 11+127=138=10001010

M: phần định trị được xác định là 001 0010 1001 0010
0000 0000 (số 32 bit)

3.4 Biểu diễn số dấu chấm động

- ☞ Để thực một phép cộng hoặc trừ hai số dấu chấm động phải tiến hành theo các bước sau:
- Tăng số mũ của số có số mũ nhỏ hơn cho bằng số có số mũ lớn hơn.
 - Cộng (hoặc trừ) các phần định trị.
 - Nếu cần thiết chuẩn hoá kết quả trả lại.

$X_1 \rightarrow M_1$ và E_1 để biểu diễn $X_1 = M_1 * R^{E1}$

$X_2 \rightarrow M_2$ và E_2 để biểu diễn $X_2 = M_2 * R^{E2}$

$$\Rightarrow X_1 * X_2 = (M_1 * M_2) * R^{E1+E2}$$

$$\Rightarrow X_1 / X_2 = (M_1 / M_2) * R^{E1-E2}$$

$$\Rightarrow X_1 \pm X_2 = (M_1 * R^{(E1-E2)} \pm M_2) * R^{E2} \text{ (với giả thiết } E1 > E2)$$

3.4 Biểu diễn số dấu chấm động

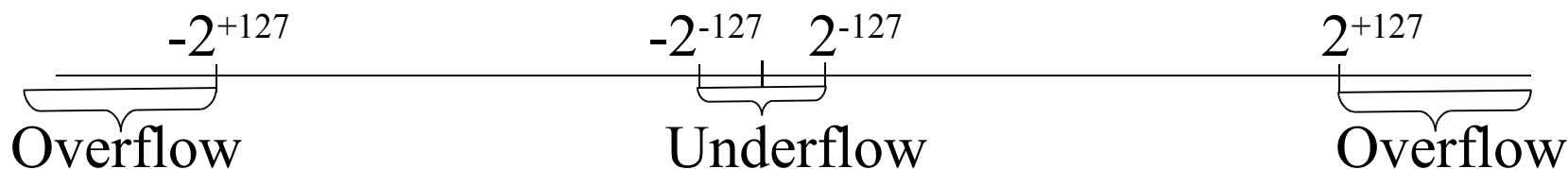
▪ Một số quy ước

Nếu $e = 255$ và $M \neq 0$ \rightarrow không phải là số

Nếu $e = 255$ và $M = 0$ \rightarrow Giá trị âm hoặc dương vô cùng

Nếu $e = 0$ và $M = 0$ \rightarrow giá trị bằng 0

Dải biểu diễn: 2^{-127} đến 2^{+127} hay tương đương 10^{-38} đến 10^{+38}



3.5 Biểu diễn ký tự.

Có hai bộ mã thường sử dụng trên máy tính:

- Bộ mã ASCII
- Bộ mã Unicode

Bộ mã ASCII (American Standard Code for Information Interchange)

- ✓ Do ANSI (American National Standard Institute) thiết kế
- ✓ Bộ mã 8 bit \rightarrow mã hoá 2^8 ký tự có mã $00_H \rightarrow FF_H$.
Trong đó

3.5 Biểu diễn ký tự.

☞ 128 ký tự chuẩn cố định có mã **(00H -> 7FH)**

✓ 33 ký tự điều khiển không thể hiện thị nên màn hình. Bao gồm các ký tự điều khiển định dạng văn bản, điều khiển truyền số liệu và điều khiển phân cách thông tin.

✓ Còn lại các ký tự còn lại hiển thị được là bao gồm:

26 ký tự hoa 41h -> 5Ah

26 ký tự thường 61h -> 7Ah

10 ký tự số 30h -> 39h

các dấu số học và ký tự đặc biệt.

3.5 Biểu diễn ký tự.

☞ 128 ký tự còn lại là ký tự mở rộng có thể thay đổi tùy ý nhà chế tạo máy tính hay người phát triển phần mềm sử dụng vào những việc riêng. Có mã $80_{\text{H}} \rightarrow \text{FF}_{\text{H}}$

Bộ mã hợp nhất Unicode:

☞ Do các hãng máy tính hàng đầu thế giới kết hợp thiết kế.

☞ Bộ mã 16 bit có thể xây dựng bộ mã toàn cầu 2^{16} ký tự với 128 ký tự đầu có mã trùng mã trong bảng mã ASCII.

☞ Có hỗ trợ các ký tự Tiếng Việt



Phần trao đổi và giải đáp

Ôn tập chương 3

- Các hệ đếm, ý nghĩa của chúng trong máy tính.
- Mã hóa dữ liệu trong máy tính.
- Mã hóa số nguyên (mã hóa số nguyên có và không dấu).
- Mã hóa số thực (số dấu chấm động)
- Mã hóa ký tự (ASCII, Unicode)
- Lưu trữ dữ liệu trong máy tính

Đặt câu hỏi

Câu 1: Kết quả hiển thị lên màn hình là bao nhiêu? Khi thực hiện đoạn lệnh sau:

```
Var a: shortint;
```

```
Begin
```

```
    a:=-1;
```

```
    writeln('Gia tri a:=',a);
```

```
    writeln('Gia tri ngan nho:=', mem[seg(a):ofs(a)]);
```

```
End.
```

Đặt câu hỏi

Câu 2: Kết quả hiển thị lên màn hình là bao nhiêu? Khi thực hiện đoạn lệnh sau:

```
Var a: shortint;
```

```
Begin
```

```
    a:=-128;
```

```
    writeln('Gia tri a:=',a);
```

```
    writeln('Gia tri ngan nho:=', mem[seg(a):ofs(a)]);
```

```
End.
```


Đặt câu hỏi

Câu 3: Kết quả hiển thị lên màn hình là bao nhiêu? Khi thực hiện đoạn lệnh sau:

```
Var a: shortint;
```

```
Begin
```

```
    a:=$6A;
```

```
    writeln('Gia tri a:=',a);
```

```
    writeln('Gia tri ngan nho:=', mem[seg(a):ofs(a)]);
```

```
End.
```

Đặt câu hỏi

Câu 4: Kết quả hiển thị lên màn hình là bao nhiêu? Khi thực hiện đoạn lệnh sau:

```
Var  b : integer absolute 3715:100;  
     a : shortint absolute 3715:100;
```

```
Begin
```

```
  b:=$00B5;
```

```
  writeln('Gia tri a:=',a);
```

```
  writeln('Gia tri ngan nho:=', mem[seg(a):ofs(a)]);
```

```
End.
```

Đặt câu hỏi

Câu 5: Kết quả hiển thị lên màn hình là bao nhiêu? Khi thực hiện đoạn lệnh sau:

```
Var    b : integer absolute 3715:100;  
      a: shortint absolute 3715:100;
```

Begin

```
  b:=-75;
```

```
  writeln('Gia tri a:=',a);
```

```
  writeln('Gia tri ngan nho:=', mem[seg(a):ofs(a)]);
```

```
  writeln('Gia tri ngan nho:=', mem[seg(a):ofs(a)+1]);
```

```
  writeln('Gia tri ngan nho:=', memw[seg(a):ofs(a)]);
```

End.

Chương 4

Bộ xử lý trung tâm

4.1 Cấu trúc của CPU

4.2 Tập lệnh (Instruction File)

4.3 Hoạt động của CPU

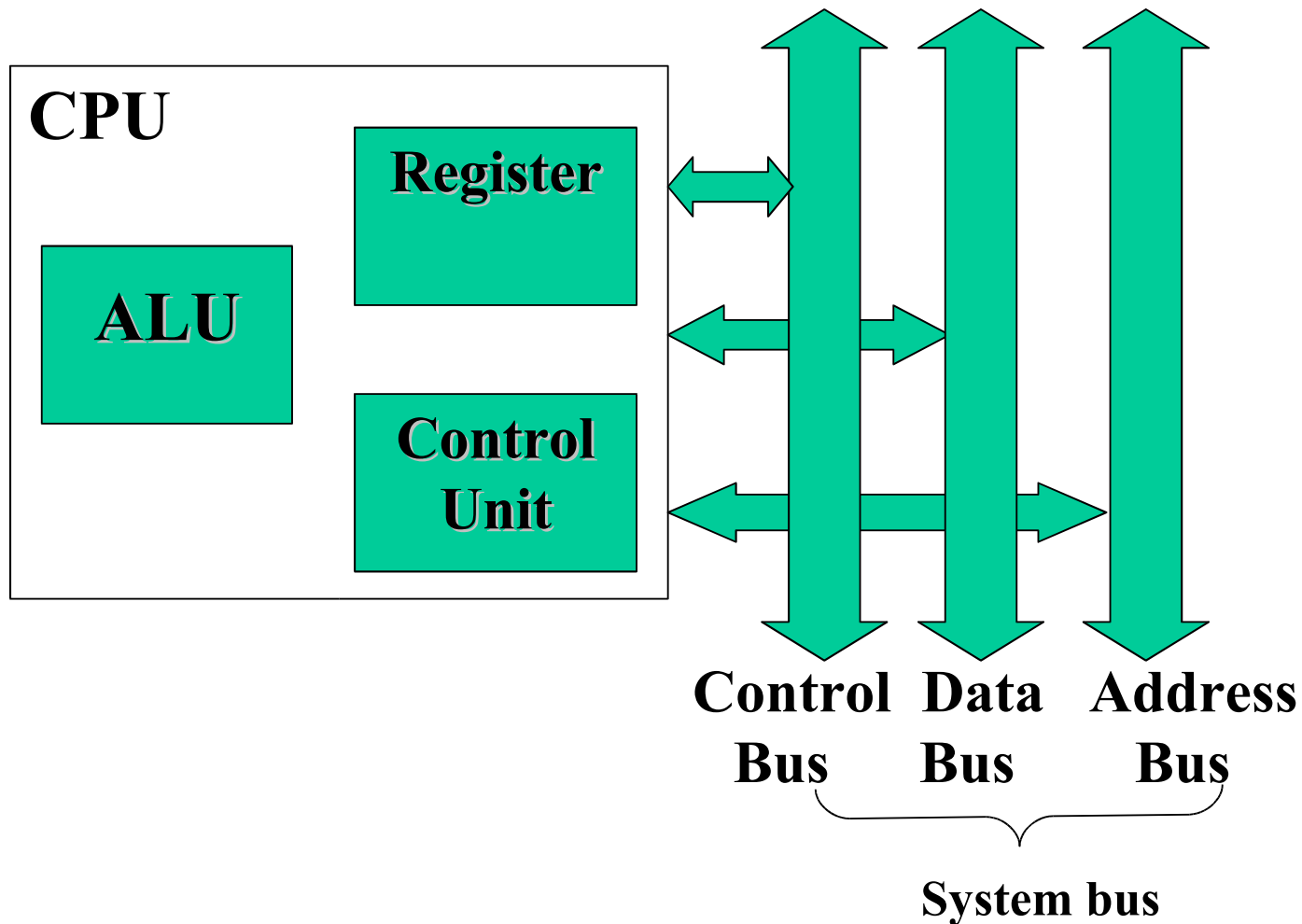
4.4 Kiến trúc Intel

4.1 Cấu trúc của CPU

Nhiệm vụ CPU: thực hiện lệnh của chương trình.

- Nhận lệnh (Fetch Instruction): CPU nhận lệnh từ bộ nhớ
- Giải mã lệnh (Decode Instruction): Xác định thao tác mà lệnh yêu cầu
- Nhận dữ liệu (Fetch Data): Nhận dữ liệu từ bộ nhớ hay cổng vào ra
- Xử lý dữ liệu (Process Data): thực hiện các phép toán số học và logic đối với dữ liệu
- Ghi dữ liệu (Write Data): Ghi dữ liệu ra bộ nhớ hay cổng vào ra.

a. Cấu trúc CPU

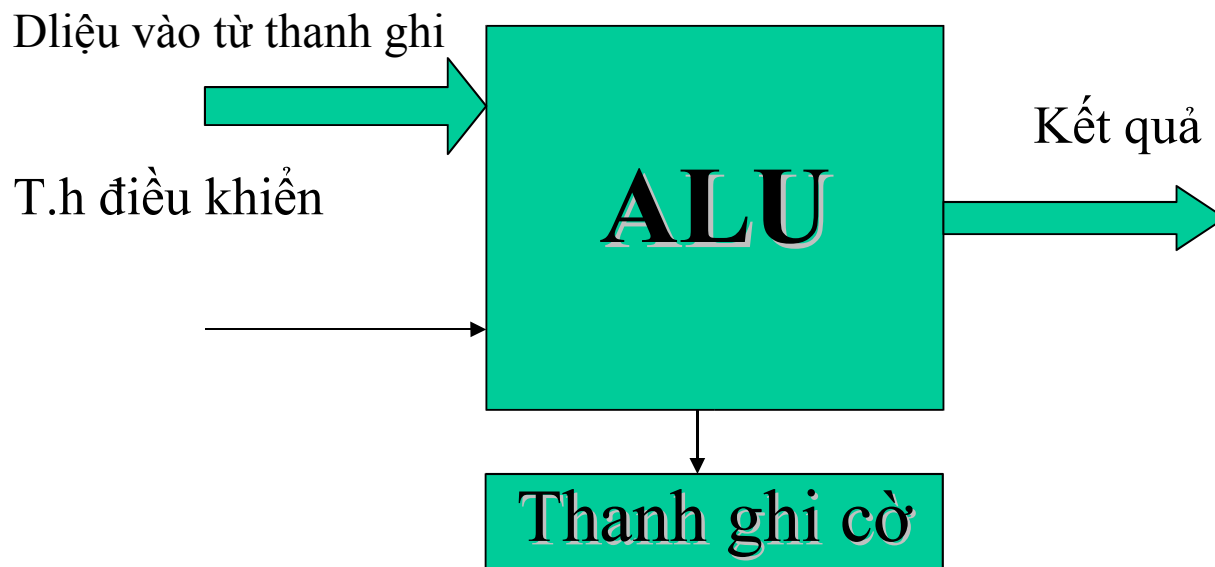


b. Đơn vị số học và logic (ALU)

Chức năng: thực các phép toán số học và logic

- Số học: cộng, trừ, nhân, chia, tăng, giảm, đảo,...
- Logic: AND, OR, XOR, NOT, dịch bit,...

■ Mô hình kết nối của ALU



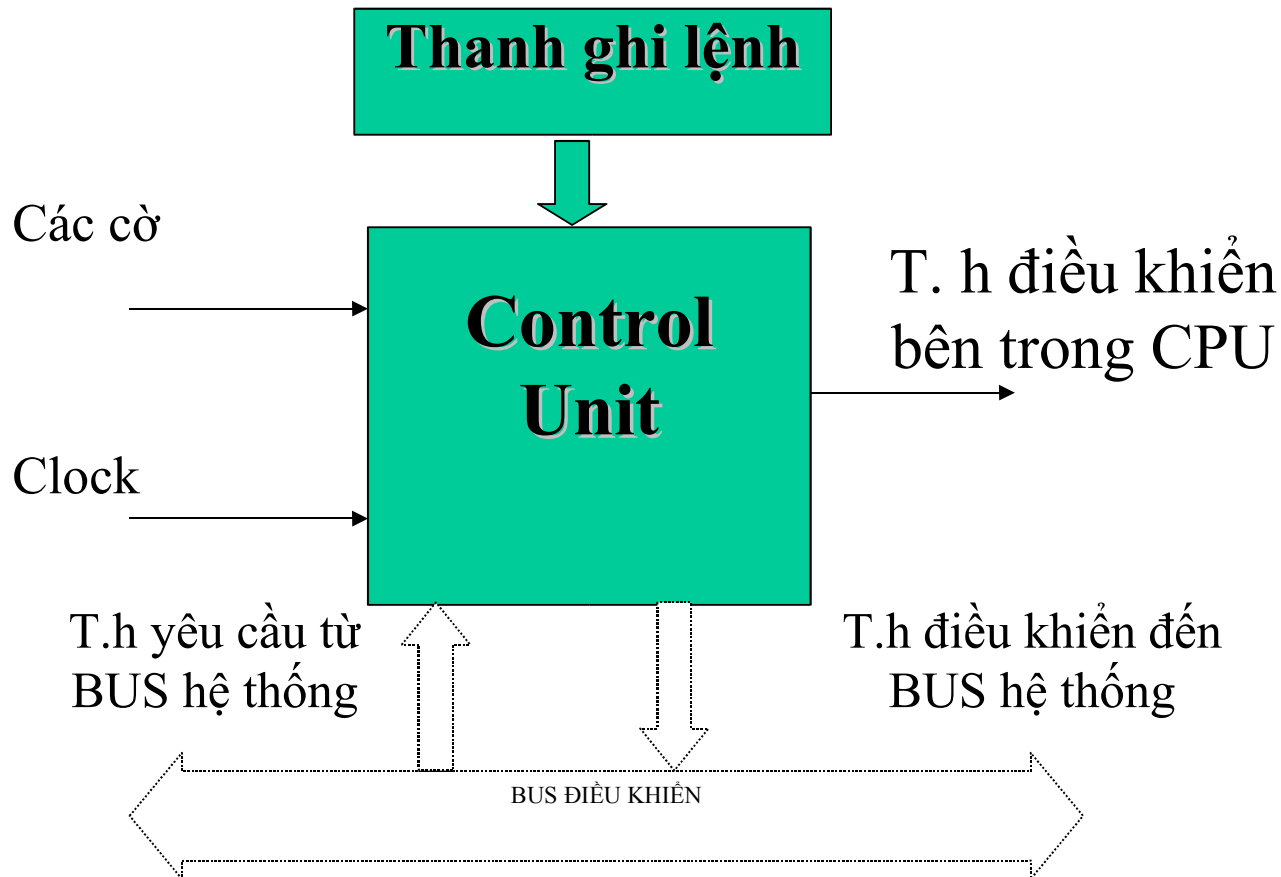
c. Đơn vị điều khiển

Chức năng:

- ✓ Nhận lệnh từ bộ nhớ đưa vào thanh ghi lệnh IP.
- ✓ Tăng nội dung thanh ghi PC mỗi khi nhận lệnh song
- ✓ Giải mã lệnh và xác định thao tác mà lệnh yêu cầu
- ✓ Phát ra tín hiệu điều khiển thực thi lệnh.
- ✓ Nhận các tín hiệu yêu cầu từ BUS hệ thống và giải quyết đáp ứng yêu cầu đó.

c. Đơn vị điều khiển

- Mô hình kết nối đơn vị điều khiển



c. Đơn vị điều khiển

- **Các thông tin kết nối đến CU**
- ✓ Clock: tín hiệu xung nhịp từ mạch tạo dao động.
- ✓ Mã lệnh từ thanh ghi lệnh đưa đến CU giải mã
- ✓ Các trạng thái cờ đưa đến cho biết trạng thái của CPU cũng như trạng thái thực hiện các phép toán trong ALU.
- ✓ Các tín hiệu điều khiển từ BUS điều khiển.
- ✓ Các tín hiệu điều khiển bên trong CPU: điều khiển thanh ghi, ALU.
- ✓ Các tín hiệu điều khiển bên ngoài CPU đó là Bộ nhớ hay cổng vào ra

4.2 Tập thanh ghi

▪ Chức năng

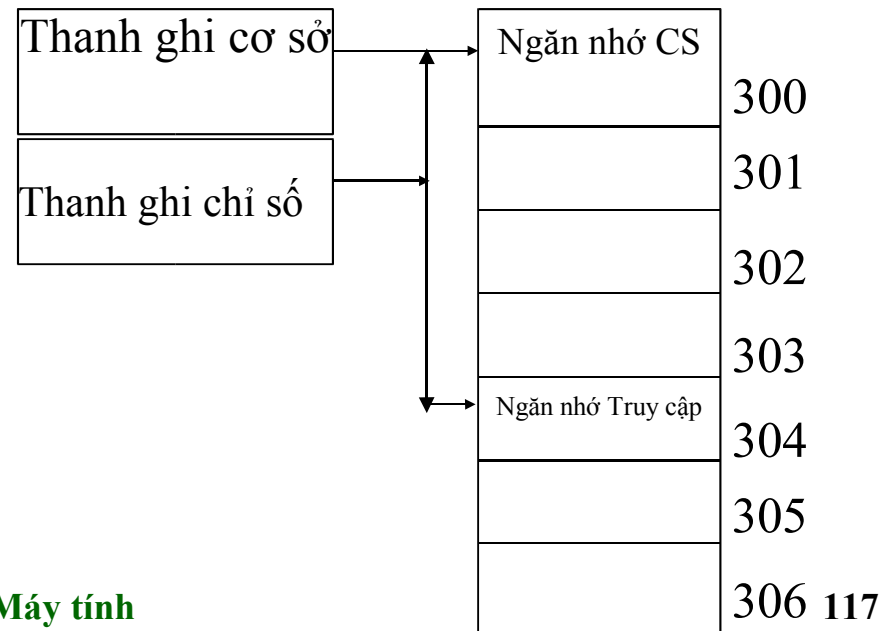
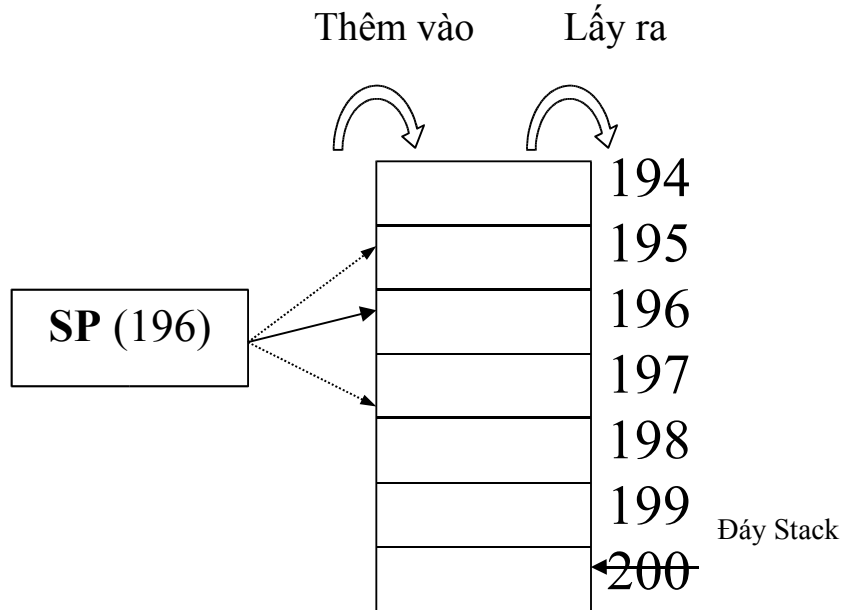
- ✓ Thực chất là vùng nhớ được CPU nhận biết qua tên thanh ghi và có tốc độ truy xuất cực nhanh.
- ✓ Chứa thông tin tạm thời phục vụ cho hoạt động ở thời điểm hiện tại của CPU
- ✓ Số lượng thanh ghi tùy thuộc vào bộ vi xử lý cụ thể -> tăng hiệu năng CPU
- ✓ Thanh ghi chia 2 loại: Loại lập trình được và loại không lập trình được

4.2 Tập thanh ghi

- **Phân loại thanh ghi theo chức năng**
- ✓ Thanh ghi địa chỉ: Thanh ghi được sử dụng để quản lý địa chỉ của ngăn nhớ hay công vào ra.
- ✓ Thanh ghi dữ liệu: Thanh ghi dùng để lưu trữ dữ liệu tạm thời
- ✓ Thanh ghi đa năng: Thanh ghi có thể chứa dữ liệu hoặc địa chỉ đều được.
- ✓ Thanh ghi điều khiển/trạng thái: Thanh ghi chứa thông tin về trạng thái CPU.
- ✓ Thanh ghi lệnh: thanh ghi chứa lệnh đang được thực hiện

4.2 Tập thanh ghi

- Một số thanh ghi điển hình
 - ✓ Bộ đếm chương trình PC
 - ✓ Ngăn xếp SS (Stack)
 - ✓ Con trỏ ngăn xếp SP



4.2 Tập thanh ghi

- **Các thanh ghi dữ liệu**
 - ✓ Chứa các dữ liệu tạm thời và kết quả trung gian.
 - ✓ Các thanh ghi số nguyên 8,16,32,64 bit.
 - ✓ Các thanh ghi số dấu chấm động.
- Thanh ghi trạng thái (State Register)
& Thanh ghi cờ (FR: Flag Register).
 - ✓ Chứa các thông tin trạng thái của CPU.
 - ✓ Các cờ phép toán báo hiệu trạng thái kết quả.
 - ✓ Các cờ điều khiển báo hiệu trạng thái của CPU
 - ✓ Ví dụ: cờ ZF, CF, SF, OF, IF (cờ ngắt =1 cho ngắt =0
cấm ngắt)

4.3 Tập lệnh

Giới thiệu chung về tập lệnh

- ❖ Mỗi bộ xử lý có tập lệnh xác định.
- ❖ Một tập lệnh thường đến vài chục đến vài nghìn lệnh
- ❖ Mỗi lệnh là chuỗi nhị phân mà bộ xử lý có thể phân tích và hiểu để thực hiện thao tác mà lệnh đó yêu cầu.
- ❖ Các lệnh khi viết thường được ánh xạ ra thành các ký hiệu gọi nhớ. ADD, MOV, IN, OUT, SHL, ROL,...
- ❖ Đây tựa của câu lệnh của hợp ngữ
(Lập ASSEMBLY)

4.3 Tập lệnh

- Các thành phần một lệnh máy 2 phần:



- ✓ Mã thao tác (Operation Code: **Opcode**): Mã chỉ ra thao tác mà bộ vi xử lý cần phải thực hiện.
- ✓ Địa chỉ toán hạng (**Operand Address**): Chỉ ra nơi chứa các toán hạng mà mã thao tác sẽ tác động.
 - Toán hạng nguồn: dữ liệu vào của thao tác
 - Toán hạng đích: dữ liệu ra của thao tác

4.3 Tập lệnh

Các kiểu thao tác

- Thao tác chuyển dữ liệu
- Thao tác xử lý số học và logic
- Thao tác vào ra dữ liệu qua cổng
- Thao tác điều khiển rẽ nhánh
- Thao tác điều khiển hệ thống
- Thao tác xử lý số dấu chấm động
- Thao tác chuyên dụng khác: xử lý ảnh, âm thanh, tiếng nói,...

4.3 Tập lệnh

Các lệnh chuyển dữ liệu

- **Lệnh Mov**
Sao chép dữ liệu từ toán hạng nguồn -> đích
- **Lệnh Load**
Nạp dữ liệu từ bộ nhớ -> bộ xử lý
- **Lệnh XCHG**
Trao đổi nội dung của hai toán hạng cho nhau
- **Lệnh PUSP**
Cất nội dung của một toán hạng nguồn vào stack
- **Lệnh POP**
Lấy nội dung ở đỉnh Stack ra toán hạng đích
- **Lệnh Set, Clear**

4.3 Tập lệnh

➤ Các lệnh số học

- **Lệnh ADD** : cộng
- **Lệnh SUB** : trừ
- **Lệnh MUL**: nhân
- **Chia DIV** : chia

➤ Các lệnh logic

- **Lệnh Test**
Thực hiện lệnh AND thiết lập cờ
- **Lệnh Shift**
Dịch trái, hoặc phải
- **Lệnh Rotate**
Quay trái hoặc quay phải
- **Lệnh Convert**
Chuyển đổi dữ liệu từ dạng này sang dạng khác
- **Lệnh AND, OR, XOR, NOT,....**

4.3 Tập lệnh

➤ Các lệnh vào ra

- **Lệnh Input:**
- **Lệnh Output**

➤ Các lệnh chuyển điều khiển

- **Lệnh Jump**
- **Lệnh Call**
- **Lệnh Return** : trở về từ chương trình con

➤ Các lệnh điều khiển hệ thống

- **Lệnh Halt** : dừng thực hiện chương trình
- **Lệnh Wait** : tạm dừng thực hiện chương trình, lặp kiểm tra cho đến khi thoả mãn thì tiếp tục thực hiện
- **No Operation**: không thực hiện gì cả
- **Lệnh Lock** : Cấm không cho chuyển nhượng BUS
- **Lệnh Unlock**: cho phép chuyển nhượng BUS

Các phương pháp định địa chỉ (Addressing Models)

- **Toán hạng của của lệnh có thể là:**
 - Một thanh ghi cụ thể
 - Nội dung của thanh ghi
 - Nội dung của ngăn nhớ hay cổng vào ra
- **Các phương pháp định địa chỉ thông dụng:**
 - Định địa chỉ tức thời
 - Định địa chỉ thanh ghi
 - Định địa chỉ trực tiếp
 - Định địa chỉ gián tiếp qua thanh ghi
 - Định địa chỉ gián tiếp
 - Định địa chỉ dịch chuyển

Định địa chỉ tức thì

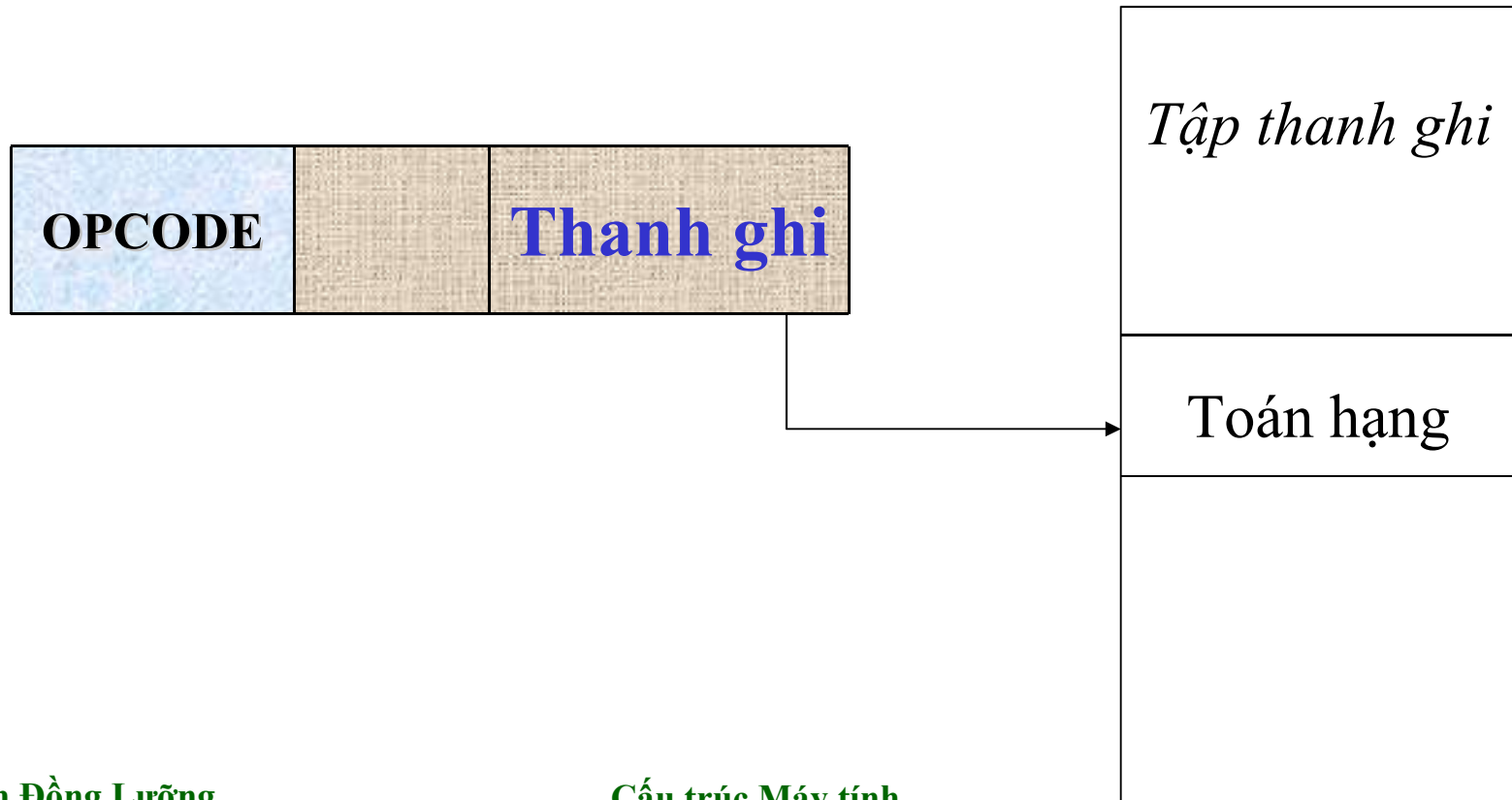
- Toán hạng là hằng số nằm ngay trong trường địa chỉ toán hạng
- Đây chỉ có thể là toán hạng nguồn
- Không tham chiếu bộ nhớ
- Truy cập toán hạng rất nhanh
- Dải giá trị toán hạng bị hạn chế

ADD R1, const



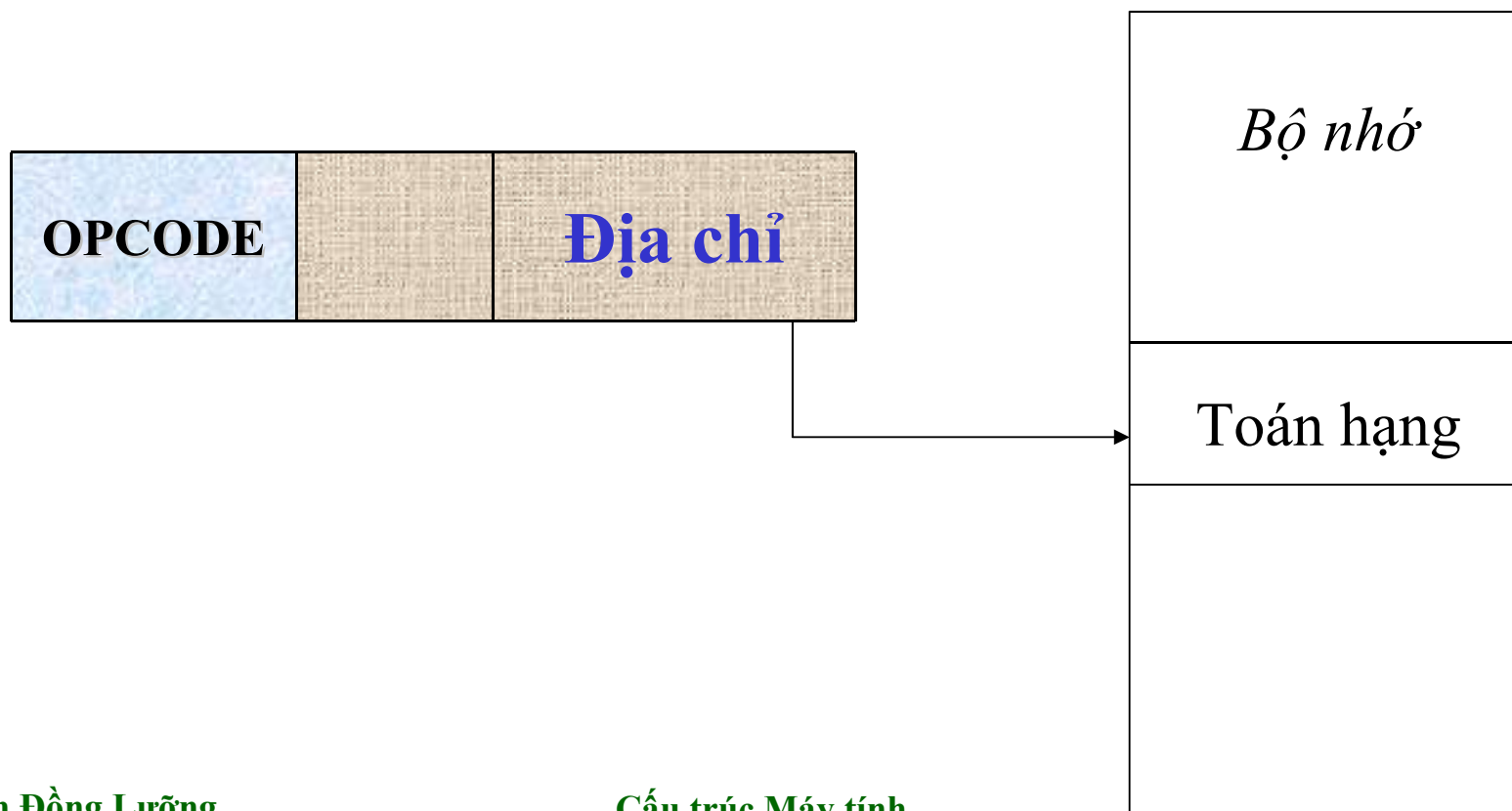
Định địa chỉ thanh ghi

- Toán hạng được chứa trong một thanh ghi, thanh ghi có tên trong trường địa chỉ toán hạng.



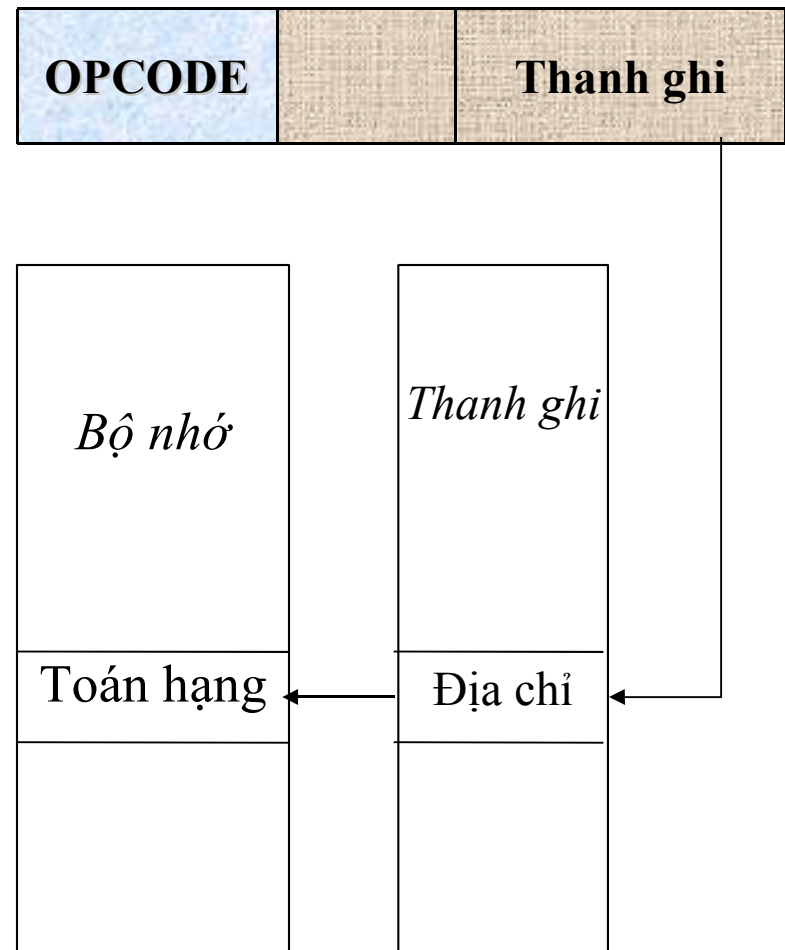
Định địa chỉ trực tiếp

- Toán hạng là ngăn nhớ có địa chỉ được chỉ ra ngay trong trường địa chỉ toán hạng

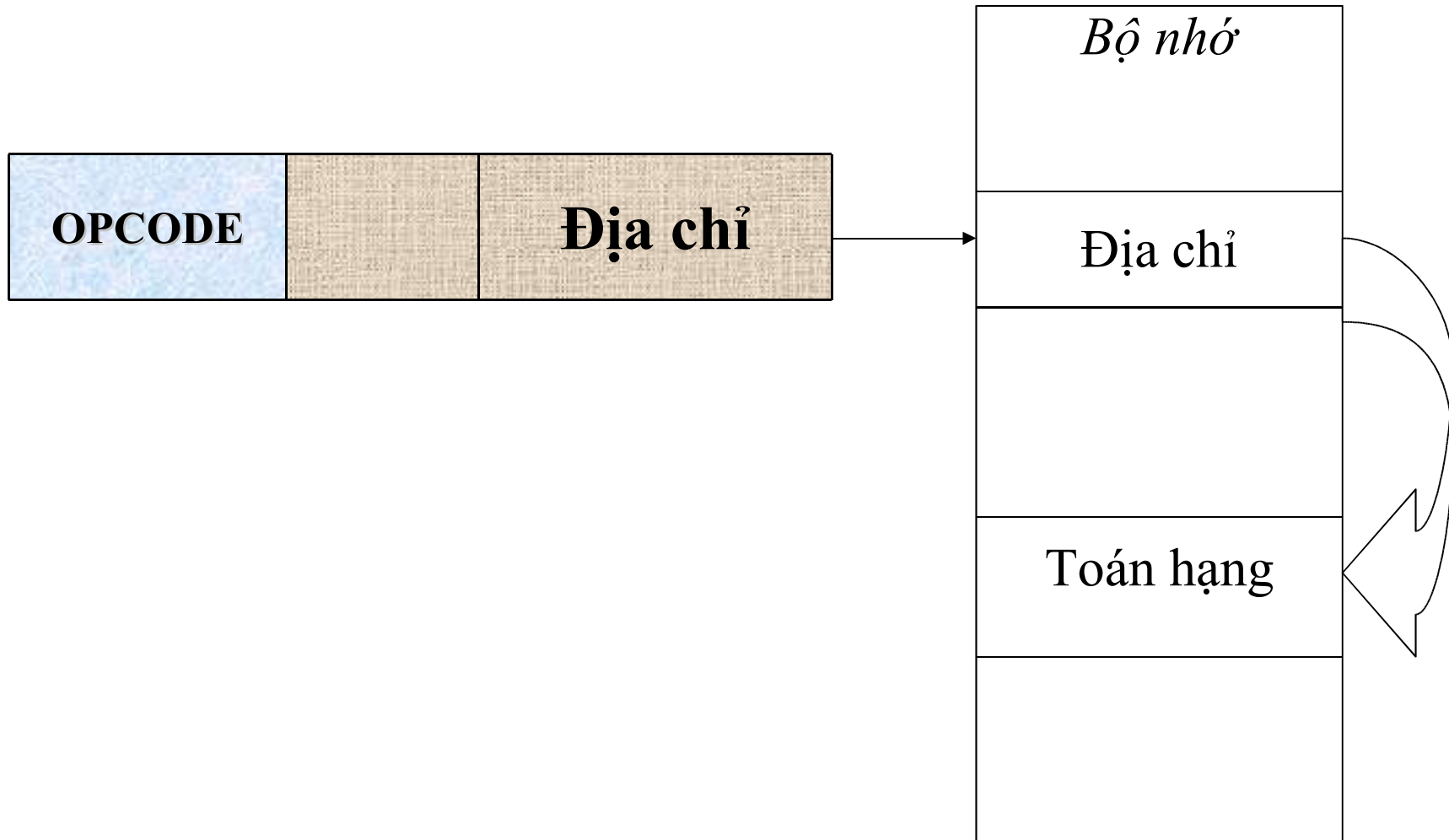


Định địa chỉ gián tiếp qua thanh ghi

- Toán hạng là ngăn ngăn nhớ có địa chỉ được chỉ ra trong thanh ghi. Trường địa chỉ toán hạng lưu trữ tên thanh ghi đó.
- Thanh ghi có thể là ngầm định
- Thanh ghi này được gọi là thanh ghi con trỏ

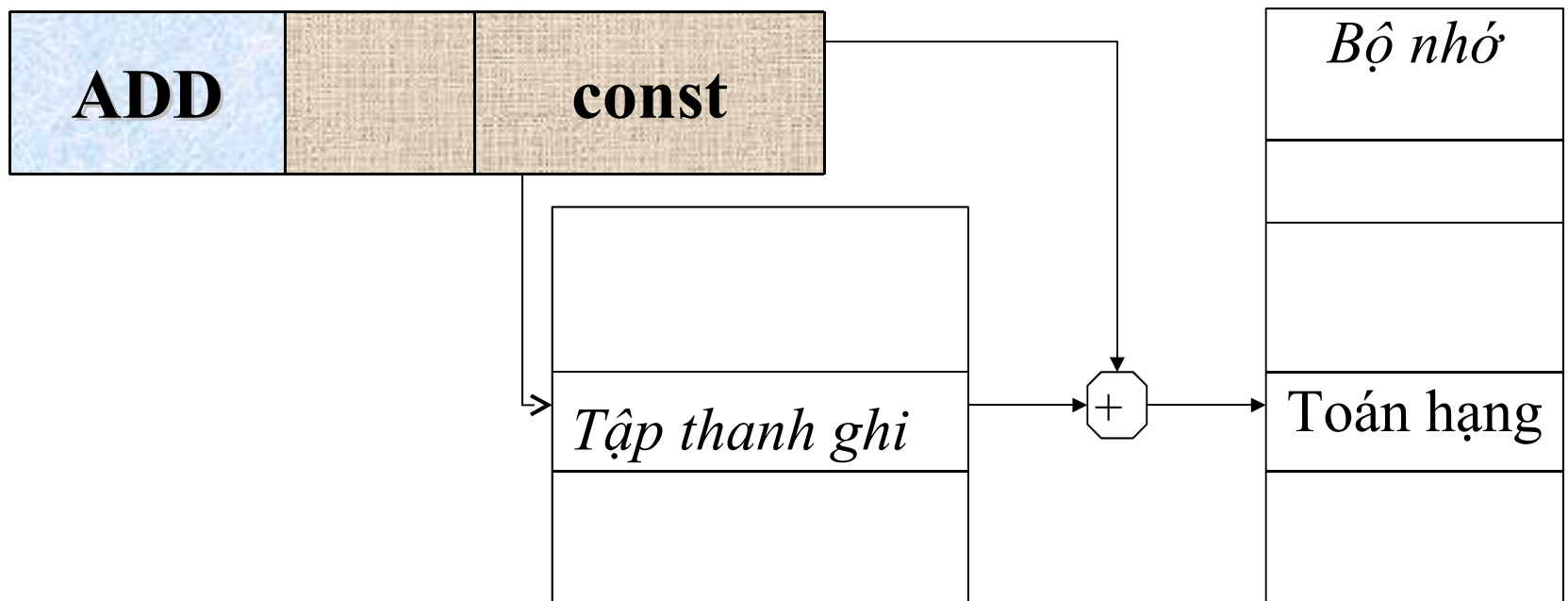


Định địa chỉ gián tiếp qua ngăn nhớ



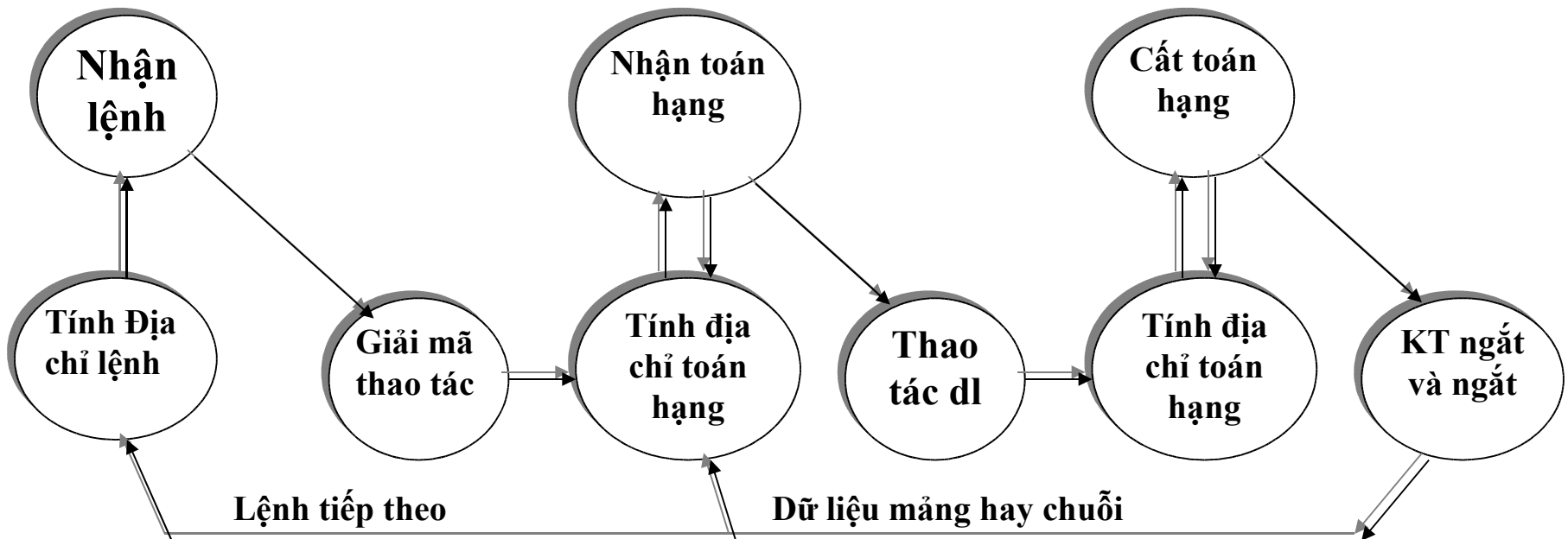
Định địa chỉ dịch chuyển

- Để xác định toán hạng gồm 2 thành phần
 - Tên thanh ghi và hằng số
 - Địa chỉ toán hạng = Nội dung thanh ghi + hằng số



4.4 Hoạt động của CPU

- Chu kỳ lệnh CPU bao gồm: Nhận lệnh, giải mã lệnh, nhận toán hạng, thực hiện lệnh, cất toán hạng, ngắt.
- Giảm đồ trạng thái chu kỳ lệnh





Phần trao đổi và giải đáp



Câu hỏi ôn tập

- Cấu trúc và chức năng của CPU
- Kiến trúc tập lệnh
- Các phương pháp tham chiếu toán hạng.
- Hoạt động cơ bản CPU
- Kiến trúc BXL tiên tiến

Chương 5

Bộ nhớ máy tính

5.1 Tổng quan bộ nhớ trong Máy tính

5.2 Bộ nhớ bán dẫn

5.3 Bộ nhớ đệm nhanh (Cache)

5.4 Bộ nhớ ngoài (bộ nhớ phụ)

5.5 Hệ thống nhớ trên máy PC hiện nay



5.1 Tổng quan

Các đặc trưng của bộ nhớ

Ví trí:

- Bên trong CPU: tập thanh ghi, cache
- Bộ nhớ trong: Bộ nhớ chính và Cache
- Bộ nhớ ngoài: các thiết bị nhớ, RAID

Dung lượng:

- Độ dài từ nhớ (tính bằng bit)
- Số lượng từ nhớ

Đơn vị truyền:

- Từ nhớ
- Khối nhớ



5.1 Tổng quan

Phương pháp truy nhập:

- Truy nhập tuần tự (băng từ)
- Truy nhập trực tiếp (các loại đĩa)
- Truy nhập ngẫu nhiên (bộ nhớ bán dẫn)
- Truy nhập liên kết (cache)

Hiệu năng:

- Thời gian truy nhập
- Chu kỳ truy xuất bộ nhớ
- Tốc độ truyền



5.1 Tổng quan

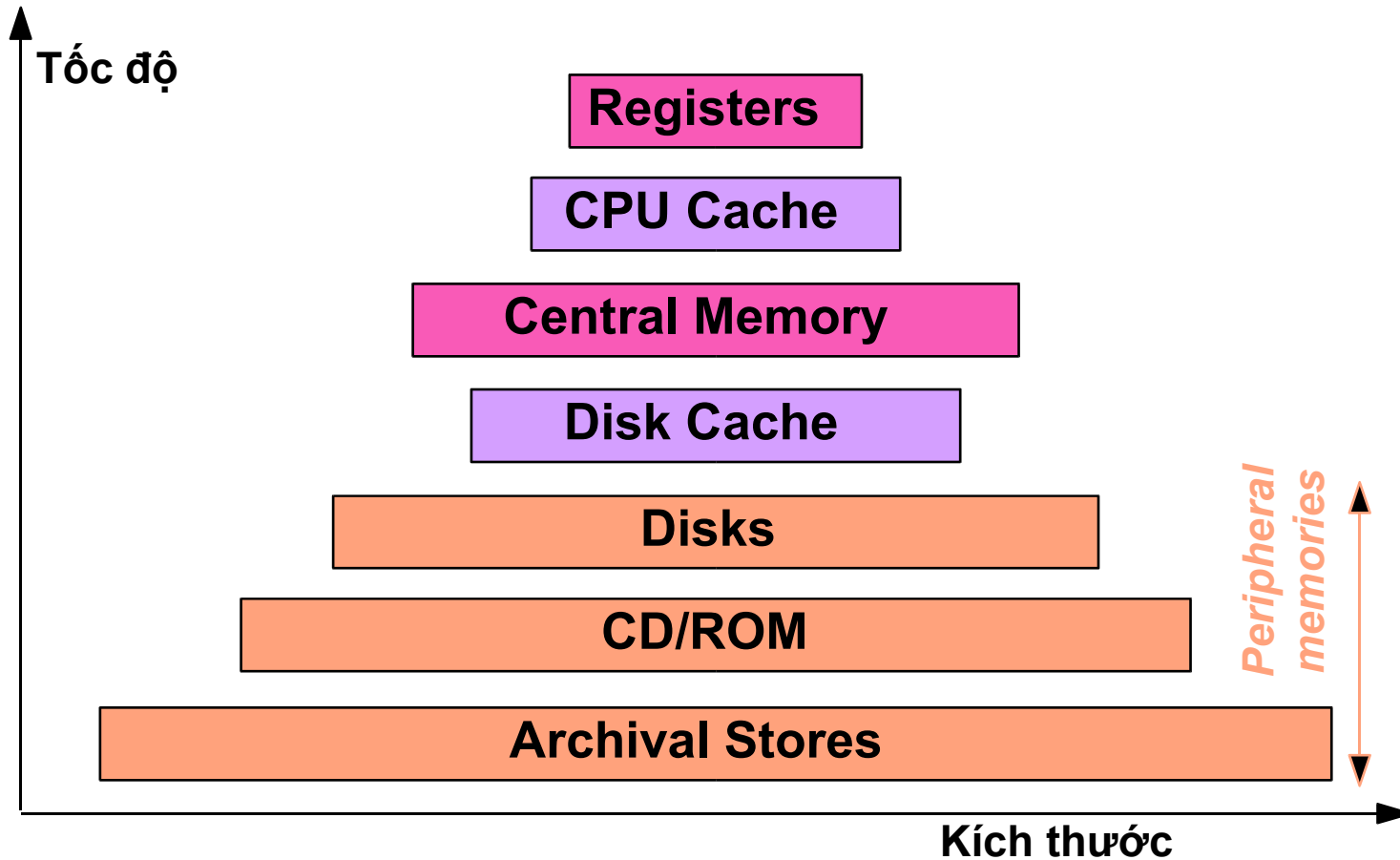
Kiểu bộ nhớ vật lý:

- Bộ nhớ bán dẫn
- Bộ nhớ từ
- Bộ nhớ quang

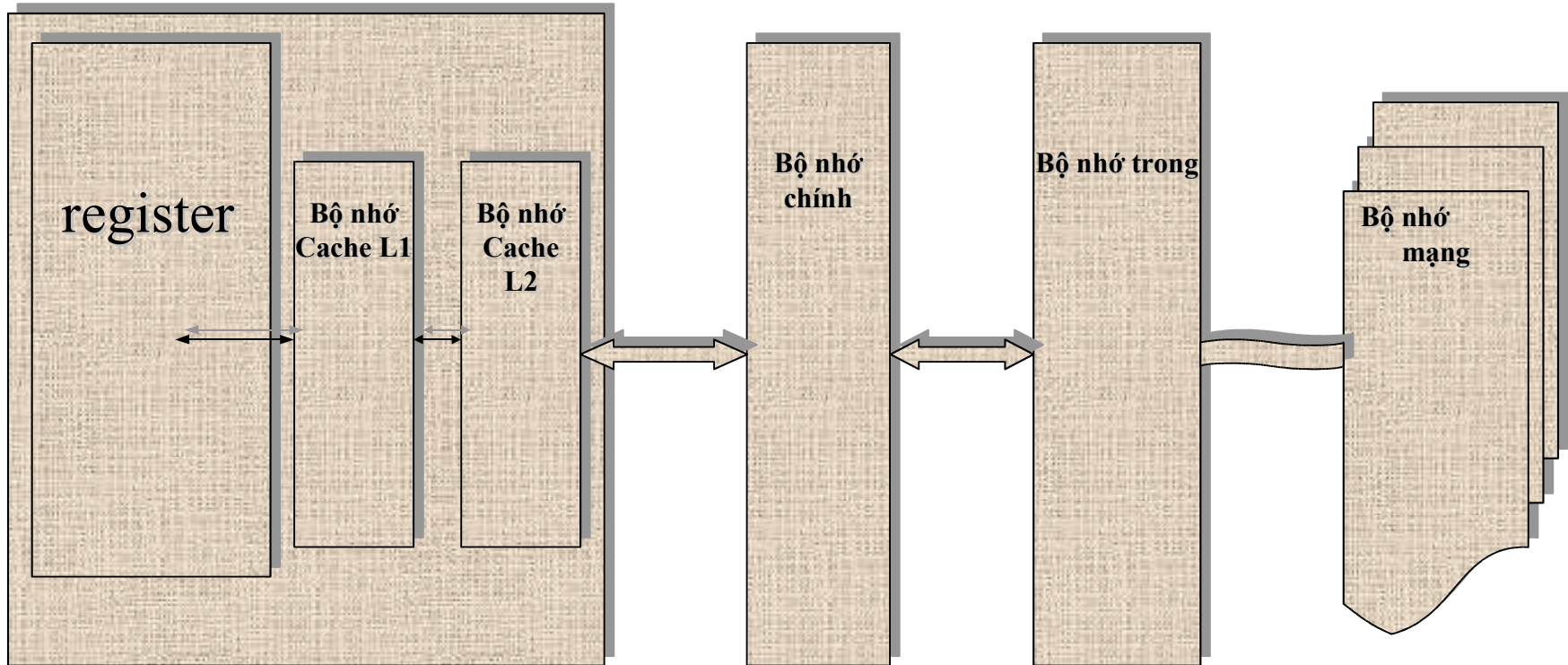
Các đặc tính vật lý:

- Khả biến/không khả biến
- Xoá được/không xoá được

Phân cấp bộ nhớ



Phân cấp bộ nhớ



Từ trái qua phải: dung lượng tăng dần, tốc độ giảm dần, giá thành tính theo đơn vị byte hoặc bit giảm dần.

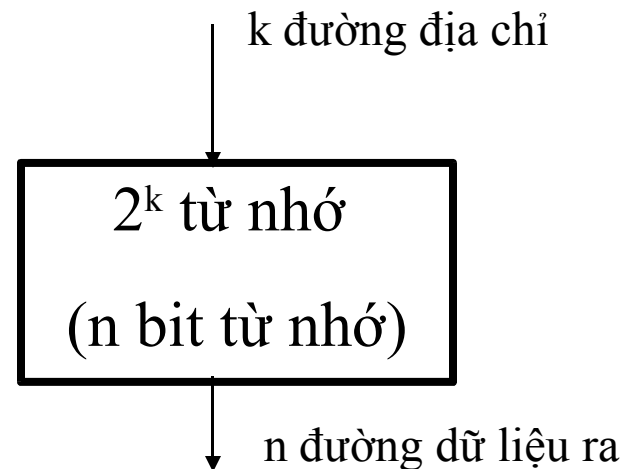
5.2 Bộ nhớ bán dẫn

Bộ nhớ chỉ đọc (ROM: Read Only Memory)

Bộ nhớ không khả biến

Sử dụng để lưu các thông tin sau:

- ❖ Thư viện các chương trình con.
- ❖ Các chương trình con điều khiển hệ thống (BIOS)
- ❖ Các bảng chức năng.



5.2 Bộ nhớ bán dẫn

Các kiểu ROM:

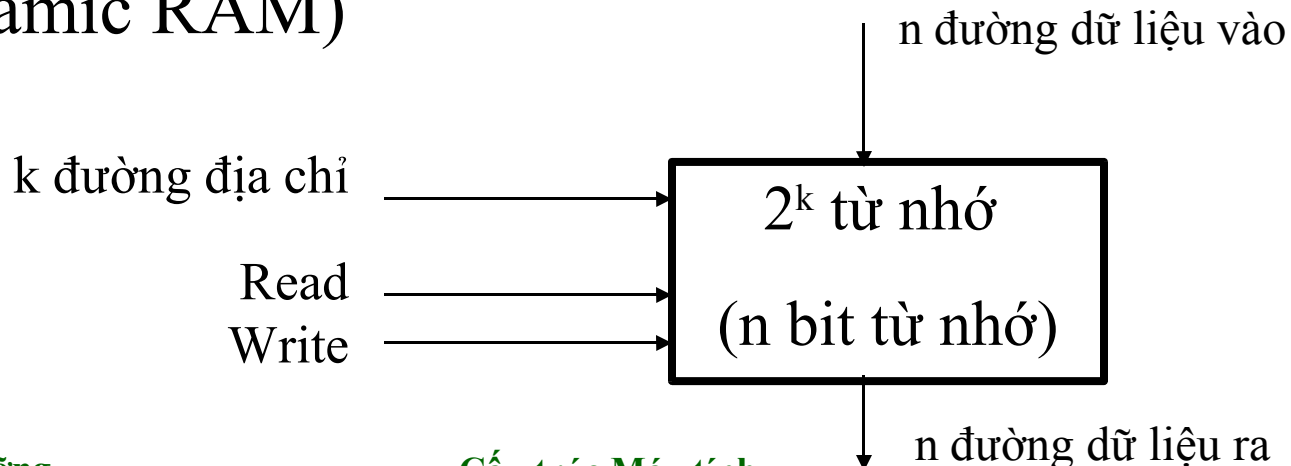
ROM mặt nạ, PROM: Programmable ROM, EPROM: Erasable PROM, EEPROM Electrically EPROM, Flash Memory (Bộ nhớ cực nhanh): Ghi theo khối, xoá bằng điện.

5.2 Bộ nhớ bán dẫn

Bộ nhớ truy cập ngẫu nhiên

(RAM : Random Access Memory)

- Bộ nhớ đọc ghi (R/W memory)
- Bộ nhớ khả biến
- Lưu thông tin tạm thời
- Có hai loại chính là SRAM (Static RAM) và DRAM (Dynamic RAM)



5.2 Bộ nhớ bán dẫn

RAM tĩnh (SRAM: Static RAM)

- Các bit được lưu dựa trên các Flip- Flop (4-8 FF lưu 1 bit)
- Thông tin lưu ổn định
- Cấu trúc phức tạp
- Dung lượng nhỏ(KB)
- Tốc độ nhanh (6-8 ns)
- Dùng làm cache
- Giá thành cao

5.2 Bộ nhớ bán dẫn

RAM động (DRAM: Dynamic RAM)

- Các bit được lưu dựa trên các tụ điện => nguyên nhân thường xuyên làm tươi.
- Dung lượng lớn.
- Tốc độ chậm (60-80ns).
- Dùng làm bộ nhớ chính
- Giá thành phải chăng.
- Các DRAM tiên tiến:

SDRAM: Synchronous Dynamic RAM, DDRAM: Double Data RAM. Ram BUS RDRAM.



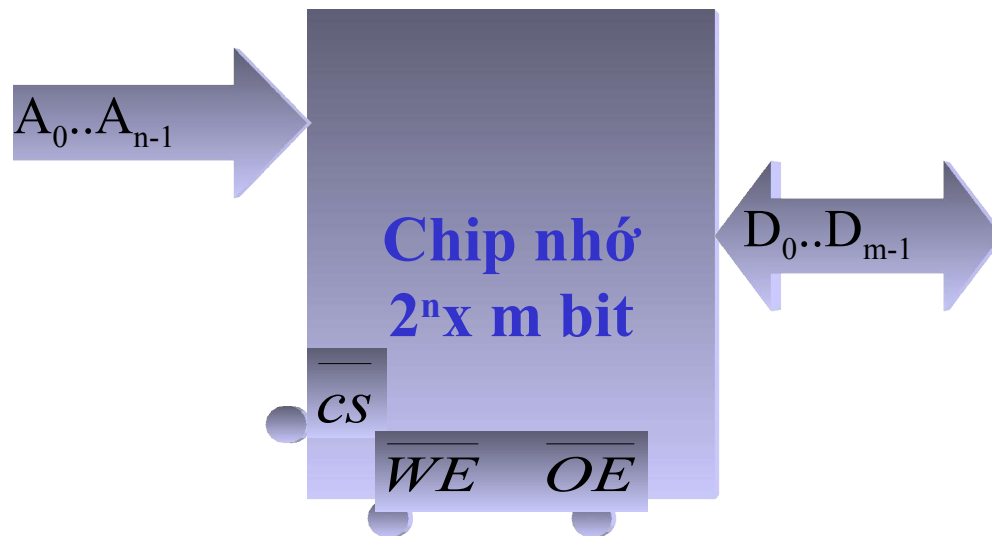
Bộ nhớ chính

Các đặc trưng cơ bản

- Tồn tại trên mọi hệ thống máy tính
- Chứa chương trình đang thực hiện và các dữ liệu có liên quan.
- Gồm các ngăn nhớ được đánh địa chỉ trực tiếp bởi CPU.
- Dung lượng bộ nhớ chính bao giờ nhỏ hơn không gian mà CPU có thể quản lý.
- Việc quản lý logic bộ nhớ phụ thuộc vào hệ điều hành.

Tổ chức của chip nhớ

- Sơ đồ cơ bản của chip nhớ



Tổ chức của chip nhớ

Các tín hiệu của chip nhớ

- ✓ Các đường địa chỉ: $A_0 \dots A_{n-1}$ để xác định 2^n ngăn nhớ.
- ✓ Các đường dữ liệu: $D_0 \dots D_{m-1}$ độ dài từ nhớ (m bit)
=> dung lượng chip nhớ = $2^n \times m$ bit
- ✓ Các tín hiệu điều khiển
 - Tín hiệu chọn chip hoạt động: CS (Chip Select)
 - Tín hiệu điều khiển đọc hoặc ghi (WE: Write Enable; OE: Output Enable)
 - Thường các tín hiệu điều khiển tích cực với mức 0

Thiết kế Module nhớ

Thiết kế module nhớ bán dẫn

- Cho chip nhớ $2^n \times m$ bit
- Yêu cầu sử dụng chip nhớ trên thiết kế module nhớ dung lượng là bội kích thước chip nhớ trên.

Giải quyết vấn đề

Có hai cách:

- Thiết kế để tăng độ dài từ nhớ, số ngăn nhớ không thay đổi.
- Thiết kế để tăng số lượng ngăn nhớ, độ dài từ nhớ không thay đổi.

Thiết kế Module nhớ

Thiết kế tăng số lượng từ nhớ

↪ **Giả thiết:** Cho các chip nhớ có dung lượng $2^n \times m$ bit.

↪ **Yêu cầu:** Thiết kế module nhớ có kích thước:

$2^n \times (k.m)$ bit

↪ **Giải quyết:**

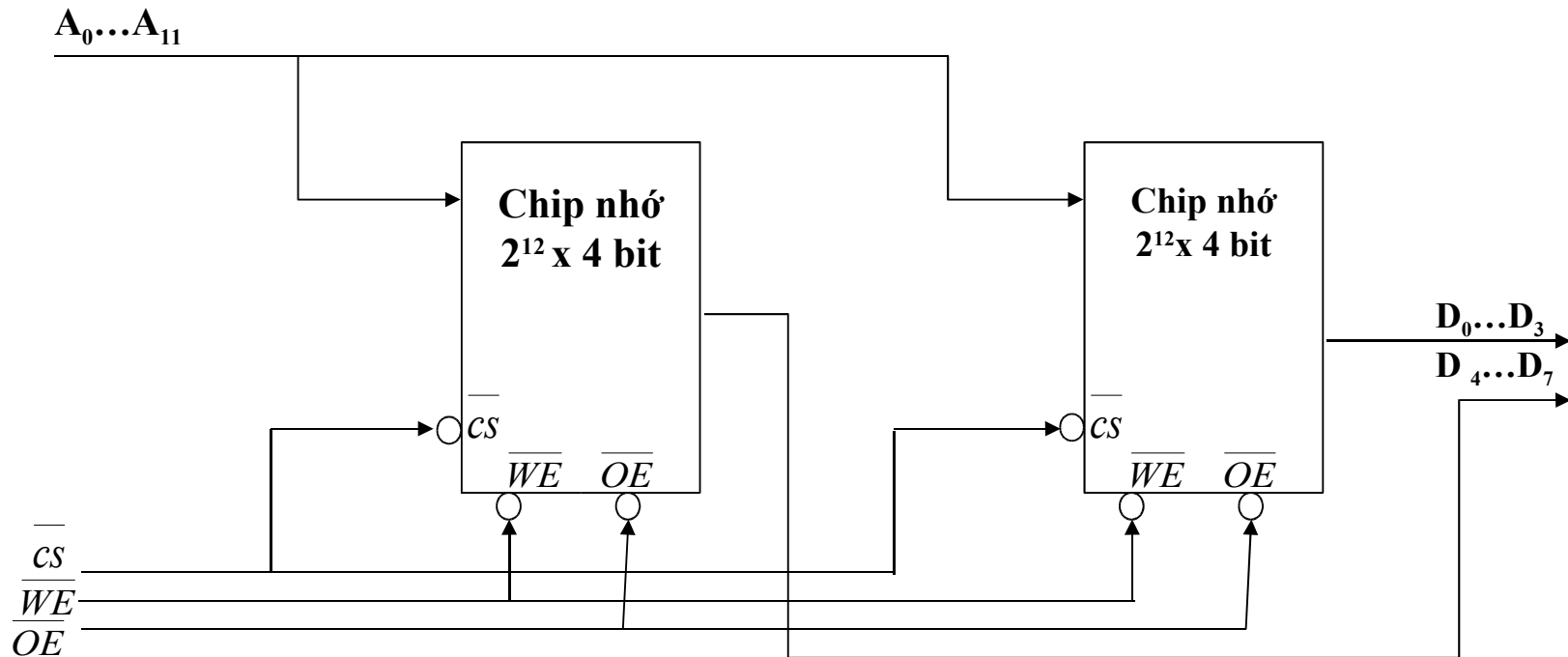
Để thiết kế được yêu cầu ta xác định hai thông số n (số đường địa chỉ) và k (số chip nhớ cần để ghép vào module thiết kế)

Thiết kế Module nhớ

Ví dụ: Cho các chip nhớ SDRAM dung lượng $4K \times 4$ bit.
Hãy thiết kế module nhớ có kích thước $4K \times 8$ bit

- Dung lượng chip nhớ $2^{12} \times 4$ bit
- Thông tin cần cho chip nhớ số đường địa chỉ $n = 12$ và số đường dữ liệu $m = 4$
- Thông tin về module nhớ số đường địa chỉ là 12 đường (số ngăn nhớ không thay đổi), số đường dữ liệu là 8 đường và số chip sử dụng thiết kế $2(k=2)$

Thiết kế Module nhớ



Thiết kế Module nhớ

Thiết kế tăng số lượng ngăn nhớ

↪ **Giả thiết:** Cho các chip nhớ có dung lượng $2^n \times m$ bit.

↪ **Yêu cầu:** Thiết kế module nhớ có kích thước:

$$2^k \cdot 2^n \times m \text{ bit}$$

↪ **Giải quyết:**

Để thiết kế được ta xác định hai thông số $n+k$ (số đường địa chỉ) và 2^k (số chip nhớ cần để ghép vào module thiết kế)

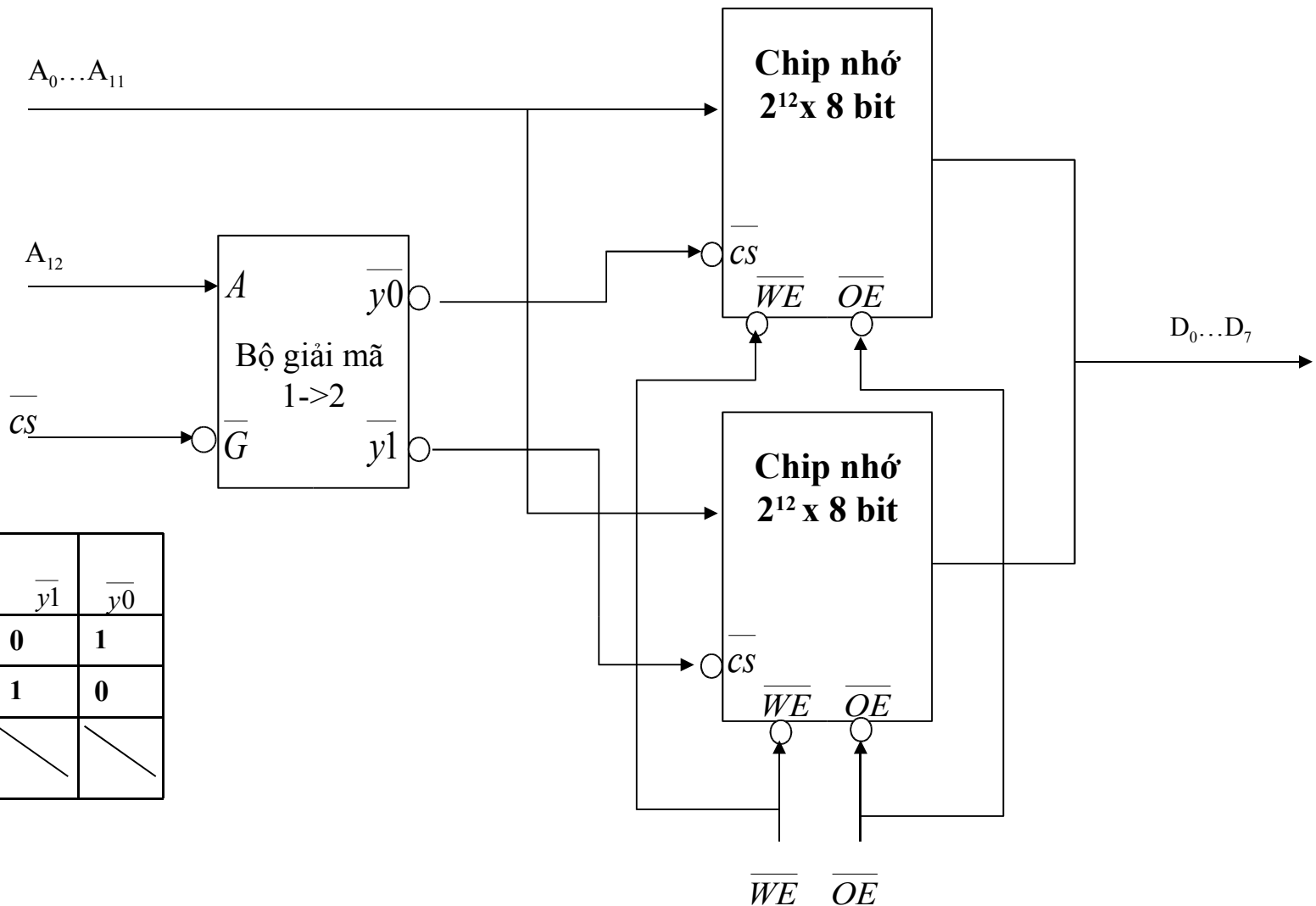
Thiết kế Module nhớ

Ví dụ : Cho các chip nhớ SDRAM dung lượng 4K x 8 bit. Hãy thiết kế module nhớ có kích thước 8K x 8 bit.

- Dung lượng chip nhớ giải thiết $2^{12} \times 8$ bit
- Thông tin cần cho chip nhớ số đường địa chỉ $n=12$ và số đường dữ liệu $m=8$
- Thông tin về module nhớ số đường địa chỉ là 13 đường (số ngăn nhớ thay đổi) và số đường dữ liệu là 8 đường (độ dài từ nhớ không đổi).

Thiết kế Module nhớ

\overline{G}	A	$\overline{y1}$	$\overline{y0}$
0	0	0	1
0	1	1	0
1	x	/	/





Bài làm thêm

- Thiết kế module nhớ 16K x 8 bit từ các chip nhớ 4K x 8 bit
- Thiết kế module nhớ 32K x 8 bit từ các chip nhớ 4K x 8 bit
- Thiết kế module nhớ 8K x 8 bit từ các chip nhớ 4K x 4 bit
- Thiết kế module nhớ 32M x 32 bit từ các chip nhớ 4M x 32 bit

Phát hiện và chỉnh lỗi trong bộ nhớ

Phát hiện và chỉnh lỗi trong bộ nhớ

Nguyên tắc chung: Trong quá trình truyền dữ liệu có thể gặp sự thay đổi các bit thông tin do nhiễu hoặc do sai hỏng của thiết bị hay module vào ra. Vì vậy, thực tế đặt ra là phải làm sao phát hiện được lỗi và có thể sửa sai được. Một trong phương pháp phát hiện lỗi (EDC: Error Detecting Code) và sửa lỗi (ECC: Error Correcting Code) là: Giả sử cần kiểm tra m bit thì người ta ghép thêm k bit kiểm tra được mã hoá theo cách nào đó rồi truyền từ ghép $m+k$ bit (k bit được truyền không mang thông tin nên gọi là bit dư thừa)

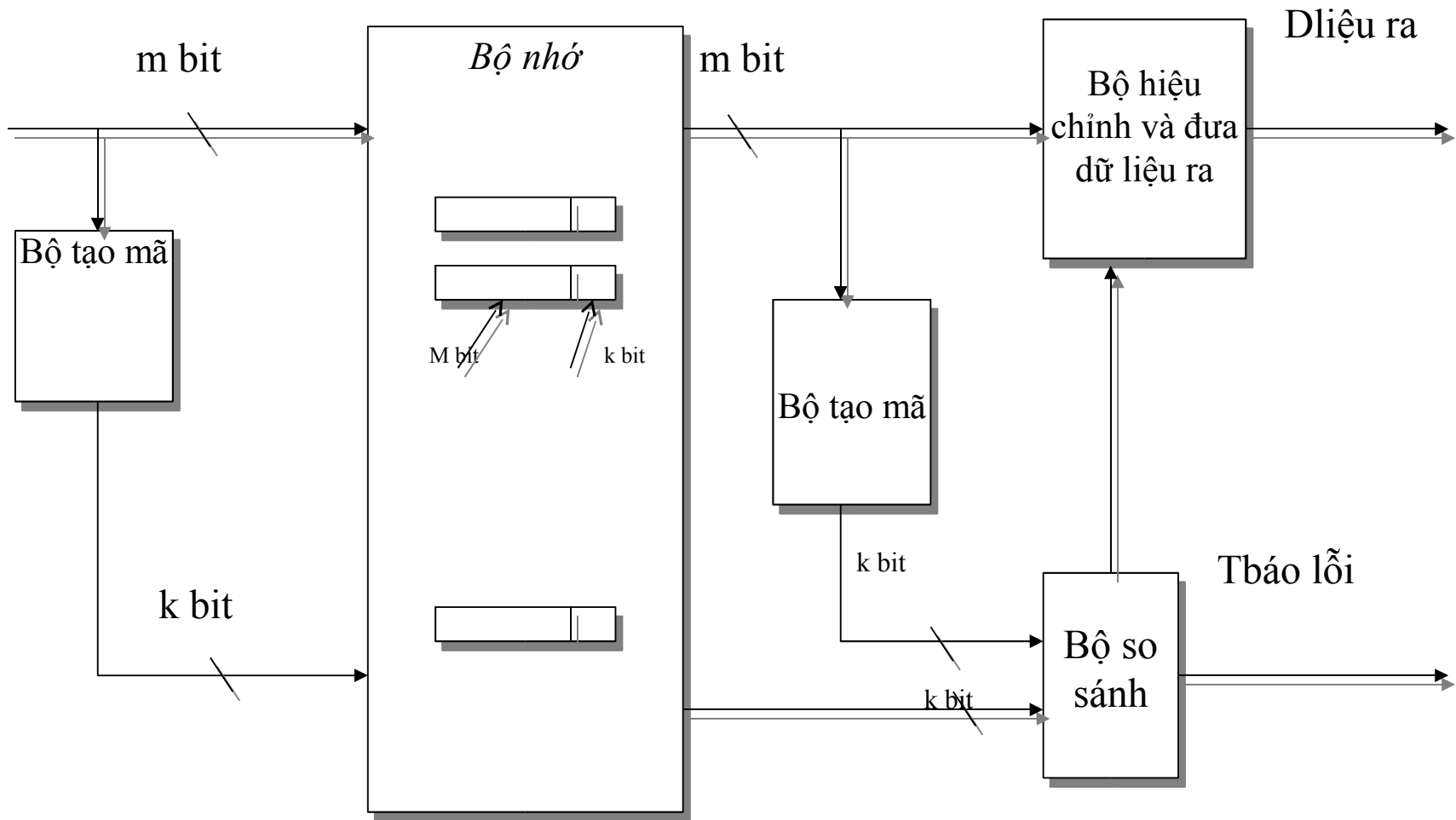
Trong đó m là số bit cần ghi vào bộ nhớ và k bit là số bit cần tạo ra kiểm tra lỗi trong m bit.

Phát hiện và chỉnh lỗi trong bộ nhớ

Khi đọc dữ liệu ra có khả năng sau:

- Không phát hiện dữ liệu có lỗi.
- Phát hiện thấy dữ liệu lỗi và có thể hiệu chỉnh dữ liệu lỗi thành đúng.
- Phát hiện thấy lỗi nhưng không có khả năng chỉ ra lỗi vì thế phát ra tín hiệu báo lỗi.
- Sơ đồ phát hiện lỗi và sửa lỗi

Phát hiện và chỉnh lỗi trong bộ nhớ



Phát hiện và chỉnh lỗi trong bộ nhớ

Ví dụ 1: Phát hiện lỗi với bit chẵn lẻ (Parity)

Mã EDC đơn giản là bit chẵn lẻ được gắn thêm vào các bit dữ liệu.

Nếu bit chẵn lẻ = 1: nếu số bit 1 trong chuỗi là lẻ

Hoặc sử dụng Nếu bit chẵn lẻ = 0: nếu số bit 1 là chẵn

Ưu điểm: đơn giản và số bit dư thừa ít.

Nhược điểm: không định vị được lỗi, hoặc nếu có sự thay đổi cả hai bit hoặc 1 hoặc 0 thì không phát hiện được. Khắc phục nhược điểm trên xây dựng mã EDC khối.

Phát hiện và chỉnh lỗi trong bộ nhớ

Ví dụ 2: Phát hiện lỗi bằng mã dư thừa CRC (Cycle Redundary Check).

Nguyên tắc: Một xâu nhị phân bất kỳ có thể coi là tập hợp các hệ số của đa thức $B(x)$ trong đó x là *hư số*. Chọn đa thức $G(x)$ là đa nào đó ta quy định trước gọi *đa thức sinh*. Ta tiến hành chia module2 đa thức $B(x)$ cho $G(x)$ ta được thương số $Q(x)$ và phần dư $R(x)$.

- ⇒ Đa thức sinh do tổ chức viên thông quốc tế quy định.
- ⇒ Khi đó ta cần truyền xâu $B(x) + R(x)$ bit
- ⇒ Để kiểm tra lỗi ta cần chia giá trị nhận được cho đa thức sinh nếu phép chia có dư thì có lỗi xuất hiện trong xâu.

Phát hiện và chỉnh lỗi trong bộ nhớ

- Ví dụ:
- Xâu gốc: $1101011011 \Leftrightarrow M(x) = x^9 + x^8 + x^6 + x^4 + x^3 + x + 1 (m=9)$
- Đa thức sinh $G(x) = x^4 + x + 1 \Leftrightarrow 10011 (r=4)$
- Xâu gốc: $11010110110000 \Leftrightarrow x^4 M(x)$
- Chia mod2 $11010110110000 \begin{array}{l} | 10011 \\ \hline 1100001010 \end{array} \rightarrow \text{thương}$
↓
1110 phần dư phép chia
- Xâu cần truyền đi: $11010110111110 \Leftrightarrow T(x)$

Phát hiện và chỉnh lỗi trong bộ nhớ

Ví dụ 3: Mã sửa lỗi Hamming

Nguyên tắc: Một từ mã Hamming gồm m bit dữ liệu và k bit kiểm tra chẵn lẻ. Mỗi bit được chọn vị trí thích hợp để phát hiện chính xác vị trí để có thể sửa lỗi được. Ví dụ chọn $m=4 \Rightarrow k=3$ ($m=2^n$; $k=n+1$)

Ta có thứ tự sau:

7	6	5	4	3	2	1
I4	I3	I2	C3	I1	C2	C1

Các bit này được mã hoá theo quy luật sau:

$$C1 = I1 \oplus I2 \oplus I4$$

$$C2 = I1 \oplus I3 \oplus I4$$

$$C3 = I2 \oplus I3 \oplus I4$$

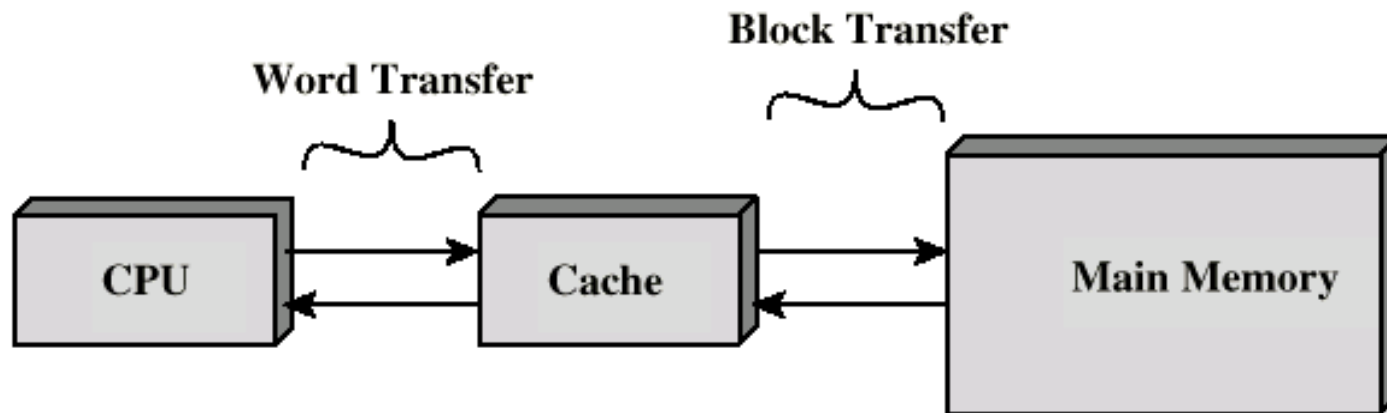
Phát hiện và chỉnh lỗi trong bộ nhớ

- Giả sử các bit cần truyền là: $I_4 I_3 I_2 I_1 = 1101$ tính các $C_3 C_2 C_1 = 010$
- Bit cần truyền 1100110
- Giả sử ta có bị lỗi, thí dụ bit I_2 từ giá trị 0 thành giá trị 1 mã nhận được 1110110 .
- Bên thu tính ra bit kiểm tra:
$$C_3 = 1 \oplus 1 \oplus 1 = 1$$
$$C_2 = 1 \oplus 1 \oplus 1 = 1$$
$$C_1 = 1 \oplus 1 \oplus 1 = 1$$
- Nếu module 2 số này ta được $111 \oplus 010 = 101$
(C_1, C_3 thay đổi và vị trí thay đổi là 101 (5))

5.3 Bộ nhớ đệm nhanh

Nguyên tắc:

- Cache có tốc độ truy xuất nhanh hơn rất nhiều bộ nhớ chính
- Cache được đặt giữa CPU và bộ nhớ chính nhằm tăng tốc độ trao đổi thông tin giữa CPU và bộ nhớ chính.
- Cache thường được đặt trong chip vi xử lý

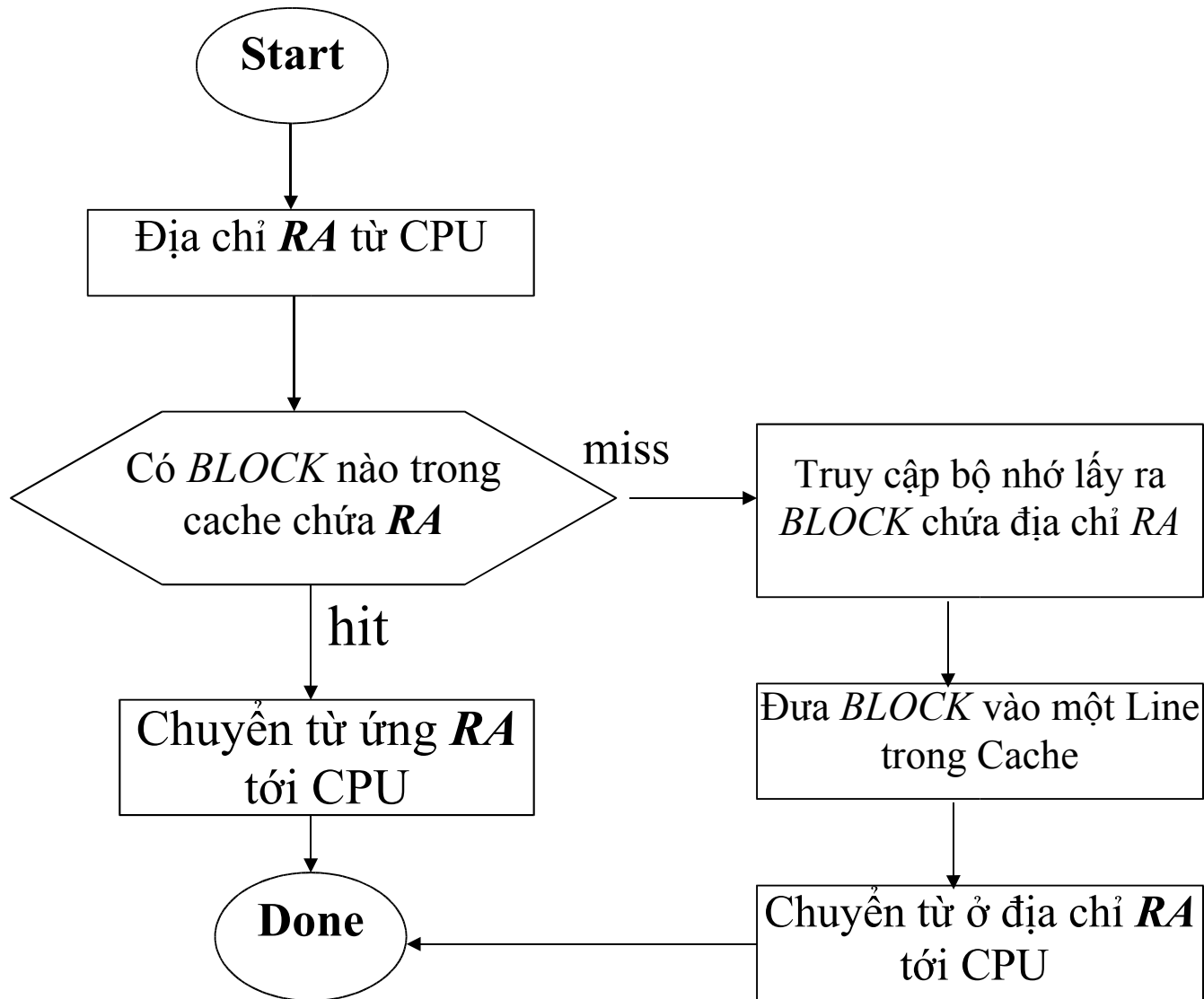


5.4 Bộ nhớ đệm nhanh

Thao tác của Cache

- CPU yêu cầu lấy nội dung của một ngăn nhớ bằng việc đưa ra một địa chỉ xác định ô nhớ.
- CPU kiểm tra xem có nội dung cần tìm trong Cache
- Nếu có: CPU nhận dữ liệu từ bộ nhớ Cache
- Nếu không có: Bộ điều khiển Cache đọc Block nhớ chứa dữ liệu CPU cần vào Cache.
- Tiếp đó chuyển dữ liệu từ Cache đến CPU
- Sơ đồ thao tác cache, bộ nhớ chính và CPU

5.4 Bộ nhớ đệm nhanh



5.4 Bộ nhớ đệm nhanh

CPU

Bộ nhớ Cache

Tag	Line 1
	Line 2
	Line 3
	...
	Line C

Bộ nhớ chính

Block 1
Block 2
Block 3
Block 4
...
Block M-2
Block M-1
Block M

5.3 Bộ nhớ đệm nhanh

- Tổ chức Cache
- ❖ Giả sử bộ nhớ chính gồm có 2^n từ nhớ đã được đánh địa chỉ (mỗi từ nhớ có địa chỉ duy nhất rộng n bit)
- ❖ Bộ nhớ chính chia thành M khối, mỗi khối có K từ nhớ
 $M=2^n/K$
- ❖ Bộ nhớ Cache có C khe mỗi khe có K từ nhớ. ($C \ll M$)
- ❖ Tại một thời điểm luôn có một tập con các khối nhớ thường trú trong cache.
- ❖ Nếu một từ sẽ được đọc thì khối chứa từ đó sẽ được chuyển vào trong cache.

5.3 Bộ nhớ đệm nhanh

Ví dụ cho phương pháp ánh xạ cụ thể trong cache

- Cho dung lượng Cache là 64KB ($m=16$)

Mỗi khối kích thước 4 bytes

$\Rightarrow C=16K(2^{14})$ lines mỗi line kích thước 4 bytes

- Cho dung lượng bộ nhớ chính 16MB ($n=24$)

Mỗi khối kích thước 4 bytes

$\Rightarrow M=4M(2^{22})$ khối mỗi khối kích thước 4 bytes

5.3 Bộ nhớ đệm nhanh

Phương pháp ánh xạ trực tiếp (Direct mapping)

- Mỗi block được ánh xạ duy nhất tới 1 line trong cache
- Địa chỉ phát ra từ CPU được chia 2 phần
- w bits có trọng số thấp để xác định duy nhất từ cần truy xuất(WORD)
- s bits còn lại xác định khối nhớ. Trong s bits chia 2 nhóm r bits LINE và $s-r$ bits TAG

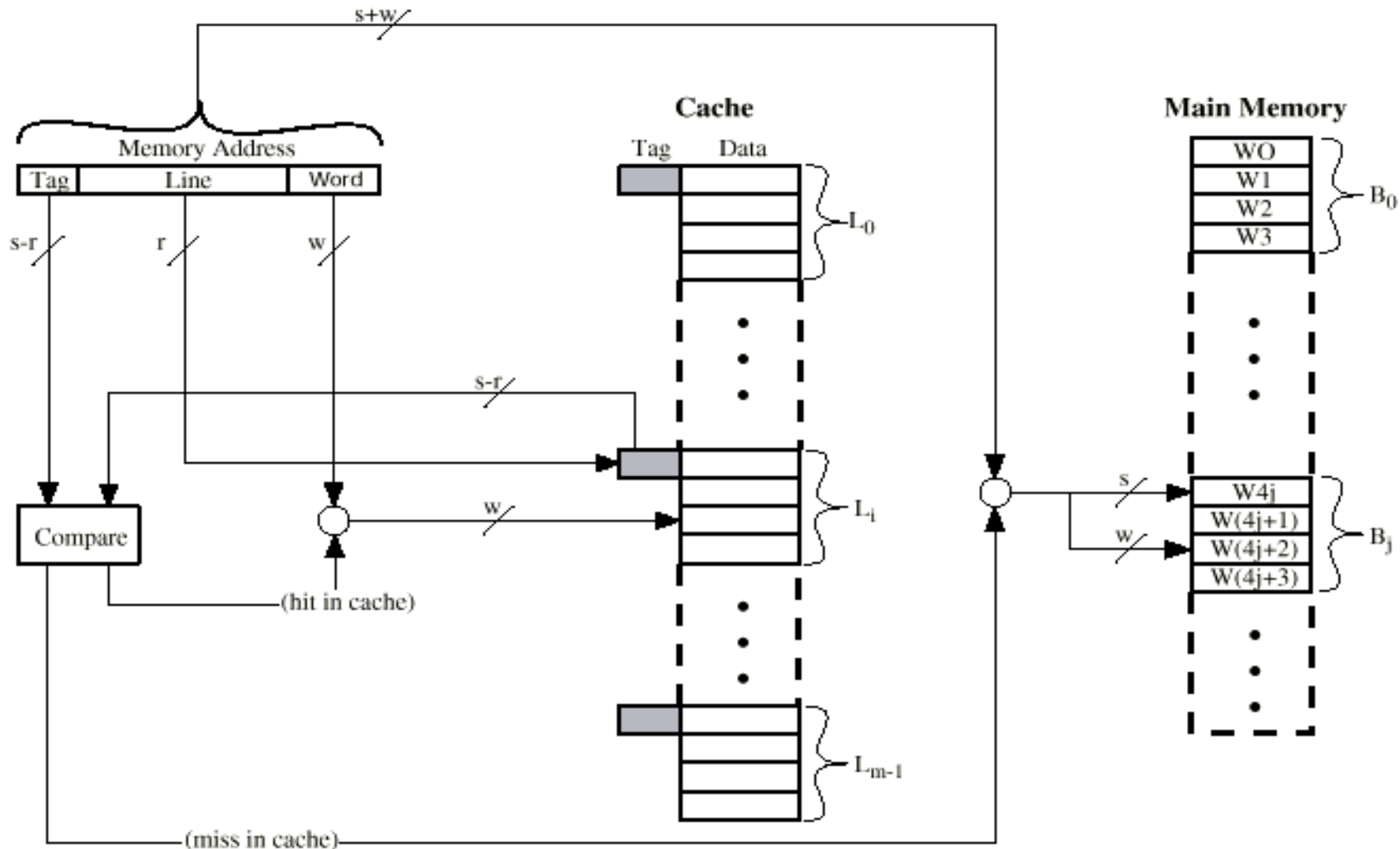
Cụ thể hóa ví dụ:

Tag $s-r$	Line or Slot r	Word w
8	14	2

5.3 Bộ nhớ đệm nhanh

- Tổng bit trong địa chỉ bộ nhớ chính $n=24$ bit: trong đó
- 2 bit phần word xác định chính xác 4 từ
 - 22 bit xác định khối(8 bit tag ($=22-14$) và 14 bit slot or line)
- Không có hai block nào trong Cache có cùng Line và Tag.
 - Kiểm tra nội dung từ tồn tại Cache chính là kiểm tra địa chỉ line và Tag

5.3 Bộ nhớ đệm nhanh



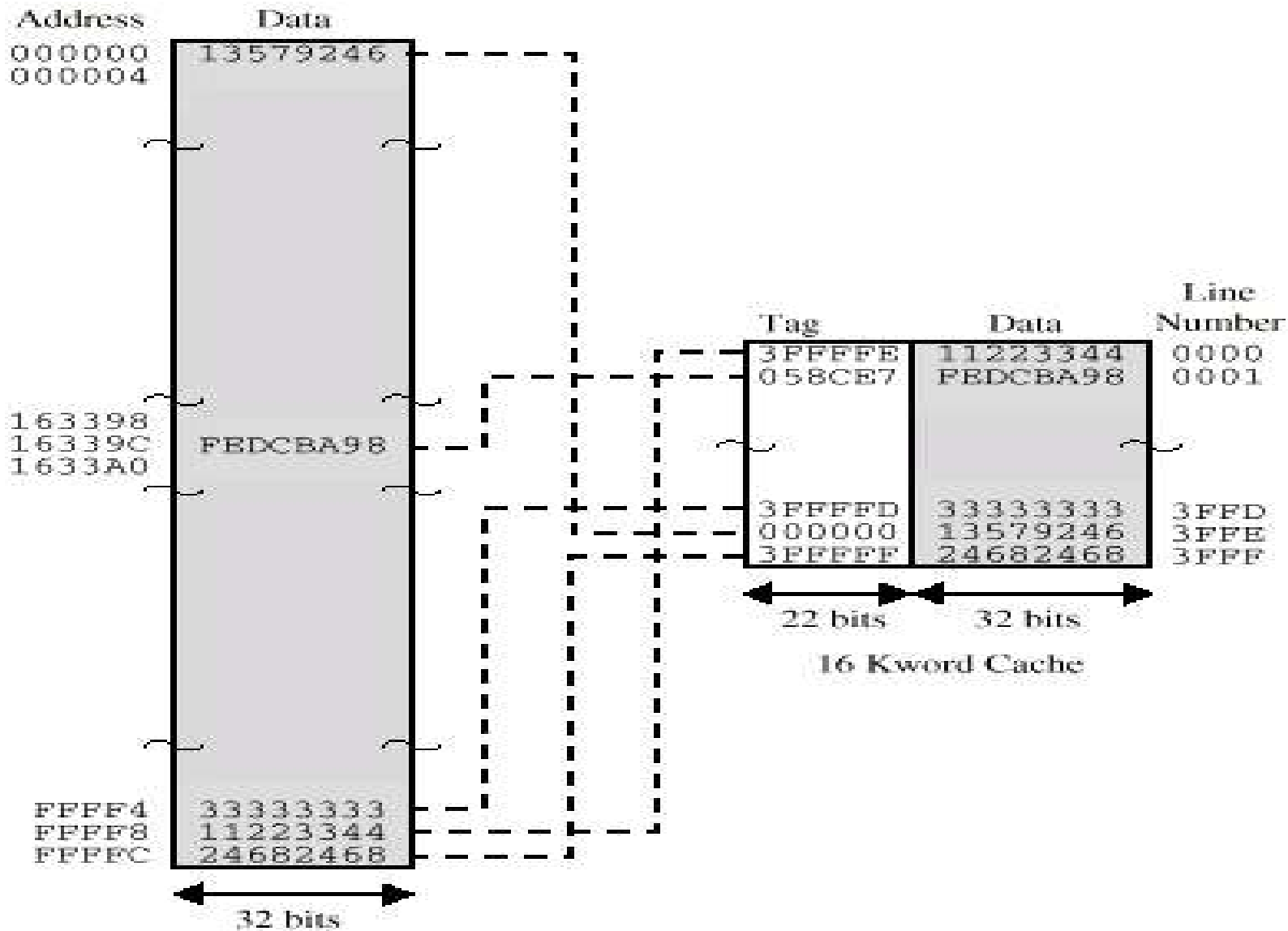
5.3 Bộ nhớ đệm nhanh

Cache line	Main Memory blocks
0	0, C, 2C, 3C... 2^s-C
1	1, C+1, 2C+1... 2^s-C+1
C-1	C-1, 2C-1, 3C-1... 2^s-1

Nhận xét:

- Đơn giản
- Chi phí ít
- Nhược điểm là sự cố định các khối trong các line của Cache. Trong trường hợp chương trình muốn truy xuất tới 2 Block liên tục mà 2 block được phân nằm trong cùng line thì khả năng Cache miss rất cao.

5.3 Bộ nhớ đệm nhanh



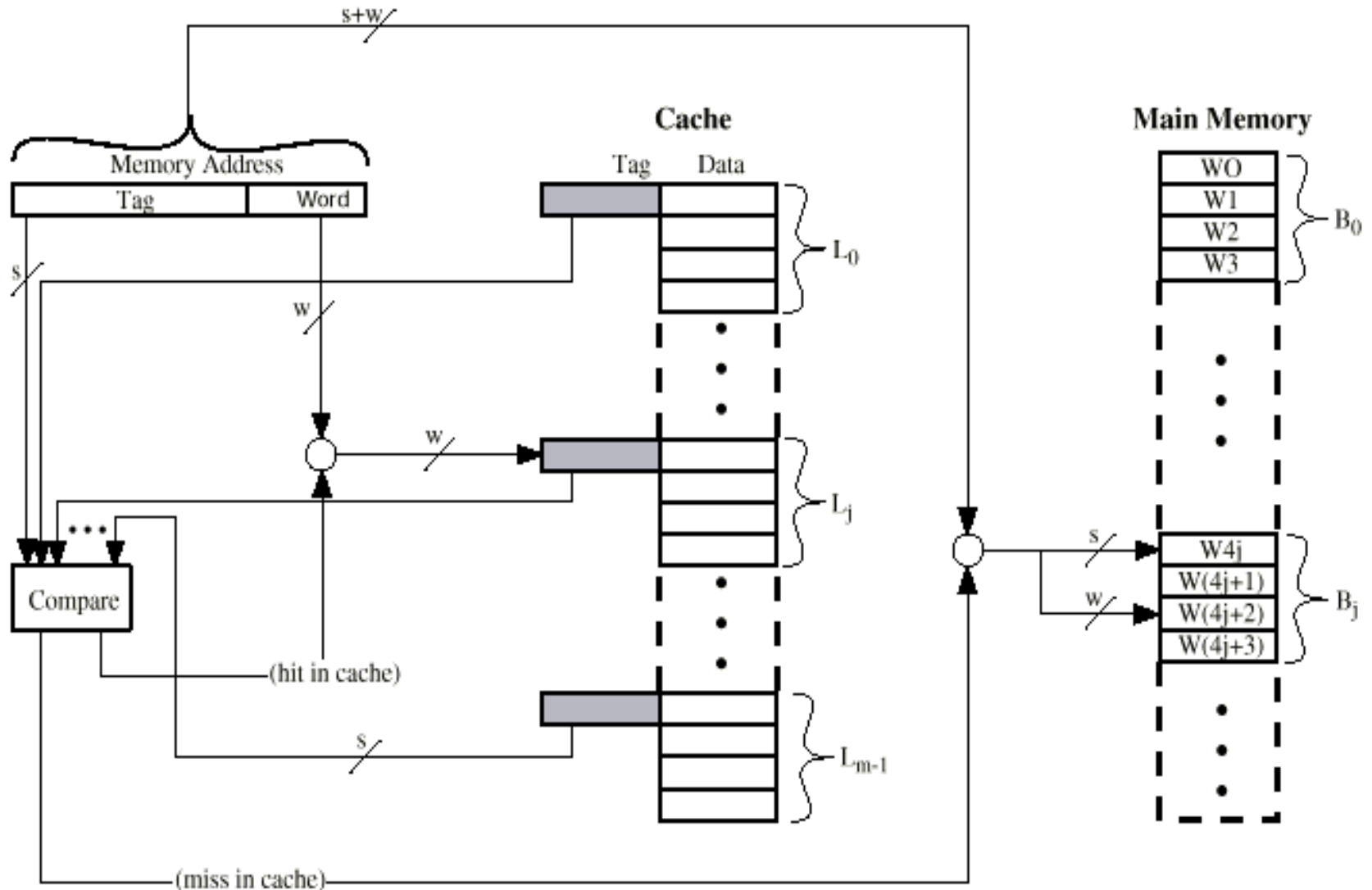
5.3 Bộ nhớ đệm nhanh

Phương pháp ánh xạ liên kết

(Associative mapping)

- Một Block của bộ nhớ chính có thể nhập bất kỳ line nào trong Cache.
- Địa chỉ CPU phát ra được chia thành 2 địa chỉ tag và word
- Địa chỉ Tag xác định khối duy nhất của bộ nhớ nằm trong Cache.
- Mỗi giá trị Tag của Line là khác nhau.
- Chi phí phương pháp này đối với Cache là cao.

5.3 Bộ nhớ đệm nhanh



5.3 Bộ nhớ đệm nhanh

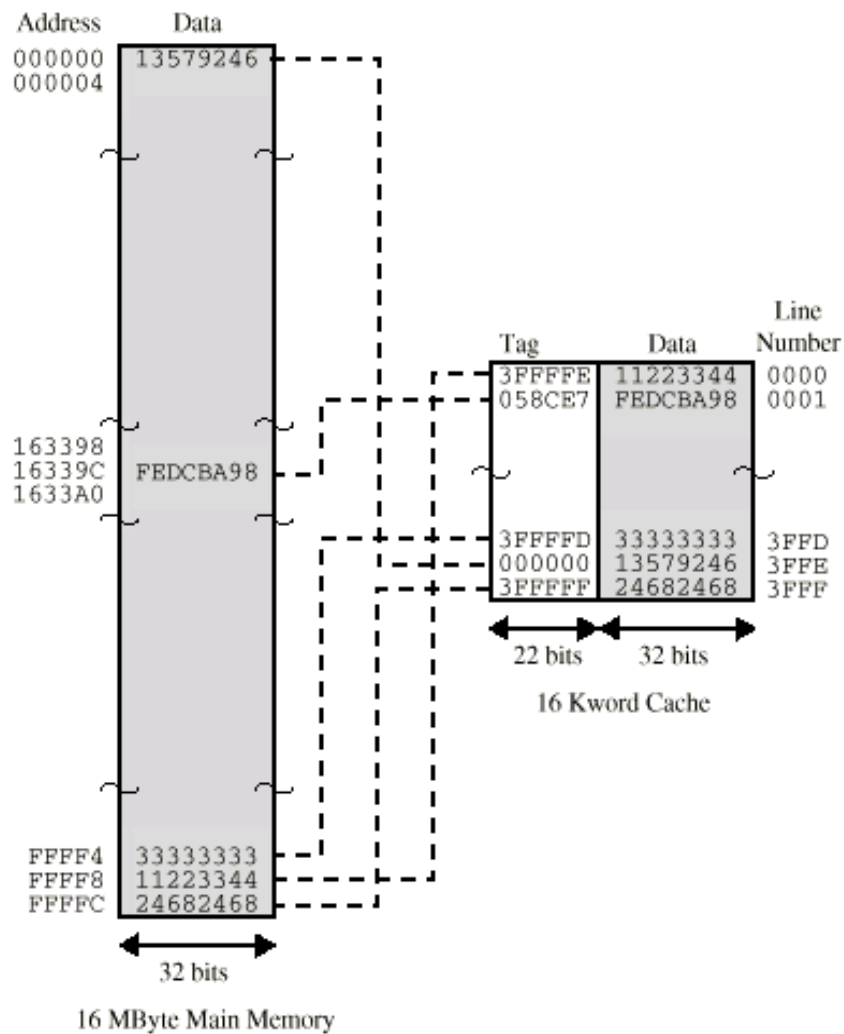
Tag 22 bit

Word
2 bit

- 22 bit Tag để lưu trữ Block 4 byte dữ liệu.
- Việc kiểm tra Cache dựa vào các giá trị Tag trong line (22 bit) để nhận biết Cache hit hay miss.
- 2 bits cuối xác định chính xác từ cần truy xuất
- Ví dụ

Địa chỉ	Tag	Dữ liệu	Cache line
FFFFFFC	FFFFFFC	24682468	3FFF

5.3 Bộ nhớ đệm nhanh

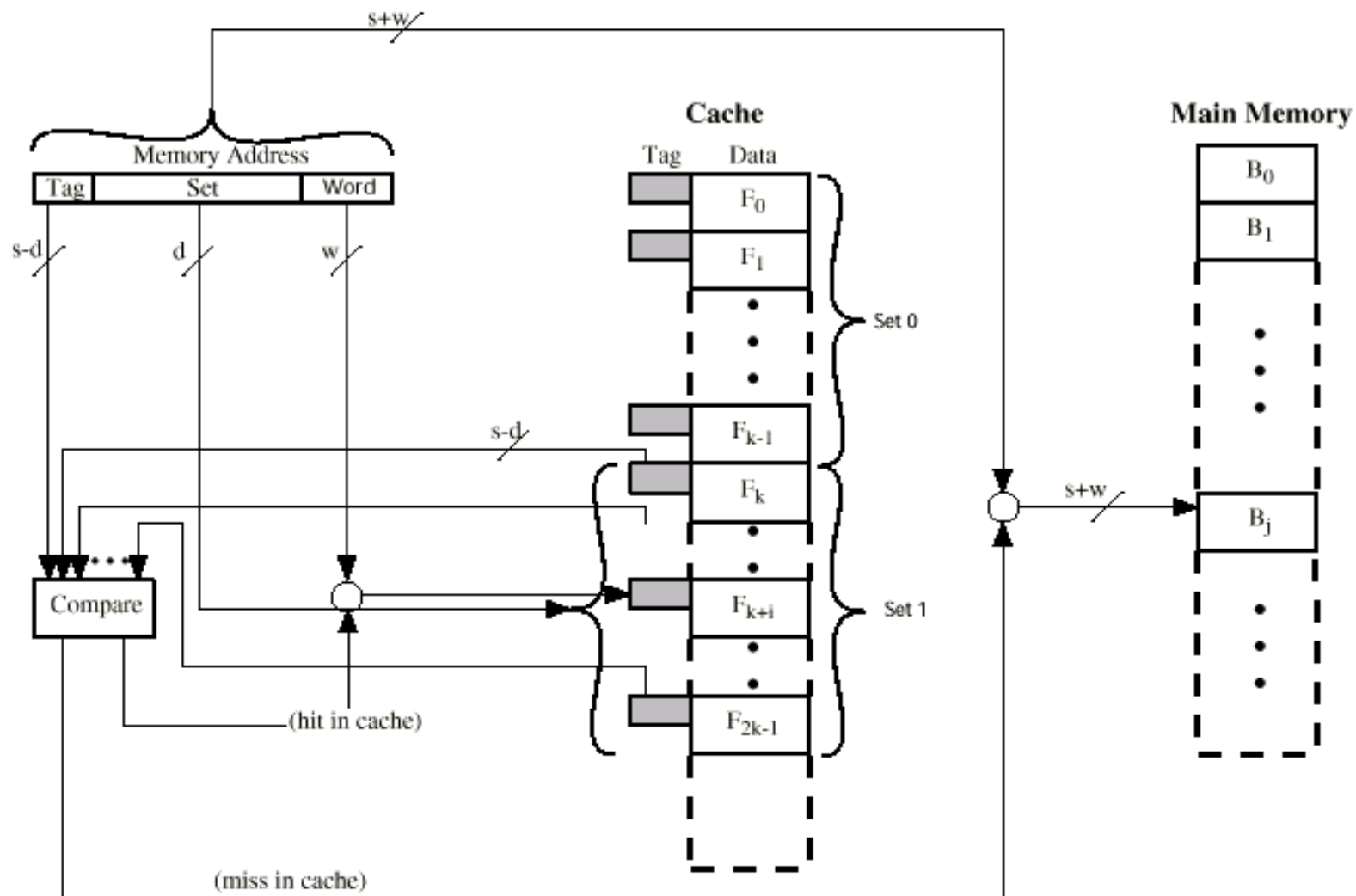


5.3 Bộ nhớ đệm nhanh

Phương pháp ánh xạ liên kết tập hợp (Set Associative mapping)

- Các line trong Cache được chia ra thành tập (nhóm) line
- Mỗi block chỉ được ánh xạ vào bất kỳ line nào trong tập nào đó mà thôi. Ví dụ Block b chỉ có thể nạp vào bất kỳ line nào trong nhóm các line thứ i. Ví dụ 2 lines một nhóm (two way associative mapping), Số Block bộ nhớ chính là modulo 2^{13}
- 000000, 00A000, 00B000, 00C000 ... ánh xạ cùng nhóm.

5.3 Bộ nhớ đệm nhanh



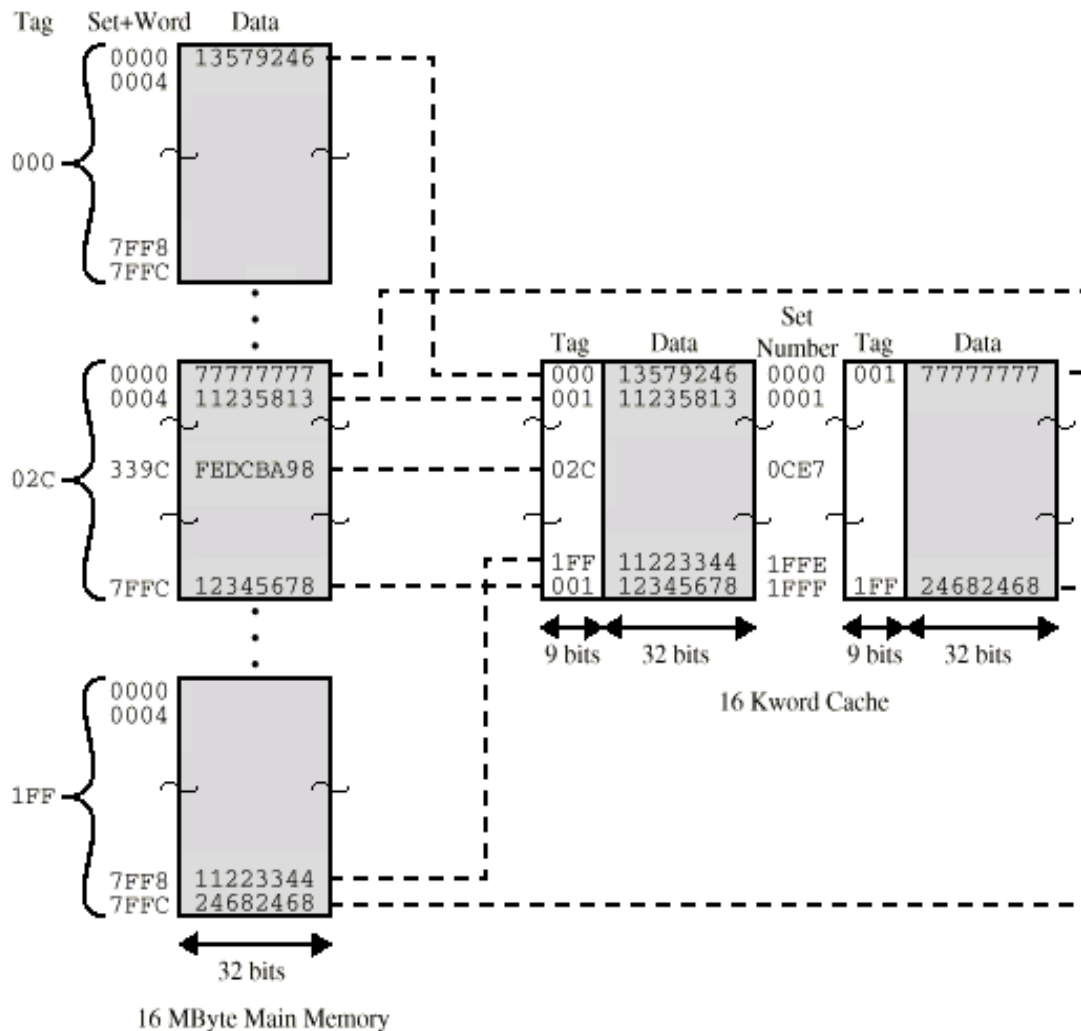
5.3 Bộ nhớ đệm nhanh

Tag 9 bit	Set 13 bit	Word 2 bit
------------------	-------------------	-------------------

- Sử dụng tập hợp để biết tập nào được truy xuất.
- So sánh trường Tag để xác định Cache hit hay miss
- Ví dụ:

Địa chỉ	Tag	Dữ liệu	số tập
1FF 7FFC	1FF	12345678	1FFF
001 7FFC	001	11223344	1FFF

5.3 Bộ nhớ đệm nhanh



5.4 Bộ nhớ đệm nhanh

Một số Block của bộ nhớ chính được nạp vào trong các line của Cache

- Nội dung thẻ *TAG* (thẻ nhớ) cho biết block nào của bộ nhớ chính hiện đang được chứa trong line
- Khi CPU truy nhập đọc hay ghi một từ nhớ của bộ nhớ chính, có 2 khả năng xảy ra :
- Từ nhớ đó có trong Cache (cache hit). Từ nhớ đó đang không có trong cache (Cache miss).

Phương pháp ghi dữ liệu khi cache hit

- Ghi xuyên qua (Write Through): nội dung sau khi xử lý xong được cập nhập vào cả Cache và bộ nhớ chính. Tốc độ chậm.

Cache trong các bộ xử lý Intel

- Ghi sau (Write back): Dữ liệu xử lý chỉ được ghi ra Cache, tốc độ nhanh. Tuy nhiên khi Block trong cache không dùng nữa thì phải ghi trả cả block tới bộ nhớ chính.

Dung lượng Cache được sử dụng cho thế hệ máy:

- 80486: có 3KB nhớ
- Pentium : có 2 cache L1 trên chip đó là Cache lệnh và cache dữ liệu (8KB). Cache L2 liên hợp
- Pentium 4: hai mức Cache L1 và L2 trên chip. Cache L1 mỗi cache 8KB. Cache L2: mỗi cache 256KB, 512KB, 1GB



5.5 Bộ nhớ ngoài

- Các kiểu bộ nhớ ngoài
- Đĩa từ
- Đĩa quang
- Bộ nhớ Flash
- RAID

Đĩa cứng (HDD: Hard Disk Driver)

- Là thành phần quan trọng lưu trữ hệ điều hành và các phần mềm tiện ích máy tính
- Một máy tính có thể một đĩa hoặc nhiều đĩa
- Dung lượng mỗi đĩa rất lớn. Năm 1993 đĩa lớn nhất 200MB đến nay 80 hay 120GB
- Tốc độ đọc ghi nhanh so các bộ nhớ ngoài khác
- Giá thành hạ
- Được sử dụng làm bộ nhớ RAID (Redundant Array of (Inexpensive) Independent Disks). Hệ thống nhớ gồm nhiều ổ đĩa cứng kết hợp với nhau mà HĐH coi như một ổ logic duy nhất.
- Dữ liệu được lưu trữ phân tán trên tất cả các đĩa
- Có thể tạo và lưu trữ thông tin dự thừa nhằm mục đích cho việc phục hồi khi đĩa nào đó bị hỏng. Độ tin cậy trong lưu trữ thông tin rất cao. Được sử dụng là bộ nhớ cho các hệ thống máy chủ.



Đĩa quang (CD-ROM, DVD)

- **CD-ROM (Compact Disk ROM)**
- **CD-R (Recordable CD)**
- **CD-RW (Rewriteable CD)**
- **Dung lượng phổ biến 650MB**
- **Ổ đĩa CD**
- **ổ CD ROM: có thể đọc dữ liệu từ đĩa CD**
- **ổ CD RW : Có thể vừa ổ đọc đĩa CD và có thể ghi dữ liệu lên đĩa CD-R, và CD-RW.**
- **Tốc độ đọc cơ sở 150KB/s**
- **Tốc độ bội lần : 40x, 50x, 60x,...**
- **DVD(Digital Video Disk): chỉ dùng trên đầu đọc**
- **DVD (Digital Versatile Disk): dùng trên ổ đĩa máy tính**
- **Dung lượng thông dụng 4.7GB**



Flash disk

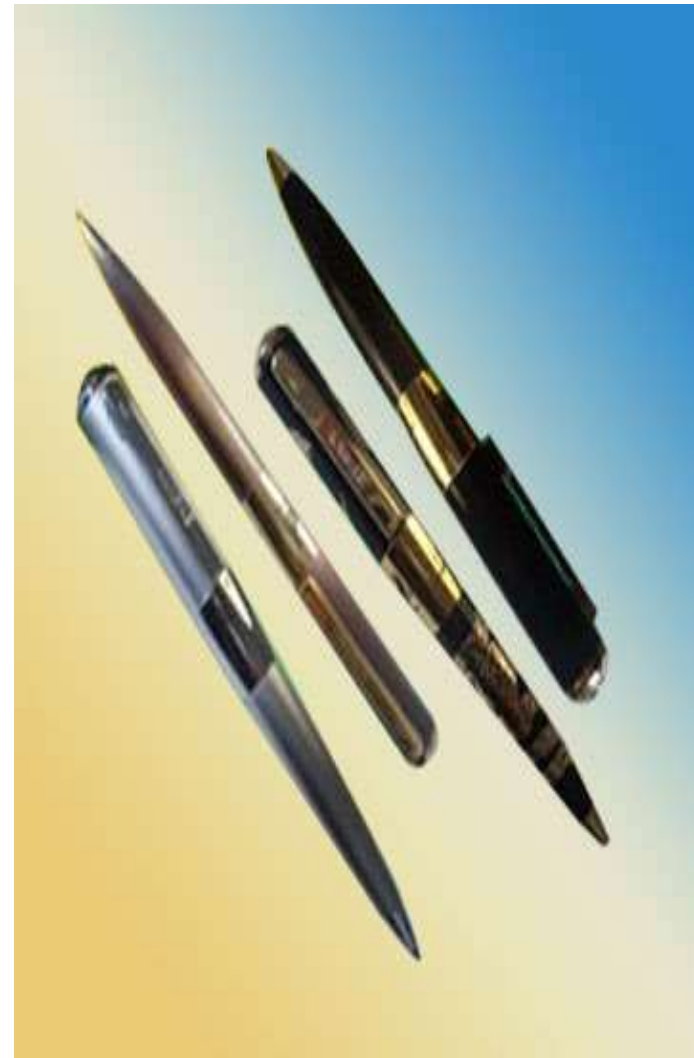
- Thường kết nối qua cổng USB
- Không phải dạng đĩa là bộ nhớ bán dẫn cực nhanh
- Dung lượng phát triển nhanh
- Gọn nhẹ và tiện lợi
- Đặc điểm đĩa Flash
 - 1) Supports USB full-speed (12MBps) transmission
 - 2) Driverless installation in Windows ME / 2000 / XP, Mac 9.0 and above, Linux 2.4 and above
 - 3) Supports boot-up by USB-HDD or USB-ZIP mode
 - 4) LED indicator displays status

Flash disk

- 5) Write protection switch
- 6) Reading and writing speed: 900k/s and 700k/s
- 7) Password protection and data encryption prevents unauthorized

access to data

- 8) Application software support in Windows OS security function
- 9) Application software resize (partition) available
- 10) Capacity: 16MB, 32MB, 64MB, 128MB, 256MB, 512MB, 1GB
- 11) Compliance: FCC(B), CE, C-Tick



GIỚI THIỆU TỔNG QUAN VỀ RAID

Xuất xứ

- ✓ RAID là cụm từ viết tắt nhóm từ **Redundant Array of Inexpensive (Independent) Disks**
- ✓ Thuật ngữ RAID được đưa ra trong một bài báo của một nhóm các nhà nghiên cứu tại Đại học tổng hợp California, Hoa Kỳ.
- ✓ RAID được đề xuất nhằm xóa bỏ khoảng trống lớn tốc độ CPU và các ổ đĩa điện cơ tương đối chậm.
- ✓ Hiệu suất thi hành vượt trội so với khi dùng một đĩa đơn lớn đắt tiền (SLED: Single Large Expensive Disk)

GIỚI THIỆU TỔNG QUAN VỀ RAID

Khái niệm

RAID: là cấu trúc đa đĩa vật lý để tạo nên một đĩa logic có kích thước lớn, độ tin cậy và khả năng vận hành cao hơn.

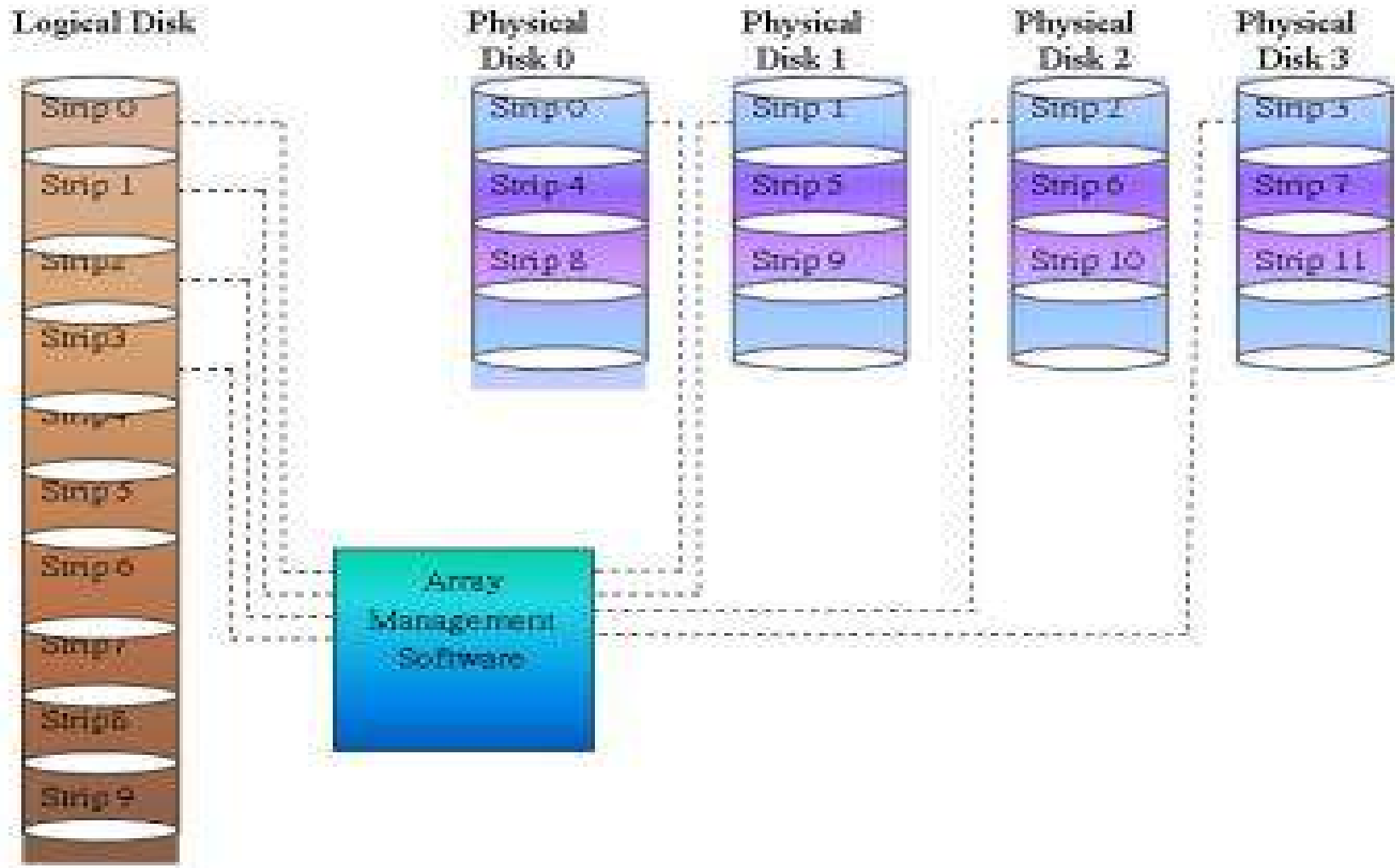
Mục đích

- ❖ **Nâng cao hiệu suất vận hành của toàn bộ hệ thống. Khả năng làm việc song song các đĩa.**
- ❖ **An toàn dữ liệu tận dụng tính dư thừa dữ liệu nhằm cải thiện độ tin cậy đĩa.**
- ❖ **Cung cấp bộ nhớ lớn**

GIỚI THIỆU TỔNG QUAN VỀ RAID

- **Đặc điểm chung của RAID**
- ❖ RAID là tập hợp các ổ đĩa vật lý được nhìn từ hệ điều hành như ổ đĩa logic đơn.
- ❖ Dữ liệu được phân bố trên mảng các ổ đĩa vật lý. Sử dụng kỹ thuật Striping. (Striping là kỹ thuật phân chia dữ liệu trên hai hay nhiều ổ đĩa làm tăng khả năng làm việc song song hệ thống)
- ❖ Dung lượng đĩa dư thừa được sử dụng để lưu trữ thông tin chẩn lẻ nhằm đảm bảo khả năng phục hồi dữ liệu trong trường hợp có hư hỏng về đĩa.

GIỚI THIỆU TỔNG QUAN VỀ RAID



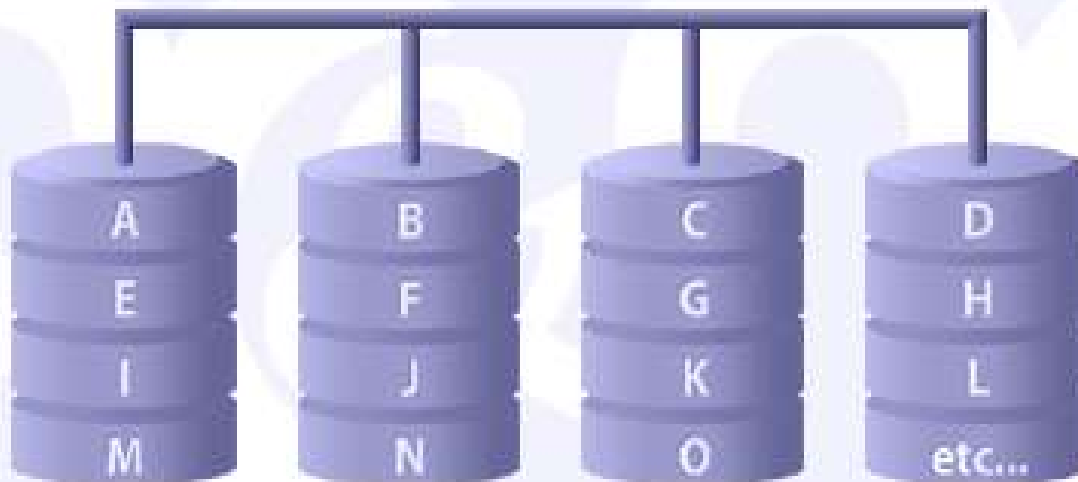


Các mức của RAID

Có 6 mức chính của RAID

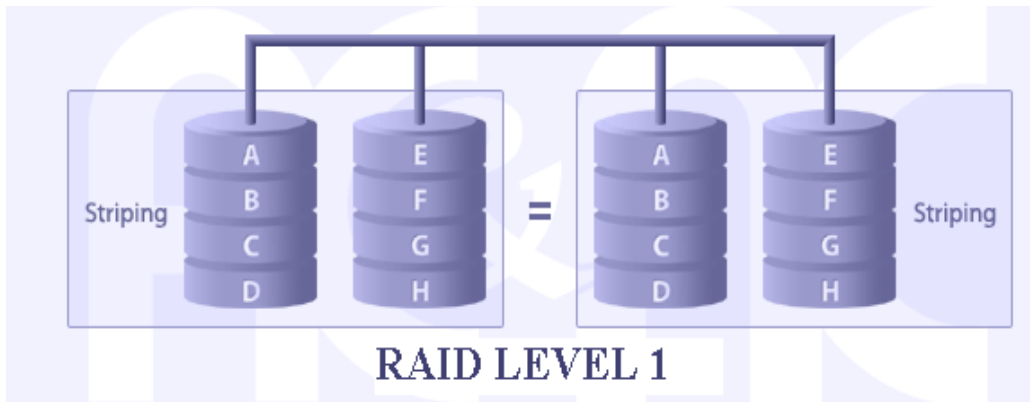
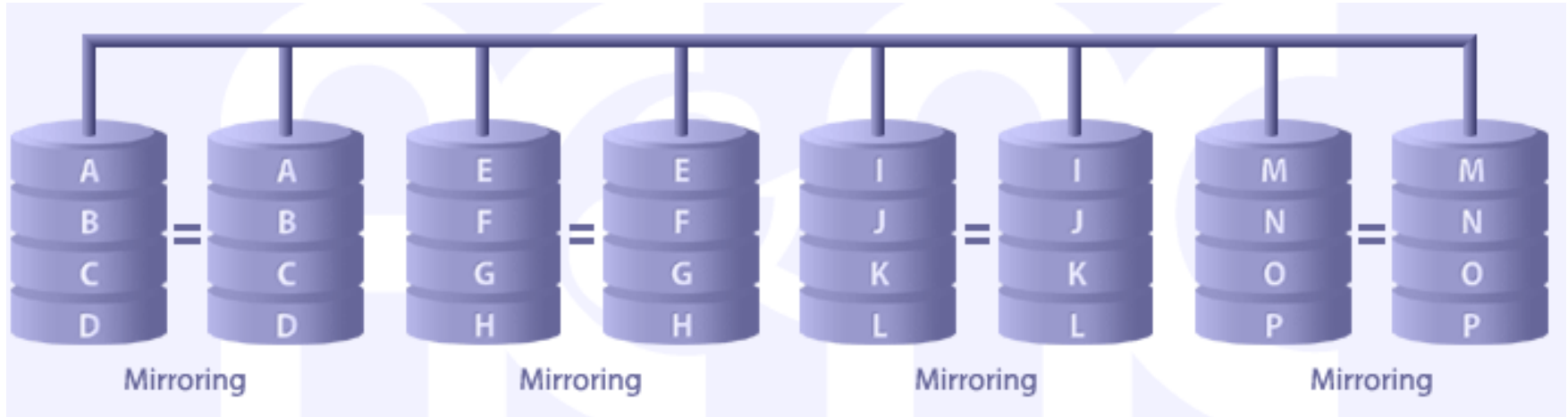
- RAID 0
- RAID 1
- RAID 2
- RAID 3
- RAID 4
- RAID 5

RAID LEVEL 0

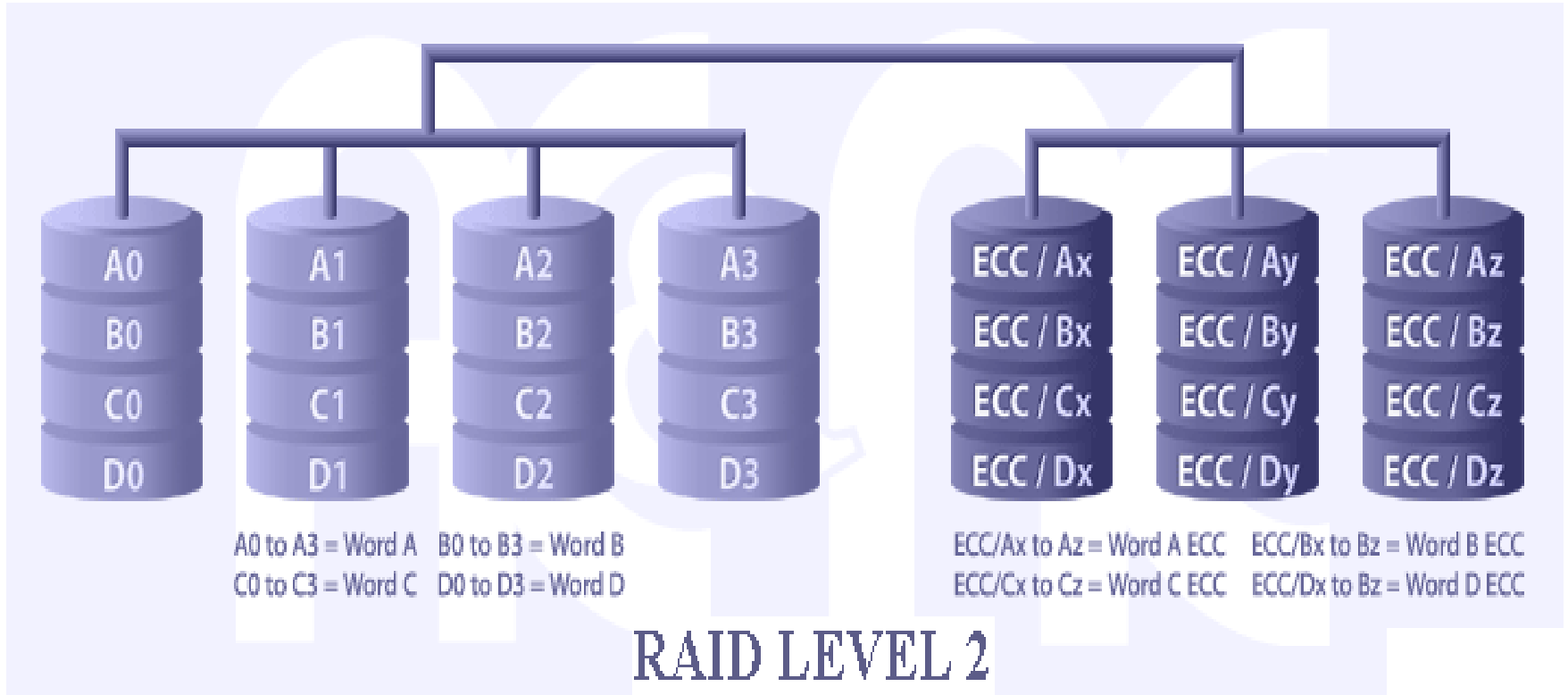


RAID LEVEL 0

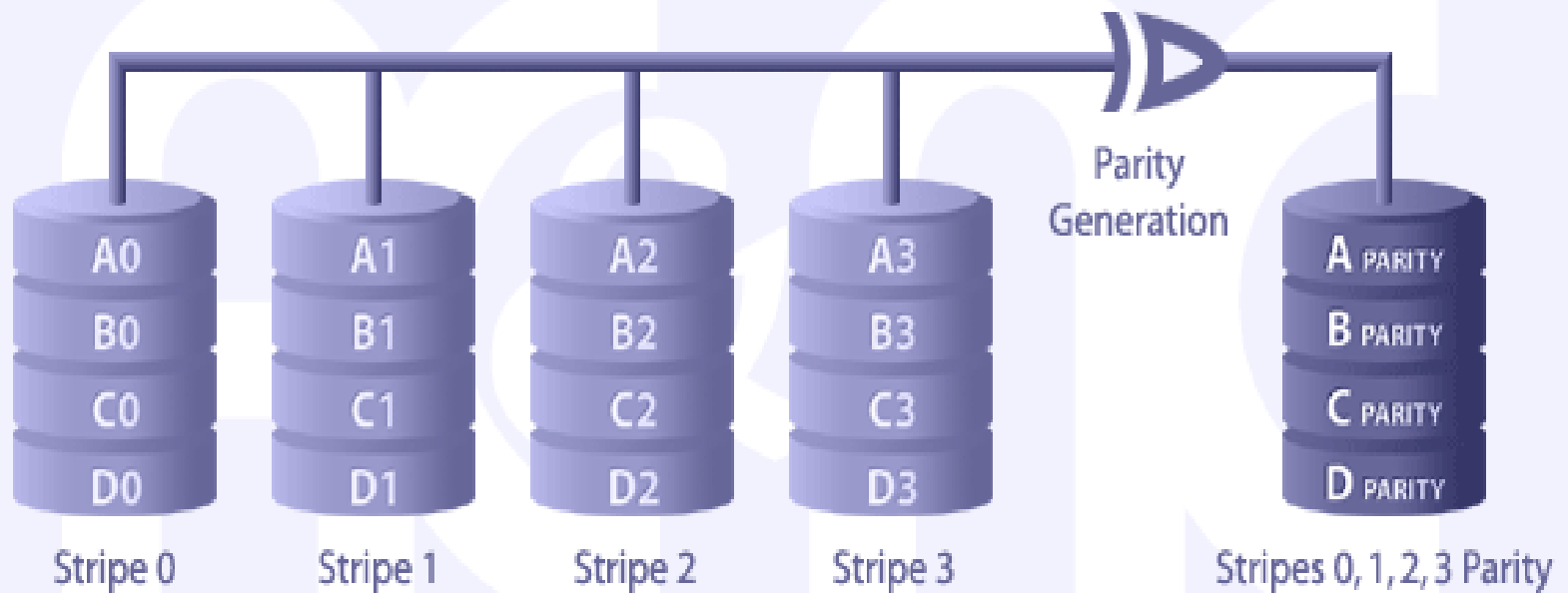
RAID LEVEL 1



RAID LEVEL 2

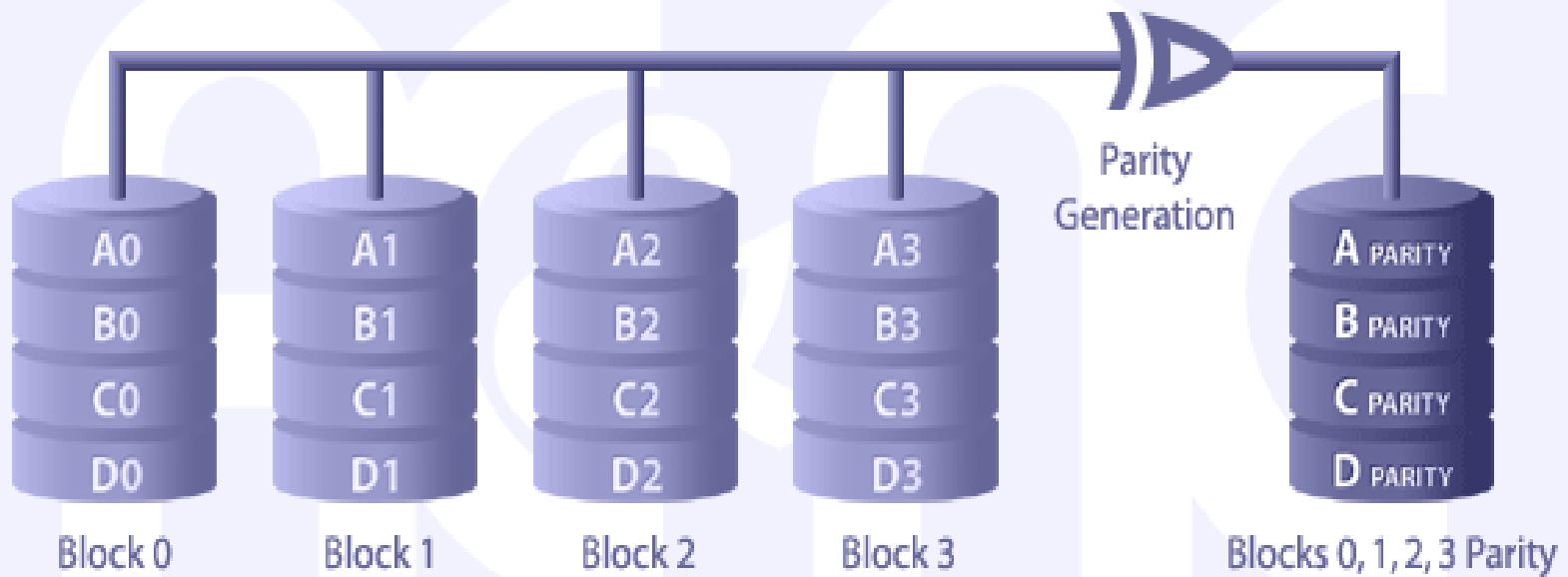


RAID LEVEL 3



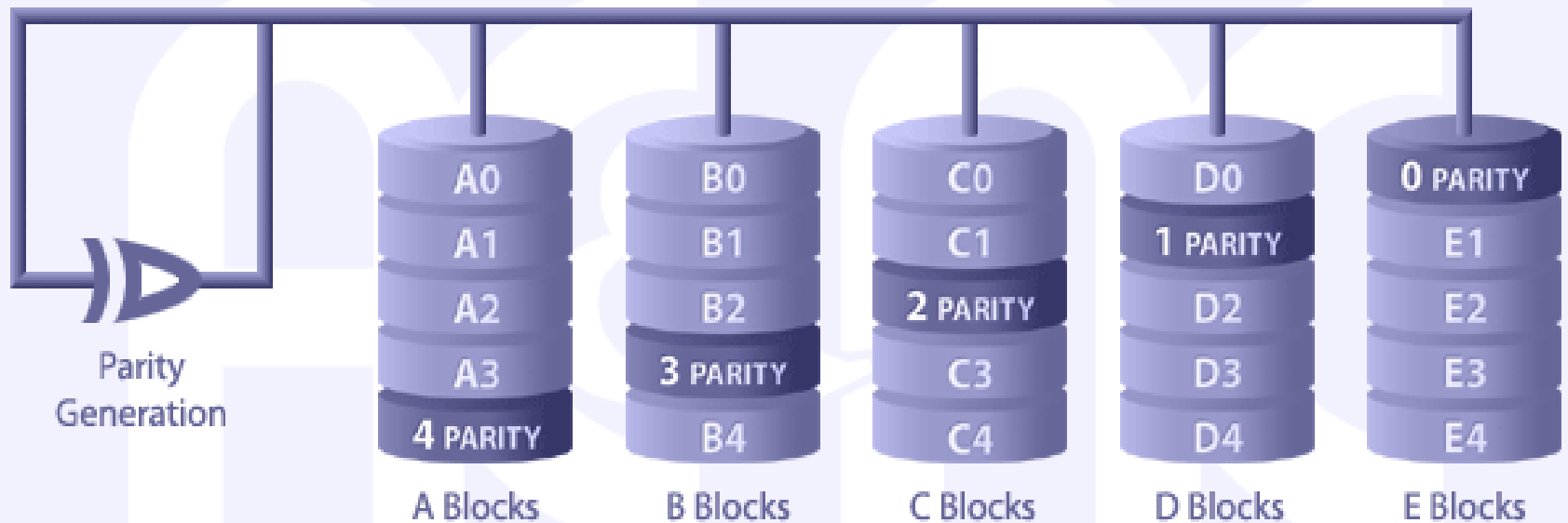
RAID LEVEL 3

RAID LEVEL 4



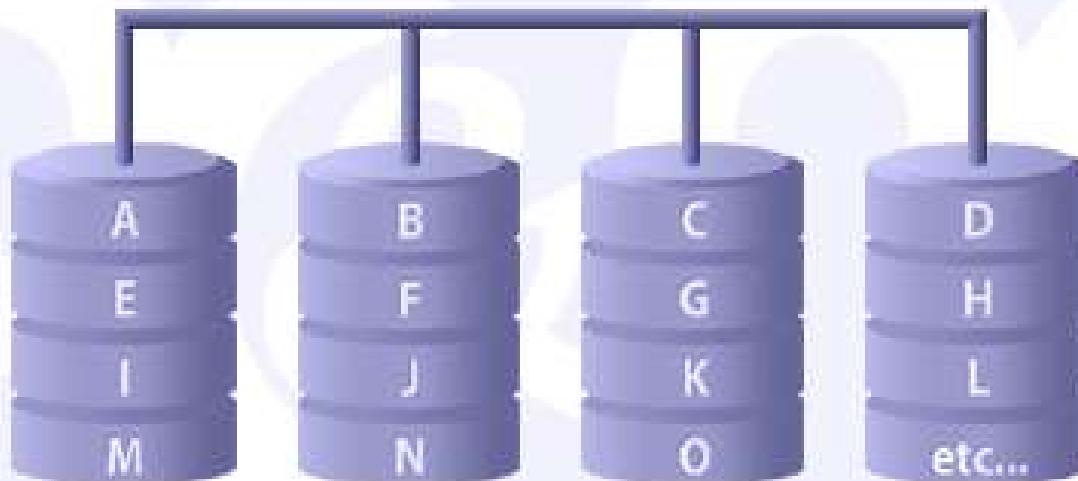
RAID LEVEL 4

RAID LEVEL 5



RAID LEVEL 5

RAID LEVEL 0



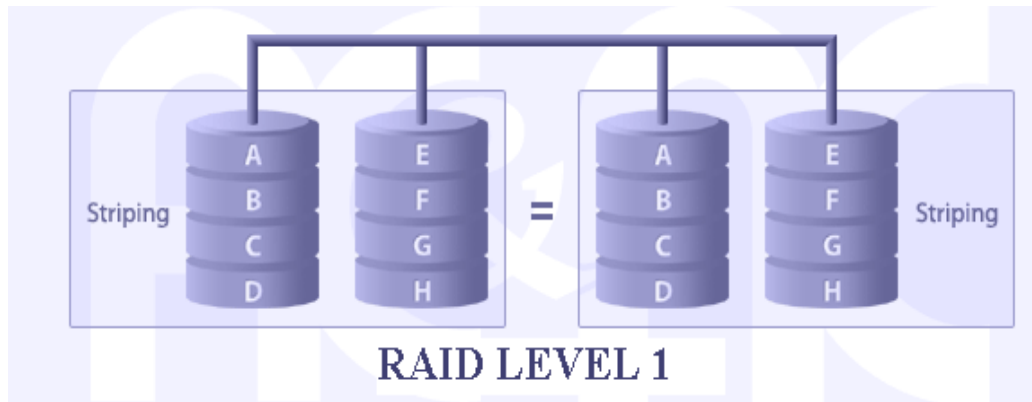
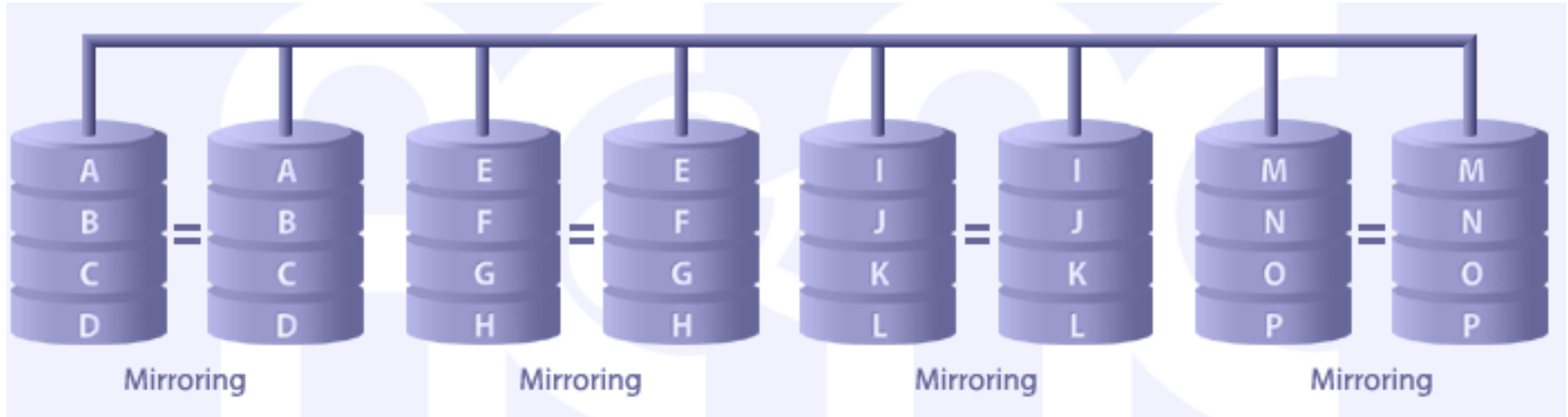
RAID LEVEL 0



Đặc điểm chung RAID mức 0

- Có thể coi RAID 0 không là thành viên của RAID
- Dữ liệu được phân chia nhiều đĩa => có khả năng truyền dữ liệu song song.
- Không lưu trữ dữ liệu dư thừa
- Phù hợp hệ thống đòi hỏi dung lượng nhớ lớn và khả năng vận hành cao hơn là độ tin cậy trong hệ thống.

RAID LEVEL 1



Đặc điểm chung RAID mức 1

- Là mức rất khác so các mức còn lại về cách lưu trữ dữ liệu dư thừa.
- Mỗi đĩa dữ liệu có một đĩa dự phòng đĩa dự phòng còn gọi **mirror disk**.

Ưu điểm:

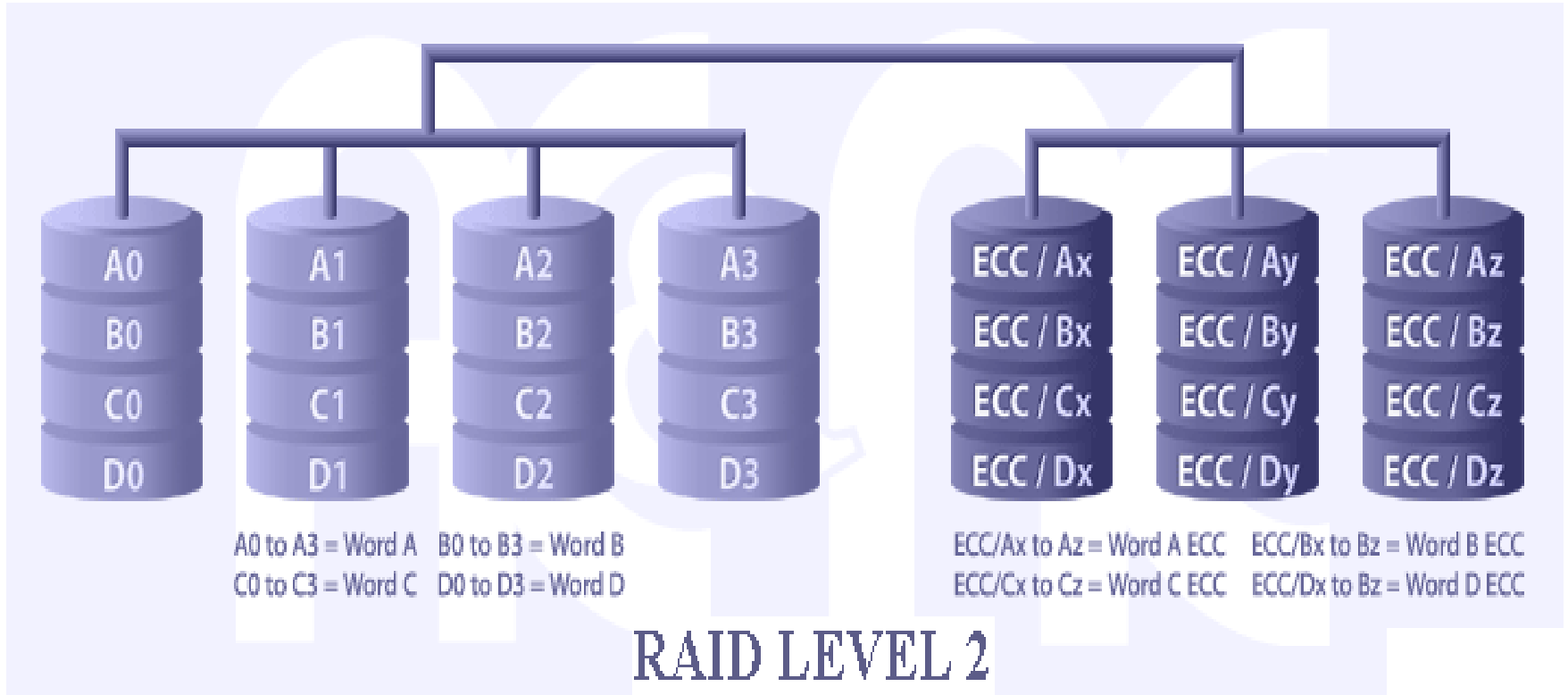
- Đáp ứng yêu cầu vào ra hệ thống
- Phục hồi dữ tốt nhất trong các mức của RAID

Nhược điểm:

- Khả năng cập nhật dữ liệu chậm
- Chi phí mua đĩa cao

KQ: Vận hành tốt cho hệ thống thường xuyên truy xuất dữ liệu

RAID LEVEL 2



Đặc điểm chung RAID mức 2

- Sử dụng công nghệ truy cập song song.
- Tất cả đĩa đều vận hành tham gia yêu cầu trao đổi dữ liệu.
- Kích thước Strip có thể byte hay word.
- Có sử dụng mã Hamming để phát hiện lỗi và sửa lỗi

Ưu điểm:

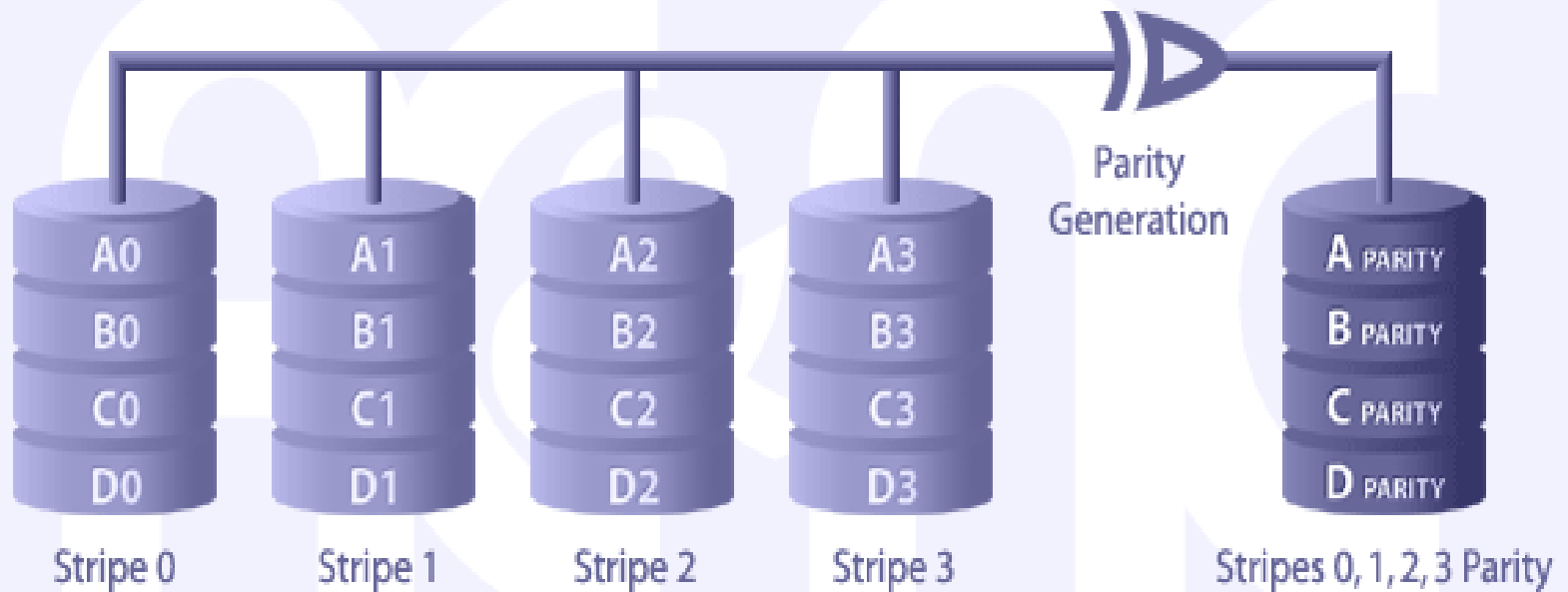
- Có khả năng phát hiện lỗi và sửa những lỗi đơn hệ thống.
- Số đĩa sử dụng ít hơn so mức RAID 1

Nhược điểm:

- Chi phí mua đĩa cao.

KQ: Ứng dụng trong hệ thống hay xuất hiện lỗi

RAID LEVEL 3



RAID LEVEL 3

Đặc điểm chung RAID mức 3

- Giống RAID 2 nhưng tổ chức đơn giản hơn. Sử dụng một đĩa dự phòng.
- Tất cả đĩa đều vận hành tham gia yêu cầu trao đổi dữ liệu.
- Kích thước Strip có thể byte hay word.
- Có sử dụng mã Parity để phục hồi dữ liệu.

Ưu điểm:

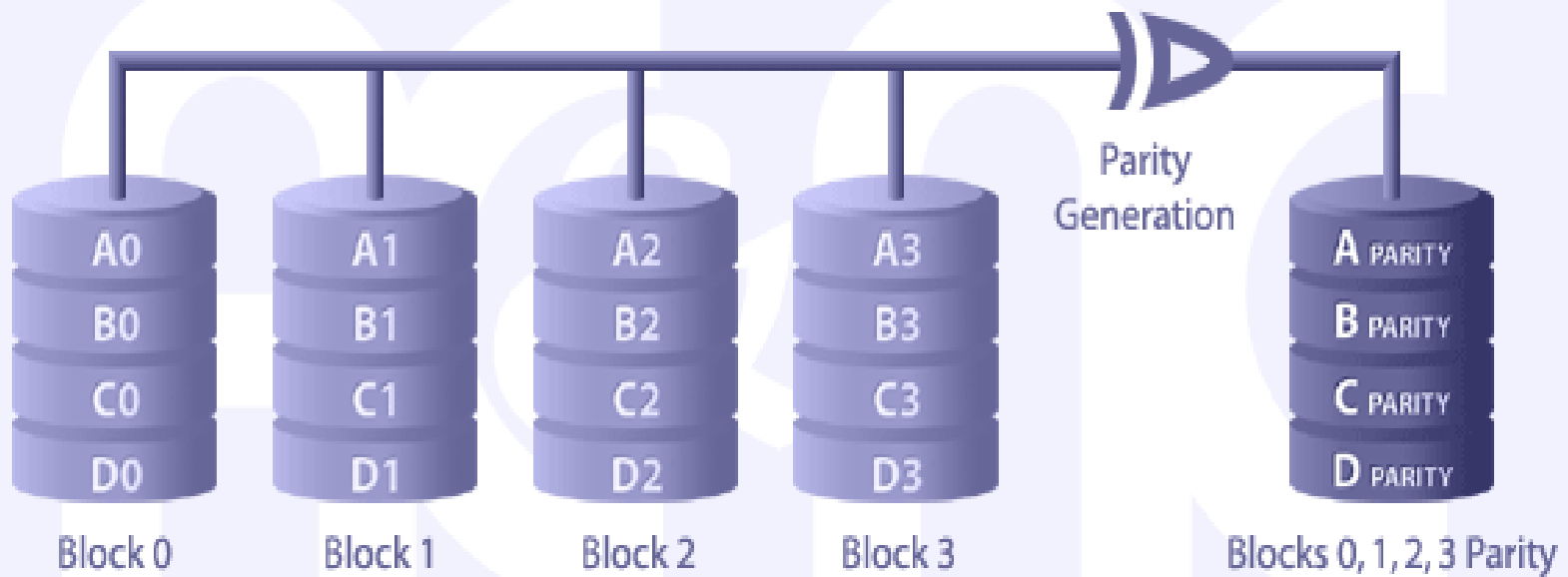
- Có khả năng truyền dữ liệu song song.
- Số đĩa sử dụng dự phòng là 1 đĩa. Chi phí thấp.

Nhược điểm:

- Tại một thời điểm chỉ thỏa mãn một yêu cầu vào ra.

KQ: Ứng dụng trong hệ thống hay xuất hiện lỗi

RAID LEVEL 4



RAID LEVEL 4

Đặc điểm chung RAID mức 4

- Giống RAID 3 những tổ chức đơn giản hơn. Sử dụng một đĩa dự phòng.
- Dữ liệu tổ chức thành khối.
- Các đĩa sử dụng phương pháp truy cập độc lập.
- Có sử dụng mã Parity để phục hồi dữ liệu.

Ưu điểm:

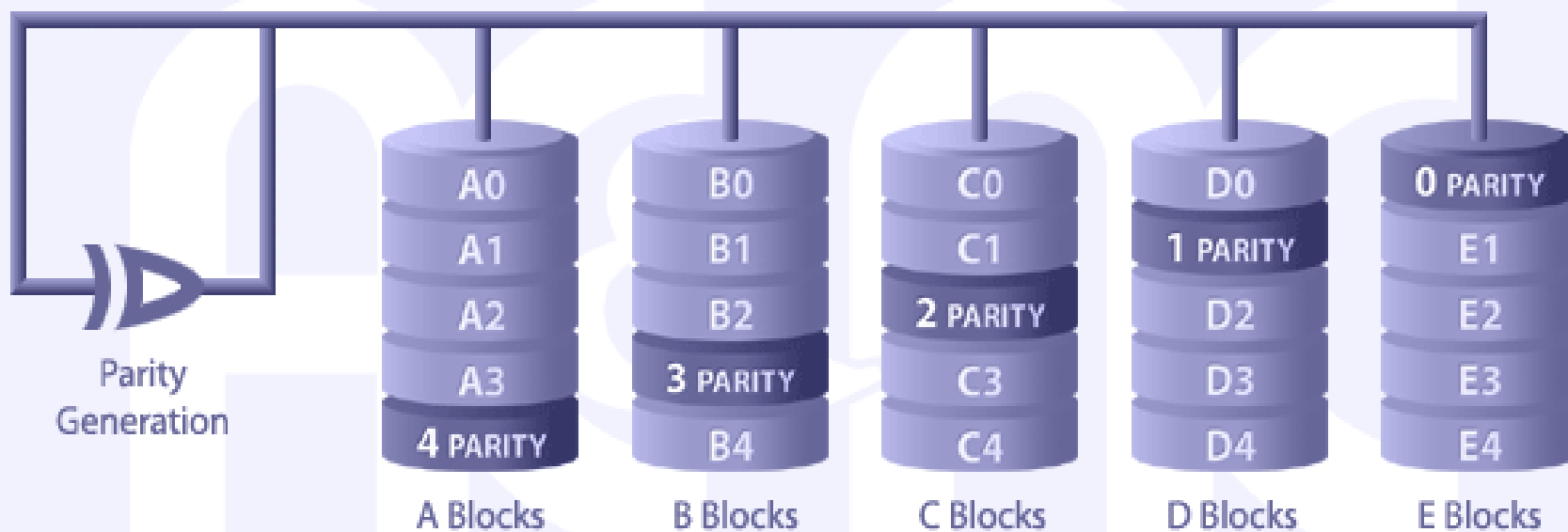
- Có khả năng đáp ứng nhiều yêu cầu vào ra đồng thời.
- Số đĩa sử dụng dự phòng là 1 đĩa. Chi phí thấp.

Nhược điểm:

- Khả năng truyền dữ liệu song song là kém.

KQ: Ứng dụng trong hệ thống hay xuất hiện lỗi

RAID LEVEL 5



RAID LEVEL 5

Đặc điểm chung RAID mức 5

- Giống RAID 4, tuy nhiên sự phân bố đều thông tin dự phòng tránh được hiện tượng tắc nghẽn(bottle neck)
- Dữ liệu tổ chức thành khối.
- Các đĩa sử dụng phương pháp truy cập độc lập.
- Có sử dụng mã Parity để phục hồi dữ liệu.

Ưu điểm:

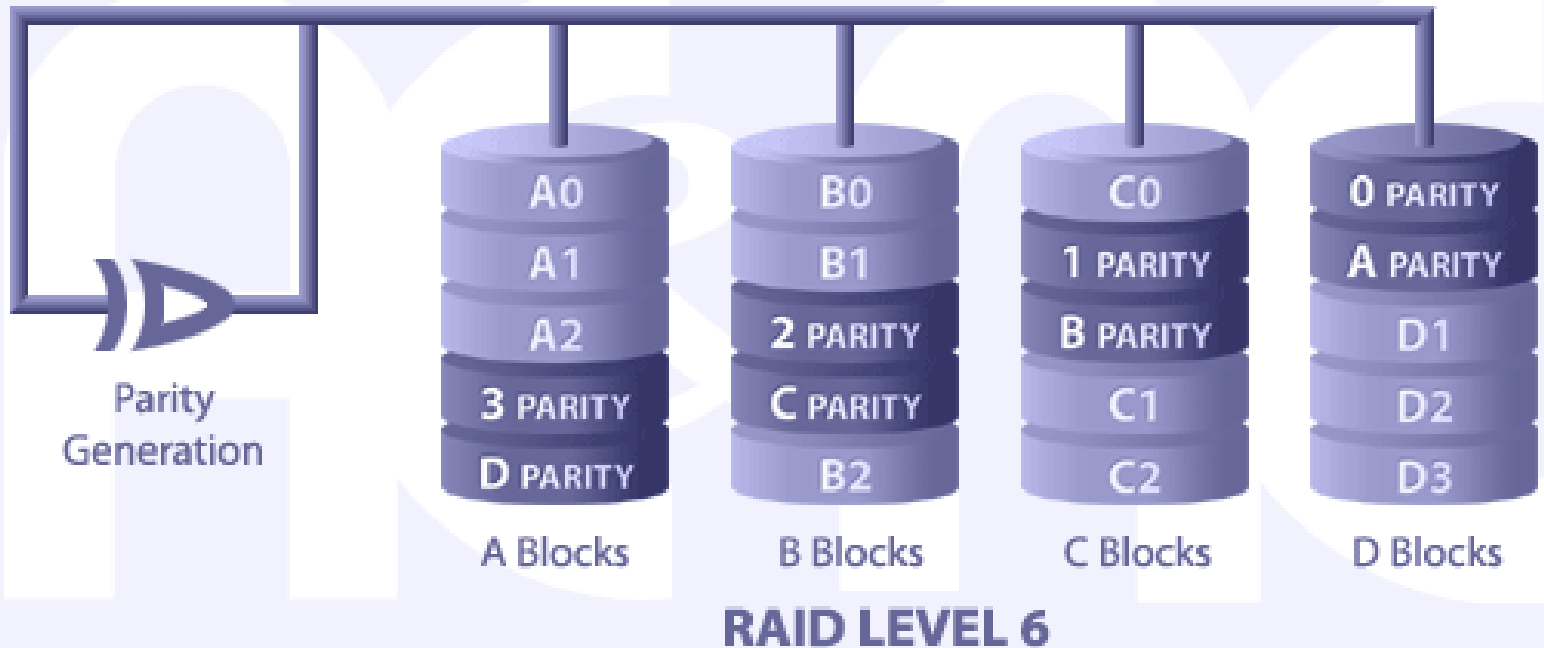
- Có khả năng đáp ứng nhiều yêu cầu vào ra đồng thời.
- Số đĩa sử dụng dự phòng là 1 đĩa. Chi phí thấp.

Nhược điểm:

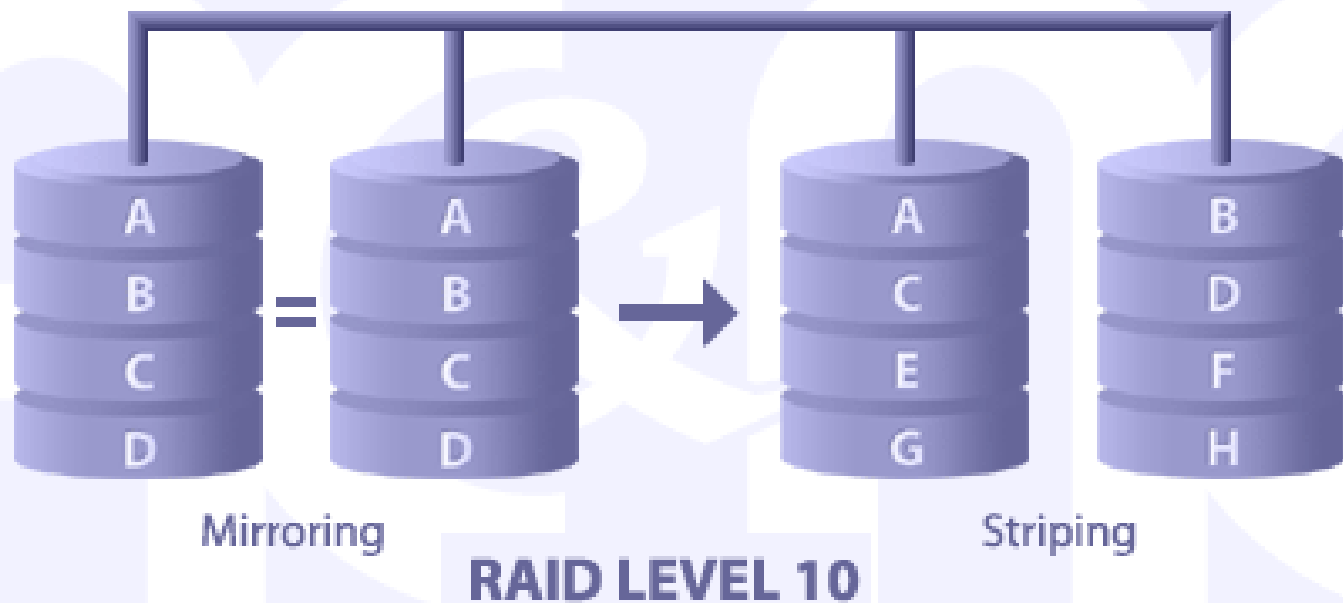
- Khả năng truyền dữ liệu song song là kém.

KQ: Ứng dụng nhiều trong thực tế.

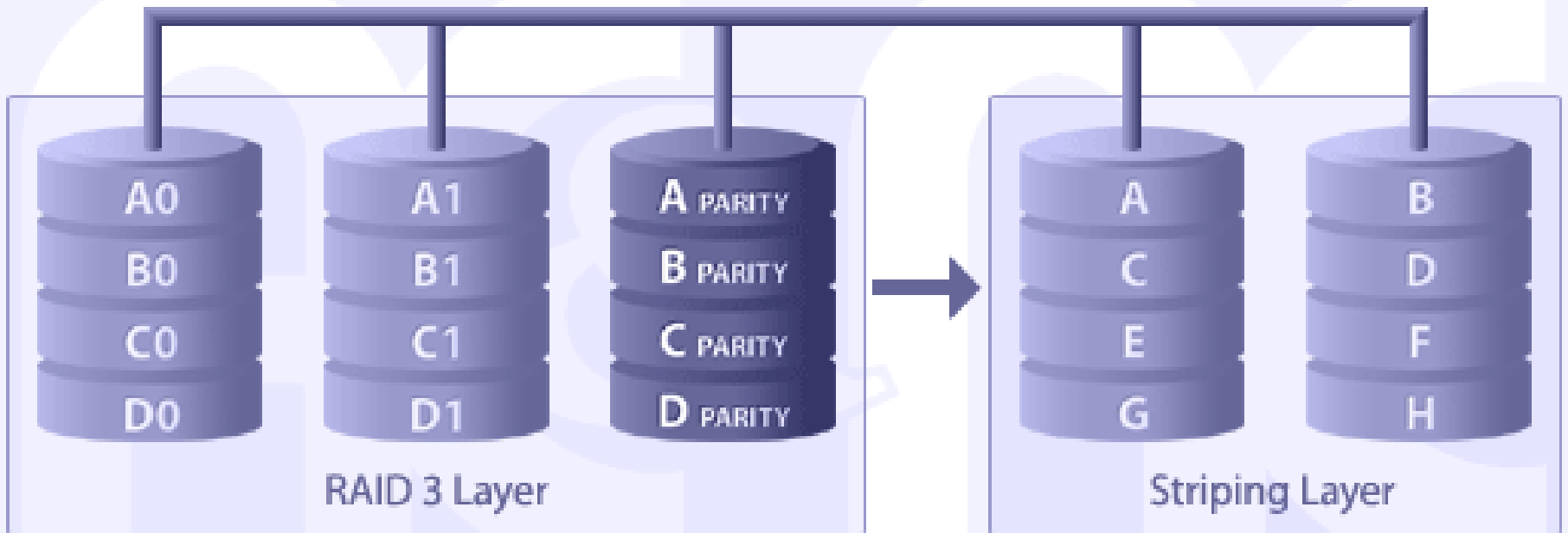
RAID LEVEL 6



RAID LEVEL 10



RAID LEVEL 50



RAID LEVEL 50 :

5.6 Hệ thống nhớ trên máy PC hiện nay

- Hệ thống Cache: tích hợp trực tiếp trên các chip vi xử lý
- Bộ nhớ chính: tồn tại dưới dạng module nhớ RAM
- SIMM: Single Inline Memory Module
- 30 pin : 8 đường dữ liệu
- 72 pin : 32 đường dữ liệu
- DIMM: Dual Inline Memory Module
- 168 pin: 64 đường dữ liệu
- RIMM: Rambus Inline Memory Module



ROM BIOS

ROM BIOS: Basic Input Output System ROM chứa chương trình sau:

- Chương trình POST (Power On Self Test)
- Chương trình CMOS setup (Complementary Metal Oxide Semiconductor)
- Chương trình Bootstrap Loader
- Chương trình điều khiển vào ra cơ bản (BIOS)
- CMOS RAM
- Chứa cấu hình hệ thống hiện thời
- Đồng hồ và ngày tháng năm hệ thống
- Có pin nuôi riêng

Chương 6

Giới thiệu chung

6.1 Tổng quan về hệ thống vào ra

6.2 Các phương pháp điều khiển vào ra

6.3 Nối ghép thiết bị ngoại vi

6.4 Các cổng vào ra thông dụng

6.1 Tổng quan về hệ thống vào ra

Giới thiệu chung hệ thống vào ra

Chức năng: Trao đổi thông tin giữa Máy tính với môi trường bên ngoài.

Các thao tác cơ bản:

- ✓ Vào dữ liệu
- ✓ Ra dữ liệu

Các thành phần chính:

- ✓ Thiết bị ngoại vi
- ✓ Module ghép nối vào ra

6.1 Tổng quan về hệ thống vào ra

Thiết bị ngoại vi

Chức năng: phương tiện chuyển đổi thông tin giữa bên trong và bên ngoài máy tính

Đặc điểm các thiết bị

Trên thị trường tồn tại rất nhiều các thiết bị ngoại vi khác nhau về: Nguyên tắc hoạt động, tốc độ, định dạng dữ liệu truyền, v.v. Đồng thời các thiết bị này có tốc độ làm việc chậm hơn CPU và RAM rất nhiều. Chính vì lý do trên cần có Module vào ra để ghép nối các thiết bị ngoại vi vào hệ thống BUS máy tính.

6.1 Tổng quan về hệ thống vào ra

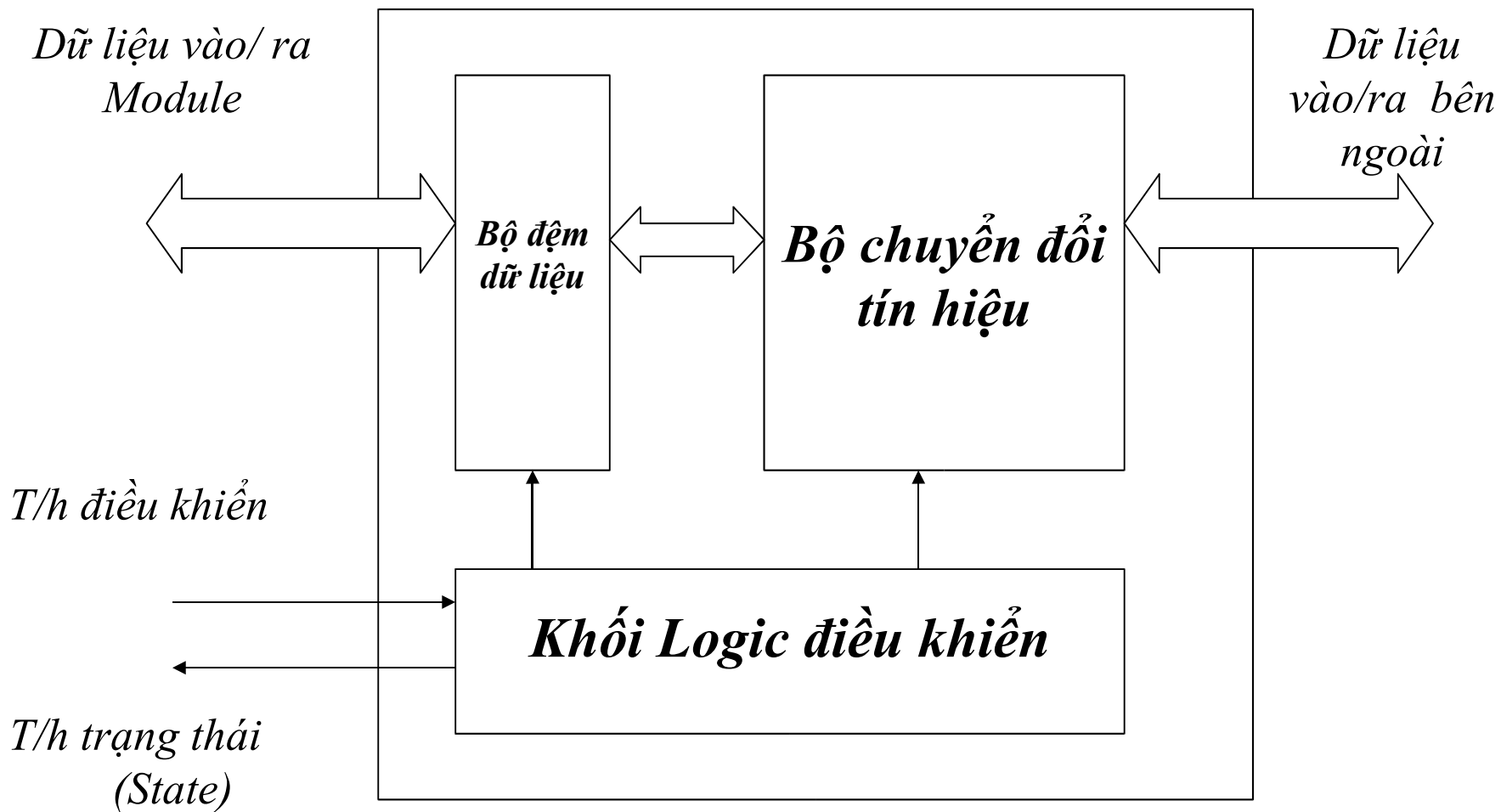
Phân loại:

- Thiết bị nhập: Keyboard, Mouse, Scan, Micro,...
- Thiết bị xuất: Monitor, Printer,
- Thiết bị xuất nhập: Modem, NIC, Driver,...

Cấu trúc tổng quát của thiết bị ngoại vi:

- ✓ **Bộ chuyển đổi tín hiệu:** chuyển đổi dữ liệu giữa bên trong và bên ngoài Máy tính
- ✓ **Bộ đệm dữ liệu:** nơi lưu trữ dữ liệu trung gian giữa Máy tính và thiết bị ngoại vi, đặt bên trong thiết bị ngoại vi.
- ✓ **Khối logic điều khiển:** điều khiển hoạt động của thiết bị ngoại vi theo tín hiệu từ Module I/O gửi tới thiết bị.

6.1 Tổng quan về hệ thống vào ra



6.1 Tổng quan về hệ thống vào ra

Module I/O

Chức năng: Nối ghép thiết bị ngoại vi với bus của máy tính.

- ✓ Điều khiển và định thời
- ✓ Trao đổi thông tin với CPU
- ✓ Trao đổi thông tin với thiết bị ngoại vi
- ✓ Đệm giữa máy tính với thiết bị ngoại vi
- ✓ Phát hiện lỗi của các thiết bị ngoại vi.

Cấu trúc chung:

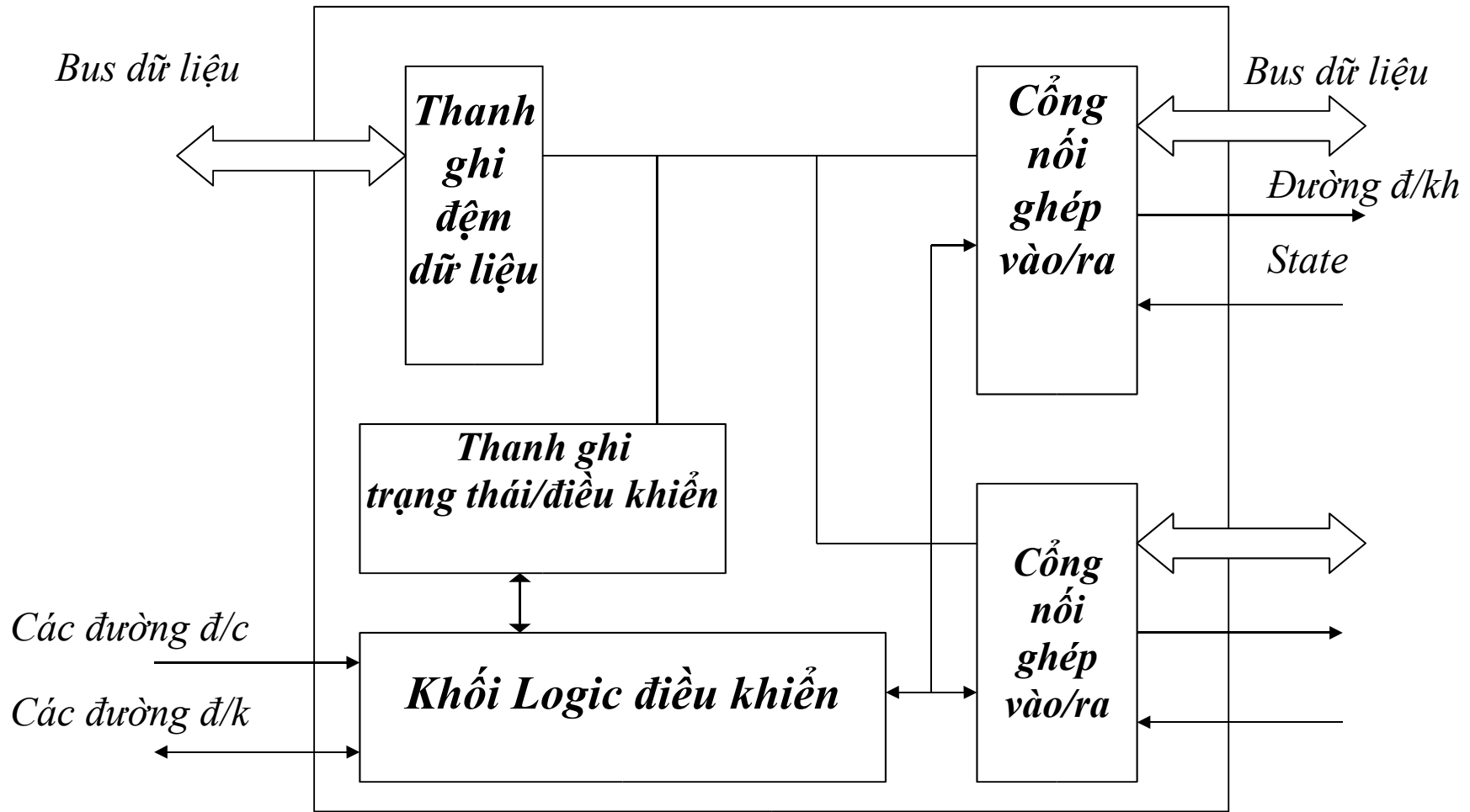
Thanh ghi đệm dữ liệu: đệm dữ liệu trong quá trình trao đổi

Cổng nối ghép vào ra: kết nối thiết bị ngoại vi, mỗi cổng có địa chỉ xác định và chuẩn kết nối riêng phụ thuộc sơ đồ chân.

Thanh ghi trạng thái/điều khiển: lưu trữ thông tin trạng thái cho các cổng vào ra

Khối logic điều khiển: điều khiển Module vào ra

6.1 Tổng quan về hệ thống vào ra



Ví dụ cổng ghép nối song song(LPT)

- Các đường dẫn của cổng song được nối với 3 thanh ghi 8 bit khác nhau:
 - ✓ Thanh ghi dữ liệu (Địa chỉ cơ sở)
 - ✓ Thanh ghi trạng thái (Địa chỉ cơ sở +1)
 - ✓ Thanh ghi điều khiển (Địa chỉ cơ sở +2)
- Các địa chỉ cổng có thể là:
 - LPT1: 378h (379h ; 37Ah)
 - LPT2: 3BCh
 - LPT3: 278h
 - LPT4: 2BCh

Ví dụ cổng ghép nối song song(LPT)

- Hợp ngữ:

Để xuất ra dữ liệu: **OUT DX, AL** hoặc **OUT DX, AX**

Để nhập vào dữ liệu: **IN AL, DX** hoặc **IN AX, DX**

(DX chứa địa chỉ; AL chứa giá trị)

- Turbo C

Để xuất ra dữ liệu: **outportb(địa_chỉ, giá_trị)**

Để nhập vào dữ liệu: **bien = inportb(địa_chỉ)**

- Turbo Pascal

Để xuất ra dữ liệu: **port[địa_chỉ]:= giá_trị**

Để nhập vào dữ liệu: **bien:=port[địa_chỉ]**

Ví dụ cổng ghép nối tiếp(COM)

- Các thanh ghi chính:
 - ✓ Thanh ghi đệm (Buffer Register) Địa chỉ cơ sở
 - ✓ Thanh ghi trạng thái (Status Register) ĐCCS+5
 - ✓ Thanh ghi điều khiển (Control Register) ĐCCS+3
- Các địa chỉ cổng có thể là:
 - COM1: 3F8h (3FDh ; 3FBh)
 - COM2: 2F8h
 - COM3: 3E8h
 - COM4: 2E8h

6.2 Các phương pháp điều khiển vào ra

Phân loại:

- ☞ Vào ra bằng chương trình
- ☞ Vào ra bằng ngắt
- ☞ Truy cập bộ nhớ trực tiếp DMA

Vào ra bằng chương trình

Nguyên tắc chung:

- ✓ Sử dụng lệnh vào ra trong chương trình để trao đổi dữ liệu với cổng vào ra.
- ✓ Khi CPU thực hiện chương trình gặp lệnh vào ra thì CPU điều khiển trao đổi dữ liệu với cổng vào ra.

Lệnh I/O:

- ✓ Với không gian địa chỉ vào ra riêng biệt: sử dụng các lệnh vào ra chuyên dụng
- ✓ Với không gian vào ra dùng chung bộ nhớ thì các lệnh trao đổi dữ liệu sử dụng như ngăn nhớ.

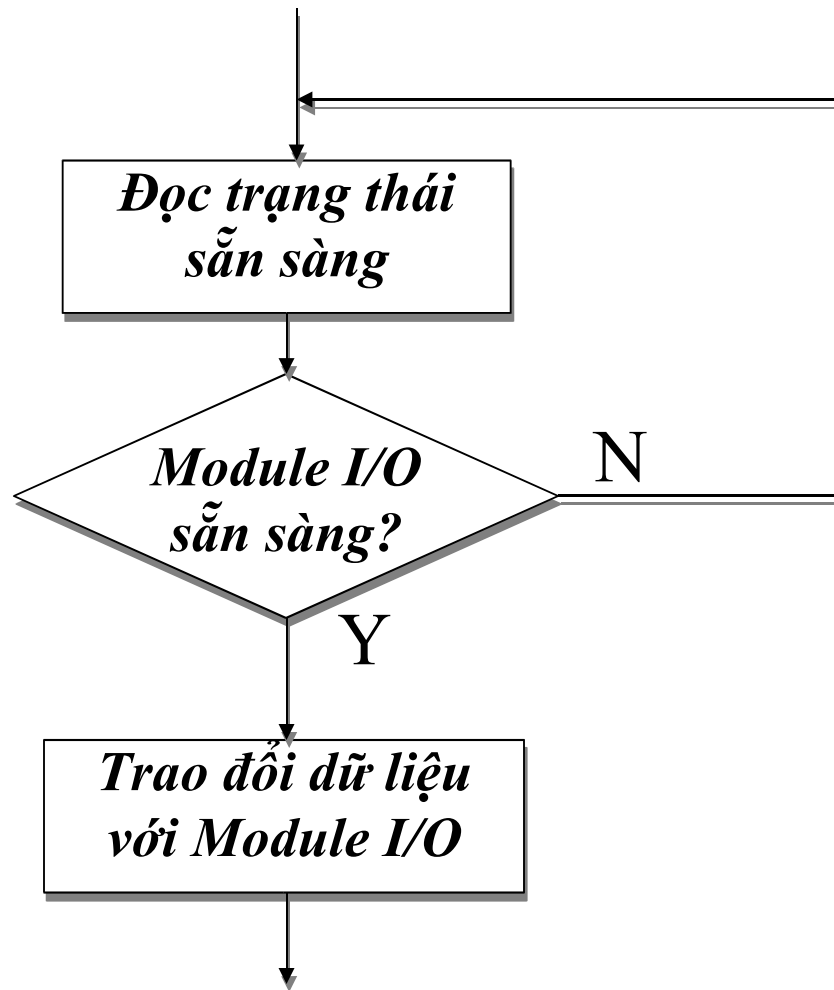
6.2 Các phương pháp điều khiển vào ra

Hoạt động vào ra bằng chương trình

- ❖ CPU gặ̣p lệnh trao đ̣i vào ra, yêu cầu thao tác vào ra
- ❖ Module vào ra thao tác vào ra
- ❖ Module vào ra thiết lập các bit trạng thái(State)
- ❖ CPU kiểm tra các bit trạng thái:
 - Nếu chưa sẵn sàng thì quay lại kiểm tra lại
 - Nếu sẵn sàng thì chuyển sang trao đ̣i dữ liệu với Module vào ra.

6.2 Các phương pháp điều khiển vào ra

Lưu đồ thực hiện chương trình:



6.2 Các phương pháp điều khiển vào ra

Nhận xét:

- ✓ CPU trực tiếp điều khiển vào ra: đọc trạng thái, kiểm tra trạng thái, thực hiện trao đổi.
- ✓ Trong trường hợp nhiều thiết bị cùng cần trao đổi dữ liệu và thiết bị chưa sẵn sàng tốn rất nhiều thời gian CPU
- ✓ Việc thực hiện trao đổi đơn giản

Vào ra bằng ngắt

Nguyên tắc chung:

- ✓ CPU không phải đợi trạng thái sẵn sàng của Module vào ra.
- ✓ Module vào ra khi nó sẵn sàng phát ra tín hiệu yêu cầu ngắt CPU
- ✓ CPU thực hiện chương trình vào ra tương ứng để trao đổi dữ liệu.
- ✓ CPU trở lại chương trình đang bị ngắt.

6.2 Các phương pháp điều khiển vào ra

Các phương pháp nối ghép

- Sử dụng nhiều đường yêu cầu ngắt.
- Kiểm tra vòng bằng phần mềm (Polling)
- Kiểm tra vòng bằng phần cứng
- Sử dụng bộ điều khiển ngắt.
- Nhiều yêu cầu ngắt đồng thời
- CPU sử dụng nhiều đường yêu cầu ngắt. Nạp vào thanh ghi yêu cầu ngắt.
- Hạn chế số lượng Module vào ra
- Các đường ngắt được quy định mức ưu tiên.

6.2 Các phương pháp điều khiển vào ra

- CPU phát ra tín hiệu chấp nhận ngắt đến Module đầu tiên.
- Nếu Module đó không gây ra ngắt thì nó gởi tín hiệu đó tới các Module kế tiếp
- Module I/O gây ngắt sẽ đặt vector lên bus dữ liệu
- CPU sử dụng ngắt để xác định chương trình con điều khiển ngắt
- Thứ tự vào ra các Module trong chuỗi xác định thứ tự ưu tiên.

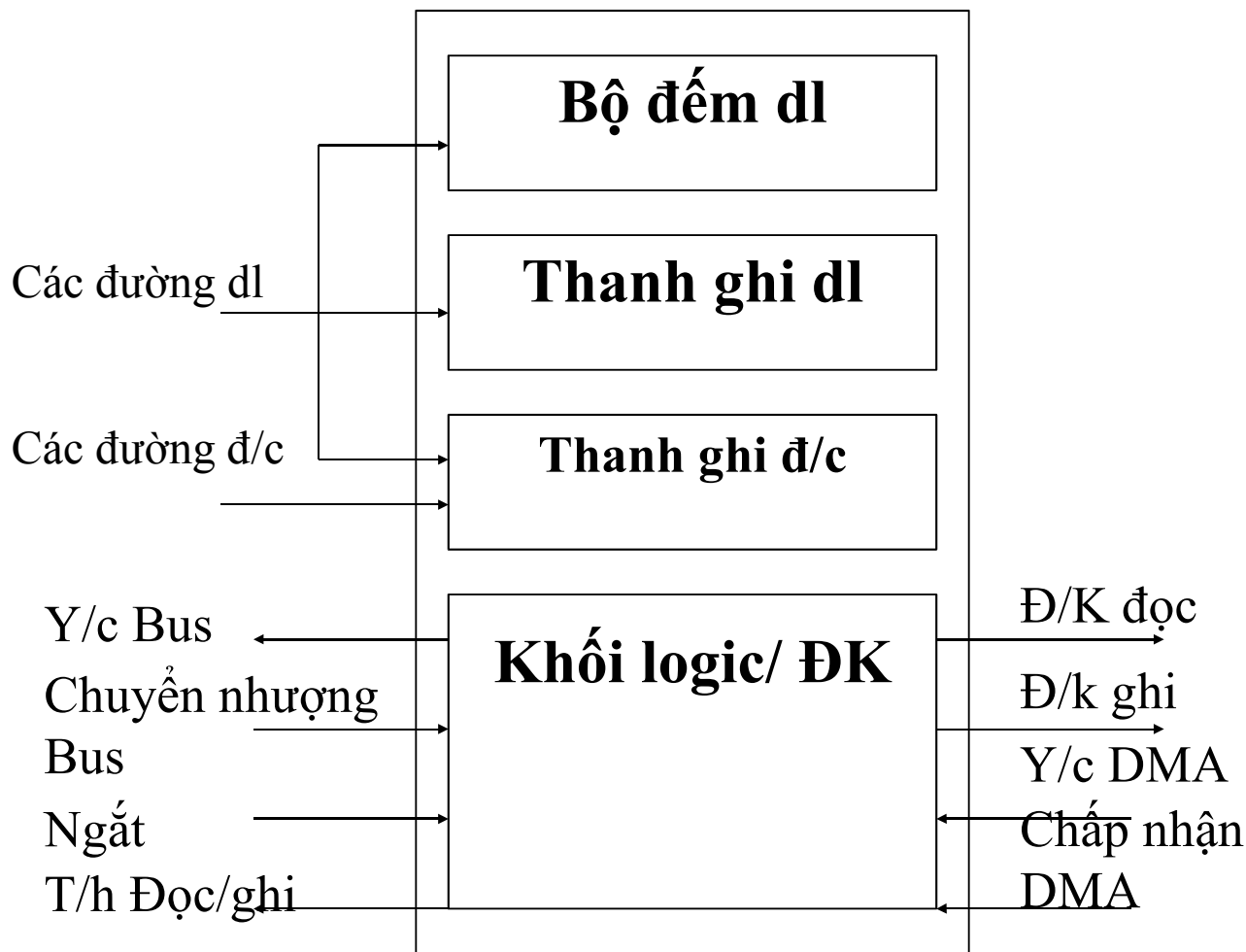
6.2 Các phương pháp điều khiển vào ra

Truy nhập bộ nhớ trực tiếp

(DMA: Direct Memory Access)

Với nhược điểm chính của hai phương pháp trên là: CPU tham gia trực tiếp vào trao đổi dữ liệu và việc trao đổi lượng dữ liệu nhỏ. Để khắc phục hai phương pháp trên một phương pháp mới có tên DMA sẽ sử dụng thêm một Module phần cứng có DMAC (DMA Controller). Vì vậy khi trao đổi dữ liệu không cần CPU.

6.2 Các phương pháp điều khiển vào ra



6.2 Các phương pháp điều khiển vào ra

Các thành phần của DMAC

- Thanh ghi dữ liệu: chứa dữ liệu trao đổi.
- Thanh ghi địa chỉ: chứa địa chỉ của ngăn nhớ dữ liệu
- Bộ đếm dữ liệu: chứa số từ dữ liệu cần trao đổi
- Khối logic điều khiển: điều khiển hoạt động của DMAC

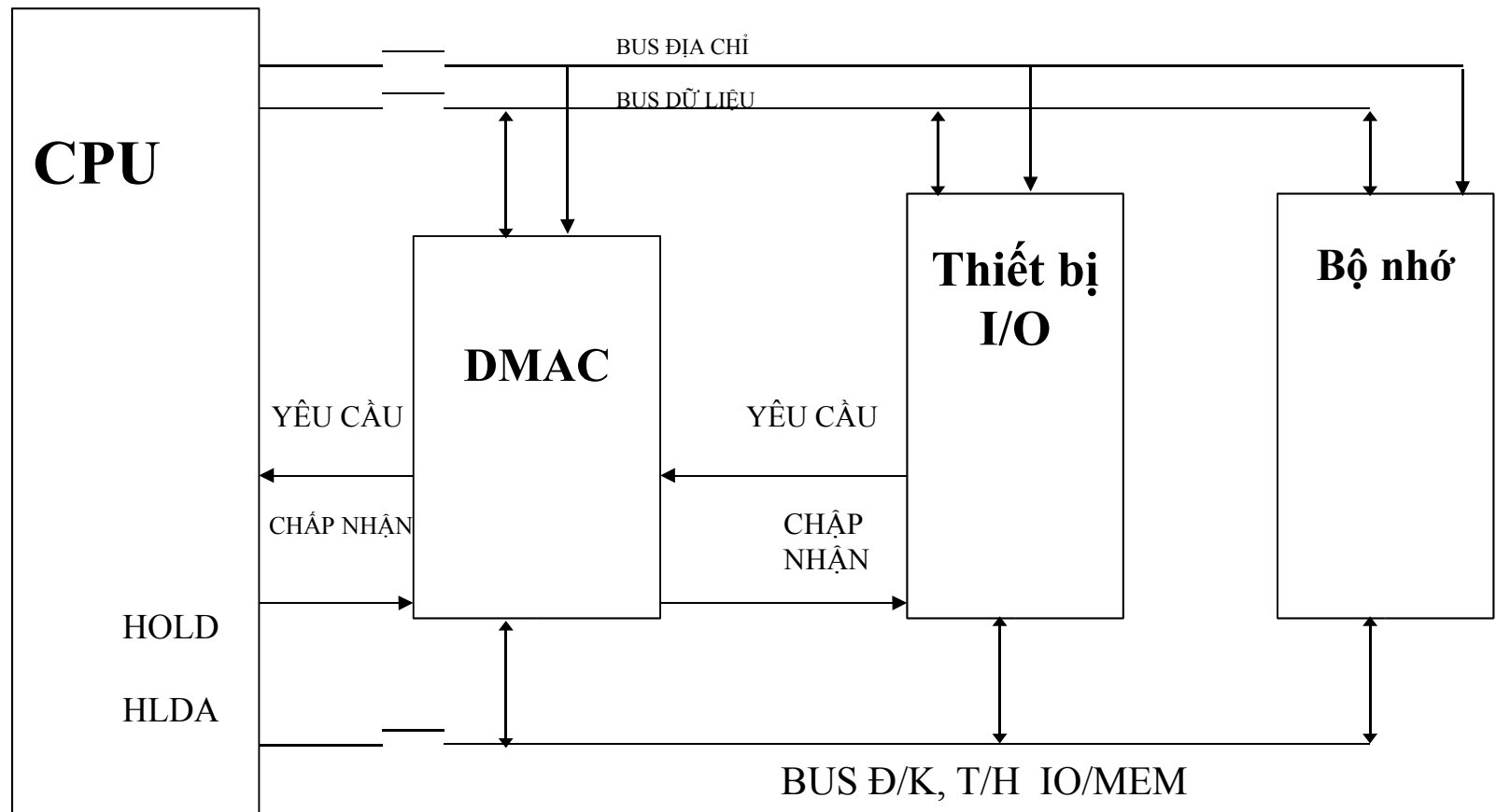
Hoạt động của DMA

- Khi cần vào ra dữ liệu thì CPU nhờ DMAC tiến hành vào ra dữ liệu với thông tin cho biết như sau:

6.2 Các phương pháp điều khiển vào ra

- Địa chỉ thiết bị vào ra
- Địa chỉ đầu của mảng nhớ chứa dữ liệu và DMAC nạp thanh ghi địa chỉ
- Số từ dữ liệu cần truyền và DMAC nạp vào bộ đếm dữ liệu
- CPU sẽ đi thực hiện việc khác
- DMAC điều khiển việc trao đổi dữ liệu sau khi truyền một từ dữ liệu thì nội dung thanh ghi địa chỉ tăng lên và nội dung bộ đếm dữ liệu giảm xuống một đơn vị.
- Khi bộ đếm bằng dữ liệu bằng 0, DMAC gửi tín hiệu ngắt CPU để báo kết thúc DMA

6.2 Các phương pháp điều khiển vào ra



6.2 Các phương pháp điều khiển vào ra

Các kiểu thực hiện DMA

- DMA truyền theo khối: DMAC sử dụng BUS để truyền cả khối dữ liệu (CPU chuyển nhượng BUS cho DMAC)
- DMA lấy chu kỳ: DMAC cưỡng bức CPU treo tạm thời từng chu kỳ BUS để thực hiện truyền một từ dữ liệu
- DMA trong suốt: DMAC nhận biết những chu kỳ nào CPU không sử dụng BUS thì chiếm BUS để trao đổi dữ liệu (DMAC lấy lên chu kỳ)

Đặc điểm DMA

- CPU không tham gia trong quá trình trao đổi dữ liệu
- DMAC điều khiển trao đổi dữ liệu giữa bộ nhớ chính và Module vào ra với tốc độ nhanh.
- Phù hợp với yêu cầu trao đổi mảng dữ liệu có kích thước lớn.

6.2 Các phương pháp điều khiển vào ra

Bộ xử lý vào ra

- Việc điều khiển vào ra được sử dụng bởi một bộ điều khiển vào ra chuyên dụng.
- Bộ xử lý vào ra hoạt động theo chương trình riêng của nó.
- Chương trình của bộ xử lý vào ra có thể nằm trong bộ nhớ chính hoặc bộ nhớ riêng.
- Hoạt động theo kiến trúc đa xử lý

6.3 Nối ghép thiết bị ngoại vi

Nối ghép thiết bị ngoại vi

Các kiểu nối ghép vào ra

- Nối ghép song song
- Nối ghép nối tiếp

Nối ghép song song

- ✓ Truyền các bit song song
- ✓ Tốc độ truyền nhanh
- ✓ Cần đường truyền song song
- ✓ Tốn nhiều dây dẫn

6.3 Nối ghép thiết bị ngoại vi

Nối ghép nối tiếp

- Truyền lần lượt từng bit
- Cần có bộ chuyển đổi từ song song sang nối tiếp
- Tốc độ chậm
- Cần ít đường truyền dữ liệu

Các cấu hình ghép nối

- ✓ Điểm tới điểm (point to point): Qua một cổng vào ra chỉ có thể ghép một thiết bị ngoại vi (PS/2, COM, LPT,...)
- ✓ Điểm tới đa điểm (Point to multipoint): Thông qua một cổng vào ra ghép nhiều thiết bị vào ra. Ví dụ: SCSI(7,15), USB (127),...

6.4 Các cổng vào ra thông dụng

Các cổng vào ra thông dụng

- PS/2 : nối ghép bàn phím và chuột
- VGA(Video Graphic Adapter): Cổng nối ghép màn hình
- LPT (Line PrinTer): nối ghép với máy in là cổng song song
- COM (COMMunication): nối ghép với Modem, chuột, và thiết bị khác. Cổng nối tiếp 9 hoặc 25 chân.
- USB: cổng nối tiếp đa năng cho phép nối ghép nối tiếp tối đa 17 thiết bị thông qua Hub.



THE END