



# KIẾN TRÚC MÁY TÍNH

## Computer Architecture

Th.s Nguyễn Hồng Nam  
[nhnam@cse.hcmut.edu.vn](mailto:nhnam@cse.hcmut.edu.vn)

# Chương 4

## *Bộ nhớ*

# Nội dung

- ❖ 1. Khái niệm về tổ chức thứ bậc của bộ nhớ
- ❖ 2. Bộ nhớ chính
- ❖ 3. Bộ nhớ Cache
- ❖ 4. Bộ nhớ ngoài
- ❖ 5. Bộ nhớ ảo

# Các đặc trưng của hệ thống nhớ

## ❖ Vị trí

- Bên trong CPU:
  - tập thanh ghi
- Bộ nhớ trong
  - bộ nhớ chính
  - bộ nhớ cache
- Bộ nhớ ngoài: các thiết bị nhớ

## ❖ Dung lượng

- Độ dài từ nhớ (tính bằng bit: 16, 32 bit)
- Số lượng từ nhớ

# Các đặc trưng của hệ thống nhớ(tt)

## ❖ **Đơn vị truyền**

- Từ nhớ (word)
- Khối nhớ (block)

## ❖ **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)

# Các đặc trưng của hệ thống nhớ(tt)

## ❖ **Hiệu năng (performance)**

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

## ❖ **Kiểu vật lý**

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

# Các đặc trưng của hệ thống nhớ(tt)

## ❖ Các đặc tính vật lý

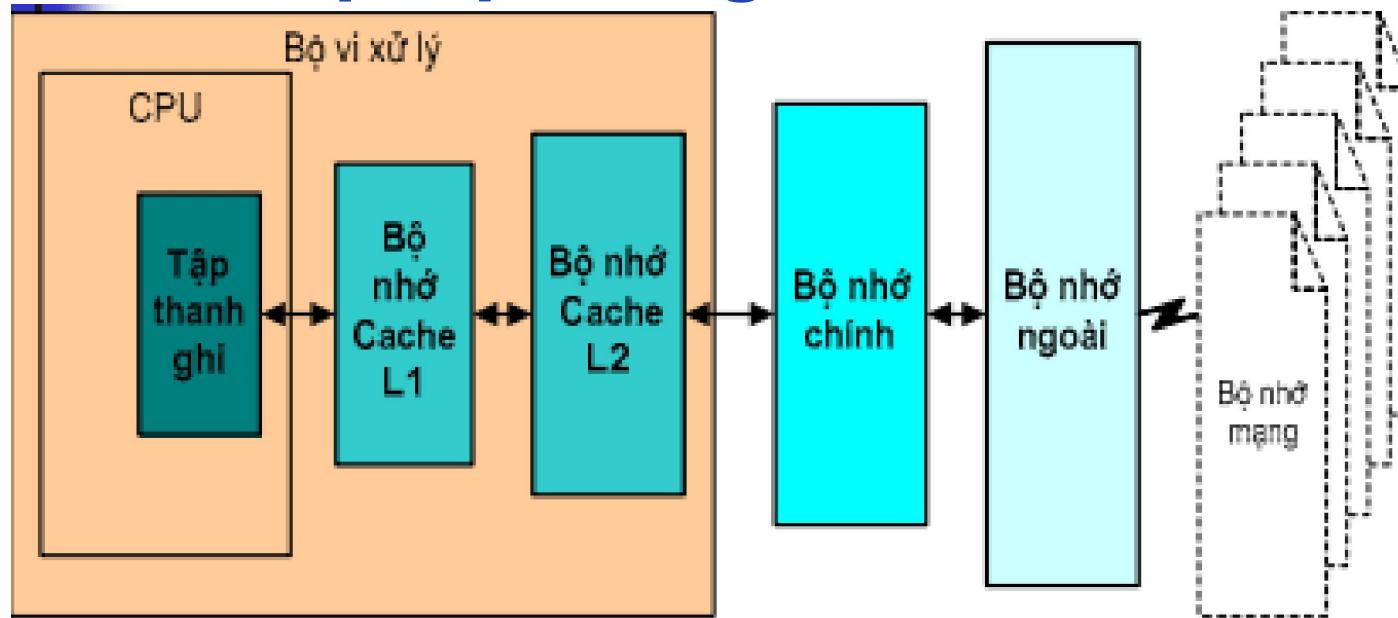
- Khả biến / Không khả biến (volatile / nonvolatile)

## ❖ Xoá được / không xoá được

- Tổ chức

# 1. Khái niệm về tổ chức thứ bậc của bộ nhớ

## ❖ Phân cấp hệ thống nhớ



Từ trái sang phải:

- dung lượng tăng dần
- tốc độ giảm dần
- giá thành/1bit giảm dần

# 1. Khái niệm về tổ chức thứ bậc của bộ nhớ

## ❖ Phân cấp hệ thống nhớ (tt)

Dung lượng

Thời gian truy cập

Giá tiền / đơn vị

CPU Registers

100s Bytes

1s ns

Cache

K Bytes

4 ns

1-0.1 cents/bit

Main Memory

M Bytes

100ns- 300ns

\$ .0001-.00001 cents /bit

Disk

G Bytes, 5 ms

(5,000,000 ns)

-5 -6

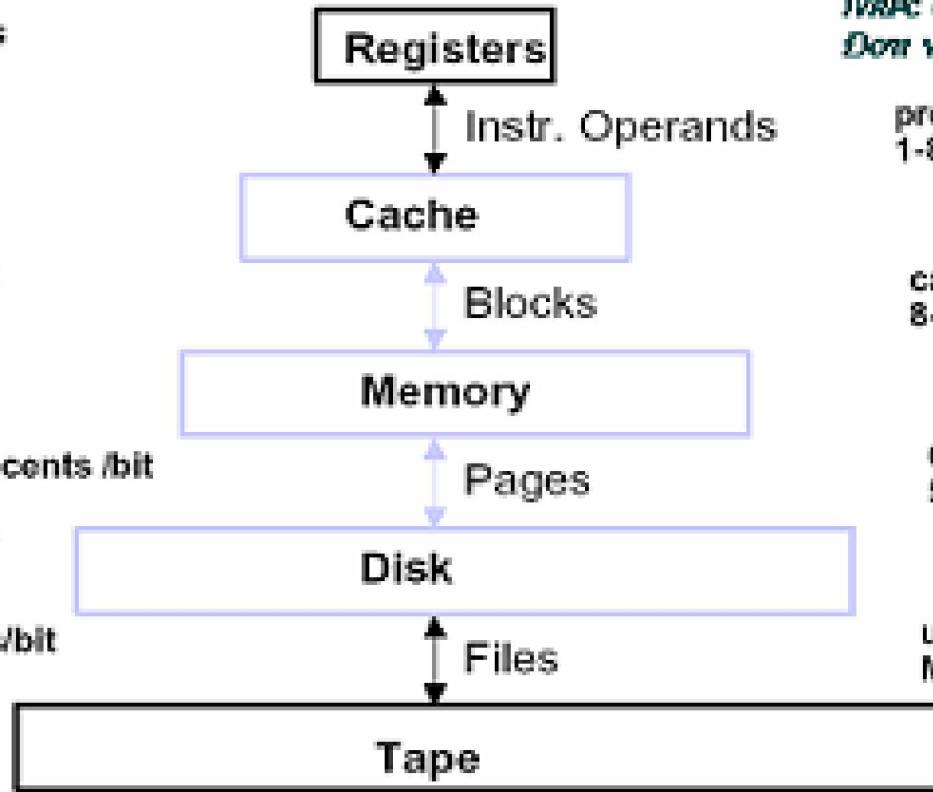
10<sup>-5</sup> - 10<sup>-6</sup> cents/bit

Tape

Infinite

sec/min

10



Mức cao

Mức quản lý

Đơn vị tham chiếu

prog./compiler  
1-8 bytes

cache ctrl  
8-128 bytes

OS  
512-4K bytes

user/operator  
Mbytes

Nhanh hơn

Làm chậm

Mức thấp

# 1. Khái niệm về tổ chức thứ bậc của bộ nhớ

## ❖ Các tham số trong phân cấp bộ nhớ(tt)

|               | Access type | Capacity      | Latency      | Bandwidth         | Cost/MB |
|---------------|-------------|---------------|--------------|-------------------|---------|
| CPU registers | Random      | 64–1024 bytes | 1–10 ns      | System clock rate | High    |
| Cache memory  | Random      | 8–512 KB      | 15–20 ns     | 10–20 MB/s        | \$500   |
| Main memory   | Random      | 16–512 MB     | 30–50 ns     | 1–2 MB/s          | \$20–50 |
| Disk memory   | Direct      | 1–20 GB       | 10–30 ms     | 1–2 MB/s          | \$0.25  |
| Tape memory   | Sequential  | 1–20 TB       | 30–10,000 ms | 1–2 MB/s          | \$0.025 |

# 1. Khái niệm về tổ chức thứ bậc của bộ nhớ

## ❖ Các thành phần phân phân cấp bộ nhớ

### ▪ Thanh ghi của CPU

- Kích thước rất nhỏ (vài chục byte tới vài KB)
- Tốc độ rất nhanh, thời gian truy cập khoảng 0.25 ns
- Giá thành đắt
- Lưu trữ tạm thời dữ liệu đầu vào và ra cho các lệnh

# 1. Khái niệm về tổ chức thứ bậc của bộ nhớ

## ❖ Các thành phần phân cấp bộ nhớ(tt)

### ▪ Cache

- Kích thước nhỏ (64KB tới 16MB)
- Tốc độ nhanh, thời gian truy cập khoảng 1 – 5ns
- Giá thành đắt
- Lưu trữ lệnh và dữ liệu cho CPU
- Còn được gọi là “bộ nhớ thông minh” (smart memory)

# 1. Khái niệm về tổ chức thứ bậc của bộ nhớ

## ❖ Các thành phần phân cấp bộ nhớ(tt)

### ▪ Bộ nhớ chính

- Kích thước lớn, dung lượng từ 256MB tới 4GB cho các hệ 32bits
- Tốc độ chậm, thời gian truy cập từ 50 – 70ns
- Lưu trữ lệnh và dữ liệu cho hệ thống và người dùng
- Giá thành rẻ

# 1. Khái niệm về tổ chức thứ bậc của bộ nhớ

## ❖ Các thành phần phân cấp bộ nhớ(tt)

### ■ Bộ nhớ phụ

- Kích thước rất lớn, dung lượng từ 20GB tới 1000GB
- Tốc độ rất chậm, thời gian truy cập khoảng 5ms
- Lưu trữ lượng dữ liệu lớn dưới dạng file trong thời gian lâu dài
- Giá thành rất rẻ

# 1. Khái niệm về tổ chức thứ bậc của bộ nhớ

## ❖ Vai trò của mô hình phân cấp

### ■ Nâng cao hiệu năng hệ thống

- Dung hòa được CPU có tốc độ cao với bộ nhớ chính và bộ
- Thời gian truy cập dữ liệu trung bình của CPU từ hệ thống bộ nhớ gần bằng thời gian truy cập cache

### ■ Giảm giá thành sản xuất

- Các thành phần đắt tiền sẽ được sử dụng với dung lượng nhỏ hơn
- Các thành phần rẻ hơn được sử dụng với dung lượng lớn hơn

# Phân loại bộ nhớ

## ❖ Dựa vào kiểu truy cập

- Bộ nhớ truy cập ngẫu nhiên (RAM: Random Access Memory)
- Bộ nhớ truy cập tuần tự (SAM: Serial Access Memory)
- Bộ nhớ chỉ đọc (ROM: Read Only Memory)

# Phân loại bộ nhớ(tt)

## ❖ Dựa vào khả năng chịu đựng/ lưu giữ thông tin

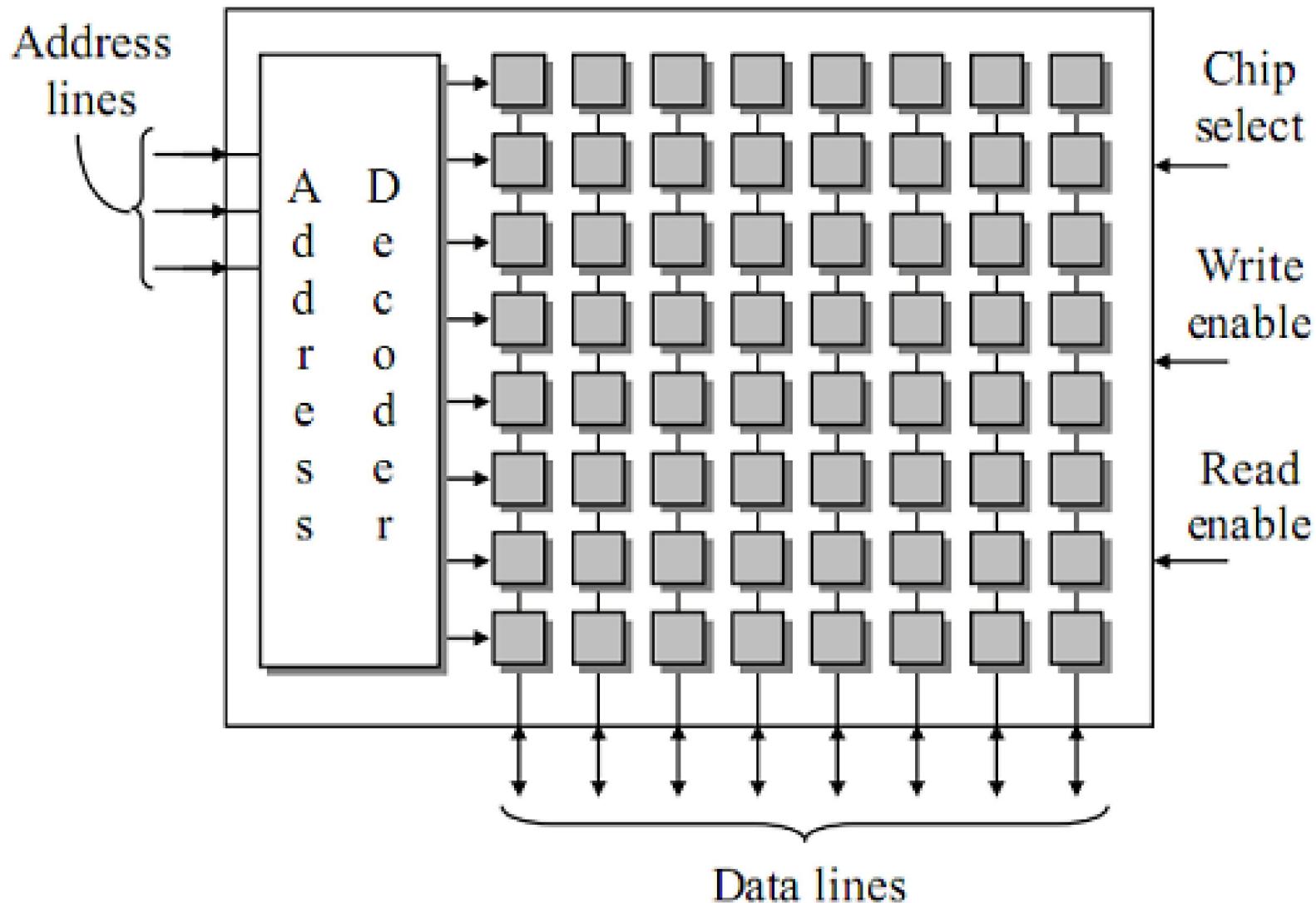
- Bộ nhớ không ổn định (volatile memory): thông tin lưu trữ bị mất khi tắt nguồn
- Bộ nhớ ổn định: thông tin lưu trữ được giữ lại khi tắt nguồn

# Phân loại bộ nhớ(tt)

## ❖ Dựa vào công nghệ chế tạo

- Bộ nhớ bán dẫn: ROM, RAM
- Bộ nhớ từ: HDD, FDD, tape
- Bộ nhớ quang: CD, DVD

# Tổ chức mạch nhớ



# Tổ chức của thiết bị nhớ

## ❖ Address lines:

- Các đường địa chỉ nối tới bus A
- Truyền tín hiệu địa chỉ từ CPU tới mạch nhớ

## ❖ Address decoder:

- Bộ giải mã địa chỉ
- Sử dụng địa chỉ để chọn ra và kích hoạt ô nhớ/dòng nhớ cần truy nhập

## ❖ Data lines

- Các đường dữ liệu kết nối với bus D
- Truyền dữ liệu từ bộ nhớ về CPU và ngược lại

# Tổ chức của thiết bị nhớ(tt)

## ❖ Chip select CS:

- Chân tín hiệu chọn chip
- Chip nhớ được kích hoạt khi CS=0. Thông thường CPU chỉ làm việc với 1 chip nhớ tại 1 thời điểm

## ❖ Write enable WE:

- Chân tín hiệu cho phép ghi
- Cho phép ghi vào đường nhớ khi WE = 0

## ❖ Read enable RE

- Chân tín hiệu cho phép đọc
- Cho phép đọc dữ liệu từ đường nhớ khi RE =0

# Băng thông của bộ nhớ

- ❖ Số dữ liệu được truy xuất từ bộ nhớ trong thời gian 1 giây.

Băng thông = dữ liệu 1 lần truy cập / thời gian một chu kỳ bộ nhớ

- ❖ Để tăng băng thông:
  - Giảm thời gian của một chu kỳ truy xuất
  - Chia bộ nhớ thành nhiều hộc (bank), mà mỗi hộc có bộ điều khiển đọc/ghi riêng biệt.

# Tốc độ của bộ nhớ phân cấp

- ❖ Hệ thống có bộ nhớ phân thành 2 cấp.
  - Cấp 1 có thời gian truy xuất  $T_1 = 0.01 \mu\text{s}$ .
  - Cấp 2 có thời gian truy xuất  $T_2 = 0.1 \mu\text{s}$ .
- ❖ Một word phải được chuyển đến Cấp 1 để có thể truy xuất bởi CPU.
  - Một “hit”: là khi CPU tìm thấy word cần tìm trong bộ nhớ cấp 1.
  - Một “miss”: ngược lại (khi đó phải truy xuất bộ nhớ cấp 2).
  - $T_{tb} = [T_1 * \text{hit\_rate} + (T_1+T_2) * \text{miss\_rate}] / \text{số lần truy cập.}$
  - Hit + Miss = số lần truy cập.

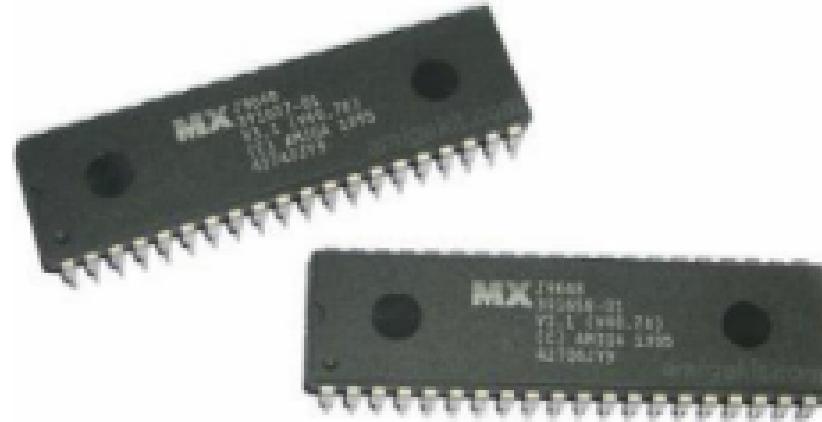
## 2. Bộ nhớ chính

### ❖ Giới thiệu ROM và RAM

- ROM và RAM là một trong những bộ nhớ chính của máy tính dùng lưu trữ các chương trình và dữ liệu trong suốt quá trình hoạt động
- ROM và RAM được tạo ra với nhiều chủng loại khác nhau, đáp ứng mọi nhu cầu sử dụng của người dùng
- ROM (Read-Only Memory – bộ nhớ chỉ đọc)
- RAM(Random Access Memory – bộ nhớ truy xuất ngẫu nhiên)

## 2. Bộ nhớ chính

### ❖ Bộ nhớ ROM



### ❖ Bộ nhớ RAM



## 2. Bộ nhớ chính

### ❖ ROM (Read Only Memory)

- Là bộ nhớ chỉ đọc
- Ghi thông tin vào ROM bằng cách sử dụng các thiết bị hoặc phương pháp đặc biệt
- ROM là bộ nhớ ổn định: tất cả thông tin vẫn được duy trì khi mất nguồn nuôi
- Là bộ nhớ bán dẫn: mỗi ô nhớ là một cổng bán dẫn
- Thường dùng để lưu trữ thông tin hệ thống: thông tin phần cứng và BIOS

## 2. Bộ nhớ chính

### ❖ ROM (Read Only Memory) (tt)

- Lưu trữ các thông tin sau:
  - Thư viện các chương trình con
  - Các chương trình điều khiển hệ thống (BIOS)
  - Các bảng chức năng
  - Vi chương trình

## 2. Bộ nhớ bán dẫn

### ❖ Các loại ROM

- ROM nguyên thủy (Ordinary ROM)
  - ROM các thế hệ đầu tiên
  - Sử dụng tia cực tím để ghi thông tin
- PROM (Programmable ROM)
  - ROM có thể lập trình
  - Thông tin có thể được ghi vào PROM nhờ thiết bị đặc biệt gọi là bộ lập trình PROM

## 2. Bộ nhớ bán dẫn

### ❖ Các loại ROM (tt)

- **EPROM (Erasable programmable read-only memory)**
  - Là ROM có thể lập trình và xóa được
  - Thông tin trong EPROM có thể xóa bằng cách chiếu các tia cực tím có cường độ cao
- **EEPROM**
  - Là EPROM nhưng nội dung có thể xóa bằng điện
  - Có thể ghi được thông tin sử dụng phần mềm chuyên dụng

## 2. Bộ nhớ bán dẫn

### ❖ Các loại ROM (tt)

#### ▪ Flash memory

- Là một dạng EEPROM nhưng có tốc độ đọc và ghi thông tin nhanh hơn
- Bộ nhớ flash chỉ có thể đọc/ ghi thông tin theo khối

## 2. Bộ nhớ bán dẫn

### ❖ RAM (Random Access Memory)

- Là bộ nhớ truy cập ngẫu nhiên (*mỗi ô nhớ được truy cập ngẫu nhiên, bằng tốc độ*)
- Bộ nhớ đọc-ghi (Read/Write Memory)
- Là bộ nhớ không ổn định (dễ bay hơi): mọi thông tin lưu trữ sẽ bị mất khi tắt nguồn
- Là bộ nhớ bán dẫn. Mỗi ô nhớ là một cổng bán dẫn

## 2. Bộ nhớ bán dẫn

### ❖ RAM (Random Access Memory) (tt)

- Sử dụng để lưu trữ thông tin hệ thống và của người dùng
  - Thông tin hệ thống: thông tin phần cứng & hệ điều hành
  - Thông tin người dùng: mã lệnh và dữ liệu các chương trình ứng dụng

## 2. Bộ nhớ bán dẫn

### ❖ Các loại RAM

- RAM tĩnh (SRAM)
- RAM động (DRAM)

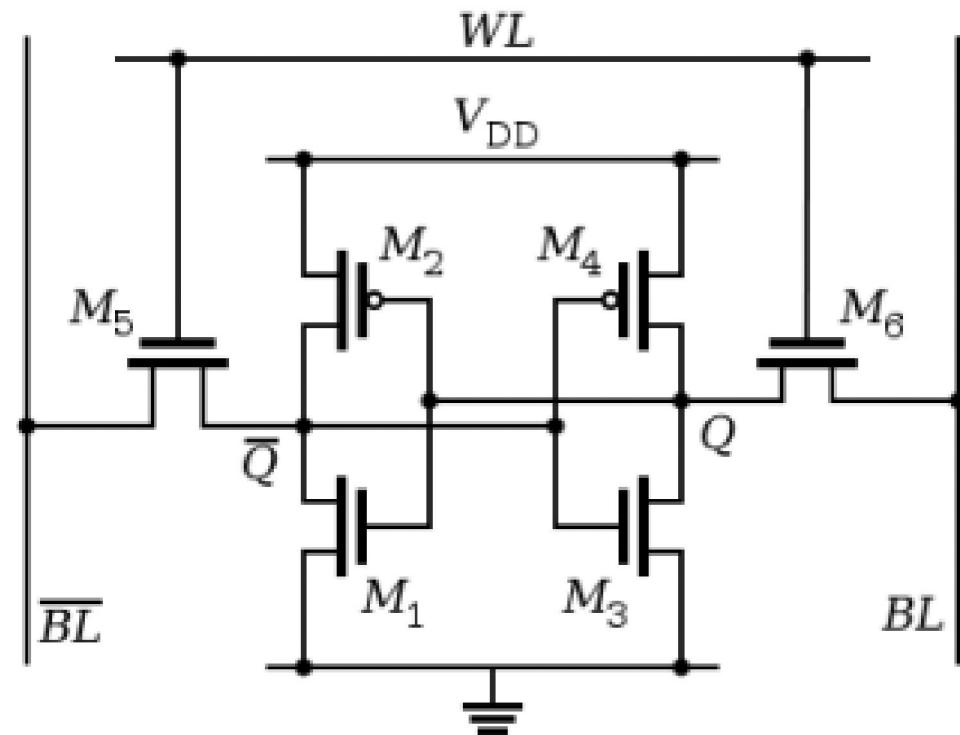
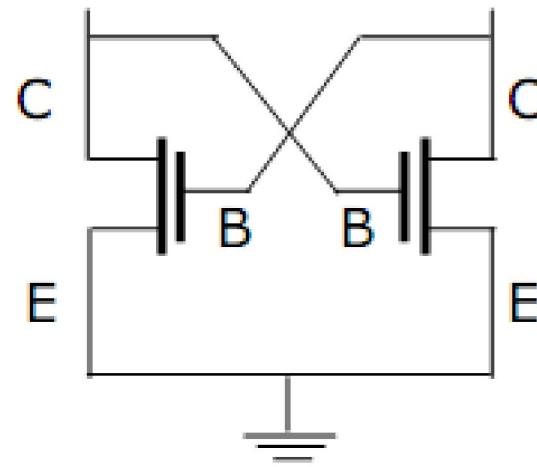
## 2. Bộ nhớ bán dẫn

### ❖ **SRAM (Static) – RAM tĩnh**

- Mỗi bit của SRAM là một mạch lật flip-flop
- Thông tin lưu trong các bit SRAM luôn ổn định, không cần phải làm tươi định kỳ
- SRAM nhanh nhưng đắt hơn DRAM

## 2. Bộ nhớ bán dẫn

### ❖ Cấu tạo của SRAM



Một mạch lật đơn giản

Một ô nhớ SRAM loại 6T

## 2. Bộ nhớ bán dẫn

### ❖ Đặc điểm của SRAM

- SRAM sử dụng mạch lật trigor lưu trữ ổn (bistable latching circuit) để lưu 1 bit thông tin
- Mỗi mạch lật lưu 1 bit thường sử dụng 6, 8, 10 transistor (gọi là mạch 6T, 8T, 10T)

## 2. Bộ nhớ bán dẫn

### ❖ Đặc điểm của SRAM(tt)

- SRAM thường có tốc độ truy cập nhanh vì:
  - Các bit có cấu trúc đối xứng
  - Các mạch nhớ SRAM chấp nhận tất cả các chân địa chỉ tại một thời điểm (không dồn kênh)
- SRAM đắt vì:
  - Nó sử dụng nhiều transistor cho một bit hơn DRAM
  - Vì cấu trúc bên trong phức tạp hơn, mật độ của SRAM thấp hơn DRAM

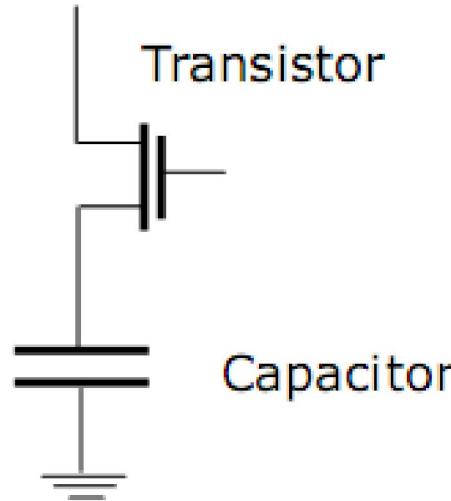
## 2. Bộ nhớ bán dẫn

### ❖ DRAM (Dynamic) – RAM động

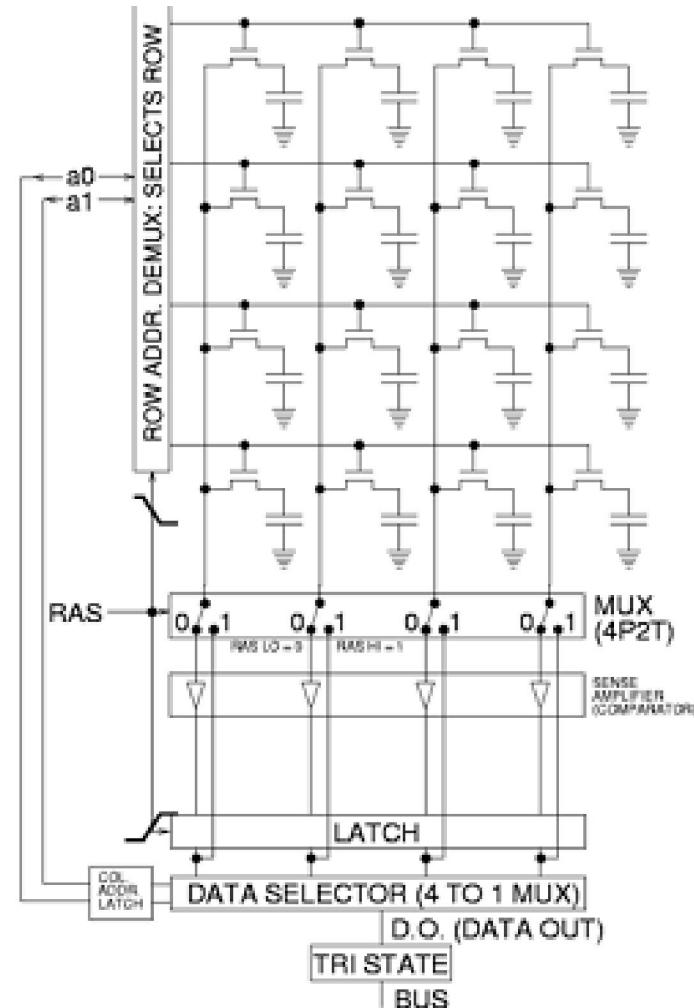
- Các bit được lưu trữ trên tụ điện
- Thông tin lưu trong bit DRAM không được ổn định, và phải được làm tươi định kỳ
- DRAM chậm hơn SRAM nhưng rẻ hơn

## 2. Bộ nhớ bán dẫn

### ❖ Cấu tạo của DRAM (Dynamic)



Một bit DRAM



## 2. Bộ nhớ bán dẫn

### ❖ Đặc điểm của DRAM

- Bit của DRAM dựa trên 1 tụ điện và 1 transistor
  - 2 mức tích điện của tụ sẽ biểu diễn 2 mức logic:
    - Không tích điện: mức 0
    - Tích đầy điện: mức 1

## 2. Bộ nhớ bán dẫn

### ❖ Đặc điểm của DRAM(tt)

- Do tụ thường tự phóng điện, điện tích trong tụ điện có xu hướng bị tổn hao
  - Cần nạp lại thông tin trong tụ thường xuyên để tránh mất thông tin
  - Việc nạp lại thông tin cho tụ là quá trình làm tươi (refresh), phải theo định kỳ
- Các bit nhớ của DRAM thường được sắp xếp thành ma trận:
  - Một tụ + một transistor -> một bit
  - Các bit được tập hợp thành các cột và dòng

## 2. Bộ nhớ bán dẫn

### ❖ Đặc điểm của DRAM(tt)

- DRAM chậm hơn SRAM vì:
  - Cần làm tươi định kỳ
  - Quá trình nạp điện tụ mất thời gian
  - Các mạch DRAM thường dùng kỹ thuật dồn kênh (địa chỉ cột/ hàng) để tiết kiệm đường địa chỉ
- DRAM rẻ hơn SRAM vì:
  - Cấu trúc đơn giản, sử dụng ít transistor
  - Mật độ cấy cao hơn

## 2. Bộ nhớ bán dẫn

### ❖ Các loại DRAM

- SDRAM: Synchronous DRAM
- SRD SDRAM: (Single Data Rate SDRAM) chấp nhận 1 thao tác đọc/ ghi và chuyển 1 từ dữ liệu trong 1 chu kỳ đồng hồ; tốc độ 100MHz, 133MHz

## 2. Bộ nhớ bán dẫn

### ❖ Các loại DRAM (tt)

- DDR SDRAM: Double Data Rate SDRAM
  - DDR1 SDRAM: DDR 266, 333, 400: có khả năng chuyển 2 từ dữ liệu trong 1 chu kỳ
  - DDR2 SDRAM: DDR2 400, 533, 800: có khả năng chuyển 4 từ dữ liệu trong 1 chu kỳ
  - DDR3 SDRAM: DDR3 800, 1066, 1333, 1600: có khả năng chuyển 8 từ dữ liệu trong 1 chu kỳ

## 2. Bộ nhớ bán dẫn

### ❖ SDR SDRAM (Single Data Rate Synchronous Dynamic RAM):

- có tốc độ Bus từ 66/100/133MHz, tổng số chân của 2 mặt là 168 chân với tốc độ rộng dữ liệu là 64 bit, điện áp hoạt động là 3.3V và giao tiếp theo dạng DIMM



## 2. Bộ nhớ bán dẫn

### ❖ DDR SDRAM (Double Data Rate Synchronous Dynamic RAM – DDRAM) :

- có tốc độ Bus từ 200/266/333/400MHz, tổng số chân của 2 mặt là 184 chân với tốc độ rộng dữ liệu là 64 bit, điện áp hoạt động là 2.5V và giao tiếp theo dạng DIMM



## 2. Bộ nhớ bán dẫn

### ❖ DDR II (Double Data Rate II Synchronous Dynamic RAM) :

- có tốc độ Bus từ 533/667/800/1066MHz, tổng số chân của 2 mặt là 240 chân, điện áp hoạt động là 1.8V và giao tiếp theo dạng DIMM



## 2. Bộ nhớ bán dẫn

### ❖ DDR III (Double Data Rate III Synchronous Dynamic RAM) :

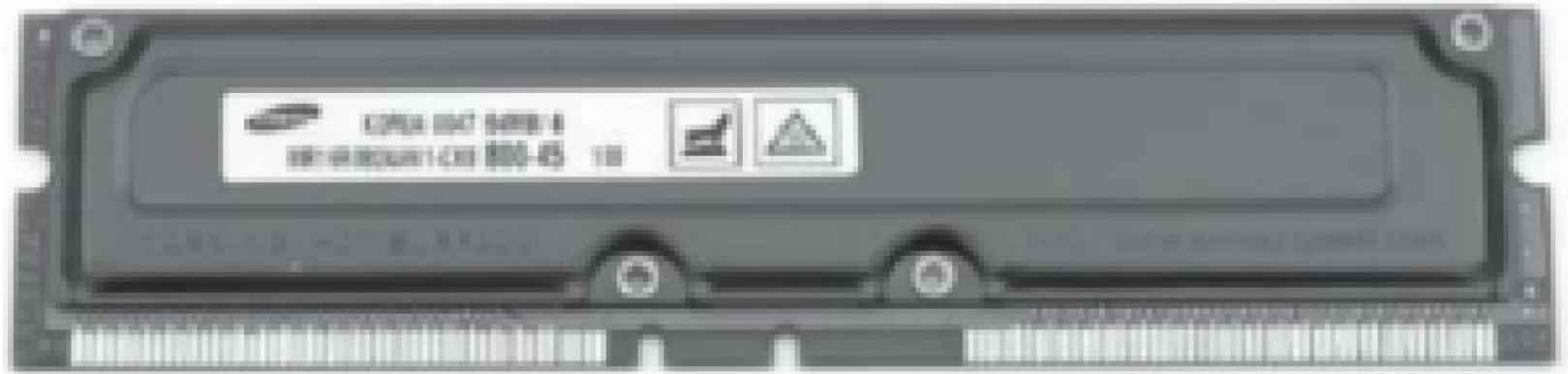
- có tốc độ Bus từ 800/1066/1333/1600MHz, tổng số chân của 2 mặt là 240 chân, điện áp hoạt động là 1.5V và giao tiếp theo dạng DIMM



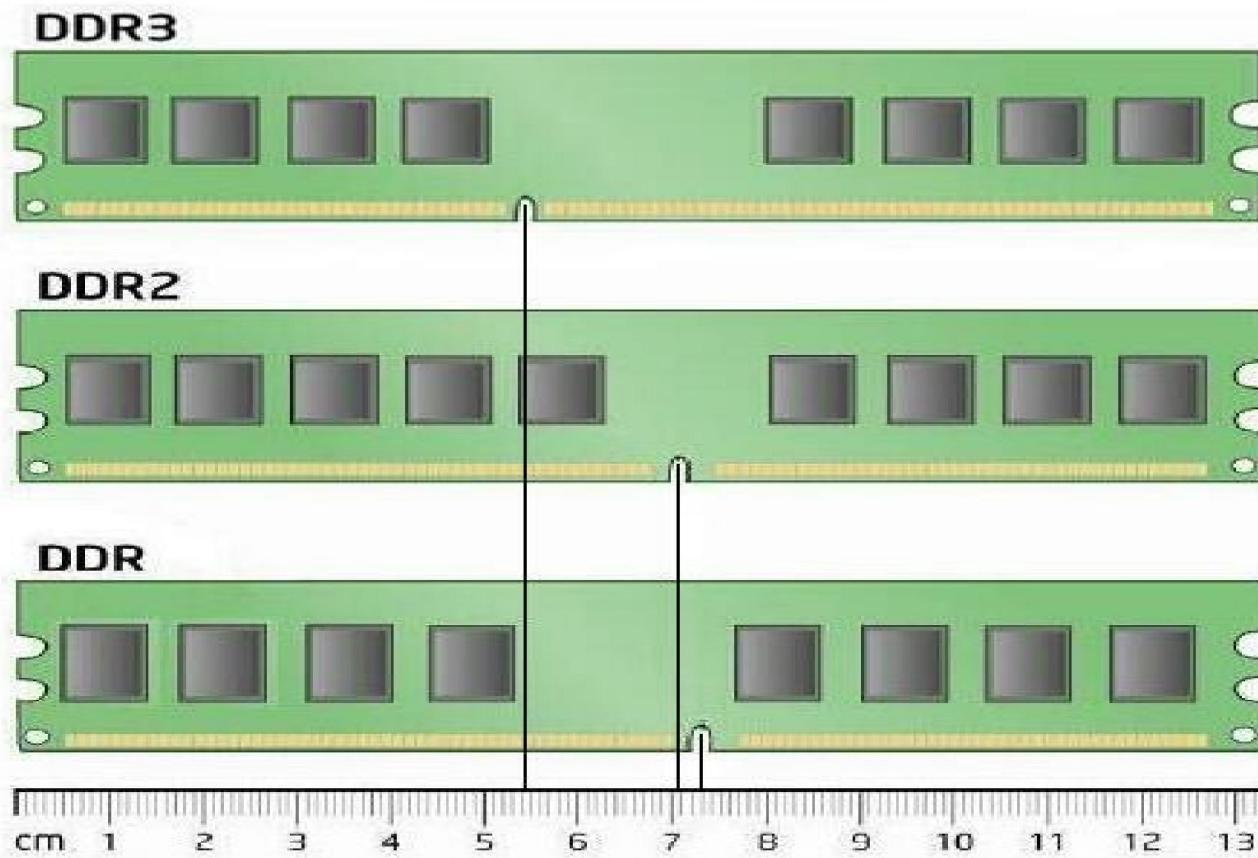
## 2. Bộ nhớ bán dẫn

### ❖ RDRAM (RAM Bus DRAM) :

- có tốc độ Bus từ 600/700/800/1066MHz, tổng số chân của 2 mặt là 184 chân, điện áp hoạt động là 2.5V và giao tiếp theo dạng RIMM

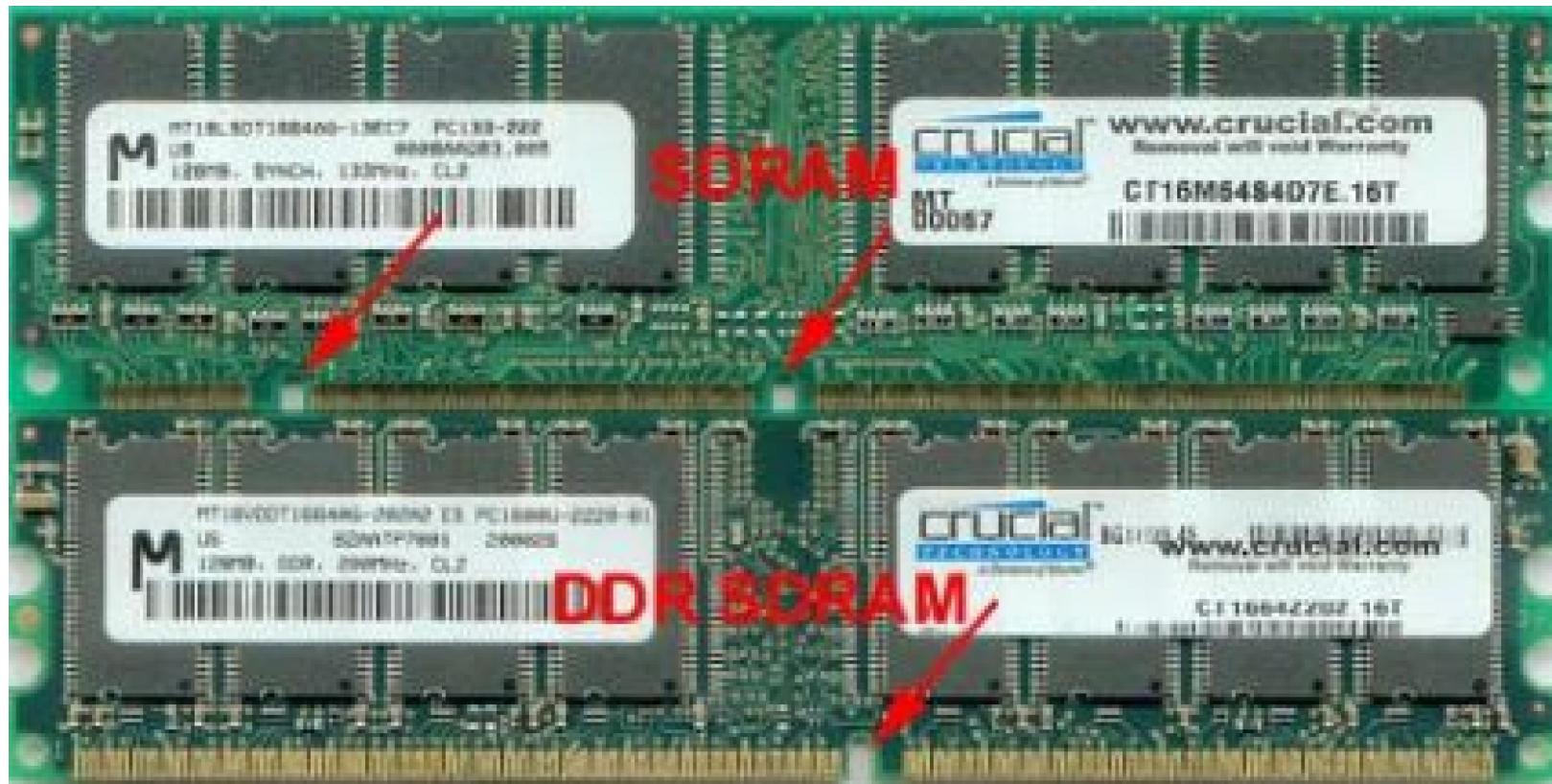


# Phân biệt DDR, DDR II và DDR III



Theo trang : <http://www.bixnet.com/>

# Phân biệt SDRAM, DDR



Theo trang : <http://www.tomshardware.com/>

# Tổng hợp các kiểu bộ nhớ BD

| Kiểu bộ nhớ                         | Loại            | Cơ chế xoá            | Cơ chế ghi | Tính bay hơi |
|-------------------------------------|-----------------|-----------------------|------------|--------------|
| RAM                                 | đọc/ghi         | bằng điện, mức byte   | bằng điện  | Có           |
| ROM                                 |                 |                       | Mặt na     |              |
| Programmable ROM (PROM)             | chỉ đọc         | Không thể xoá         |            |              |
| Erasable PROM                       |                 | Tia cực tím, mức chip |            |              |
| Electrically Erasable PROM (EEPROM) | hầu hết chỉ đọc | bằng điện, mức byte   | bằng điện  | Không        |
| Flash Memory                        |                 | bằng điện, mức khôi   |            |              |

# Xác suất truy cập dữ liệu trong BNT

- ❖ Nguyên tắc được áp dụng cho việc thâm nhập dữ liệu: **nguyên tắc về không gian** và **nguyên tắc về thời gian**
  - **Nguyên tắc về thời gian:** cho biết các ô nhớ được hệ thống xử lý thâm nhập có khả năng sẽ được thâm nhập trong tương lai gần.
  - **Nguyên tắc về không gian:** cho biết, bộ xử lý thâm nhập vào một ô nhớ thì có nhiều khả năng thâm nhập vào ô nhớ có địa chỉ kế tiếp do các lệnh được sắp xếp thành chuỗi có thứ tự.

# 3. Bộ nhớ Cache

## ❖ **Cache** là thành phần nhớ trong sơ đồ phân cấp bộ nhớ máy tính

- Nó hoạt động như thành phần trung gian, trung chuyển dữ liệu từ bộ nhớ chính về CPU và ngược lại

## ❖ **Vị trí**

- Với các hệ thống cũ, cache thường nằm ngoài CPU
- Với các CPU mới, cache thường được tích hợp vào trong CPU

# 3. Bộ nhớ Cache

- ❖ **Dung lượng thường nhỏ**
  - Với các hệ thống cũ: 16K, 32K,..., 128K
  - Với các hệ thống mới: 256K, 512K, 1MB, 2MB, ...
- ❖ **Tốc độ truy nhập của cache nhanh hơn so với tốc độ bộ nhớ chính**
- ❖ **Giá thành cache (tính trên bit) thường đắt hơn so với bộ nhớ chính**
- ❖ **Với các hệ thống CPU mới, cache thường được chia thành nhiều mức:**
  - Mức 1: 16 – 32 KB có tốc độ rất cao
  - Mức 2: 1 -16MB có tốc độ khá cao

# 3. Bộ nhớ Cache

## ❖ Vai trò của Cache

- Nâng cao hiệu năng hệ thống:
  - Dung hòa giữa CPU có tốc độ cao và bộ nhớ chính tốc độ thấp (giảm số lượng truy cập trực tiếp của CPU vào bộ nhớ chính)
  - Thời gian trung bình CPU truy cập hệ thống bộ nhớ gần bằng thời gian truy cập cache
- Giảm giá thành sản xuất
  - Nếu 2 hệ thống có cùng hiệu năng thì hệ thống có cache sẽ rẻ hơn
  - Nếu 2 hệ thống cùng giá thành, hệ thống có cache sẽ nhanh hơn

# 3. Bộ nhớ Cache

## ❖ Các nguyên lý hoạt động của Cache

- Cache được coi là bộ nhớ thông minh:
  - Cache có khả năng đoán trước yêu cầu về lệnh và dữ liệu của CPU
  - Dữ liệu và lệnh cần thiết được chuyển trước từ bộ nhớ chính về cache -> CPU chỉ truy nhập cache -> giảm thời gian truy nhập bộ nhớ
- Cache hoạt động dựa trên 2 nguyên lý cơ bản:
  - Nguyên lý cục bộ/ lân cận về không gian (spatial locality)
  - Nguyên lý cục bộ/ lân cận về thời gian (temporal locality)

# 3. Bộ nhớ Cache

## ❖ Các nguyên lý hoạt động của Cache (tt)

- Cục bộ (lân cận) về không gian:
  - Nếu một vị trí bộ nhớ được truy cập, thì khả năng/ xác suất các vị trí gần đó được truy cập trong thời gian gần tới là cao
  - Áp dụng với các mục dữ liệu và các lệnh có thứ tự tuần tự theo chương trình
  - Hầu hết các lệnh trong chương trình có thứ tự tuần tự, do đó cache đọc một khối lệnh trong bộ nhớ, mà bao gồm cả các phần tử xung quanh vị trí phần tử hiện tại được truy cập

# 3. Bộ nhớ Cache

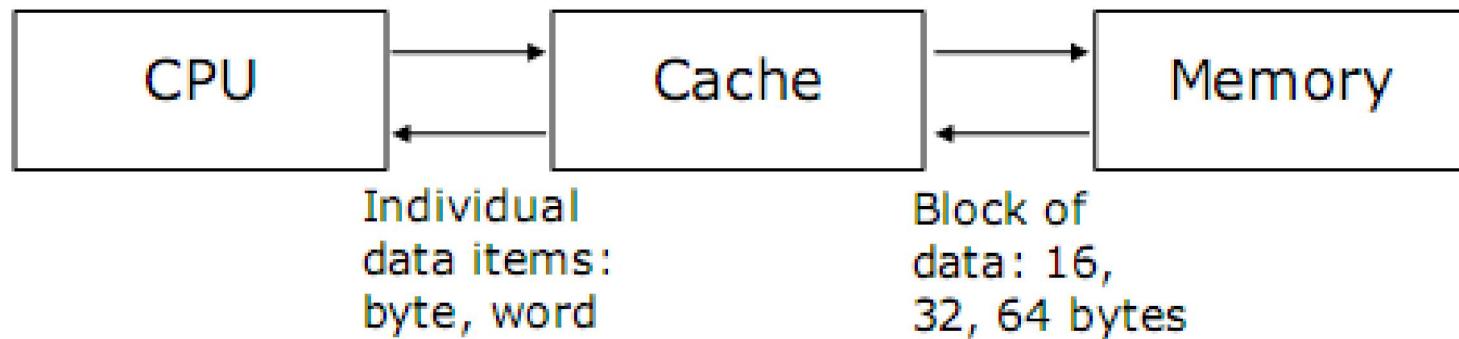
## ❖ Các nguyên lý hoạt động của Cache (tt)

- Cục bộ (lân cận) về thời gian:
  - Nếu một vị trí bộ nhớ được truy cập, thì khả năng nó sẽ được truy cập trong thời gian gần tới là cao
  - Áp dụng với các mục dữ liệu và các lệnh trong vòng lặp
  - Cache đọc khối dữ liệu trong bộ nhớ bao gồm tất cả các thành phần trong vòng lặp

# 3. Bộ nhớ Cache

## ❖ Trao đổi dữ liệu

- CPU đọc/ ghi từng mục dữ liệu riêng biệt từ/ vào cache
- Cache đọc/ ghi khối dữ liệu từ/ vào bộ nhớ



# 3. Bộ nhớ Cache

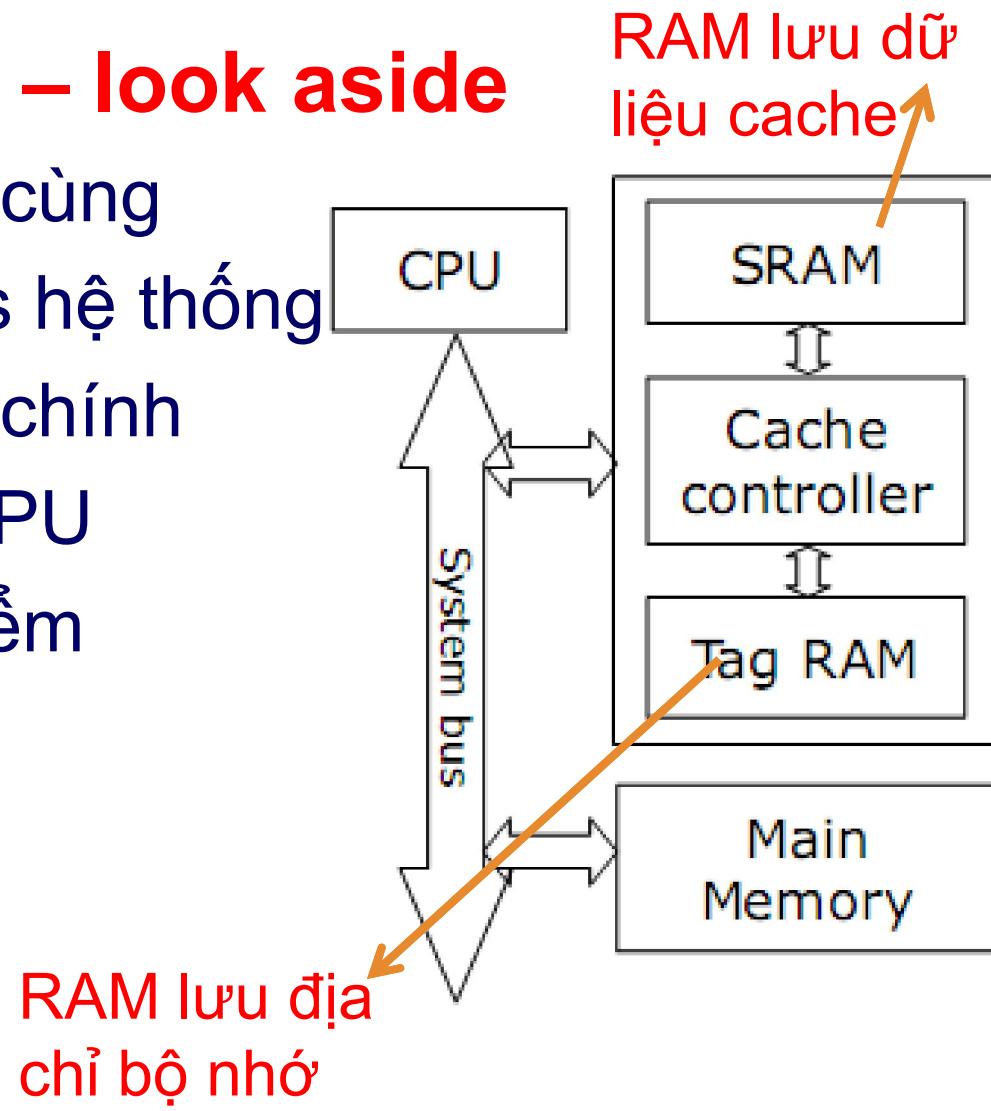
## ❖ Hệ số Hit và Miss của Cache

- Hit là sự kiện CPU truy cập tới mục dữ liệu/ mục tin mà tìm được trong cache
  - Xác suất xảy ra Hit được gọi là hệ số hit H
  - $0 \leq H \leq 1$
  - H càng cao thì cache càng tốt
- Miss là sự kiện CPU truy cập tới mục dữ liệu mà không tìm thấy nó trong cache
  - Khả năng xảy ra Miss gọi là hệ số miss hay  $1-H$
  - $0 \leq (1-H) \leq 1$
  - Hệ số miss thấp thì hiệu quả cache cao

# 3. Bộ nhớ Cache

## ❖ Kiến trúc Cache – look aside

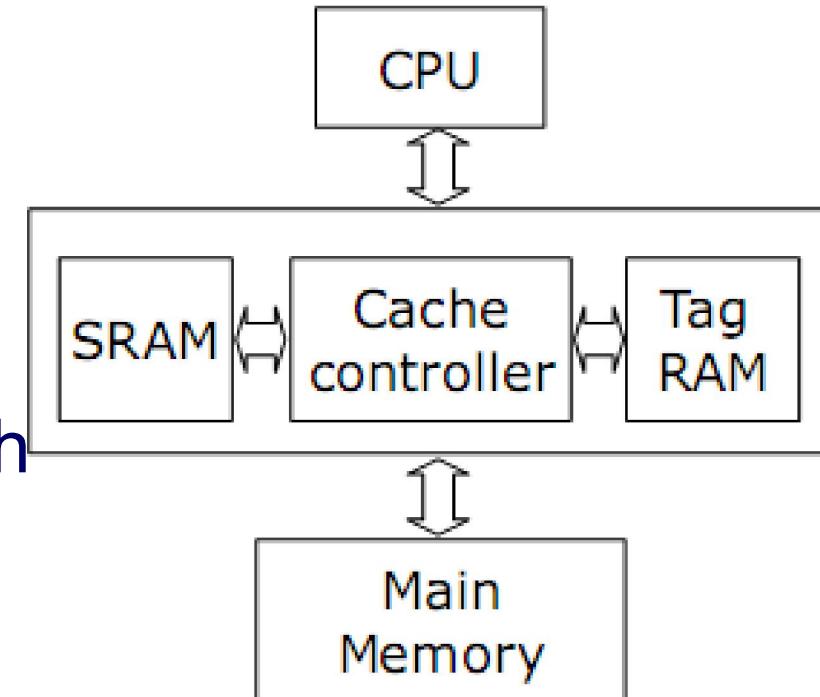
- Cache và bộ nhớ cùng được kết nối tới bus hệ thống
- Cache và bộ nhớ chính “thấy” chu kỳ bus CPU tại cùng một thời điểm
- Ưu:
  - Thiết kế đơn giản
  - Miss nhanh
- Nhược
  - Hit chậm



# 3. Bộ nhớ Cache

## ❖ Kiến trúc Cache – look through

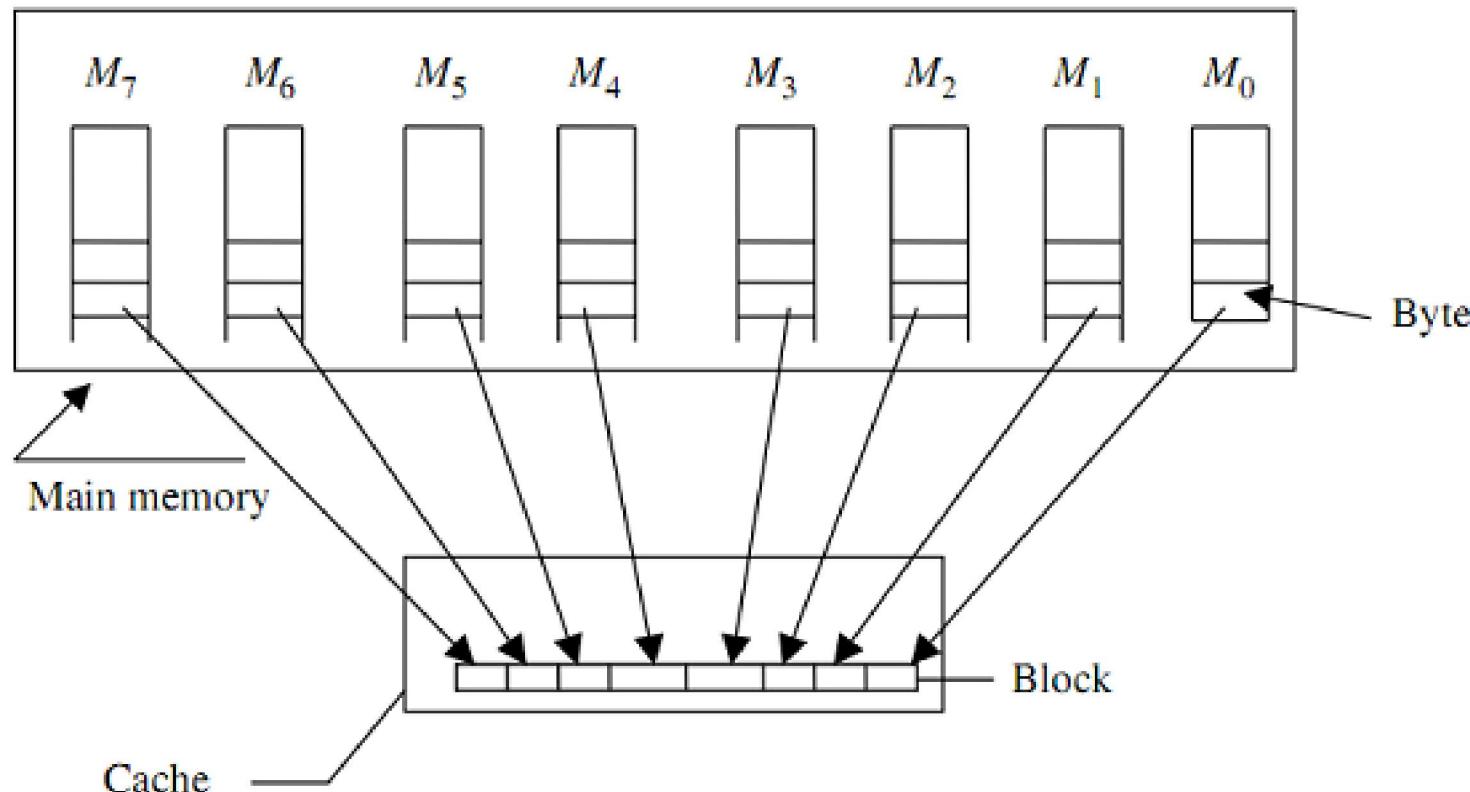
- Cache nằm giữa CPU và bộ nhớ chính
- Cache “thấy” chu kỳ bus CPU trước sau đó nó “truyền” lại cho bộ nhớ chính
- **Ưu:**
  - Hit nhanh
- **Nhược**
  - Thiết kế phức tạp
  - Đắt; Miss chậm



# 3. Bộ nhớ Cache

## ❖ Tổ chức Cache

- Tổ chức cache giải quyết vấn đề cache và bộ nhớ chính phối hợp làm việc với nhau như thế nào



# 3. Bộ nhớ Cache

## ❖ Các kỹ thuật tổ chức Cache

- Ánh xạ trực tiếp (direct mapping):
  - Đơn giản, nhanh
  - Ánh xạ cố định
- Ánh xạ kết hợp đầy đủ (fully associative mapping):
  - Phức tạp, chậm
  - Ánh xạ linh hoạt
- Ánh xạ tập kết hợp/ theo bộ (set): (set associative mapping)
  - Phức tạp
  - Nhanh, ánh xạ linh hoạt

# 3. Bộ nhớ Cache

## ❖ Ánh xạ trực tiếp

- Cache: Được chia thành n khối hoặc dòng (block or line), từ Line (0 tới n-1)
- Bộ nhớ:
  - Được chia thành m trang (page), từ page 0 tới page m-1
  - Mỗi trang bộ nhớ có kích thước bằng cache
  - Mỗi trang có n lines, từ Line 0 tới Line n-1

# 3. Bộ nhớ Cache

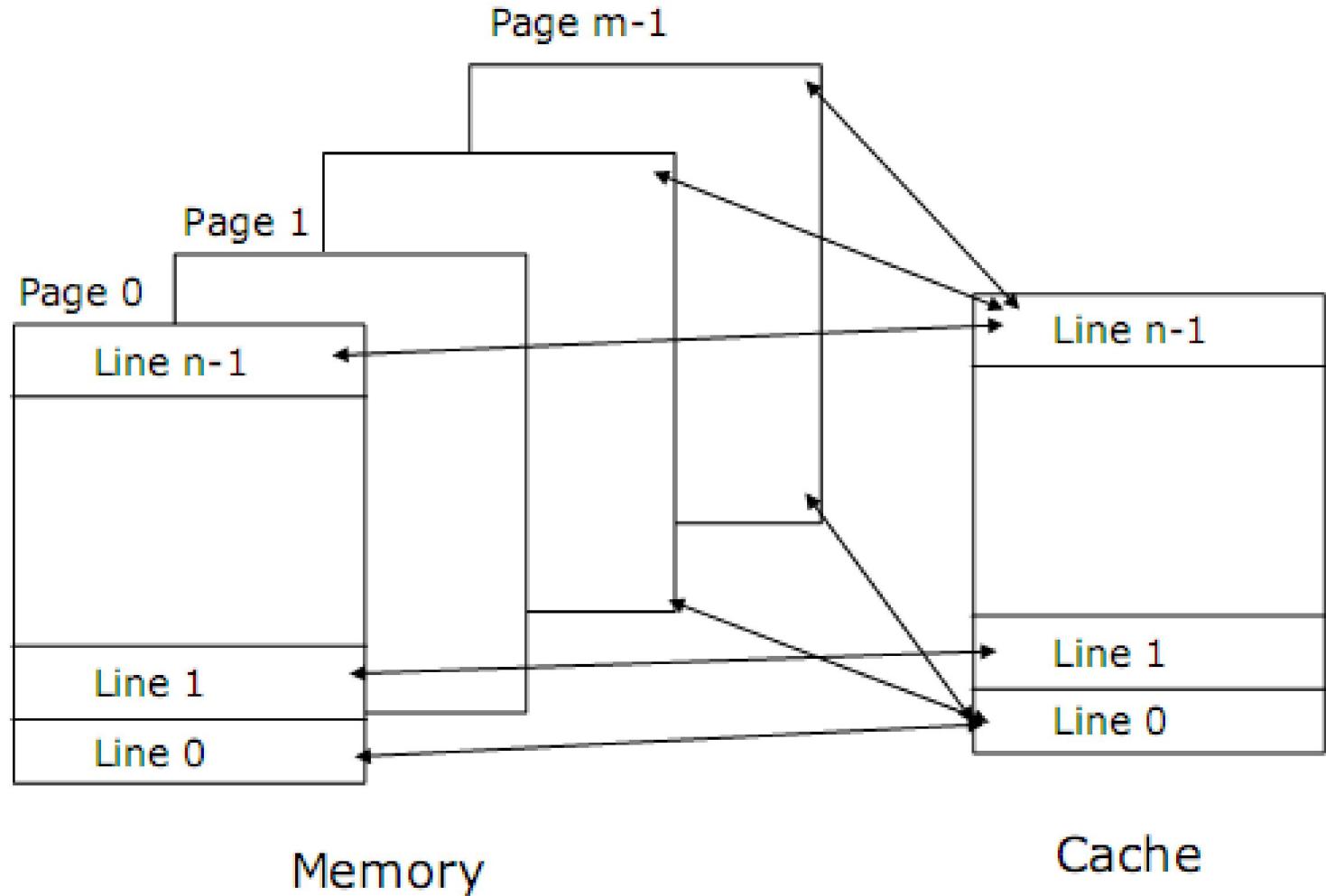
## ❖ Ánh xạ trực tiếp (tt)

### ▪ Ánh xạ:

- Line 0 của (page 0 tới page m-1) được ánh xạ tới Line 0 của cache
- Line 1 của (page 0 tới page m-1) được ánh xạ tới Line 1 của cache
- ....
- Line n-1 của (page 0 tới page m-1) được ánh xạ tới Line n-1 của cache

# 3. Bộ nhớ Cache

## ❖ Ánh xạ trực tiếp (tt)



# 3. Bộ nhớ Cache

## ❖ Địa chỉ ánh xạ trực tiếp

|     |      |      |
|-----|------|------|
| Tag | Line | Word |
|-----|------|------|

- Tag (bit): là địa chỉ của trang trong bộ nhớ
- Line (bit): là địa chỉ của line trong cache
- Word (bit): là địa chỉ của word trong line

# 3. Bộ nhớ Cache

## ❖ Địa chỉ ánh xạ trực tiếp (tt)

### ❖ Ví dụ:

- Đầu vào:
  - Kích thước bộ nhớ: 4GB
  - Kích thước cache: 1MB
  - Kích thước line: 32 byte
- Đầu ra
  - Kích thước line: 32 byte = 25 -> Word = 5 bit
  - Kích thước cache: 1 MB = 2<sup>20</sup> -> có  $2^{20} / 2^5 = 2^5$  lines -> Line = 5 bit
  - Tag = địa chỉ 32 bit – Line – Word = 32 – 5 – 5 = 22 bit

# 3. Bộ nhớ Cache

## ❖ Ánh xạ trực tiếp

### ▪ Ưu điểm :

- Thiết kế đơn giản
- Nhanh vì ánh xạ cố định: khi biết địa chỉ bộ nhớ có thể tìm nó trong cache rất nhanh

### ▪ Nhược điểm:

- Vì ánh xạ cố định nên khả năng xảy ra xung đột cao
- Tỷ lệ hit thấp

### 3. Bộ nhớ Cache

#### ❖ Ánh xạ kết hợp (liên kết) đầy đủ

- Cache: Được chia thành n khối hoặc dòng (block or line), từ Line 0 tới Line n-1
- Bộ nhớ:
  - Được chia thành m khối hay dòng, từ Line 0 tới Line m-1
  - Kích thước mỗi dòng cache bằng kích thước một dòng bộ nhớ
  - Số lượng dòng trong bộ nhớ có thể lớn hơn nhiều số lượng dòng của cache ( $m >> n$ )

### 3. Bộ nhớ Cache

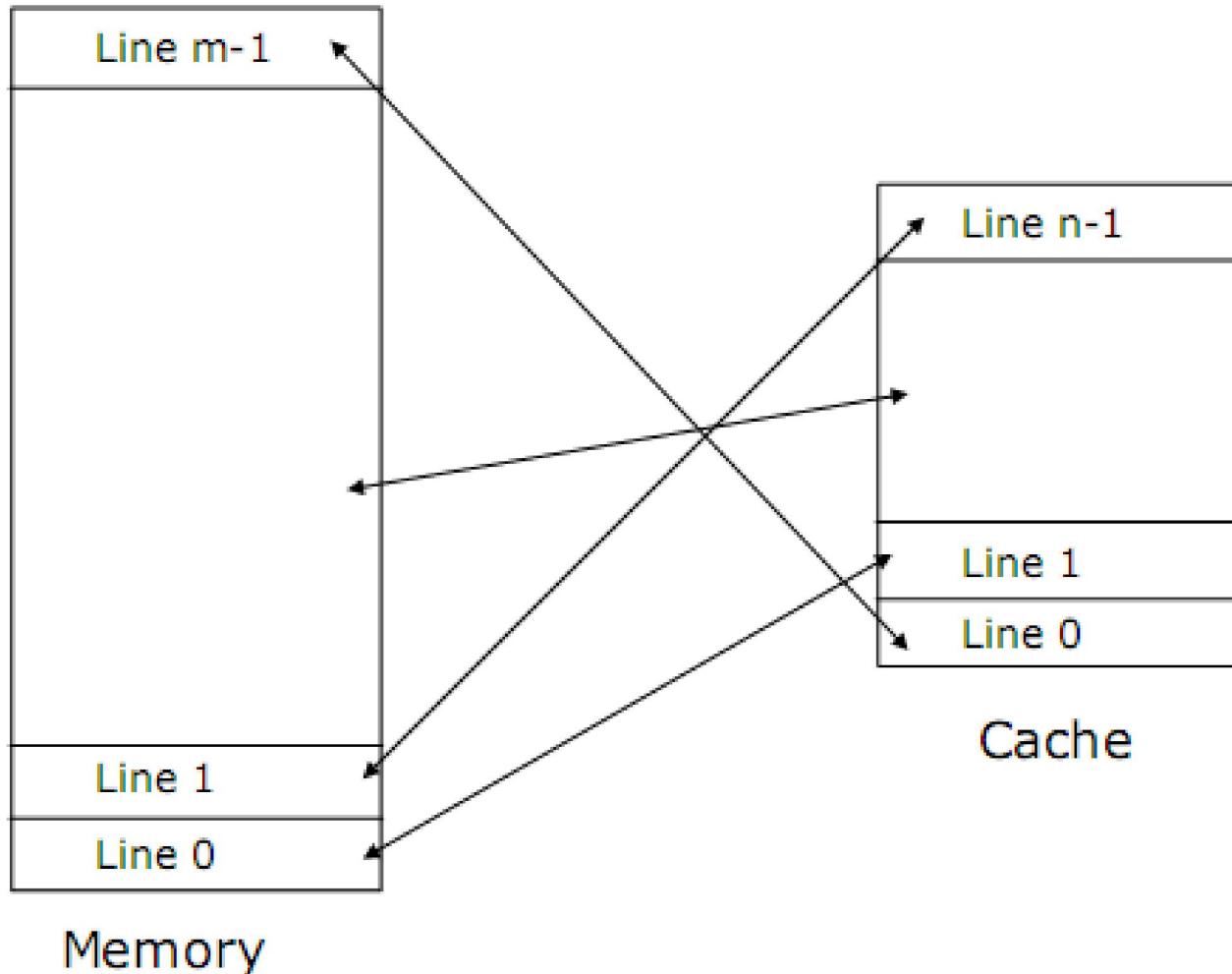
#### ❖ Ánh xạ kết hợp (liên kết) đầy đủ (tt)

- Ánh xạ :

- Một dòng của bộ nhớ có thể ánh xạ tới dòng bất kì của cache
- Line i của bộ nhớ có thể ánh xạ tới line j của cache

# 3. Bộ nhớ Cache

## ❖ Ánh xạ kết hợp (liên kết) đầy đủ (tt)



### 3. Bộ nhớ Cache

#### ❖ Địa chỉ ánh xạ kết hợp đầy đủ



- Tag (bit) là địa chỉ của line trong bộ nhớ (page =1)
- Word (bit) là địa chỉ của từ trong line

### 3. Bộ nhớ Cache

❖ **Địa chỉ ánh xạ kết hợp đầy đủ (tt)**

❖ **Ví dụ:**

▪ Input:

- Kích thước bộ nhớ = 4GB
- Kích thước cache = 1MB
- Kích thước line = 32 byte

▪ Output:

- Kích thước Line = 32 byte =  $2^5$  => Word = 5 bit
- Tag = 32 bit – word =  $32 - 5 = 27$  bit

# 3. Bộ nhớ Cache

## ❖ Ánh xạ kết hợp đầy đủ

- **Ưu:**
  - Ít xung đột vì ánh xạ linh hoạt
  - Tỉ lệ hit cao hơn
- **Nhược:**
  - Chậm vì phải tìm kiếm địa chỉ bộ nhớ trong cache
  - Phức tạp vì có thêm n bộ so sánh địa chỉ trong cache
- **Thường sử dụng cho cache có kích thước nhỏ**

# 3. Bộ nhớ Cache

## ❖ Ánh xạ tập kết hợp (liên kết theo nhóm)

- Cache:
  - Được chia thành k đường (ways) có kích thước bằng nhau
  - Mỗi đường được chia thành n dòng (block or line), từ Line 0 tới Line n-1
- Bộ nhớ:
  - Được chia thành m trang, từ page 0 tới page m-1
  - Kích thước trang page bằng kích thước way của cache
  - Mỗi trang có n line, từ Line 0 tới Line n-1

# 3. Bộ nhớ Cache

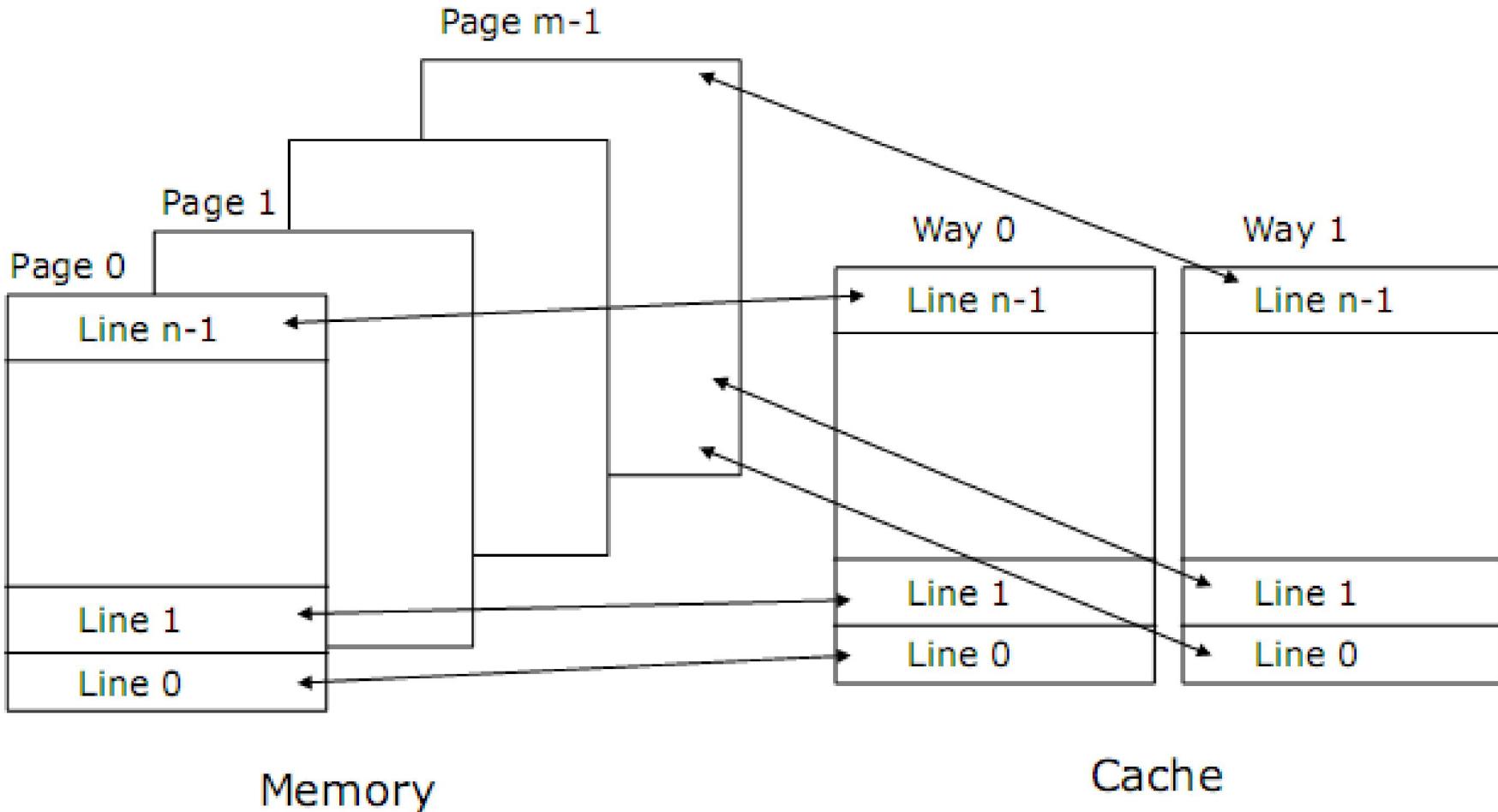
## ❖ Ánh xạ tập kết hợp

### ▪ Ánh xạ:

- Page được ánh xạ tới Way (ánh xạ linh hoạt):
  - Một page bộ nhớ có thể được ánh xạ tới way bất kì của cache
- Line của page ánh xạ tới line của way (ánh xạ cố định):
  - Line0 của page i được ánh xạ tới Line0 của way j;
  - Line1 của page I được ánh xạ tới Line1 của way j;
  - ...
  - Line n-1 của page I được ánh xạ tới Linen-1 của way j;

# 3. Bộ nhớ Cache

## ❖ Ánh xạ tập kết hợp (tt)



# 3. Bộ nhớ Cache

## ❖ Địa chỉ ánh xạ tập kết hợp

|     |     |      |
|-----|-----|------|
| Tag | Set | Word |
|-----|-----|------|

Tag (bit): là địa chỉ của trang trong bộ nhớ

Set (bit): là địa chỉ của line trong way của cache

Word (bit): là địa chỉ của word trong line

# 3. Bộ nhớ Cache

## ❖ Địa chỉ ánh xạ tập kết hợp (tt)

- Ví dụ:
  - Input:
    - Kích thước bộ nhớ: 4GB
    - Kích thước cache: 1MB, 2 ways
    - Kích thước Line: 32 byte
  - Output:
    - Line size = 32 byte =  $2^{5}$  □ Word = 5 bit
    - Cache size = 1MB =  $2^{21}$  □ có  $2^{21} / 2^5 / 2 = 24$  lines trong 1 way
    - Set = 4 bit
    - Tag = 32bit địa chỉ – Set – Word =  $32 - 4 - 5 = 23$  bit.

# 3. Bộ nhớ Cache

## ❖ Ánh xạ tập kết hợp

- Ưu:
  - Nhanh vì ánh xạ trực tiếp được sử dụng cho ánh xạ dòng
  - Ít xung đột vì ánh xạ từ trang nhớ tới đường của cache là linh hoạt
  - Tỷ lệ tìm thấy (hit) cao
- Nhược:
  - Thiết kế và điều khiển phức tạp vì cache được chia thành các way.

# 3. Bộ nhớ Cache

## ❖ Đọc/ ghi thông tin trong cache

- Thao tác ghi:

- Trường hợp tìm thấy (hit case):

- Write through (ghi thẳng): mục dữ liệu được ghi vào cache và ghi vào bộ nhớ đồng thời
- Write back (ghi trễ): mục dữ liệu trước tiên được ghi vào cache và cả dòng (block) chứa nó ở trong cache sẽ được ghi lại vào bộ nhớ sau đó, khi mà dòng đó bị thay thế

# 3. Bộ nhớ Cache

## ❖ Đọc/ ghi thông tin trong cache (tt)

- Thao tác ghi:

➤ Trường hợp không tìm thấy (miss case):

- Write allocate (ghi có đọc lại): mục dữ liệu trước hết được ghi vào bộ nhớ sau đó cả dòng chứa nó sẽ được đọc vào cache
- Write non-allocate (ghi không đọc lại): mục dữ liệu chỉ được ghi vào bộ nhớ

# 3. Bộ nhớ Cache

## ❖ **Các chính sách thay thế dòng cache**

- Tại sao phải thay thế dòng cache?
  - Ánh xạ dòng (bộ nhớ) -> dòng (cache) thường là ánh xạ nhiều -> một
  - Nhiều dòng bộ nhớ chia sẻ một dòng cache -> các dòng bộ nhớ được nạp vào cache sử dụng một thời gian và được thay thế bởi dòng khác theo yêu cầu thông tin phục vụ CPU

# 3. Bộ nhớ Cache

## ❖ Các chính sách thay thế dòng cache (tt)

- Chính sách thay thế (replacement policies): xác định cách thức lựa chọn các dòng trong cache để thay thế khi có dòng mới từ bộ nhớ cần chuyển vào
- 3 chiến lược chính:
  - Thay thế ngẫu nhiên (random replacement)
  - Vào trước ra trước FIFO (First In First Out)
  - Thay thế các dòng ít được sử dụng gần đây nhất LRU (Least Recently Used)

### 3. Bộ nhớ Cache

#### ❖ Các chính sách thay thế cache

- Thay thế ngẫu nhiên:

- Các dòng trong cache được chọn ngẫu nhiên để thay

- Đơn giản

- Tỷ lệ miss cao vì phương pháp này không xét tới dòng cache nào đang thực sự được sử dụng

- Nếu một dòng cache đang được sử dụng mà bị thay thế thì sự kiện miss xảy ra và cần phải đọc lại vào cache

# 3. Bộ nhớ Cache

## ❖ Các chính sách thay thế cache (tt)

- Thay thế kiểu vào trước ra trước FIFO
  - Dựa trên nguyên lý FIFO
  - Các dòng cache được đọc vào cache trước sẽ được chọn để thay trước
  - Tỷ lệ miss thấp hơn so với phương pháp ngẫu nhiên
  - Tỷ lệ miss vẫn cao vì phương pháp này vẫn chưa thực sự xem xét tới block nào đang thực sự được sử dụng
  - Cài đặt phức tạp vì cần thêm mạch để giám sát thứ tự nạp các dòng bộ nhớ vào cache

### 3. Bộ nhớ Cache

#### ❖ Các chính sách thay thế cache (tt)

- Thay thế các dòng ít được sử dụng gần đây nhất LRU
  - Các dòng cache ít được sử dụng nhất trong thời gian gần đây sẽ được chọn để thay
  - Xét tới các dòng đang được sử dụng
  - Tỷ lệ miss thấp nhất so với phương pháp ngẫu nhiên và FIFO
  - Cài đặt phức tạp vì cần thêm mạch để giám sát tần suất sử dụng các dòng cache

# 3. Bộ nhớ Cache

## ❖ Hiệu năng cache

Thời gian truy cập trung bình của hệ thống nhớ có cache:

$$t_{access} = (\text{Hit cost}) + (\text{miss rate}) * (\text{miss penalty})$$

$$t_{access} = t_{cache} + (1 - H) * (t_{memory}); H \text{ là hệ số hit}$$

Nếu  $t_{cache} = 5\text{ns}$ ,  $t_{memory} = 60\text{ns}$  và  $H=80\%$ , ta có:

$$t_{access} = 5 + (1 - 0.8) * (60) = 5+12 = 17\text{ns}$$

Nếu  $t_{cache} = 5\text{ns}$ ,  $t_{memory} = 60\text{ns}$  và  $H=95\%$ , ta có:

$$t_{access} = 5 + (1 - 0.95) * (60) = 5+3 = 8\text{ns}$$

### 3. Bộ nhớ Cache

#### ❖ Các yếu tố ảnh hưởng tới hiệu năng

- Kích thước cache
- Phân chia cache: một cache cho dữ liệu, một cache cho lệnh
- Tạo nhiều mức cache

### 3. Bộ nhớ Cache

#### ❖ Các yếu tố ảnh hưởng – kích thước cache

- Số liệu thống kê cho thấy:

- Kích thước cache ***không*** ảnh hưởng nhiều đến hệ số miss
- Hệ số miss của cache lệnh thấp hơn nhiều so với cache dữ liệu
- 8KB cache lệnh có hệ số miss < 1%
- 256KB cache lệnh có hệ số miss < 0.002%  
----> tăng kích thước cache lệnh không giảm miss hiệu quả.
- 8KB cache dữ liệu có hệ số miss < 4%
- 256KB cache dữ liệu có hệ số miss < 3%  
----> tăng kích thước cache dữ liệu lên 32 lần, hệ số miss giảm 25% (từ 4% xuống 3%).

# 3. Bộ nhớ Cache

## ❖ Các yếu tố ảnh hưởng – kích thước cache

### ▪ Kích thước cache lớn:

- Có thể lưu trữ nhiều khối dữ liệu trong bộ nhớ hơn
- Giảm tần suất trao đổi các khối dữ liệu của chương trình khác nhau giữa bộ nhớ và cache
- Cache lớn chậm hơn cache nhỏ:
- Xu hướng: cache càng ngày càng lớn
  - Hỗ trợ đa nhiệm (multi-tasking ) tốt hơn
  - Hỗ trợ tốt hơn cho xử lý song song
  - Hỗ trợ tốt hơn cho các hệ thống CPU đa nhân

### 3. Bộ nhớ Cache

#### ❖ Các yếu tố ảnh hưởng - Cache nhiều mức

- Nâng cao hiệu năng hệ thống vì cache nhiều mức có thể dung hòa tốc độ của CPU và MEM tốt hơn cache một mức

| CPU | L1  | L2   | L3   | Main memory |
|-----|-----|------|------|-------------|
| 1ns | 5ns | 15ns | 30ns | 60ns        |
| 1ns | 5ns |      | 60ns |             |

- Thực tế, cache thường có 2 mức: L1 và L2. Một số cache có 3 mức: L1, L2 và L3
- Giảm giá thành hệ thống nhớ

### 3. Bộ nhớ Cache

#### ❖ Các yếu tố ảnh hưởng – phân chia cache

- Cache có thể được chia thành cache lệnh và cache dữ liệu để hiệu năng tốt hơn vì:
  - Dữ liệu và lệnh khác nhau về tính cục bộ
  - Dữ liệu có tính chất cục bộ về thời gian hơn
  - Lệnh có tính chất cục bộ về không gian hơn
- Cache lệnh chỉ hỗ trợ thao tác đọc còn cache dữ liệu hỗ trợ cả 2 thao tác đọc ghi
  - Đễ tối ưu cache hơn
  - Hỗ trợ nhiều thao tác đọc/ ghi cùng lúc => giảm xung đột tài nguyên
  - Kết hợp các chức năng khác như giải mã trước lệnh vào trong cache lệnh => xử lý lệnh tốt hơn

### 3. Bộ nhớ Cache

#### ❖ Các yếu tố ảnh hưởng – phân chia cache (tt)

- Trong thực tế, hầu hết cache L1 được chia thành 2 phần:
  - I\_cache (Instruction cache): cache lệnh
  - D\_cache (data cache): cache dữ liệu.
- Các cache mức cao hơn không được chia
  - Chia cache L1 có lợi ích cao nhất vì nó gần CPU nhất. CPU đọc/ ghi trực tiếp lên cache L1. Cache L1 cần hỗ trợ nhiều lệnh truy nhập đồng thời và các biện pháp tối ưu hóa
  - Chia các cache mức cao hơn không đem lại hiệu quả cao và làm phức tạp trong hệ thống điều khiển cache

# 4. Bộ nhớ ngoài

## ❖ **Đĩa từ:**

- FDD
- HDD

## ❖ **Đĩa quang**

- CD
- DVD

## ❖ **RAID**

## ❖ **NAS**

## ❖ **SAN**

## 4. Bộ nhớ ngoài

### ❖ **Đĩa từ là các phương tiện lưu trữ:**

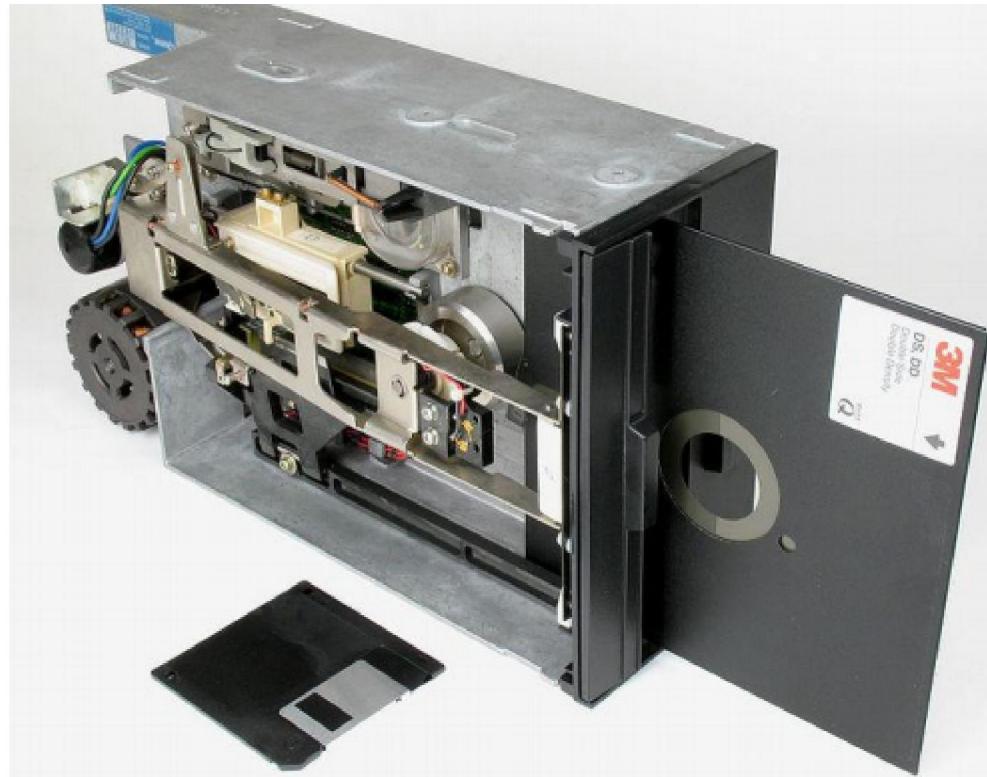
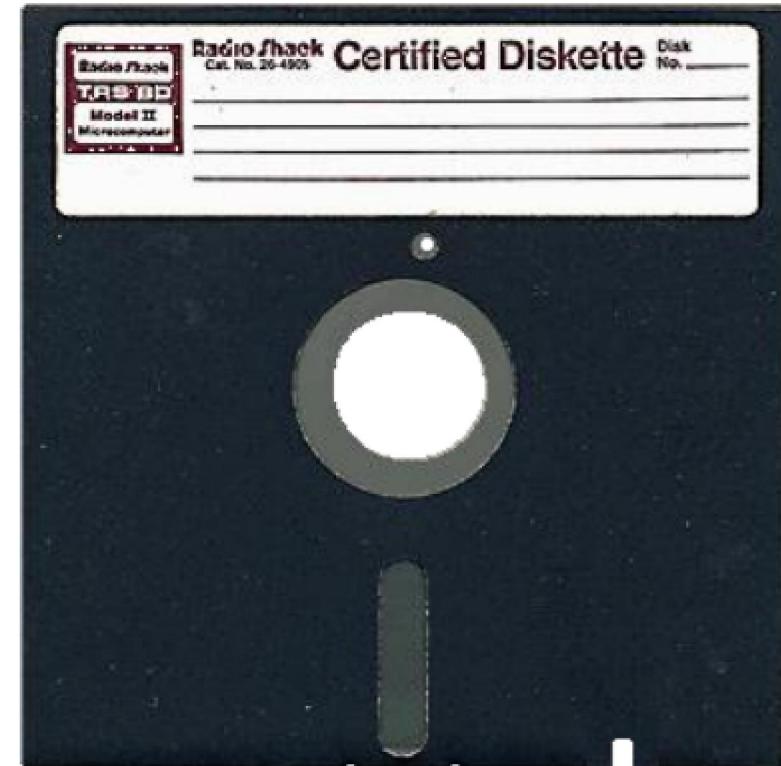
- Thiết bị lưu trữ thông tin kiểu ổn định
- Thiết bị lưu trữ lớn
- Dựa trên các nguyên lý từ và vật liệu sắt từ phủ mặt đĩa để lưu thông tin
- Thường dưới dạng đĩa nhựa hoặc kim loại

# 4. Bộ nhớ ngoài

## ❖ Các kiểu đĩa từ

- Đĩa mềm (FD: Floppy Disk): làm bằng plastic, dung lượng nhỏ, tốc độ chậm, dễ hỏng
  - Sử dụng ổ đĩa mềm (FDD: Floppy Disk Drive) để đọc ghi đĩa mềm
- Đĩa cứng (HD: Hard Disk): thường làm bằng kim loại, dung lượng lớn, tốc độ cao
  - Thường được gắn với ổ đĩa trong một hộp kín bảo vệ

# 4. Bộ nhớ ngoài



# 4. Bộ nhớ ngoài



# 4. Bộ nhớ ngoài



Techfuels.com

# 4. Bộ nhớ ngoài



# 4. Bộ nhớ ngoài

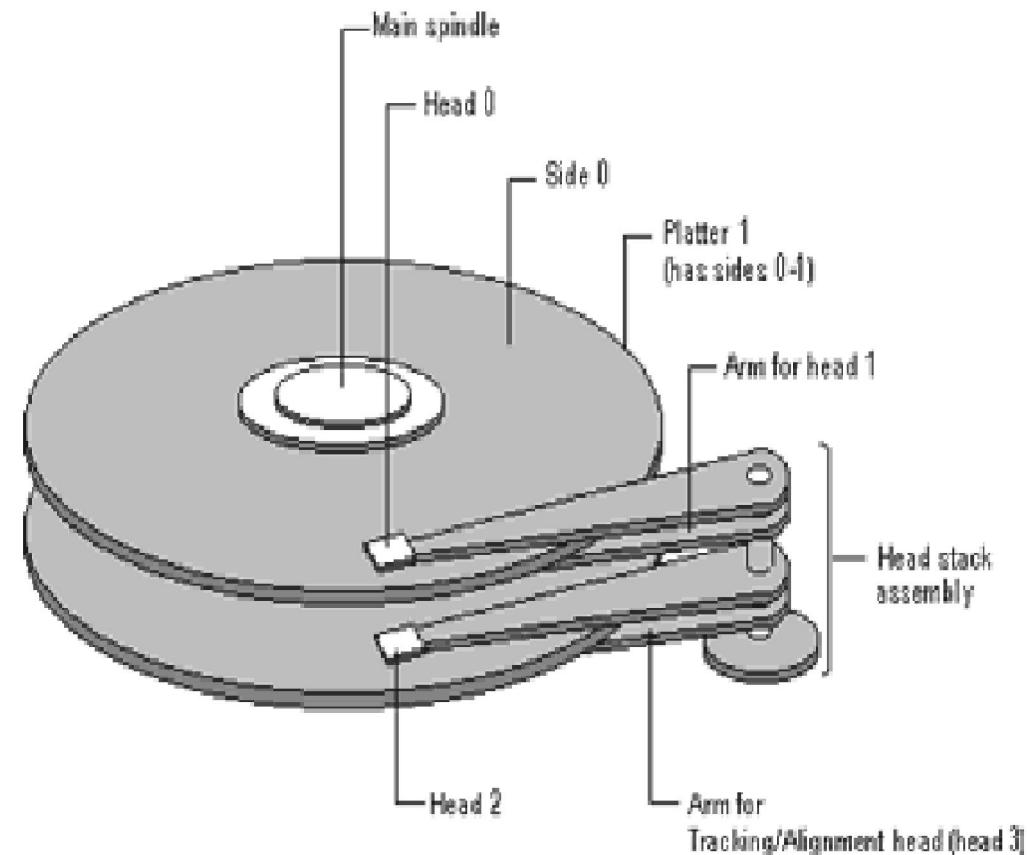
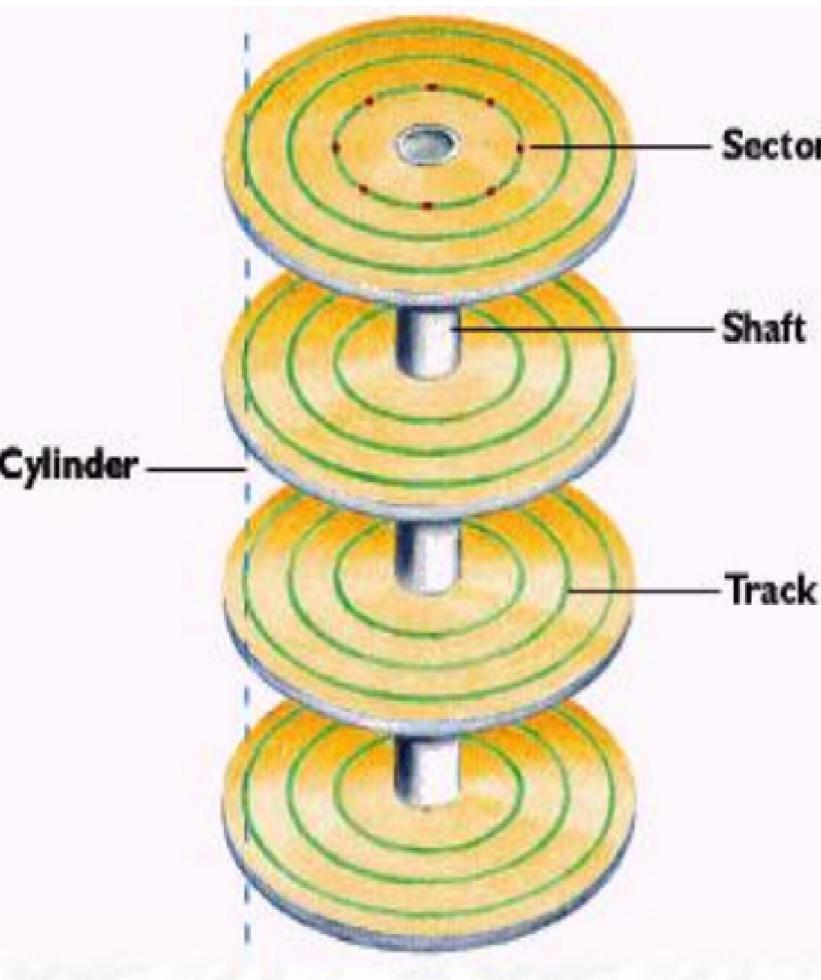
## ❖ Cấu tạo đĩa cứng

### ▪ Đĩa từ (Disks)

- HDD có thể có thể gồm một hoặc nhiều đĩa kim loại được lắp đồng trục (đặt trên cùng một trục quay)
- Đĩa thường phẳng và được chế tạo bằng nhôm hoặc thủy tinh
- Lớp bột từ tính phủ trên mặt đĩa để lưu trữ thông tin rất mỏng, chỉ khoảng 10 – 20nm
  - Oxide sắt 3 ( $Fe_2O_3$ ) được sử dụng trong các HDD cũ
  - Trong HDD hiện tại, sử dụng hợp kim coban và sắt
- Một đĩa có 2 mặt (side): mặt 0 và 1

# 4. Bộ nhớ ngoài

## ❖ Cấu tạo đĩa cứng



# 4. Bộ nhớ ngoài

## ❖ HDD: Các phần tử

- Đầu từ (head):

- Được sử dụng để đọc và ghi thông tin trên bề mặt đĩa
- Đầu từ không tiếp xúc mà chỉ “bay” trên bề mặt đĩa
- Số lượng đầu từ của mỗi ổ đĩa thường rất khác nhau: 4, 8, 12, 16, 24, 32, 64, ...

- Rãnh (tracks):

- Là các đường tròn đồng tâm trên bề mặt đĩa
- Được đánh số từ ngoài (0) vào trong
- Có hàng nghìn rãnh trên bề mặt 31/2 HDD

# 4. Bộ nhớ ngoài

## ❖ HDD: Các phần tử (tt)

- Cylinder (mặt trục):
  - Gồm tập các rãnh ở cùng vị trí đầu từ
- Sector (cung):
  - Là một phần của rãnh
  - Thông thường là 512 byte
  - Là đơn vị quản lý nhỏ nhất của đĩa

# 4. Bộ nhớ ngoài

## ❖ HDD: Các phần tử (tt)

- Các tham số HDD quan trọng để tính dung lượng:

- Số lượng cylinder (C)
- Số lượng đầu từ (H)
- Số lượng sector/ rãnh (S)
- Dung lượng = C x H x S x 512 (byte)

# 4. Bộ nhớ ngoài

## ❖ Định dạng đĩa cứng (format)

- Đĩa cứng có thể được định dạng theo 2 mức:
- Định dạng mức thấp (low level format):
  - Do BIOS thực hiện
  - Là quá trình gán địa chỉ (ID) cho các sector vật lý
  - Đĩa cứng phải được định dạng ở mức thấp trước khi sử dụng (tiếp tục với format mức cao)
  - Các HDD hiện đại thường được định dạng mức thấp bởi nhà sản xuất

# 4. Bộ nhớ ngoài

## ❖ Định dạng đĩa cứng (format)

- Định dạng mức cao (high level format):
  - Do hệ điều hành thực hiện
  - Là quá trình gán địa chỉ cho các sector logic vào tạo hệ thống file
  - HDD cũng phải được định dạng ở mức cao trước khi được sử dụng để lưu thông tin

# 4. Bộ nhớ ngoài

## ❖ Giao diện ghép nối HDD

- Các dạng giao diện ghép nối ổ đĩa cứng với máy tính gồm:
  - Parallel ATA (PATA hoặc IDE/EIDE - Integrated Drive Electronics) – Advanced Technology Attachments
  - Serial ATA (SATA)
  - SCSI – Small Computer System Interface
  - Serial Attached SCSI (SAS)
  - iSCSI – Internet SCSI

# 4. Bộ nhớ ngoài

## ❖ Giao diện ghép nối ATA

- ATA/ IDE sử dụng cáp dẹt (pin cables) 40 hoặc 80 chân để nối HDD với bảng mạch chủ
- Mỗi cáp thường hỗ trợ ghép nối với 2 ổ:
  - Một là ổ đĩa chủ (master)
  - Một là ổ đĩa tớ (slave)
- Tốc độ truyền dữ liệu:
  - Bandwidth: 16 bit
  - Thông lượng: 16, 33, 66, 100 và 133MB/s

# 4. Bộ nhớ ngoài

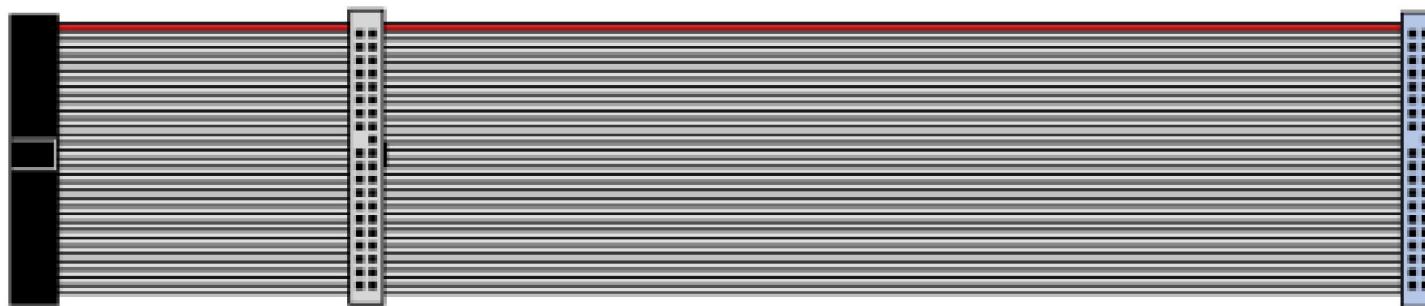
## ❖ Giao diện ghép nối ATA/PATA/IDE/EIDE



Đầu cắm IDE



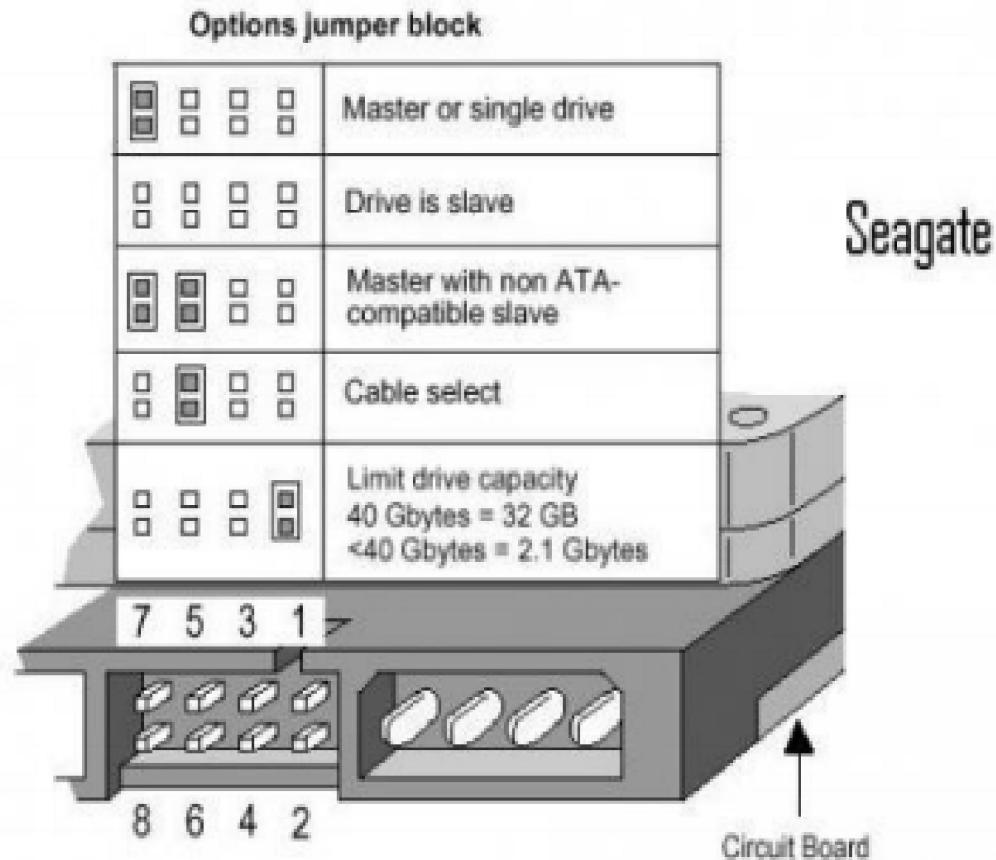
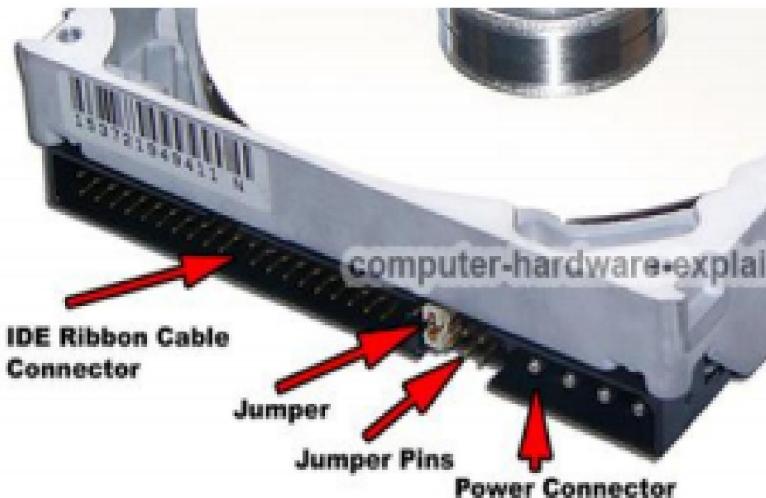
Khe cắm IDE



Cáp IDE

# 4. Bộ nhớ ngoài

## ❖ Giao diện ghép nối ATA/PATA/IDE/EIDE



# 4. Bộ nhớ ngoài

## ❖ Giao diện ghép nối SATA

- SATA sử dụng cùng tập lệnh mức thấp giống như ATA nhưng SATA sử dụng đường truyền tin nối tiếp tốc độ cao trên 2 cặp dây dẫn
- Bộ điều khiển SATA sử dụng chuẩn AHCI (Advanced Host Controller Interface)

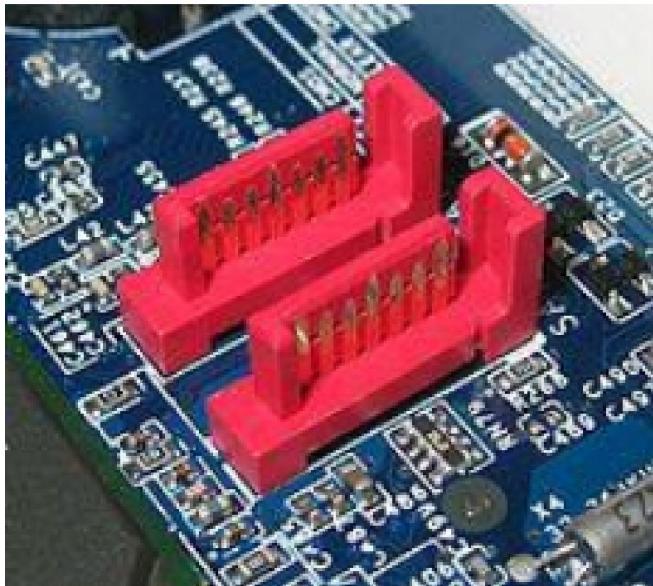
# 4. Bộ nhớ ngoài

## ❖ Giao diện ghép nối SATA (tt)

- SATA có nhiều đặc tính ưu việt hơn ATA:
  - Truyền dữ liệu nhanh và hiệu quả hơn
  - Hot plug
  - Số lượng dây cáp ít hơn
- Tốc độ truyền dữ liệu SATA:
  - Thế hệ 1: 1.5 Gb/s
  - Thế hệ 2: 3.0 Gb/s
  - Thế hệ 3: 6 Gb/s

# 4. Bộ nhớ ngoài

## ❖ Giao diện ghép nối SATA (tt)



SATA sockets



SATA data plug



SATA power plug

# 4. Bộ nhớ ngoài

## ❖ Phân khu (Partitions) đĩa cứng

- Đĩa cứng vật lý có thể được chia thành nhiều phần để dễ sử dụng và quản lý. Mỗi phần được gọi là một phân đoạn hay phân khu (partition):
  - Một phân khu chính (primary)
  - Một hoặc một số phân khu mở rộng (extended partitions)
- Một phân khu có thể được chia thành một hoặc một số ổ đĩa logic:
  - Phân khu chính chỉ có thể chứa duy nhất 1 ổ đĩa logic
  - Phân khu mở rộng có thể được chia thành một hoặc một số ổ đĩa logic

## 4. Bộ nhớ ngoài

### ❖ **Bảng phân khu (HDD Partition Table)**

- Lưu thông tin về các phân khu đĩa cứng
- Bảng có một số bản ghi (record), mỗi bản ghi chứa thông tin về một phân khu:
  - Phân khu này hoạt động hay không
  - Cylinder, đầu từ, sector bắt đầu của partition
  - Cylinder, đầu từ, sector cuối của partition
  - Kiểu định dạng của phân khu (FAT, NTFS)
  - Kích thước của phân khu tính theo số lượng sector

## 4. Bộ nhớ ngoài

### ❖ Boot sector (cung khởi động)

- Là sector đặc biệt trên đĩa:
  - Sector đầu tiên của ổ đĩa logic
  - Chứa chương trình mồi khởi động (Bootstrap loader) là đoạn chương trình nhỏ có nhiệm vụ kích hoạt việc nạp hệ điều hành từ HDD vào bộ nhớ trong

# 4. Bộ nhớ ngoài

## ❖ Boot sector (cung khởi động) (tt)

- Quá trình khởi động máy:
  - Bật nguồn
  - Đọc thông tin trong ROM-BIOS, tự kiểm tra cấu hình; kiểm tra các thiết bị & tình trạng sẵn sàng làm việc
  - Nạp và thực hiện chương trình mồi khởi động
  - Nạp các thành phần khởi động của hệ điều hành
  - Nạp nhân hệ điều hành
  - Nạp giao diện người dùng hệ điều hành -> sẵn sàng

## 4. Bộ nhớ ngoài

### ❖ Giới thiệu về đĩa quang

- Đĩa quang hoạt động dựa trên các nguyên lý quang học
  - Đĩa được tạo bằng plastic
  - Một lớp nhôm rất mỏng được đặt trên một mặt của đĩa để phản xạ tia laser
  - Mặt đĩa được “khắc” rãnh và mức lõm của rãnh (các mẫu pit và land) được sử dụng để biểu diễn các bit thông tin

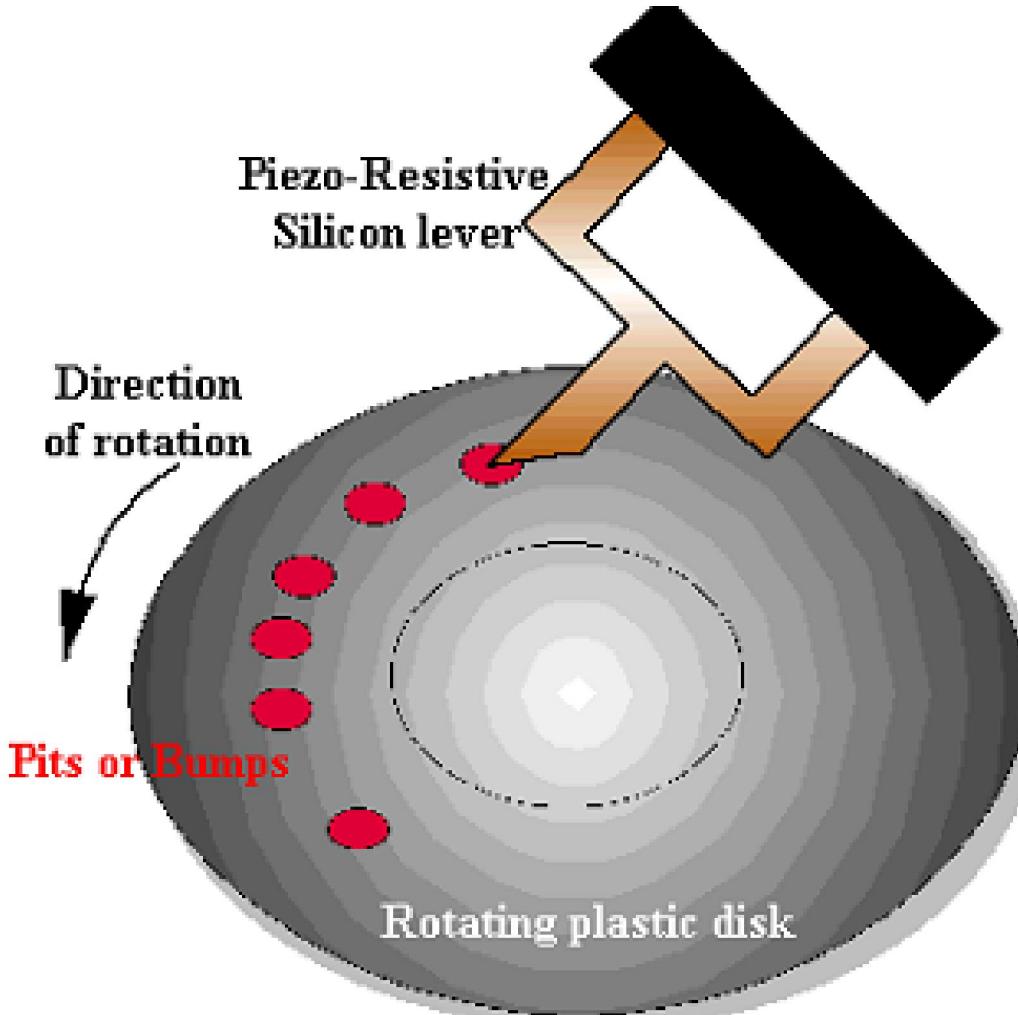
## 4. Bộ nhớ ngoài

### ❖ Giới thiệu về đĩa quang (tt)

- Cách thức tạo CD-ROMs
  - Tạo bản CD/ DVD chủ chứa thông tin ở dạng “âm bản”
  - Sử dụng bản CD/DVD chủ này để “in” thông tin lên các CD/DVD trắng

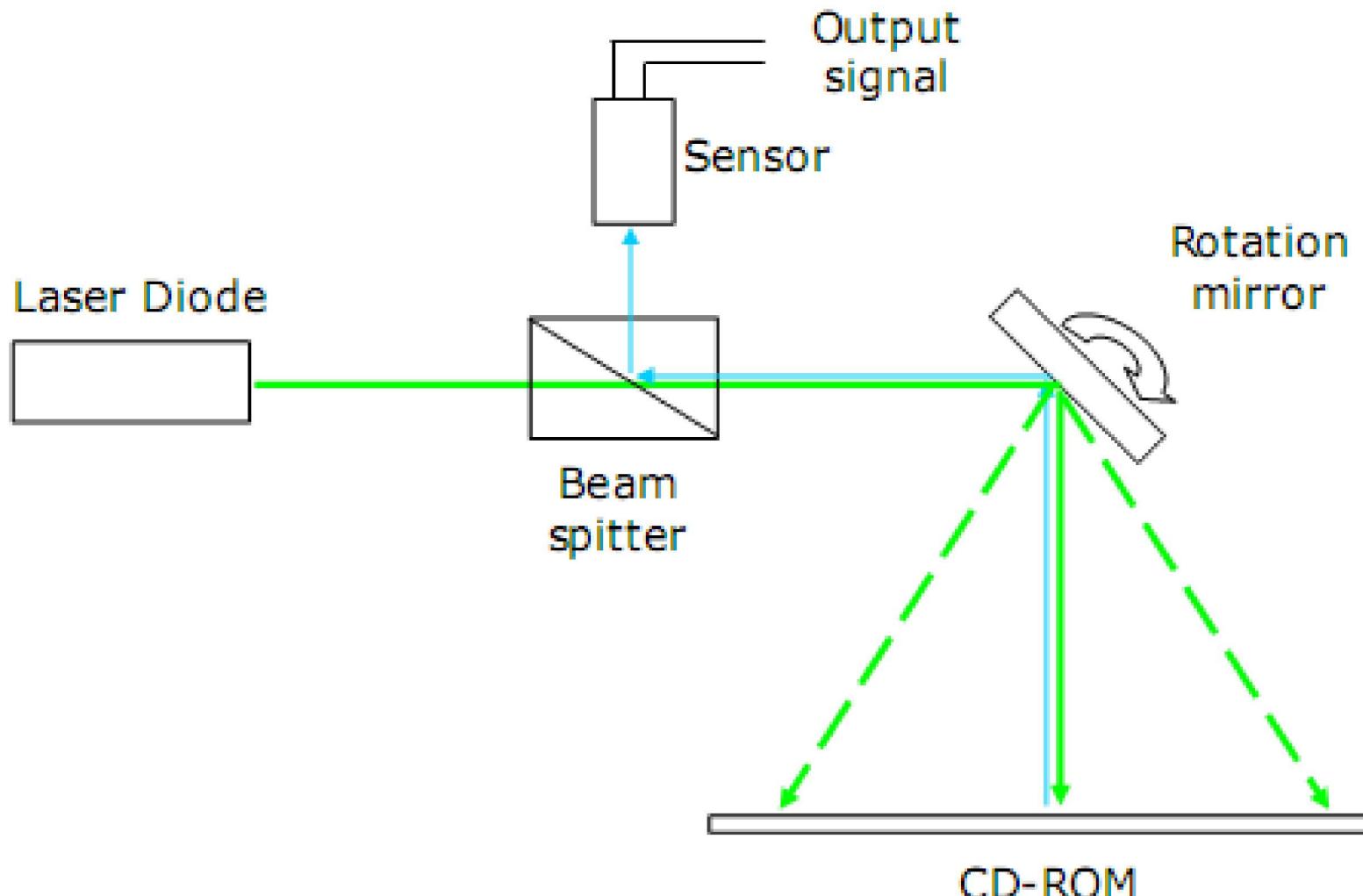
# 4. Bộ nhớ ngoài

## ❖ Đĩa quang – lưu trữ thông tin



## 4. Bộ nhớ ngoài

### ❖ Nguyên lý đọc thông tin CD-ROM



## 4. Bộ nhớ ngoài

### ❖ Nguyên lý đọc thông tin CD-ROM

- Tia laser từ đốt phát laser đi qua bộ tách tia (beam splitter) đến gương quay (rotation mirror)
- Gương quay được điều khiển bởi tín hiệu đọc, lái tia laser đến vị trí cần đọc trên mặt đĩa
- Tia phản xạ từ mặt đĩa phản ánh mức lồi lõm trên mặt đĩa quay trở lại gương quay
- Gương quay chuyển tia phản xạ về bộ tách tia và sau đó tới bộ cảm biến quang điện (sensor)
- Bộ cảm biến quang điện chuyển đổi tia laser phản xạ thành tín hiệu đầu ra. Cường độ tia laser được biểu diễn thành mức tín hiệu ra

# 4. Bộ nhớ ngoài

## ❖ Các loại đĩa quang

- CD (Compact Disk)
  - CD-ROM: Read Only CD
  - CD-R: Recordable CD: ghi 1 lần
  - CD-RW: Rewritable CD: ghi lại
- DVD (Digital Video Disk)
  - DVD-ROM: Read Only DVD
  - DVD-R: Recordable DVD
  - DVD-RW: Rewritable DVD
  - HD-DVD: High-density DVD
  - Blu-ray DVD: Ultra-high density DVD

## 4. Bộ nhớ ngoài

### ❖ Đĩa CD (Compact disk)

- Dung lượng tối đa là 700 MB
- Sử dụng tia laser hồng ngoại có bước sóng 780nm
- Tốc độ truyền dữ liệu:
  - Tốc độ cơ sở: 150KB/s
  - Tốc độ thực tế: hệ số nhân x tốc độ cơ sở:
    - $4x = 4 \times 150\text{KB/s} = 600\text{KB/s}$
    - $50x = 50 \times 150\text{KB/s} = 7500\text{KB/s}$

# Giới thiệu về RAID

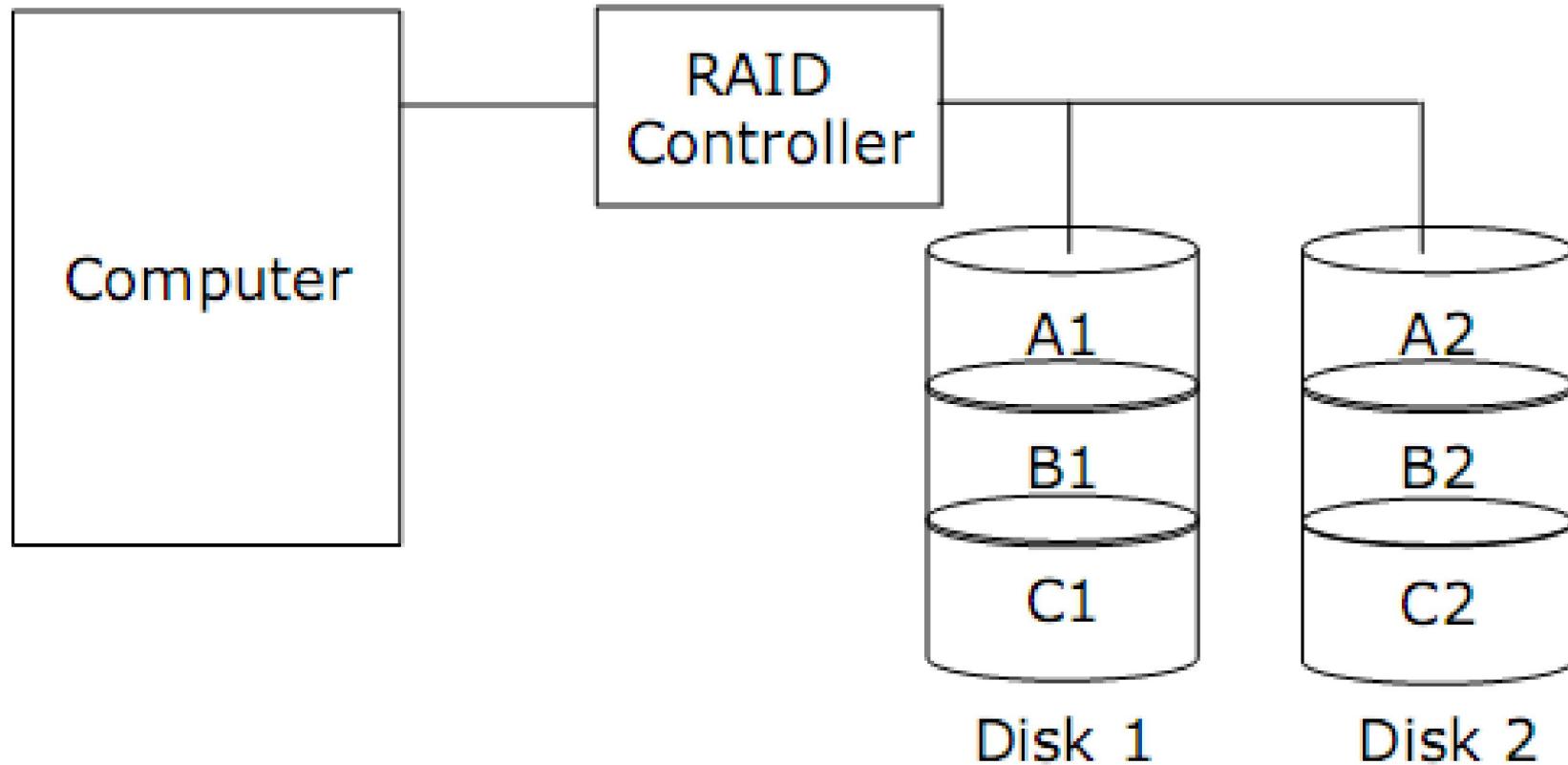
- ❖ RAID (Redundant Array of Independent Disks) là công nghệ tạo các thiết bị lưu trữ tiên tiến trên cơ sở đĩa cứng những mục đích sau:
  - Hiệu năng, tốc độ cao (high performance/ speed)
  - Độ tin cậy cao (high reliability)
  - Dung lượng lớn (large volume)
- ❖ RAID là một tập/ mảng các HDD nhưng được HDH coi như 1 ổ đĩa logic
- ❖ Các đĩa cứng theo chuẩn SATA và SCSI mới hỗ trợ tạo RAID
- ❖ Dữ liệu được phân tán trên các đĩa vật lý
- ❖ Đĩa dư thừa được sử dụng để lưu trữ thông tin parity => đảm bảo khôi phục dữ liệu

# Các kỹ thuật RAID

- ❖ 2 kỹ thuật cơ bản được sử dụng trong RAID:
  - Disk stripping (tạo lát đĩa):
    - Dữ liệu được chia thành các khối và mỗi khối được ghi đồng thời vào một đĩa độc lập
    - Sau đó, các khối dữ liệu có thể được đọc từ HDD một cách đồng thời
  - => Cải thiện tốc độ truy cập
- Disk mirroring :
  - Dữ liệu được chia thành các khối và mỗi khối được ghi vào một số đĩa
  - Tại thời điểm bất kì, luôn có nhiều hơn 1 bản sao dữ liệu => độ tin cậy tăng

# Các kỹ thuật RAID

## ❖ RAID – disk stripping



Disk stripping technique

# Các kỹ thuật RAID

❖ RAID – disk stripping (tt)

❖ Các đặc điểm:

- Dựa trên kỹ thuật disk stripping (đọc/ ghi song song)
- Dữ liệu được phân bố trên các đĩa trong mảng
- Tối thiểu cần 2 HDD

# Các kỹ thuật RAID

## ❖ RAID – disk stripping (tt)

- Các đặc điểm:

- Dựa trên kỹ thuật disk stripping (đọc/ ghi song song)
- Dữ liệu được phân bố trên các đĩa trong mảng
- Tối thiểu cần 2 HDD

# Các kỹ thuật RAID

## ❖ RAID – disk stripping (tt)

- Ưu:

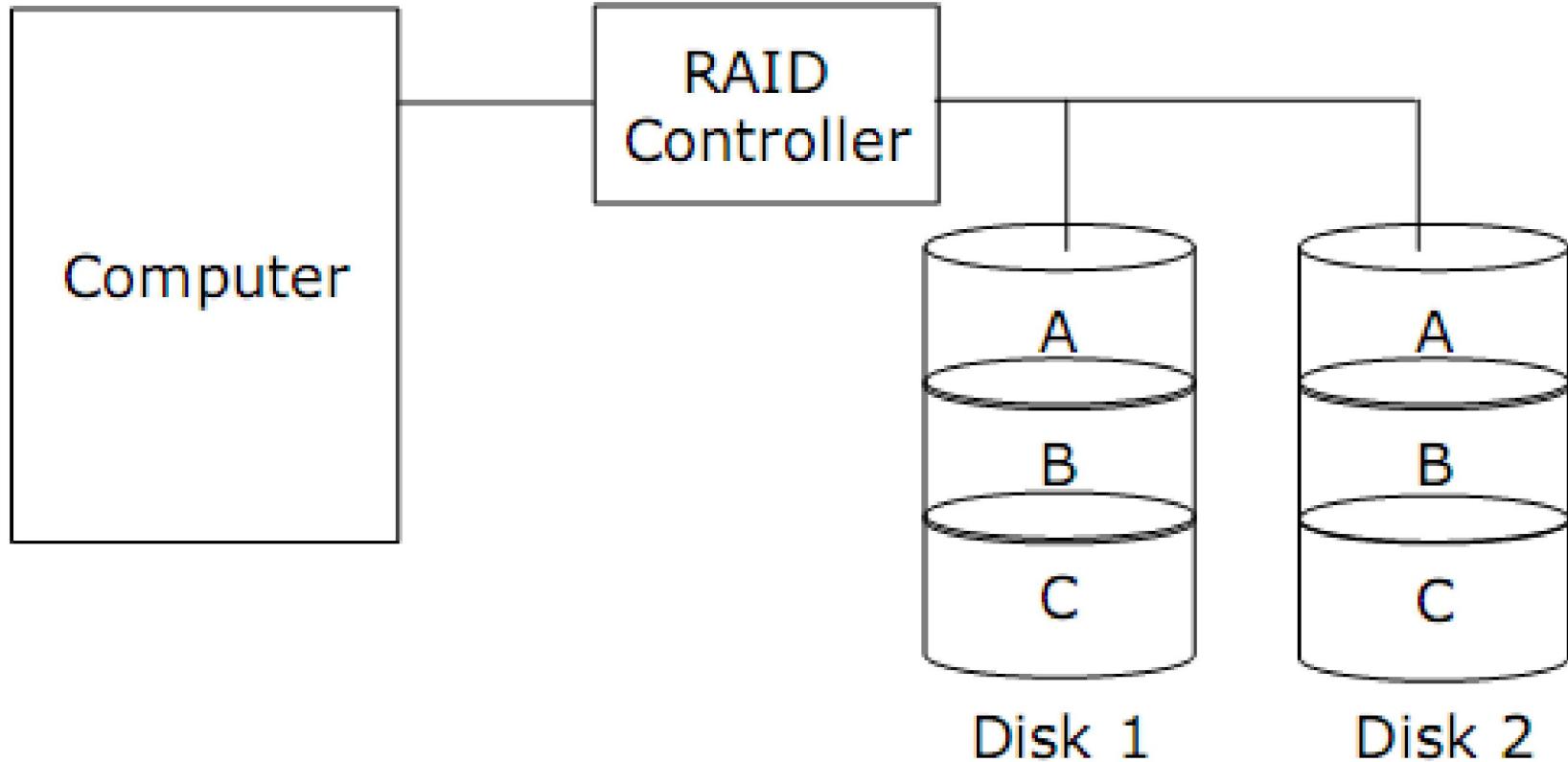
- Tốc độ nhanh
- Đáp ứng tốt các hệ thống nhu cầu I/O cao
- Dung lượng là tổng của tất cả các đĩa

- Nhược:

- Độ tin cậy như một đĩa

# Các kỹ thuật RAID

## ❖ RAID – disk mirroring



Disk mirroring technique

# Các kỹ thuật RAID

## ❖ RAID – disk mirroring

- Các đặc điểm:

- Dựa trên kỹ thuật disk mirroring (nhiều bản sao)
- Tính dư thừa có được đơn giản bằng cách sao tất cả dữ liệu
- Tối thiểu cần 2 HDD
- Dữ liệu cũng phân mảnh (data stripping) như RAID 0 nhưng mỗi mảnh logic được ánh xạ tới 2 đĩa vật lý khác nhau

=> Mỗi đĩa trong mảng có một bản sao cùng dữ liệu (mirror)

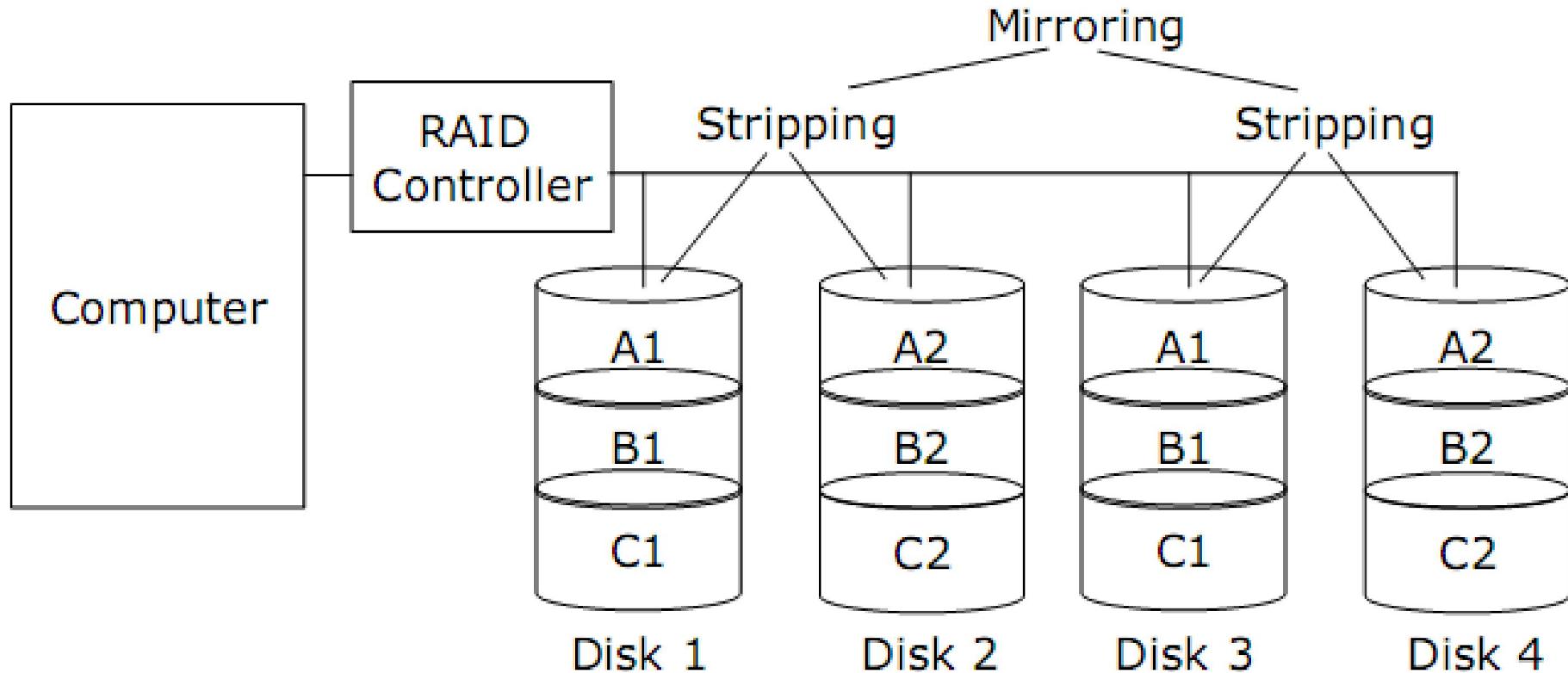
# Các kỹ thuật RAID

## ❖ RAID – disk mirroring (tt)

- Ưu:
  - Độ tin cậy cao
- Nhược:
  - Dung lượng thực sự bằng  $\frac{1}{2}$  tổng số đĩa
  - Chi phí cao

# Các kỹ thuật RAID

## ❖ RAID 10 – disk stripping & mirroring



# Các kỹ thuật RAID

## ❖ RAID 10 – disk stripping & mirroring (tt)

- Các đặc điểm:
  - Tối thiểu cần 4 HDD
  - Dựa trên kỹ thuật disk mirroring và stripping
- Ưu:
  - Nhanh hơn so với một đĩa
  - Tin cậy hơn so với một đĩa
- Nhược:
  - Dung lượng bằng một nửa dung lượng tổng số đĩa

## 4. Bộ nhớ ảo (Virtual Memory)

- ❖ Khi độ dài của chương trình vượt quá giới hạn dung lượng bộ nhớ → phải có sự quản lý việc trao đổi thông tin giữa bộ nhớ và đĩa từ.
- ❖ Bộ nhớ ảo làm cho việc trao đổi thông tin này được thực hiện một cách tự động.
- ❖ Bộ nhớ ảo xác định một cơ chế vận chuyển tự động số liệu giữa bộ nhớ trong và bộ nhớ ngoài (đĩa từ).

# 4. Bộ nhớ ảo (Virtual Memory)

- ❖ Bộ nhớ ảo là kỹ thuật do hệ điều hành, có hỗ trợ của phần cứng
- ❖ Cho phép thực hiện chương trình lớn hơn bộ nhớ trong bằng cách sử dụng bộ nhớ ngoài
- ❖ Hoạt động dạng overlay tự động
- ❖ Bộ nhớ ảo = bộ nhớ trên đĩa
- ❖ Bộ nhớ ảo bao gồm bộ nhớ trong và bộ nhớ ngoài được phân tích thành khối để có thể cung cấp cho mỗi chương trình một số khối cần thiết cho việc thực hiện chương trình đó.

# 4. Bộ nhớ ảo (Virtual Memory)

- ❖ Chương trình được viết trên không gian địa chỉ ảo, là thông số của CPU và hệ điều hành
- ❖ Khi thực thi, hệ điều hành:
  - nạp chương trình vào bộ nhớ
  - chuyển đổi địa chỉ ảo thành địa chỉ vật lý
  - truy xuất trên bộ nhớ vật lý

# 4. Bộ nhớ ảo (Virtual Memory)

địa chỉ ảo

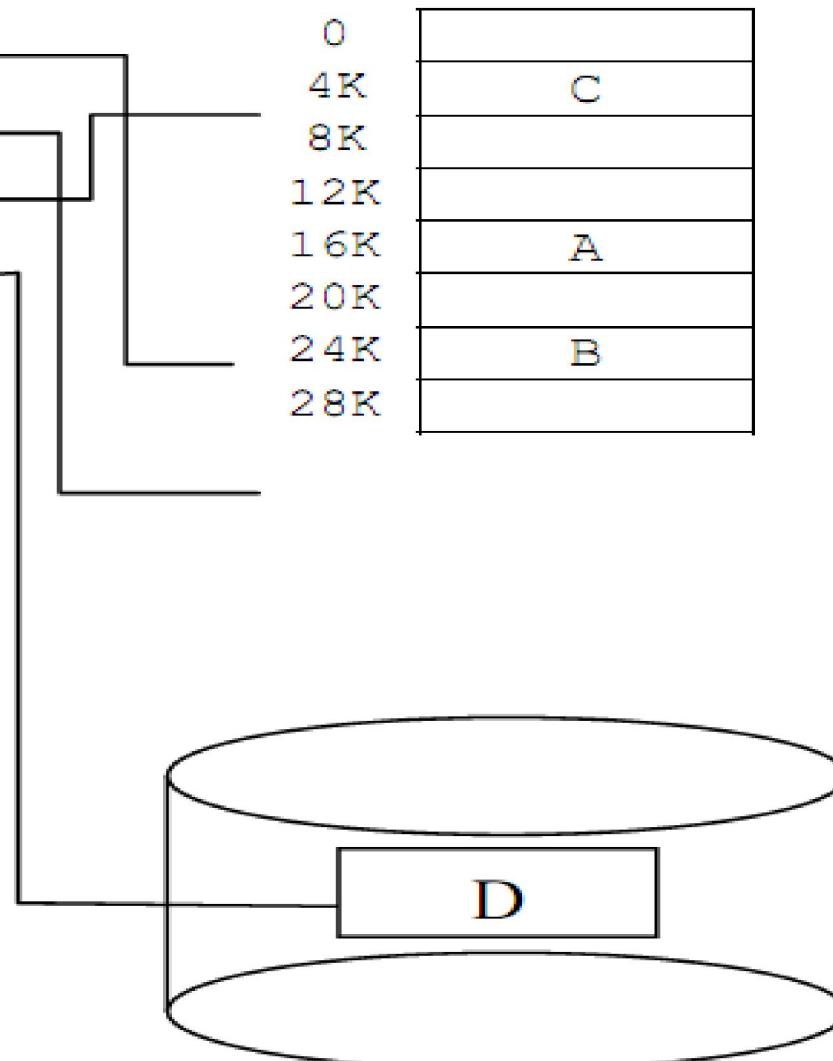
|     |     |
|-----|-----|
| 0   | A   |
| 4K  | B   |
| 8K  | C   |
| 12K | D   |
| 16K |     |
| 20K |     |
| 24K |     |
| 28K |     |
| 32K |     |
| ... | ... |
| ... | ... |
| 16M |     |

địa chỉ vật lý

|     |   |
|-----|---|
| 0   |   |
| 4K  | C |
| 8K  |   |
| 12K |   |
| 16K | A |
| 20K |   |
| 24K | B |
| 28K |   |

bộ nhớ trong

bộ nhớ ảo



Đĩa cứng

## 4. Bộ nhớ ảo (Virtual Memory)

- ❖ Bộ nhớ ảo đơn giản hóa việc nạp chương trình vào bộ nhớ để thi hành nhờ một cơ chế được gọi là sự tái định địa chỉ (address relocation). Cơ chế này cho phép một chương trình có thể được thi hành khi nó nằm ở bất cứ vị trí nào trong bộ nhớ.

# 4. Bộ nhớ ảo (Virtual Memory)

| Tham số   | Cache   | Bộ nhớ ảo  |
|---|---|--|
| Chiều dài mỗi khối (trang)  | 16 - 128 byte                                   | 4096 - 65536 bytes   |
| Thời gian thâm nhập thành công  | 1 - 2 xung nhịp                                 | 40 - 100 xung nhịp   |
| Trừng phạt khi thất bại<br>(Thời gian thâm nhập)<br>(Di chuyển số liệu) | 8 - 100 xung nhịp<br>6 - 60 xung<br>2 - 40 xung | 700.000 - 6 triệu xung<br>500.000 - 4 triệu xung<br>200.000 - 2 triệu xung |
| Tỉ số thất bại  | 0,5% - 10%                                      | 0,00001% - 0,001%  |
| Dung lượng  | 8 KB – 8MB                                      | 16 MB – 8GB  |

Bảng IV.3: Đại lượng điển hình cho bộ nhớ cache và bộ nhớ ảo.  
So với bộ nhớ cache thì các tham số của bộ nhớ ảo tăng từ 10 đến 100.000 lần

# 4. Bộ nhớ ảo (Virtual Memory)

❖ Những khác biệt khác giữa bộ nhớ cache và bộ nhớ ảo là:

- Khi thất bại cache, sự thay thế một khối trong cache được điều khiển bằng phần cứng, trong khi sự thay thế trong bộ nhớ ảo là chủ yếu do hệ điều hành.
- Không gian định vị mà bộ xử lý quản lý là không gian định vị của bộ nhớ ảo, trong lúc đó dung lượng bộ nhớ cache không thuộc vào không gian định vị bộ xử lý.
- Bộ nhớ ngoài còn được dùng để lưu trữ tập tin ngoài nhiệm vụ là hậu phương của bộ nhớ trong (trong các cấp bộ nhớ).
- Bộ nhớ ảo cũng được thiết kế bằng nhiều kỹ thuật đặc thù cho chính nó.

# 4. Bộ nhớ ảo (Virtual Memory)

## ❖ Các dạng bộ nhớ ảo

- Dạng phân trang (paging)
- Dạng phân đoạn (segmentation)
- Dạng phân đoạn có phân trang (Paged segmentation /Segmentation with paging)

# 4. Bộ nhớ ảo (Virtual Memory)

## ❖ Bộ nhớ ảo dạng phân trang

- Tổ chức phân trang
- Chuyển đổi địa chỉ
- Nạp trang
- Thay thế trang

# 4. Bộ nhớ ảo (Virtual Memory)

## ❖ Tổ chức phân trang

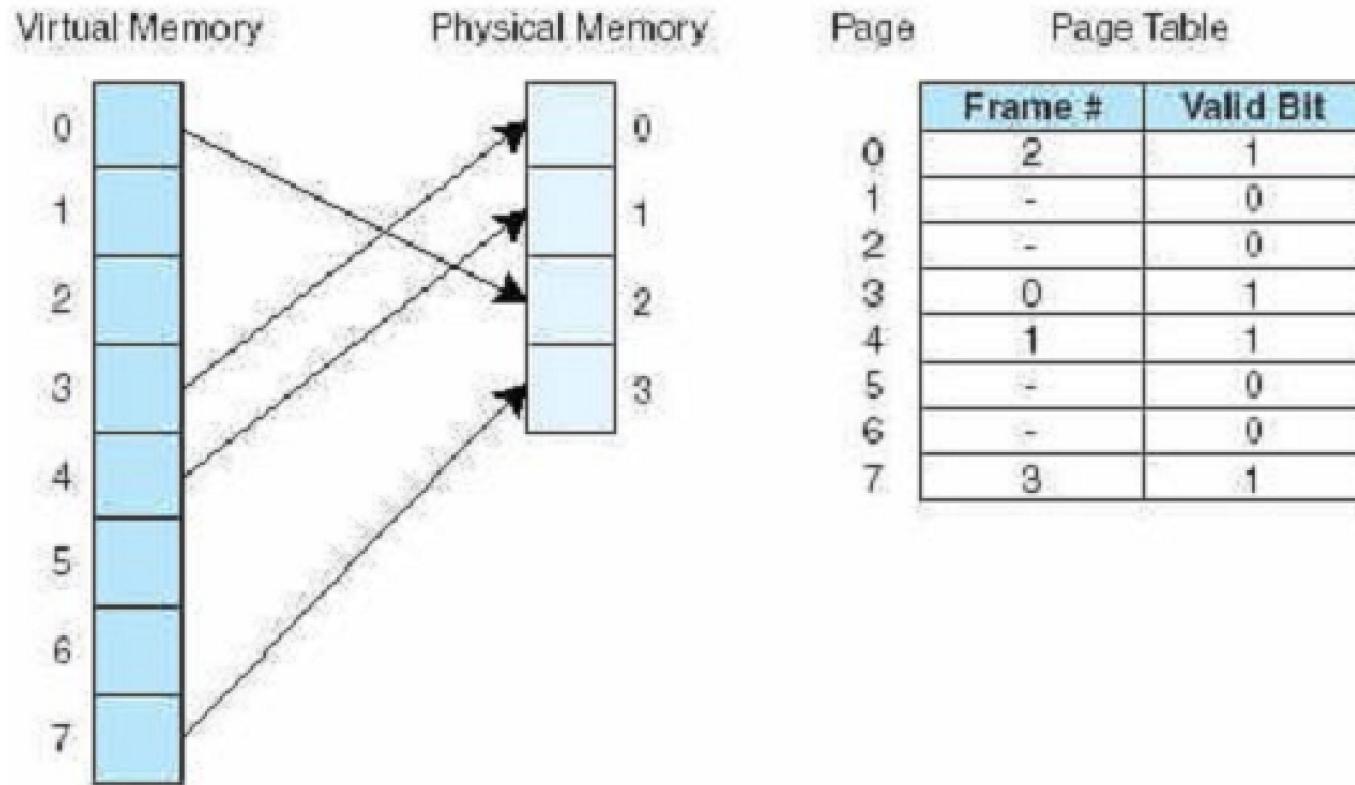
- Không gian địa chỉ ảo chia thành các trang (page)
- Bộ nhớ được chia thành các khung trang (page frame), chứa được một trang
  - Các trang của một chương trình khi ở trên bộ nhớ không cần ở các vị trí liên tục
  - Số trang >> số khung
- Có bảng trang (page table) quản lý các trang

# 4. Bộ nhớ ảo (Virtual Memory)

## ❖ Bảng trang

- Có N dòng, với N là số trang
  - → mỗi dòng tương ứng với 1 trang
- Cấu trúc 1 dòng:
  - Valid bit
    - Valid = 0 nếu trang chưa có trên bộ nhớ
    - Valid = 1 nếu trang đang có trên bộ nhớ
  - Frame Number (Frame #)
    - Số thứ tự khung trang đang chứa trang

# 4. Bộ nhớ ảo (Virtual Memory)



Bộ nhớ ảo có 8 trang, bộ nhớ vật lý có 4 khung  
Bảng trang có 8 phần tử

# 4. Bộ nhớ ảo (Virtual Memory)

## ❖ Chuyển đổi địa chỉ

- Địa chỉ ảo theo phân trang
  - Page Number: số thứ tự trang
  - Offset: địa chỉ trong trang
- Địa chỉ vật lý 

|             |        |
|-------------|--------|
| Page Number | Offset |
|-------------|--------|

# 4. Bộ nhớ ảo (Virtual Memory)

## ❖ Chuyển đổi địa chỉ

### ▪ Địa chỉ ảo theo phân trang

- Page Number: số thứ tự trang

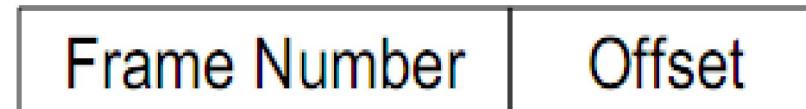
- Offset: địa chỉ trong trang



### ▪ Địa chỉ vật lý

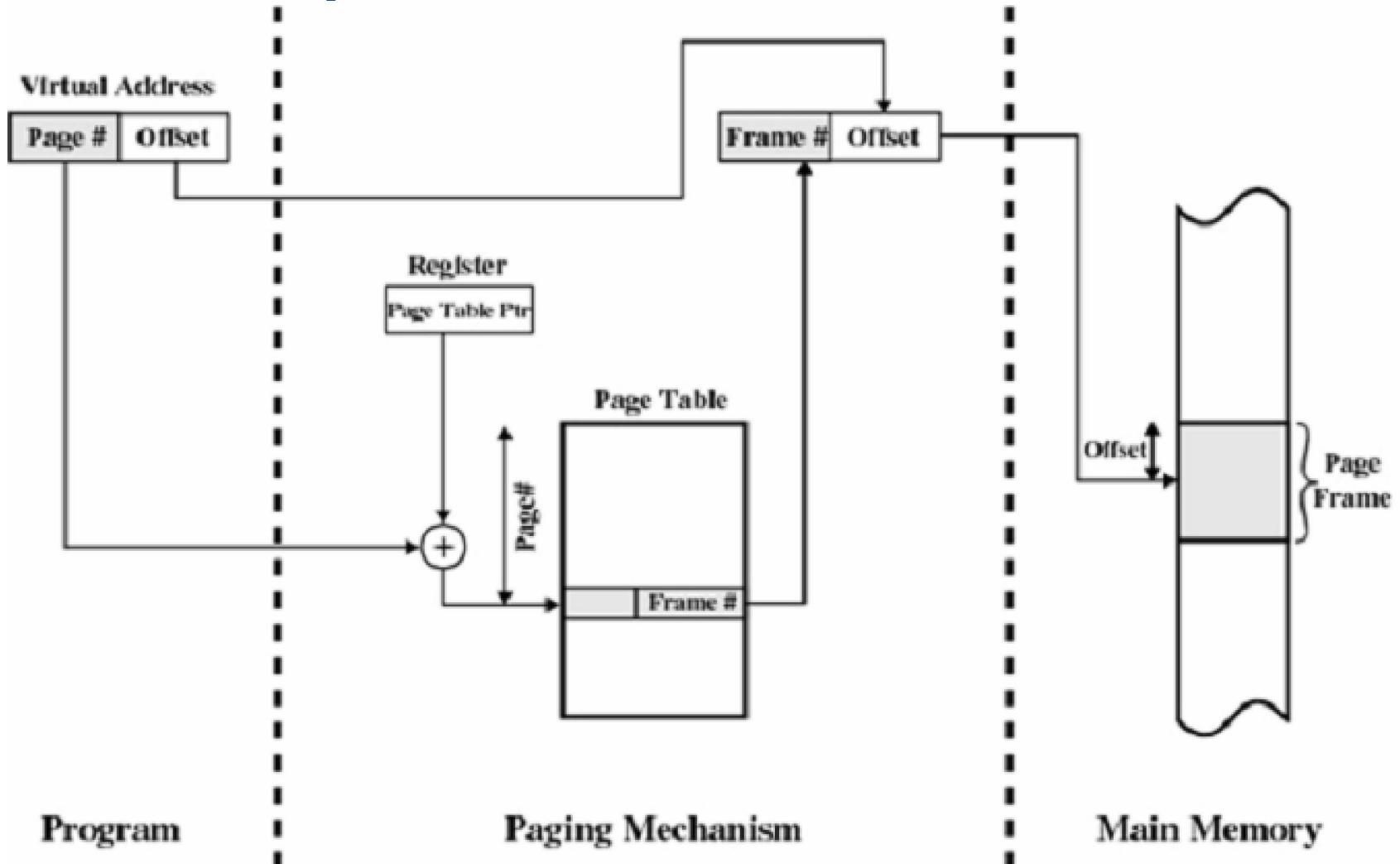
- Page Number: số thứ tự khung trang

- Offset: địa chỉ trong khung trang



# 4. Bộ nhớ ảo (Virtual Memory)

## ❖ Sơ đồ chuyển đổi địa chỉ



# 4. Bộ nhớ ảo (Virtual Memory)

❖ Truy xuất dữ liệu theo địa chỉ ảo

- Tách page # và offset từ địa chỉ ảo
- Chuyển page # thành frame # bằng cách truy xuất bảng trang
  - A. Tìm phần tử quản lý trang
  - B. Kiểm tra valid bit
- 1. Valid = 1
  - a. Thay page # bằng frame #
  - b. Truy xuất dữ liệu trên khung với vị trí offset

# 4. Bộ nhớ ảo (Virtual Memory)

❖ Truy xuất dữ liệu theo địa chỉ ảo (tt)

- 2.  $\text{Valid} = 0 \rightarrow$  lỗi trang
  - a. Tìm trang trên đĩa
  - b. Tìm một khung trống (có thể phải thay thế trang nếu các khung đầy)
  - c. Sao chép trang vào khung trống
  - d. Cập nhật bảng trang ( $\text{valid} = 1$ , frame # mới)
  - e. Thực hiện truy xuất như bước 1

# 4. Bộ nhớ ảo (Virtual Memory)

## ❖ Nạp trang

- Có 2 phương pháp:
  - Nạp theo yêu cầu: Nạp trang khi có lỗi trang
  - Nạp trước : Nạp trước các trang theo các điều kiện xác định

# 4. Bộ nhớ ảo (Virtual Memory)

## ❖ Thay thế trang

- Khi các khung đã đầy mà cần nạp thêm trang thì phải thay thế một trang đang có trên khung.
  - Nếu trang bị thay thế có thay đổi nội dung thì phải đưa ra đĩa
- Có các phương pháp chọn phần tử thay thế: Optimal, FIFO, LRU (thông dụng)
- Least Recently Used (LRU): chọn trang không được truy xuất trong khoảng thời gian lớn nhất

# 4. Bộ nhớ ảo (Virtual Memory)

## ❖ Bộ nhớ ảo dạng phân đoạn

### ▪ Tổ chức phân đoạn

- Bộ nhớ ảo bao gồm các đoạn (segment) có kích thước không cố định
- Khi nạp đoạn vào bộ nhớ thì hệ điều hành tìm khoảng trống đủ để nạp đoạn
- Có bảng đoạn quản lý các đoạn

# 4. Bộ nhớ ảo (Virtual Memory)

## ❖ Nhận xét về phân trang, phân đoạn

- Trang trong suốt đối với người lập trình
- Phân trang tránh được phân mảnh bên ngoài
- Người lập trình sử dụng được đoạn
- Phân đoạn phù hợp với lập trình theo khối, cấu trúc dữ liệu thay đổi, dùng chung và bảo vệ bộ nhớ

# 4. Bộ nhớ ảo (Virtual Memory)

- ❖ Bộ nhớ ảo dạng phân đoạn có phân trang
  - Kết hợp các ưu điểm của phân đoạn và phân trang
  - **Tổ chức:**
    - Bộ nhớ ảo bao gồm các đoạn
    - Trong mỗi đoạn thực hiện phân trang