

# Chương 4: Bộ nhớ máy tính

- 4.1. Tổng quan về hệ thống nhớ
- 4.2. Bộ nhớ bán dẫn
- 4.3. Bộ nhớ chính
- 4.4. Bộ nhớ cache
- 4.5. Bộ nhớ ngoài
- 4.6. Bộ nhớ ảo
- 4.7. Hệ thống nhớ trên máy tính cá nhân

# 4.1. Tổng quan về hệ thống nhớ

## 1. 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)
  - Số lượng từ nhớ

# Các đặc trưng của hệ thống nhớ (tiếp)

## ■ Đơn vị truyền

- Từ nhớ
- Khối nhớ

## ■ 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 (tiếp)

## ■ 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

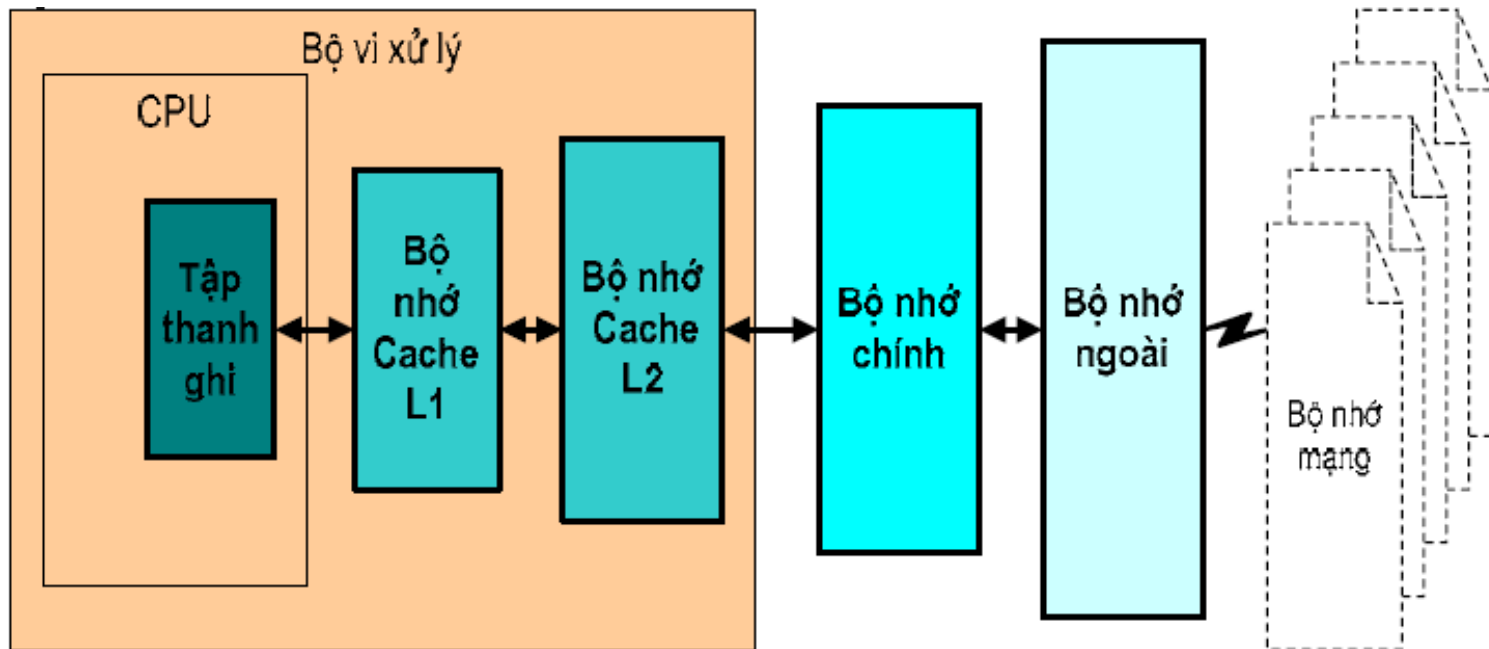
# Các đặc trưng của hệ thống nhớ (tiếp)

## ■ Các đặc tính vật lý

- Khả biến / Không khả biến  
(volatile / nonvolatile)
- Xoá được / không xoá được

## ■ Tổ chức

## 2. Phân cấp hệ thống nhớ



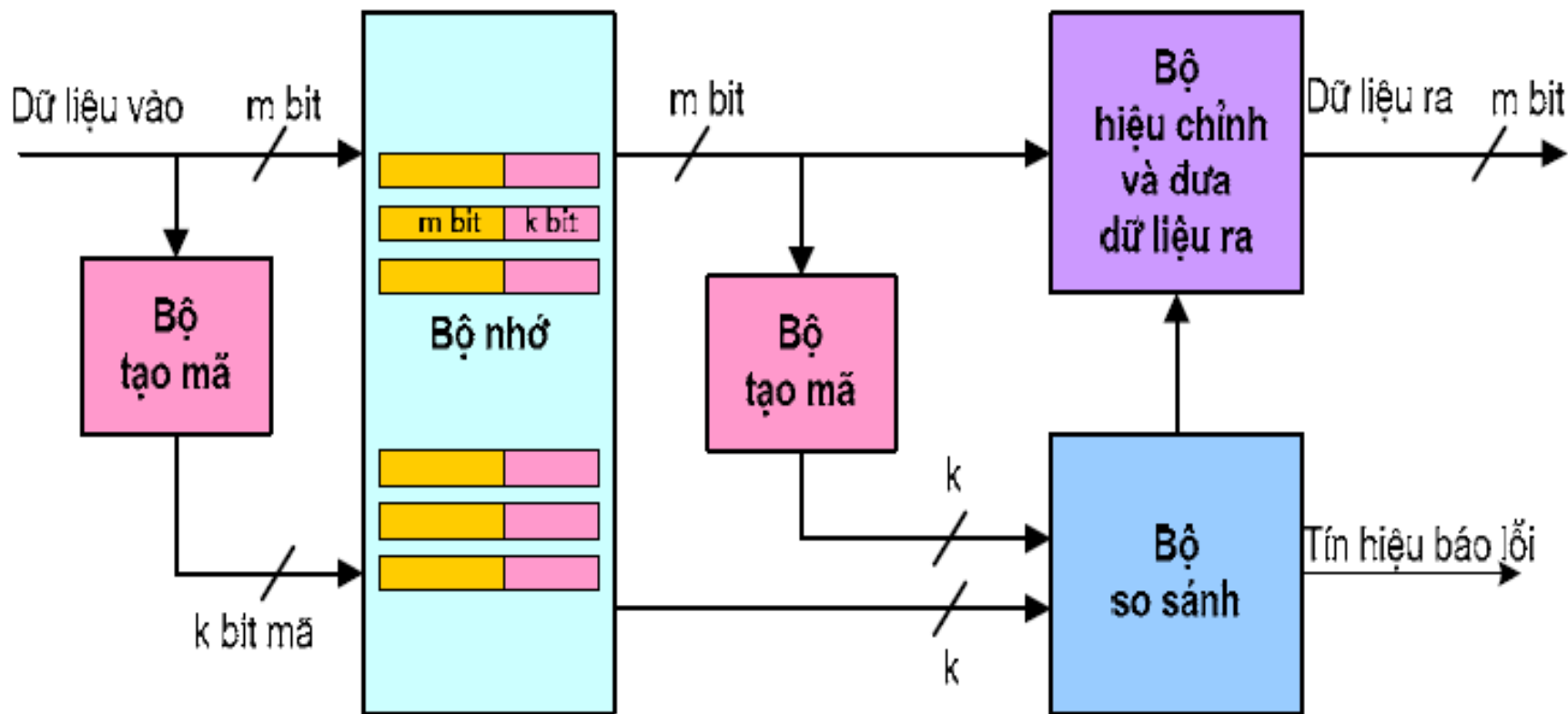
Từ trái sang phải:

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

# 3. Phát hiện và hiệu chỉnh lỗi trong bộ nhớ

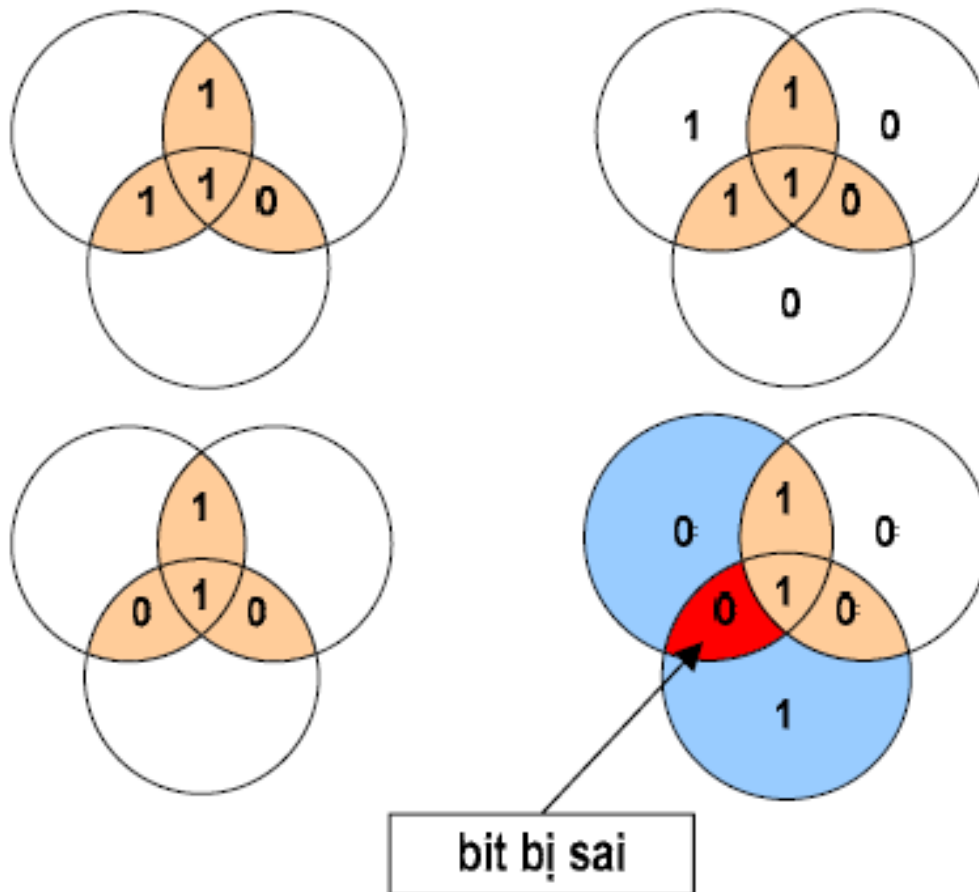
- Nguyên tắc chung: cần tạo ra và lưu trữ thêm thông tin dự thừa.
- Từ dữ liệu cần ghi vào bộ nhớ:  $m$  bit
- Cần tạo ra và lưu trữ từ mã:  $k$  bit  
→ Lưu trữ  $(m+k)$  bit
- Khi đọc ra có các khả năng sau:
  - Không phát hiện thấy dữ liệu lỗi
  - Phát hiện thấy dữ liệu lỗi và có thể hiệu chỉnh dữ liệu thành đúng
  - Phát hiện thấy lỗi nhưng không có khả năng hiệu chỉnh → cần phát ra tín hiệu báo lỗi.

# Sơ đồ phát hiện và hiệu chỉnh lỗi





# Ví dụ mã sửa lỗi Hamming ( $m=4$ , $k=3$ )



## 4.2. Bộ nhớ bán dẫn

### 1. Phân loại

Kiểu bộ nhớ	Tiêu chuẩn	Khả năng xóa	Cơ chế ghi	Tính khả biến
Read Only Memory (ROM)	Bộ nhớ chỉ đọc	Không xóa được	Mặt nạ	Không khả biến
Programmable ROM (PROM)			Bảng điện	
Erasable PROM (EPROM)	Bộ nhớ hầu như chỉ đọc	bằng tia cực tím, cả chip		
Electrically Erasable PROM (EEPROM)		bằng điện, mức từng byte		
Flash memory	Bộ nhớ đọc-ghi	bằng điện, từng khối	Bảng điện	Khả biến
Random Access Memory (RAM)		bằng điện, mức từng byte		

# ROM (Read Only Memory)

- Bộ nhớ không khả biến
- 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

# Các kiểu ROM

## ■ ROM mặt nạ:

- thông tin được ghi khi sản xuất
- rất đắt

## ■ PROM (Programmable ROM)

- Cần thiết bị chuyên dụng để ghi bằng chương trình → chỉ ghi được một lần

## ■ EPROM (Erasable PROM)

- Cần thiết bị chuyên dụng để ghi bằng chương trình → ghi được nhiều lần
- Trước khi ghi lại, xóa bằng tia cực tím

# Các kiểu ROM (tiếp)

- EEPROM (Electrically Erasable PROM)
  - Có thể ghi theo từng byte
  - Xóa bằng điện
- Flash memory (Bộ nhớ cực nhanh)
  - Ghi theo khối
  - Xóa bằng điện

# RAM (Random Access Memory)

- Bộ nhớ đọc-ghi (Read/Write Memory)
- Khả biến
- Lưu trữ thông tin tạm thời
- Có hai loại: SRAM và DRAM  
(Static and Dynamic)

# SRAM (Static) – RAM tĩnh

- Các bit được lưu trữ bằng các Flip-Flop → thông tin ổn định
- Cấu trúc phức tạp
- Dung lượng chip nhỏ
- Tốc độ nhanh
- Đắt tiền
- Dùng làm bộ nhớ cache

# DRAM (Dynamic) – RAM động

- Các bit được lưu trữ trên tụ điện  
→ cần phải có mạch làm tươi
- Cấu trúc đơn giản
- Dung lượng lớn
- Tốc độ chậm hơn
- Rẻ tiền hơn
- Dùng làm bộ nhớ chính

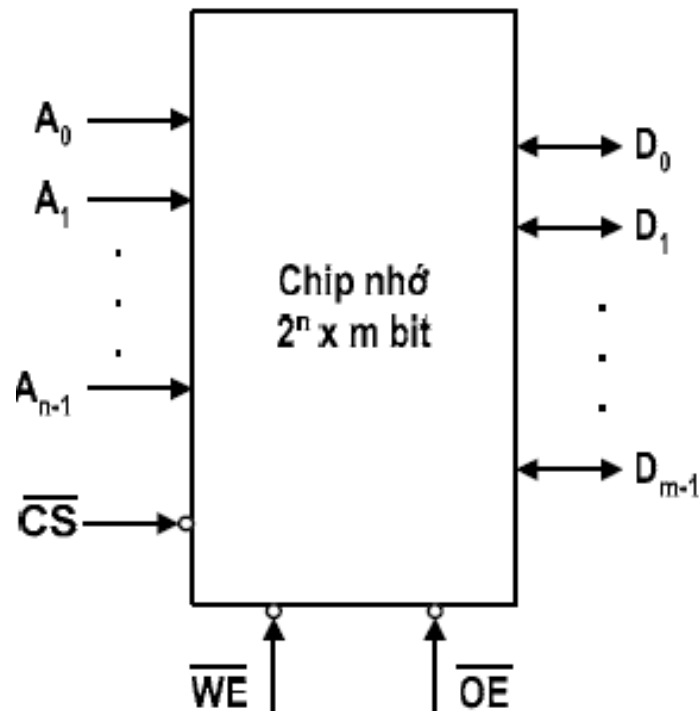


# Các DRAM tiên tiến

- Enhanced DRAM
- Cache DRAM
- Synchronous DRAM (SDRAM): làm việc được đồng bộ bởi xung clock
- DDR-SDRAM (Double Data Rate SDRAM)
- Rambus DRAM (RDRAM)

## 2. Tổ chức của chip nhớ

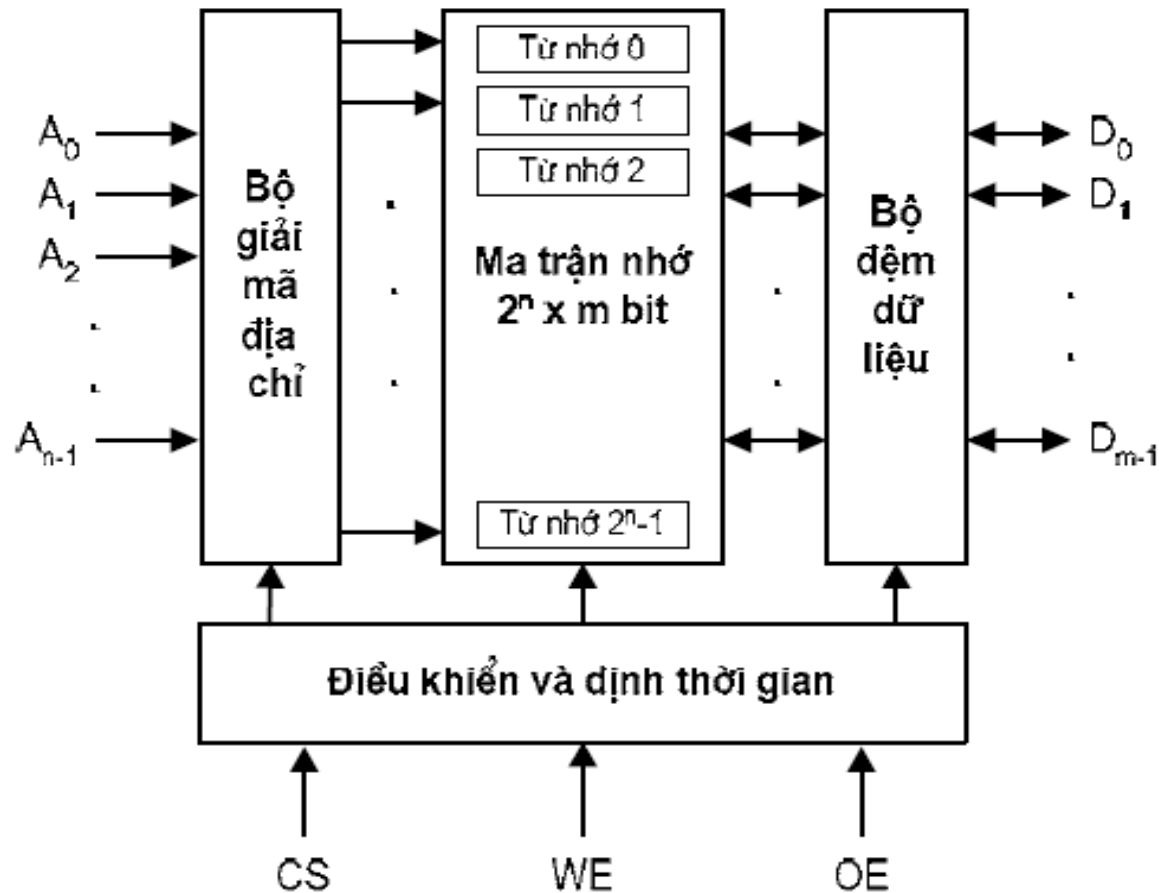
### ■ Sơ đồ cơ bản của chip nhớ



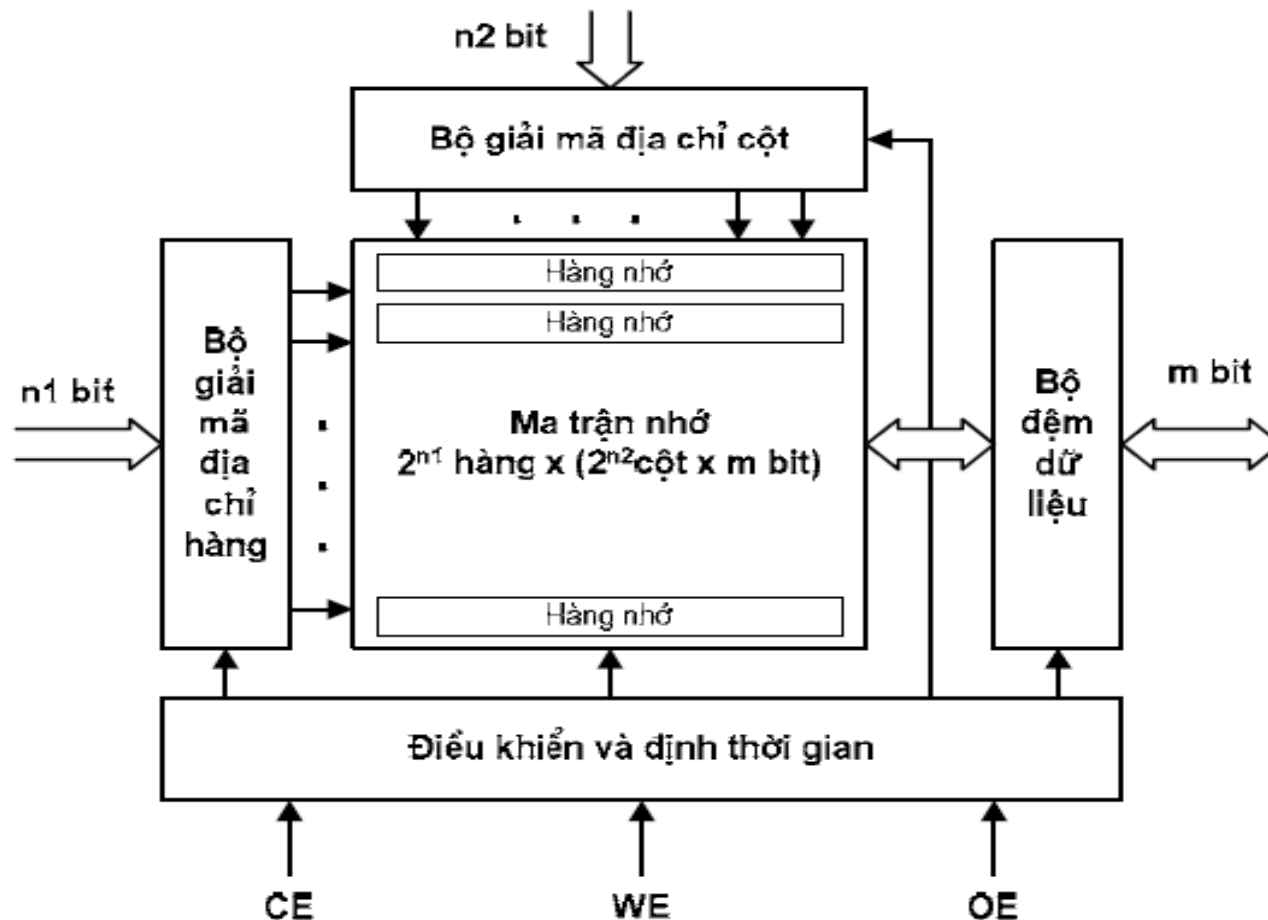
# Các tín hiệu của chip nhớ

- Các đường địa chỉ:  $A_{n-1} \div A_0 \rightarrow$  có  $2^n$  từ nhớ
- Các đường dữ liệu:  $D_{m-1} \div D_0 \rightarrow$  độ dài từ nhớ = m bit
- Dung lượng chip nhớ =  $2^n \times m$  bit
- Các đường điều khiển:
  - Tín hiệu chọn chip CS (Chip Select)
  - Tín hiệu điều khiển đọc OE (Output Enable)
  - Tín hiệu điều khiển ghi WE (Write Enable)(Các tín hiệu điều khiển thường tích cực với mức 0)

# Tổ chức bộ nhớ một chiều



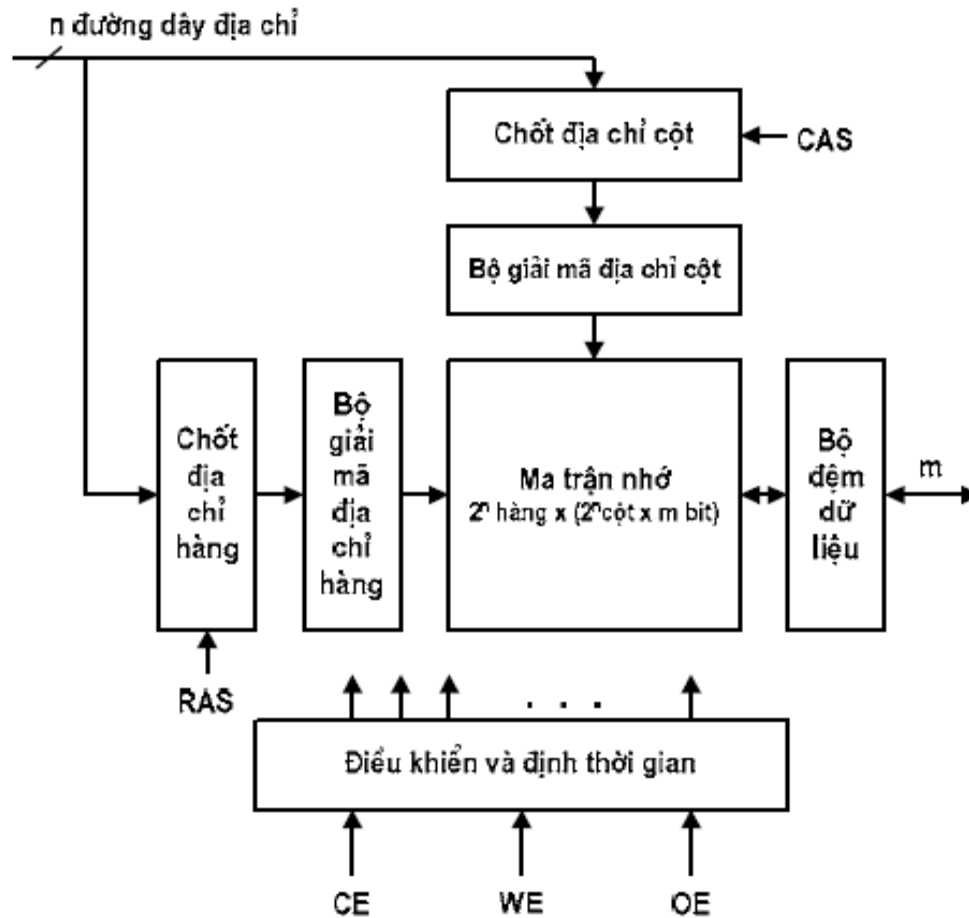
# Tổ chức bộ nhớ hai chiều



# Tổ chức bộ nhớ hai chiều

- Có  $n$  đường địa chỉ:  $n = n_1 + n_2$ 
  - $2^{n_1}$  hàng,
  - mỗi hàng có  $2^{n_2}$  từ nhớ,
- Có  $m$  đường dữ liệu:
  - mỗi từ nhớ có độ dài  $m$ -bit.
- Dung lượng của chip nhớ:  
 $[2^{n_1} \times (2^{n_2} \times m)] \text{ bit} = (2^{n_1+n_2} \times m) \text{ bit} = (2^n \times m) \text{ bit}.$
- Hoạt động giải mã địa chỉ:
  - Bước 1: bộ giải mã hàng chọn 1 trong  $2^{n_1}$  hàng.
  - Bước 2: bộ giải mã cột chọn 1 trong  $2^{n_2}$  từ nhớ (cột) của hàng đã được chọn.

# Tổ chức của DRAM

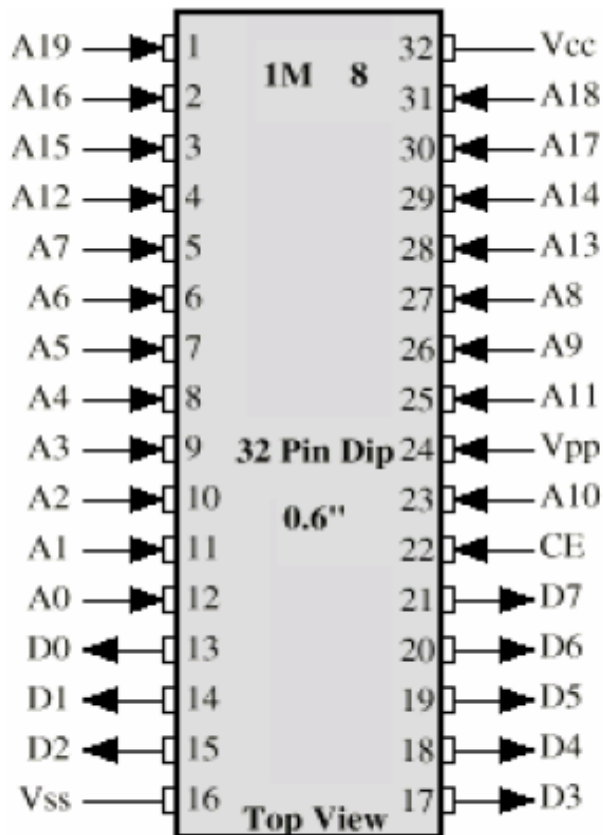


# Tổ chức của DRAM

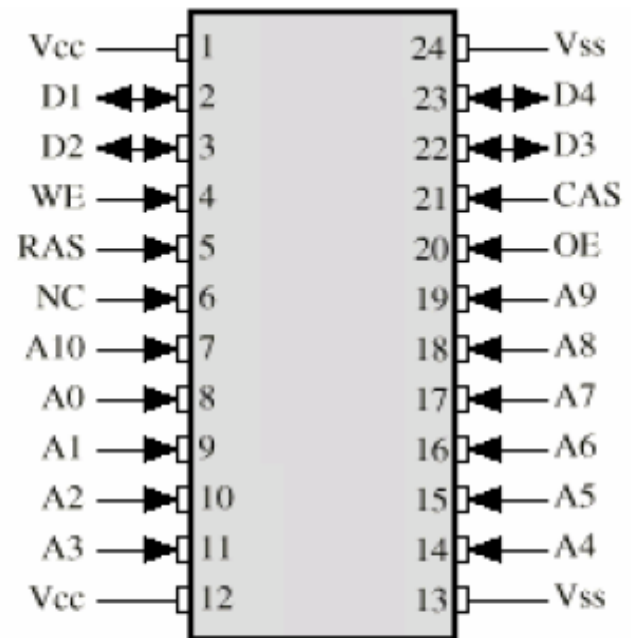
- Dùng  $n$  đường địa chỉ dẫn kênh  $\rightarrow$  cho phép truyền  $2^n$  bit địa chỉ
- Tín hiệu chọn địa chỉ hàng RAS (Row Address Select)
- Tín hiệu chọn địa chỉ cột CAS (Column Address Select)
- Dung lượng của DRAM =  $2^{2n} \times m$  bit



# Chip nhớ

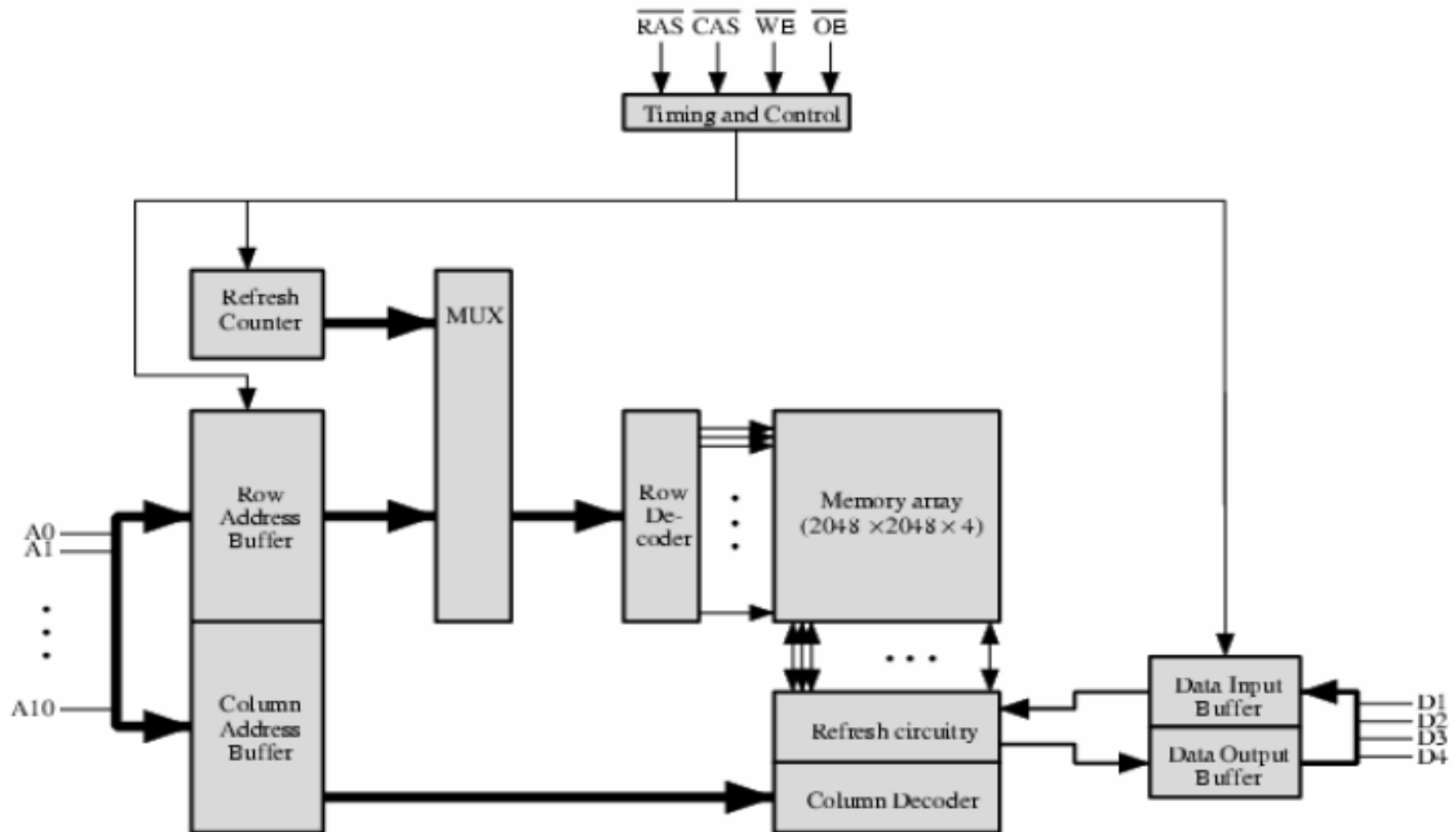


(a) 8 Mbit EPROM

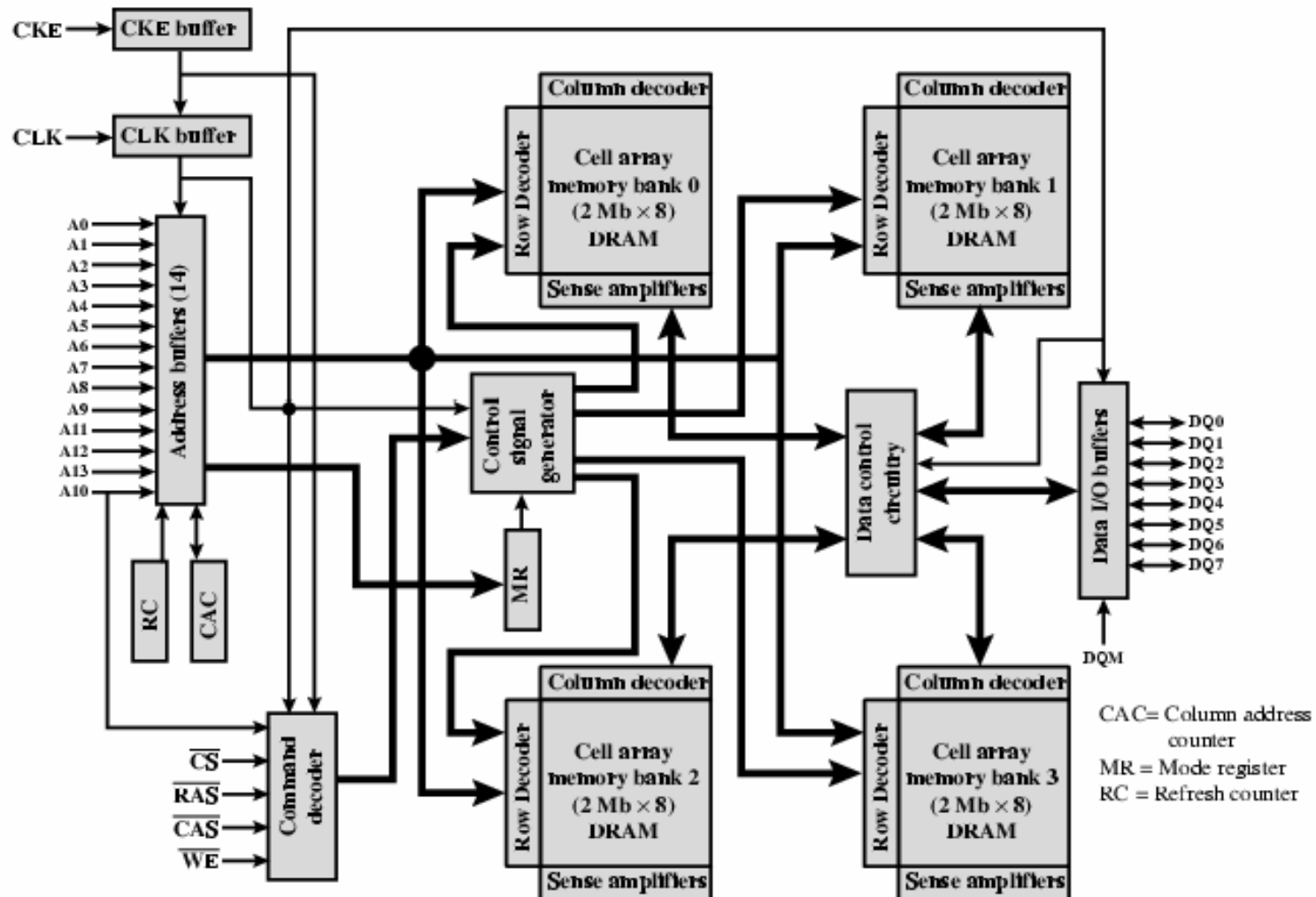


(b) 16 Mbit DRAM

# Ví dụ chip nhớ 16 Mb DRAM (4M x 4)



# IBM 64Mb SDRAM



### 3. Thiết kế mô-đun nhớ bán dẫn

- Dung lượng chip nhớ  $2^n \times m$  bit
- Cần thiết kế để tăng dung lượng:
  - Thiết kế tăng độ dài từ nhớ
  - Thiết kế tăng số lượng từ nhớ
  - Thiết kế kết hợp

# Tăng độ dài từ nhớ

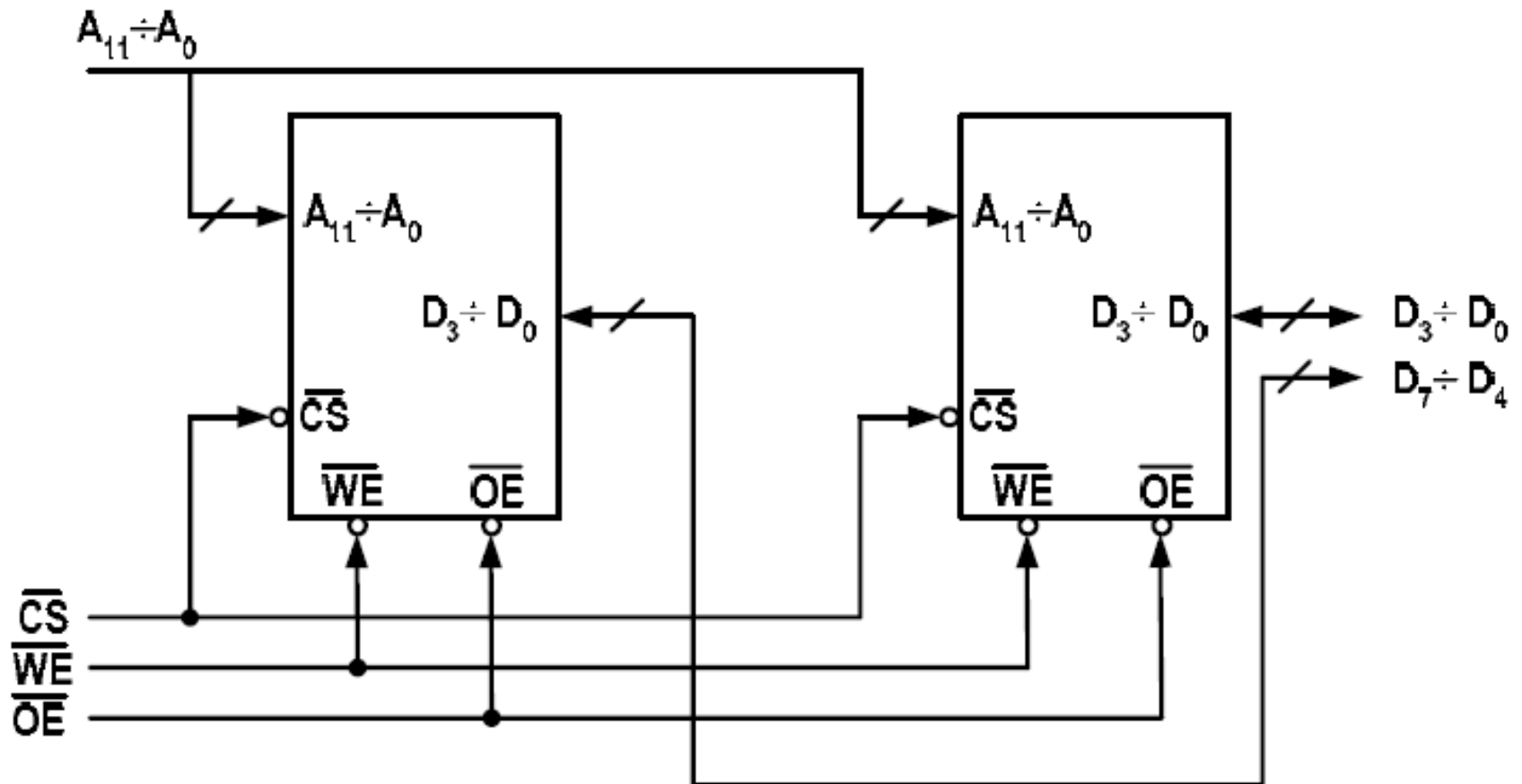
VD1:

- Cho chip nhớ SRAM 4K x 4 bit
- Thiết kế mô-đun nhớ 4K x 8 bit

Giải:

- Dung lượng chip nhớ =  $2^{12}$  x 4 bit
- chip nhớ có:
  - 12 chân địa chỉ
  - 4 chân dữ liệu
- mô-đun nhớ cần có:
  - 12 chân địa chỉ
  - 8 chân dữ liệu

# Ví dụ tăng độ dài từ nhớ



# Bài toán tăng độ dài từ nhớ tổng quát

- Cho chip nhớ  $2^n \times \text{mbit}$
- Thiết kế mô-đun nhớ  $2^n \times (\text{k.m}) \text{ bit}$
- Dùng  $k$  chip nhớ

# Tăng số lượng từ nhớ

VD2:

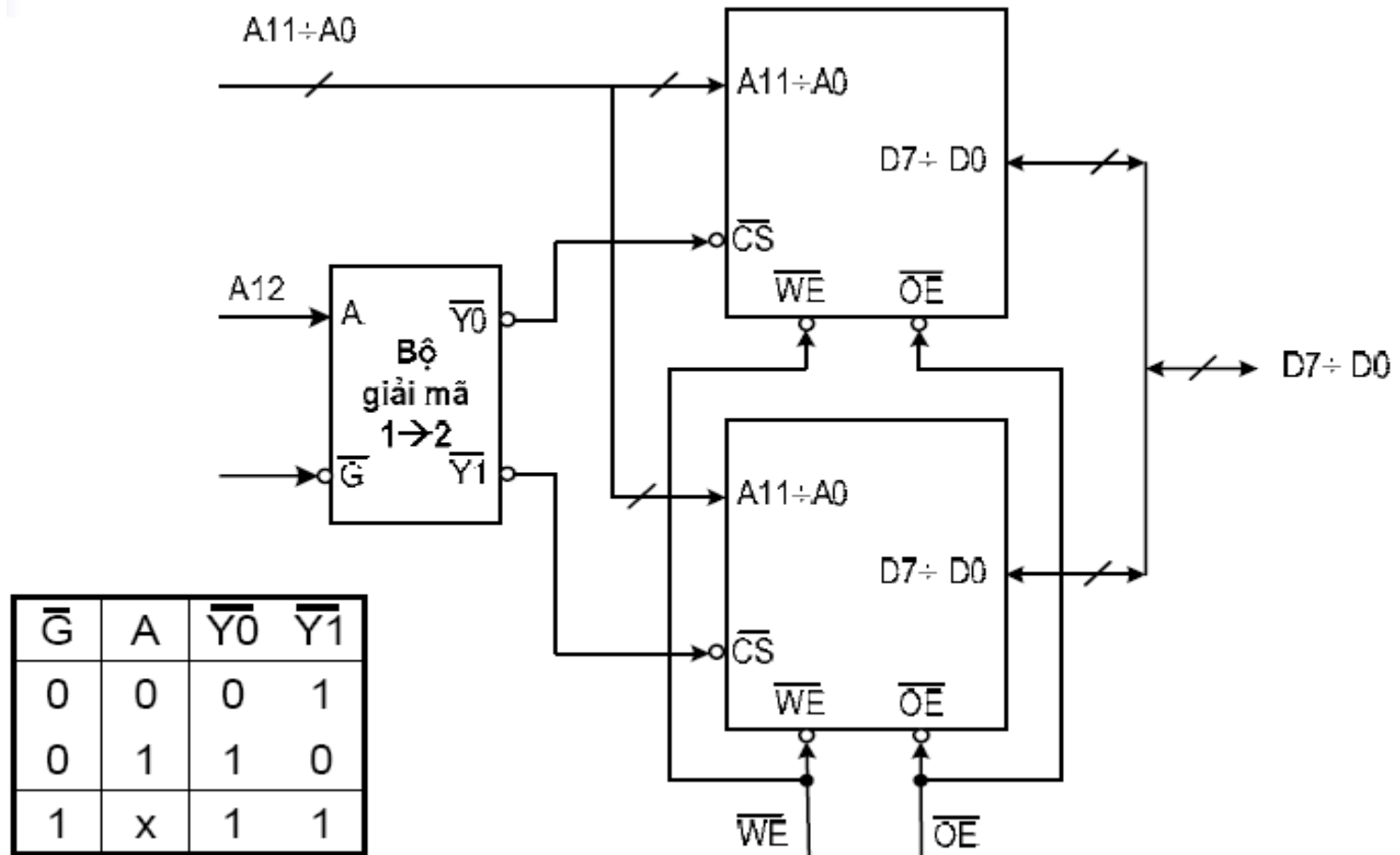
- Cho chip nhớ SRAM 4K x 8 bit
- Thiết kế mô-đun nhớ 8K x 8 bit

Giải:

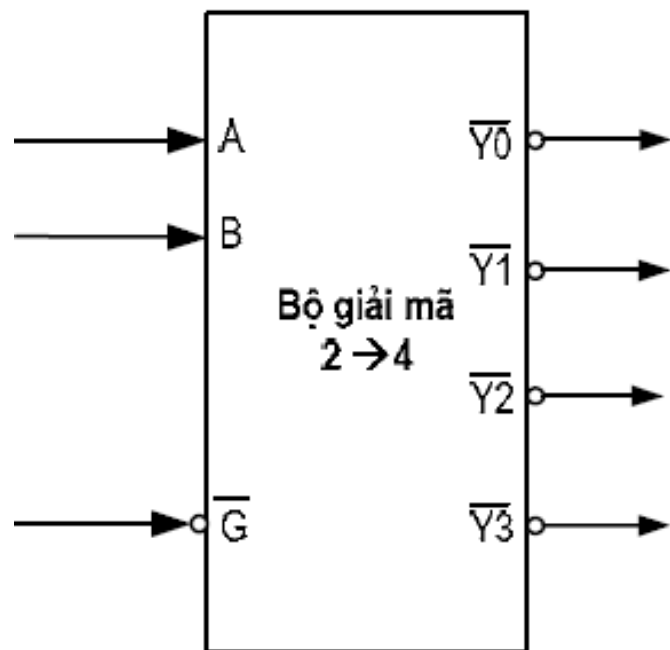
- Dung lượng chip nhớ =  $2^{12} \times 8$  bit
- chip nhớ có:
  - 12 chân địa chỉ
  - 8 chân dữ liệu
- Dung lượng mô-đun nhớ  $2^{13} \times 8$ 
  - 13 chân địa chỉ
  - 8 chân dữ liệu



# Tăng số lượng từ nhớ



# Bộ giải mã $2 \rightarrow 4$



$\overline{G}$	B	A	$\overline{Y_0}$	$\overline{Y_1}$	$\overline{Y_2}$	$\overline{Y_3}$
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0
1	x	x	1	1	1	1

# Bài tập

## 1. Tăng số lượng từ gấp 4 lần:

- Cho chip nhớ SRAM 4K x 8 bit
- Thiết kế mô-đun nhớ 16K x 8 bit

## 2. Tăng số lượng từ gấp 8 lần:

- Cho chip nhớ SRAM 4K x 8 bit
- Thiết kế mô-đun nhớ 32K x 8 bit

## 3. Thiết kế kết hợp:

- Cho chip nhớ SRAM 4K x 4 bit
- Thiết kế mô-đun nhớ 8K x 8 bit

## 4.3. Bộ nhớ chính

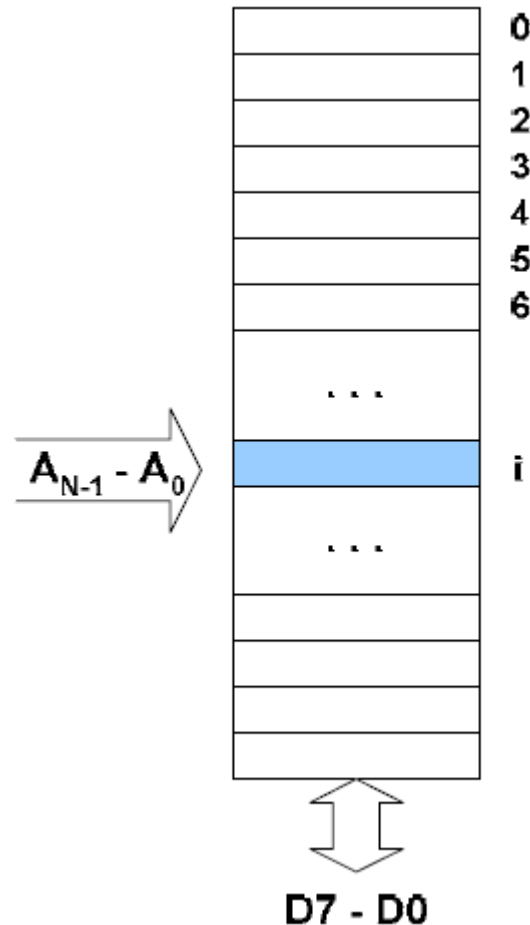
### 1. Các đặc trưng cơ bản

- Chứa các chương trình đang thực hiện và các dữ liệu đang được sử dụng
- Tồn tại trên mọi hệ thống máy tính
- Bao gồm các ngăn nhớ được đánh địa chỉ trực tiếp bởi CPU
- Dung lượng của bộ nhớ chính nhỏ hơn không gian địa chỉ bộ nhớ mà CPU có khả năng quản lý.
- Việc quản lý tham chiếu logic của bộ nhớ chính tùy thuộc vào hệ điều hành

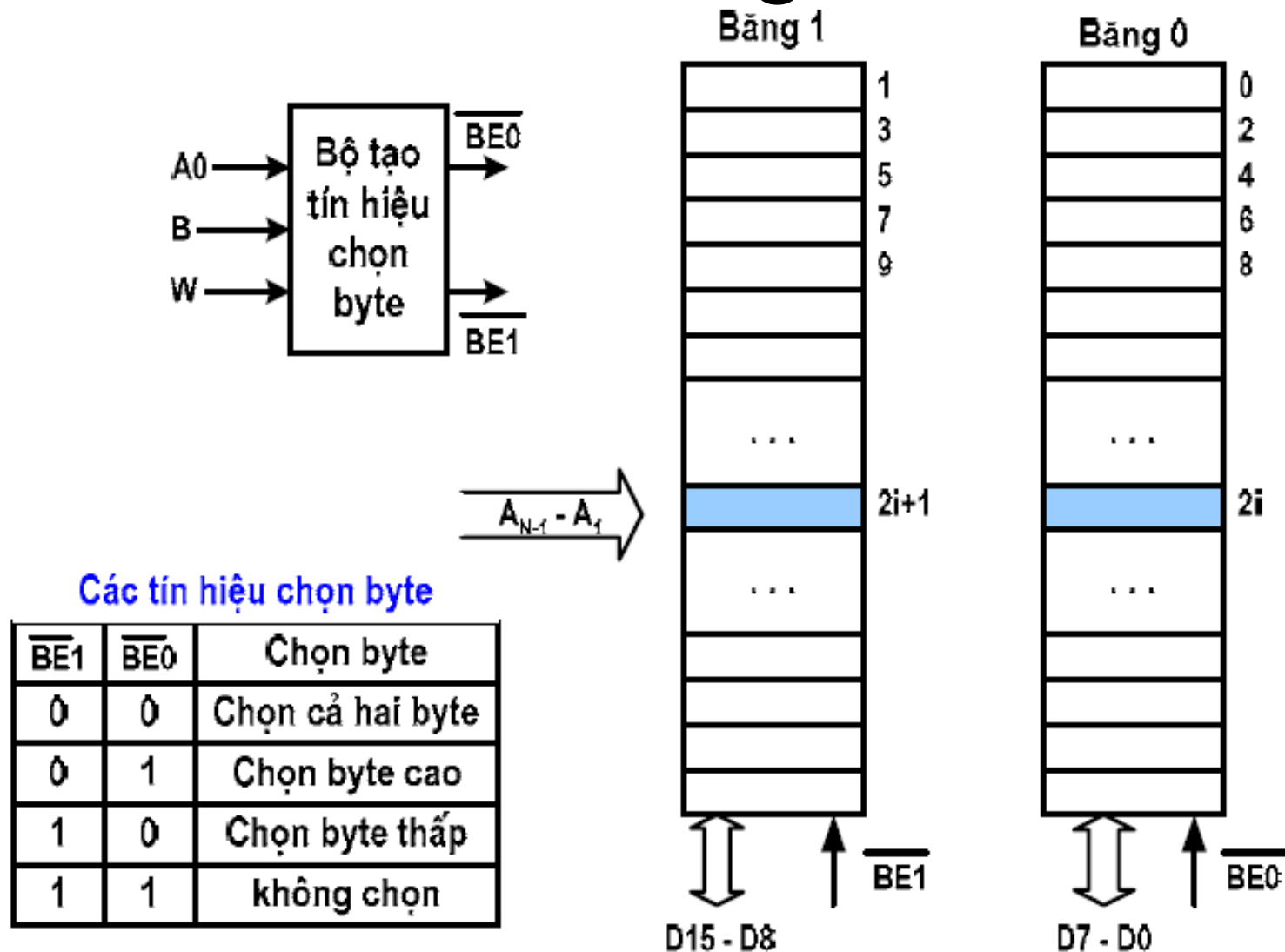
## 2. Tổ chức bộ nhớ đan xen (interleaved memory)

- Độ rộng của bus dữ liệu để trao đổi với bộ nhớ:  $m = 8, 16, 32, 64, 128 \dots$  Bit
- Các ngăn nhớ được tổ chức theo byte  
→ tổ chức bộ nhớ vật lý khác nhau

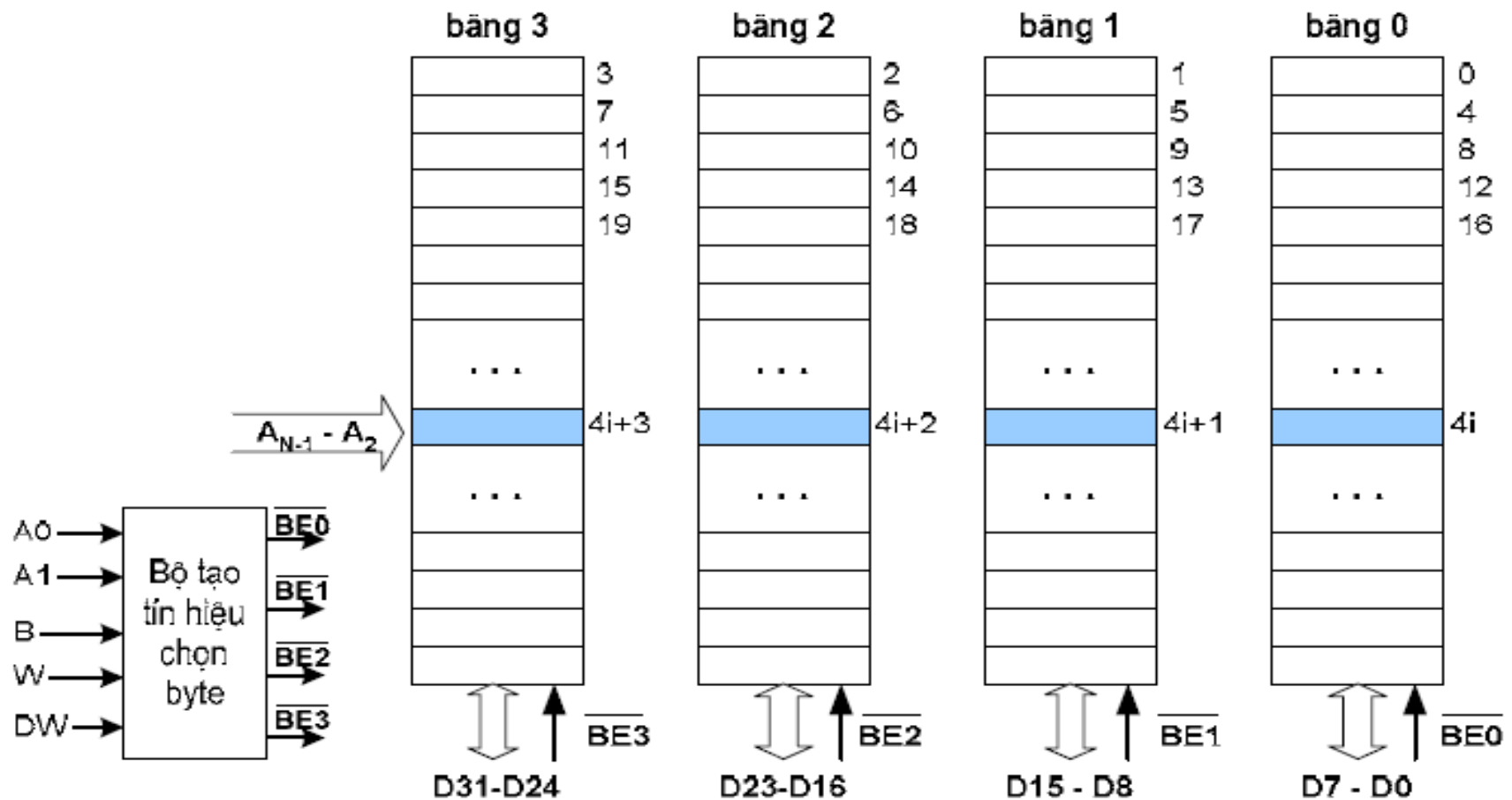
$m=8\text{bit} \rightarrow$  một bảng nhớ tuyến tính



$m=16\text{bit} \rightarrow$  hai bảng nhớ đan xen

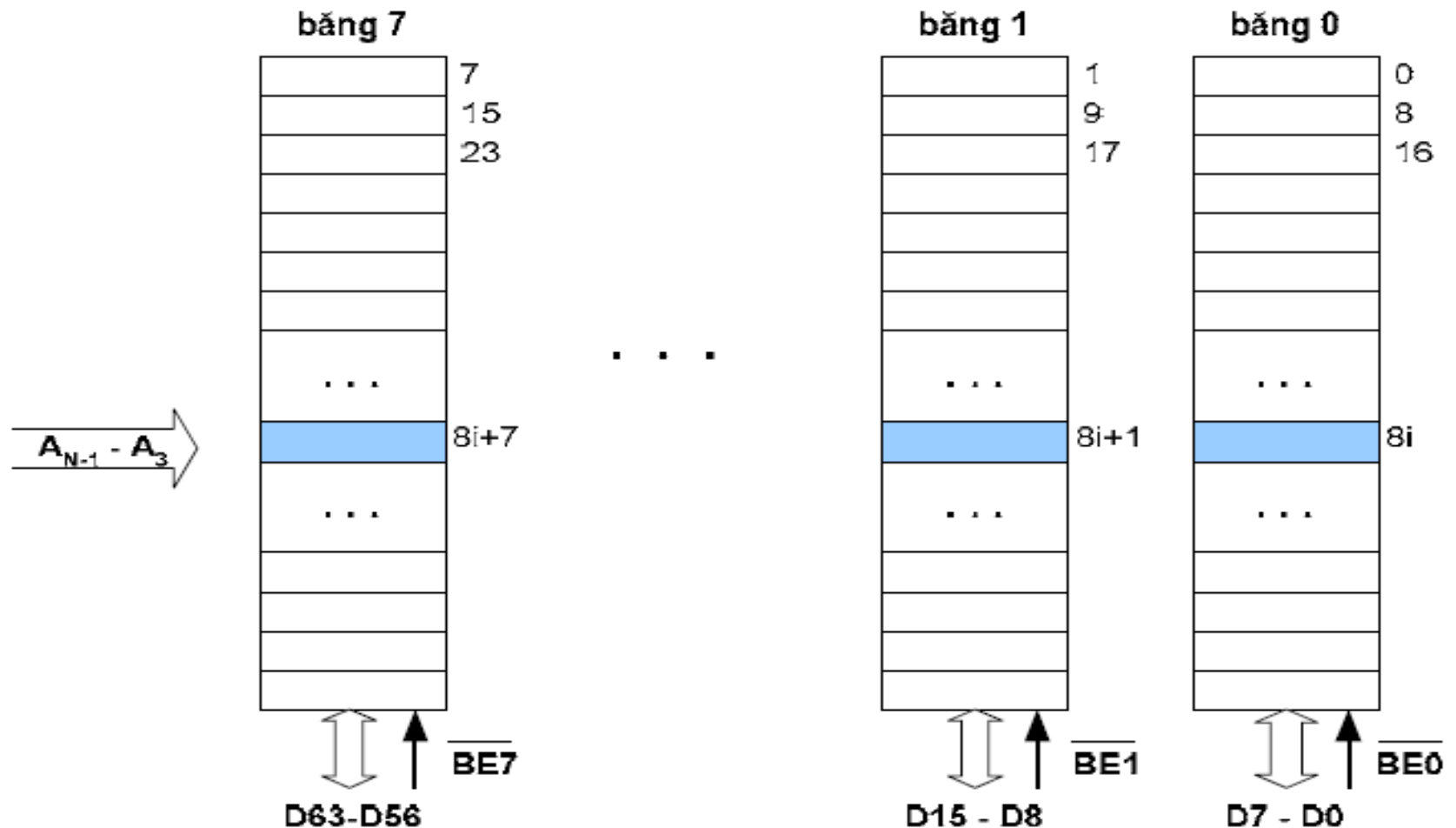


$m=32\text{bit} \rightarrow$  bốn băng nhớ đơn xen





$m=64\text{bit} \rightarrow$  tám băng nhớ đơn xen



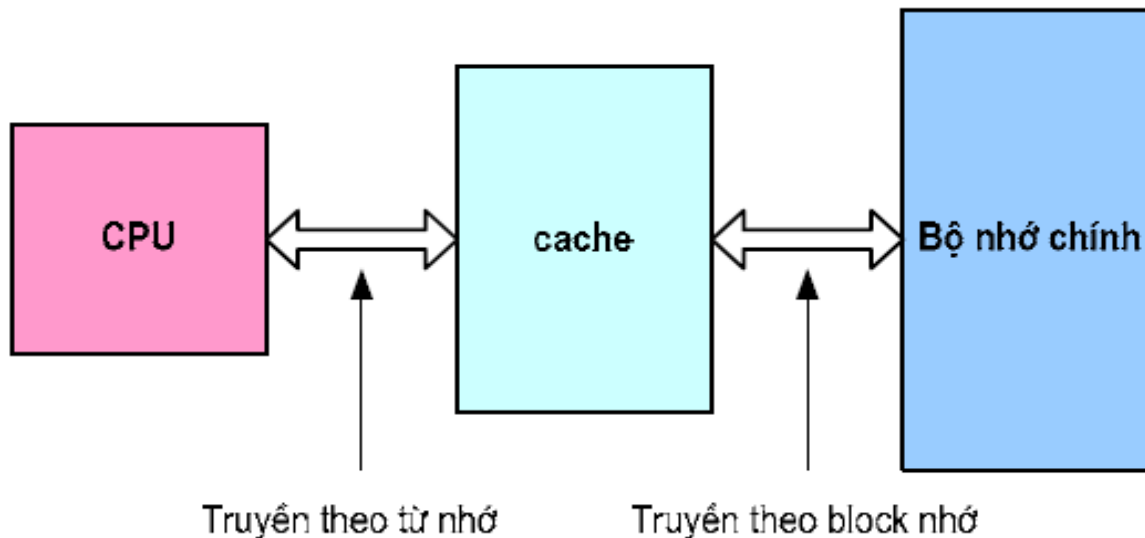
## 4.4. Bộ nhớ đệm nhanh (cache memory)

### 1. Nguyên tắc chung của cache

- Nguyên lý cục bộ hoá tham chiếu bộ nhớ: Trong một khoảng thời gian đủ nhỏ CPU thường chỉ tham chiếu các thông tin trong một khối nhớ cục bộ
- Ví dụ:
  - Cấu trúc chương trình tuần tự
  - Vòng lặp có thân nhỏ
  - Cấu trúc dữ liệu mảng

# Nguyên tắc chung của cache (tiếp)

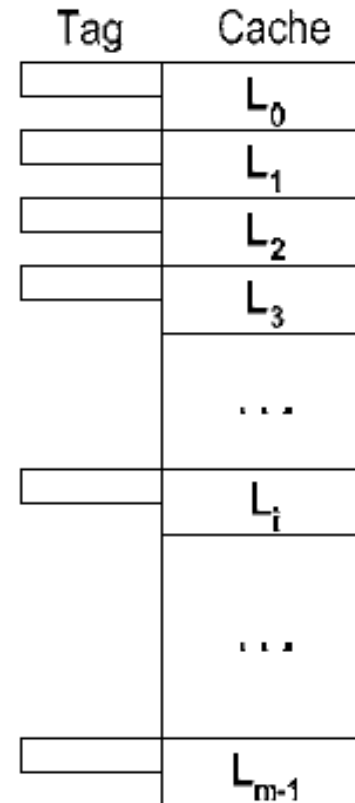
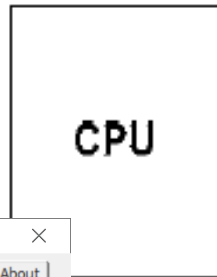
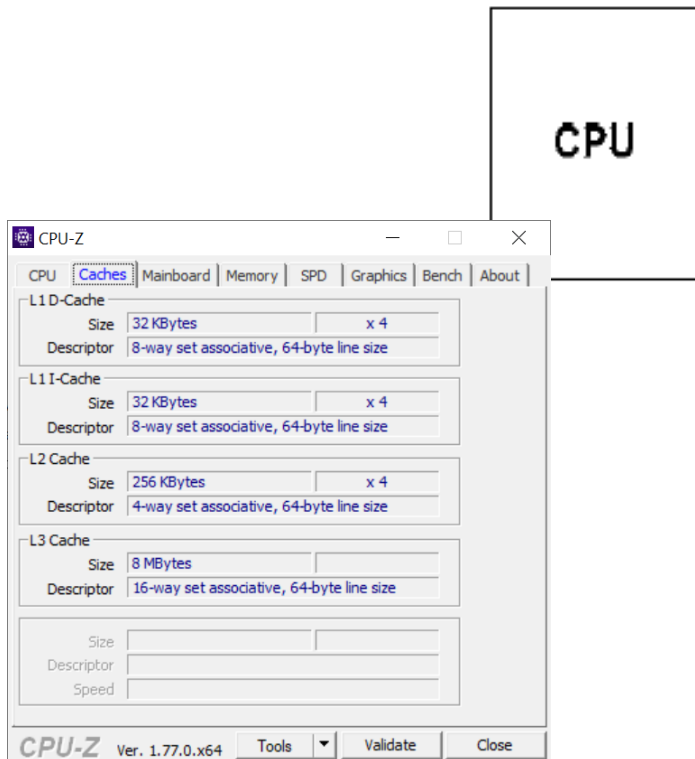
- Cache có tốc độ nhanh hơn bộ nhớ chính
- Cache được đặt giữa CPU và bộ nhớ chính nhằm tăng tốc độ CPU truy cập bộ nhớ
- Cache có thể được đặt trên chip CPU



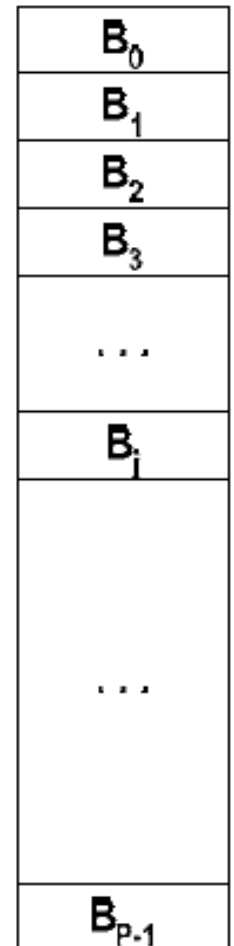
# Ví dụ về thao tác của cache

- CPU yêu cầu nội dung của ngăn nhớ
- CPU kiểm tra trên cache với dữ liệu này
- Nếu có, CPU nhận dữ liệu từ cache(nhanh)
- Nếu không có, đọc Block nhớ chứa dữ liệu từ bộ nhớ chính vào cache
- Tiếp đó chuyển dữ liệu từ cache vào CPU

# Cấu trúc chung của cache / bộ nhớ chính



Bộ nhớ chính



# Cấu trúc chung của cache / bộ nhớ chính (tiếp)

- Bộ nhớ chính có  $2^N$  byte nhớ
- Bộ nhớ chính và cache được chia thành các khối có kích thước bằng nhau
  - Bộ nhớ chính:  $B_0, B_1, B_2, \dots, B_{p-1}$  (p Blocks)
  - Bộ nhớ cache:  $L_0, L_1, L_2, \dots, L_{m-1}$  (m Lines)
  - Kích thước của Block = 8,16,32,64,128 byte

# Cấu trúc chung của cache / bộ nhớ chính (tiếp)

- Một số Block của bộ nhớ chính được nạp vào các Line của cache.
- Nội dung Tag (thẻ nhớ) cho biết Block nào của bộ nhớ chính hiện đang được chứa ở Line đó.
- Khi CPU truy nhập (đọc/ghi) một từ nhớ, có hai khả năng xảy ra:
  - Từ nhớ đó có trong cache (cache hit)
  - Từ nhớ đó không có trong cache (cache miss).

## 2. Các phương pháp ánh xạ

(Chính là các phương pháp tổ chức bộ nhớ cache)

- Ánh xạ trực tiếp

(Direct mapping)

- Ánh xạ liên kết toàn phần

(Fully associative mapping)

- Ánh xạ liên kết tập hợp

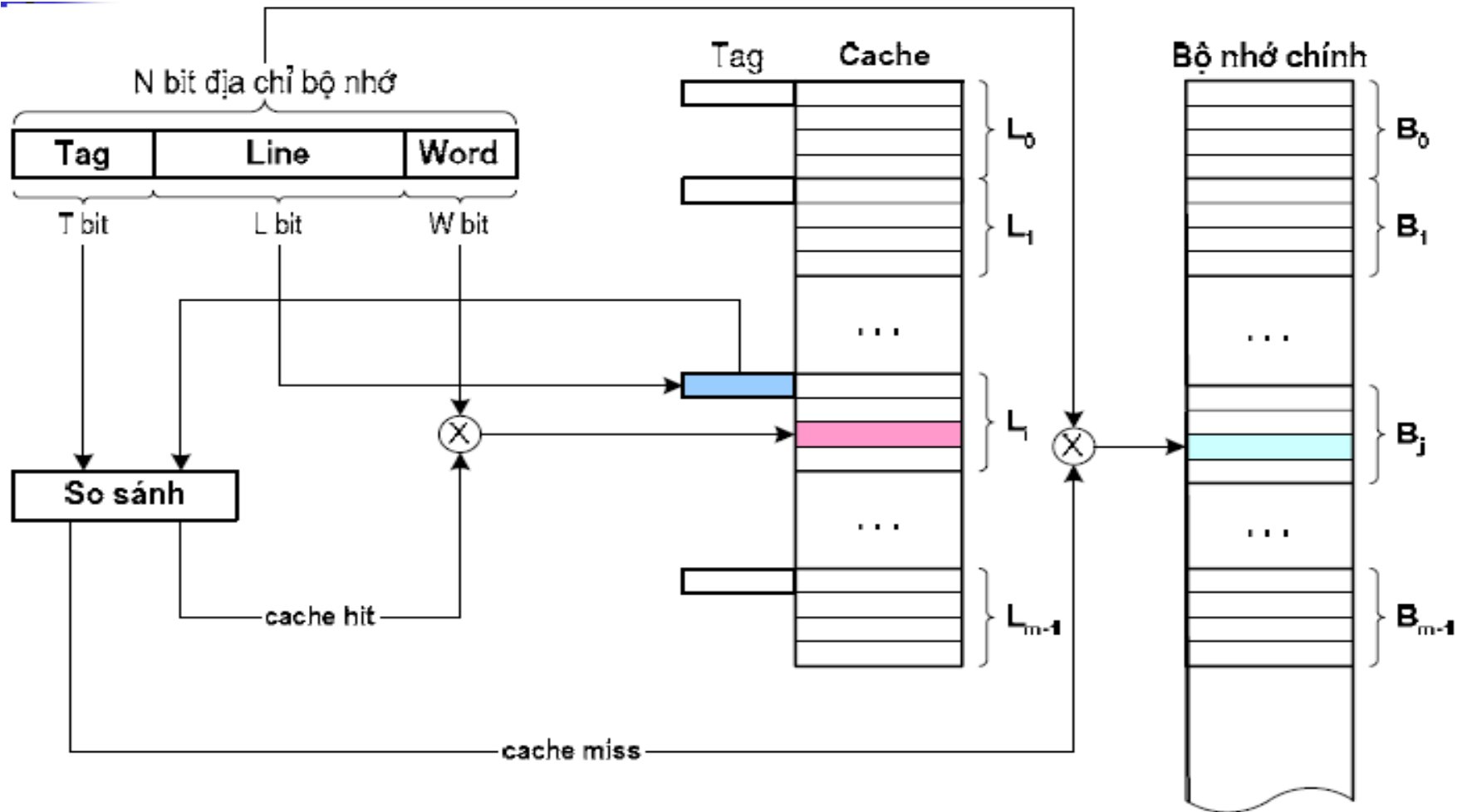
(Set associative mapping)



# Ánh xạ trực tiếp

- Mỗi Block của bộ nhớ chính chỉ có thể được nạp vào một Line của cache:
  - $B_0 \rightarrow L_0$
  - $B_1 \rightarrow L_1$
  - ....
  - $B_{m-1} \rightarrow L_{m-1}$
  - $B_m \rightarrow L_0$
  - $B_{m+1} \rightarrow L_1$
  - ....
- Tổng quát
  - $B_j$  chỉ có thể nạp vào  $L_{j \bmod m}$
  - $m$  là số *Line* của *cache*.

# Minh họa ánh xạ trực tiếp



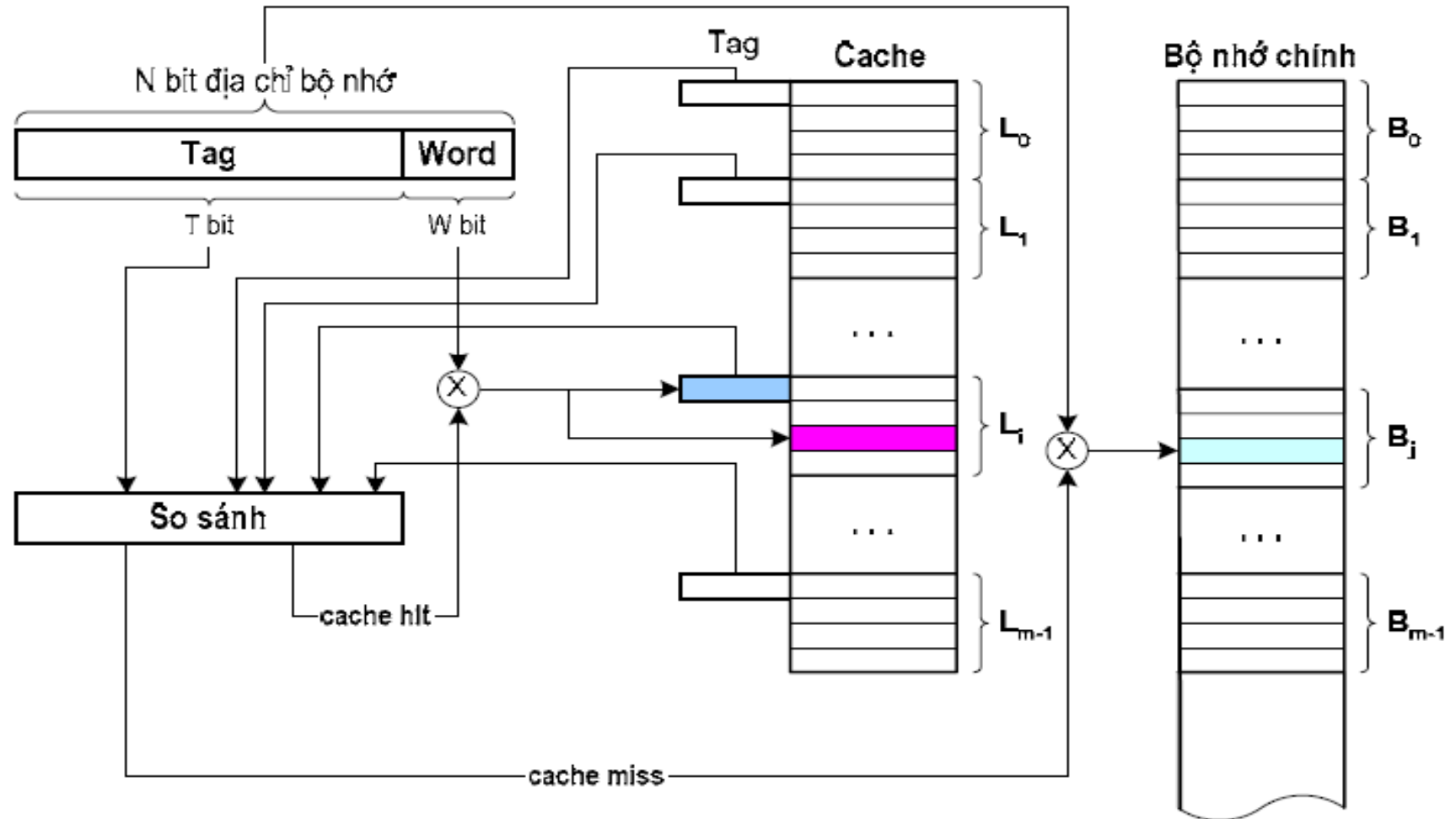
# Đặc điểm của ánh xạ trực tiếp

- Mỗi một địa chỉ  $N$  bit của bộ nhớ chính gồm ba trường:
  - Trường **Word** gồm  $W$  bit xác định một từ nhớ trong *Block* hay *Line*:
$$2^W = \text{kích thước của } Block \text{ hay } Line$$
  - Trường **Line** gồm  $L$  bit xác định một trong số các *Line* trong *cache*:
$$2^L = \text{số } Line \text{ trong } cache = m$$
  - Trường **Tag** gồm  $T$  bit:
$$T = N - (W+L)$$
- Bộ so sánh đơn giản
- Xác suất *cache hit* thấp

# Ánh xạ liên kết toàn phần

- Mỗi *Block* có thể nạp vào bất kỳ *Line* nào của cache.
- Địa chỉ của bộ nhớ chính bao gồm hai trường:
  - Trường Word giống như trường hợp ở trên.
  - Trường Tag dùng để xác định Block của bộ nhớ chính.
- Tag xác định Block đang nằm ở Line đó

# Minh hoạ ánh xạ liên kết toàn phần



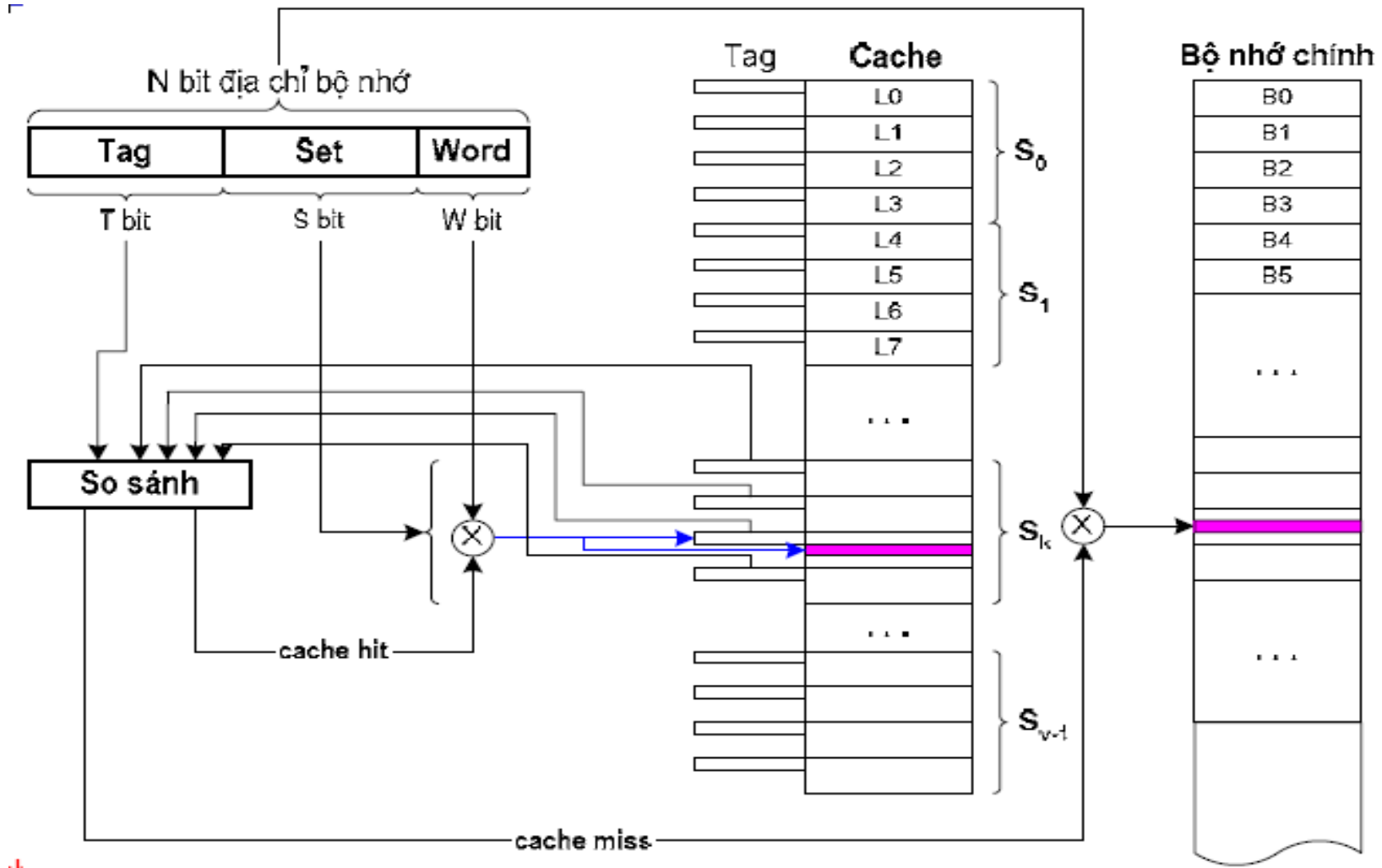
# Đặc điểm của ánh xạ liên kết toàn phần

- So sánh đồng thời với tất cả các Tag → mất nhiều thời gian
- Xác suất cache hit cao.
- Bộ so sánh phức tạp.

# Ánh xạ liên kết tập hợp

- Cache được chia thành các Tập (Set)
- Mỗi một Set chứa một số Line
- Ví dụ:
  - 4 Line/Set  $\rightarrow$  4-way associative mapping
- Ánh xạ theo nguyên tắc sau:
  - $B_0 \rightarrow S_0$
  - $B_1 \rightarrow S_1$
  - $B_2 \rightarrow S_2$
  - .....

# Minh hoạ ánh xạ liên kết tập hợp





# Đặc điểm của ánh xạ liên kết tập hợp

- Kích thước *Block* =  $2^W$  Word
- Trường *Set* có  $S$  bit dùng để xác định một trong số  $V = 2^S$  *Set*
- Trường *Tag* có  $T$  bit:  $T = N - (W+S)$
- Tổng quát cho cả hai phương pháp trên
- Thông thường 2,4,8,16Lines/Set

# Ví dụ về ánh xạ địa chỉ

- Không gian địa chỉ bộ nhớ chính = 4GB
- Dung lượng bộ nhớ cache là 256KB
- Kích thước Line (Block) = 32byte.
- Xác định số bit của các trường địa chỉ cho ba trường hợp tổ chức:
  - Ánh xạ trực tiếp
  - Ánh xạ liên kết toàn phần
  - Ánh xạ liên kết tập hợp 4 đường

# Với ánh xạ trực tiếp

- Bộ nhớ chính = 4GB =  $2^{32}$  byte  $\rightarrow N = 32$  bit
- Cache = 256 KB =  $2^{18}$  byte.
- Line = 32 byte =  $2^5$  byte  $\rightarrow W = 5$  bit
- Số Line trong cache =  $2^{18} / 2^5 = 2^{13}$  Line  
 $\rightarrow L = 13$  bit
- $T = 32 - (13 + 5) = 14$  bit

Tag	Line	Word
14 bit	13 bit	5 bit

# Với ánh xạ liên kết toàn phần

- Bộ nhớ chính = 4GB =  $2^{32}$  byte  $\rightarrow$  N = 32 bit
- *Line* = 32 byte =  $2^5$  byte  $\rightarrow$  W = 5 bit
- Số bit của trường *Tag* sẽ là:  $T = 32 - 5 = 27$  bit

Tag 27 bit	Word 5 bit
---------------	---------------

# Với ánh xạ liên kết tập hợp 4 đường

- Bộ nhớ chính = 4GB =  $2^{32}$  byte  $\rightarrow$  N = 32 bit
- $Line = 32$  byte =  $2^5$  byte  $\rightarrow$  W = 5 bit
- Số  $Line$  trong  $cache = 2^{18} / 2^5 = 2^{13}$   $Line$
- Một  $Set$  có 4  $Line = 2^2$   $Line$   
 $\rightarrow$  số  $Set$  trong  $cache = 2^{13} / 2^2 = 2^{11}$   $Set \rightarrow$   
S = 11 bit
- Số bit của trường  $Tag$  sẽ là:  $T = 32 - (11 + 5)$   
= 16 bit

Tag	Set	Word
16 bit	11 bit	5 bit

### 3. Thuật giải thay thế (1): Ánh xạ trực tiếp

- Không phải lựa chọn
- Mỗi Block chỉ ánh xạ vào một Line xác định
- Thay thế Block ở Line đó

# Thuật giải thay thế (2): Ánh xạ liên kết

- Được thực hiện bằng phần cứng (nhanh)
- Random: Thay thế ngẫu nhiên
- FIFO (First In First Out): Thay thế *Block* nào nằm lâu nhất ở trong *Set* đó
- LFU (Least Frequently Used): Thay thế *Block* nào trong *Set* có số lần truy nhập ít nhất trong cùng một khoảng thời gian
- LRU (Least Recently Used): Thay thế *Block* ở trong *Set* tương ứng có thời gian lâu nhất không được tham chiếu tới.

## 4. Phương pháp ghi dữ liệu khi cache hit

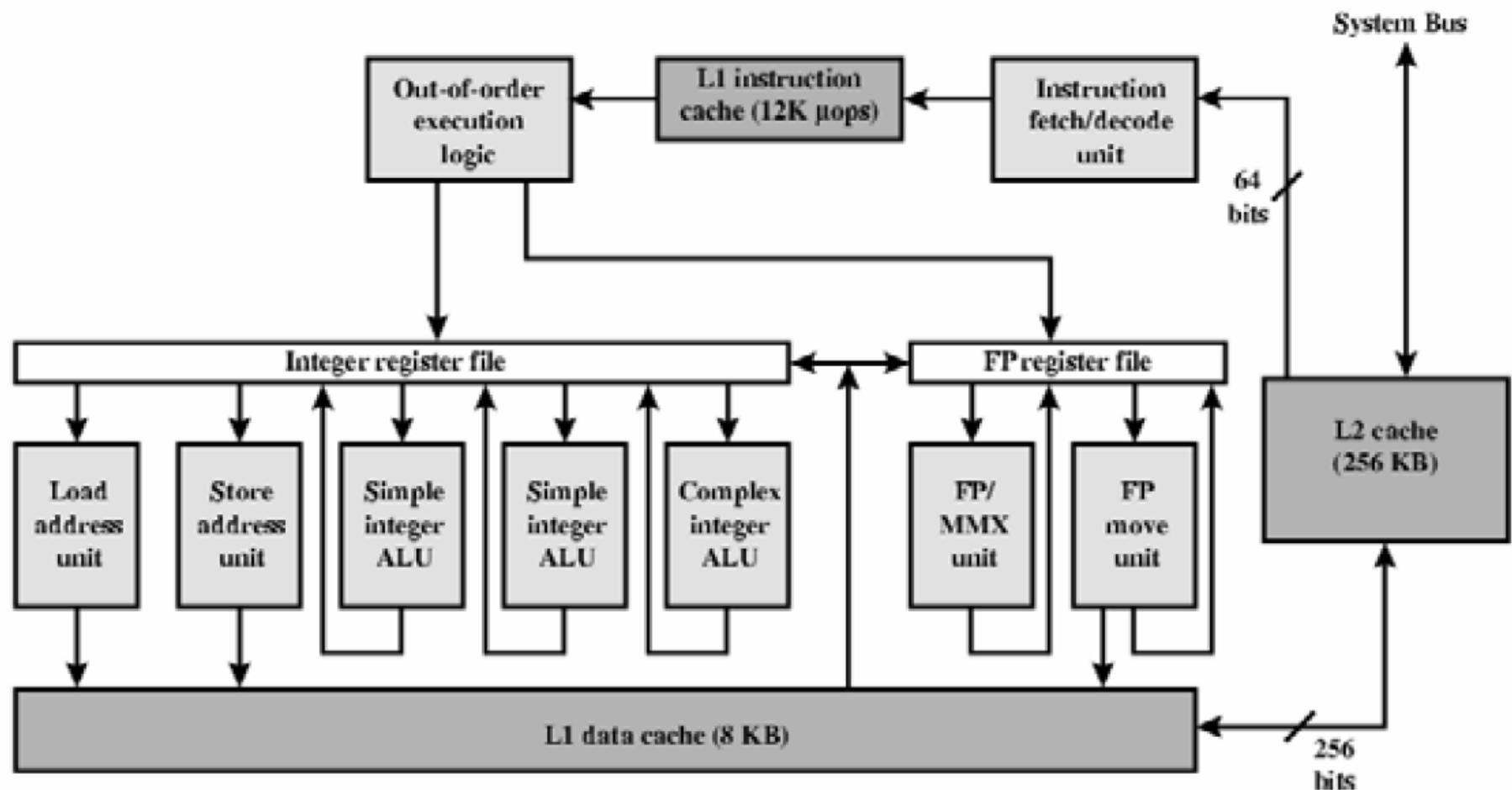
- Ghi xuyên qua (Write-through):
  - ghi cả cache và cả bộ nhớ chính
  - tốc độ chậm
- Ghi trả sau (Write-back):
  - chỉ ghi ra cache
  - tốc độ nhanh
  - khi Block trong cache bị thay thế cần phải ghi trả cả Block về bộ nhớ chính



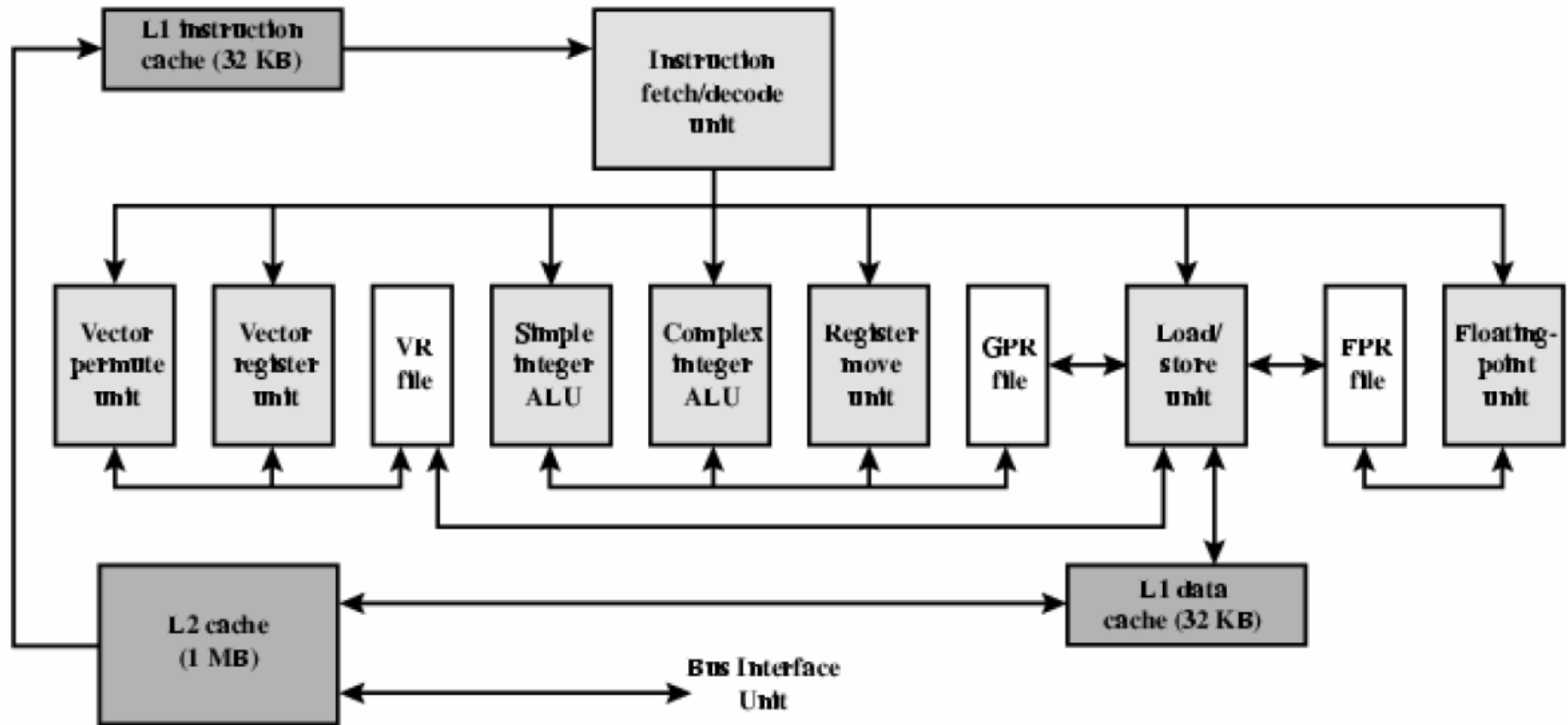
# 5. Cache trên các bộ xử lý Intel

- 80486: 8KB cache L1 trên chip
- Pentium: có hai cache L1 trên chip
  - Cache lệnh = 8KB
  - Cache dữ liệu = 8KB
- Pentium 4 (2000): hai mức cache L1 và L2 trên chip
  - Cache L1:
    - mỗi cache 8KB
    - Kích thước Line = 64 byte
    - ánh xạ liên kết tập hợp 4 đường
  - cache L2
    - 256KB
    - Kích thước Line = 128 byte
    - ánh xạ liên kết tập hợp 8 đường

# Sơ đồ Pentium 4



# PowerPC G4 (dùng cho Power Mac)

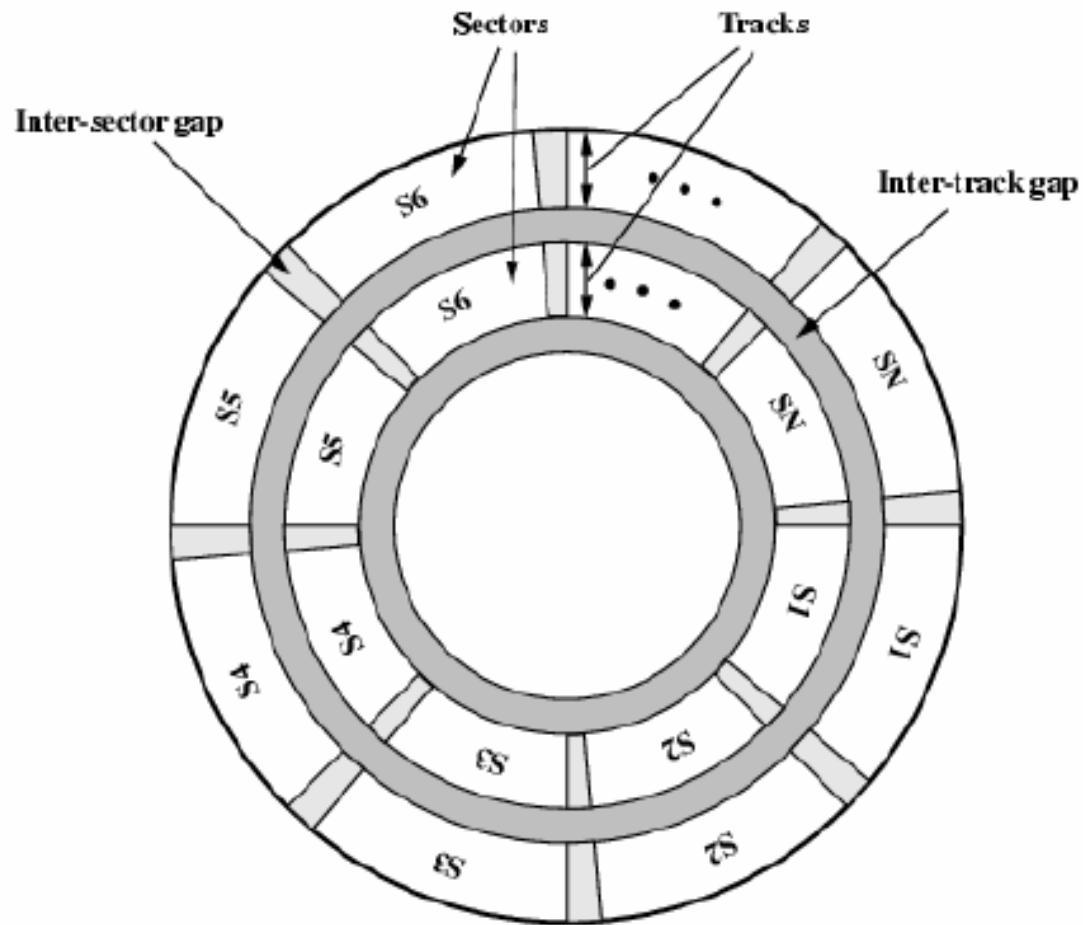


## 4.5. Bộ nhớ ngoài

### 1. Các kiểu bộ nhớ ngoài

- Băng từ
- Đĩa từ
- Đĩa quang
- Flash Disk

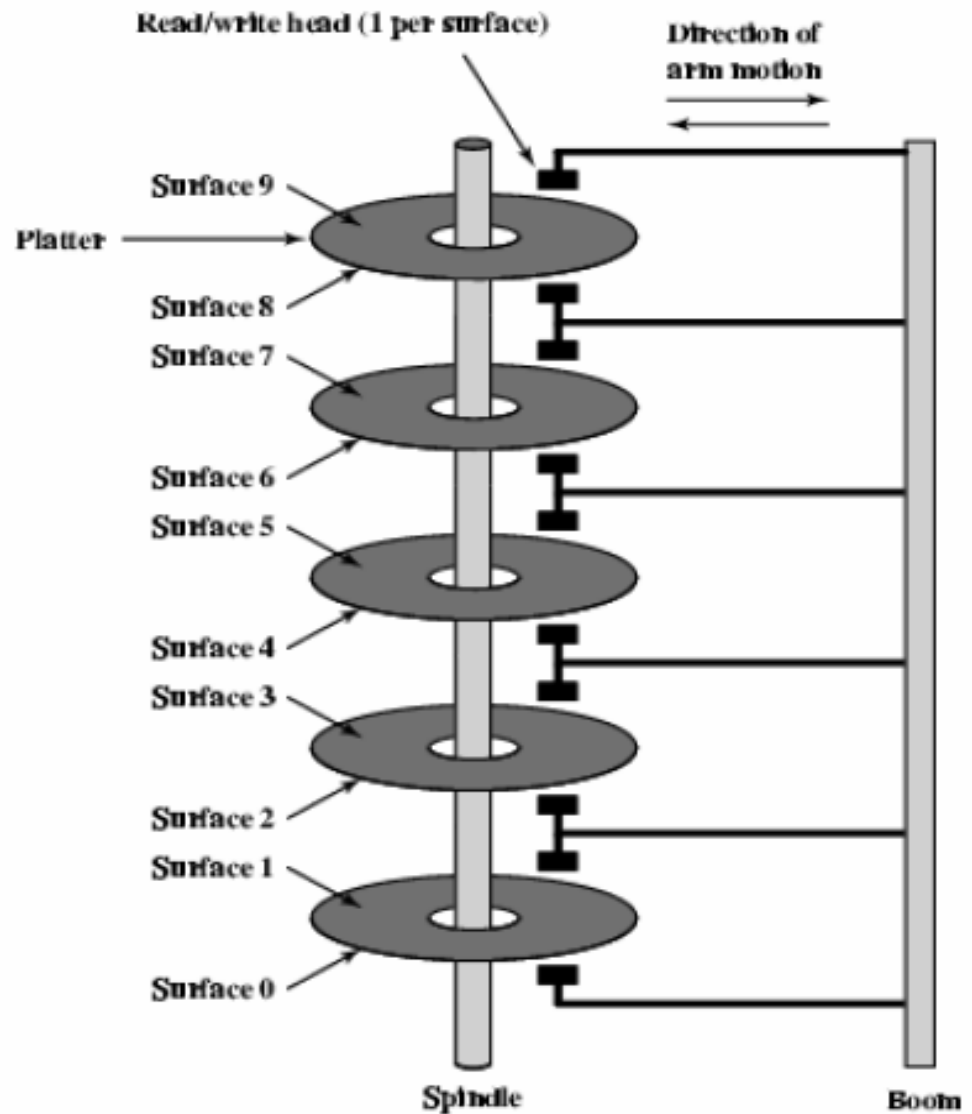
## 2. Đĩa từ



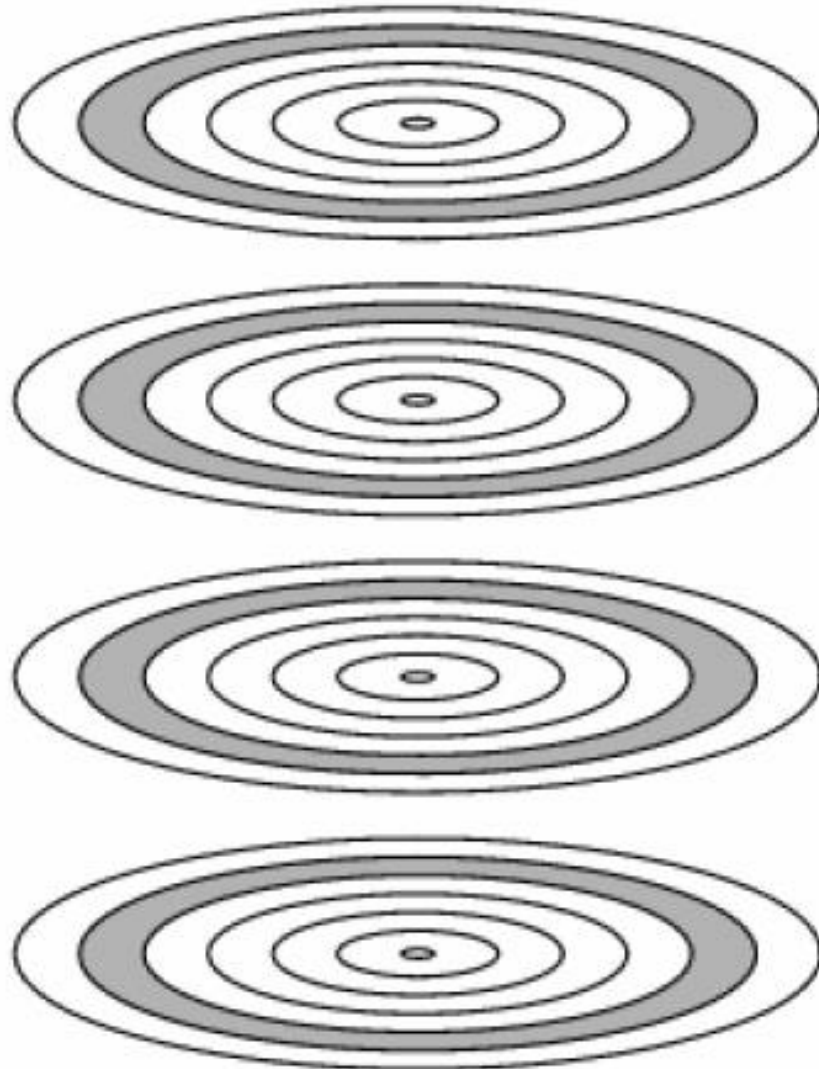
# Các đặc tính đĩa từ

- Đầu từ cố định hay đầu từ di động
- Đĩa cố định hay thay đổi
- Một mặt hay hai mặt
- Một đĩa hay nhiều đĩa
- Cơ chế đầu từ
  - Tiếp xúc (đĩa mềm)
  - Không tiếp xúc

# Nhiều đĩa



# Cylinders





# Đĩa mềm

- 8", 5.25", 3.5"
- Dung lượng nhỏ: chỉ tới 1.44Mbyte
- Tốc độ chậm
- Thông dụng
- Rẻ tiền
- Tương lai có thể không dùng nữa ?

# Đĩa cứng

- Một hoặc nhiều đĩa
- Thông dụng
- Dung lượng tăng lên rất nhanh
  - 1993: 200MB
  - 2004: 30GB, 40GB
  - 2013: 1TB, 2TB
- Tốc độ đọc/ghi nhanh
- Rẻ tiền

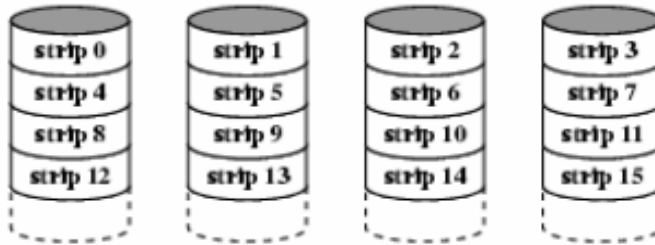
# RAID

- Redundant Array of Inexpensive Disks
- Redundant Array of Independent Disks
- Hệ thống nhớ dung lượng lớn

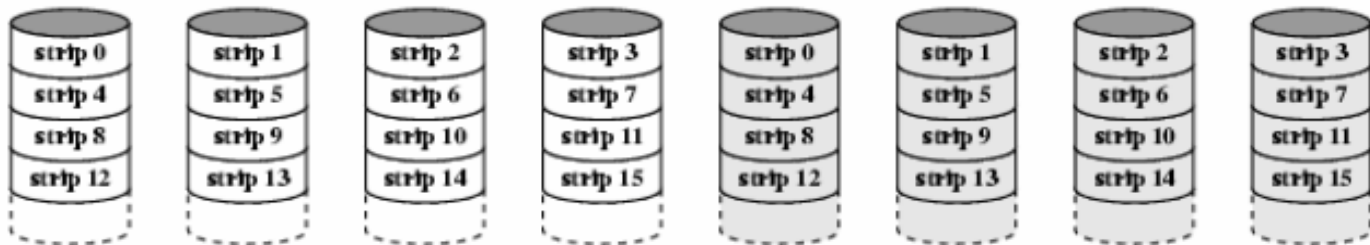
# Đặc điểm của RAID

- Tập các đĩa cứng vật lý được OS coi như một ổ logic duy nhất → dung lượng lớn
- Dữ liệu được lưu trữ phân tán trên các ổ đĩa vật lý → truy cập song song (nhanh)
- Có thể sử dụng dung lượng dư thừa để lưu trữ các thông tin kiểm tra chẵn lẻ, cho phép khôi phục lại thông tin trong trường hợp đĩa bị hỏng → an toàn thông tin
- 7 loại phổ biến (RAID 0 – 6)

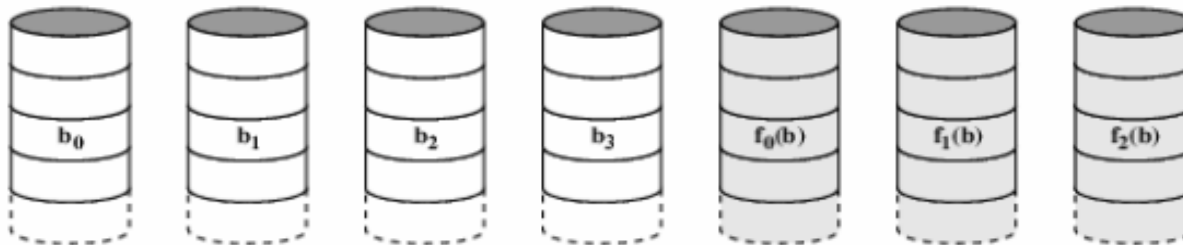
# RAID 0, 1, 2



(a) RAID 0 (non-redundant)

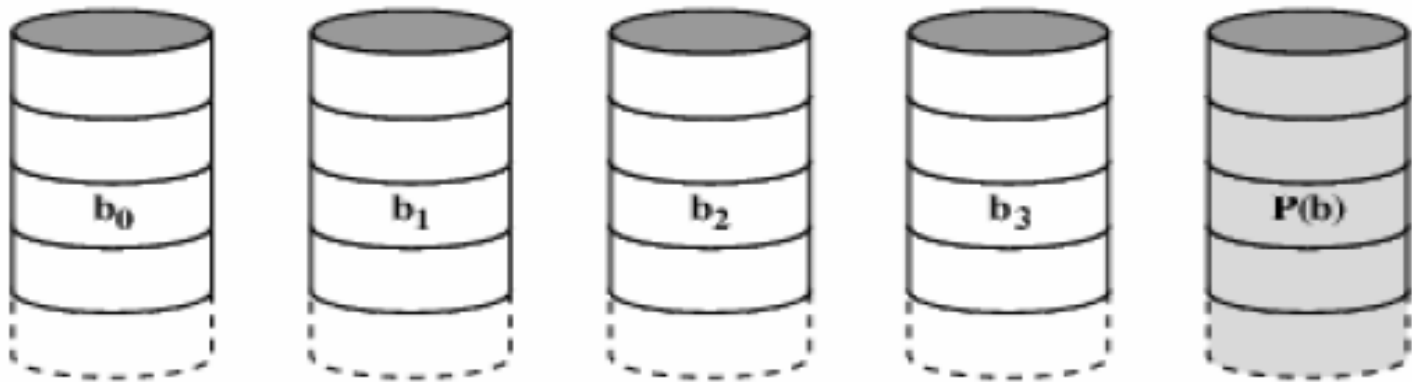


(b) RAID 1 (mirrored)

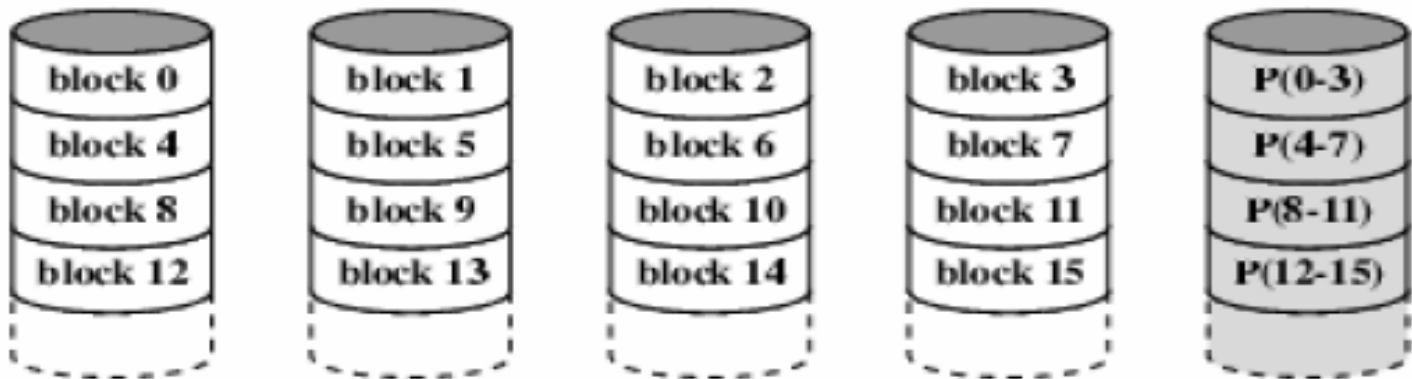


(c) RAID 2 (redundancy through Hamming code)

# RAID 3 & 4

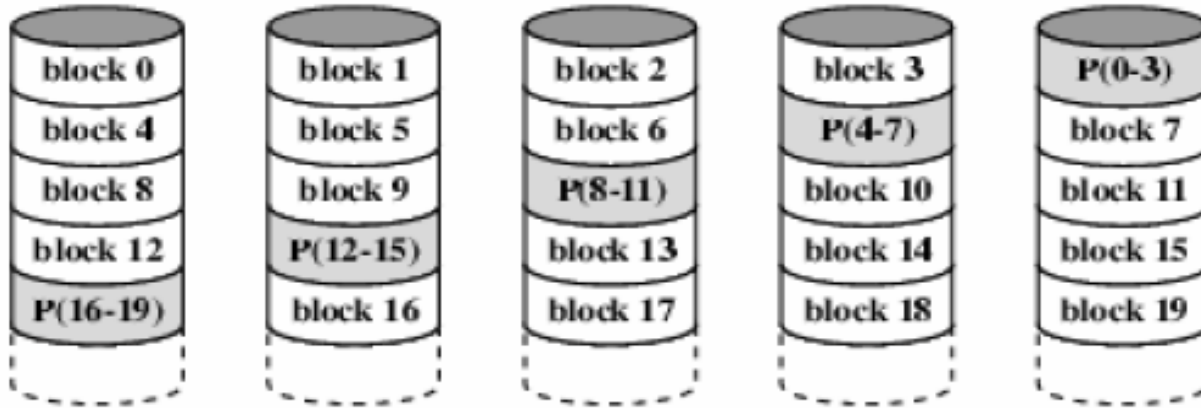


(d) RAID 3 (bit-interleaved parity)

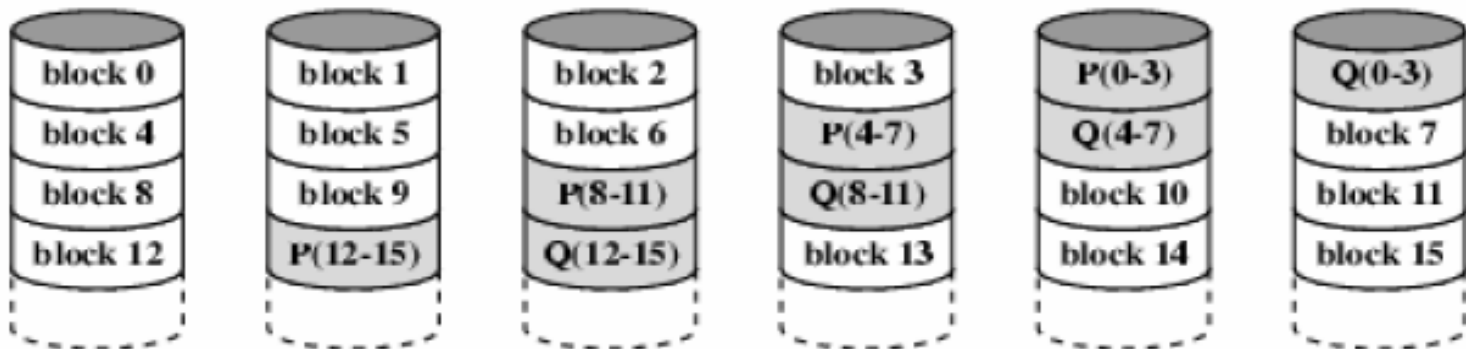


(e) RAID 4 (block-level parity)

# RAID 5 & 6

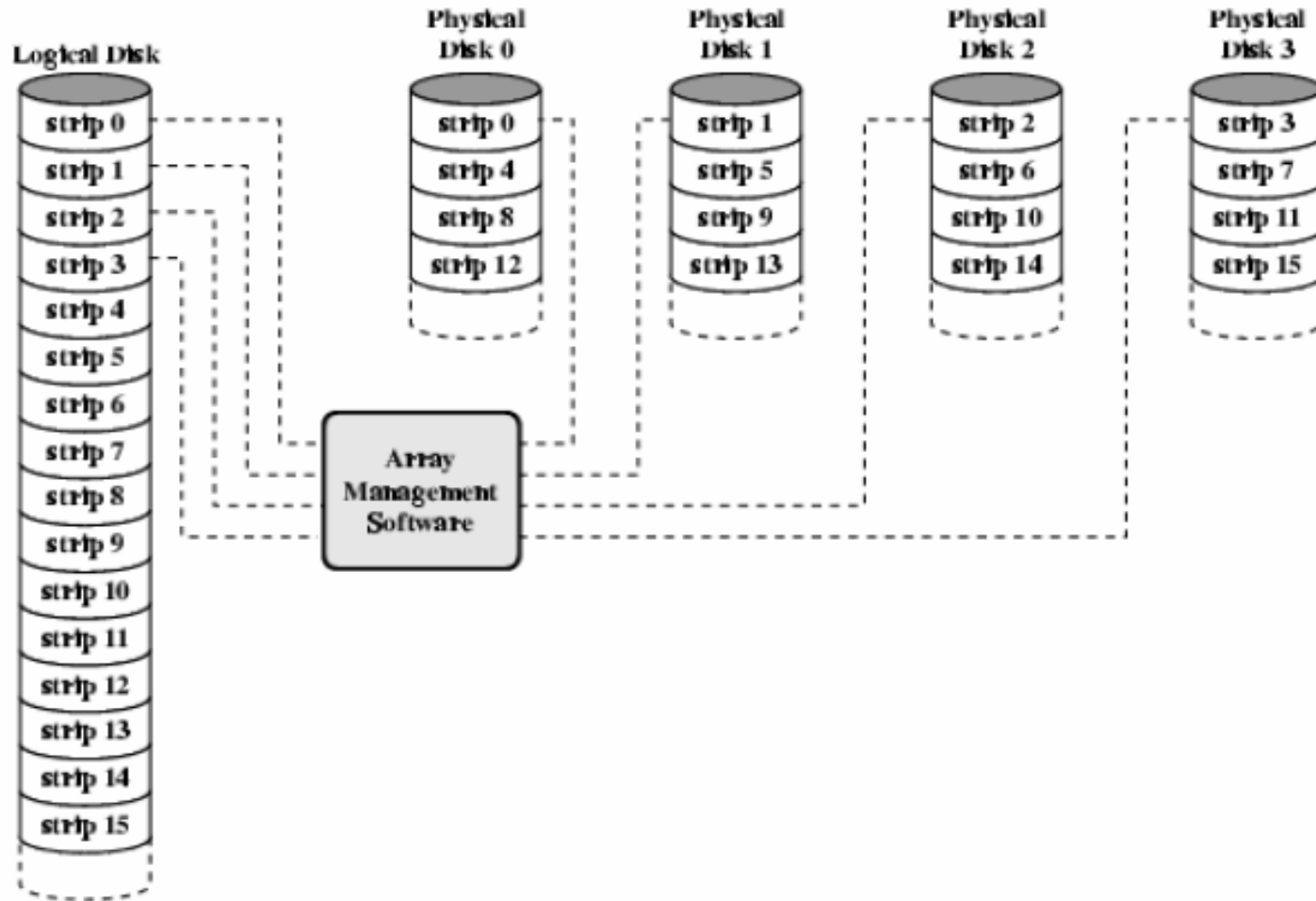


(f) RAID 5 (block-level distributed parity)



(g) RAID 6 (dual redundancy)

# Ảnh xạ dữ liệu của RAID 0





# Cách tính dung lượng đĩa

Dung lượng của ổ đĩa cứng quyết định bởi yếu tố công nghệ:

- Mật độ ghi (bits/inch): Số lượng bit có thể nén được trên độ dài 1 inch của một rãnh.
- Mật độ rãnh (tracks/inch): Số lượng các rãnh có thể nén được trên 1 inch theo hướng bán kính.
- Mật độ tính theo diện tích (bits/inch<sup>2</sup>): tích của mật độ ghi, mật độ rãnh

# Cách tính dung lượng đĩa

- Ứng dụng tính dung lượng của ổ đĩa cứng nếu biết Số byte trên sector là 512, số sector trên rãnh trung bình là 300, số rãnh trên một mặt là 20,000, số mặt trên một đĩa là 2, số đĩa trên ổ đĩa là 5?

Dung lượng = (#bytes/sector) x (#sectors/track<sub>(trung bình)</sub>) x (#tracks/surface) x (#surfaces/platter) x (#platters/disk)

$$\begin{aligned}\text{Dung lượng:} &= 512 \times 300 \times 20000 \times 2 \times 5 \\ &= 30,720,000,000\end{aligned}$$

# 3. Đĩa quang

- CD-ROM (Compact Disk ROM)
- CD-R (Recordable CD)
- CD-RW (Rewriteable CD)
- Dung lượng thông dụng 650MB
- Ổ đĩa CD:
  - Ổ CD-ROM
  - Ổ CD-Writer: Ghi một phiên hoặc ghi nhiều phiên
  - Ổ CD-RW
- Tốc độ đọc cơ sở 150KByte/s.
- Tốc độ bội, ví dụ: 48x, 52x,...

# 3. Đĩa quang (tiếp)

## ■ DVD

- Digital Video Disk: chỉ dùng trên ổ đĩa xem video
- Digital Versatile Disk: ổ trên máy tính
- Ghi một hoặc hai mặt
- Một hoặc hai lớp trên một mặt
- Thông dụng: 4,7GB/lớp

## 4. Flash Disk

- Thường kết nối qua cổng USB
- Không phải dạng đĩa
- Bộ nhớ bán dẫn cực nhanh (flash memory)
- Dung lượng tăng nhanh
- Thuận tiện

# Các dạng flash memory

- Ổ nhớ kết nối qua cổng USB
- Thẻ nhớ
- Ổ SSD (Solid State Drive): kết nối nhiều chip nhớ flash và cho phép truy cập song song



## 4.6. Bộ nhớ ảo (Virtual Memory)

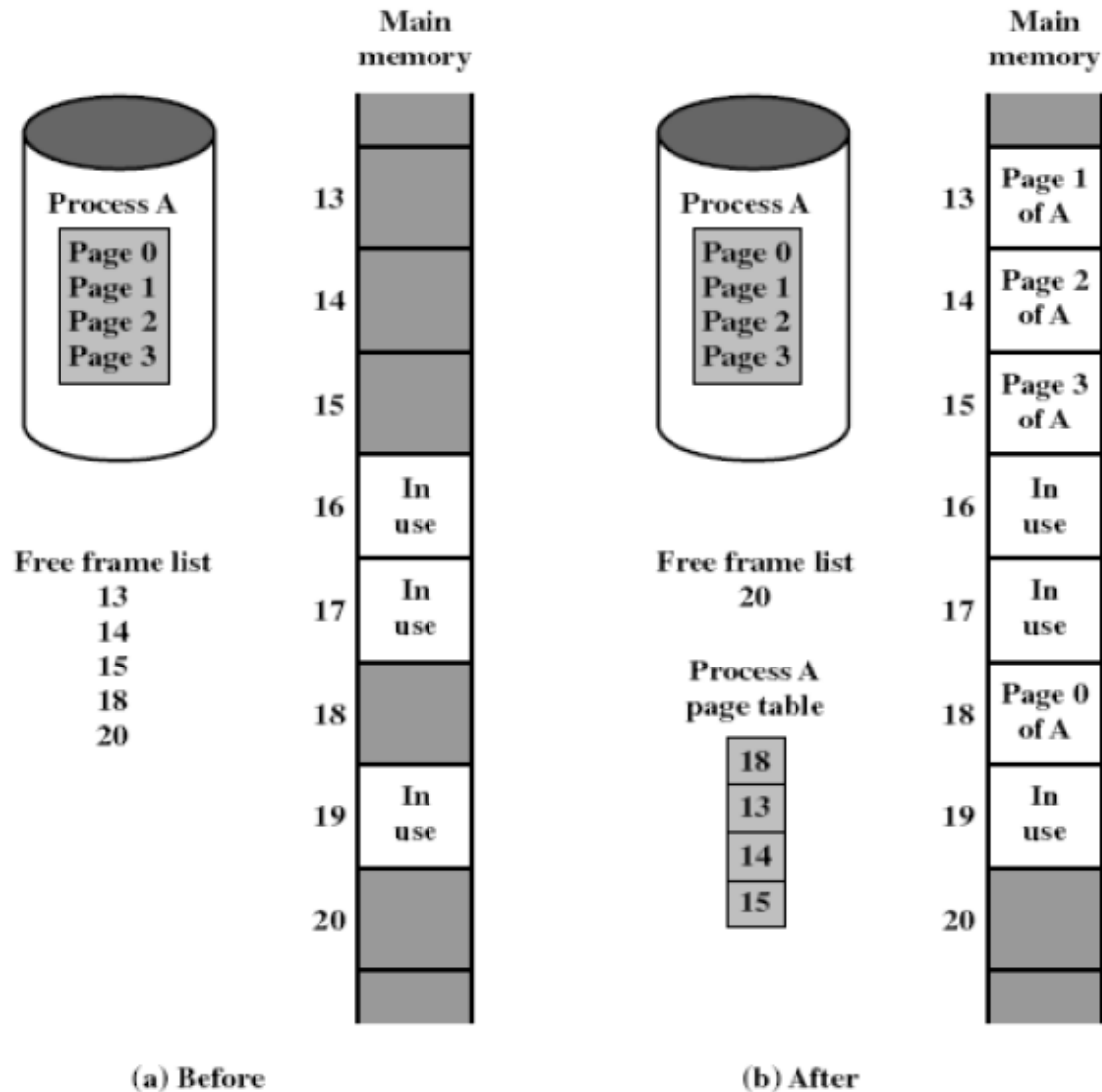
- Khái niệm bộ nhớ ảo: gồm bộ nhớ chính và bộ nhớ ngoài mà được CPU coi như là một bộ nhớ duy nhất (bộ nhớ chính).
- Các kỹ thuật thực hiện bộ nhớ ảo:
  - Kỹ thuật phân trang: Chia không gian địa chỉ bộ nhớ thành các trang nhớ có kích thước bằng nhau và nằm liền kề nhau  
Thông dụng: kích thước trang = 4KBytes
  - Kỹ thuật phân đoạn: Chia không gian nhớ thành các đoạn nhớ có kích thước thay đổi, các đoạn nhớ có thể gộp lên nhau.

# Phân trang

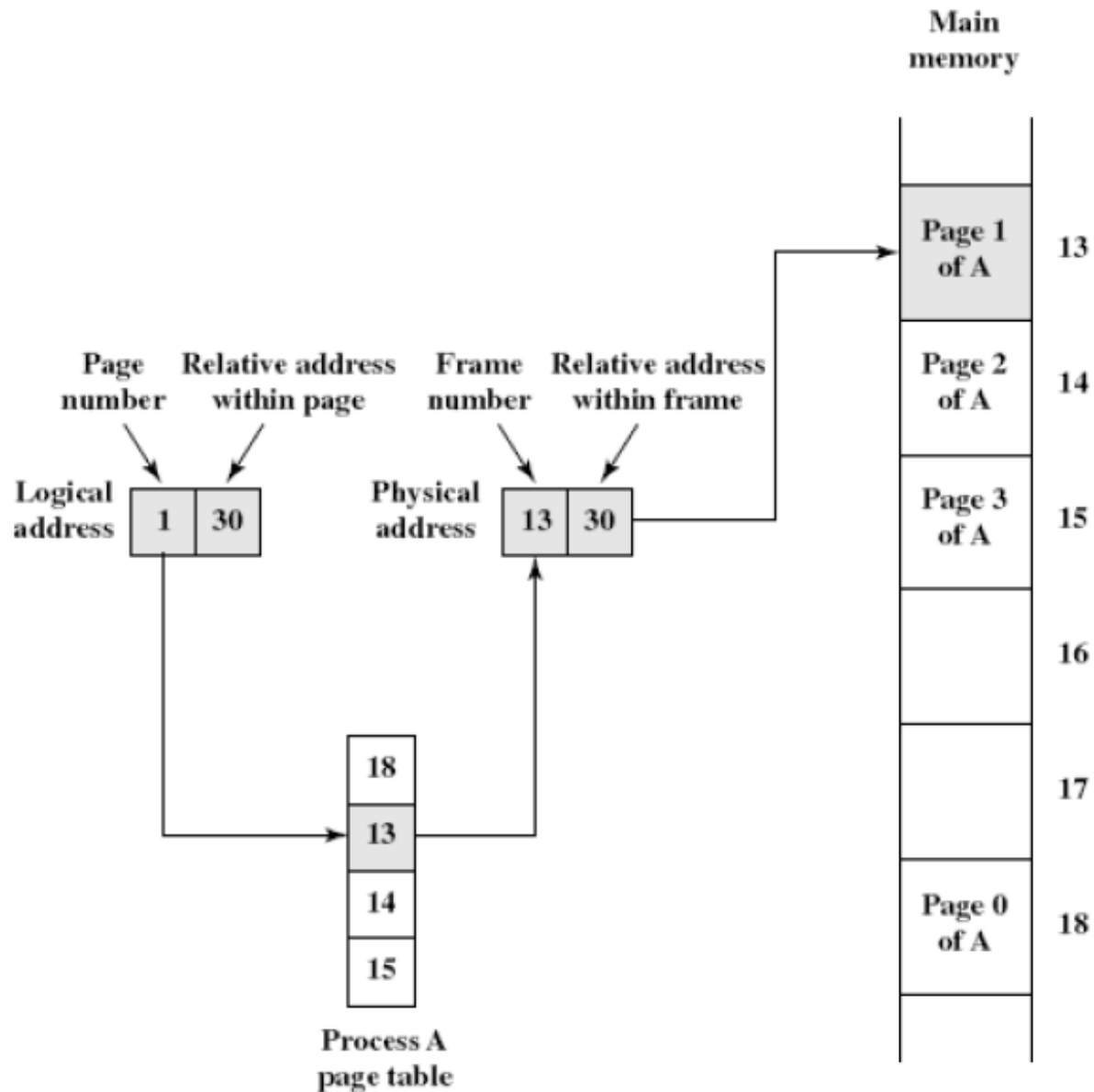
- Phân chia bộ nhớ thành các phần có kích thước bằng nhau gọi là các khung trang
- Chia chương trình (tiến trình) thành các trang
- Cấp phát số hiệu khung trang yêu cầu cho tiến trình
- HĐH duy trì danh sách các khung trang nhớ trống
- Tiến trình không yêu cầu các khung trang liên tiếp
- Sử dụng bảng trang để quản lý



# Cấp phát các khung trang



# Địa chỉ logic và địa chỉ vật lý của phân trang



# Nguyên tắc làm việc của bộ nhớ ảo phân trang

## ■ Phân trang theo yêu cầu

- Không yêu cầu tất cả các trang của tiến trình nằm trong bộ nhớ
- Chỉ nạp vào bộ nhớ những trang được yêu cầu

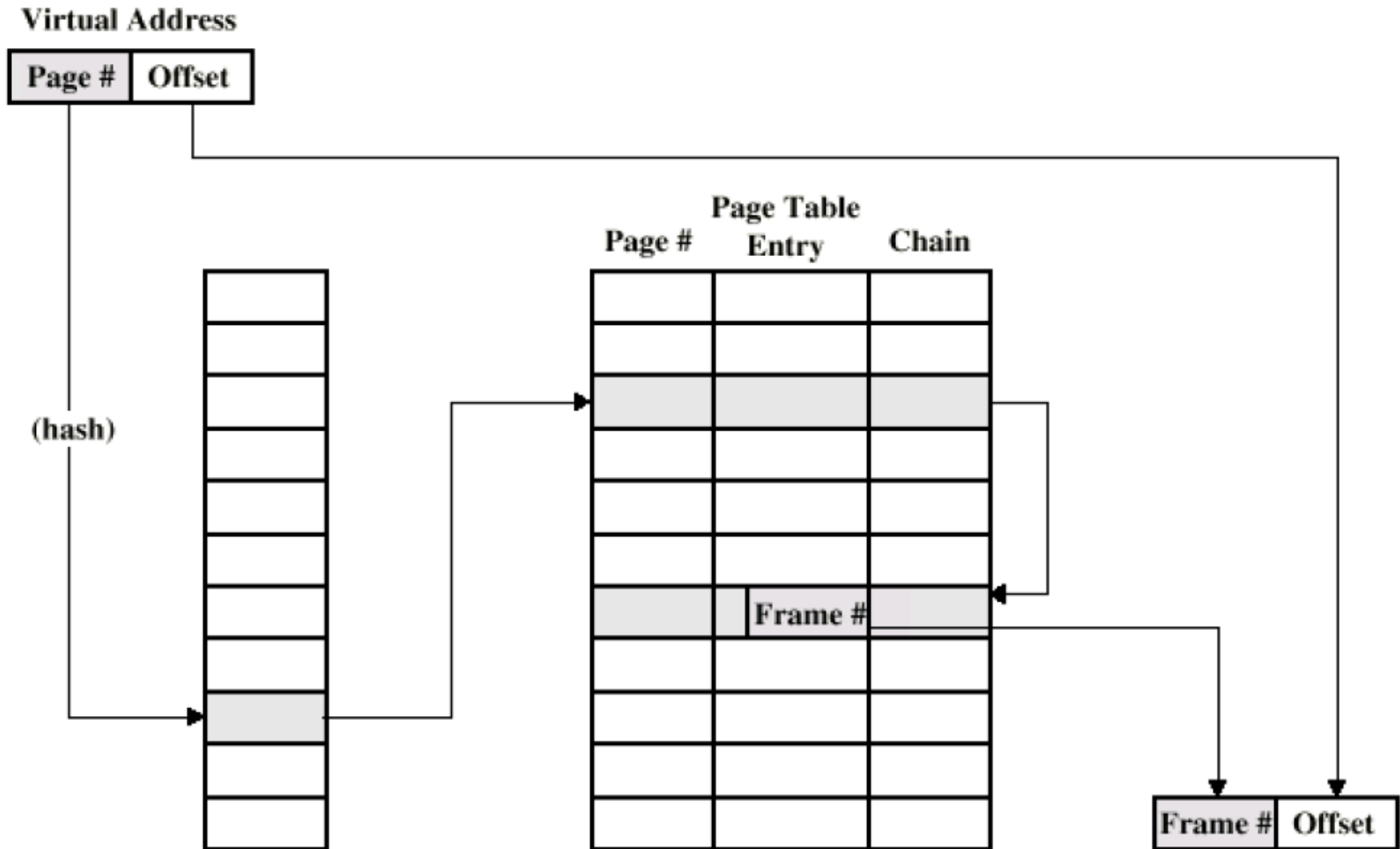
## ■ Lỗi trang

- Trang được yêu cầu không có trong bộ nhớ
- HĐH cần hoán đổi trang yêu cầu vào
- Có thể cần hoán đổi một trang nào đó ra để lấy chỗ
- Cần chọn trang để đưa ra

# Thất bại

- Quá nhiều tiến trình trong bộ nhớ quá nhỏ
- HĐH tiêu tốn toàn bộ thời gian cho việc hoán đổi
- Có ít hoặc không có công việc nào được thực hiện
- Đĩa luôn luôn sáng
- Giải pháp:
- Thuật toán thay trang
- Giảm bớt số tiến trình đang chạy
- Thêm bộ nhớ

# Cấu trúc bảng trang



## 4.7. Hệ thống nhớ trên PC hiện nay

- Bộ nhớ cache: tích hợp trên chip vi xử lý
- Bộ nhớ chính: Tồn tại dưới dạng các mô-đun nhớ RAM
  - SIMM – Single Inline Memory Module
    - 30 chân: 8 đường dữ liệu
    - 72 chân: 32 đường dữ liệu
  - DIMM – Dual Inline Memory Module
    - 64 đường dữ liệu
  - RIMM – Rambus DRAM

# Hệ thống nhớ trên PC hiện nay (tiếp)

- ROM BIOS chứa các chương trình sau:
  - Chương trình POST (Power On Self Test)
  - Chương trình CMOS Setup
  - Chương trình Bootstrap loader
  - Các trình điều khiển vào-ra cơ bản (BIOS)
- CMOS RAM:
  - Chứa thông tin cấu hình hệ thống
  - Đồng hồ hệ thống
  - Có pin nuôi riêng
- Video RAM: quản lý thông tin của màn hình
- Các loại bộ nhớ ngoài

■ **Hết Chương 4**