HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÀI GIẢNG MÔN

KIẾN TRÚC MÁY TÍNH

ThS. Nguyễn Trọng Huân

Khoa Kỹ thuật Điện tử 2

2021

CHƯƠNG 4
INTERNAL MEMORY

Nội dung

1. Tổng quan bộ nhớ máy tính
2. Bộ nhớ chính
3. Bộ nhớ cache

1. TỔNG QUAN BỘ NHỚ MÁY TÍNH

Đặc trưng của bộ nhớ

- Vị trí
- Bên trong CPU: tập thanh ghi
- Bộ nhớ trong:
  - ✓ Bộ nhớ chính: ROM, RAM
  - ✓ Bộ nhớ đệm (cache)
- Bộ nhớ ngoài: các thiết bị lưu trữ
- Dung lượng
- Độ dài từ nhớ (tính bằng bit)
- Số lượng từ nhớ

Đặc trưng của bộ nhớ

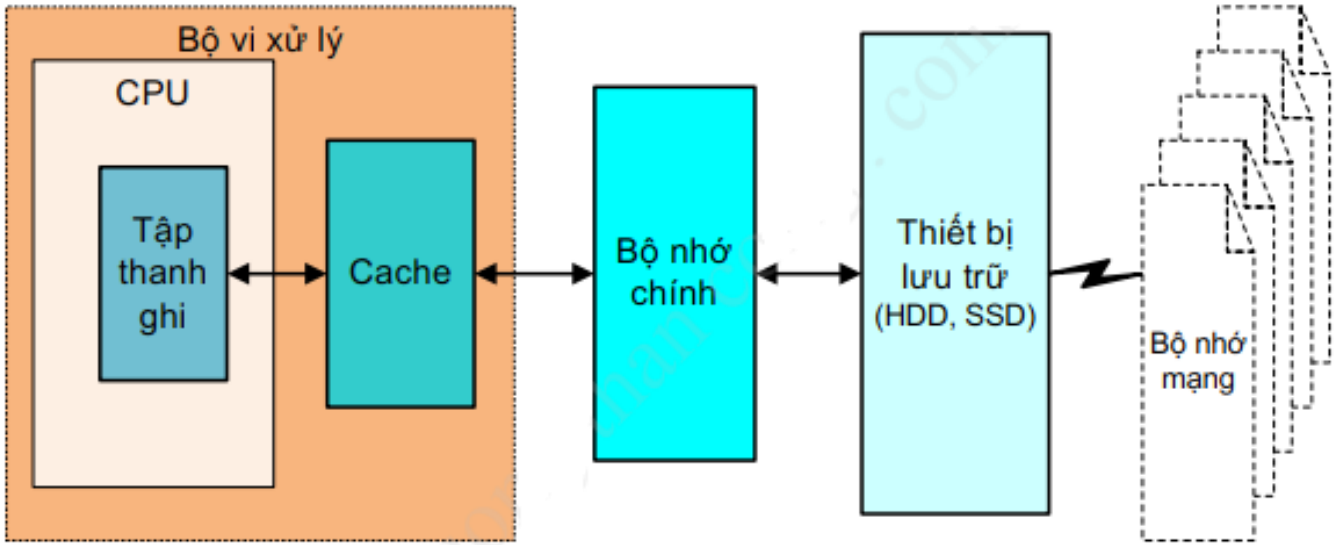
- Đơn vị truyền
  - Từ nhớ (word)
  - Khối nhớ (block)
- Phương pháp truy nhập
  - Truy nhập tuần tự (băng từ)
  - Truy nhập trực tiếp (các loại đĩa)
  - Truy nhập ngẫu nhiên (bộ nhớ bán dẫn)
  - Truy nhập liên kết (cache)

Đặc trưng của bộ nhớ

- Hiệu năng (performance)
  - Thời gian truy nhập
  - Chu kỳ nhớ
  - Tốc độ truyền
- Kiểu vật lý
  - Bộ nhớ bán dẫn: ROM, RAM
  - Bộ nhớ từ: đĩa mềm, đĩa cứng
  - Bộ nhớ quang: CD, DVD…
- Các đặc tính vật lý
  - Thay đổi/Không thay đổi được (volatile / nonvolatile)
  - Xoá được/không xoá được

Phân cấp bộ nhớ

- Từ trái sang phải:
- Dung lượng tăng dần
- Tốc độ giảm dần
- Giá thành cùng dung lượng giảm dần



Phương thức truy xuất

➢ Phương pháp truy xuất tuần tự (Sequential access)

- Bắt đầu ở first location – đọc theo thứ tự
- Thời gian truy cập phụ thuộc vào vị trí dữ liệu và vị trí trước đó
- Ví dụ: Băng từ

➢ Phương thức truy xuất trực tiếp (Direct access)

- Các khối dữ liệu riêng có địa chỉ duy nhất
- Truy xuất bằng cách:
  - ✓ Nhảy đến vùng kế cận
  - ✓ Tìm kiếm tuần tự (hoặc đợi, ví dụ như đợi đĩa quay)
- Thời gian truy cập phụ thuộc vào vị trí đích và vị trí trước đó
- Ví dụ: Ổ đĩa

Phương thức truy xuất

➢ Phương thức truy xuất ngẫu nhiên (Random access)

- Các địa chỉ riêng xác định các vị trí cụ thể
- Thời gian truy cập độc lập vị trí đích hoặc vị trí trước đó
- Ví dụ: RAM

➢ Phương thứ truy xuất liên kết (Associative access)

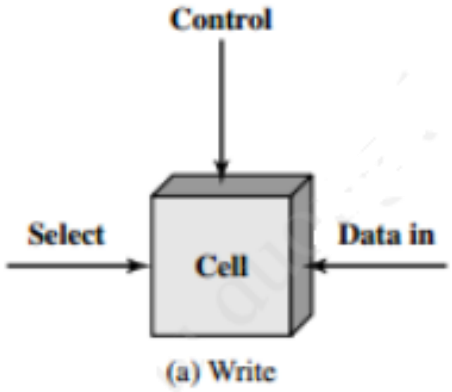
- Dữ liệu được định vị bằng cách so sánh với nội dung của một phần dữ liệu được lưu trữ
- Thời gian truy cập độc lập với vị trí dữ liệu và vị trí truy xuất trước đó.
- Ví dụ: cache

2. BỘ NHỚ CHÍNH

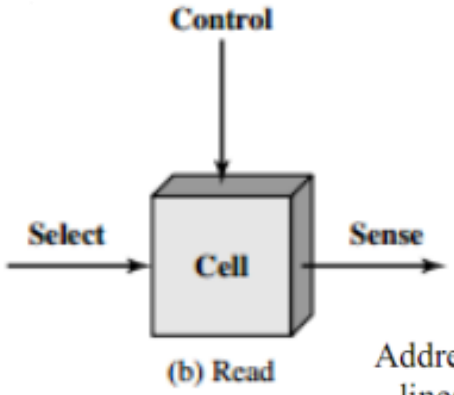
➢ Đặc trưng:

- Chứa các chương trình đang được 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 quản lý
- Việc quản lý logic bộ nhớ chính tùy thuộc vào hệ điều hành

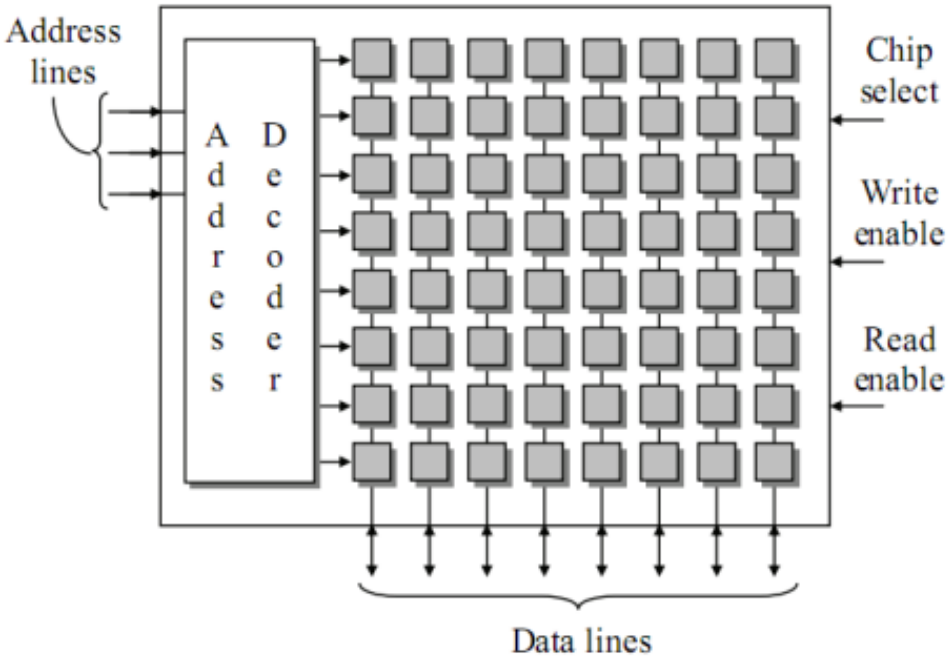
Tổ chức của 1 ô nhớ và bộ nhớ



(a) Write



(b) Read



ROM (Read Only Memory)

- Bộ nhớ chỉ đọc
- Lưu trữ các thông tin sau:
- Thư viện các chương trình con
- Các chương trình điều khiển hệ thống (BIOS)
- Vi chương trình

Phân loại

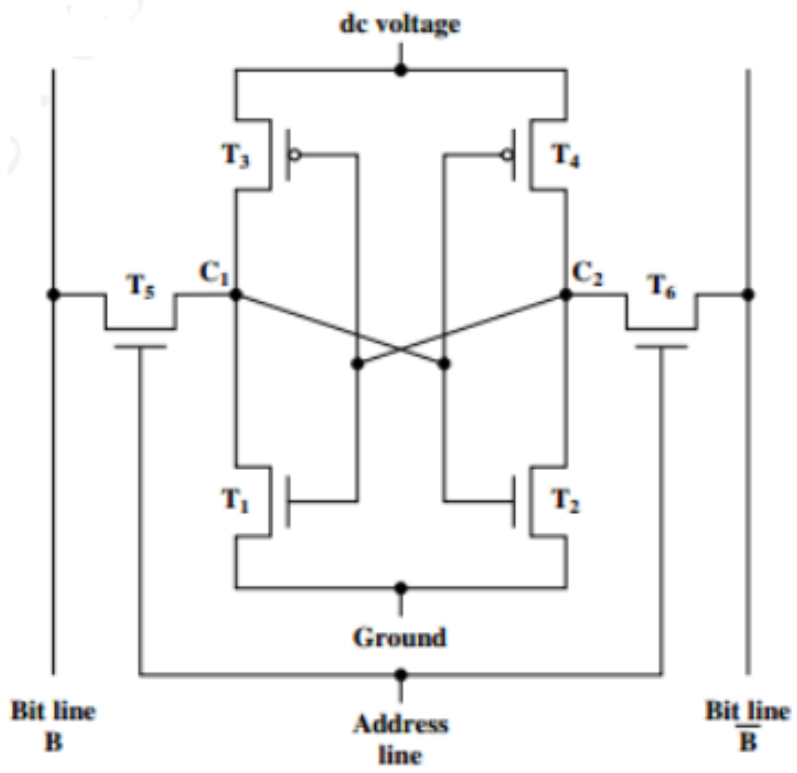
- ROM mặt nạ:
  - Thông tin được ghi khi sản xuất
- PROM (Programmable ROM)
  - Cần thiết bị chuyên dụng để ghi
  - Chỉ ghi được một lần
- EPROM (Erasable PROM)
  - Cần thiết bị chuyên dụng để ghi
  - Xóa được bằng tia tử ngoại
  - Ghi lại được nhiều lần
- EEPROM (Electrically Erasable PROM)
  - Có thể ghi theo từng byte
  - Xóa bằng điện
- FLASH
  - Ghi theo khối
  - Xóa bằng điện
  - Dung lượng lớn

RAM (Radom Access Memory)

- Bộ nhớ truy xuất ngẫu nhiên
- Có khả năng đọc/ghi (Read/Write Memory)
- Lưu trữ thông tin tạm thời
- Có 2 loại: SRAM (Static) và DRAM (Dynamic)

SRAM (Static RAM)

- Các bit được lưu trữ bằng các Flip-Flop nên thông tin ổn định
- Cấu trúc phức tạp
- Dung lượng chip nhỏ
- Tốc độ nhanh (6-8ns)
- Đắt tiền
- Dùng làm bộ nhớ cache

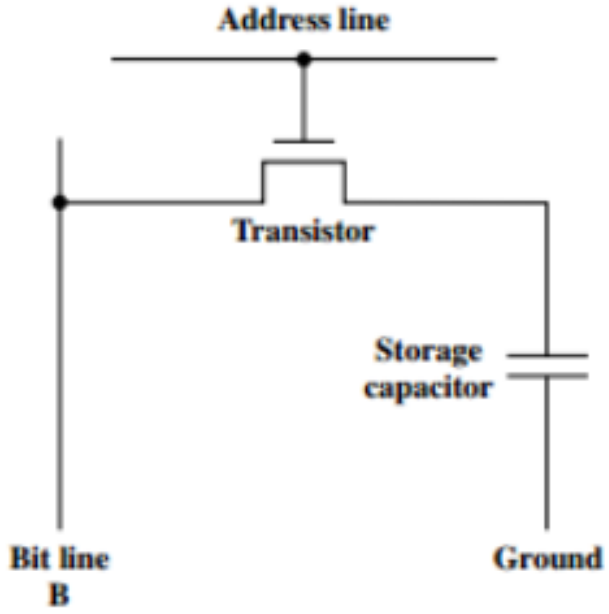


Logic 1:
C1=high, C2=low
T1,T4: off
T2,T3: on

Logic 0:
C1=low, C2=high
T1,T4: on
T2,T3: off

DRAM (Dynamic RAM)

- Các bit được lưu trữ trên tụ điện do đó cần phải có mạch làm tươi
- Cấu trúc đơn giản
- Dung lượng lớn
- Tốc độ chậm hơn (60-80ns)
- Rẻ tiền hơn
- Dùng làm bộ nhớ chính



DRAM tiên tiến

- Enhanced DRAM
  - DRAM có bao gồm một phần nhỏ SRAM
  - Cache DRAM (1Mb DRAM, 8kb SRAM)
- Synchronous DRAM (SDRAM)
  - Đồng bộ hóa với xung nhịp của CPU
- DDR-SDRAM (Double Data Rate SDRAM), DDR3, DDR4
  - Gấp đôi tốc độ của SDRAM, 184-pin
- Rambus DRAM (RDRAM)
  - 2 kênh truyền thông riêng biệt (dual channel)
  - Tốc độ chuyển dữ liệu lên tới 3.2Gbytes/giây

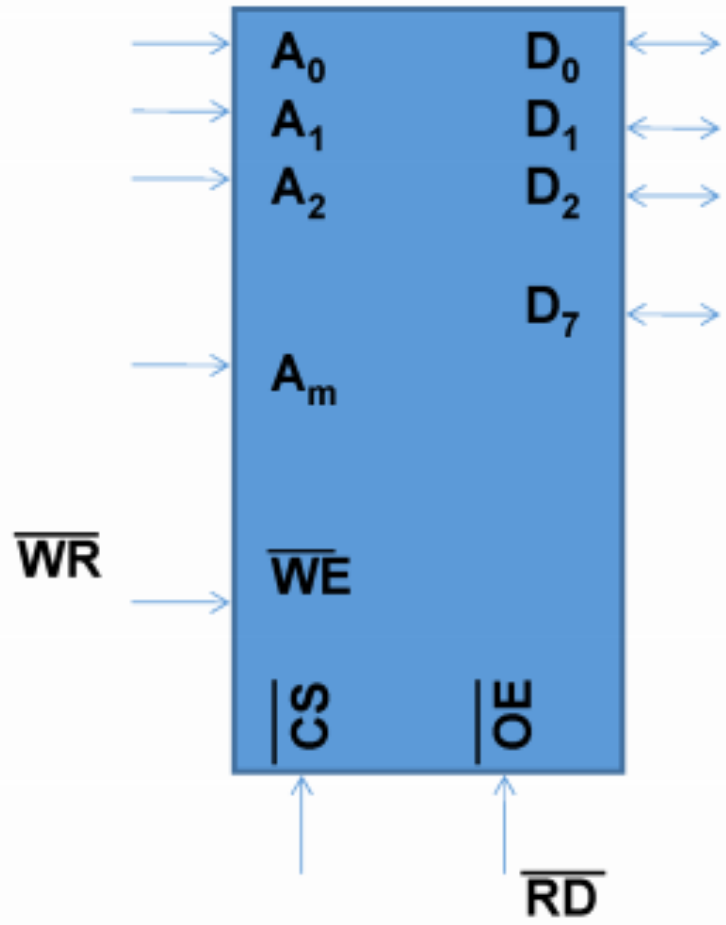
THIẾT KẾ BỘ NHỚ

- Dung lượng chip nhớ 2^N x 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

Cấu trúc tổng quát của bộ nhớ (IC nhớ)

- A_1-A_m: Địa chỉ
- D_0-D_7: Dữ liệu
- WE: Cho phép ghi
- OE: Cho phép ra
- CS: Kích hoạt



Tăng độ dài từ nhớ

➢ VD1:

Cho chip nhớ SRAM 4K x 4 bit

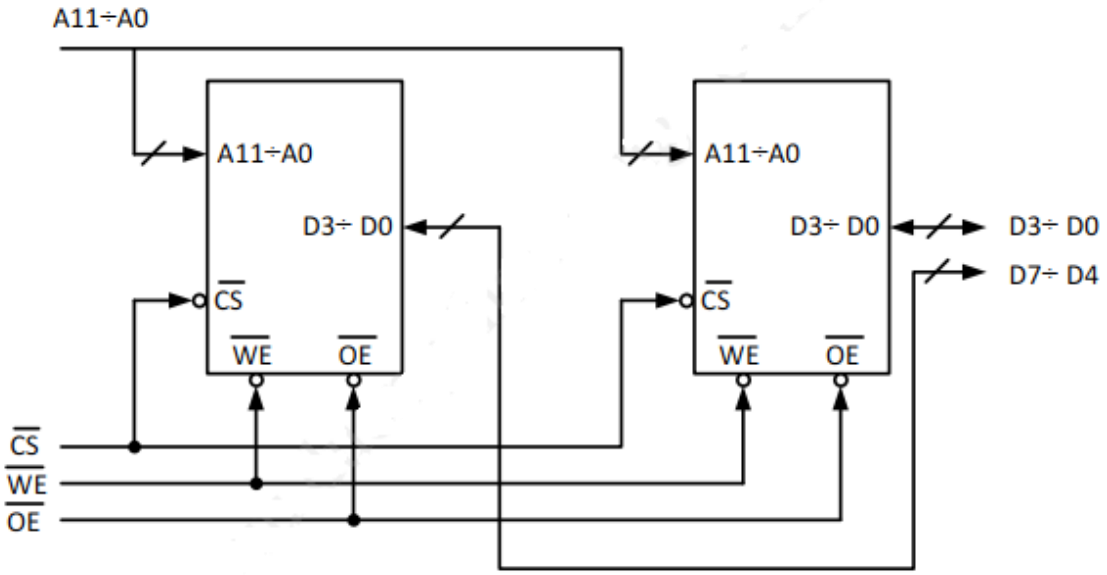
Thiết kế mô-đun nhớ 4K x 8 bit

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



Tăng độ dài từ nhớ

- Cho chip nhớ 2^n x m bit
- Thiết kế mô-đun nhớ 2^n x (k.m) bit
- Dùng k chip nhớ

Tăng số lượng từ nhớ

➢ VD1:

Cho chip nhớ SRAM 4K x 8 bit

Thiết kế mô-đun nhớ 8K x 8 bit

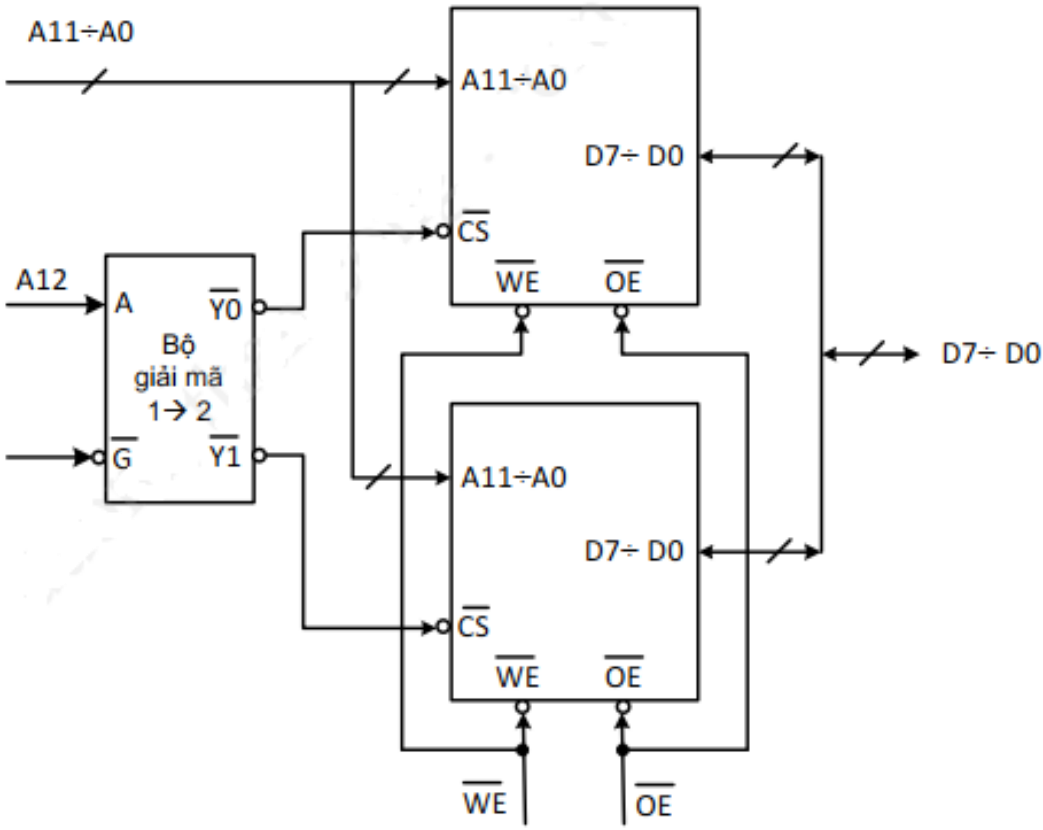
Giải:

Dung lượng chip nhớ = 2^{12} x 8 bit

chip nhớ có: 12 chân địa chỉ, 8 chân dữ liệu

mô-đun nhớ 2^{13} x 8 bit

cần có: 12 chân địa chỉ, 8 chân dữ liệu



Thiết kế kết hợp

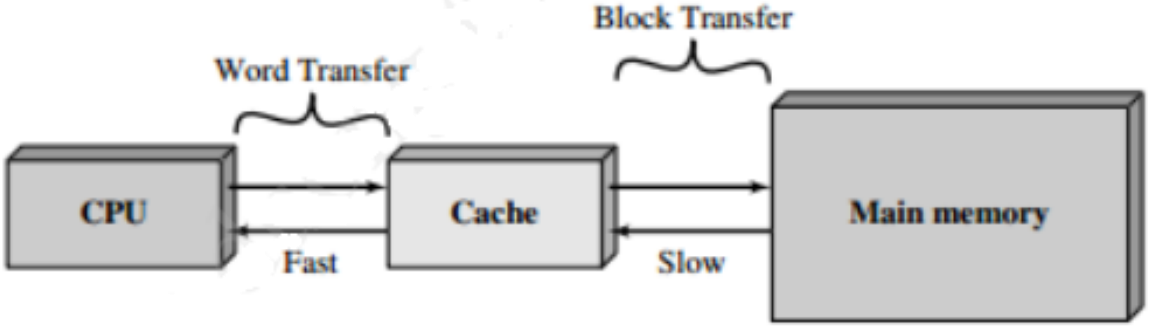
- VD3:

Cho chip nhớ SRAM 4K x 4 bit

Thiết kế mô-đun nhớ 8K x 8 bit

3. BỘ NHỚ CACHE (ĐỆM)

- Cache có tốc độ nhanh hơn bộ nhớ chính
- Cache được đặt giữa CPU và bộ nhớ chính.
- Cache có thể được đặt trên chip của CPU



- Với các hệ thống máy tính cũ, dung lượng cache là 16KB, 32KB,..., 128KB;
- Với các hệ thống máy tính gần đây, dung lượng cache lớn hơn: 256KB, 512KB, 1MB, 2MB, 4MB, 8MB và 16MB

Nguyên lý

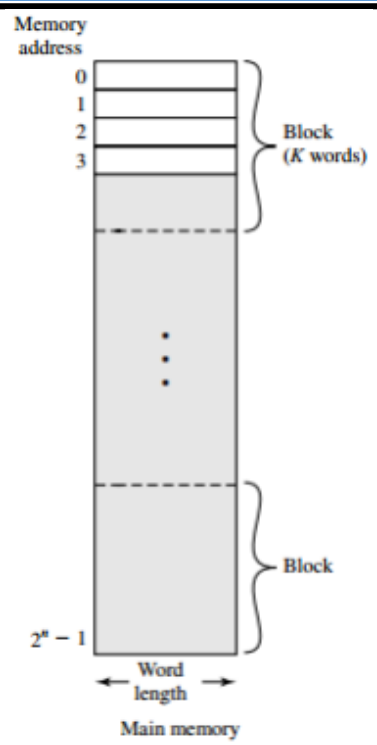
- Bộ nhớ cache là một giải pháp tăng tốc truy xuất bộ nhớ chính. Cache có khả năng đoán trước yêu cầu về dữ liệu và lệnh của CPU. Dữ liệu và lệnh cần thiết được chuyển trước từ bộ nhớ chính về cache
- Cache chứa một phần bộ nhớ chính. Vì vậy khi muốn truy xuất một ngăn nhớ, CPU sẽ tìm trong cache, nếu có sẽ lấy từ nhớ rất nhanh (hit), nếu không thấy sẽ lấy từ bộ nhớ chính (miss) và nạp nguyên khối nhớ chứa ngăn nhớ đó vào cache.
- Lý do nạp cả khối nhớ vào cache: trên cơ sở dự đoán các truy xuất tiếp theo sẽ tham chiếu đến các từ nhớ lân cận từ nhớ vừa truy xuất và như vậy truy xuất kế tiếp sẽ nhanh hơn

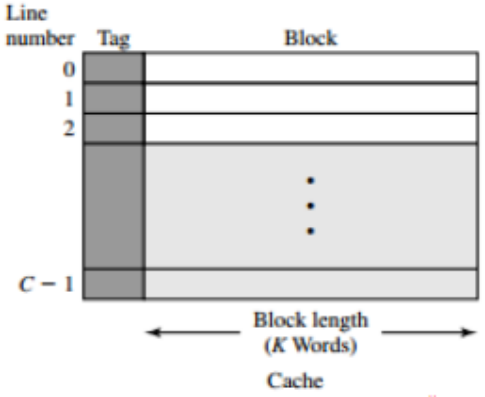
Nguyên lý

- Nguyên lý lân cận về không gian "Nếu một ô nhớ đang được truy nhập thì xác xuất các ô nhớ liền kề với nó được truy nhập trong tương lai gần là rất cao".
- Nguyên lý lân cận về thời gian chú trọng hơn đến tính lặp lại của việc truy nhập các mẩu thông tin trong một khoảng thời gian tương đối ngắn: "Nếu một ô nhớ đang được truy nhập thì xác xuất nó được truy nhập lại trong tương lai gần là rất cao"

Cấu trúc

- Một số Block của bộ nhớ chính được nạp vào các Line của cache.
- Tag (thẻ nhớ) cho biết block nào của bộ nhớ chính hiện đang được chứa ở line đó.
- Khi CPU truy xuất một từ nhớ, có thể:
- Từ nhớ có trong cache (cache hit)
- Từ nhớ không có trong cache (cache miss)
- Vì số line của cache ít hơn số block của bộ nhớ chính, cần có một thuật giải ánh xạ thông tin trong bộ nhớ chính vào cache





Ánh xạ bộ nhớ vào cache

➢ Có 3 phương pháp ánh xạ chủ yếu:

- Ánh xạ trực tiếp
- Ánh xạ liên kết toàn phần
- Ánh xạ liên kết tập hợp

Ánh xạ trực tiếp (Direct mapping)

- Mỗi block của bộ nhớ chính chỉ có thể được nạp vào 1 line duy nhất của cache.
- Quy ước nạp:

B0 → L0

B1 → L1

.....

Bm-1 → Lm-1

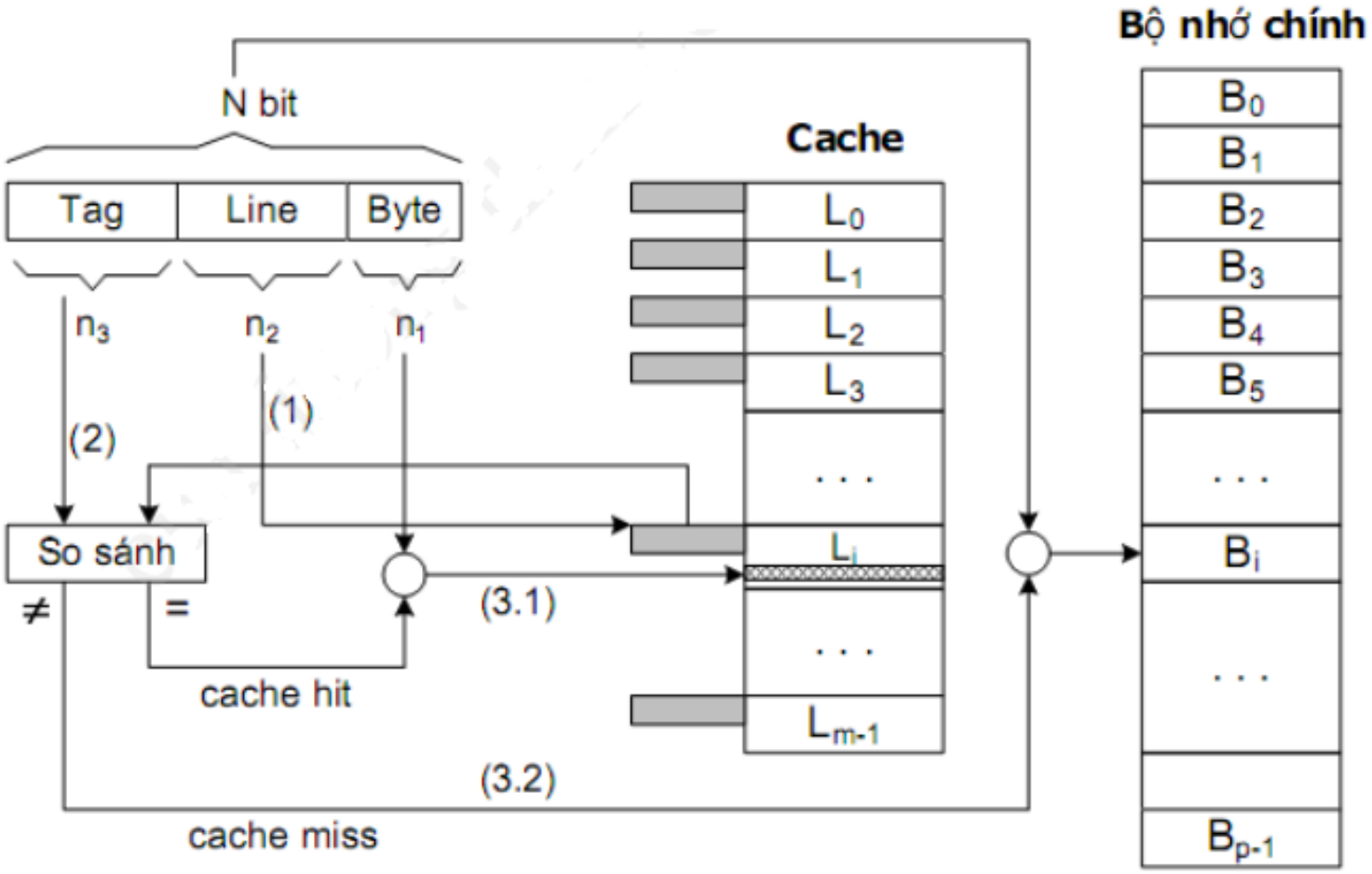
Bm → L0

Bm+1 → L1

L0 : B0, Bm, B2m ...

L1 : B1, Bm+1, B2m+1 ..

Bj chỉ có thể được nạp vào L (j mod m)


Bộ nhớ chính
N bit
Cache
Tag
Line
Byte
n_3
n_2
n_1
(1)
(2)
So sánh
≠
=
cache hit
(3.1)
(3.2)
cache miss
L_0
L_1
L_2
L_3
L_i
L_{m-1}
B_0
B_1
B_2
B_3
B_4
B_5
B_i
B_{p-1}

- Địa chỉ CPU phát ra có N bit, được chia thành 3 trường:
- Trường Byte (có n1 bit) để xác định byte nhớ trong Line (Block): 2^{n1} = kích thước 1 Line
- Trường Line (có n2 bit) để xác định Line trong Cache: 2^{n2} = số Line trong Cache

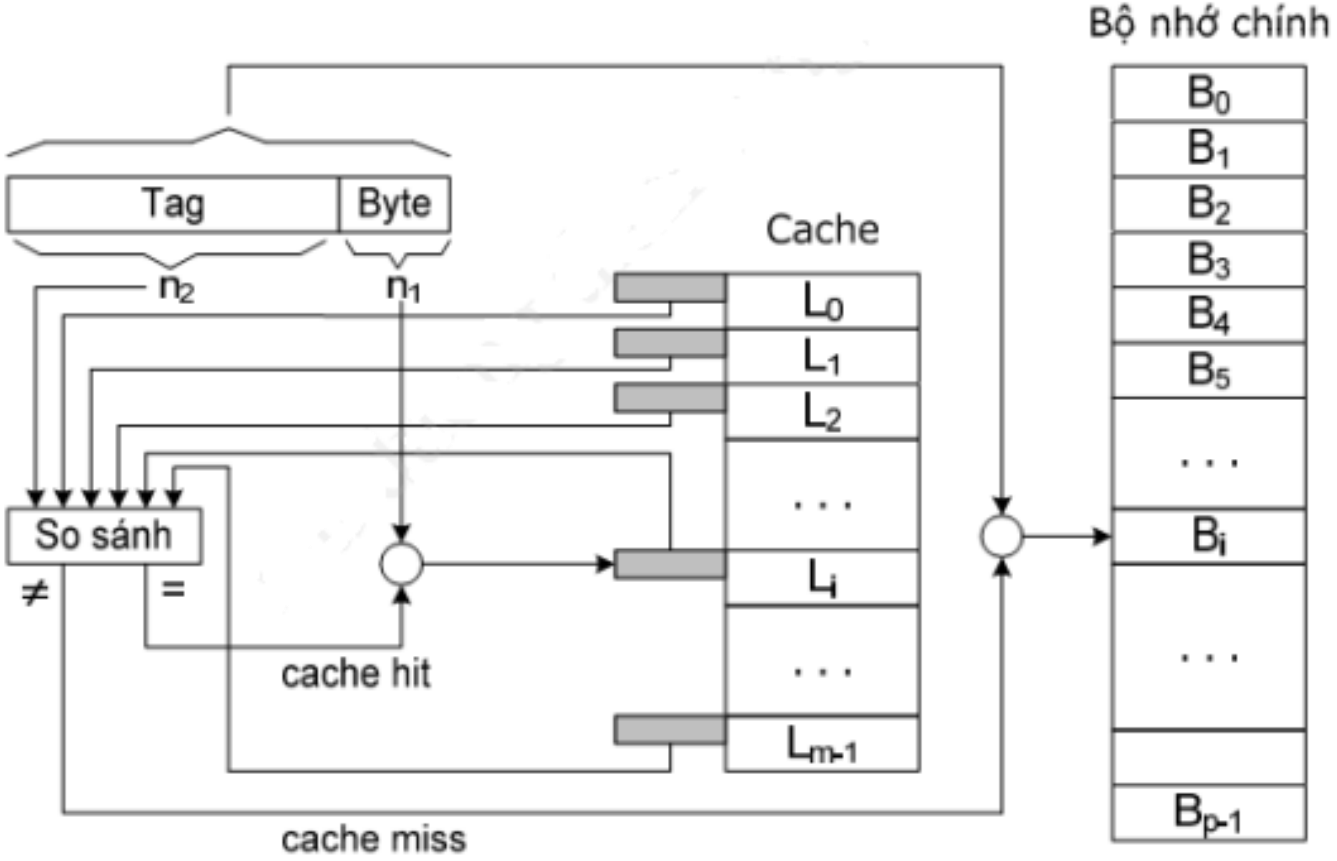
  → Dung lượng Cache = $2^{n1} * 2^{n2} = 2^{n1+n2}$
- Trường Tag (có n3 bit): số bit còn lại:

  $n3 = N - (n1+ n2) > 0$ vì $2^N >> 2^{n1+n2}$

Ánh xạ liên kết toàn phần (Fully Associative Mapping)

- Mỗi block có thể được nạp vào bất kỳ line nào của cache.
- Địa chỉ bộ nhớ do CPU phát ra được chia thành 2 phần: tag và byte.
- Để kiểm tra xem một block có trong cache hay không, phải đồng thời kiểm tra tất cả tag của các line trong cache.

→ Cần các mạch phức tạp để kiểm tra, thiết kế các thuật toán thay thế để tối đa hệ số tìm thấy.


Bộ nhớ chính
Tag
Byte
Cache
n_2
n_1
L_0
L_1
L_2
L_i
L_{m-1}
So sánh
≠
=
cache hit
cache miss
B_0
B_1
B_2
B_3
B_4
B_5
B_i
B_{p-1}

Ánh xạ liên kết tập hợp (Set Associative Mapping)

- Là phương pháp dung hòa của 2 phương pháp trên
- Chia cache thành các tập: S0, S1, S2 ...
- Mỗi Set có một số Line (2, 4, 8, 16 Line)
- Ví dụ: Mỗi Set có 2 line: 2-way Set Associative Mapping. Mỗi block được nạp vào 1 line nào đó trong Set nhất định:

  B0 → S0

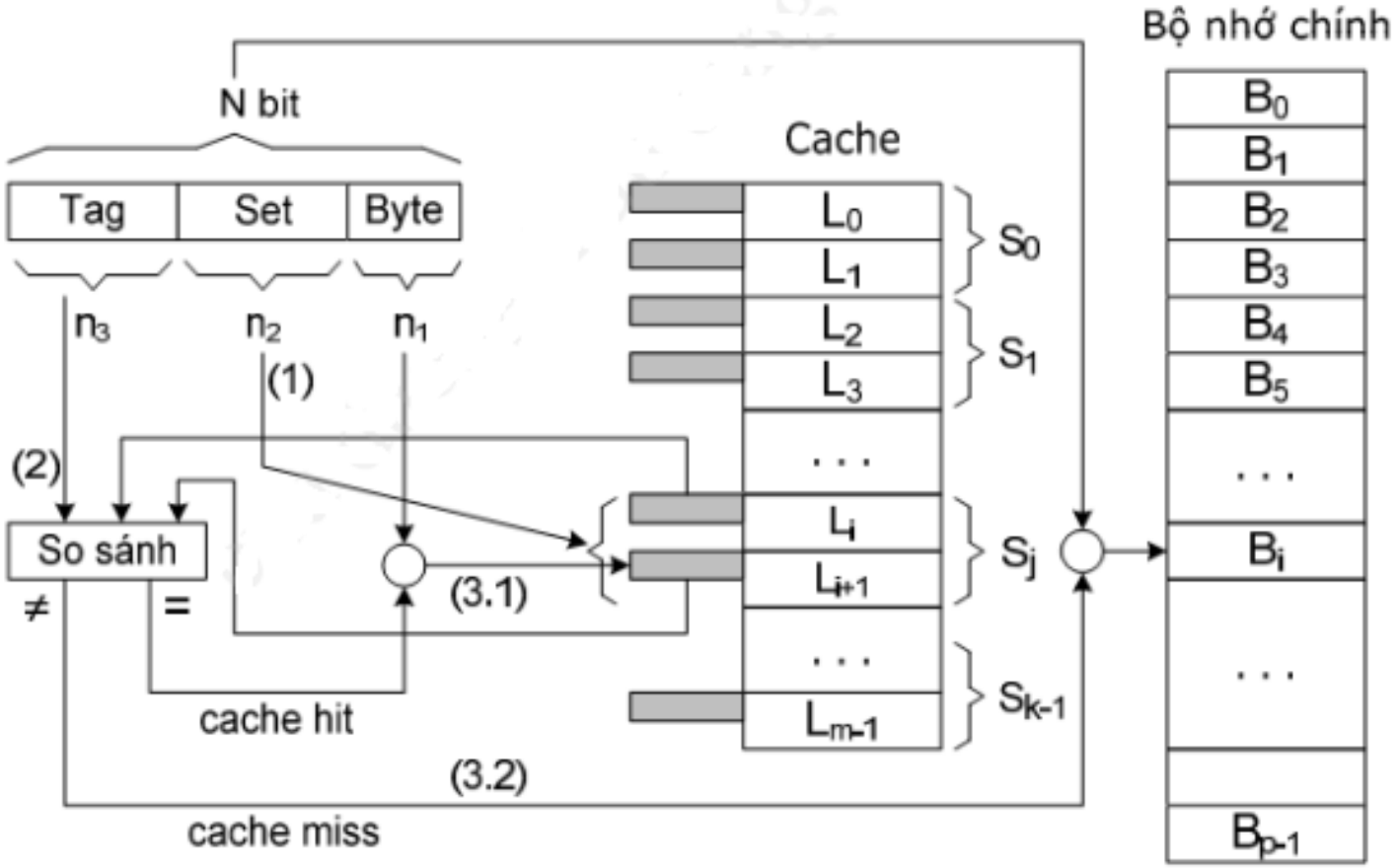
  B1 → S1

  .....

  Bk-1 → Sk-1

  Bk → S0

  Địa chỉ do CPU phát ra có 3 trường: Tag, Set, Byte


Bộ nhớ chính
N bit
Cache
Tag
Set
Byte
n_3
n_2
n_1
(1)
(2)
So sánh
≠
=
(3.1)
cache hit
(3.2)
cache miss
L_0
L_1
L_2
L_3
L_i
L_{i+1}
L_{m-1}
S_0
S_1
S_j
S_{k-1}
B_0
B_1
B_2
B_3
B_4
B_5
B_i
B_{p-1}

- Ví dụ:

  Hệ thống có:

  Bộ nhớ chính = 256 MB

  Cache = 128 KB

  Line = 16 Byte

  Xác định số bit của các trường địa chỉ khi

  a) Ánh xạ trực tiếp

  b) Ánh xạ liên kết tập hợp 4 Line/Set

a.

$2^N = 256. 2^{20} = 2^{28} \rightarrow N = 28$ bit

Tính cho trường Byte:

Kích thước line = 16 = 2^4 Byte → n1 = 4 bit

Tính cho trường Line:

Số line trong Cache: 128. $2^{10}/16 = 2^{13} \rightarrow$ n2 = 13 bit

Tính cho trường Tag:

n3 = N - (n1 + n2) = 28 - (4 + 13) = 11 bit

b.

- Trường Byte: n1 = 4 bit
- Trường Set: Số Set = Số line/4 = $2^{13}/4 = 2^{11} \rightarrow$ n2=11 bit
- Trường Tag: n3 = N - (n1 + n2) = 28 - (4 + 11) = 13 bit

Giải thuật thay thế block trong cache

- Khi CPU truy nhập một thông tin mà không có trong cache (cache miss) thì nạp block chứa thông tin đó vào trong cache để thay thế block cũ trong cache.
- Ánh xạ trực tiếp → chỉ có 1 cách nạp → không cần thuật giải để nạp.
  - Mỗi Block chỉ ánh xạ vào một Line xác định
  - Thay thế Block ở Line đó
- 2 phương pháp ánh xạ liên kết → cần có thuật giải để lựa chọn thay thế

Giải thuật thay thế block trong cache

Với ánh xạ liên kết: cần có thuật giải thay thế:

- 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 truy cập

  → Tối ưu, hiệu quả nhất: LRU

Phương pháp ghi dữ liệu khi cache hit

- ➢ Ghi xuyên qua (Write through)
- Ghi cả cache và 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

HẾT CHƯƠNG 4