HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

BÀI GIẢNG MÔN

KIẾN TRÚC MÁY TÍNH

CHƯƠNG 3.2 – BỘ NHỚ CACHE

Giảng viên: TS. Hoàng Xuân Dậu

Điện thoại/E-mail: dau@ekabiz.vn

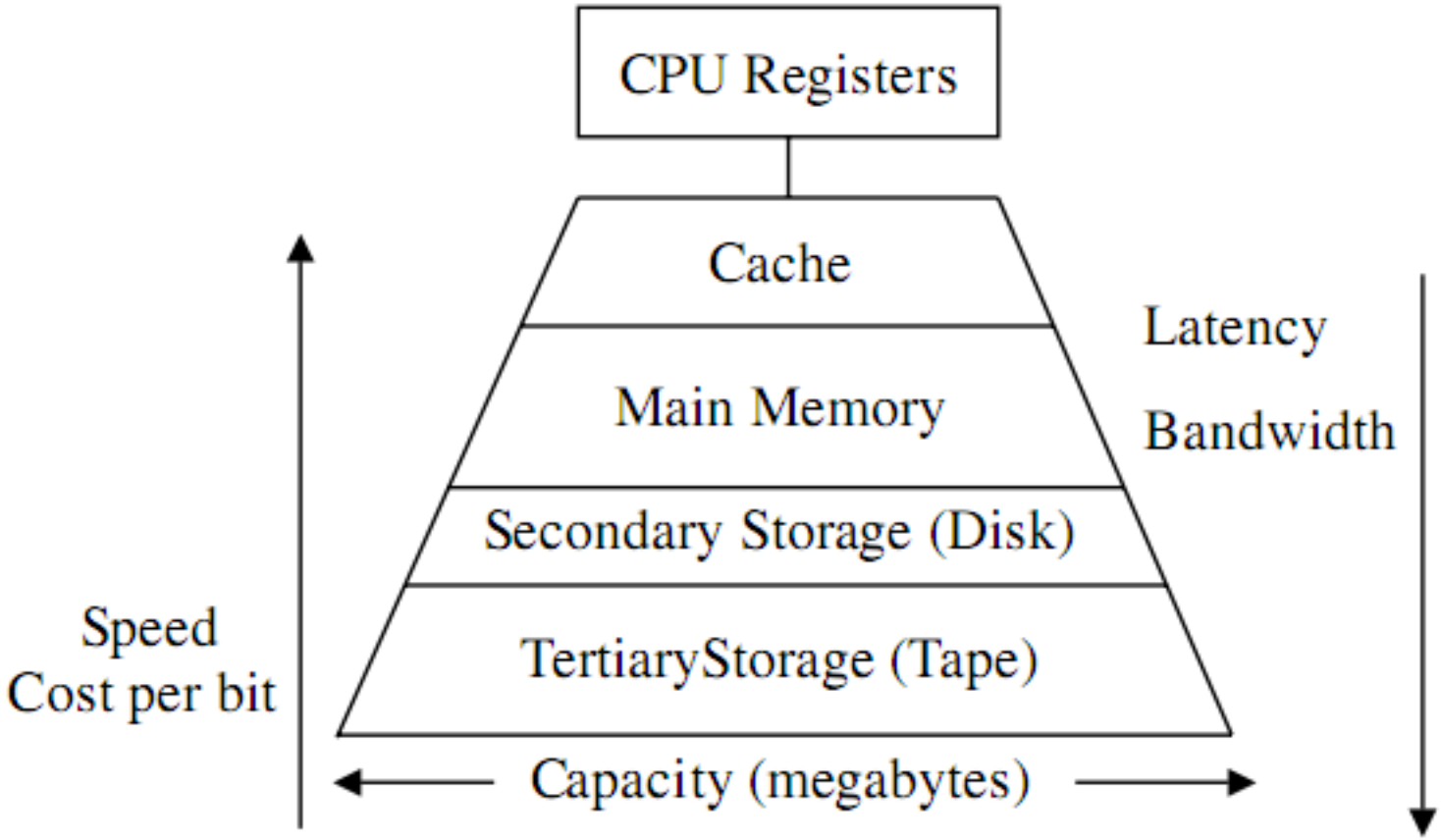
Bộ môn: Khoa học máy tính - Khoa CNTT1

Học kỳ/Năm biên soạn: Học kỳ 2 năm học 2009-2010

NỘI DUNG

1. Giới thiệu về bộ nhớ trong và cấu trúc phân cấp của bộ nhớ
2. Phân loại bộ nhớ và tổ chức mạch nhớ, ROM và RAM
3. Bộ nhớ cache
4. Câu hỏi ôn tập

3.2.1 Hệ thống nhớ - mô hình phân cấp



4.1 Hệ thống nhớ - tham số

	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

3.2.1 Hệ thống nhớ - Các thành phần

- CPU registers (các thanh ghi của CPU):
  - Dung lượng rất nhỏ, khoảng từ vài chục bytes đến vài KB
  - Tốc độ truy nhập rất cao (các thanh ghi hoạt động với tốc độ của CPU); thời gian truy nhập khoảng 0,25ns
  - Giá thành đắt
  - Sử dụng để lưu toán hạng đầu vào và kết quả của các lệnh.
- Cache (bộ nhớ cache):
  - Dung lượng tương đối nhỏ (khoảng 64KB đến 16MB)
  - Tốc độ truy nhập cao; thời gian truy nhập khoảng 1-5ns
  - Giá thành đắt
  - Còn được gọi là “bộ nhớ thông minh” (smart memory)
  - Sử dụng để lưu lệnh và dữ liệu cho CPU xử lý.

4.1 Hệ thống nhớ - Các thành phần

- Main memory (bộ nhớ chính):
  - Gồm ROM và RAM, có kích thước khá lớn; với hệ thống 32 bít, dung lượng khoảng 256MB-4GB
  - Tốc độ truy nhập chậm; thời gian truy nhập khoảng 50-70ns
  - Giá thành tương đối rẻ
  - Sử dụng để lưu lệnh và dữ liệu của hệ thống và của người dùng
- Secondary memory (bộ nhớ thứ cấp – bộ nhớ ngoài):
  - Có dung lượng rất lớn, khoảng từ 20GB-1000GB
  - Tốc độ truy nhập rất chậm; thời gian truy nhập khoảng 5ms
  - Giá thành rẻ
  - Sử dụng để lưu dữ liệu lâu dài dưới dạng các tệp (files)

3.2.1 Hệ thống nhớ - Vai trò của mô hình phân cấp

- Tăng hiệu năng hệ thống
  - Dung hoà được CPU có tốc độ cao và phần bộ nhớ chính và bộ nhớ ngoài có tốc độ thấp;
  - Thời gian trung bình CPU truy nhập dữ liệu từ hệ thống nhớ tiệm cận thời gian truy nhập cache.
- Giảm giá thành sản xuất
  - Các thành phần đắt tiền (thanh ghi và cache) được sử dụng với dung lượng nhỏ;
  - Các thành phần rẻ tiền hơn (bộ nhớ chính và bộ nhớ ngoài) được sử dụng với dung lượng lớn;

  ➔ tổng giá thành của hệ thống nhớ theo mô hình phân cấp sẽ rẻ hơn so với hệ thống nhớ không phân cấp có cùng tốc độ.

3.2.2 Phân loại bộ nhớ

- Dựa trên kiểu truy nhập:
  - Random Access Memory (RAM): bộ nhớ truy nhập ngẫu nhiên
  - Serial Access Memory (SAM) : bộ nhớ truy nhập tuần tự
  - Read Only Memory (ROM): bộ nhớ chỉ đọc
- Dựa trên khả năng duy trì dữ liệu:
  - Volatile memory: bộ nhớ không ổn định; thông tin mất khi mất nguồn nuôi: RAM.
  - Non-volatile memory: bộ nhớ ổn định; thông tin vẫn được duy trì khi mất nguồn nuôi: ROM,HDD,...
- Dựa trên công nghệ chế tạo:
  - Bộ nhớ bán dẫn (Semiconductor memory): ROM, RAM
  - Bộ nhớ từ tính (Magnetic memory): HDD, FDD, băng từ
  - Bộ nhớ quang học (Optical memory): CD, DVD

4.3 Phân loại bộ nhớ trong-Bộ nhớ ROM

- ROM là bộ nhớ chỉ đọc (Read Only Memory)
  - Việc ghi thông tin vào ROM chỉ có thể được thực hiện bằng các thiết bị hoặc phương pháp đặc biệt;
- ROM là bộ nhớ ổn định
  - Thông tin trong ROM vẫn được duy trì khi mất nguồn nuôi
- ROM là bộ nhớ bán dẫn: mỗi ô nhớ của ROM là một cổng bán dẫn
- ROM thường được sử dụng để lưu chương trình khởi động của máy tính
  - Đọc các thông tin về phần cứng hệ thống trong RAM CMOS (Basic Input Output System – hệ thống vào ra cơ sở).

4.3 Bộ nhớ ROM – Ví dụ

4.3 Bộ nhớ ROM – Các loại ROM

- ROM nguyên thuỷ (Orginal ROM):
  - ROM các thế hệ đầu tiên;
- PROM (Programmable ROM):
  - ROM có thể lập trình được;
  - Thông tin có thể được ghi vào PROM nhờ một thiết bị đặc biệt gọi là bộ lập trình PROM.
- EPROM (Erasable programmable read-only memory):
  - Là ROM có thể lập trình và xoá được;
  - Thông tin trong EPROM có thể xoá được sử dụng tia cực tím có cường độ cao.

4.3 Bộ nhớ ROM – Các loại ROM

- EEPROM: (Electrically Erasable PROM):
  - Là PROM có thể xoá được thông tin bằng điện
  - Có thể ghi được thông tin sử dụng phần mềm chuyên dụng
- Flash memory:
  - Là một dạng EEPROM nhưng có tốc độ ghi và đọc thông tin nhanh hơn.
  - Bộ nhớ flash chỉ có thể đọc/ghi theo khối.

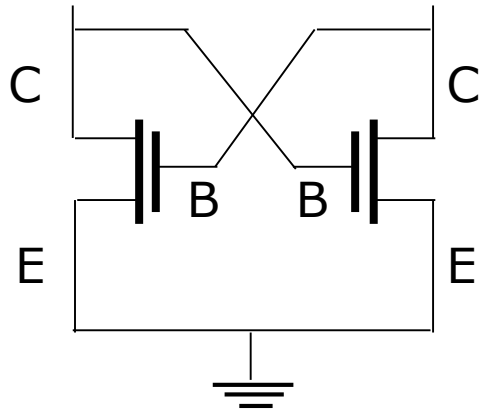
3.2.2 Bộ nhớ RAM – Giới thiệu

- RAM (Random Access Memory) là bộ nhớ truy nhập ngẫu nhiên
  - Mỗi ô nhớ của RAM có thể được truy nhập một cách ngẫu nhiên không theo trật tự nào;
  - Tốc độ truy nhập các ô nhớ là tương đương.
- RAM là bộ nhớ không ổn định:
  - Tất cả thông tin trong RAM sẽ bị mất khi mất nguồn nuôi
- RAM là bộ nhớ bán dẫn: mỗi ô nhớ của RAM là một cổng bán dẫn
- RAM được sử dụng để lưu các thông tin của hệ thống và của người dùng:
  - Thông tin của hệ thống: thông tin phần cứng và hệ điều hành
  - Thông tin của người dùng: các chương trình ứng dụng và dữ liệu.

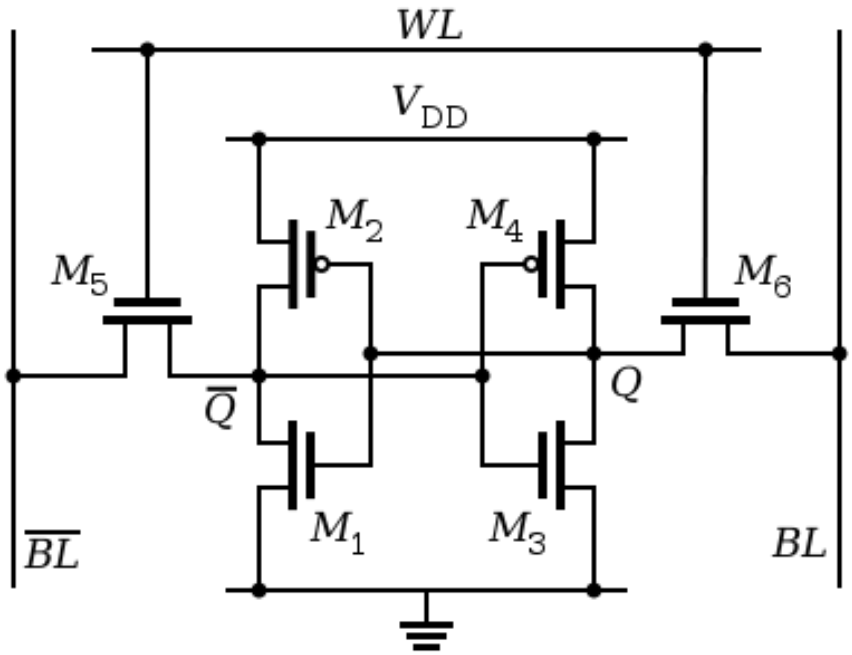
4.4 RAM – Các loại RAM

- Hai loại RAM cơ bản:
  - RAM tĩnh (Static RAM – SRAM):
    - Mỗi bít SRAM là một mạch lật – flip-flop
    - Thông tin lưu trong các bit SRAM luôn ổn định và không phải "làm tươi" định kỳ
    - SRAM nhanh hơn nhưng đắt hơn DRAM.
  - RAM động (Dynamic RAM – DRAM):
    - Mỗi bít DRAM dựa trên một tụ điện
    - Thông tin lưu trong các bit DRAM không ổn định và phải được "làm tươi" định kỳ
    - DRAM chậm hơn nhưng rẻ hơn SRAM.

4.4.1 SRAM – Cấu tạo



Một mạch lật (flip-flop) đơn giản

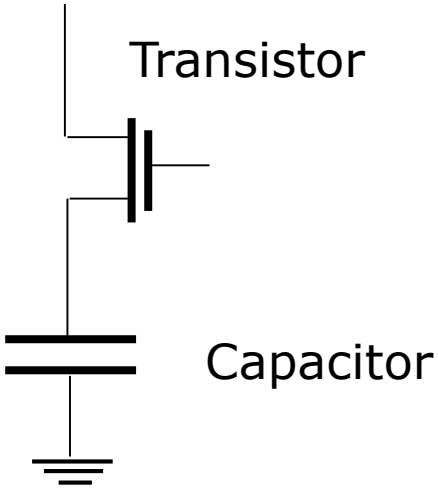


Một ô nhớ SRAM loại 6T

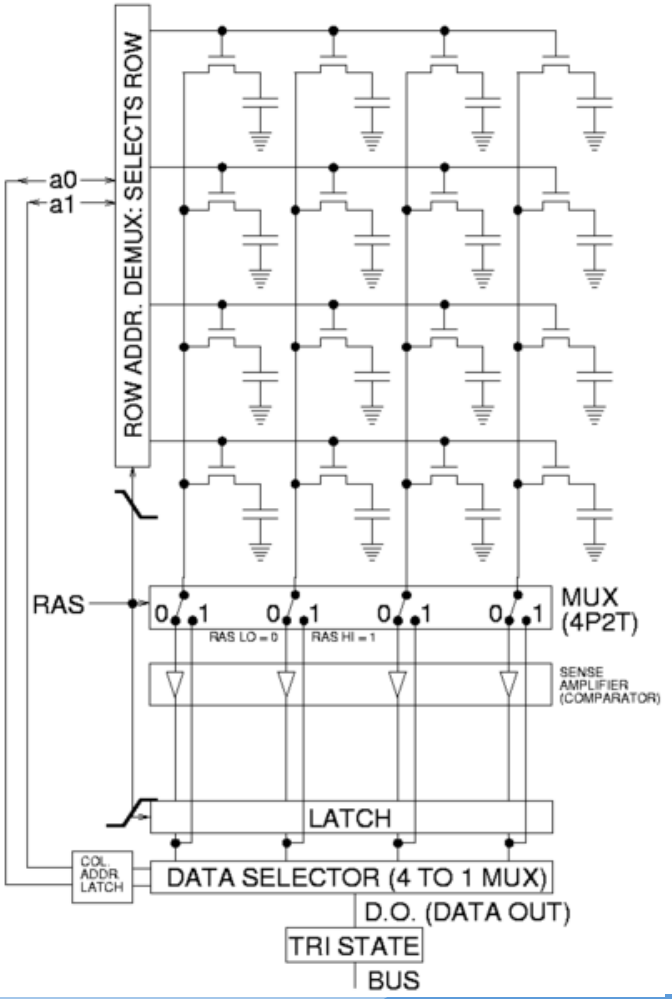
4.4.1 SRAM – Đặc điểm

- SRAM sử dụng một mạch lật trigơ lưỡng ổn (*bistable latching circuit*) để lưu một bit thông tin;
- Mỗi mạch lật lưu 1 bit thường sử dụng 6, 8 hoặc 10 transitors (gọi là mạch 6T, 8T hoặc 10T);
- SRAM thường có tốc độ truy nhập nhanh do:
  - Các bit của SRAM 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 thường đắt hơn so với DRAM do:
  - Mỗi bít SRAM dùng nhiều transitor hơn so với 1 bit DRAM
  - Do cấu trúc bên trong của SRAM bit phức tạp hơn nên mật độ SRAM thường thấp.

4.4.2 DRAM – Cấu tạo



Một bit DRAM



4.4.2 DRAM – Đặc điểm

- Mỗi bit DRAM dựa trên một tụ điện và một transitor:
  - Hai mức tích điện của tụ biểu diễn 2 mức logic 0 và 1:
    - Không tích điện: mức 0
    - Tích đầy điện: mức 1
- Do tụ thường tự phóng điện, điện tích trong tụ 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.

4.4.2 DRAM – Đặc điểm

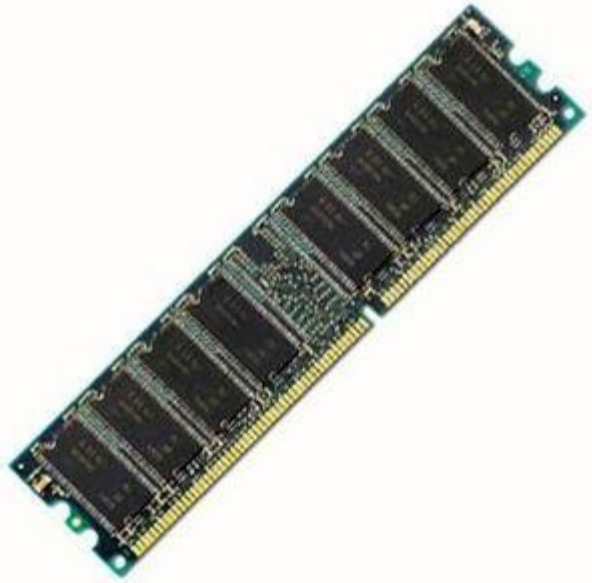
- Các bit nhớ của DRAM thường được sắp xếp thành ma trận:
  - Một tụ + một transitor → một bit
  - Các bít được tập hợp thành các dòng và cột
- DRAM thường chậm hơn SRAM do:
  - Cần quá trình làm tươi
  - Việc nạp điện cho tụ cũng mất nhiều thời gian
  - Các mạch DRAM thường dùng kỹ thuật dồn kênh (địa chỉ cột/hàng) để tiết kiệm đường địa chỉ.
- DRAM thường rẻ hơn SRAM do:
  - Cấu trúc đơn giản, dùng ít transitor
  - Mật độ cấy cao

4.4.2 DRAM – Các loại DRAM

- SDRAM(Synchronous DRAM): DRAM đồng bộ (với nhịp đồng hồ của bus)
- SRD SDRAM (Single Data Rate SDRAM): chấp nhận một thao tác đọc/ghi và chuyển 1 từ dữ liệu trong 1 chu kỳ đồng hồ; tốc độ 100MHz, 133MHz
- DDR SDRAM (Double Data Rate SDRAM)
  - DDR1 SDRAM: DDR 266, 333, 400: có khả năng chuyển 2 từ dữ liệu trong 1 chu kỳ đồng hồ;
  - DDR2 SDRAM: DDR2 400, 533, 800 : có khả năng chuyển 4 từ dữ liệu trong 1 chu kỳ đồng hồ;
  - DDR3 SDRAM: DDR3 800, 1066, 1333, 1600 : có khả năng chuyển 8 từ dữ liệu trong 1 chu kỳ đồng hồ;

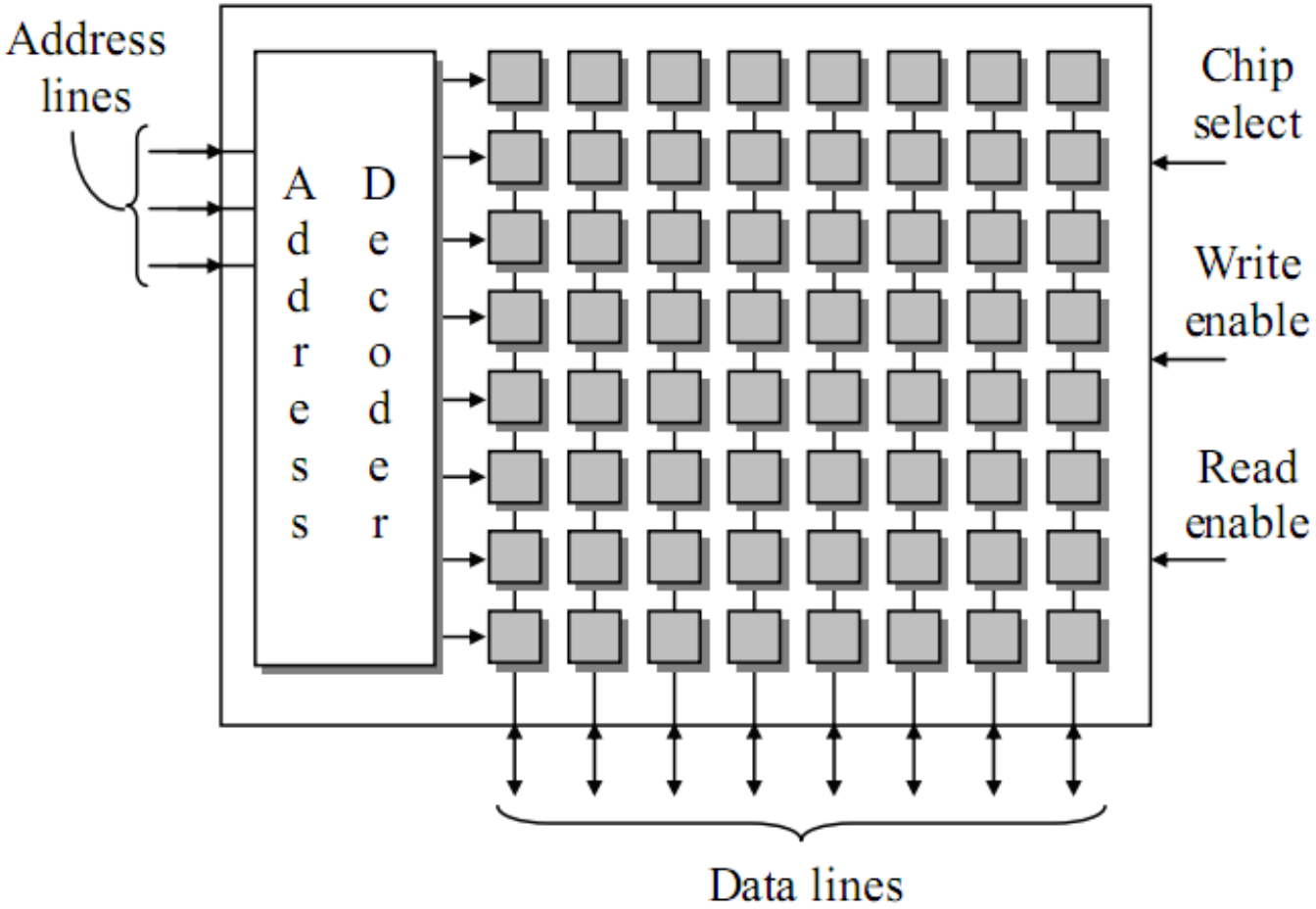
4.4.2 DRAM – Các loại DRAM

SDRAM PC133



DDR3 1066 SDRAM

3.2.2 Tổ chức mạch nhớ



4.2 Tổ chức mạch nhớ

- Address lines:
  - Các đường địa chỉ kết nối với bus A;
  - Chuyển tín hiệu địa chỉ từ CPU đến mạch nhớ
- Address decoder:
  - Bộ giải mã địa chỉ
  - Sử dụng tín hiệu 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;
  - Truyễn dữ liệu từ bộ nhớ về CPU và ngược lại.

4.2 Tổ chức mạch nhớ

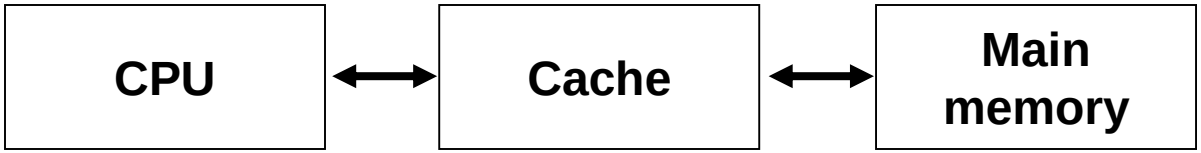
- Chip select (CS):
  - Chân tín hiệu chọn chip;
  - Chip nhớ được kích hoạt khi CS = 0; Thông thường, CPU chỉ có thể làm việc với một chip nhớ tại một thời điểm.
- Write enable (WE):
  - Chân tín hiệu cho phép ghi;
  - Cho phép ghi vào dòng nhớ khi WE = 0.
- Read enable (RE):
  - Chân tín hiệu cho phép đọc;
  - Cho phép đọc dữ liệu từ dòng nhớ khi RE = 0.

3.2.3 Bộ nhớ cache

1. Cache là gì?
2. Vai trò của cache
3. Các nguyên lý hoạt động cơ bản của cache
4. Trao đổi dữ liệu
5. Các tham số hiệu năng của bộ nhớ
6. Các kiến trúc cache
7. Tổ chức cache-Các phương pháp ánh xạ
8. Quá trình đọc/ghi thông tin trong cache
9. Các chính sách thay thế của cache
10. Các biện pháp cải thiện hiệu năng cache

1. Cache là gì?

- Cache là một thành phần trong hệ thống nhớ phân cấp của máy tính:
  - Cache đóng vai trong trung gian, trung chuyển dữ liệu từ bộ nhớ chính về CPU và ngược lại;
- Vị trí của cache:
  - Với các hệ thống CPU cũ, cache thường nằm ngoài CPU
  - Với các CPU mới, cache thường được tích hợp vào trong CPU



1. Cache là gì?

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

2. Vai trò của cache

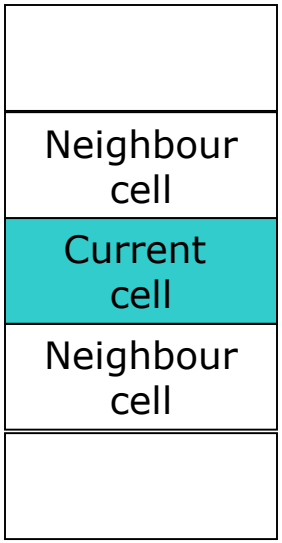
- Tăng hiệu năng hệ thống
  - Dung hoà được CPU có tốc độ cao và bộ nhớ chính có tốc độ thấp;
  - Thời gian trung bình CPU truy nhập dữ liệu từ bộ nhớ chính tiệm cận thời gian truy nhập cache.
- Giảm giá thành sản xuất
  - Nếu hai hệ thống nhớ có cùng giá thành, hệ thống nhớ có cache có tốc độ truy nhập nhanh hơn;
  - Nếu hai hệ thống nhớ có cùng tốc độ, hệ thống nhớ có cache có giá thành rẻ hơn.

3. Các nguyên lý hoạt động của cache

- Cache được coi là bộ nhớ thông minh:
  - 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 → CPU chỉ truy nhập cache → giảm thời gian truy nhập hệ thống nhớ.
- Cache hoạt động dựa trên 2 nguyên lý cơ bản:
  - Nguyên lý lân cận về không gian (Spatial locality)
  - Nguyên lý lân cận về thời gian (Temporal locality)

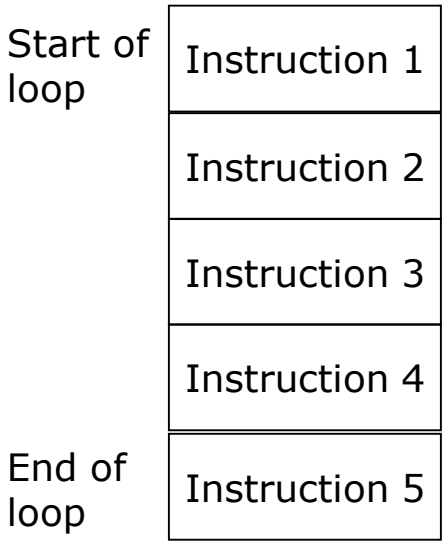
3. Các nguyên lý hoạt động của cache

- 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;
- Áp dụng:
  - Lân cận về không gian được áp dụng cho nhóm lệnh/dữ liệu có tính tuần tự cao;
- Giải thích:
  - Do các lệnh trong một chương trình thường tuần tự → cache đọc cả khối lệnh từ bộ nhớ chính → phủ được lân cận của ô nhớ đang được truy nhập.



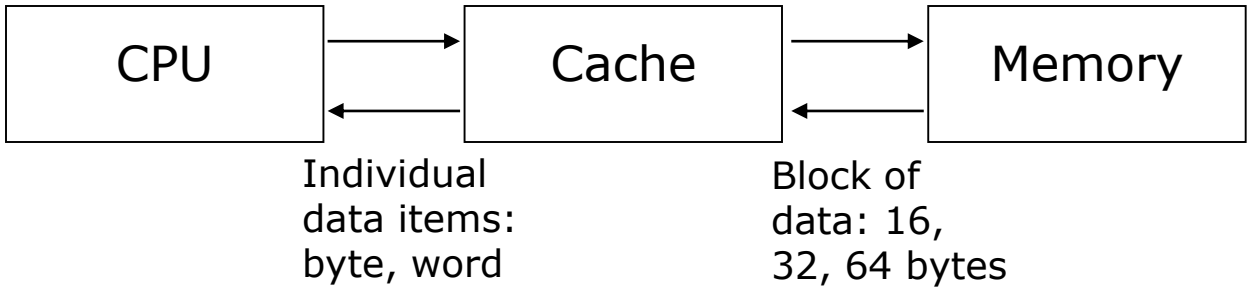
4. Các nguyên lý hoạt động của cache

- Nguyên lý lân cận về thời gian:
  - Nếu một ô nhớ đang được truy nhập thì xác xuất nó được truy nhập lại trong tương lai gần là rất cao;
- Áp dụng:
  - Lân cận về thời gian được áp dụng cho dữ liệu và nhóm lệnh trong vòng lặp;
- Giải thích:
  - Các phần tử dữ liệu thường được cập nhật, sửa đổi thường xuyên;
  - Cache đọc cả khối lệnh từ bộ nhớ chính → phủ được cả khối lệnh của vòng lặp.



4. Trao đổi dữ liệu giữa CPU-Cache-Mem

- CPU đọc/ghi các phần tử dữ liệu đơn lẻ với cache
  - Tại sao?
- Cache đọc/ghi các khối dữ liệu lớn với bộ nhớ chính
  - Tại sao?



5. Các tham số hiệu năng của bộ nhớ cache

- *Hit* là một sự kiện mà CPU truy nhập một mục tin có ở trong cache:
  - Xác suất để có một hit gọi là hệ số hit, hoặc *H*.
  - 0 <= H <= 1
  - Hệ số hit càng cao thì hiệu quả của cache càng cao.
- *Miss* là một sự kiện mà CPU truy nhập một mục tin không có ở trong cache:
  - Xác suất của một miss gọi là hệ số miss, hoặc 1-*H*.
  - 0 <= (1 – H) <= 1
  - Hệ số miss thấp thì hiệu quả của cache càng cao.
- Thời gian trễ truy cập trung bình

5. Hiệu năng cache

- Cache Hit và Cache Miss
- Thời gian truy nhập truy bình của một hệ thống nhớ có cache:

$t_{access} = H*t_{cache} + M*(t_{memory}+t_{cache})$

$t_{access} = t_{cache} + (1 - H) * (t_{memory})$

trong đó H hệ số hit.

If t_{cache} = 5ns, t_{memory} = 60ns và H=80%, ta có:

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

If t_{cache} = 5ns, t_{memory} = 60ns và H=95%, ta có:

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

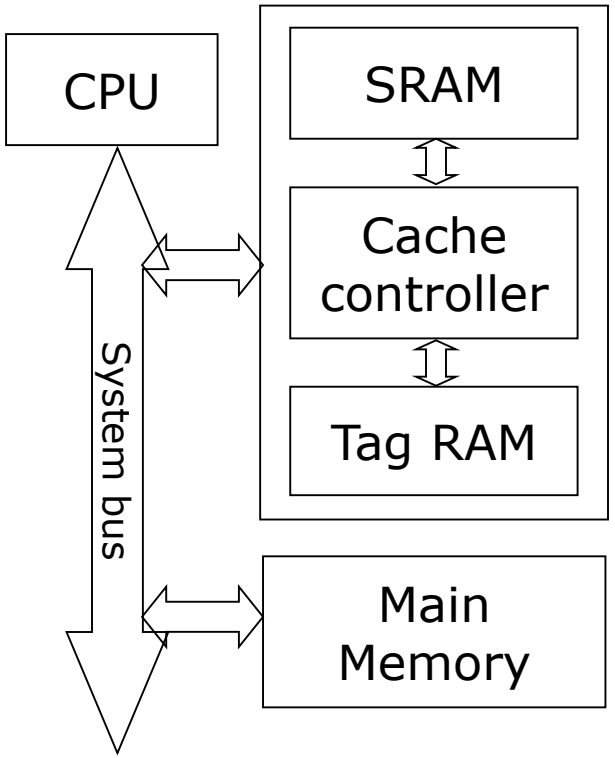
➔ Thời gian truy nhập truy bình tiệm cận thời gian truy nhập cache.

6. Kiến trúc cache – Look aside

- Cache và bộ nhớ chính cùng kết nối với bus hệ thống;
- Cache và bộ nhớ chính "thấy" chu kỳ bus của CPU tại cùng một thời điểm;
- Ưu:
  - Thiết kế đơn giản
  - Miss nhanh (tại sao?)
- Nhược:
  - Hit chậm (tại sao?)

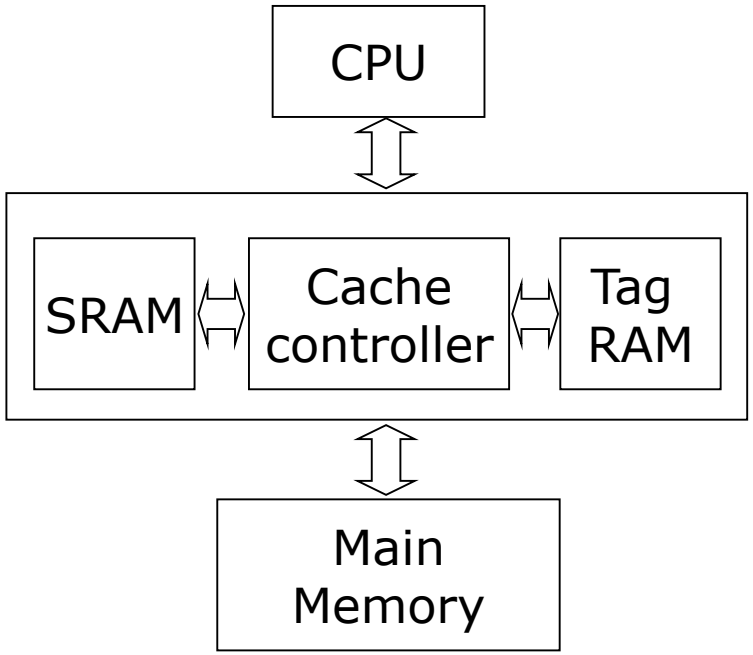
SRAM: RAM lưu dữ liệu cache

Tag RAM: RAM lưu địa chỉ bộ nhớ



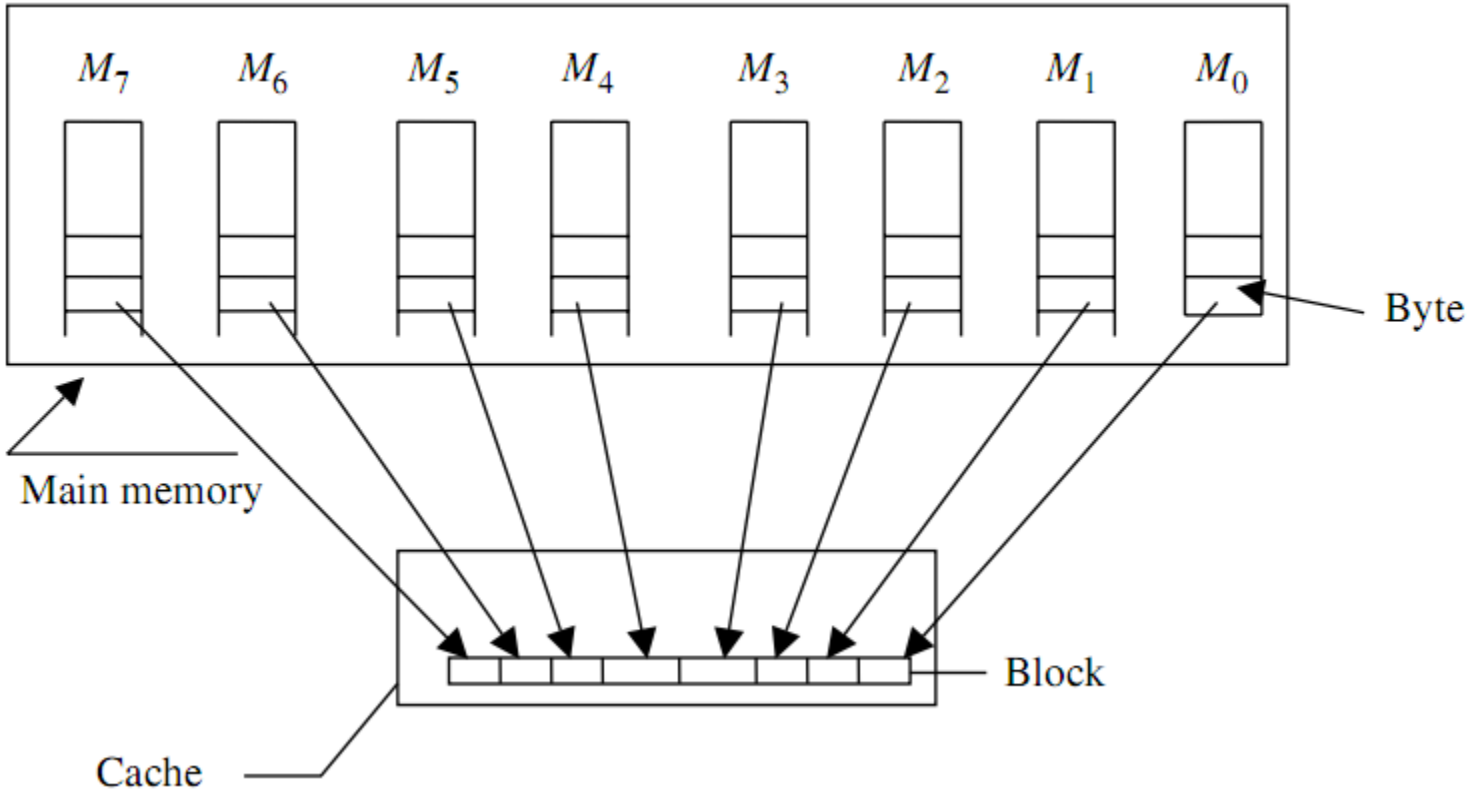
6. Kiến trúc cache – Look through

- Cache nằm giữa CPU và bộ nhớ chính;
- Cache "thấy" chu kỳ bus của CPU trước, sau đó nó chuyển chu kỳ bus cho bộ nhớ chính;
- Ưu:
  - Hit nhanh (tại sao?)
- Nhược:
  - Thiết kế phức tạp
  - Đắt tiền
  - Miss chậm (tại sao?)



7. Tổ chức cache-Các Phương pháp ánh xạ

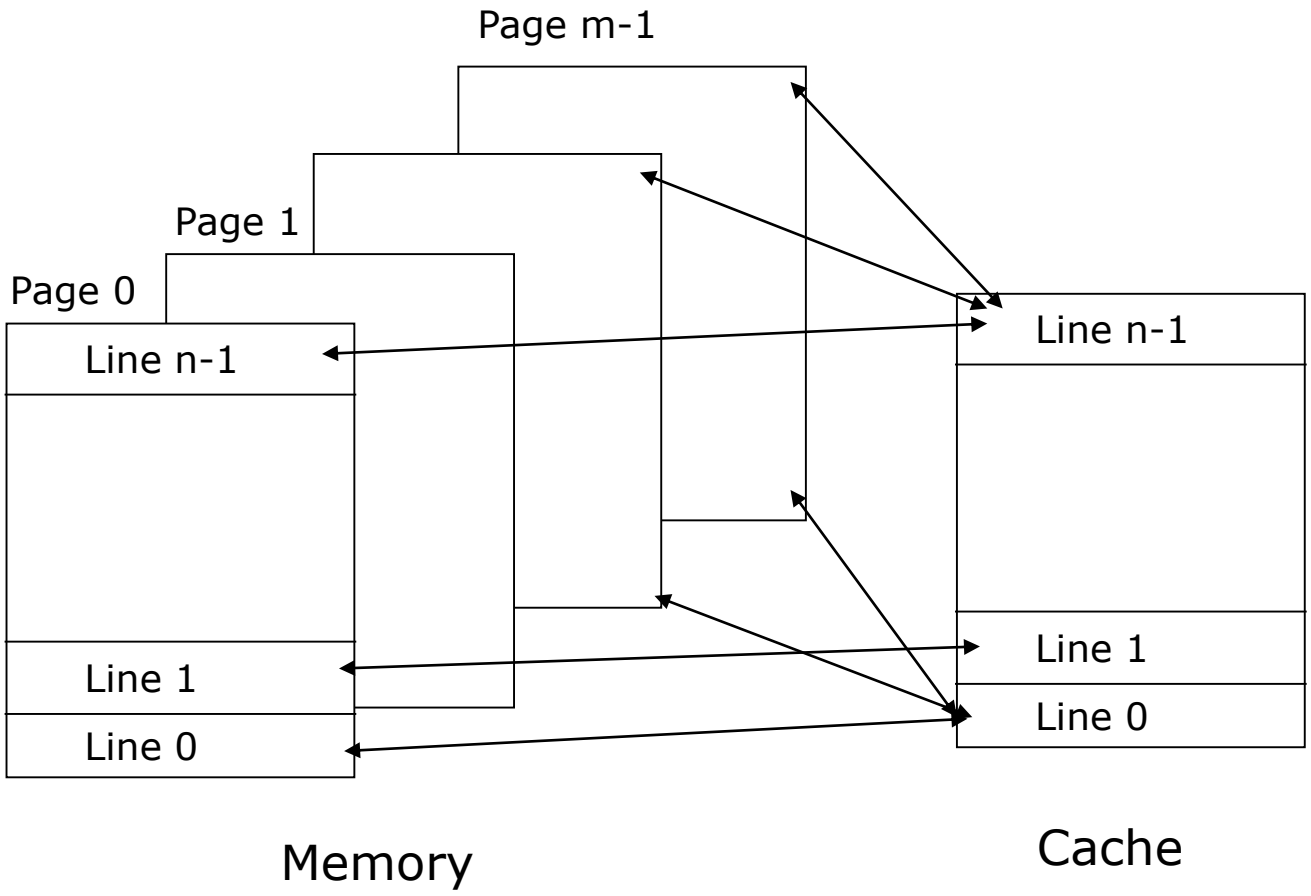
- Tổ chức cache giải quyết vấn đề cache và bộ nhớ chính phối hợp làm việc như thế nào?



7. Tổ chức cache - Các phương pháp

- Ánh xạ trực tiếp (Direct mapping)
  - Đơn giản và nhanh
  - Ánh xạ cứng dễ gây xung đột
- Ánh xạ kết hợp đầy đủ (Fully associative mapping)
  - Phức tạp và chậm
  - Ánh xạ mềm, ít xung đột
- Ánh xạ tập kết hợp (Set associative mapping)
  - Phức tạp và nhanh
  - Ánh xạ mềm, ít xung đột

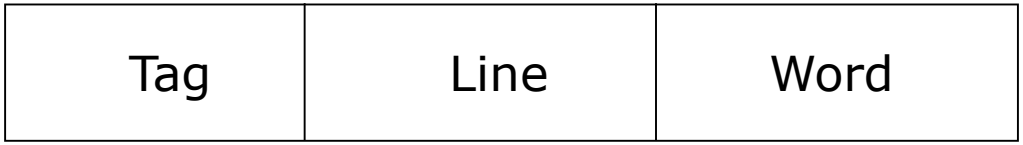
7. Tổ chức cache – Ánh xạ trực tiếp



7. Tổ chức cache – Ánh xạ trực tiếp

- Cache:
  - Chia thành *n* khối hoặc dòng, từ $Line_0$ đến $Line_{n-1}$
- Bộ nhớ chính:
  - Chia thành *m* trang, từ $page_0$ đến $page_{m-1}$.
  - Một trang bộ nhớ có kích thước bằng cache
  - Mỗi trang có *n* dòng, từ $Line_0$ đến $Line_{n-1}$
- Ánh xạ:
  - $Line_0$ của ($page_0$ đến $page_{m-1}$) ánh xạ đến $Line_0$ của cache;
  - $Line_1$ của ($page_0$ đến $page_{m-1}$) ánh xạ đến $Line_1$ của cache;
  - ....
  - $Line_{n-1}$ của ($page_0$ đến $page_{m-1}$) ánh xạ đến $Line_{n-1}$ của cache;

7. Tổ chức cache – Địa chỉ ánh xạ trực tiếp



- *Tag* (bit): địa chỉ trang trong bộ nhớ
- *Line* (bit): địa chỉ dòng trong cache
- *Word* (bit): địa chỉ của từ trong dòng

7. Tổ chức cache – Địa chỉ ánh xạ trực tiếp

- Ví dụ
  - Vào:
    - Dung lượng bộ nhớ = 4GB
    - Dung lượng cache = 1MB
    - Kích thước dòng = 32 byte
    - Dung lượng word= 8 bit =1B
  - Ra:
    - Ta có kích thước dòng Line = 32 byte = 2^5 B→ Số lượng word trong 1 dòng = DL dòng cache/ word máy tính = 2^5 B/1B = 2^5 (words)

      → Word (bit) = 5 bit
    - Ta có dung lượng Cache = 1MB = 2^{20} B→ Số lượng dòng cache trong cache= DL cache/DL dòng cache= 2^{20} B/ 2^5 B= 2^{15} (dòng)

      → Line (bit) = 15 bit
    - Cách 1: Ta có dung lượng bộ nhớ 4GB = 2^{32} B ➔ Số lượng word=DL bộ nhớ/DL word = 2^{32} B/1B= 2^{32} (words)→ tổng cộng có 32 bít địa chỉ để địa chỉ hoá các ô nhớ:

      Tag (bit) = 32 bit địa chỉ – Line (bit) – Word (bit)= 32 – 5 – 15 = 12 bit.

7. Tổ chức cache – Địa chỉ ánh xạ trực tiếp

- Ví dụ
  - Vào:
    - Dung lượng bộ nhớ = 4GB
    - Dung lượng cache = 1MB
    - Kích thước dòng = 32 byte
    - Dung lượng word= 8 bit =1B
  - Ra:
    - Ta có kích thước dòng Line = 32 byte = 2^5 B→ Số lượng word trong 1 dòng = DL dòng cache/ word máy tính = 2^5 B/1B = 2^5 (words)

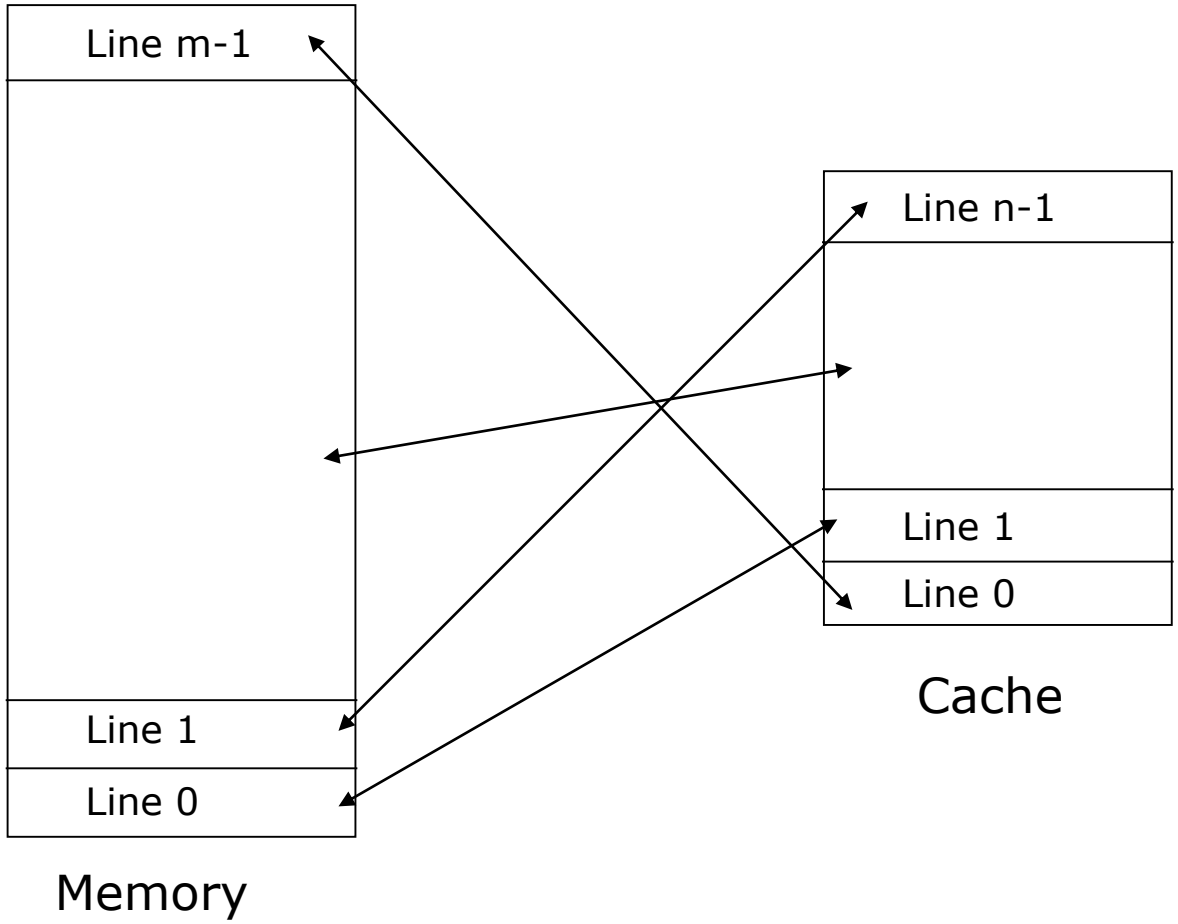
      → Word (bit) = 5 bit
    - Ta có dung lượng Cache = 1MB = 2^{20} B→ Số lượng dòng cache trong cache= DL cache/DL dòng cache= 2^{20} B/ 2^5 B= 2^{15} (dòng)

      → Line (bit) = 15 bit
    - Cách 2: Ta có dung lượng bộ nhớ 4GB = 2^{32} B ➔ Số lượng pages =DL bộ nhớ/DL cache = 2^{32} B/ 2^{20} B = 2^{12} (pages)
      → Tag (bit) = 12 bit.

7. Tổ chức cache – Ánh xạ trực tiếp

- Ưu:
  - Thiết kế đơn giản
  - Nhanh do ánh xạ là cố định: khi biết địa chỉ ô nhớ có thể tìm được vị trí của nó trong cache rất nhanh chóng.
- Nhược:
  - Do ánh xạ cố định nên dễ gây xung đột (tại sao?)
  - Hệ số hit không cao.

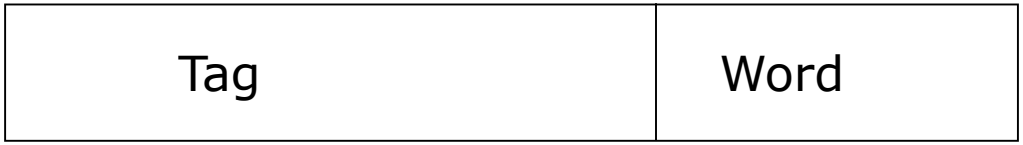
7. T.chức cache – Ánh xạ kết hợp đầy đủ



7. T.chức cache – Ánh xạ kết hợp đầy đủ

- Cache:
  - Được chia thành *n* khối hoặc dòng, từ $Line_0$ đến $Line_{n-1}$
- Bộ nhớ:
  - Được chia thành *m* khối hoặc dòng, từ $Line_0$ đến $Line_{m-1}$.
  - Kích thước mỗi dòng cache bằng kích thước một dòng bộ nhớ
  - Số dòng trong bộ nhớ rất lớn so với số dòng của cache (m >> n).
- Ánh xạ:
  - Một dòng trong bộ nhớ có thể được ánh xạ vào một dòng bất kỳ trong cache;
  - $Line_i$ trong bộ nhớ có thể được ánh xạ vào $Line_j$ của cache;

7. T.chức cache – Ánh xạ kết hợp đầy đủ



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

7. T.chức cache – Ánh xạ kết hợp đầy đủ

- Ví dụ về địa chỉ cache:
  - Vào:
    - Dung lượng bộ nhớ = 4GB
    - Dung lượng cache = 1MB
    - Kích thước dòng = 32 byte
  - Ra:
    - Ta có kích thước dòng Line = 32 byte = 2^5 B→ Số lượng word trong 1 dòng = DL dòng cache/ word máy tính = 2^5 B/1B = 2^5 (words)

      → Word (bit) = 5 bit
    - Cách 1: Ta có dung lượng bộ nhớ 4GB = 2^{32} B ➔ Số lượng word=DL bộ nhớ/DL word = 2^{32} B/1B= 2^{32} (Word)→ tổng cộng có 32 bít địa chỉ để địa chỉ hoá các ô nhớ:

      Tag (bit) = 32 bit địa chỉ – Word (bit)= 32 – 5 = 27 bit.

7. T.chức cache – Ánh xạ kết hợp đầy đủ

❖ Ví dụ về địa chỉ cache:

- Vào:
  - Dung lượng bộ nhớ = 4GB
  - Dung lượng cache = 1MB
  - Kích thước dòng = 32 byte
- Ra:
  - Ta có kích thước dòng Line = 32 byte = 2^5 B→ Số lượng word trong 1 dòng = DL dòng cache/ word máy tính = 2^5 B/1B = 2^5 (words)

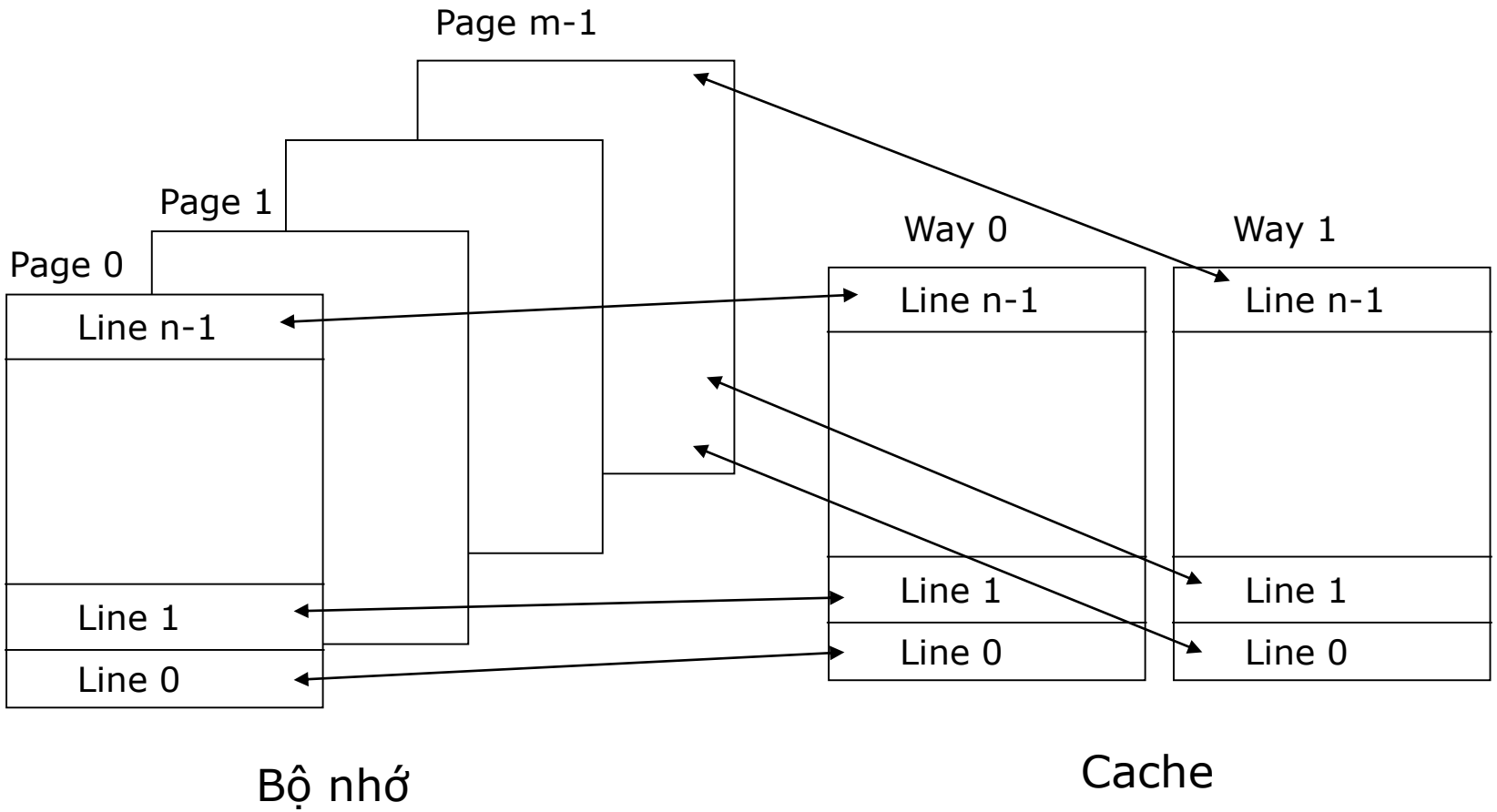
    → Word (bit) = 5 bit
  - Cách 2: Ta có dung lượng bộ nhớ 4GB = 2^{32} B ➔ Số lượng dòng=DL bộ nhớ/DL dòng = 2^{32} B/2^5 B= 2^{27} dòng

    → Tag (bit) = 27 bit.

7. T.chức cache – Ánh xạ kết hợp đầy đủ

- Ưu:
  - Giảm được xung đột do ánh xạ là không cố định
  - Hệ số Hit cao hơn ánh xạ trực tiếp.
- Nhược:
  - Chậm do cần phải tìm địa chỉ ô nhớ trong cache
  - Phức tạp do cần có n bộ so sánh địa chỉ bộ nhớ trong cache.
- Thường được sử dụng với cache có dung lượng nhỏ.

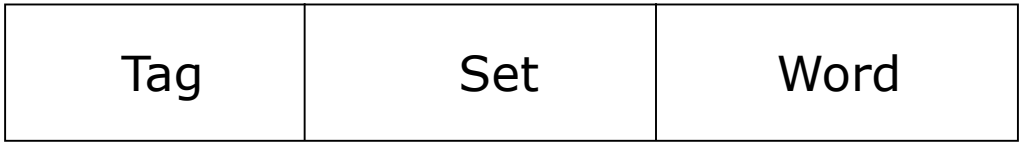
7. T.chức cache – Ánh xạ tập kết hợp



7. T.chức cache – Ánh xạ tập kết hợp

- Cache:
  - Được chia thành k đường (way) với kích thước bằng nhau;
  - Mỗi đường được chia thành *n* dòng, từ $Line_0$ đến $Line_{n-1}$
- Bộ nhớ:
  - Được chia thành *m* trang, từ $page_0$ đến $page_{m-1}$.
  - Kích thước một trang bộ nhớ bằng kích thước đường của cache
  - Mỗi trang có *n* dòng, từ $Line_0$ đến $Line_{n-1}$
- Ánh xạ:
  - Ánh xạ trang đến đường (ánh xạ mềm dẻo):
    - Một trang của bộ nhớ có thể ánh xạ đến một đường bất kỳ của cache.
  - Ánh xạ dòng của trang đến dòng của đường (ánh xạ cố định):
    - $Line_0$ của $page_i$ ánh xạ đến $Line_0$ của way_j;
    - $Line_1$ của $page_i$ ánh xạ đến $Line_1$ của way_j;
    - ....
    - $Line_{n-1}$ của $page_i$ ánh xạ đến $Line_{n-1}$ của way_j;

7. T.chức cache – Ánh xạ tập kết hợp



- Địa chỉ cache:
  - *Tag* (bit) là địa chỉ của trang trong bộ nhớ
  - *Set* (bit) là địa chỉ của dòng trong đường cache
  - *Word* (bit) là địa chỉ của từ trong dòng

7. T.chức cache – Ánh xạ tập kết hợp

- Ví dụ:
  - Vào:
    - Dung lượng bộ nhớ = 4GB
    - Dung lượng cache = 1MB, 2 đường
    - Kích thước dòng = 32 byte
  - Ra:
    - Kích thước dòng Line = 32 byte = 2^5 →Số lượng Word .....

    →Word (bit) = 5 bit
    - Dung lượng Cache = 1MB = 2^{20} →Số lượng line trong Cache Way = Dung lượng đường cache/dung lượng line->Số lượng line trong đường cache = 2^{20} / 2 đường / 2^5 = 2^{14} dòng/đường → Set = 14 bit
    - Cách 1: Ta có dung lượng bộ nhớ 4GB = 2^{32} B ➔ Số lượng word=DL bộ nhớ/DL word = 2^{32} B/1B= 2^{32} (Word)→ tổng cộng có 32 bít địa chỉ để địa chỉ hoá các ô nhớ:

      Tag = 32bit địa chỉ – Set – Word = 32 – 14 – 5 = 13 bit.

7. T.chức cache – Ánh xạ tập kết hợp

- Ví dụ:
  - Vào:
    - Dung lượng bộ nhớ = 4GB
    - Dung lượng cache = 1MB, 2 đường
    - Kích thước dòng = 32 byte
  - Ra:
    - Kích thước dòng Line = 32 byte = 2^5 →Số lượng Word

    →Word (bit) = 5 bit
    - Dung lượng Cache = 1MB = 2^{20} →Số lượng line trong Cache Way = Dung lượng đường cache/dung lượng line->Số lượng line trong đường cache = 2^{20} / 2 đường / 2^5 = 2^{14} dòng/đường → Set = 14 bit
    - Cách 2: Ta có dung lượng bộ nhớ 4GB = 2^{32} B ➔ Số lượng pages =DL bộ nhớ/(DL cache/Số đường cache) = 2^{32} B*2/ 2^{20} B = 2^{13} (pages)
    → Tag (bit) = 13 bit.

7. T.chức cache – Ánh xạ tập kết hợp

- Ưu:
  - Nhanh do ánh xạ trực tiếp được sử dụng cho ánh xạ dòng (chiếm số lớn ánh xạ);
  - Giảm xung đột do ánh xạ từ các trang bộ nhớ đến các đường cache là mềm dẻo.
  - Hệ số Hit cao hơn.
- Nhược:
  - Phức tạp trong thiết kế và điều khiển vì cache được chia thành một số đường.

8. Đọc/ghi thông tin trong cache

❖ Đọc thông tin:

- Trường hợp hit (mẩu tin cần đọc có trong cache)
  - Mẩu tin được đọc từ cache vào CPU;
  - Bộ nhớ chính không tham gia.
- Trường hợp miss (mẩu tin cần đọc không có trong cache)
  - Mẩu tin trước hết được đọc từ bộ nhớ chính vào cache;
  - Sau đó nó được chuyển từ cache vào CPU.

  ➔ đây là trường hợp miss penalty: thời gian truy nhập mẩu tin bằng tổng thời gian truy nhập cache và bộ nhớ chính.

8. Đọc/ghi thông tin trong cache

- Ghi thông tin:
  - Trường hợp hit (mẩu tin cần ghi có trong cache)
    - Ghi thẳng (write through): mẩu tin được ghi ra cache và bộ nhớ chính đồng thời;
    - Ghi trễ (write back): mẩu tin trước hết được ghi ra cache và dòng chứa mẩu tin được ghi ra bộ nhớ chính khi dòng đó bị thay thế.
  - Trường hợp miss (mẩu tin cần ghi không có trong cache)
    - Ghi có đọc lại (write allocate / fetch on write): mẩu tin trước hết được ghi ra bộ nhớ chính và sau đó dòng chứa mẩu tin được đọc vào cache;
    - Ghi không đọc lại (write non-allocate): mẩu tin chỉ được ghi ra bộ nhớ chính (dòng chứa mẩu tin không được đọc vào cache).

9. Các chính sách thay thế dòng cache

- Vì 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.
- Chính sách thay thế (replacement policies) xác định các dòng cache nào được chọn bị thay thế bởi các dòng khác từ bộ nhớ.
- Các chính sách thay thế:
  - Ngẫu nhiên (Random)
  - Vào trước ra trước (FIFO)
  - Thay thế các dòng ít được sử dụng gần đây nhất (LRU).

9. Các chính sách thay thế dòng cache

- Thay thế ngẫu nhiên (Radom Replacement):
  - Các dòng cache được chọn ngẫu nhiên để thay thế
  - Ưu:
    - Cài đặt đơn giản
  - Nhược:
    - Hệ số miss cao:
      - Thay thế ngẫu nhiên không xem xét đến các dòng cache đang thực sự được sử dụng
      - Nếu một dòng cache đang được sử dụng và bị thay thế → xảy ra miss và nó lại cần được đọc từ bộ nhớ chính vào cache.

9. Các chính sách thay thế dòng cache

- Thay thế kiểu vào trước ra trước (FIFO-First In First Out):
  - Các dòng cache được đọc vào cache trước sẽ bị thay thế trước
  - Ưu:
    - Có hệ số miss thấp hơn o với thay thế ngẫu nhiên (tại sao?)
  - Nhược:
    - Hệ số miss vẫn còn cao
      - Thay thế vẫn chưa thực sự xem xét đến các dòng cache đang được sử dụng. Một dòng cache "già" vẫn có thể đang được sử dụng.
    - Cài đặt phức tạp do cần có mạch điện tử để theo dõi trật tự nạp các dòng bộ nhớ vào cache.

9. Các chính sách thay thế dòng cache

- Thay thế các dòng ít được sử dụng gần đây nhất (LRU-Least Recently Used):
  - Các dòng cache ít được sử dụng gần đây nhất được lựa chọn để tha thế.
  - Ưu:
    - Có hệ số miss thấp nhất so với thay thế ngẫu nhiên và thay thế FIFO
    - Do thay thế LRU có xem xét đến các dòng đang được sử dụng
  - Nhược:
    - Cài đặt phức tạp do cần có mạch điện tử để theo dõi tần suất sử dụng các dòng cache.

10. Hiệu năng cache – Các yếu tố ảnh hưởng

❖ Các yếu tố ảnh hưởng đến hiệu năng cache:

- Kích thước cache:
  - Kích thước cache nên lớn hay nhỏ?
- Tách cache:
  - Cache được tách thành 2 phần: cache lệnh (I-Cache) và D-Cache
- Tạo cache thành nhiều mức:
  - Cache được thiết kế thành nhiều mức: L1 – L2 – L3, ... với kích thước tăng dần.

10. Hiệu năng 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%).

10. Hiệu năng cache – Các yếu tố ảnh hưởng

- Kích thước cache:
  - Cache có kích thước lớn:
    - Có thể tăng được số dòng bộ nhớ lưu trong cache
    - Giảm tần suất tráo đổi các dòng cache của các chương trình khác nhau với bộ nhớ chính
    - Cache lớn thường chậm hơn cache nhỏ (tại sao?)
      - Không gian tìm kiếm địa chỉ ô nhớ lớn hơn
  - Xu hướng tương lai: cache càng lớn càng tốt (tại sao?)
    - Hỗ trợ đa nhiệm tốt hơn
    - Hỗ trợ xử lý song song tốt hơn
    - Hỗ trợ tốt hơn các hệ thống CPU nhiều nhân

10. Hiệu năng cache – Các yếu tố ảnh hưởng

- Tách cache:
  - Cache có thể được tách thành cache lệnh (I-Cache) và cache dữ liệu (D-Cache) để cải thiện hiệu năng, do:
    - Dữ liệu và lệnh có tính lân cận khác nhau;
    - Dữ liệu thường có tính lân cận về thời gian cao hơn lân cận về không gian; lệnh có tính lân cận về không gian cao hơn lân cận về thời gian;
    - Cache lệnh chỉ cần hỗ trợ thao tác đọc; cache dữ liệu cần hỗ trợ cả 2 thao tác đọc và ghi → tách cache giúp tối ưu hoá dễ dàng hơn;
    - Tách cache hỗ trợ nhiều lệnh truy nhập đồng thời hệ thống nhớ → giảm xung đột tài nguyên cho CPU pipleline.

10. Hiệu năng cache – Các yếu tố ảnh hưởng

- Tách cache:
  - Trên thực tế, hầu hết cache L1 được tách thành 2 phần:
    - I-Cache (Instruction Cache): cache lệnh
    - D-Cache (Data Cache): cache dữ liệu
    - I-Cache và D-Cache thường hỗ trợ nhiều lệnh truy nhập
  - Các cache ở mức cao hơn không được tách. Tại sao?
    - Cache L1 được tách vì nó ở gần CPU nhất; CPU trực tiếp đọc ghi lên cache L1. Cache L1 cần hỗ trợ nhiều lệnh truy nhập đồng thời và các biện pháp tối ưu hoá;
    - Các mức cao hơn của cache ít được tách do:
      - Điều khiển phức tạp
      - Hiệu quả mang lại không thực sự cao, do CPU không trực tiếp đọc/ghi các mức cache này. Hơn nữa, các mức cache cao hơn trao đổi dữ liệu với cache L1 theo khối, nên việc hỗ trợ nhiều lệnh truy nhập đồng thời không có nhiều ý nghĩa.

10. Hiệu năng cache – Các yếu tố ảnh hưởng

❖ Tạo cache thành nhiều mức:

- Cải thiện được hiệu năng hệ thống do hệ thống cache nhiều mức có khả năng dung hoà tốt hơn tốc độ của CPU với bộ nhớ chính.

CPU	L1	L2	L3	Bộ nhớ chính
1ns	5ns	15ns	30ns	60ns
1ns	5ns			60ns

- Trên thực tế, đa số cache được tổ chức thành 2 mức: L1 và L2. Một số cache có 3 mức: L1, L2 và L3.
- Giảm giá thành hệ thống nhớ.

10. H.năng cache – Các biện pháp giảm miss

- Cache tốt:
  - Hệ số hit cao
  - Hệ số miss thấp
  - Nếu xảy ra miss thì không quá chậm
- Các loại miss:
  - Miss bắt buộc (Compulsory misses): thường xảy ra tại thời điểm chương trình được kích hoạt, khi mã chương trình đang được tải vào bộ nhớ và chưa được nạp vao cache.
  - Miss do dung lượng (Capacity misses): thường xảy ra do kích thước của cache hạn chế, đặc biết trong môi trường đa nhiệm. Do kích thước cach nhỏ nên mã của các chương trình thường xuyên bị tráo đổi giữa bộ nhớ và cache.
  - Miss do xung đột (Conflict misses): xảy ra khi có nhiều dòng bộ nhớ cùng cạnh tranh một dòng cache.

10. H.năng cache – Các biện pháp giảm miss

- Tăng kích thước dòng cache:
  - Giảm miss bắt buộc
    - Dòng kích thước lớn sẽ có khả năng bao phủ lận tốt hơn → giảm miss bắt buộc;
  - Tăng miss do xung đột
    - Dòng kích thước lớn sẽ làm giảm số dòng cache → tăng mức độ cạnh tranh → tăng miss do xung đột
  - Dòng kích thước lớn có thể gây lãng phí dung lượng cache. Do dòng lớn nên có thể có nhiều phần của dòng cache không bao giờ được sử dụng.
  - Kích thước dòng thường dùng hiện nay là 64 bytes.

10. H.năng cache – Các biện pháp giảm miss

- Tăng mức độ liên kết của cache (tăng số đường cache):
  - Giảm miss xung đột:
    - Tăng số đường cache → tăng tính mềm dẻo của ánh xạ trang bộ nhớ - đường cache (nhiều lựa chọn hơn) → giảm miss xung đột.
  - Làm cache chậm hơn:
    - Tăng số đường cache → tăng không gian tìm kiếm địa chỉ ô nhớ → làm cache chậm hơn.

3.2.4 Câu hỏi ôn tập

1. Hệ thống bộ nhớ phân cấp: đặc điểm, vai trò.
2. ROM là gì? các loại ROM.
3. RAM, SRAM, DRAM là gì? Cấu tạo của SRAM và DRAM.
4. Bộ nhớ cache:
   a. Cache là gì? vai trò và nguyên lý hoạt động.
   b. Kiến trúc cache
   c. Tổ chức/ánh xạ cache
   d. Đọc ghi thông tin trong cache
   e. Các chính sách thay thế dòng cache
   f. Hiệu năng, các yếu tố ảnh hưởng và các biện pháp cải thiện hiệu năng cache.