KIẾN TRÚC MÁY TÍNH

Bộ môn Mạng và an toàn thông tin
Giảng viên: TS. Đoàn Thị Quế

Chương 4

Bộ nhớ cache

Nội dung chính

4.1 Tổng quan về bộ nhớ máy tính
4.2 Nguyên lý của bộ nhớ cache
4.3 Các yếu tố trong thiết kế bộ nhớ cache
4.4 Tổ chức cache của Pentium 4
4.5 Tổ chức cache trong ARM

4.1 Tổng quan về bộ nhớ máy tính

Các đặc điểm của bộ nhớ máy tính

Vị trí Bên trong (vd: thanh ghi, cache, bộ nhớ chính Bên ngoài (vd: đĩa quang, đĩa từ, băng từ) **Dung lượng** Số lượng từ Số lượng byte **Đơn vị truyền** Từ Khối **Phương pháp truy cập** Tuần tự Trực tiếp Ngẫu nhiên Kết hợp	**Hiệu suất** Thời gian truy cập Chu kỳ bộ nhớ Tốc độ truyền tải **Loại vật lý** Bán dẫn Từ Quang học Quang-từ **Tính chất vật lý** Khả biến/Không khả biến Có thể xóa/không xóa được **Tổ chức** Module bộ nhớ

Các đặc điểm của bộ nhớ máy tính

a. Vị trí

- Bộ nhớ ngoài bao gồm các thiết bị lưu trữ ngoại vi có thể truy cập vào bộ xử lý thông qua bộ điều khiển I/O
- Bộ nhớ trong:
  - Bộ nhớ chính (Main Memory)
  - Bộ nhớ đệm (Cache)
  - Các thanh ghi (Registers): bộ nhớ cục bộ riêng của bộ xử lý

b. Dung lượng

- Bộ nhớ thường được biểu diễn dưới dạng byte
  - Đối với bộ nhớ trong, dung lượng thường được biểu diễn dưới dạng byte hoặc từ (word). Độ dài từ phổ biến là 8, 16 và 32 bit.
  - Dung lượng bộ nhớ ngoài thường được biểu diễn bằng byte.

c. Đơn vị truyền

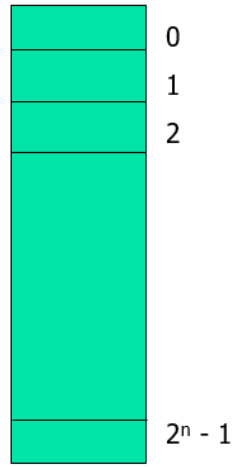
- Đối với bộ nhớ trong, đơn vị truyền bằng số lượng đường điện đi vào và ra khỏi module bộ nhớ
- Với bộ nhớ ngoài, đơn vị truyền thường lớn hơn rất nhiều, thường được gọi là các khối (block)

Một số khái niệm đối với bộ nhớ trong

- **Từ (word):**
  - Từ (word) đơn vị "tự nhiên" của bộ nhớ.
  - Kích thước từ thường bằng số bit biểu diễn một số nguyên và bằng kích thước lệnh. (Intel x86 có nhiều độ dài lệnh khác nhau nhưng kích thước từ là 32b)
- **Đơn vị đánh địa chỉ**:
  - Đơn vị đánh địa chỉ có thể là byte hoặc từ (word).
  - Số bit địa chỉ là *n* thì số lượng các đơn vị đánh địa chỉ là 2^n
- **Dung lượng bộ nhớ (DLBN)**
  - $DLBN = 2^n \cdot W_{đv}$ (byte)

    2^n: Số lượng đơn vị đánh địa chỉ, gọi tắt là số địa chỉ

    $W_{đv}$: Kích thước của một đơn vị đánh địa chỉ (tính theo byte)



- **Đơn vị truyền:**
  - Với bộ nhớ trong, đơn vị truyền bằng số lượng các bit được gửi đến hoặc đi từ bộ nhớ ở cùng một thời điểm (số đường dữ liệu đến hoặc đi từ bộ nhớ).
  - Đơn vị truyền không nhất thiết phải bằng kích thước một word hoặc đơn vị đánh địa chỉ.

Ví dụ

1. Vi xử lý Intel x86-32b, kết nối bus gồm: 32 đường địa chỉ, 16 đường dữ liệu. Giả sử với bộ nhớ tổ chức dưới dạng các đơn vị đánh địa chỉ là 16b. Hãy cho biết:
   a. Kích thước word của bộ nhớ trên
   b. Dung lượng tối đa của bộ nhớ mà vi xử lý có thể quản lý được
   c. Đơn vị truyền của bộ nhớ trên
   d. Để thực hiện một lệnh: cộng 2 số (trong bộ nhớ) và ghi kết quả vào các ngăn nhớ khác thì vi xử lý sẽ phải thực hiện bao nhiêu thao tác đọc, ghi bộ nhớ

d. Phương pháp truy cập

Truy cập tuần tự	Truy cập trực tiếp	Truy cập ngẫu nhiên	Kết hợp
• Bộ nhớ được tổ chức thành các đơn vị dữ liệu được gọi là bản ghi (record) • Truy cập được thực hiện tuần tự • Thời gian truy cập biến đổi • Ví dụ: băng từ	• Có một cơ chế đọc-ghi chia sẻ • Mỗi khối hoặc bản ghi có một địa chỉ duy nhất dựa trên vị trí vật lý • Thời gian truy cập biến đổi • Ví dụ: đĩa từ	• Mỗi vị trí trong bộ nhớ có một cơ chế định địa chỉ riêng • Thời gian truy cập vào một vị trí nhất định không đổi và không phụ thuộc vào chuỗi các truy cập trước đó • Một vị trí bất kỳ có thể được chọn ngẫu nhiên, định địa chỉ và truy cập trực tiếp • Ví dụ: bộ nhớ chính	• Một word được truy xuất dựa trên một phần nội dung thay vì địa chỉ của nó • Mỗi vị trí có cơ chế định địa chỉ riêng. Thời gian truy xuất là không đổi, không phụ thuộc vào vị trí hoặc các truy cập trước đó • Ví dụ: bộ nhớ Cache

e. Hiệu năng

Hai đặc điểm quan trọng nhất của bộ nhớ: dung lượng và hiệu năng

Ba tham số hiệu năng được sử dụng:

Thời gian truy cập (Access time)

- Đối với bộ nhớ truy cập ngẫu nhiên, nó là thời gian cần để thực hiện 1 thao tác đọc hoặc ghi
- Đối với bộ nhớ truy cập không ngẫu nhiên, nó là thời gian cần để định vị đầu đọc-ghi vào vị trí mong muốn

Chu kỳ bộ nhớ (Memory cycle time)

- Là khoảng thời gian ngắn nhất giữa 2 lần truy cập kế tiếp
- Với bộ nhớ truy cập ngẫu nhiên: Thời gian truy cập cộng với thời gian cần thiết trước khi lần truy cập thứ hai có thể bắt đầu
- Khoảng thời gian thêm này có thể cần thiết để các để các tín hiệu cũ trên đường truyền triệt tiêu hoặc để khôi phục lại dữ liệu bị hỏng
- Chu kỳ bộ nhớ liên quan đến hệ thống bus, không liên quan bộ xử lý

Tốc độ truyền tải (Transfer rate)

- Tốc độ truyền dữ liệu vào hoặc ra khỏi bộ nhớ

f. Đặc tính vật lý của bộ nhớ

- Các dạng phổ biến nhất là: Bộ nhớ bán dẫn, Bộ nhớ bề mặt từ, Bộ nhớ quang, Bộ nhớ quang từ

- Một số đặc điểm vật lý quan trọng:

1. Đặc điểm lưu trữ dữ liệu

- Bộ nhớ khả biến (Volatile memory): thông tin bị suy yếu hoặc bị mất khi nguồn điện tắt

  VD: RAM, Cache

- Bộ nhớ không khả biến(Non-volatile memory): thông tin một khi đã được ghi thì sẽ không bị mất trừ khi cố tình thay đổi kể cả không có nguồn cung cấp

  VD: ROM, USB, HDD,…

2. Công nghệ sản xuất:

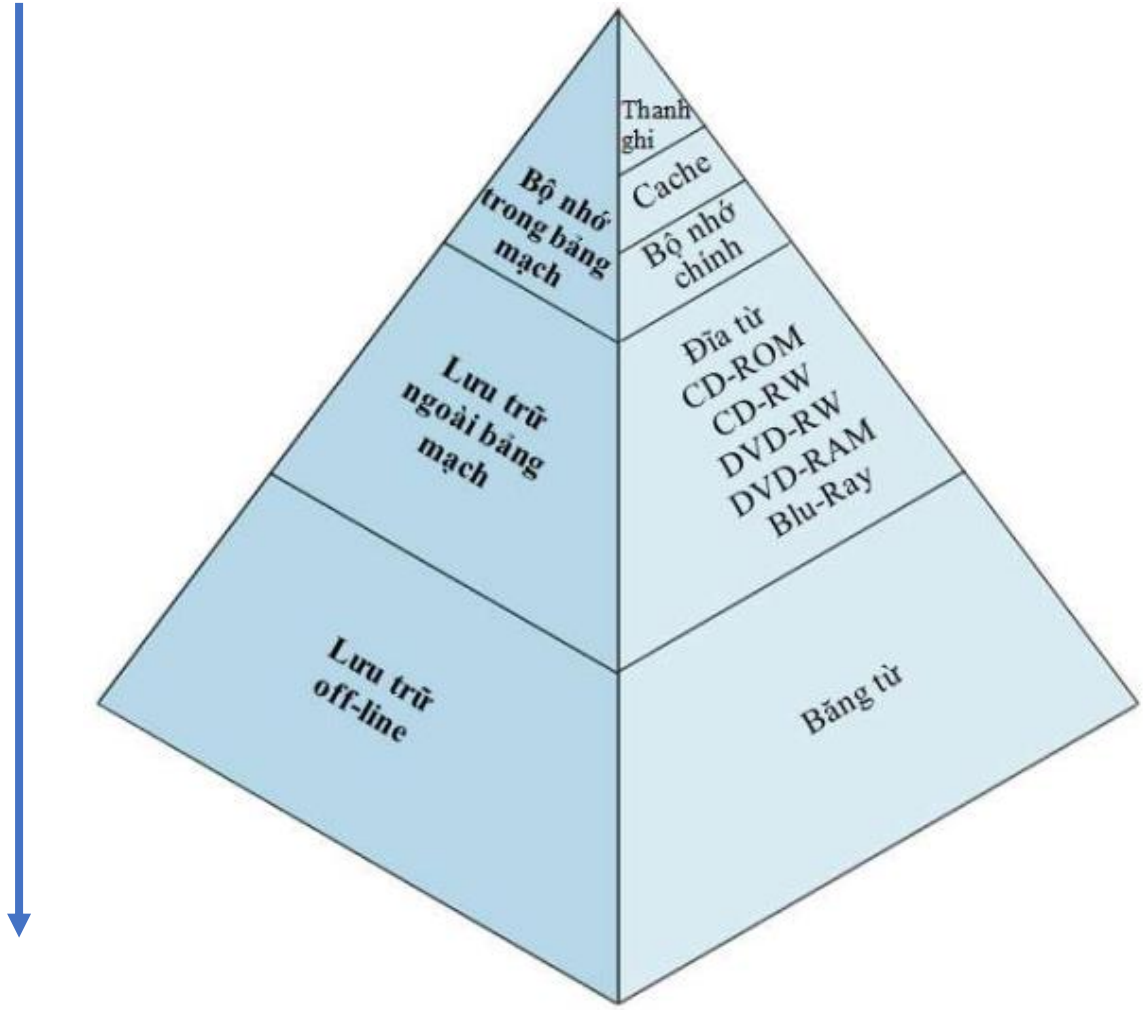
- Bộ nhớ bề mặt từ (Magnetic-surface memories): HDD, Tape
- Bộ nhớ bán dẫn (Semiconductor memory): RAM, ROM, Cache,…
- Bộ nhớ không xoá được (Nonerasable memory): Không thể thay đổi, trừ khi phá hủy các khối lưu trữ. VD: ROM

g. Tổ chức bộ nhớ: mô hình phân cấp bộ nhớ

- Thiết kế bộ nhớ của máy tính cần trả lời ba câu hỏi:
  - Dung lượng bao nhiêu là đủ?
  - Tốc độ ra sao?
  - Giá thành như thế nào?
- Cần có sự cân đối giữa dung lượng, thời gian truy cập và chi phí
  - Thời gian truy cập nhanh hơn, chi phí lớn hơn cho mỗi bit
  - Dung lượng lớn hơn, chi phí nhỏ hơn cho mỗi bit
  - Dung lượng lớn hơn, thời gian truy cập chậm hơn
- Giải pháp:
  - Không dựa hoàn toàn vào một thành phần hoặc công nghệ bộ nhớ
  - Sử dụng một hệ thống phân cấp bộ nhớ

Bộ nhớ phân cấp - Sơ đồ

- Dung lượng tăng
- Chi phí trên bit giảm
- Thời gian truy cập tăng



Nội dung chính

4.1 Tổng quan về bộ nhớ máy tính

4.2 Nguyên lý của bộ nhớ cache

4.3 Các yếu tố trong thiết kế bộ nhớ cache

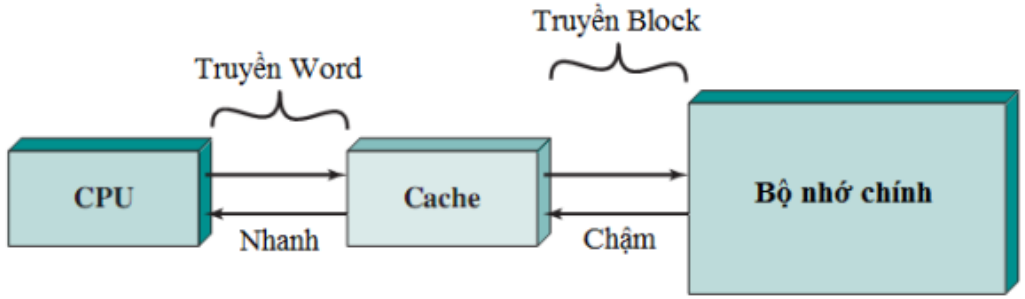
4.4 Tổ chức cache của Pentium 4

4.5 Tổ chức cache trong ARM

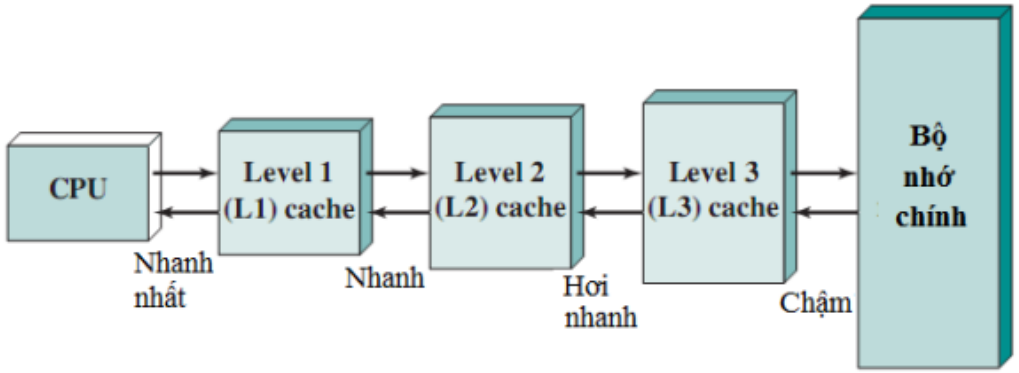
4.2. Nguyên lý bộ nhớ cache

Bộ nhớ cache và bộ nhớ chính

- Bộ xử lý truy xuất lệnh/dữ liệu từ bộ nhớ chính với tốc độ chậm (do tốc độ bộ nhớ chính, tốc độ bus chậm hơn bộ xử lý)
- Bộ nhớ đệm (cache) được thiết kế để cải thiện thời gian truy cập bộ nhớ:
  - Dựa vào tính cục bộ của dữ liệu và lệnh lưu trữ trong bộ nhớ chính
  - Bộ nhớ cache có tốc độ cao nhưng dung lượng thấp hơn bộ nhớ chính
  - Bộ nhớ cache chứa **bản sao** của một phần dữ liệu của bộ nhớ chính.



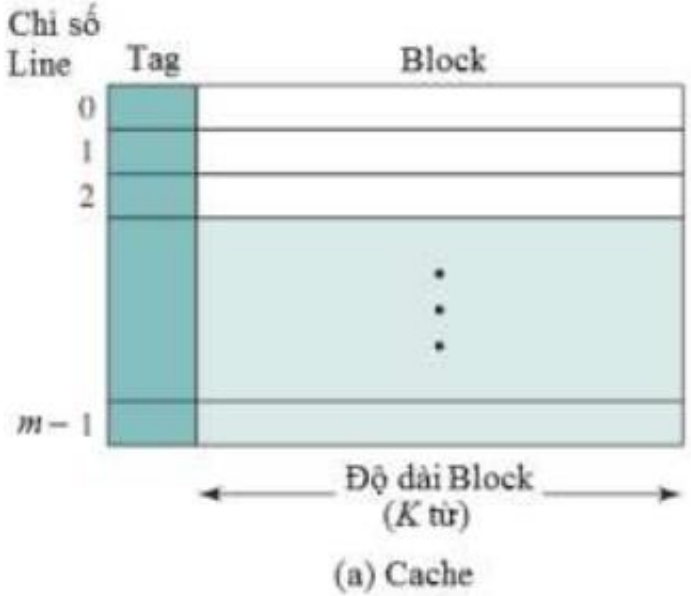
(a) Cache đơn



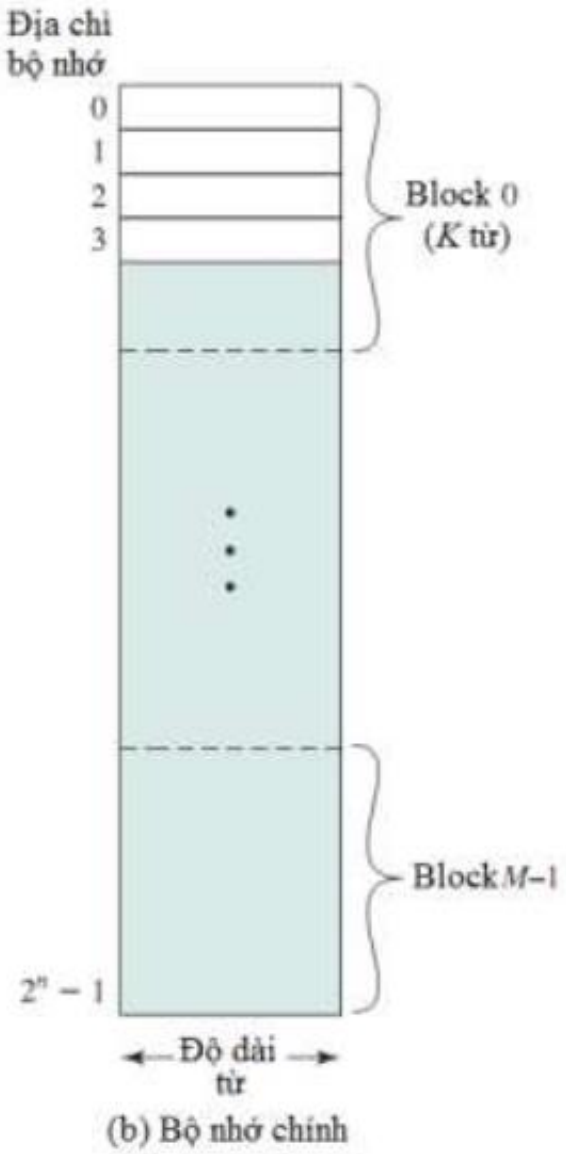
(b) Tổ chức cache ba level

Hình 4.2 Cache và bộ nhớ chính

Cấu trúc bộ nhớ chính/cache



(a) Cache



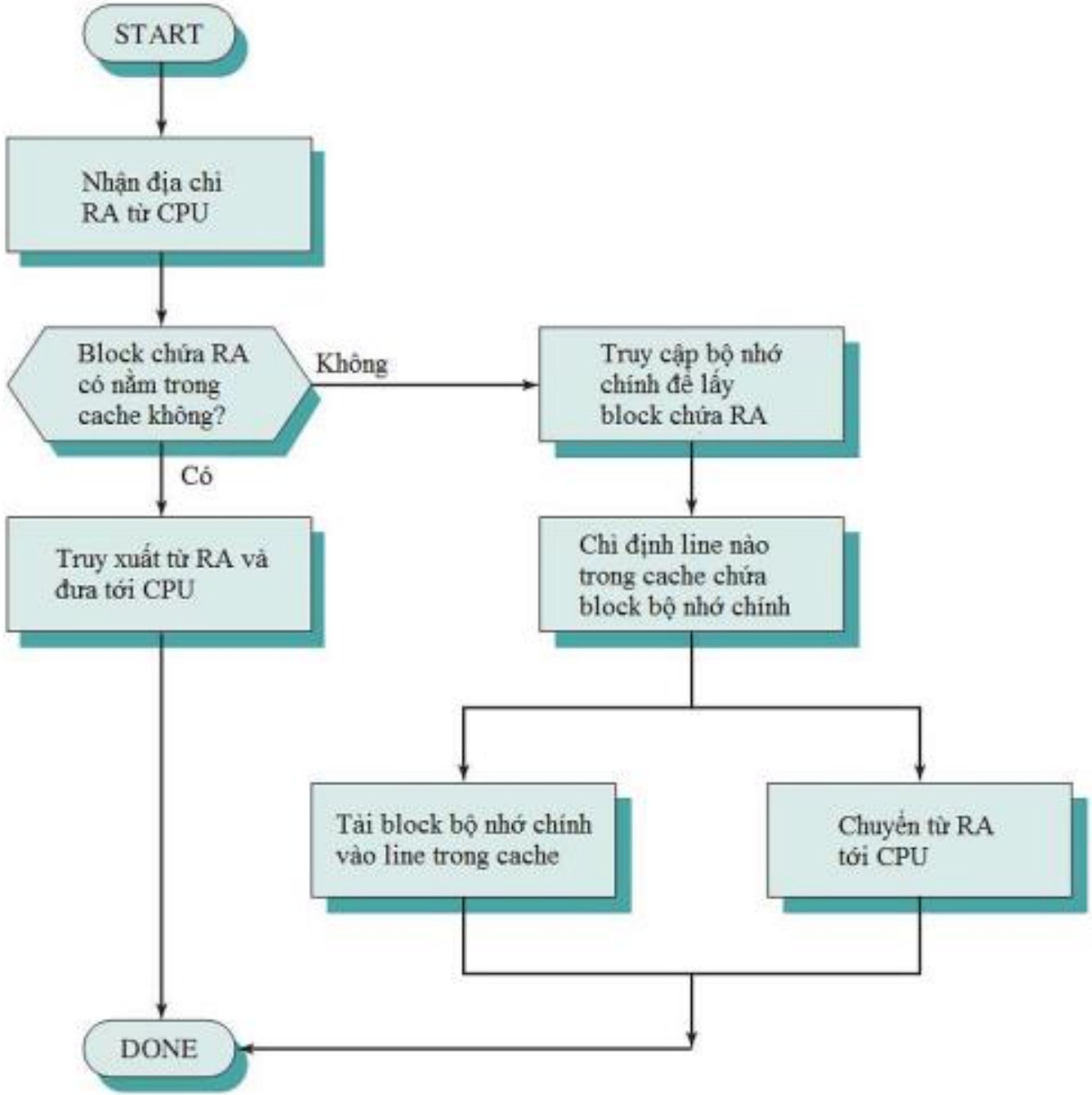
(b) Bộ nhớ chính

- Bộ nhớ chính gồm 2^n ***từ nhớ (word)*** được đánh địa chỉ (*n* bit địa chỉ)
- Bộ nhớ chính được chia thành các ***khối (block)*** có kích thước cố định: ***K word***.
  - Bộ nhớ chính có $M = \frac{2^n}{K}$ khối
- Bộ nhớ cache được chia thành các ***đường (line)***, mỗi đường có ***K word.***
- Mỗi **block** của bộ chính được ánh xạ vào một **line** của Cache

Nguyên lý hoạt động của Cache

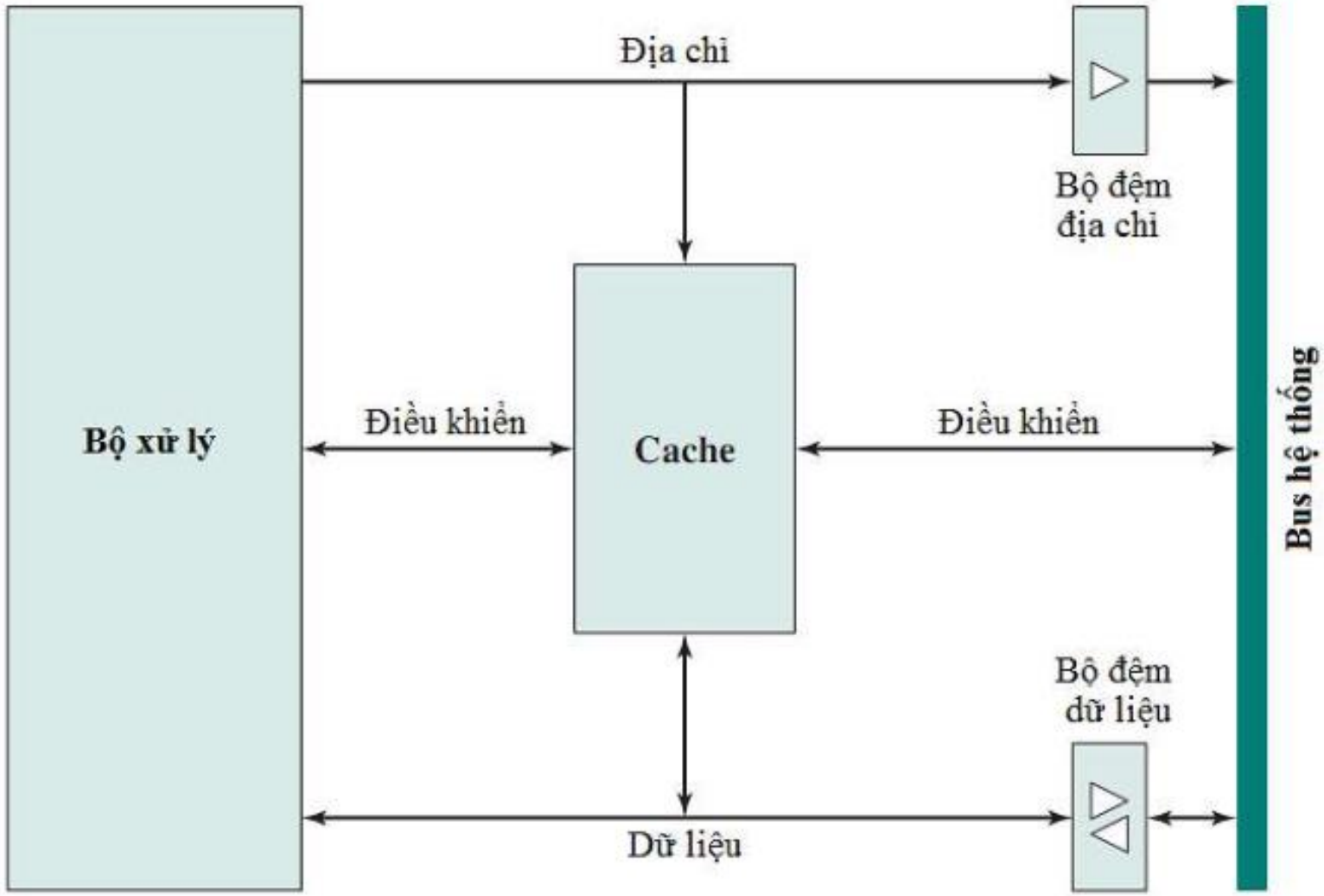
- Khi bộ xử lý muốn đọc một word của bộ nhớ nó sẽ kiểm tra xem word đó có nằm trong bộ nhớ cache hay không.
  - Nếu có: word này được gửi đến bộ vi xử lý.
  - Nếu không: một khối dữ liệu từ bộ nhớ chính (chứa từ mà bộ xử lý đang muốn truy cập), được đọc vào bộ nhớ cache và sau đó word cần tìm được gửi đến bộ xử lý.

Thao tác đọc Cache



(RA – read address): địa chỉ đọc của từ cần đọc

Tổ chức bộ nhớ cache điển hình



Nội dung chính

4.1 Tổng quan về bộ nhớ máy tính

4.2 Nguyên lý của bộ nhớ cache

4.3 Các yếu tố trong thiết kế bộ nhớ cache

4.4 Tổ chức cache của Pentium 4

4.5 Tổ chức cache trong ARM

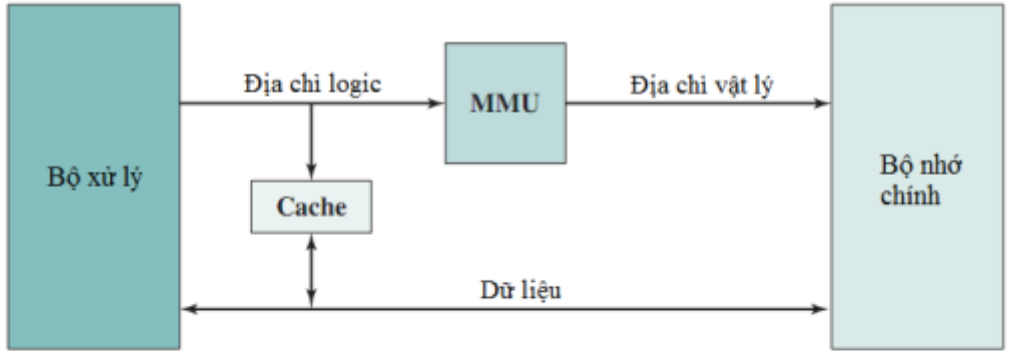
4.3. Các yếu tố trong thiết kế Cache

a. **Địa chỉ bộ nhớ cache** Logic Vật lý b. **Ánh xạ bộ nhớ** Trực tiếp Kết hợp Kết hợp tập hợp c. **Thuật toán thay thế** Least recently used (LRU) First in first out (FIFO) Least frequently used (LFU) Random	d. **Chính sách ghi** Ghi xuôi Ghi ngược e. **Cache nhiều cấp** Một hoặc hai cấp Thống nhất hoặc phân chia g. **Kích thước bộ nhớ cache** h. **Kích thước line**

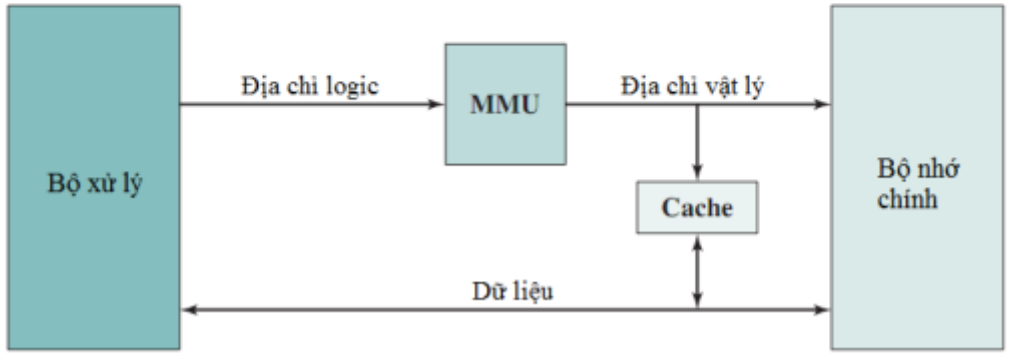
a. Địa chỉ bộ nhớ cache

- Bộ xử lý hỗ trợ bộ nhớ ảo:
  - ✓Quản lý bộ nhớ thông qua địa chỉ logic
  - ✓Các trường địa chỉ trong lệnh là các địa chỉ ảo
  - ✓Để thực hiện các thao tác đọc/ghi vào bộ nhớ chính, khối quản lý bộ nhớ (MMU – Memory Management Unit) sẽ dịch từng địa chỉ ảo sang địa chỉ vật lý trong bộ nhớ chính

- Cache logic (Logical Cache):
  - ✓ Bộ nhớ cache đặt giữa bộ xử lý và MMU
  - ✓ Địa chỉ được sử dụng là địa chỉ logic
- Cache vật lý (Physical Cache):
  - ✓ Bộ nhớ cache đặt giữa MMU và bộ nhớ chính
  - ✓ Địa chỉ được sử dụng là địa chỉ vật lý



(a) Cache logic



(b) Cache vật lý

b. Ánh xạ bộ nhớ

- Bởi vì số đường (line) của cache ít hơn số khối (block) của bộ nhớ chính, cần có một thuật toán ánh xạ các khối bộ nhớ chính vào các đường của bộ nhớ cache
- Ba kỹ thuật có thể được sử dụng:

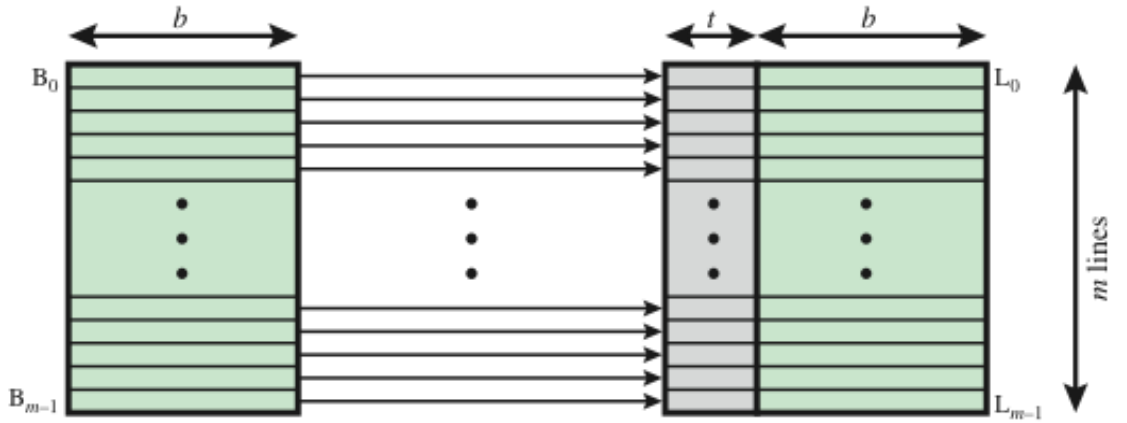
Trực tiếp	Kết hợp	Tập kết hợp
• Mỗi khối của bộ nhớ chính được ánh xạ vào một line duy nhất • Đơn giản nhất	• Cho phép một khối của bộ nhớ chính được nạp vào line bất kỳ • Logic điều khiển cache diễn giải địa chỉ bộ nhớ bằng một trường Tag và trường Word • Để xác định một khối có ở trong một cache không, logic điều khiển cache phải cùng lúc kiểm tra Tag của tất cả các line	• Kết hợp hai phương pháp trên • Thể hiện ưu điểm của cả phương pháp trực tiếp và kết hợp, đồng thời giảm nhược điểm

Ánh xạ trực tiếp (Direct mapping)

- Mỗi **khối (block)** của bộ nhớ chính được ánh xạ vào một **đường (line)** nhất định của bộ nhớ cache.
- Block thứ j trong bộ nhớ chính sẽ được ánh xạ vào line thứ $j \bmod m$ (phép chia lấy dư), m là số line

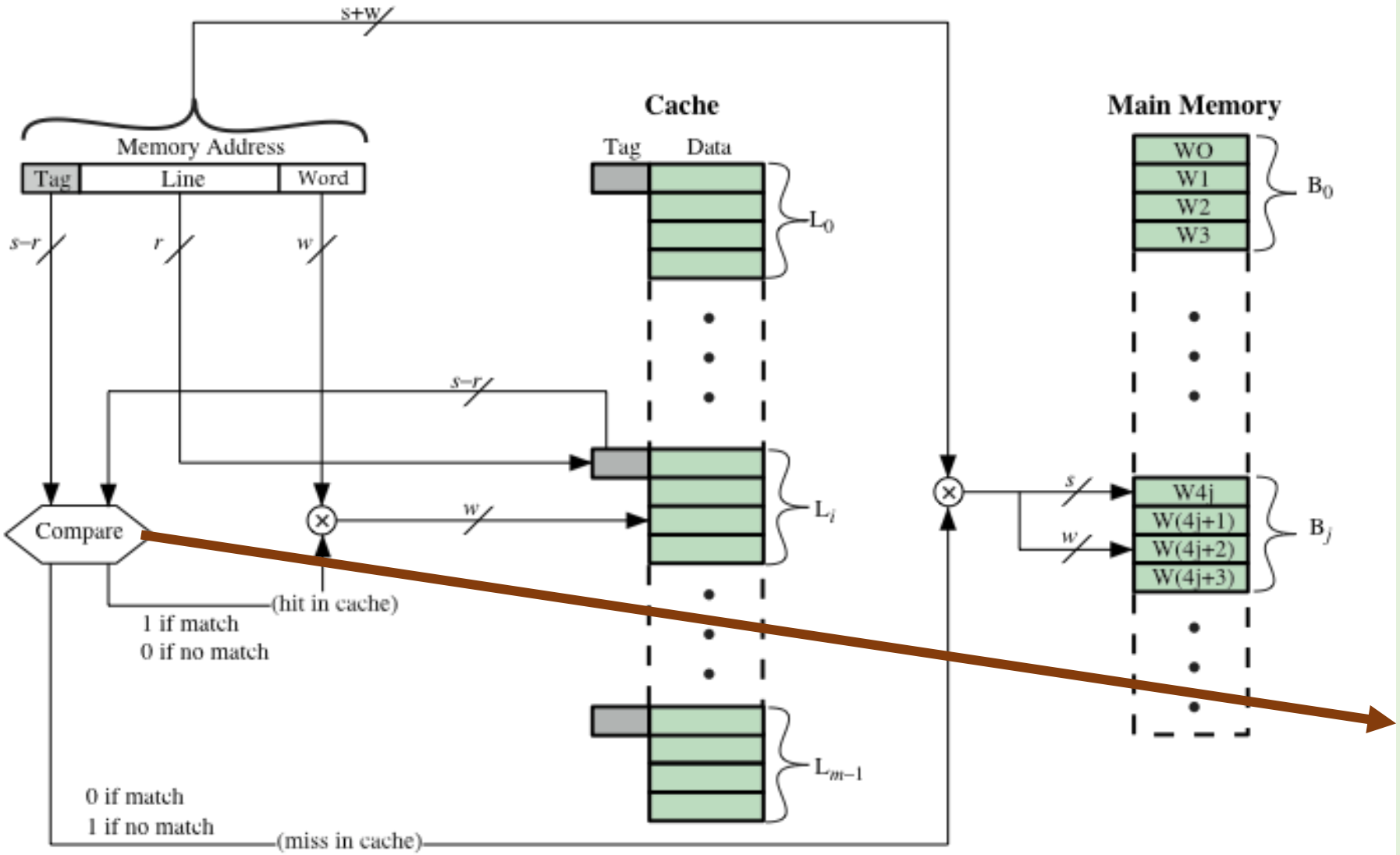
$$B_j \text{ ánh xạ vào } L_{j \bmod m}$$

- Do vậy, nhiều block sẽ được ánh xạ vào một line. Để xác định block nào đang được ánh xạ vào cache: sử dụng trường tag



(a) Direct mapping

Tổ chức cache ánh xạ trực tiếp



Khi truy xuất một word, **địa chỉ bộ nhớ** (s+w bit) được tách thành 3 trường:

- Word (w bit): xác định một word (hoặc một đơn vị đánh địa chỉ) trong block/line
- Line (r bit): xác định một line trong 2^r line của cache
- Tag (s-r bit): xác định block nào đang được ánh xạ vào line đó

So sánh **tag** của địa chỉ bộ nhớ với **tag** của line trong cache để xác định xem có phải block đó đang ở trong cache không

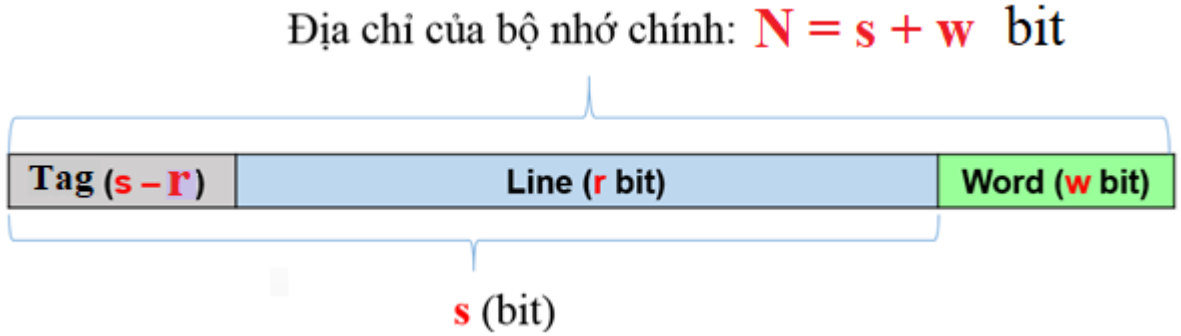
Nhận xét ánh xạ trực tiếp

- Ưu điểm:
  - Thực hiện đơn giản
- Nhược điểm:
  - Các khối lưu cố định tại một đường trong bộ nhớ cache. Do vậy, khi chương trình tham chiếu các từ lặp lại từ hai khối mà cùng ánh xạ đến một đường thì cache liên tục phải đổi dữ liệu từ memory vào. Điều này làm giảm hiệu suất (hiện tượng *thrashing)*

Tổng kết ánh xạ trực tiếp

B_j được ánh xạ vào $L_{j \bmod m}$

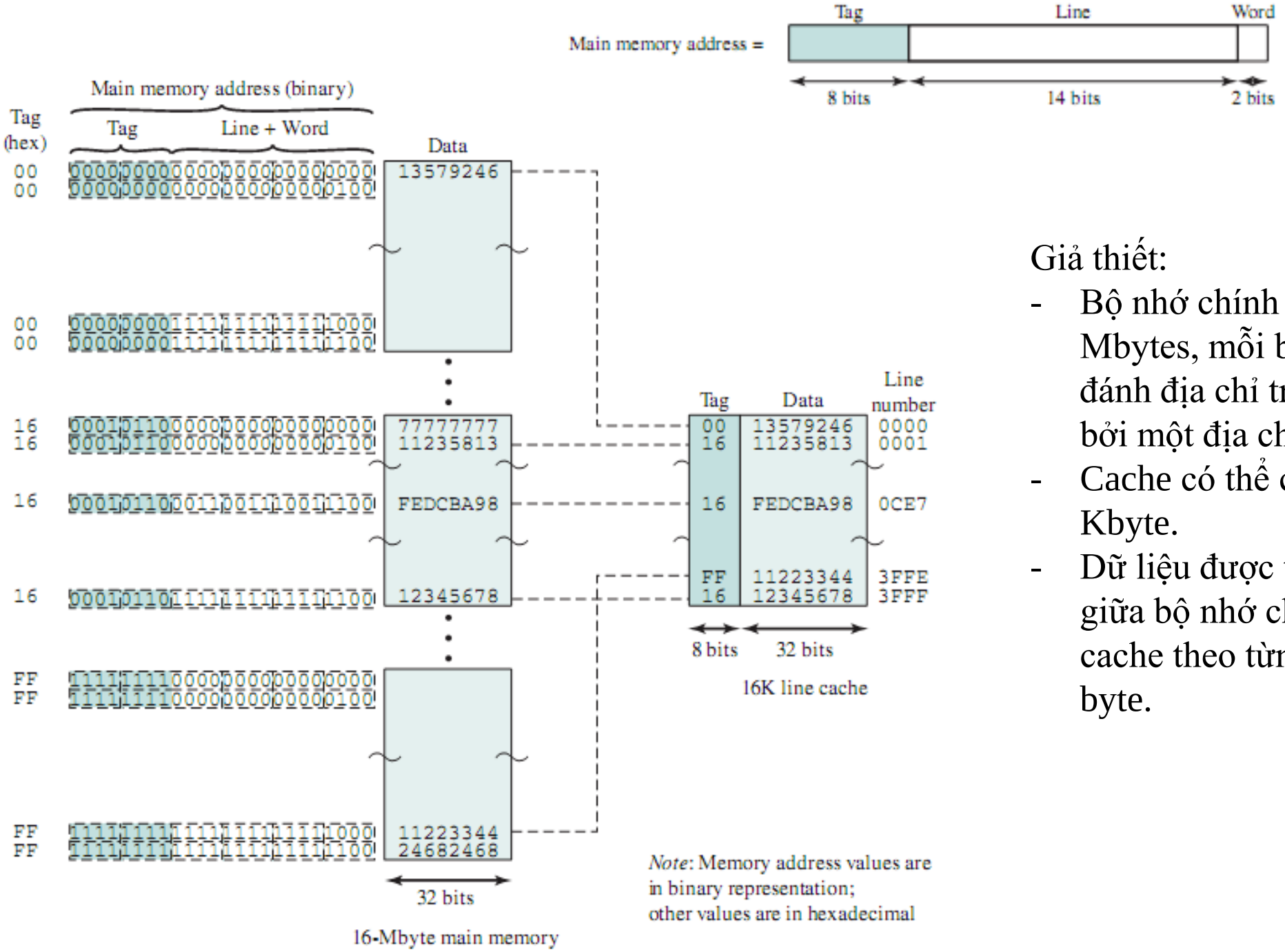
m là số đường (line) trong bộ nhớ Cache



- Độ dài địa chỉ bộ nhớ chính: $N = (s + w)$ bit
- Kích thước đơn vị đánh địa chỉ: $W_{đv}$ byte
- Số lượng đơn vị đánh địa chỉ của bộ nhớ chính = $2^N = 2^{s+w}$
- Dung lượng bộ nhớ chính = $2^N \cdot W_{đv}$ byte =$2^{s+w} \cdot W_{đv}$ byte

- Số lượng đơn vị đánh địa chỉ của một khối (hoặc một đường) = 2^w
- Kích thước khối=kích thước đường = $2^w \cdot W_{đv}$ byte
- Số khối trong bộ nhớ chính = $2^{s+w}/2^w = 2^s$
- Số đường trong bộ nhớ cache = $m = 2^r$
- Kích thước của tag = $(s - r)$ bit = $(N - r - w)$ bit

Ví dụ ánh xạ trực tiếp



Giả thiết:

- Bộ nhớ chính gồm 16 Mbytes, mỗi byte được đánh địa chỉ trực tiếp bởi một địa chỉ .
- Cache có thể chứa 64 Kbyte.
- Dữ liệu được truyền giữa bộ nhớ chính và cache theo từng block 4 byte.

Bài tập áp dụng về ánh xạ trực tiếp

Bộ nhớ chính: 2^{16} byte, kích thước khối 8 byte, ánh xạ trực tiếp vào cache 32 đường.

a. 16 bit địa chỉ bộ nhớ chính được chia thành các trường Tag, Line và Word. Xác định kích thước của các trường Tag, Line và Word.

b. Các địa chỉ sau sẽ được lưu ở đường nào của cache?

0001 0001 0001 1011 1101 0000 0001 1101

1100 0011 0011 0100 1010 1010 1010 1010

c. Giả sử byte có địa chỉ 0001 1010 0001 1010 được lưu ở cache, các byte nào của bộ nhớ chính cũng được lưu trên đường đó?

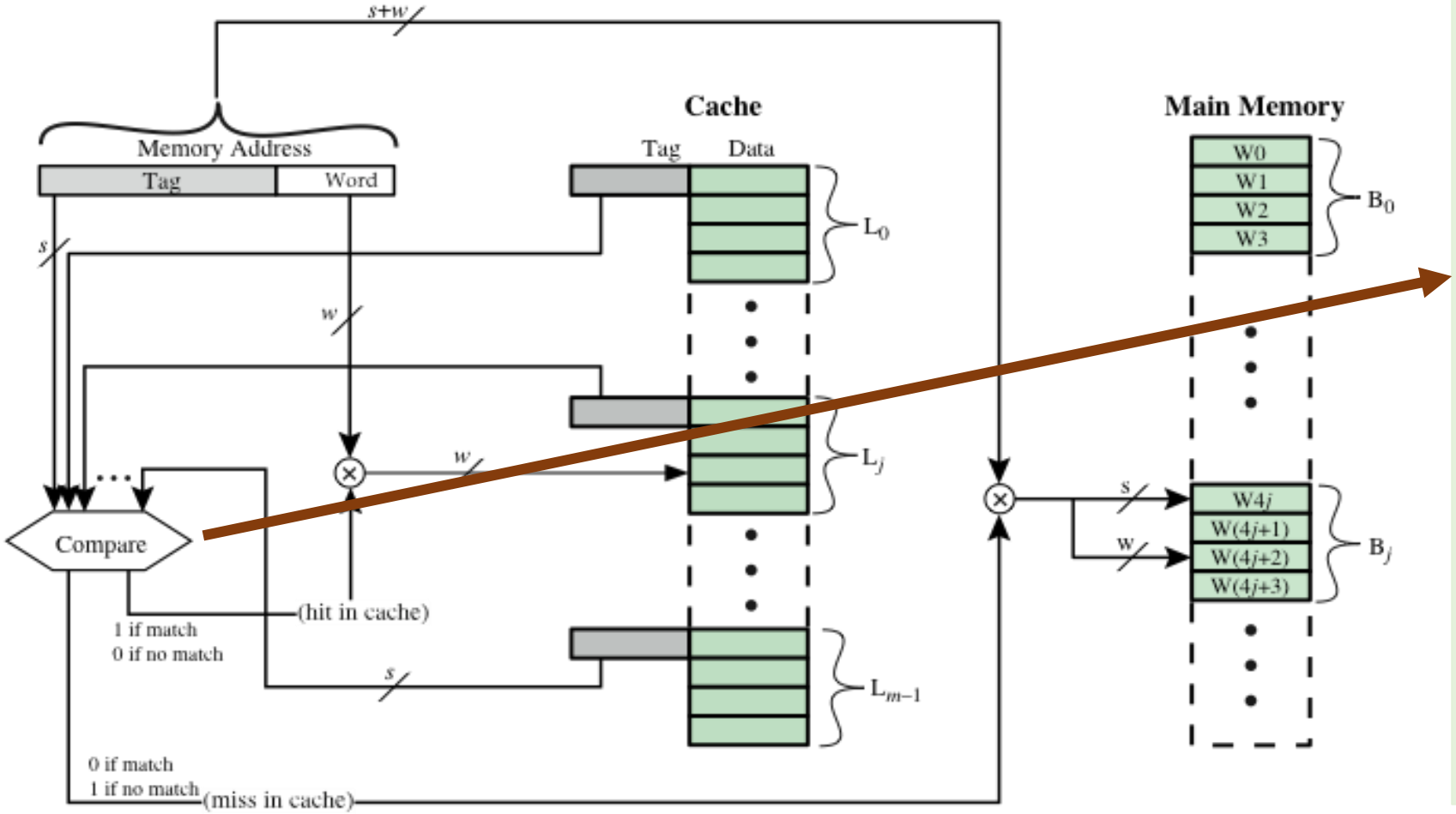
d. Có bao nhiêu byte có thể được lưu trên cache?

e. Khối nhớ thứ 36 được ánh xạ vào đường nào?

Ánh xạ kết hợp (Associative Mapping)

- Ánh xạ kết hợp khắc phục nhược điểm của ánh xạ trực tiếp bằng cách cho phép mỗi khối được nạp vào một đường bất kỳ của bộ nhớ cache
- *Bộ logic điều khiển bộ nhớ cache (cache control logic)* tách địa chỉ bộ nhớ thành hai trường: Tag và Word. Trường Tag hiển thị duy nhất một khối bộ nhớ chính.
- Để xác định liệu một khối có trong bộ nhớ cache, *bộ logic điều khiển bộ nhớ cache* phải cùng lúc kiểm tra mỗi Tag của các đường để so sánh

Tổ chức ánh xạ kết hợp



Khi truy xuất một word, logic cache tách **địa chỉ bộ nhớ** thành 2 trường:

- Word (w bit): xác định một word (hoặc một đơn vị đánh địa chỉ) trong block/line
- Tag (s bit): xác định block nào đang được ánh xạ vào line đó

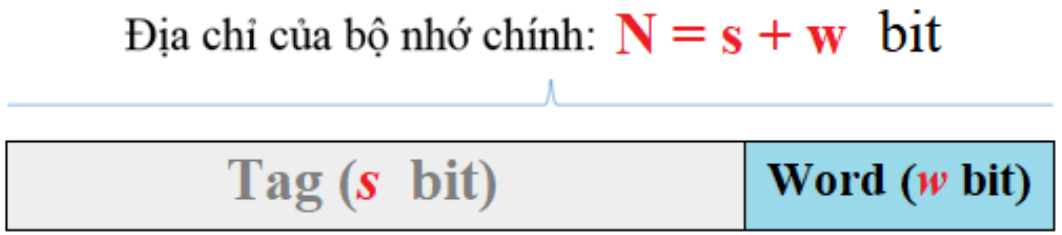
So sánh **Tag** của địa chỉ bộ nhớ với **Tag** của các line trong cache để xác định xem block đó có đang được ánh xạ vào cache không.

Nhận xét ánh xạ kết hợp

- **Ưu điểm**:
  - Linh hoạt khi thay thế một khối và đọc một khối mới vào cache. Các thuật toán thay thế được xây dựng để tối ưu hóa tỷ lệ truy cập.
- **Nhược điểm**:
  - Mạch phức tạp để thực hiện việc kiểm tra tất cả trường Tag của các đường trong cache một cách song song

Tổng kết ánh xạ kết hợp

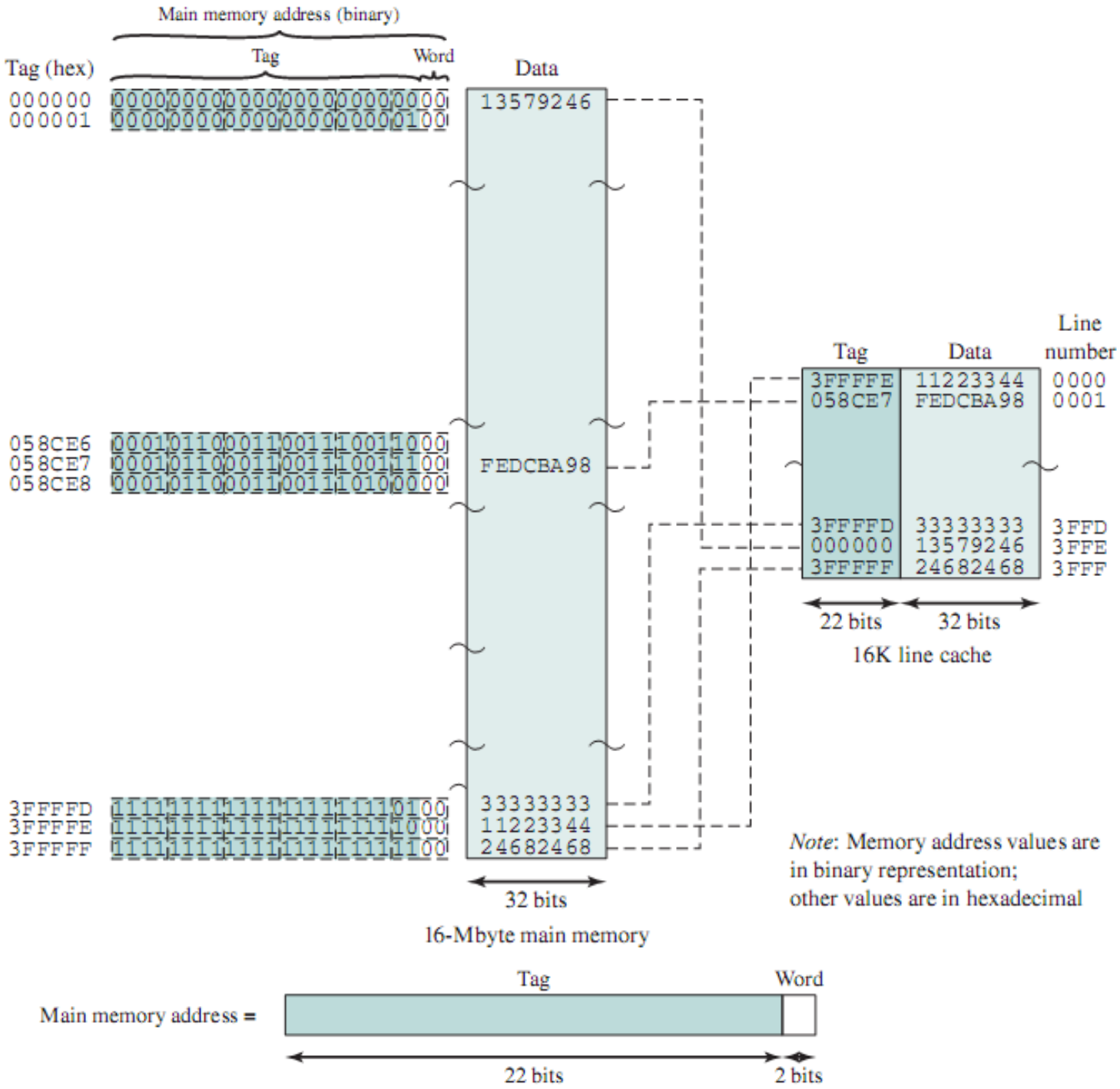
B_j được ánh xạ vào đường bất kỳ



- Độ dài địa chỉ bộ nhớ chính: $N = (s + w)$ bit
- Kích thước đơn vị đánh địa chỉ: $W_{đv}$ byte
- Số lượng đơn vị đánh địa chỉ của bộ nhớ chính = $2^N = 2^{s+w}$
- Dung lượng bộ nhớ chính = $2^N \cdot W_{đv}$ byte =$2^{s+w} \cdot W_{đv}$ byte

- Số lượng đơn vị đánh địa chỉ của một khối (hoặc một đường) = 2^w
- Kích thước khối=kích thước đường = $2^w \cdot W_{đv}$ byte
- Số khối trong bộ nhớ chính = $2^{s+w}/2^w = 2^s$
- Kích thước của tag = s bit = $(N - w)$ bit

Ví dụ ánh xạ kết hợp



Giả thiết:

- Bộ nhớ chính gồm 16 Mbytes, mỗi byte được đánh địa chỉ trực tiếp bởi một địa chỉ .
- Cache có thể chứa 64 Kbyte.
- Dữ liệu được truyền giữa bộ nhớ chính và cache theo từng block 4 byte.

Ánh xạ tập kết hợp (Set Associative Mapping)

- Tận dụng các ưu điểm của cả phương pháp trên đồng thời giảm nhược điểm của chúng
- Chia cache thành một số **Tập (set)**
  - Mỗi Tập chứa một số đường
- Một khối sẽ được ánh xạ vào một đường bất kỳ trong một Tập nhất định
  - Ví dụ:
    - 1 Tập có 2 đường
    - Một khối có thể ánh xạ vào 1 trong 2 đường bất kỳ của một Tập

Tổ chức cache *k*-Way Set Associative

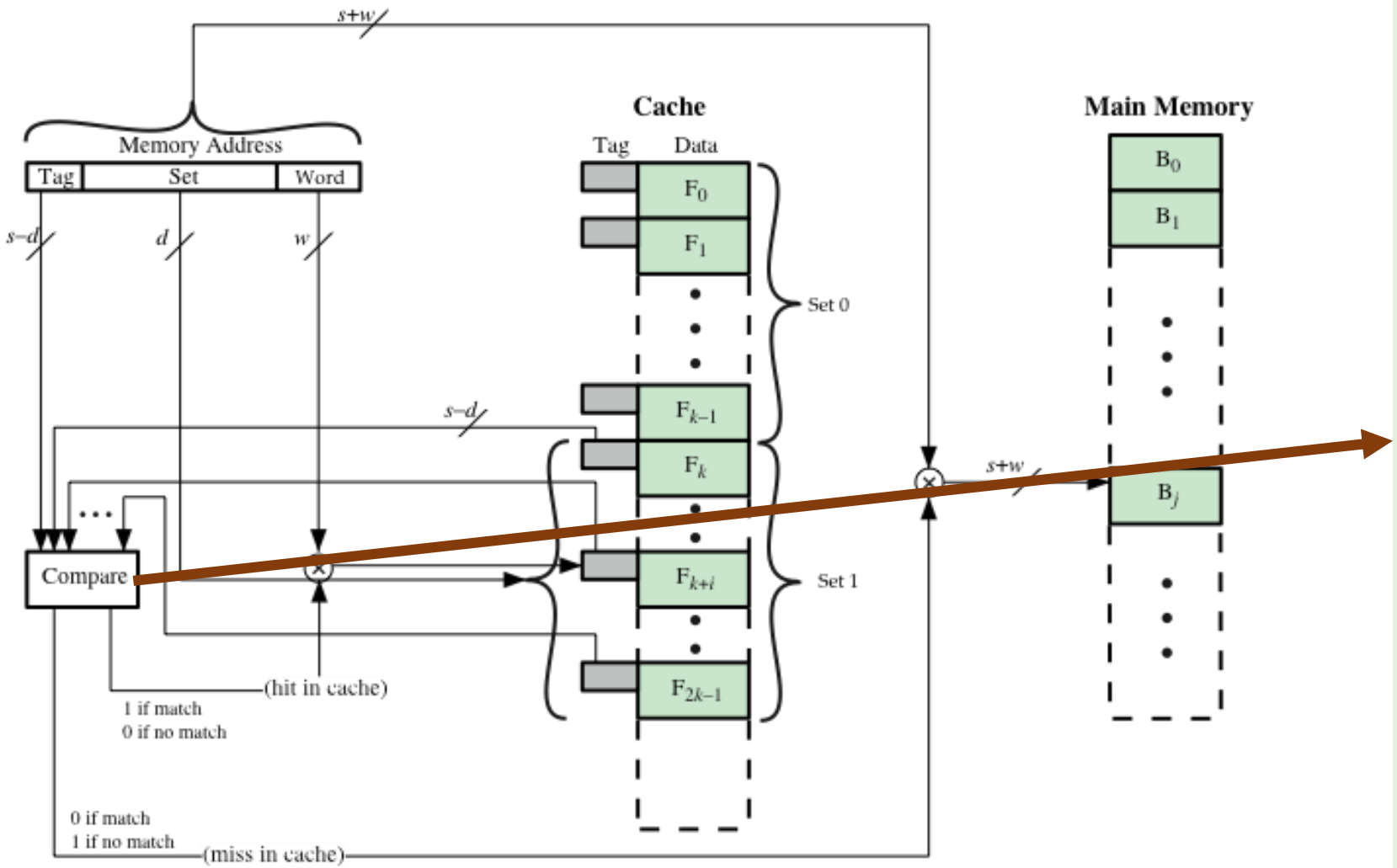


Figure 4.14 *k*-Way Set Associative Cache Organization

Khi truy xuất một word, logic cache tách **địa chỉ bộ nhớ** thành 3 trường:

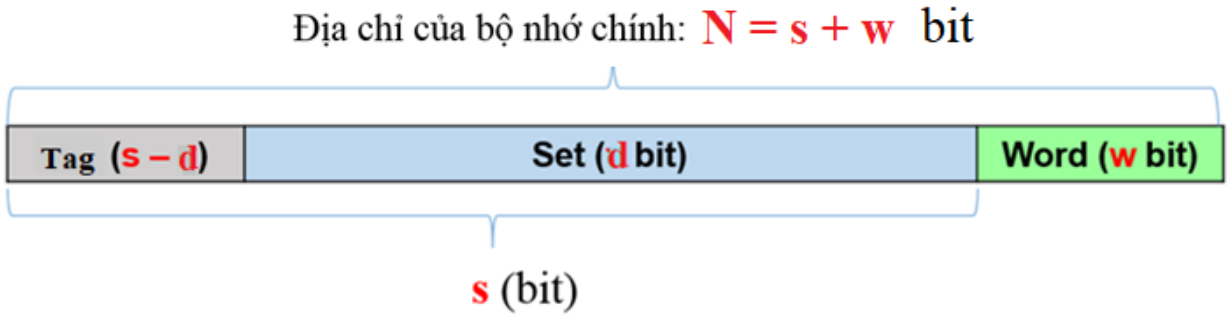
- Word (w bit): xác định một word (hoặc một đơn vị đánh địa chỉ) trong block
- Set (d bit): xác định block đó được ánh xạ vào set thứ tự bao nhiêu trong cache
- Tag (s-d bit): xác định block nào đang được ánh xạ vào line nào trong Set đó

So sánh **tag** của địa chỉ bộ nhớ với **tag** của các line trong Set đó để xác định xem có phải block đó đang được ánh xạ vào cache không

Tổng kết ánh xạ tập kết hợp

B_j được ánh xạ vào đường bất kỳ của tập $S_{j \bmod v}$

v = *số lượng Tập có trong cache*



- Độ dài địa chỉ bộ nhớ chính: $N = (s + w)$ bit
- Kích thước đơn vị đánh địa chỉ: $W_{đv}$ byte
- Số lượng đơn vị đánh địa chỉ của bộ nhớ chính = 2^N = 2^{s+w}
- Dung lượng bộ nhớ chính = $2^N \cdot W_{đv}$ byte =$2^{s+w} \cdot W_{đv}$ byte
- Số lượng đơn vị đánh địa chỉ của một khối (hoặc một đường) = 2^w
- Kích thước khối = kích thước đường = $2^w \cdot W_{đv}$ byte
- Số khối trong bộ nhớ chính = $2^{s+w}/2^w = 2^s$

- Số đường trong Cache = m
- Số đường trong mỗi Tập = k
- Số tập trong bộ nhớ cache = $v = m/k = 2^d$
- Kích thước của tag = $(s - d)$ bit = $(N - d - w)$ bit

Ví dụ ánh tập kết hợp

Giả thiết:

- Bộ nhớ chính gồm 16 Mbytes, mỗi byte được đánh địa chỉ trực tiếp bởi một địa chỉ .
- Cache có thể chứa 64 Kbyte.
- Dữ liệu được truyền giữa bộ nhớ chính và cache theo từng block 4 byte.
- 2 đường trong một tập (tập kết hợp 2-way)

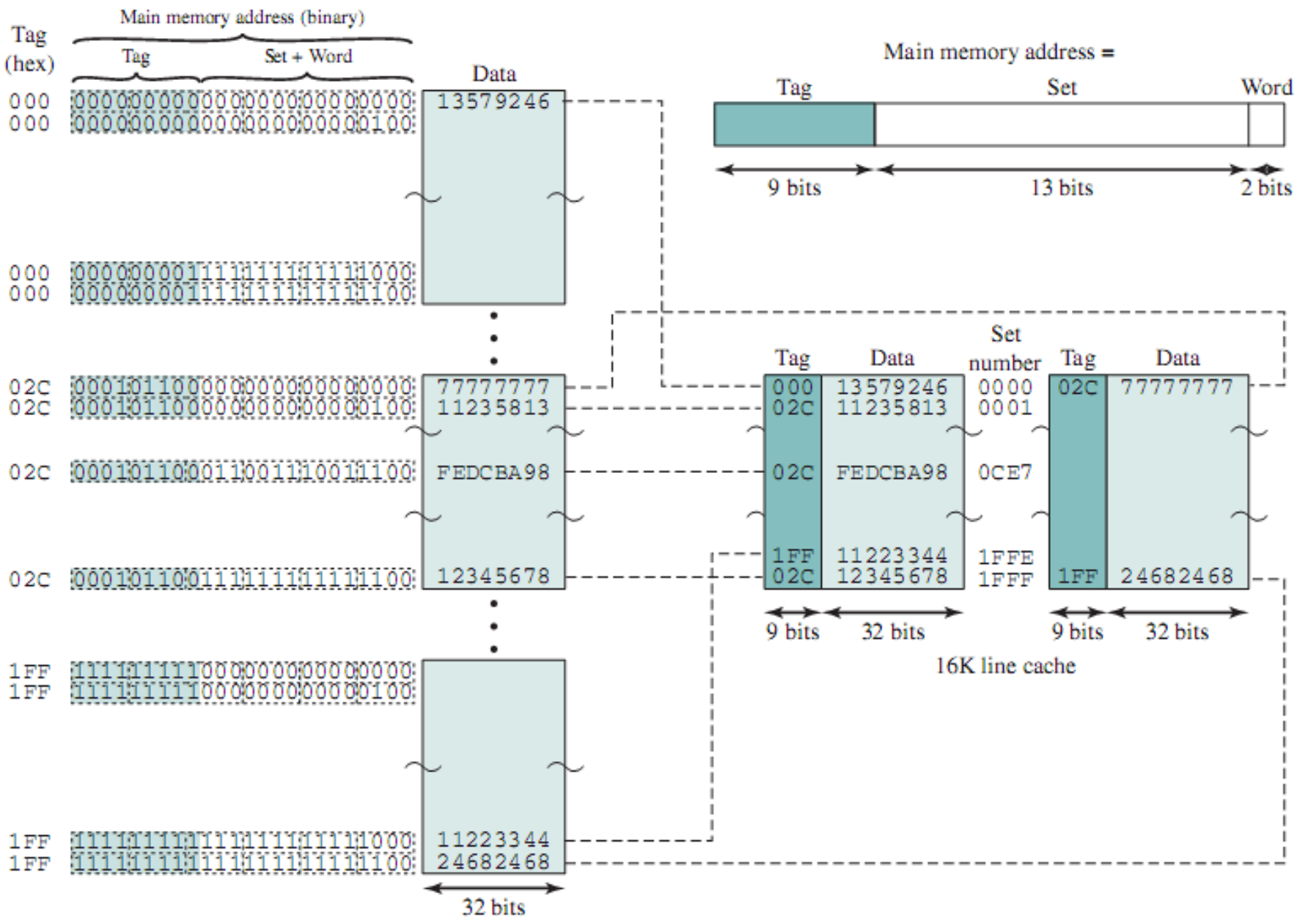


Figure 4.15 Two-Way Set-Associative Mapping Example

Bài tập về ánh xạ tập kết hợp

Bài 1

Xét vi xử lý 32 bit, địa chỉ 32b. Bộ nhớ cache 16-Kbyte, ánh xạ tập kết hợp 4-way, bộ nhớ đánh địa chỉ theo từ. Giả sử một đường gồm 4 từ 32-bit.

a. Xác định các trường của địa chỉ được sử dụng để ánh xạ cache.

b. Từ nhớ có địa chỉ $ABCDE8F8_H$ được ánh xạ vào vị trí nào trong cache.

Bài 2

Bộ nhớ Cache có 64 đường, sử dụng ánh xạ tập kết hợp 4 đường. Bộ nhớ chính có 4K khối, mỗi khối có kích thước 128 từ, bộ nhớ đánh địa chỉ theo từ.

Xác định định dạng địa chỉ bộ nhớ và tính kích thước các trường tương ứng.

So sánh hiệu suất của các PP ánh xạ

- Cache hit: số lần truy cập cache thành công
- Cache miss: số lần truy cập cache không thành công
- Hit ratio: tỷ lệ truy cập thành công

- Hiệu suất cache được đánh giá dựa trên

$$\boldsymbol{hit\ ratio = \frac{cache\ hit}{tổng\ số\ lần\ truy\ \ cập}}$$

c. Thuật toán thay thế

- Khi bộ nhớ cache đã đầy, nếu một khối mới được đưa vào cache, một trong những khối hiện có phải được thay thế
  - Đối với ánh xạ trực tiếp: một khối bất kỳ chỉ có thể ánh xạ vào 1 đường cụ thể. Nên khi cần đưa 1 khối mới vào cache buộc phải xóa dữ liệu cũ trên đường tương ứng đi.
  - Đối với ánh xạ kết hợp và tập kết hợp, một khối có thể được ánh xạ vào một trong một số đường. Vậy khi đưa 1 khối mới vào cache, ta cần xác định xem khối đó sẽ được ánh xạ vào đường nào: **thuật toán thay thế**
- Để đạt được tốc độ cao, thuật toán phải được thực hiện trong phần cứng

4 thuật toán thay thế phổ biến nhất

- Least recently used (LRU)
  - Hiệu quả nhất
  - Thay thế khối nằm trong cache lâu nhất mà không có tham chiếu đến nó
  - Do triển khai đơn giản, LRU là thuật toán thay thế phổ biến nhất
- First-in-first-out (FIFO)
  - Thay thế khối đã nằm trong cache lâu nhất
  - Dễ dàng thực hiện như một kỹ thuật vòng đệm hoặc round-robin
- Least frequently used (LFU)
  - Thay thế khối có ít tham chiếu đến nó nhất
  - Ở mỗi line thêm vào một bộ đếm, mỗi khi có tham chiếu đến line nào, bộ đếm của line đó tăng thêm 1 đơn vị
- Ngẫu nhiên
  - Có thể thay thế bất cứ khối nào
  - Các nghiên cứu đã chỉ ra: thay thế ngẫu nhiên chỉ làm giảm hiệu suất của hệ thống đi một chút so với các thuật toán thay thế ở trên

d. Chính sách ghi

- **Ghi xuyên qua (Write through):** khi VXL ghi dữ liệu ra bộ nhớ, dữ liệu sẽ được ghi đồng thời ra cả cache và BNC
  - Kỹ thuật đơn giản nhất
  - Tất cả các thao tác ghi được thực hiện cho bộ nhớ chính cũng như cache
  - Nhược điểm: tạo ra lưu lượng bộ nhớ đáng kể và có thể tạo ra nút cổ chai
- **Ghi trả sau (Write back):** khi VXL ghi dữ liệu, dữ liệu chỉ được ghi trên cache, sử dụng 1 bit (gọi là dirty bit) thiết lập giá trị để đánh dấu là dữ liệu đã bị thay đổi. Việc cập nhật dữ liệu lên BNC chỉ xảy ra khi thay thế khối mới vào cache
  - Giảm thao tác ghi bộ nhớ
  - Dữ liệu trên BNC không có hiệu lực. Cơ chế DMA bắt buộc phải thực hiệu qua cache
  - Nhược điểm: mạch phức tạp và khả năng có nút cổ chai

e. Cache nhiều cấp

- Với các hệ thống ngày nay, người ta thường sử dụng nhiều cache trong kiến trúc
- *Cache trên chip (cache on chip)* làm giảm hoạt động bus ngoài của bộ xử lý; tăng tốc thời gian xử lý và tăng hiệu năng toàn hệ thống
  - Khi lệnh hoặc dữ liệu được tìm thấy trong cache → không cần truy cập bus
  - Truy cập bộ nhớ *cache trên chip* nhanh hơn đáng kể
  - Trong giai đoạn này, bus tự do hỗ trợ các lượt truyền khác
- *Cache 2 cấp (cache 2 level)* : sử dụng một *cache on chip (cache L1)* và một cache bên ngoài (*cache L2)*
  - Cải thiện hiệu suất
  - Việc sử dụng cache nhiều cấp làm cho các vấn đề thiết kế liên quan đến cache phức tạp hơn, gồm kích thước, thuật toán thay thế, chính sách ghi

Cache thống nhất/Cache phân chia

- Cache thống nhất: lệnh và dữ liệu được lưu trên cùng một cache
  - Ưu điểm:
    - Tự động cân bằng giữa việc nạp dữ liệu và lệnh => tỷ lệ truy cập thành công cao hơn cache phân chia
    - Chỉ cần thiết kế và thực hiện một bộ nhớ cache
- Cache phân chia: lệnh và dữ liệu được lưu trên hai cache khác nhau, thường là cache L1
  - Ưu điểm: Loại bỏ sự cạnh tranh cache giữa khối tìm nạp/giải mã lệnh và khối thực hiện
- Xu hướng: cache phân chia ở L1 và cache thống nhất ở các level cao hơn

g. Kích thước cache (cache size)

- Kích thước của bộ nhớ đệm đủ nhỏ để chi phí trung bình cho mỗi bit gần bằng bộ nhớ chính và đủ lớn để tổng thể thời gian truy cập trung bình gần bằng thời gian truy cập của riêng bộ đệm.
- Bộ đệm càng lớn thì số lượng cổng liên quan đến việc đánh địa chỉ bộ đệm càng lớn. Kết quả là bộ đệm lớn có xu hướng chậm hơn một chút so với những bộ đệm có kích thước nhỏ hơn
- Do hiệu suất của bộ nhớ đệm rất nhạy cảm với bản chất của khối lượng công việc, nên không thể đạt được một kích thước bộ đệm “tối ưu” duy nhất.

Processor	Type	Year of Introduction	L1 Cache[a]	L2 cache	L3 Cache
IBM 360/85	Mainframe	1968	16 to 32 kB	—	—
PDP-11/70	Minicomputer	1975	1 kB	—	—
VAX 11/780	Minicomputer	1978	16 kB	—	—
IBM 3033	Mainframe	1978	64 kB	—	—
IBM 3090	Mainframe	1985	128 to 256 kB	—	—
Intel 80486	PC	1989	8 kB	—	—
Pentium	PC	1993	8 kB/8 kB	256 to 512 KB	—
PowerPC 601	PC	1993	32 kB	—	—
PowerPC 620	PC	1996	32 kB/32 kB	—	—
PowerPC G4	PC/server	1999	32 kB/32 kB	256 KB to 1 MB	2 MB
IBM S/390 G6	Mainframe	1999	256 kB	8 MB	—
Pentium 4	PC/server	2000	8 kB/8 kB	256 KB	—
IBM SP	High-end server/ supercomputer	2000	64 kB/32 kB	8 MB	—
CRAY MTA[b]	Supercomputer	2000	8 kB	2 MB	—
Itanium	PC/server	2001	16 kB/16 kB	96 KB	4 MB
Itanium 2	PC/server	2002	32 kB	256 KB	6 MB
IBM POWER5	High-end server	2003	64 kB	1.9 MB	36 MB
CRAY XD-1	Supercomputer	2004	64 kB/64 kB	1MB	—
IBM POWER6	PC/server	2007	64 kB/64 kB	4 MB	32 MB
IBM z10	Mainframe	2008	64 kB/128 kB	3 MB	24-48 MB
Intel Core i7 EE 990	Workstaton/ server	2011	6 × 32 kB/32 kB	1.5 MB	12 MB
IBM zEnterprise 196	Mainframe/ Server	2011	24 × 64 kB/ 128 kB	24 × 1.5 MB	24 MB L3 192 MB L4

Kích thước cache trong một số bộ xử lý

a-Hai giá trị cách nhau bằng dấu / là cache chỉ thị và cache dữ liệu.

b-Cả hai cache đều là cache chỉ thị; Không có cache dữ liệu.

h. Kích thước Line

- Khi kích thước khối tăng, ban đầu tỷ lệ truy cập sẽ tăng do nguyên tắc cục bộ → dữ liệu hữu ích (dữ liệu có khả năng lớn được truy cập trong câu lệnh tiếp theo) được đưa vào cache
- Tuy nhiên, khi kích thước khối quá lớn, tỷ lệ này giảm đi do:

  1, Các khối lớn hơn làm giảm số lượng đường trong một cache

  2, Khi kích thước khối lớn, mỗi word thêm vào lại càng xa word yêu cầu → ít có khả năng truy xuất

Nội dung chính

4.1 Tổng quan về bộ nhớ máy tính

4.2 Nguyên lý của bộ nhớ cache

4.3 Các yếu tố trong thiết kế bộ nhớ cache

4.4 Tổ chức cache của Pentium 4

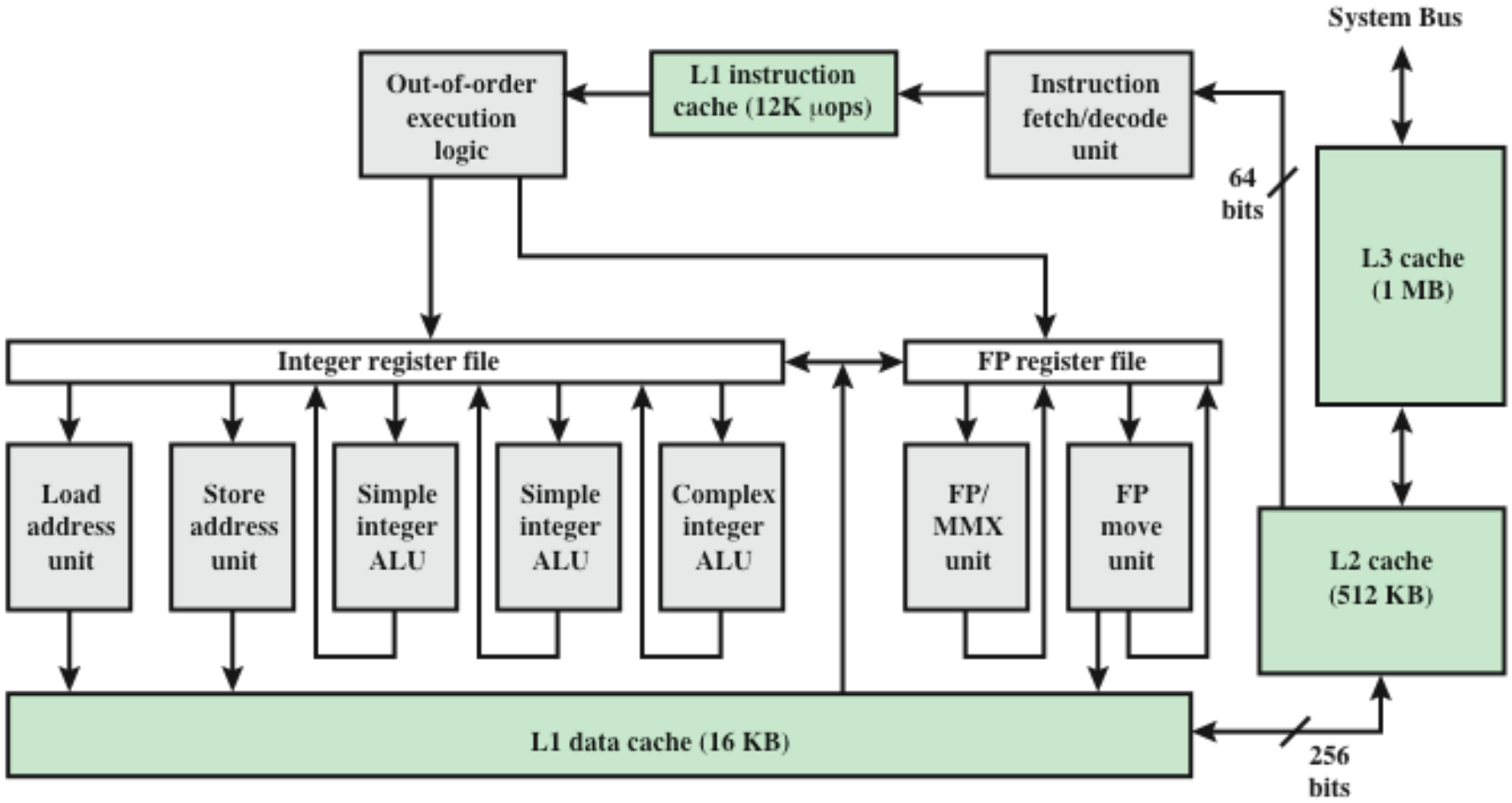
4.5 Tổ chức cache trong ARM

4.4.Tổ chức Cache Pentium 4

Problem	Solution	Processor on which Feature First Appears
External memory slower than the system bus.	Add external cache using faster memory technology.	386
Increased processor speed results in external bus becoming a bottleneck for cache access.	Move external cache on-chip, operating at the same speed as the processor.	486
Internal cache is rather small, due to limited space on chip	Add external L2 cache using faster technology than main memory	486
Contention occurs when both the Instruction Prefetcher and the Execution Unit simultaneously require access to the cache. In that case, the Prefetcher is stalled while the Execution Unit's data access takes place.	Create separate data and instruction caches.	Pentium
Increased processor speed results in external bus becoming a bottleneck for L2 cache access.	Create separate back-side bus that runs at higher speed than the main (front-side) external bus. The BSB is dedicated to the L2 cache.	Pentium Pro
	Move L2 cache on to the processor chip.	Pentium II
Some applications deal with massive databases and must have rapid access to large amounts of data. The on-chip caches are too small.	Add external L3 cache.	Pentium III
	Move L3 cache on-chip.	Pentium 4

Bảng 4.4 Intel Cache Evolution

Sơ đồ khối Pentium 4



Các chế độ hoạt động Cache Pentium 4

Control Bits	Operating Mode		
NW	Cache Fills	Write Throughs	Invalidates
0	Enabled	Enabled	Enabled
0	Disabled	Enabled	Enabled
1	Disabled	Disabled	Disabled

Note: CD = 0; NW = 1 là kết hợp không hợp lệ.

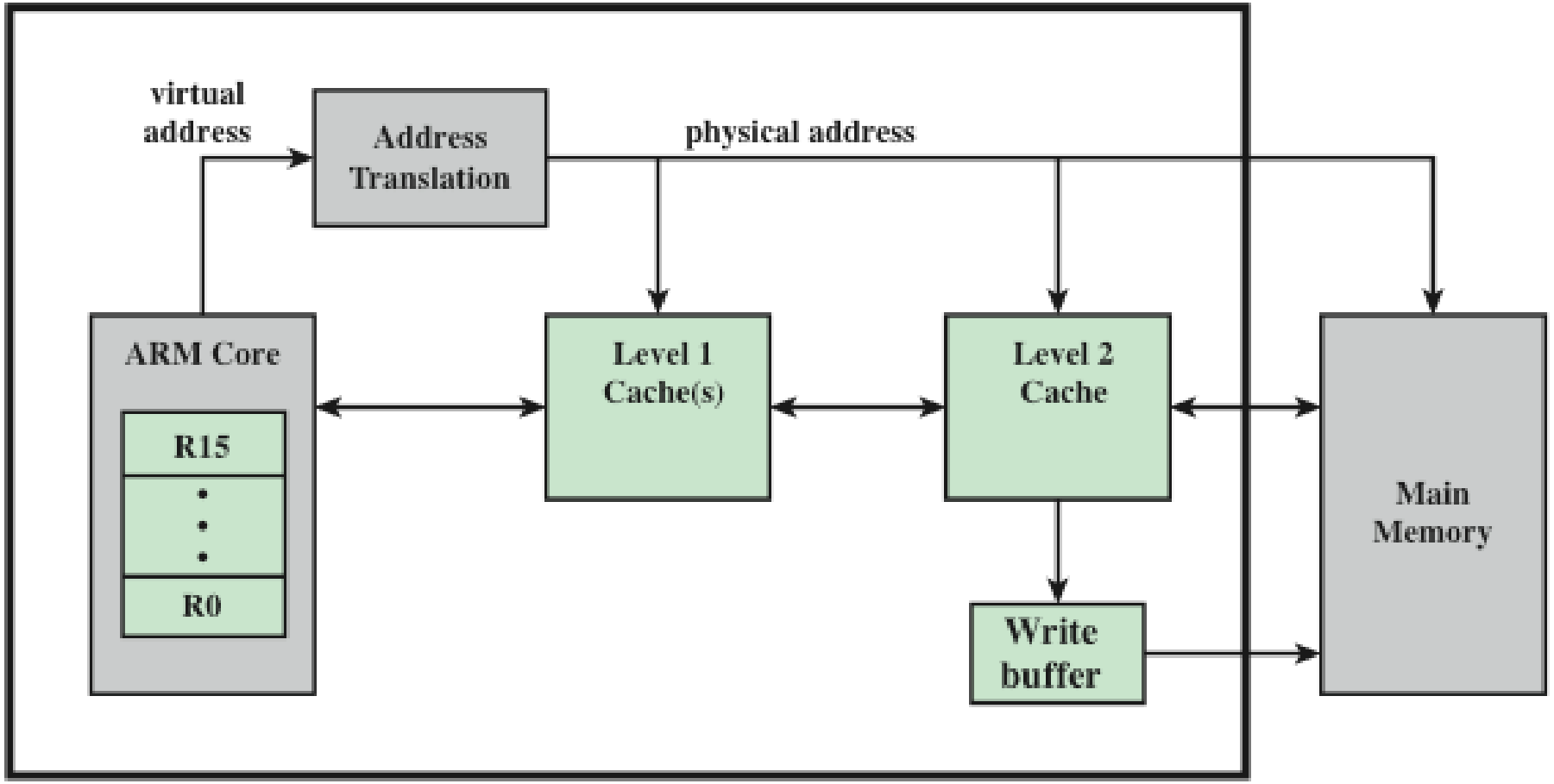
Nội dung chính

4.1 Tổng quan về bộ nhớ máy tính
4.2 Nguyên lý của bộ nhớ cache
4.3 Các yếu tố trong thiết kế bộ nhớ cache
4.4 Tổ chức cache của Pentium 4
4.5 Tổ chức cache trong ARM

4.5. Tổ chức cache ARM
Đặc tính Cache ARM

Core	Cache Type	Cache Size (kB)	Cache Line Size (words)	Associativity	Location	Write Buffer Size (words)
ARM720T	Unified	8	4	4-way	Logical	8
ARM920T	Split	16/16 D/I	8	64-way	Logical	16
ARM926EJ-S	Split	4-128/4-128 D/I	8	4-way	Logical	16
ARM1022E	Split	16/16 D/I	8	64-way	Logical	16
ARM1026EJ-S	Split	4-128/4-128 D/I	8	4-way	Logical	8
Intel StrongARM	Split	16/16 D/I	4	32-way	Logical	32
Intel Xscale	Split	32/32 D/I	8	32-way	Logical	32
ARM1136-JF-S	Split	4-64/4-64 D/I	8	4-way	Physical	32

ARM Cache và tổ chức bộ đệm ghi



Từ khóa

- Cache: bộ nhớ cache
- Cache hit: truy cập thành công vào bộ nhớ cache
- Cache miss: truy cập bộ nhớ cache không thành công
- Hit ratio: tỷ lệ truy cập thành công
- Cache mapping: ánh xạ bộ nhớ cache
- Write policy: chính sách ghi
- Cache L1, L2, L3: bộ nhớ cache mức 1, 2,3
- Line: đường trong cache
- Set: tập trong cache
- Block: khối trong bộ nhớ chính

Tổng kết

Chương 4: Bộ nhớ Cache

- Đặc điểm của hệ thống bộ nhớ
  - Vị trí
  - Dung lượng
  - Đơn vị truyền
- Bộ nhớ phân cấp
  - Dung lượng bao nhiêu là đủ?
  - Tốc độ ra sao?
  - Giá thành như thế nào?
- Nguyên lý bộ nhớ Cache
- Các yếu tố trong thiết kế cache
  - Địa chỉ bộ nhớ cache
  - Ánh xạ bộ nhớ
  - Thuật toán thay thế
  - Chính sách ghi
  - Cache nhiều cấp
  - Kích thước bộ nhớ cache
  - Kích thước line
- Tổ chức cache Pentium 4
- Tổ chức cache ARM

Câu hỏi ôn tập

1. Liệt kê và mô tả các đặc trưng của bộ nhớ máy tính.
2. Một số khái niệm đối với bộ nhớ trong: Từ (word), đơn vị đánh địa chỉ (Addressable units), đơn vị truyền (Unit of transfer)
3. Sự khác nhau giữa truy cập tuần tự, trực tiếp và ngẫu nhiên là gì?
4. Nêu mối quan hệ giữa thời gian truy cập, giá thành và dung lượng bộ nhớ.
5. Nguyên lý hoạt động của bộ nhớ đệm.
6. Tổ chức của bộ nhớ chính và bộ nhớ đệm.
7. Các yếu tố chính trong thiết kế cache là gì?
8. Khái niệm Cache logic (Logical Cache) và Cache vật lý (Physical Cache).
9. Trình bày các phương pháp ánh xạ trong Cache: ánh xạ trực tiếp, kết hợp và tập kết hợp.

Câu hỏi ôn tập

10. Đối với cache ánh xạ trực tiếp, một địa chỉ bộ nhớ chính gồm bao nhiêu trường? Là những trường gì?
11. Đối với cache ánh xạ kết hợp, một địa chỉ bộ nhớ chính gồm bao nhiêu trường? Là những trường gì?
12. Đối với cache ánh xạ tập kết hợp, một địa chỉ bộ nhớ chính gồm bao nhiêu trường? Là những trường gì?
13. Trình bày các thuật toán thay thế trong bộ nhớ Cache.
14. Ghi xuyên qua (Write though) và ghi trả sau (write back) khác nhau như thế nào? Kỹ thuật nào giảm số lần truy cập bus hệ thống nhiều hơn?
15. Kích thước của đường có ảnh hưởng như thế nào đến hiệu suất cache (Hit ratio)?
16. Khái niệm cache phân chia và cache thống nhất.

Hình ảnh và nội dung trong bài giảng này tham khảo từ cuốn sách và slide bài giảng “Computer Organization and Architecture”, 10th Edition, của tác giả William Stallings.