

## BÀI TẬP CHƯƠNG 6: HỆ THỐNG BỘ NHỚ PHÂN CẤP

### DẠNG 1: ÁNH XẠ LIÊN KẾT

**Bài tập 1 [5.2/484 - Computer Organization and Design]:** Bộ nhớ đệm (cache) đóng vai trò quan trọng trong việc cung cấp một hệ thống phân cấp bộ nhớ hiệu suất cao cho các bộ xử lý. Dưới đây là danh sách các tham chiếu địa chỉ bộ nhớ 32-bit, được biểu diễn dưới dạng địa chỉ từ:

3, 180, 43, 2, 191, 88, 190, 14, 181, 44, 186, 253

- a) Đối với từng tham chiếu này, hãy xác định địa chỉ nhị phân, thẻ (tag), và chỉ số (index) với bộ nhớ đệm ánh xạ trực tiếp (direct-mapped cache) có 16 khối, mỗi khối chứa một từ. Đồng thời liệt kê xem mỗi tham chiếu là "hit" hay "miss", giả sử bộ nhớ đệm ban đầu rỗng.
- b) Đối với từng tham chiếu này, hãy xác định địa chỉ nhị phân, thẻ (tag), và chỉ số (index) với bộ nhớ đệm ánh xạ trực tiếp (direct-mapped cache) có kích thước khối là hai từ và tổng cộng 8 khối. Đồng thời liệt kê xem mỗi tham chiếu là "hit" hay "miss", giả sử bộ nhớ đệm ban đầu rỗng.

**Bài tập 2:** Bộ nhớ đệm (cache) đóng vai trò quan trọng trong việc cung cấp một hệ thống phân cấp bộ nhớ hiệu suất cao cho các bộ xử lý. Dưới đây là danh sách các tham chiếu địa chỉ bộ nhớ 32-bit, được biểu diễn dưới dạng địa chỉ byte:

12, 432, 87, 8, 765, 352, 764, 56, 433, 88, 748, 1012

- a) Đối với từng tham chiếu này, hãy xác định địa chỉ nhị phân, thẻ (tag), chỉ số (index), và bù (offset) với bộ nhớ đệm ánh xạ trực tiếp (direct-mapped cache) có 16 khối, mỗi khối chứa 4 byte. Đồng thời liệt kê xem mỗi tham chiếu là "hit" hay "miss", giả sử bộ nhớ đệm ban đầu rỗng.
- b) Đối với từng tham chiếu này, hãy xác định địa chỉ nhị phân, thẻ (tag), chỉ số (index), và bù (offset) với bộ nhớ đệm ánh xạ trực tiếp (direct-mapped cache) có kích thước khối là 8 byte và tổng cộng 8 khối. Đồng thời liệt kê xem mỗi tham chiếu là "hit" hay "miss", giả sử bộ nhớ đệm ban đầu rỗng.

## DÀNG 2: LIÊN KẾT TẬP HỢP

**Bài tập 1 [5.13/492 - Computer Organization and Design]:** Trong bài tập này, chúng ta sẽ nghiên cứu cách các chính sách thay thế ảnh hưởng đến tỷ lệ cache miss. Giả sử một bộ nhớ đệm liên kết theo tập hợp 2-way (2-way set associative cache) với 4 khối (blocks). Để giải quyết các bài toán trong bài tập này, bạn có thể vẽ một bảng như ví dụ bên dưới, với dãy địa chỉ là “0, 1, 2, 3, 4”.

| Address of Memory Block Accessed | Miss or hit | Evicted Block | Contents of Cache Blocks After Reference |        |        |        |
|----------------------------------|-------------|---------------|--|--------|--------|--------|
|                                  |             |               | Set 0                                    | Set 0  | Set 1  | Set 1  |
| 0                                | Miss        |               | Mem[0]                                   |        |        |        |
| 1                                | Miss        |               | Mem[0]                                   |        | Mem[1] |        |
| 2                                | Miss        |               | Mem[0]                                   | Mem[2] | Mem[1] |        |
| 3                                | Miss        |               | Mem[0]                                   | Mem[2] | Mem[1] | Mem[3] |
| 4                                | Miss        | 0             | Mem[4]                                   | Mem[2] | Mem[1] | Mem[3] |
| ...                              |             |               |  |        |        |        |

Xét dãy địa chỉ sau:

0, 2, 4, 8, 10, 12, 14, 16, 0

- a) Giả sử sử dụng chính sách thay thế LRU (Least Recently Used), dãy địa chỉ trên có bao nhiêu cache hit?
- b) Giả sử sử dụng chính sách thay thế MRU (Most Recently Used), dãy địa chỉ trên có bao nhiêu cache hit?
- c) Mô phỏng một chính sách thay thế ngẫu nhiên bằng cách tung đồng xu. Ví dụ, nếu kết quả là "mặt ngửa" thì loại bỏ khối đầu tiên trong tập hợp; nếu kết quả là "mặt sấp" thì loại bỏ khối thứ hai trong tập hợp. Dãy địa chỉ trên có bao nhiêu cache hit?
- d) Địa chỉ nào nên bị thay thế tại mỗi lần để tối ưu hóa số lượng cache hit? Dãy địa chỉ trên có bao nhiêu cache hit nếu bạn tuân theo chính sách "tối ưu"?
- e) Giải thích tại sao khó có thể triển khai một chính sách thay thế bộ nhớ đệm mà tối ưu cho mọi dãy địa chỉ.
- f) Giả sử bạn có thể quyết định tại mỗi lần truy cập bộ nhớ xem có muốn lưu trữ địa chỉ được yêu cầu vào cache hay không. Điều này có thể ảnh hưởng như thế nào đến tỷ lệ cache miss?

**Bài tập 2:** Bộ nhớ đệm (cache) rất quan trọng trong việc cung cấp một hệ thống phân cấp bộ nhớ hiệu năng cao cho bộ xử lý. Dưới đây là danh sách các địa chỉ bộ nhớ 32-bit, được cho dưới dạng địa chỉ theo từ (word address):

**4, 32, 23, 7, 12, 8, 15, 2, 13, 24, 14, 13**

Với mỗi địa chỉ trên, xác định địa chỉ bộ nhớ khói, lập bảng và chỉ rõ là **trúng (hit)** hay **trượt (miss)**, cho biết:

- + Bộ nhớ đệm **liên kết tập hợp** với 4 khói, mỗi tập hợp (set) gồm 2 khói, mỗi khói gồm 2 từ nhớ.
- + Giả sử bộ nhớ đệm ban đầu rỗng.
- + Sử dụng chính sách thay thế **LRU (Least Recently Used)**.

### DẠNG 3: LIÊN KẾT TOÀN PHẦN

**Bài tập 1:** Bộ nhớ đệm (cache) rất quan trọng trong việc cung cấp một hệ thống phân cấp bộ nhớ hiệu năng cao cho bộ xử lý. Dưới đây là danh sách các địa chỉ bộ nhớ 32-bit, được cho dưới dạng địa chỉ theo từ (word address):

**1, 4, 6, 4, 7, 8, 9, 7, 1, 3, 5, 9**

Với mỗi địa chỉ trên, xác định địa chỉ khói bộ nhớ, lập bảng và chỉ rõ là **trúng (hit)** hay **trượt (miss)**, cho biết khói nào thay thế khói nào (nếu có), cho biết:

- + Bộ nhớ đệm **liên kết toàn phần** (fully associative cache) với 4 khói (blocks), mỗi khói chứa 1 từ.
- + Giả sử bộ nhớ đệm ban đầu rỗng.
- + Sử dụng chính sách thay thế **LRU (Least Recently Used)**.

### DẠNG 4: TÌM OFFSET, INDEX VÀ TAG

**Bài tập 1:** Xác định các thành phần trong địa chỉ: Giả sử một bộ nhớ được tổ chức thành 16 dòng (**lines**) trong một cache trực tiếp (**direct-mapped**) với mỗi dòng chứa 4 từ (**words**). Bộ nhớ chính có tổng cộng  $2^{10}$  từ.

Tính kích thước của **block offset, index, tag**.

**Bài tập 2:** Mô phỏng truy cập bộ nhớ: Cho một hệ thống với:

- + Bộ nhớ chính có  $2^{16}$  từ.
- + Cache có 64 dòng (**lines**) và mỗi dòng chứa 16 từ.
- + Bộ nhớ được ánh xạ theo phương pháp **direct-mapped**.

1. Xác định số bit của **block offset**, **index**, và **tag**.
2. Với địa chỉ 0x2A3C, xác định:

- + **Index** tương ứng.
- + **Tag** tương ứng.
- + **Offset** tương ứng.

**Bài tập 3:** Cache tương giao đầy đủ (Fully Associative Cache): Một cache tương giao đầy đủ có:

- + 128 dòng.
- + Mỗi dòng chứa 4 từ.
- + Bộ nhớ chính có kích thước  $2^{20}$  từ.

1. Tính kích thước của **block offset**.
2. Tính kích thước của **tag**.
3. Địa chỉ 0x3F2A5 sẽ được lưu trữ như thế nào trong cache?

**Bài tập 4:** Cache ánh xạ tập hợp (Set-Associative Cache)

Cho một cache ánh xạ tập hợp với các thông số:

- + 4 set.
- + Mỗi set chứa 8 dòng.
- + Mỗi dòng chứa 2 từ.
- + Bộ nhớ chính có kích thước  $2^{14}$  từ.

1. Tính số bit của **block offset**, **index**, và **tag**.
  2. Với địa chỉ 0x1D34:
- + Xác định **set index**.
  - + Xác định **tag**.

**Bài tập 5:** Tính toán hiệu năng cache: Một chương trình truy cập các địa chỉ trong dãy: 0x4A0, 0x4A4, 0x4A8, và 0x4AC. Cache có các thông số:

- + 8 dòng, mỗi dòng chứa 4 từ.
  - + Ánh xạ theo phương pháp **direct-mapped**.
  - + Bộ nhớ chính có  $2^{12}$  từ.
1. Xác định các giá trị **index**, **tag**, và **block offset** cho từng địa chỉ.
  2. Phân tích hit/miss với cache ban đầu trống.

Donate:  
TRAN MINH PHU

