

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.

$$\text{index} = \log_2(16) = 4 \text{ bit}$$

$$\text{offset} = \log_2(1) = 0 \text{ bit}$$

$$\text{tag} = 32 - 4 = 28 \text{ bit}$$

Địa chỉ word	Nhị phân	Tag	Index	Hit/Miss
3	00...0011	0	3	Miss
180	00... 10110100	11	4	Miss
43	00...101011	2	11	Miss
2	00...0010	0	2	Miss
191	00...10111111	11	15	Miss
88	00...1011000	5	8	Miss
190	00...10111110	11	14	Miss
14	00...1110	0	14	Miss
181	00...10110101	11	5	Miss
44	00...101100	2	12	Miss
186	00...10111010	11	10	Miss
253	00...11111101	15	13	Miss

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.

$$\text{index} = \log_2(8) = 3 \text{ bit}$$

$$\text{offset} = \log_2(2) = 1 \text{ bit}$$

$$\text{tag} = 32 - 3 - 1 = 28 \text{ bit}$$

Địa chỉ word	Nhị phân	Tag	Index	Hit/Miss
3	00...0011	0	1	Miss
180	00... 10110100	11	2	Miss
43	00...101011	2	3	Miss
2	00...0010	0	1	Hit
191	00...10111111	11	7	Miss
88	00...1011000	5	4	Miss
190	00...10111110	11	7	Hit
14	00...1110	0	7	Miss
181	00...10110101	11	2	Hit
44	00...101100	2	6	Miss
186	00...10111010	11	5	Miss
253	00...11111101	15	6	Miss

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.

$$\text{index} = \log_2(16) = 4 \text{ bit}$$

$$\text{offset} = \log_2(4) = 2 \text{ bit}$$

$$\text{tag} = 32 - 4 - 2 = 26 \text{ bit}$$

Địa chỉ word	Nhị phân	Tag	Index	Hit/Miss
12	00...1100	0	3	Miss
432	00...110110000	6	12	Miss
87	00...1010111	0	10	Miss
8	00...1000	0	2	Miss
765	00...1011111101	11	15	Miss
352	00...101100000	6	8	Miss
764	00...1011111100	11	15	Hit
56	00...111000	0	14	Miss
433	00...110110001	6	12	Hit
88	00...1011000	1	6	Miss
748	00...1011101100	11	11	Miss
1012	00...1111110100	15	13	Miss

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.

DẠNG 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?

LRU: Thay thế khối ít được sử dụng gần đây nhất trong tập hợp

Address	Miss/Hit	Evicted	Set 0 (block 0, block 1)	Set 1 (block 0, block 1)
0	Miss	-	(0, -)	(-, -)
2	Miss	-	(0, -)	(2, -)
4	Miss	-	(0, 4)	(2, -)
8	Miss	0	(4, 8)	(2, -)
10	Miss	-	(4, 8)	(2, 10)
12	Miss	4	(8, 12)	(2, 10)
14	Miss	2	(8, 12)	(10, 14)
16	Miss	8	(12, 16)	(10, 14)
0	Miss	12	(16, 0)	(10, 14)

→ Số cache hit: 0

b) Giả sử sử dụng chính sách thay thế MRU (Most Recently Used), dãy địa chỉ trên có bao nhiêu cache hit?

MRU: Thay thế khối vừa được sử dụng gần đây nhất trong tập hợp

Address	Miss/Hit	Evicted	Set 0 (block 0, block 1)	Set 1 (block 0, block 1)
0	Miss	-	(0, -)	(-, -)
2	Miss	-	(0, -)	(2, -)
4	Miss	-	(0, 4)	(2, -)
8	Miss	4	(0, 8)	(2, -)
10	Miss	-	(0, 8)	(2, 10)
12	Miss	8	(0, 12)	(2, 10)
14	Miss	10	(0, 12)	(2, 14)
16	Miss	12	(0, 16)	(2, 14)
0	Hit	-	(0, 16)	(2, 14)

→ Số cache hit: 1

c) Mô phỏng một chính sách thay thế ngẫu nhiên bằng cách tung đồng xu. Ví dụ, nếu kết quả là "mặt ngựa" 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?

Tung đồng xu: Ngựa (thay khối 0), Sấp (thay khối 1). Giả sử: Ngựa, Sấp, Ngựa, Sấp, Ngựa.

Address	Miss/Hit	Evicted	Set 0 (block 0, block 1)	Set 1 (block 0, block 1)
0	Miss	-	(0, -)	(-, -)
2	Miss	-	(0, -)	(2, -)
4	Miss	-	(0, 4)	(2, -)
8	Miss	0 (Ngựa)	(8, 4)	(2, -)
10	Miss	-	(8, 4)	(2, 10)
12	Miss	4 (Sấp)	(8, 12)	(2, 10)
14	Miss	2 (Ngựa)	(8, 12)	(10, 14)
16	Miss	8 (Sấp)	(12, 16)	(10, 14)
0	Miss	12 (Ngựa)	(0, 16)	(10, 14)

→ Số cache hit: 0

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"?

Thay thế khối không được sử dụng trong tương lai gần nhất (dựa trên dãy địa chỉ).

Address	Miss/Hit	Evicted	Set 0 (block 0, block 1)	Set 1 (block 0, block 1)
0	Miss	-	(0, -)	(-, -)
2	Miss	-	(0, -)	(2, -)
4	Miss	-	(0, 4)	(2, -)
8	Miss	-	(0, 8)	(2, -)
10	Miss	-	(0, 8)	(2, 10)
12	Miss	8	(0, 12)	(2, 10)
14	Miss	10	(0, 12)	(2, 14)
16	Miss	12	(0, 16)	(2, 14)
0	Hit	-	(0, 16)	(2, 14)

→ Số cache hit: 1

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.

Chính sách tối ưu yêu cầu biết trước toàn bộ dãy truy cập địa chỉ trong tương lai, điều này không khả thi trong thực tế vì các chương trình có hành vi truy cập bộ nhớ động và không thể dự đoán chính xác.

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?

Nếu có thể chọn không lưu địa chỉ vào cache (bypass) khi biết địa chỉ đó không được tái sử dụng, tỷ lệ cache miss giảm vì tránh thay thế các khối hữu ích. Điều này tăng hiệu quả cache, đặc biệt trong các trường hợp truy cập địa chỉ chỉ dùng một lần.

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 khối, lập bảng và chỉ rõ là trúng (hit) hay trượt (miss), cho biết:

+ Bộ nhớ đệm liên kết tập hợp với 4 khối, mỗi tập hợp (set) gồm 2 khối, mỗi khối gồm 2 từ nhớ.

+ Giả sử bộ nhớ đệm ban đầu rỗng.

+ Sử dụng chính sách thay thế LRU (Least Recently Used).

Bài làm

Word Address	Nhị phân (32-bit)	Nhị phân Block	Set	Hit/Miss	Evicted Block	Set 0 (Block 0, Block 1)	Set 1 (Block 0, Block 1)
4	00...100	0010	1	Miss	-	(-, -)	(2, -)
32	00...100000	10000	0	Miss	-	(16, -)	(2, -)
23	00...10111	1011	1	Miss	-	(16, -)	(2, 11)
7	00...111	0011	1	Miss	2	(16, -)	(11, 3)
12	00...1100	0110	1	Miss	11	(16, -)	(3, 6)

8	00...1000	0100	0	Miss	-	(16, 4)	(3, 6)
15	00...1111	0111	1	Miss	3	(16, 4)	(6, 7)
2	00...10	0001	1	Miss	6	(16, 4)	(7, 1)
13	00...1101	0110	1	Miss	7	(16, 4)	(1, 6)
24	00...11000	1100	0	Miss	16	(4, 12)	(1, 6)
14	00...1110	0111	1	Miss	1	(4, 12)	(6, 7)
13	00...1101	0110	1	Hit	-	(4, 12)	(6, 7)

- Số cache hit: 1
- Số cache miss: 11

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 khối bộ nhớ, lập bảng và chỉ rõ là trúng (hit) hay trượt (miss), cho biết khối nào thay thế khối nào (nếu có), cho biết:

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

Bài làm

Địa chỉ	Khối bộ nhớ	Hit/Miss	Thay thế	Trạng thái bộ nhớ
1	1	Miss	-	[1]
4	4	Miss	-	[1, 4]
6	6	Miss	-	[1, 4, 6]
4	4	Hit	-	[1, 6, 4]
7	7	Miss	-	[1, 6, 4, 7]
8	8	Miss	1	[6, 4, 7, 8]
9	9	Miss	6	[4, 7, 8, 9]
7	7	Hit	-	[4, 8, 9, 7]
1	1	Miss	4	[8, 9, 7, 1]

3	3	Miss	8	[9, 7, 1, 3]
5	5	Miss	9	[7, 1, 3, 5]
9	9	Miss	7	[1, 3, 5, 9]

Tổng cộng: 2 Hit, 10 Miss

Thay thế xảy ra 5 lần:

- Bước 6: 8 thay thế 1
- Bước 7: 9 thay thế 6
- Bước 9: 1 thay thế 4
- Bước 10: 3 thay thế 8
- Bước 11: 5 thay thế 9