



## Ôn tập KTMT C4,5 - ktmt

Tiếng anh CNTT (Trường Đại học Công nghiệp Hà Nội)

## CHƯƠNG 4: KIẾN TRÚC HỆ THỐNG NHỚ

### 4.11.

CPU có 24 bit địa chỉ, bộ nhớ chính 256 KB chia làm 512 block nhớ, bộ nhớ cache có dung lượng 8 KB. Khi CPU được lệnh phát ra địa chỉ truy nhập bộ nhớ là 2F0825h. Hãy trình bày chi tiết phương pháp đọc cache theo kỹ thuật ánh xạ trực tiếp cho trường hợp phát ra địa chỉ trên và địa chỉ ô nhớ cần truy cập trong block theo từng trường hợp.

#### *Bài làm*

Dung lượng 1 block = Dung lượng bộ nhớ / Số block nhớ

$$C_{\text{block}} = C_{\text{memory}} / N_{\text{block}} = 256\text{KB} / 512 = 256 \times 1024 / 512 = 0.5\text{KB} = 512 = 2^9$$

→ Số bit để đánh địa chỉ lệch trong block là  $n_1 = 9$  bits

Số line của cache = Dung lượng cache / Dung lượng 1 line (1 block)

$$N_{\text{line}} = C_{\text{cache}} / C_{\text{block}} = 8\text{KB} / 0.5\text{KB} = 16 = 2^4 (\text{lines})$$

→ Số bit để đánh số hiệu line trong cache là  $n_2 = 4$  bits

Số block nhớ trong bộ nhớ chính như đã cho là 512 block

$$N_{\text{block}} = 512 = 2^9 \rightarrow \text{Số bit để đánh số hiệu block nhớ là } n_3 = 9 \text{ bits}$$

Số bits địa chỉ hợp lệ là:  $N = n_1 + n_2 + n_3 = 9 + 4 + 9 = 22$  bits

Thực hiện pp đọc cache với địa chỉ là 2F0825H:

+ Đổi sang nhị phân: 2F0825H = 1011110000100000100101B

+ CPU ktra địa chỉ truy nhập này gồm 22 bits = Số bit địa chỉ hợp lệ

+ CPU xác định số hiệu line (LI) cần truy nhập được xác định bởi  $n_2$  bit là 0100B → LI = 0100B = 4 bits

+ CPU xác định số hiệu block nhớ (BI) cần truy nhập được xác định bởi các bit cao còn lại tiếp theo sau các bit  $n_2$  → BI = 101111000B = 376

+ CPU xác định tính hợp lệ tiếp theo của địa chỉ phải thỏa mãn:  $LI = BI \bmod 16$ . Ta thấy  $376 \bmod 16 = 8$  → Block nạp không đúng line → Địa chỉ không hợp lệ.

### 4.12.

Trình bày phương pháp ánh xạ liên kết hoàn toàn trong tổ chức bộ nhớ cache.

#### *Gợi ý*

Trình bày phương pháp tổ chức cache theo kỹ thuật ánh xạ liên kết hoàn toàn cùng sơ đồ tổ chức cache.

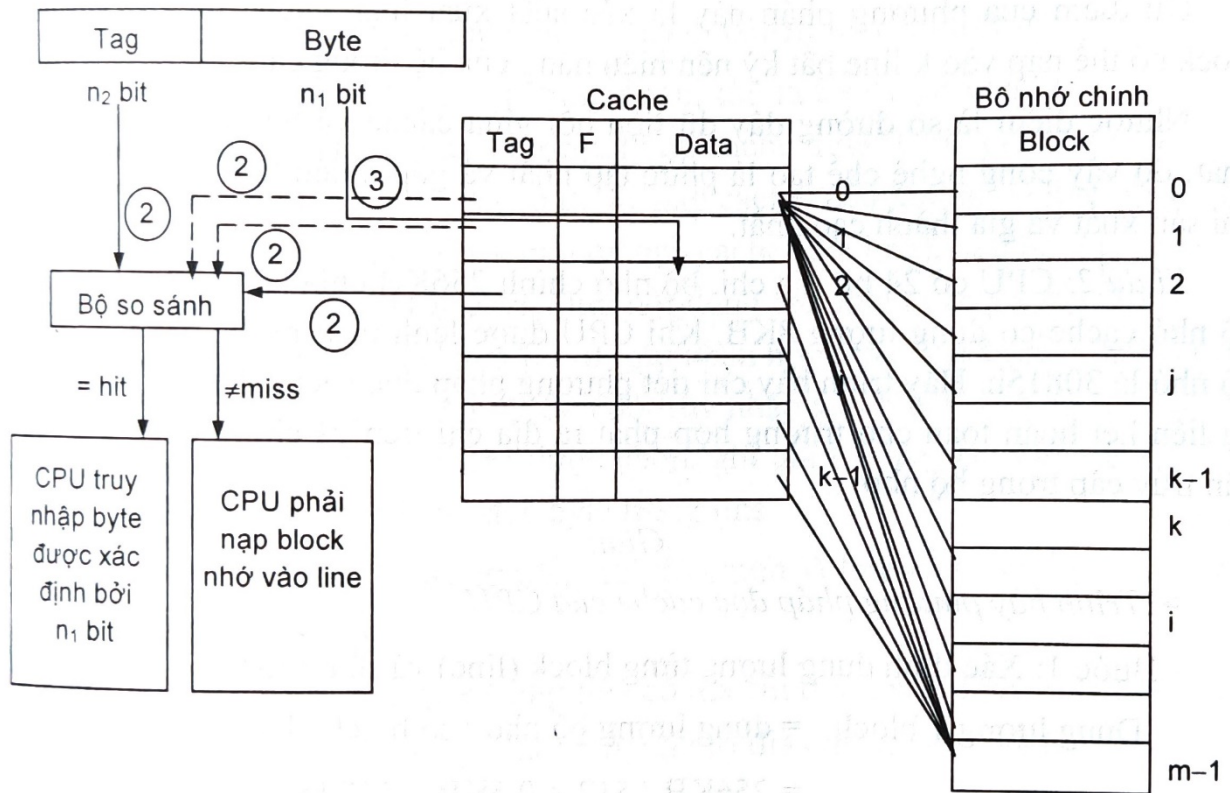
### ***Bài làm***

Phương pháp ánh xạ liên kết hoàn toàn:

- Cache: Được chia thành n khối hoặc dòng (block or line), từ Line 0 tới Line n-1
- Bộ nhớ:
  - + Được chia thành m khối hay dòng, từ Line 0 tới Line m-1
  - + Kích thước mỗi dòng cache bằng kích thước một dòng bộ nhớ
  - + Số lượng dòng trong bộ nhớ có thể lớn hơn nhiều số lượng dòng của cache ( $m \gg n$ )
- Ánh xạ :
  - + Một dòng của bộ nhớ có thể ánh xạ tới dòng bất kì của cache
  - + Line i của bộ nhớ có thể ánh xạ tới line j của cache
- Địa chỉ ánh xạ kết hợp đầy đủ

Tag	Word
-----	------

  - + Tag (bit) là địa chỉ của line trong bộ nhớ (page = 1)
  - + Word (bit) là địa chỉ của từ trong line
- Ưu điểm :
  - + Ít xung đột vì ánh xạ linh hoạt
  - + Tỷ lệ hit cao hơn
- Nhược điểm:
  - + Chậm vì phải tìm kiếm địa chỉ bộ nhớ trong cache
  - + Phức tạp vì có thêm n bộ so sánh địa chỉ trong cache
  - + Thường sử dụng cho cache có kích thước nhỏ



Sơ đồ tổ chức cache theo kỹ thuật ánh xạ liên kết hoàn toàn

#### 4.13.

Trình bày thao tác đọc cache theo phương pháp ánh xạ liên kết hoàn toàn trong tổ chức bộ nhớ cache.

##### Bài làm

- Bước 1: CPU kiểm tra tính hợp lệ của địa chỉ truy nhập, nếu số bit có nghĩa biểu diễn địa chỉ truy nhập lớn hơn số bit hợp lệ (lớn hơn  $n$ ), địa chỉ này không hợp lệ, tiến trình phải dừng.
- Bước 2: Bộ điều khiển cache đọc lần lượt các giá trị trên tag thuộc từng line trong cache và so sánh với giá trị do  $n_2$  bit xác định.
- + Nếu đây là trường hợp cache hit, thực hiện bước 3 của thao tác.
- + Nếu là trường hợp cache miss thì CPU phải chọn 1 line nào đó để nạp cache, nếu không có line rỗng CPU phải chọn 1 line đang chứa block nhớ để nạp cache (đổi line), nếu bit cờ F của line được chọn bằng 1 thì CPU phải sao lưu nội dung của line ra block nhớ có số hiệu bằng giá trị trên tag của line, sửa  $F = 0$  (ghi nhận nội dung line trùng với nội dung của 1 block trên bộ nhớ chính). Sau đó truy nhập vào block nhớ có địa chỉ xác định bởi  $n_2$  bit cao nhất trong  $n$

bit địa chỉ hợp lệ và nạp nội dung của nó vào line được chọn, sửa tag của line bằng số hiệu block nhớ vừa nạp. Và cuối cùng thực hiện Bước 3 của thao tác.

- Bước 3: Byte dữ liệu cần đọc có địa chỉ do n1 bit thấp xác định trong line được đọc vào CPU.

#### 4.14.

CPU có 24 bit địa chỉ, bộ nhớ chính 256 KB chia làm 512 block nhớ, bộ nhớ cache có dung lượng 8 KB. Khi CPU được lệnh phát ra địa chỉ truy nhập bộ nhớ là 3280Ah. Hãy trình bày chi tiết phương pháp đọc cache theo kỹ thuật ánh xạ liên kết hoàn toàn và chỉ ra địa chỉ ô nhớ cần truy cập trong bộ nhớ theo trường hợp phát ra địa chỉ trên.

#### **Gợi ý**

- Hoàn thành và nắm chắc nội dung câu 4.13 và câu 4.14.
- Xem **ví dụ 2** trong nội dung **phương pháp ánh xạ liên kết hoàn toàn** thuộc mục 4.1.3.3. – tài liệu giáo trình kiến trúc máy tính, Đại học CNHN.

#### **Bài làm**

Dung lượng 1 block = Dung lượng bộ nhớ / Số block nhớ

$$C_{\text{block}} = C_{\text{memory}} / N_{\text{block}} = 256\text{KB} / 512 = 256 \times 1024 / 512 = 0.5\text{KB} = 512 = 2^9$$

→ Số bit để đánh địa chỉ lệch trong block là n1 = 9 bits

Số line của cache = Dung lượng cache / Dung lượng 1 line (1 block)

$$N_{\text{line}} = C_{\text{cache}} / C_{\text{block}} = 8\text{KB} / 0.5\text{KB} = 16 = 2^4 (\text{lines})$$

Số block nhớ trong bộ nhớ chính như đã cho là 512 block

$$N_{\text{block}} = 512 = 2^9 \rightarrow \text{Số bit để đánh số hiệu block nhớ là } n2 = 9 \text{ bits}$$

Số bits địa chỉ hợp lệ là:  $N = n1 + n2 = 9 + 9 = 18 \text{ bits}$

Thực hiện pp đọc cache với địa chỉ là 3280AH:

3280AH = 110010100000001010B → Chiều dài địa chỉ là 18 bits → Hợp lệ về chiều dài → Địa chỉ hợp lệ

→ CPU phát địa chỉ này lên bus địa chỉ.

+ Bộ điều khiển cache xác định giá trị của trường Tag là 9 bit cao của địa chỉ trên để xác định số hiệu block bộ nhớ cần truy cập: 110010100 = 404

+ Bộ điều khiển cache sẽ so sánh trường Tag của tất cả các line với 404.

Có 2 khả năng xảy ra.

+ Trường hợp 1: Không có Tag của line nào có giá trị 404  $\rightarrow$  Block cần truy cập chưa có trong cache (miss cache). CPU phải nạp block có số hiệu 404 từ bộ nhớ chính vào cache. CPU tìm line nào trong cache sử dụng kém hiệu quả nhất để thay bằng block mới. Nếu bit F của line bị thay thế = 1 thì nội dung của line đó sẽ được ghi trả lại bộ nhớ vào đúng block có giá trị bằng trường Tag của line đó. Sau đó xóa F = 0 rồi nạp block nhớ có số hiệu 404 vào line này và cập nhật trường Tag của nó bằng 404. Sau đó CPU sẽ đọc byte cần trong line này.

+ Trường hợp 2: Có một Tag của line nào đó = 404 (hit cache). CPU sẽ đọc byte có số hiệu là 9 bit thấp của địa chỉ: 000001010 = 10  $\rightarrow$  đọc byte có số hiệu 10 của line đó.

$\rightarrow$  Địa chỉ vật lý của ô nhớ cần truy cập chính là: 3280AH

#### 4.15.

CPU có 24 bit địa chỉ, bộ nhớ chính 256 KB chia làm 512 block nhớ, bộ nhớ cache có dung lượng 8 KB. Khi CPU được lệnh phát ra địa chỉ truy nhập bộ nhớ là 7280Ah. Hãy trình bày chi tiết phương pháp đọc cache theo kỹ thuật ánh xạ liên kết hoàn toàn và chỉ ra địa chỉ ô nhớ cần truy cập trong bộ nhớ theo trường hợp phát ra địa chỉ trên.

#### **Bài làm**

Dung lượng 1 block = Dung lượng bộ nhớ / Số block nhớ

$$C_{\text{block}} = C_{\text{memory}} / N_{\text{block}} = 256\text{KB} / 512 = 256 \times 1024 / 512 = 0.5\text{KB} = 512 = 2^9$$

$\rightarrow$  Số bit để đánh địa chỉ lệch trong block là  $n_1 = 9$  bits

Số line của cache = Dung lượng cache / Dung lượng 1 line (1 block)

$$N_{\text{line}} = C_{\text{cache}} / C_{\text{block}} = 8\text{KB} / 0.5\text{KB} = 16 = 2^4 (\text{lines})$$

Số block nhớ trong bộ nhớ chính như đã cho là 512 block

$$N_{\text{block}} = 512 = 2^9 \rightarrow \text{Số bit để đánh số hiệu block nhớ là } n_2 = 9 \text{ bits}$$

Số bits địa chỉ hợp lệ là:  $N = n_1 + n_2 = 9 + 9 = 18$  bits

Thực hiện pp đọc cache với địa chỉ là 7280AH:

7280AH= 111001010001010B → Chiều dài địa chỉ là 19 bits → Không hợp lệ về chiều dài → Địa chỉ không hợp lệ

#### 4.16.

Trình bày phương pháp ánh xạ liên kết tập hợp trong tổ chức bộ nhớ cache.

#### **Gợi ý**

Trình bày phương pháp xây dựng cache theo kỹ thuật ánh xạ liên kết tập hợp cùng sơ đồ tổ chức cache.

#### **Bài làm**

Phương pháp xây dựng cache theo kỹ thuật ánh xạ liên kết tập hợp:

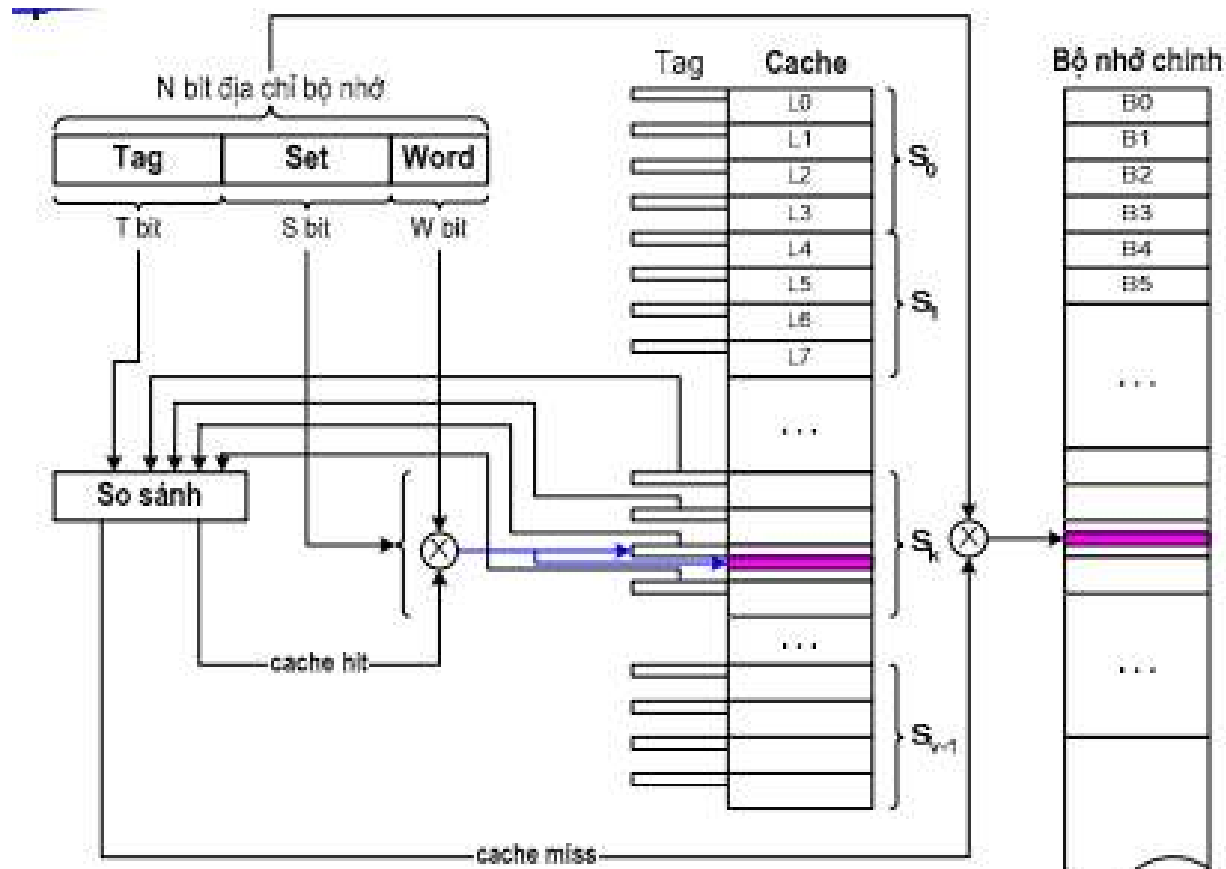
- Cache:
  - + Được chia thành k đường (ways) có kích thước bằng nhau
  - + Mỗi đường được chia thành n dòng (block or line), từ Line 0 tới Line n-1
- Bộ nhớ:
  - + Được chia thành m trang, từ page 0 tới page m-1
  - + Kích thước trang page bằng kích thước way của cache
  - + Mỗi trang có n line, từ Line 0 tới Line n-1
- Ánh xạ:
  - + Page được ánh xạ tới Way (ánh xạ linh hoạt):
    - Một page bộ nhớ có thể được ánh xạ tới way bất kì của cache
  - + Line của page ánh xạ tới line của way (ánh xạ cố định):
    - Line0 của page i được ánh xạ tới Line0 của way j;
    - Line1 của page I được ánh xạ tới Line1 của way j;
    - ...
    - Line n-1 của page I được ánh xạ tới Linen-1 của way j;
- Địa chỉ ánh xạ tập kết hợp

Tag	Set	Word
-----	-----	------

+Tag (bit): là địa chỉ của trang trong bộ nhớ

+ Set (bit): là địa chỉ của line trong way của cache

- +Word (bit): là địa chỉ của word trong line
- Ưu điểm:
  - + Nhanh vì ánh xạ trực tiếp được sử dụng cho ánh xạ dòng
  - + Ít xung đột vì ánh xạ từ trang nhớ tới đường của cache là linh hoạt
  - + Tỷ lệ tìm thấy (hit) cao
- Nhược điểm:
  - + Thiết kế và điều khiển phức tạp vì cache được chia thành các way.



Sơ đồ tổ chức cache theo kỹ thuật ánh xạ liên kết tập hợp

- Đặc điểm của ánh xạ liên kết tập hợp

N bits địa chỉ hợp lệ được chia làm 3 phần:

- + W bits thấp nhất dùng để xác định số hiệu của từ nhớ trong line (block) cần truy cập → kích thước của một block =  $2^W$
- + S bits kế tiếp dùng để xác định một trong  $2^S$  set cần truy cập.



+ T bits trọng số cao nhất dùng để xác định block cần truy cập.

$$\rightarrow N = T + S + W$$

#### 4.17.

Trình bày thao tác đọc cache theo phương pháp ánh xạ liên kết tập hợp trong tổ chức bộ nhớ cache.

##### **Bài làm**

- SI (Set Index)– là số hiệu Set cần truy nhập.
- BI (Block Index)- là số hiệu Block nhớ cần truy nhập.

+ Bước 1: CPU kiểm tra tính hợp lệ của địa chỉ truy nhập.

- Trường hợp 1: Nếu số bit có nghĩa biểu diễn địa chỉ truy nhập lớn hơn số bit hợp lệ (lớn hơn n), địa chỉ này không hợp lệ, tiến trình phải dừng.
- Trường hợp 2: Nếu  $SI \neq (BI \bmod )$  thì địa chỉ này không hợp lệ, tiến trình phải dừng (Số lượng set trong cache là ).

Ngược lại, CPU thực hiện phát ra địa chỉ truy nhập lên Bus A, sang bước 2.

+ Bước 2: Căn cứ vào n2 bit dùng để đánh số hiệu cho set, bộ điều khiển cache các định được set cần truy nhập.

+ Bước 3: Bộ điều khiển cache đọc lần lượt từng giá trị trên tag thuộc từng line trong set vừa tìm được và so sánh với giá trị BI do n3 bit xác định.

- Nếu đây là trường hợp cache hit, thực hiện bước 3 của thao tác
- Nếu là trường hợp cache miss thì CPU phải tiến hành nạp cache: CPU phải chọn 1 line nào đó trong set để nạp cache, nếu không có line rỗng CPU phải chọn 1 line đang chứa block nhớ để nạp cache, nếu bit cờ F của line được chọn bằng 1 thì CPU phải sao lưu nội dung của line ra block nhớ có số hiệu bằng giá trị trên tag của line, sửa F = 0 (ghi nhận nội dung line trùng với nội dung của 1 block trên bộ nhớ chính). Sau đó truy nhập vào block nhớ có địa chỉ xác định bởi n3 bit cao trong n bit và nạp nội dung của nó vào line được chọn, sửa tag thành số hiệu block vừa nạp. Và cuối cùng thực hiện Bước 4 của thao tác.

+ Bước 4: Từ dữ liệu hoặc byte dữ liệu cần đọc có địa chỉ do n1 bit thấp xác định trong line được đọc vào CPU.

#### 4.18.

CPU có 24 bit địa chỉ, bộ nhớ chính 256 KB chia làm 512 block nhớ, bộ nhớ cache có dung lượng 8 KB chia làm 4 set. Khi CPU được lệnh phát ra địa chỉ truy nhập bộ nhớ là 22A07h. Hãy trình bày chi tiết phương pháp đọc cache và chỉ ra địa chỉ ô nhớ cần truy cập trong bộ nhớ theo trường hợp phát ra địa chỉ trên.

### **Gợi ý**

- Hoàn thành và nắm chắc nội dung câu 4.13 và câu 4.14.
- Xem **ví dụ 3** trong nội dung **phương pháp ánh xạ liên kết tập hợp** thuộc mục 4.1.3.3. – tài liệu giáo trình kiến trúc máy tính, Đại học CNHN.

### **Bài làm**

Bước 1: Xác định dung lượng của 1 block (line) nhớ.

$$C_{\text{block}} = C_{\text{memory}}/N_{\text{block}} = 256\text{KB}/512 = 256 \times 1024/512 = 0.5\text{KB} = 512 = 2^9$$

$$\rightarrow W = 9 \text{ bits}$$

Bước 2: Xác định số bit để đánh số hiệu cho tập (set) của cache

$$n_{\text{set}} = 4 = 2^2 \rightarrow S = 2 \text{ bits}$$

+ Tính số lines của cache:

$$n_{\text{line}} = C_{\text{cache}}/C_{\text{block}} = 8\text{KB}/0.5\text{KB} = 16 = 2^4(\text{lines})$$

+ Tính số line trên 1 set là:

$$n_{\text{line/set}} = n_{\text{line}}/n_{\text{set}} = 16/4 = 4 (\text{line/1set})$$

Bước 3: Xác định số bit để đánh số hiệu cho block

$$N_{\text{block}} = 512 = 2^9 \rightarrow T = 9 \text{ bits}$$

Bước 4: Chiều dài tối đa của bits địa chỉ hợp lệ là :

$$N = T + S + W = 9 + 2 + 9 = 20 \text{ bits}$$

Bước 5: Xác định địa chỉ CPU phát ra có hợp lệ không:

22A07H = 100010101000000111B  $\rightarrow$  Chiều dài địa chỉ là 18 bits chiều dài tối đa địa chỉ hợp lệ  $\rightarrow$  Địa chỉ hợp lệ về chiều dài(1)

$$SI = 01B = 1$$

$$BI = 1000101B = 69$$

Theo ánh xạ liên kết tập hợp thì block 69 phải nạp vào set sau:

$69 \bmod 4 = 1 \rightarrow$  Block nạp đúng set (2)  $\rightarrow$  Từ (1) và (2)  $\rightarrow$  Địa chỉ CPU phát là hợp lệ

Bước 6: Truy xuất cache

Từ địa chỉ CPU phát ra thì đơn vị điều khiển cache sẽ đi kiểm tra 4 line của set 2 xem trường tag có bằng với 69 không?

Có 2 khả năng xảy ra:

- Không có trường tag nào bằng 69  $\rightarrow$  miss cache, Khi đó CPU phải nạp block nhớ có số hiệu 69 vào một line của Set 2  $\rightarrow$  như trường hợp 2.
- Có 1 tag của line nào đó trong set 2 bằng 69  $\rightarrow$  hit cache  $\rightarrow$  CPU đọc ô nhớ có số hiệu (địa chỉ lệch)  $W = 000000111B = 7$

Địa chỉ vật lý:  $TW = \underline{001000101000000111}B = 08A07H$

#### 4.19.

CPU có 24 bit địa chỉ, bộ nhớ chính 256 KB chia làm 512 block nhớ, bộ nhớ cache có dung lượng 8 KB. Khi CPU được lệnh phát ra địa chỉ truy nhập bộ nhớ là 2280Ah. Hãy trình bày chi tiết phương pháp đọc cache theo kỹ thuật ánh xạ liên kết tập hợp và chỉ ra địa chỉ ô nhớ cần truy cập trong bộ nhớ theo trường hợp phát ra địa chỉ trên.

#### ***Bài làm***

Bước 1: Xác định dung lượng của 1 block (line) nhớ.

$$C_{\text{block}} = C_{\text{memory}} / N_{\text{block}} = 256KB / 512 = 256 \times 1024 / 512 = 0.5KB = 512 = 2^9$$

$$\rightarrow W = 9 \text{ bits}$$

Bước 2: Xác định số bit để đánh số hiệu cho tập (set) của cache

$$n_{\text{set}} = 4 = 2^2 \rightarrow S = 2 \text{ bits}$$

+ Tính số lines của cache:

$$n_{\text{line}} = C_{\text{cache}} / C_{\text{block}} = 8KB / 0.5KB = 16 = 2^4 \text{ (lines)}$$

+ Tính số line trên 1 set là:

$$n_{\text{line/set}} = n_{\text{line}} / n_{\text{set}} = 16 / 4 = 4 \text{ (line/1set)}$$

Bước 3: Xác định số bit để đánh số hiệu cho block

$$N_{\text{block}} = 512 = 2^9 \rightarrow T = 9 \text{ bits}$$

Bước 4: Chiều dài tối đa của bits địa chỉ hợp lệ là :

$$N = T + S + W = 9 + 2 + 9 = 20 \text{ bits}$$

Bước 5: Xác định địa chỉ CPU phát ra có hợp lệ không:

2280AH = 100010100000001010B → Chiều dài địa chỉ là 18 bits chiều dài tối đa địa chỉ hợp lệ → Địa chỉ hợp lệ về chiều dài(1)

$$SI = 00B = 0$$

$$BI = 000001010B = 10$$

Theo ánh xạ liên kết tập hợp thì block 10 phải nạp vào set sau:

$10 \bmod 4 = 2 \rightarrow$  Block nạp không đúng set (2) → Từ (1) và (2) → Địa chỉ CPU phát là không hợp lệ

#### 4.20.

CPU có 24 bit địa chỉ, bộ nhớ chính 256 KB chia làm 512 block nhớ, bộ nhớ cache có dung lượng 8 KB. Khi CPU được lệnh phát ra địa chỉ truy nhập bộ nhớ là 120812h. Hãy trình bày chi tiết phương pháp đọc cache theo kỹ thuật ánh xạ liên kết tập hợp và chỉ ra địa chỉ ô nhớ cần truy cập trong bộ nhớ theo trường hợp phát ra địa chỉ trên.

#### ***Bài làm***

Bước 1: Xác định dung lượng của 1 block (line) nhớ.

$$C_{\text{block}} = C_{\text{memory}} / N_{\text{block}} = 256\text{KB} / 512 = 256 \times 1024 / 512 = 0.5\text{KB} = 512 = 2^9$$

$$\rightarrow W = 9 \text{ bits}$$

Bước 2: Xác định số bit để đánh số hiệu cho tập (set) của cache

$$n_{\text{set}} = 4 = 2^2 \rightarrow S = 2 \text{ bits}$$

+ Tính số lines của cache:

$$n_{\text{line}} = C_{\text{cache}} / C_{\text{block}} = 8\text{KB} / 0.5\text{KB} = 16 = 2^4 \text{ (lines)}$$

+ Tính số line trên 1 set là:

$$n_{\text{line/set}} = n_{\text{line}} / n_{\text{set}} = 16 / 4 = 4 \text{ (line/1set)}$$

Bước 3: Xác định số bit để đánh số hiệu cho block

$$N_{\text{block}} = 512 = 2^9 \rightarrow T = 9 \text{ bits}$$

Bước 4: Chiều dài tối đa của bits địa chỉ hợp lệ là :

$$N = T + S + W = 9 + 2 + 9 = 20 \text{ bits}$$

Bước 5: Xác định địa chỉ CPU phát ra có hợp lệ không:

$120812H = 100100000100000010010B \rightarrow$  Chiều dài địa chỉ là 21 bits chiều dài tối đa địa chỉ hợp lệ  $\rightarrow$  Địa chỉ không hợp lệ về chiều dài

$\rightarrow$  Địa chỉ CPU phát là không hợp lệ

#### 4.21. Làm trong vở

Giả sử bộ nhớ vật lý có dung lượng 512 MB, chương trình gồm 5 module, xác định địa chỉ vật lý tương ứng với các địa chỉ logic phát ra là  $\langle 2, 43h \rangle$ ,  $\langle 3, 4FFh \rangle$ ,

$\langle 4, 600h \rangle$ ,  $\langle 5, 100h \rangle$ , cho biết bảng quản lý phân đoạn như sau:

P	A	L
1	100h	200h
0	-	400h
1	300h	700h
0	-	500h
1	A00h	600h

#### Bài làm

##### a. Với các địa chỉ logic phát ra là $\langle 2, 43h \rangle$ (S,d)

- Bảng SCB cho thấy chương trình có 5 đoạn, đánh số từ 0 đến 4.
- Số hiệu đoạn cần truy nhập là  $S = 2 \leq 4$ , địa chỉ đoạn cần truy nhập là hợp lệ.
- Phần tử 2 trong bảng SCB, cho biết đoạn 2 có độ dài là  $L = 700h$  byte, so sánh :  $L = 700h > d = 43h$ , nên địa chỉ lệch cần truy nhập là hợp lệ
- Phần tử 2 trong bảng SCB có  $P = 1$ , cho biết đoạn 2 đã được nạp vào vùng nhớ có địa chỉ đầu  $A = 300h$ .
- Địa chỉ vật lý cần truy nhập  $= A + d = 300h + 43h = 343h$

##### b. Với các địa chỉ logic phát ra là $\langle 4, 4FFh \rangle$

##### c. Với các địa chỉ logic phát ra là $\langle 3, 4FFh \rangle$

- Bảng SCB cho thấy chương trình có 5 đoạn, đánh số từ 0 đến 4.
- Số hiệu đoạn cần truy nhập là  $S = 3 < 5$ , địa chỉ đoạn cần truy nhập là hợp lệ.
- Phần tử 3 trong bảng SCB, cho biết đoạn 3 có độ dài là  $L = 500h$  byte, so sánh :  $L = 500h > d = 4FFh$ , nên địa chỉ lệch cần truy nhập là hợp lệ

- Phần tử 2 trong bảng SCB có  $P = 0$ , cho biết đoạn 2 đã chưa được nạp bộ nhớ chính. Do vậy hệ thống cần thực hiện nạp đoạn hoặc đổi đoạn để tiến trình tiếp tục thực thi.

**d. Với các địa chỉ logic phát ra là  $\langle 1, 306h \rangle$**

**e. Với các địa chỉ logic phát ra là  $\langle 4, 600h \rangle$**

- Bảng SCB cho thấy chương trình có 5 đoạn, đánh số từ 0 đến 4.  
 - Số hiệu đoạn cần truy nhập là  $S = 3 < 5$ , địa chỉ đoạn cần truy nhập là hợp lệ.

- Phần tử 2 trong bảng SCB, cho biết đoạn 3 có độ dài là  $L = 500h$  byte, so sánh :  $L = 600h \leq d = 600h$ , nên địa chỉ lệch cần truy nhập không hợp lệ, tiến trình dừng.

**f. Với các địa chỉ logic phát ra là  $\langle 2, 701h \rangle$**

**g. Với các địa chỉ logic phát ra là  $\langle 5, 33h \rangle$**

#### 4.22.

Giả sử bộ nhớ vật lý, kích thước 32 bytes, chia làm 8 trang vật lý. Chương trình có kích thước 18 bytes có nội dung '0123456789ABCDEFGH'. Và bảng quản lý trang (PCB) có nội dung sau:

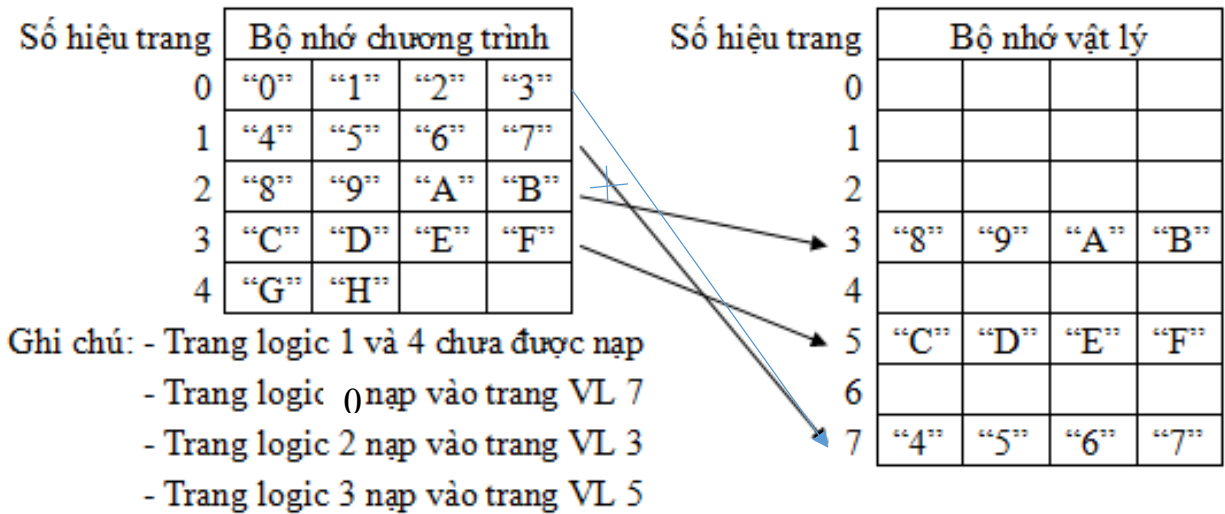
P	A
1	7
0	-
1	3
1	5
0	-

- Xác định địa chỉ vật lý truy nhập tương ứng với các địa chỉ logic sau:  $\langle 0,2 \rangle$ ;  $\langle 2,3 \rangle$ ;  $\langle 1,3 \rangle$ ;  $\langle 4,1 \rangle$ ;  $\langle 2,4 \rangle$ ;  $\langle 3,5 \rangle$ ;  $\langle 5,1 \rangle$
- Cho biết nội dung các ô nhớ tương ứng với các địa chỉ trên.

#### Bài làm

##### \* Phần tính toán chung

- Kích thước một trang vật lý là:  $L = \text{dung lượng bộ nhớ vật lý} / \text{số trang vật lý}$ .  
 $= 32/8 = 4 \text{ bytes} = 2^2$ .
- Vậy số bit để đánh độ lệch trong trang là:  $n_1 = 2$ .
- Số bit để đánh số hiệu trang vật lý là:  $8 = 2^3$ , vậy  $n_2 = 3$ .
- Số bit để đánh địa chỉ vật lý là:  $32 = 2^5$ , vậy  $n = 5$  ( $n = n_2 + n_1$ ).
- Số trang logic của chương trình = dung lượng chương trình/  $L = 18/4 = 4,5$ . Vậy số trang logic của chương trình phải là 5. (bảng PCB trên là phù hợp).
- Căn cứ vào bảng PCB, dung lượng bộ nhớ vật lý và kích thước của một trang, ta có sơ đồ nạp trang như sau:



**a. Xác định địa chỉ vật lý (ĐCVL) truy nhập tương ứng với địa chỉ logic <0,2>**

- Số hiệu trang logic cần truy nhập là:  $N_p = 0 < 5$  (số trang logic của chương trình) nên số hiệu trang cần truy nhập là hợp lệ.
- Địa chỉ lệch cần truy nhập trong trang là:  $d = 2 < 4$  (độ dài trang), nên địa chỉ lệch cần truy nhập trong trang là hợp lệ.
- Phần tử 0 trong bảng PCB có  $P = 1$ , cho biết trang logic 0 đã được nạp vào trang vật lý  $A = 7$ .  $A = 7 < 8$  (số trang vật lý), nên giá trị này là hợp lệ.
- ĐCVL cần truy nhập là:

$$\begin{aligned} \text{ĐCVL} &= A \text{ SHL } n_1 \text{ OR } d = 7 \text{ SHL } 2 \text{ OR } 2 = 0111b \text{ SHL } 2 \text{ OR } 10b \\ &= 11100b \text{ OR } 10b = 11110b = 30. \end{aligned}$$

- Nhìn vào sơ đồ ta thấy nội dung ô nhớ có địa chỉ 30 là "2".

**b. Xác định địa chỉ vật lý truy nhập tương ứng với địa chỉ logic <2,3>**

$N_p = 2 < 5 \rightarrow$  Số hiệu trang cần truy nhập là hợp lệ.

Phần tử 2 trong bảng PCB có:

- $P = 1 \rightarrow$  Trang 2 đã nạp vào RAM
- $A = 3 \rightarrow$  Trang 2 đã nạp vào trong địa chỉ vật lý

$d = 3 < 4 \rightarrow$  Địa chỉ lệch cần truy nhập trong trang là hợp lệ  $\rightarrow$  Địa chỉ vật lý cần truy nhập là:

$$PA = A * L + d = 3 * 4 + 3 = 15$$

Hoặc

$$\text{Dịch trái } A = 3 \text{ đi } n_1 = 2 \text{ bit OR } d = 3$$

$$\begin{array}{r}
 001100b \\
 + \quad 0011b \\
 \hline
 001111b = 15
 \end{array}$$

8	12
9	13
A	14
B	15

→ Nội dung: “B”

**c. Xác định địa chỉ vật lý truy nhập tương ứng với địa chỉ logic <1,3>**

- Số hiệu trang logic cần truy nhập là:  $N_p = 1 < 5$  (số trang logic của chương trình). Nên số hiệu trang cần truy nhập là hợp lệ.
- Địa chỉ lệch cần truy nhập trong trang là:  $d = 3 < 4$  (độ dài trang), nên địa chỉ lệch cần truy nhập trong trang là hợp lệ.
- Phần tử 1 trong bảng PCB có  $P = 0$ , cho biết trang logic 0 chưa được nạp vào bộ nhớ vật lý, nên hệ thống cần nạp trang hoặc đổi trang.

**d. Xác định địa chỉ vật lý truy nhập tương ứng với địa chỉ logic <4,1>**

- Số hiệu trang logic cần truy nhập là:  $N_p = 4 < 5$  (số trang logic của chương trình). Nên số hiệu trang cần truy nhập là hợp lệ.
  - Địa chỉ lệch cần truy nhập trong trang là:  $d = 1 < 4$  nên địa chỉ lệch cần truy nhập trong trang là hợp lệ.
  - Phần tử 4 trong bảng PCB có  $P = 0 \rightarrow$  Trang 4 không nạp vào RAM
- Hệ thống cần nạp trang hoặc đổi trang

**e. Xác định địa chỉ vật lý truy nhập tương ứng với địa chỉ logic <3,5>**

- Số hiệu trang logic cần truy nhập là:  $N_p = 3 < 5$  (số trang logic của chương trình). Nên số hiệu trang cần truy nhập là hợp lệ.
- Địa chỉ lệch cần truy nhập trong trang là:  $d = 5 \geq 4$  (độ dài trang), nên địa chỉ lệch cần truy nhập trong trang là không hợp lệ. Do vậy tiến trình phải dừng.

**f. Xác định địa chỉ vật lý truy nhập tương ứng với địa chỉ logic <4,4>**



- Số hiệu trang logic cần truy nhập là:  $N_p = 4 < 5$  (số trang logic của chương trình).  
Nên số hiệu trang cần truy nhập là hợp lệ.

- Địa chỉ lệch cần truy nhập trong trang là:  $d = 4 < 4$  (độ dài trang), nên địa chỉ lệch cần truy nhập trong trang là không hợp lệ. Do vậy tiến trình phải dừng.

**g. Xác định địa chỉ vật lý truy nhập tương ứng với địa chỉ logic <5,3>**

- Số hiệu trang logic cần truy nhập là:  $N_p = 5 < 5$  (số trang logic của chương trình).  
Nên số hiệu trang cần truy nhập là không hợp lệ.

**4.23.**

Giả sử bộ nhớ vật lý (BNVL) có dung lượng 64MB, chương trình gồm 4 modul, xác định địa chỉ truy cập trong chiến lược phân trang – phân đoạn, với:

SCB			PCB <sub>0</sub>		PCB <sub>2</sub>		PCB <sub>3</sub>	
Ps	Ap	Ls	P	A	P	A	P	A
1	400400h	4	0		1	407h	1	40Ah
0	-	4	1	405h	0	-	1	40Fh
1	400C00h	3	1	406h	1	408h	0	-
1	401000h	4	0	-			1	40Bh

Biết kích thước 1 trang là 4KB, địa chỉ đầu của bảng phân đoạn  $R_s = 400000h$ .

Xác định địa chỉ vật lý cần truy cập theo địa chỉ logic sau:

<0, 2, 20Bh>; <0, 1, 7Ah>; <1, 0, 46h>; <1, 1, 1001h>;  
<2, 1, 1000h>; <2,3,200h>; <4, 1, 57h>

**\* Phân tích toán chung**

- Kích thước một trang vật lý là:

$$L = 4 \text{ KB} = 2^2 * 2^{10} \text{ bytes} = 2^{12} \text{ bytes}$$

$$= 10000000000000 \text{ B bytes} = 1000 \text{ H bytes}$$

- Vậy số bit để đánh độ lệch trong trang là:  $n_1 = 12 \text{ bit}$ .

- Số trang vật lý = dung lượng BNVL / kích thước 1 trang

$$= 64 \text{ MB} / 4 \text{ KB} = 64 * 2^{10} \text{ KB} / 4 \text{ KB} = 16 * 2^{10} = 2^4 * 2^{10} = 2^{14}$$

$$= 1000000000000000 \text{ B} = 4000 \text{ H}$$

- Số bit để đánh số hiệu trang vật lý là:  $n_2 = 14$ .

- Số bit để đánh địa chỉ vật lý là:  $64 \text{ MB} = 2^{26}$ , vậy  $n = 26$  ( $n = n_2 + n_1$ ).

**a. Xác định địa chỉ vật lý cần truy cập theo địa chỉ logic <0, 2, 20Bh>**

- Số hiệu đoạn cần truy nhập là:  $S = 0 < 4$  (số đoạn của chương trình), địa chỉ đoạn cần truy nhập là hợp lệ.

- Số hiệu trang logic cần truy nhập là:  $N_p = 2 < L_0 = 4$  (độ dài đoạn 0 – xem phần tử 0 bảng SCB), số hiệu trang logic cần truy nhập là hợp lệ.
- Địa chỉ lệch cần truy nhập trong trang là:  $d = 20BH < L = 1000H$ , địa chỉ lệch cần truy nhập trong trang là hợp lệ.
- Phần tử 0 trong SCB có  $P_s = 1$ , cho biết đoạn 0 đã được nạp và  $A_p = 400400h$  là địa chỉ đầu của bảng  $PCB_0$  trong bộ nhớ vật lý.
- Truy cập bảng  $PCB_0$ , ta thấy phần tử 2 (quản lý trang  $N_p = 2$ ) cho biết trang  $N_p = 2$  đã được nạp ( $P = 1$ ) vào trang vật lý có số hiệu  $A = 406H$ .
- Vậy địa chỉ vật lý cần truy nhập là:

$$\begin{aligned} DCVL &= A \text{ SHL } n_1 \text{ OR } d = 406H \text{ SHL } 12 \text{ OR } 20BH \\ &= 406000H \text{ OR } 20BH = 40620BH. \end{aligned}$$

***b. Xác định địa chỉ vật lý cần truy cập theo địa chỉ logic <0, 1, 7Ah>***

- Số hiệu đoạn cần truy nhập là:  $S = 0 < 4$  (số đoạn của chương trình), địa chỉ đoạn cần truy nhập là hợp lệ. Phần tử 0 trong SCB có:
- $P_s = 1 \rightarrow$  Đoạn 0 đã nạp
- $A_p = 400400h \rightarrow$  Bảng  $PCB_0$  đã bắt đầu tại địa chỉ 400400h
- $L_p = 4 \rightarrow$  Đoạn 0 có kích thước 4 trang
- Số hiệu trang logic cần truy nhập là:  $N_p = 1 < 4 \rightarrow$  Thỏa mãn. Từ mục vào 1 của bảng  $PCB_0$  có:
- $P = 1 \rightarrow$  Trang 1 của của module 1 đã nạp
- $A = 405h \rightarrow$  Trang 1 nạp vào trang vật lý  $405h < 4000h \rightarrow$  Thỏa mãn
- Địa chỉ lệch cần truy nhập trong trang là:  $d = 7Ah < L = 1000h \rightarrow$  Thỏa mãn
- $\rightarrow$  Địa chỉ vật lý cần truy nhập là:

$$DCVL = A * L + d = 405h * 12 + 7Ah = 405000h + 7Ah = 40507Ah$$

***c. Xác định địa chỉ vật lý cần truy cập theo địa chỉ logic <1, 0, 46h>***

- Số hiệu đoạn cần truy nhập là:  $S = 1 < 4$  (số đoạn của chương trình), địa chỉ đoạn cần truy nhập là hợp lệ.
- Số hiệu trang logic cần truy nhập là:  $N_p = 0 < L_1 = 4$  (độ dài đoạn 0 – xem phần tử 0 bảng SCB), số hiệu trang logic cần truy nhập là hợp lệ.
- Địa chỉ lệch cần truy nhập trong trang là:  $d = 46H < L = 1000H$ , địa chỉ lệch cần truy nhập trong trang là hợp lệ.

- Phần tử 1 trong SCB có  $P_s = 0$ , cho biết đoạn 1 chưa được nạp vào bộ nhớ vật lý. Hệ thống cần nạp đoạn hoặc đổi đoạn.

***d. Xác định địa chỉ vật lý cần truy cập theo địa chỉ logic <1, 1, 1001h>***

- Số hiệu đoạn cần truy nhập là:  $S = 1 < 4$  (số đoạn của chương trình), địa chỉ đoạn cần truy nhập là hợp lệ.
- Phần tử 1 trong bảng PCB có:  $P_s = 0 \rightarrow$  Đoạn 1 không được nạp  $\rightarrow$  Hệ thống cần nạp đoạn hoặc đổi đoạn

***e. Xác định địa chỉ vật lý cần truy cập theo địa chỉ logic <2, 1, 1000h>***

- Số hiệu đoạn cần truy nhập là:  $S = 2 < 4$  (số đoạn của chương trình), địa chỉ đoạn cần truy nhập là hợp lệ.
- Số hiệu trang logic cần truy nhập là:  $N_p = 1 < L_2 = 4$  (độ dài đoạn 0 – xem phần tử 0 bảng SCB), số hiệu trang logic cần truy nhập là hợp lệ.
- Địa chỉ lệch cần truy nhập trong trang là:  $d = 1000H \geq L = 1000H$ , địa chỉ lệch cần truy nhập trong trang là không hợp lệ, do vậy tiến trình phải dừng.

***f. Xác định địa chỉ vật lý cần truy cập theo địa chỉ logic <2,3, 200h>***

- Số hiệu đoạn cần truy nhập là:  $S = 2 < 4$  (số đoạn của chương trình), địa chỉ đoạn cần truy nhập là hợp lệ.
  - Phần tử 2 của bảng PCB có:  $P_s = 1 \rightarrow$  Đoạn 2 đã được nạp
  - $A_p = 400C00h \rightarrow$  Bảng PCB<sub>2</sub> bắt đầu tại địa chỉ 400C00h
  - $L_p = 3 \rightarrow$  Đoạn 2 có kích thước 3 trang
  - $N_p = 3 > 3 \rightarrow$  Không thỏa mãn.
- $\rightarrow$  Địa chỉ truy nhập không hợp lệ

***g. Xác định địa chỉ vật lý cần truy cập theo địa chỉ logic <4, 1, 57h>***

- Số hiệu đoạn cần truy nhập là:  $S = 4 < 4$  (số đoạn của chương trình)  $\rightarrow$  Địa chỉ đoạn cần truy nhập là không hợp lệ.

## Chương 5 : Kỹ thuật đường ống và RISC

**Câu 5.1.** Trình bày kỹ thuật đường ống (pipeline) để đưa ra công thức tính thời gian thực hiện n lệnh, với ví dụ một đường ống gồm 6 công đoạn: Nhận lệnh, giải mã lệnh, tính địa chỉ toán hạng, nhận toán hạng, thi hành lệnh, ghi toán hạng.

### **Bài làm**

- Kỹ thuật đường ống gồm 6 công đoạn:
  - + Nhận lệnh (IF: Instruction Fetch).
  - + Giải mã lệnh (ID: Instruction decode).
  - + Tính địa chỉ toán hạng (Operand calculation - OC).
  - + Nhận toán hạng (Operands Fetch – OF).
  - + Thực hiện lệnh (Instruction execution – IE).
  - + Ghi kết quả (WO – Write operation).
- Công thức tính thời gian thực hiện lệnh n:

$$T_s = n * m * T_c$$

$$T_p = m * T_c + (n-1) * T_c$$

Trong đó: m là số công đoạn

n là số câu lệnh

$T_c$  là thời gian thực hiện 1 công đoạn

$T_p$ : thời gian thực hiện n câu lệnh

**Câu 5.2.** Trình bày các khó khăn và hướng khắc phục với kỹ thuật đường ống đơn vị lệnh

### **Bài làm**

Khó khăn: có 3 xung đột:

- Xung đột về cấu trúc: Tại một thời điểm xử lý n lệnh, sẽ có thể xảy ra trường hợp có một số công đoạn của lệnh khác nhau cần dùng chung 1 tài nguyên phần cứng
- Xung đột về mặt dữ liệu: Ví dụ thực hiện 2 lệnh sau.

ADDR1,R2,R3;R3=R1+R2

ADD R3,R4,R5 ;R5=R3+R4

+ Ta thấy lệnh thứ 2 muốn thực hiện thì lệnh thứ nhất phải hoàn thành xong, xong 2 lệnh nằm sát cạnh nhau nên lệnh 2 ở công đoạn nhận toán hạng cần R3 thì lệnh một mới ở công đoạn xử lý lệnh. Như vậy lệnh 2 phải chờ, hiệu quả thực hiện pipeline giảm đi

- Xung đột về mặt điều khiển: Khi gặp lệnh rẽ nhánh có điều kiện (IF\_THEN... ELSE) thì khi chưa kịp giải mã lệnh, không biết lệnh rẽ theo nhánh nào. Do vậy phải chờ.

Khắc phục:

- Siêu đường ống: Tăng số công đoạn thực hiện cho mỗi lệnh (8 đến 10 công đoạn) Tăng số mạch phân cứng trong mỗi đường ống.
- Siêu hướng: Tăng số đường ống lên.
- Ví dụ Pentium có:
  - + 2 đường ống
  - + Dự đoán rẽ nhánh
- Cho phép hệ thống được nạp lệnh tiếp theo cho cả 2 nhánh khi giải mã xong điều kiện nhả, sẽ chỉ lấy kết quả thì hành, của lệnh thỏa mãn điều kiện, hoặc hệ thống phải có khả năng dự đoán rẽ nhánh chính xác, tuy nhiên xử lý vấn đề xung đột này không phải đơn giản

**Câu 5.3.** Giả sử một lệnh được chia làm 5 công đoạn: nhận lệnh, giải mã lệnh, nhận toán hạng, xử lý, cất kết quả. Mỗi công đoạn thực hiện trong  $5 \cdot 10^{-9}$  giây. Một lệnh nếu thực hiện tuần tự hết  $22 \cdot 10^{-9}$  giây. So sánh thời gian thực hiện 40 lệnh giữa kỹ thuật pipeline và kỹ thuật tuần tự.

**Gợi ý**

- Áp dụng công thức  $T_p = m \cdot T_c + (n-1) \cdot T_c$ , trong đó:
  - $T_p$ : thời gian thực hiện n lệnh theo kỹ thuật pipeline.
  - $T_c$ : thời gian thực hiện một công đoạn lớn nhất trong m công đoạn theo kỹ thuật pipeline.
  - n: số lệnh.
  - M: số công đoạn trong một lệnh.
- Và công thức  $T_t = n \cdot T$ , trong đó:
  - T: thời gian thực hiện 1 lệnh khi thực hiện tuần tự.

### ***Bài làm***

Ta có:  $T_t = n * T = 40 * 22 * 10^{-9}$  (giây)

$$T_p = m * T_c + (n-1) * T_c = 5 * (5 * 10^{-9}) + (40 - 1) * (5 * 10^{-9})$$
$$= 44 * (5 * 10^{-9}) \text{ (giây)}$$

Vậy thực hiện 40 lệnh với kỹ thuật pipeline nhanh hơn 40 lệnh với kỹ thuật tuần tự là :  $T_t/T_p = 40 * 22 * 10^{-9} / (44 * 5 * 10^{-9}) = 8 * 22 / 44 = 4$  lần.

### **Câu 5.4.**

Giả sử một lệnh được chia làm 6 công đoạn: nhận lệnh, giải mã lệnh, tính địa chỉ toán hạng, nhận toán hạng, xử lý, cất kết quả. Mỗi công đoạn thực hiện trong thời gian  $5 * 10^{-9}$  giây. Một lệnh nếu thực hiện tuần tự hết  $22 * 10^{-9}$  giây. So sánh thời gian thực hiện 50 lệnh giữa kỹ thuật pipeline và kỹ thuật tuần tự.

### ***Bài làm***

Ta có:  $T_t = n * T = 50 * 22 * 10^{-9}$  (giây)

$$T_p = m * T_c + (n-1) * T_c = 6 * (5 * 10^{-9}) + (50 - 1) * (5 * 10^{-9})$$
$$= 55 * (5 * 10^{-9}) \text{ (giây)}$$

Vậy thực hiện 50 lệnh với kỹ thuật pipeline nhanh hơn 40 lệnh với kỹ thuật tuần tự là :  $T_t/T_p = 50 * 22 * 10^{-9} / (55 * 5 * 10^{-9}) = 4$  lần.

### **Câu 5.5.**

Trình bày tóm tắt các đặc điểm trong kỹ thuật cài đặt RISC.

### ***Bài làm***

Đặc điểm trong kỹ thuật cài đặt RISC

- Kích thước lệnh cố định 4 bytes.
- Sử dụng lệnh load/store.
- Nhiều thanh ghi: 32 thanh ghi 32 bit cho số nguyên, 32 thanh ghi 64 bit dành cho số thực.
- Đánh địa chỉ đơn giản.
- Tập lệnh nhỏ.
- Hỗ trợ ít kiểu dữ liệu.
- Các lệnh thực hiện nhanh chỉ cần 1 xung clock.
- Có bus riêng cho dữ liệu và mã lệnh

### **Câu 5.6.**

Cho biết ý nghĩa của cụm từ RISC (Reduced Instruction Set Computer)

**Bài làm**

RISC là viết tắt của Bộ hướng dẫn giảm. Nó được thiết kế để giảm thời gian thực hiện bằng cách đơn giản hóa tập lệnh. Nó sử dụng các hướng dẫn tối ưu hóa cao. Bộ xử lý dựa trên RISC thường được sử dụng cho các thiết bị di động như điện thoại di động và máy tính bảng vì chúng hiệu quả hơn.

**5. Trình bày ngắn gọn sự khác biệt giữa kiến trúc CISC và kiến trúc RISC.**

**Bài làm**

	RISC	CISC
Viết tắt	Reduced Instruction Set Computer	Complex Instruction Set Computer
Định nghĩa	Bộ xử lý RISC có một bộ hướng dẫn nhỏ hơn với một vài nút địa chỉ	Bộ xử lý CISC có một bộ hướng dẫn lớn hơn với nhiều nút địa chỉ.
Đơn vị bộ nhớ	Nó không có đơn vị bộ nhớ và sử dụng một phần cứng riêng để thực hiện các hướng dẫn.	Nó có một đơn vị bộ nhớ để thực hiện các hướng dẫn phức tạp.
Chương trình	Nó có một đơn vị lập trình cứng.	Nó có một đơn vị lập trình vi mô.
Thiết kế	Nó là một thiết kế phức tạp	Nó là một thiết kế dễ dàng hơn
Tính toán	Các tính toán nhanh hơn và chính xác.	Các tính toán là chậm và chính xác
Giải mã	Giải mã hướng dẫn rất đơn giản.	Giải mã các hướng dẫn là phức tạp.
Thời gian	Thời gian thực hiện là rất ít.	Thời gian thực hiện rất cao
Bộ nhớ ngoài	Nó không yêu cầu bộ nhớ ngoài để tính toán.	Nó đòi hỏi bộ nhớ ngoài để tính toán
Đường ống	Pipelining không hoạt động chính xác.	Đường ống không hoạt động chính xác.
Ngừng	Sự đình trệ chủ yếu là giảm trong bộ xử lý.	Các bộ xử lý thường bị đình trệ.
Mở rộng mã	Mở rộng mã có thể là một vấn đề.	Mở rộng mã không phải là một vấn đề.

Không gian đĩa	Không gian được lưu.	Không gian bị lãng phí
Các ứng dụng	Được sử dụng trong các ứng dụng cao cấp như xử lý video, viễn thông và xử lý hình ảnh.	Được sử dụng trong các ứng dụng cấp thấp như hệ thống bảo mật, tự động hóa gia đình...