

# Chương 1 - Tổng quan về Kiến trúc máy tính

## KHÁI NIỆM VỀ KIẾN TRÚC VÀ TỔ CHỨC MÁY TÍNH

- Kiến trúc máy tính (Computer Architecture) và Tổ chức máy tính (Computer Organization) là hai trong số các khái niệm cơ bản của ngành Công nghệ máy tính (Computer Engineering).
- Có thể nói kiến trúc máy tính là bức tranh toàn cảnh về hệ thống máy tính, còn tổ chức máy tính là bức tranh cụ thể về các thành phần phần cứng của hệ thống máy tính.
- Kiến trúc máy tính là khoa học về việc lựa chọn và kết nối các thành phần phần cứng để tạo ra các máy tính đạt được các yêu cầu về chức năng (functionality), hiệu năng (performance) và giá thành (cost).

# Chương 1 - Tổng quan về Kiến trúc máy tính

## KHÁI NIỆM VỀ KIẾN TRÚC VÀ TỔ CHỨC MÁY TÍNH

- Kiến trúc máy tính được cấu thành từ 3 thành phần con: (i) Kiến trúc tập lệnh (Instruction Set Architecture), (ii) Vi kiến trúc (Micro Architecture) và Thiết kế hệ thống (System Design).
  - **Kiến trúc tập lệnh** là hình ảnh của một hệ thống máy tính ở mức ngôn ngữ máy. Kiến trúc tập lệnh bao gồm các thành phần: tập lệnh, các chế độ địa chỉ, các thanh ghi, **khuôn dạng địa chỉ và dữ liệu**.
  - **Vi kiến trúc** là mô tả mức thấp về các thành phần của hệ thống máy tính, phối ghép và việc trao đổi thông tin giữa chúng. Vi kiến trúc giúp trả lời hai câu hỏi (1) Các thành phần phần cứng của máy tính kết nối với nhau như thế nào? và (2) Các thành phần phần cứng của máy tính tương tác với nhau như thế nào để thực thi tập lệnh?
  - **Thiết kế hệ thống**: bao gồm tất cả các thành phần phần cứng của hệ thống máy tính, bao gồm: Hệ thống phối ghép (các bus và các chuyên mạch), Hệ thống bộ nhớ, Các cơ chế giảm tải cho CPU (như truy nhập trực tiếp bộ nhớ) và Các vấn đề khác (như đa xử lý và xử lý song song).

# Chương 1 - Tổng quan về Kiến trúc máy tính

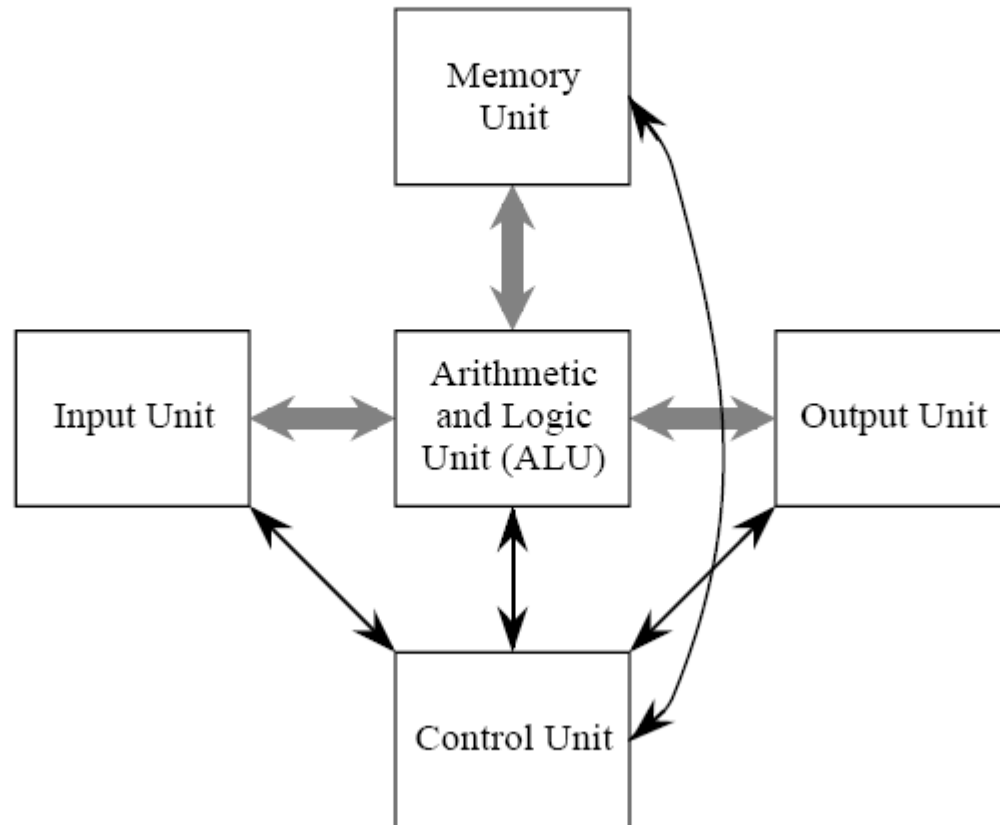
## Cấu trúc chung của máy vi tính

- Máy tính bao gồm các khối chính là: đơn vị xử lý trung tâm CPU, bộ nhớ chính và các thiết bị ngoại vi. Các khối này liên kết với nhau qua bus hệ thống và các mạch ghép nối (giao diện) vào/ra.
- ✓ Cách thức tổ chức phổ biến của một máy tính hiện nay theo kiểu Von-Neumann là các hoạt động được xảy ra một cách nối tiếp nhau và số liệu cũng như lệnh đều được lưu trữ trong cùng một bộ nhớ.

# Chương 1 - Tổng quan về Kiến trúc máy tính

## Kiến trúc máy tính Von Neumann

- The von Neumann model consists of five major components: (1) input unit; (2) output unit; (3) arithmetic logic unit; (4) memory unit; (5) control unit.



# Chương 1 - Tổng quan về Kiến trúc máy tính

## □ Các đặc điểm của kiến trúc von-Neumann

- Kiến trúc von-Neumann dựa trên 3 khái niệm cơ sở:

(1) Lệnh và dữ liệu được lưu trữ trong bộ nhớ đọc ghi chia sẻ - một bộ nhớ duy nhất được sử dụng để lưu trữ cả lệnh và dữ liệu,

(2) Bộ nhớ được đánh địa chỉ theo vùng, không phụ thuộc vào nội dung nó lưu trữ và

(3) Các lệnh của một chương trình được thực hiện tuần tự. **Quá trình thực hiện lệnh** được chia thành 3 giai đoạn (stages) chính:

(i) CPU đọc (fetch) lệnh từ bộ nhớ,

(ii) CPU giải mã và thực hiện lệnh;

- nếu lệnh yêu cầu dữ liệu, CPU đọc dữ liệu từ bộ nhớ; và

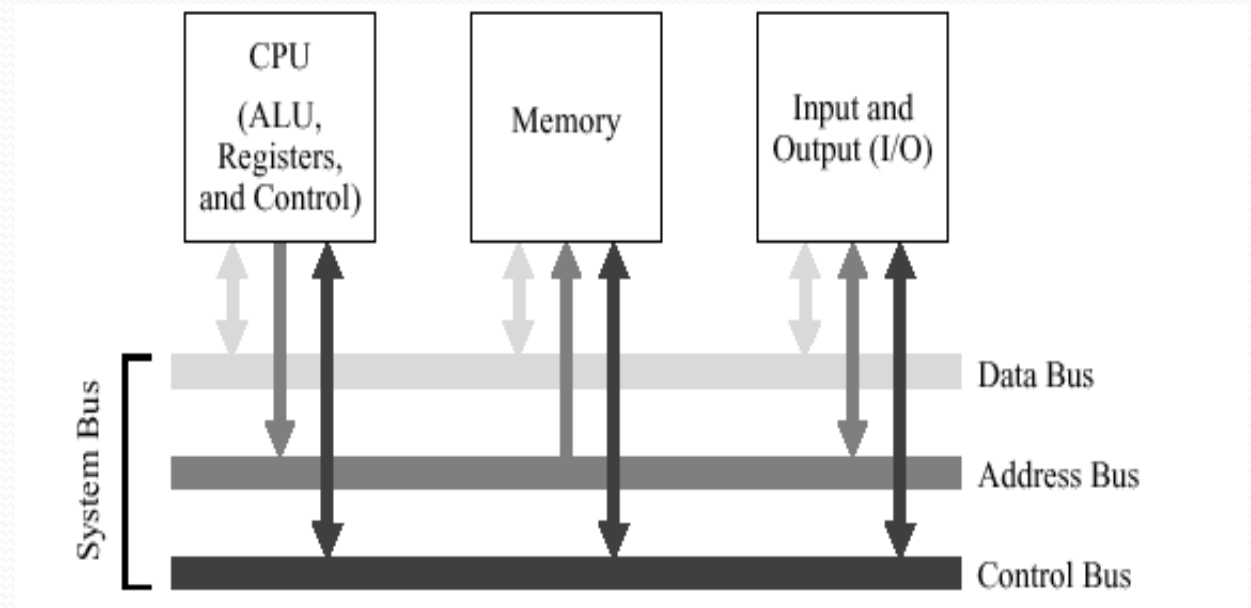
(iii) CPU ghi kết quả thực hiện lệnh vào bộ nhớ (nếu có)

# Chương 1 - Tổng quan về Kiến trúc máy tính

- *Khối xử lý trung tâm CPU* (central processing unit): **xử lý và thực hiện các lệnh**. Bên trong CPU bao gồm các mạch điều khiển logic, mạch tính số học và logic, và một bộ nhớ nhỏ có tốc độ truy nhập cao là các thanh ghi. Nếu CPU được xây dựng trên một hoặc vài vi mạch, thường được đóng trong 1 chip, thì được gọi là bộ **vi xử lý** (microprocessor).
- *Bộ nhớ*: lưu trữ các lệnh và dữ liệu dùng cho bộ VXL. Bao gồm 2 loại: bộ nhớ trong (được tạo bởi các vi mạch nhớ bán dẫn có tốc độ truy cập cao) và bộ nhớ ngoài (được tạo bởi các môi trường nhớ khác như đĩa từ, đĩa quang).
- *Các thiết bị ngoại vi*: gồm các thiết bị vào ra (I/O) dùng để nhập hoặc xuất dữ liệu. Các thiết bị ngoại vi liên hệ với CPU qua các **mạch ghép nối** vào ra (I/O interface). Mạch này cho phép nối 2 bộ phận độc lập nhằm làm cho chúng có thể tương hợp và thông tin được với nhau. Các mạch ghép nối cụ thể giữa Bus hệ thống với thế giới bên ngoài thường được gọi là **cổng**.

# Chương 1 - Tổng quan về Kiến trúc máy tính

- Để duy trì hoạt động chính xác và đồng bộ hóa CPU cùng các bộ phận liên quan, máy tính còn phải có một mạch tạo các xung điện có tần số rất chính xác và ổn định gọi là *đồng hồ hệ thống* (system control). Tần số này quyết định tốc độ hoạt động của máy tính nhanh hay chậm.
- *Bus hệ thống*: là một tập hợp các đường dây qua đó CPU có thể giao tiếp với các bộ phận khác.





# Chương 1 - Tổng quan về Kiến trúc máy tính

- **Bus địa chỉ** (*Address Bus*) thường có từ 16, 20, 24 đến 32 đường dây song song chuyển tải thông tin của các bit địa chỉ. Khi đọc/ghi bộ nhớ, CPU sẽ đưa ra trên bus này địa chỉ của ô nhớ liên quan. Khả năng phân biệt địa chỉ (số lượng địa chỉ cho ô nhớ mà CPU có khả năng phân biệt được) phụ thuộc vào số bit của bus địa chỉ. Ví dụ, nếu 1 CPU có số đường dây địa chỉ là  $N = 16$  thì nó có khả năng địa chỉ hóa được  $2^N = 64$  KB. Khi đọc/ghi với cổng vào/ra CPU cũng đưa ra trên bus địa chỉ các bit địa chỉ tương ứng với cổng.
- **Bus dữ liệu** (*Data Bus*): Bus dữ liệu có từ 8, 16, 32 đến 64 đường dây tùy theo các bộ VXL cụ thể. Số lượng đường dây này quyết định số bit dữ liệu mà CPU có khả năng xử lý cùng một lúc. Bus dữ liệu là bus 2 chiều, nghĩa là dữ liệu có thể được truyền đi từ CPU (ra) hoặc truyền đến CPU (vào).



# Chương 1 - Tổng quan về Kiến trúc máy tính

- *Bus điều khiển (Control bus)*: thường gồm hàng chục đường dây tín hiệu khác nhau.
  - Mỗi tín hiệu điều khiển có 1 chiều nhất định.
  - Vì khi hoạt động CPU đưa tín hiệu điều khiển tới các khối khác nhau trong hệ thống, đồng thời nó cũng nhận các tín hiệu điều khiển từ các khối đó để phối hợp hoạt động của toàn hệ nên các tín hiệu này trên hình vẽ được thể hiện bởi các đường có **mũi tên 2 chiều**, điều đó không phải để chỉ tính 2 chiều của 1 tín hiệu mà là **tính 2 chiều của cả nhóm tín hiệu**.

# Chương 1 - Tổng quan về Kiến trúc máy tính

## Nhiệm vụ của CPU:

- **Nhận lệnh** (Fetch Instruction): đọc lệnh từ bộ nhớ
- **Giải mã lệnh** (Decode Instruction): xác định thao tác mà lệnh yêu cầu.
- **Nhận dữ liệu** (Fetch Data): nhận dữ liệu từ bộ nhớ hoặc các cổng vào/ra.
- **Xử lý dữ liệu** (Process Data): thực hiện phép toán số học hay phép toán logic với các dữ liệu).
- **Ghi dữ liệu** (Write Data): ghi dữ liệu ra bộ nhớ hay cổng vào/ra.

# Chương 1 - Tổng quan về Kiến trúc máy tính

## Các thành phần cơ bản của CPU:

- Đơn vị điều khiển (Control Unit - CU).
- Đơn vị số học và logic (Arithmetic and Logic Unit - ALU).
- Tập thanh ghi (Register File - RF).
- Đơn vị ghép nối Bus (Bus Interface Unit - BIU).
- Bus bên trong (Internal Bus).

# Chương 1 - Tổng quan về Kiến trúc máy tính

Chức năng của đơn vị điều khiển (Control Unit - CU):

- Nhận lệnh từ bộ nhớ đưa vào thanh ghi lệnh.
- Tăng nội dung thanh ghi PC để trở sang lệnh kế tiếp.
- Giải mã lệnh để xác định thao tác mà lệnh yêu cầu.
- Phát ra các tín hiệu điều khiển thực hiện lệnh.
- Nhận các tín hiệu yêu cầu từ bus hệ thống và đáp ứng các yêu cầu đó.

# Chương 1 - Tổng quan về Kiến trúc máy tính

Các tín hiệu đưa đến đơn vị điều khiển:

- Clock: tín hiệu nhịp từ mạch tạo dao động bên ngoài.
- Mã lệnh từ thanh ghi lệnh đưa đến để giải mã.
- Các cờ từ thanh ghi cờ cho biết trạng thái của CPU.
- Các tín hiệu yêu cầu từ bus điều khiển.

# Chương 1 - Tổng quan về Kiến trúc máy tính

Các tín hiệu phát ra từ đơn vị điều khiển:

- Các tín hiệu điều khiển bên trong CPU
  - Điều khiển các thanh ghi.
  - Điều khiển ALU.
- Các tín hiệu điều khiển bên ngoài CPU.
  - Điều khiển bộ nhớ.
  - Điều khiển các modul vào/ra.



# Chương 1 - Tổng quan về Kiến trúc máy tính

## Một số khái niệm liên quan khác:

- **BIOS** (*Basic Input/Output System*) là một tập hợp chương trình sơ cấp để hướng dẫn các hoạt động cơ bản của máy tính, bao gồm cả thủ tục khởi động và việc quản lý các tín hiệu nhập vào từ bàn phím. BIOS được nạp cố định trong một chip nhớ chỉ đọc (ROM) lắp trên board mẹ.
- **Truy cập trực tiếp bộ nhớ** (*DMA - Direct Memory Access*), bộ điều khiển (controller) DMA là một mạch điện tử tích hợp, có trang bị các chức năng vi xử lý, được lắp cố định trên board mẹ, phù hợp với một kiểu VXL nhất định. Chip DMA cho phép máy tính có thể di chuyển dữ liệu từ các ổ đĩa hoặc các ngoại vi khác trực tiếp vào bộ nhớ máy tính mà không ảnh hưởng đến công việc của bộ vi xử lý chính nên làm tăng tốc độ của máy tính lên rất nhiều.

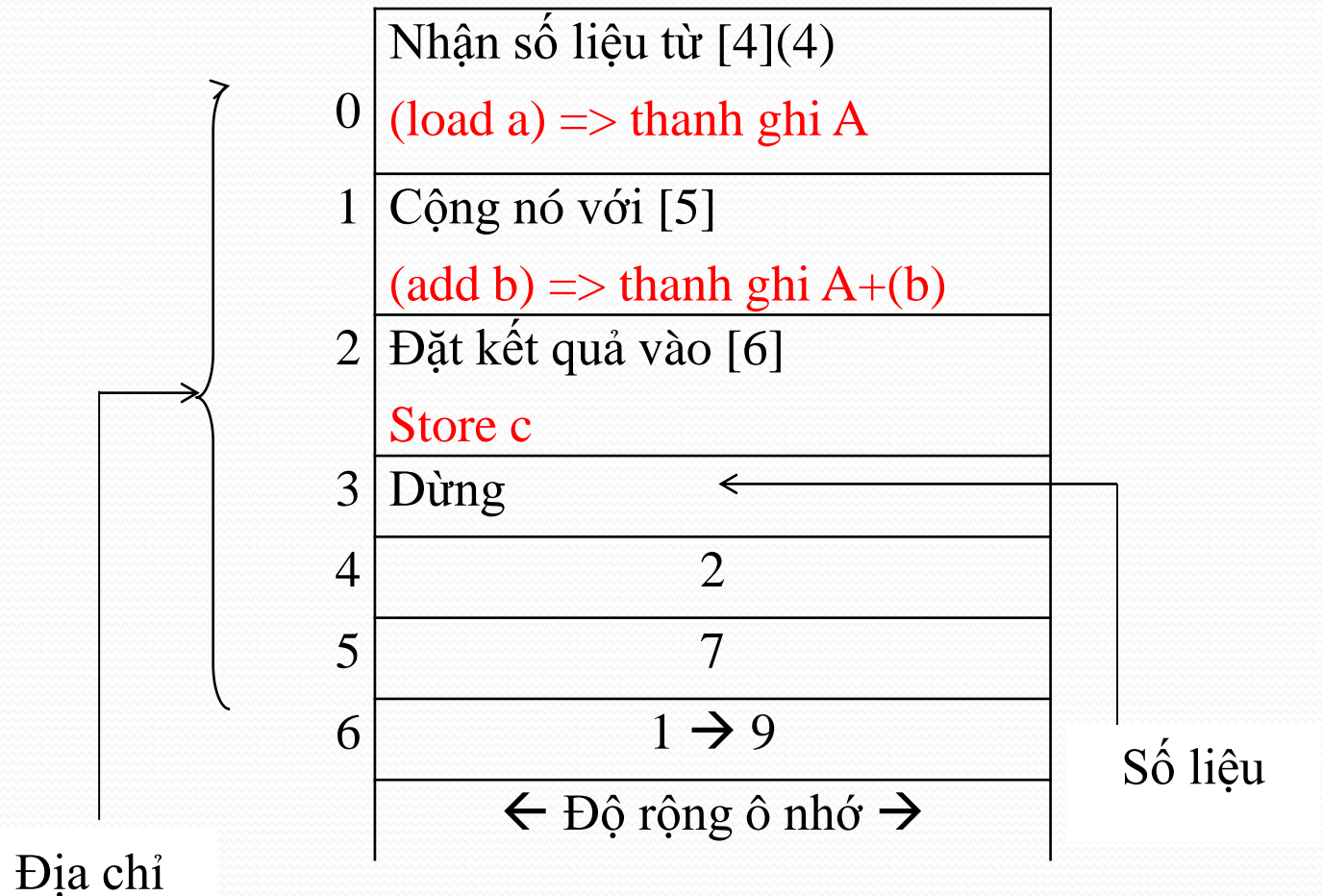
# Chương 1 - Tổng quan về Kiến trúc máy tính

**Quá trình hoạt động của máy tính:** Với một máy tính hoạt động theo kiểu tuần tự ta có thể dùng một ví dụ để minh họa cách thức mà CPU thực thi các lệnh. Cách tổ chức theo nguyên tắc xây dựng *các hoạt động xảy ra nối tiếp nhau*.

- Tại một vùng nào đó trong bộ nhớ là một chương trình được lưu trữ trước khi chạy. Chương trình trong một máy tính bao gồm một tập hợp các lệnh được sắp xếp theo một trật tự nào đó.
- Khi chạy chương trình, CPU sẽ đọc lần lượt từng lệnh một từ bộ nhớ và thực hiện nó.
- Các số liệu ban đầu thuộc về chương trình cũng như số liệu tính toán trung gian cũng được lưu trữ ngay trong bộ nhớ như các lệnh (chỉ có khác là các lệnh được lưu trữ ở 1 vùng, còn các số liệu được lưu trữ ở 1 vùng khác).
- Một máy tính được tổ chức như vậy gọi là máy tính kiểu Von Neumann.

# Chương 1 - Tổng quan về Kiến trúc máy tính

Quá trình hoạt động của máy tính:



# Chương 1 - Tổng quan về Kiến trúc máy tính

## Quá trình hoạt động của máy tính:

Trong thực tế, CPU thực thi từng lệnh theo 1 chuỗi thao tác như sau:

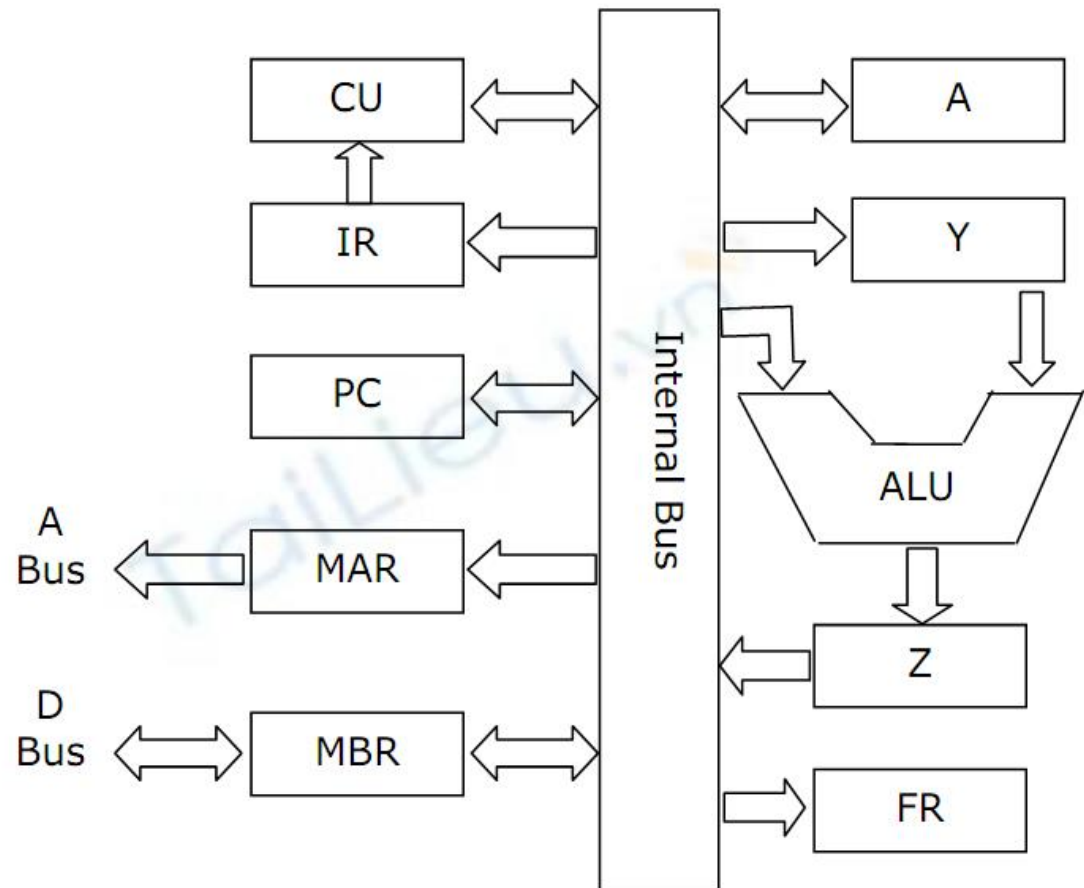
- (1). Lấy lệnh kế tiếp từ bộ nhớ vào thanh ghi lệnh IR.
- (2). Chuyển nội dung bộ đếm chương trình (trở) tới địa chỉ lệnh tiếp theo (PC hoặc IP).
- (3). Xác định loại lệnh vừa lấy.
- (4). Nếu lệnh sử dụng dữ liệu trong bộ nhớ, xác định nơi chứa dữ liệu
- (5). Tìm-nạp dữ liệu vào các thanh ghi trong CPU (nếu có)
- (6). Thực thi lệnh
- (7). Lưu các kết quả vào nơi thích hợp.
- (8). Trở lại bước 1 để thực thi lệnh kế tiếp.

Đây là một chu kỳ *lấy lệnh – giải mã lệnh – thực thi lệnh* ở hầu hết các máy tính.

# Chương 1 - Tổng quan về Kiến trúc máy tính

**Minh họa quá trình hoạt động của một bộ xử lý đơn giản:**

*Sơ đồ khối tổng quát của CPU*



# Chương 1 - Tổng quan về Kiến trúc máy tính

Minh họa quá trình hoạt động của một bộ xử lý đơn giản:

*Sơ đồ khối tổng quát của CPU (các thành phần):*

- Bộ điều khiển (Control Unit – CU): giải mã lệnh
- Bộ tính toán số học và logic (Arithmetic and Logic Unit)
- Bus trong CPU (CPU Internal Bus)
- Các thanh ghi của CPU:
  - Thanh ghi tích lũy A (Accumulator), AC, ACC
  - Bộ đếm chương trình PC (Program Counter) hay IP
  - Thanh ghi lệnh IR (Instruction Register)
  - Thanh ghi địa chỉ bộ nhớ MAR (Memory Address Register)
  - Thanh ghi đệm dữ liệu MBR (Memory Buffer Register)
  - Các thanh ghi tạm thời Y và Z
  - Thanh ghi cờ FR (Flag Register): trạng thái của VXL



# Chương 1 - Tổng quan về Kiến trúc máy tính

## Minh họa quá trình hoạt động của một bộ xử lý đơn giản:

*Chu trình xử lý lệnh:* Như đã trình bày ở trên, nhiệm vụ chủ yếu của CPU là đọc lệnh từ bộ nhớ, giải mã và thực hiện lệnh của chương trình. Khoảng thời gian để CPU thực hiện xong một lệnh kể từ khi CPU cấp phát tín hiệu địa chỉ ô nhớ chứa lệnh đến khi nó hoàn tất việc thực hiện lệnh được gọi là chu kỳ lệnh (Instruction Cycle). Mỗi chu kỳ lệnh của CPU được mô tả theo các bước sau:

- 1) Khi một chương trình được kích hoạt, hệ điều hành (OS - Operating System) nạp mã chương trình vào bộ nhớ trong;
- 2) **Địa chỉ của ô nhớ** chứa lệnh đầu tiên của chương trình được nạp vào bộ đếm chương trình PC;
- 3) Địa chỉ ô nhớ chứa lệnh từ PC được chuyển đến bus địa chỉ thông qua thanh ghi MAR;
- 4) Bus địa chỉ chuyển địa chỉ ô nhớ đến **đơn vị quản lý bộ nhớ** (MMU - Memory Management Unit);
- 5) MMU chọn ra ô nhớ và thực hiện lệnh đọc nội dung ô nhớ;

# Chương 1 - Tổng quan về Kiến trúc máy tính

Minh họa quá trình hoạt động của một bộ xử lý đơn giản:

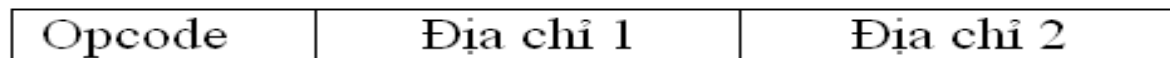
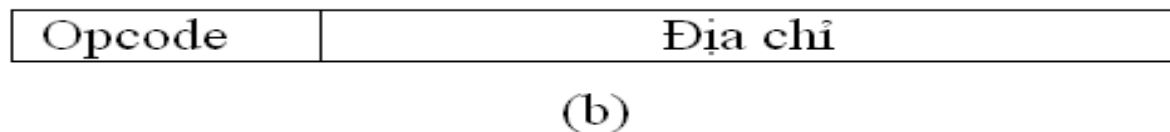
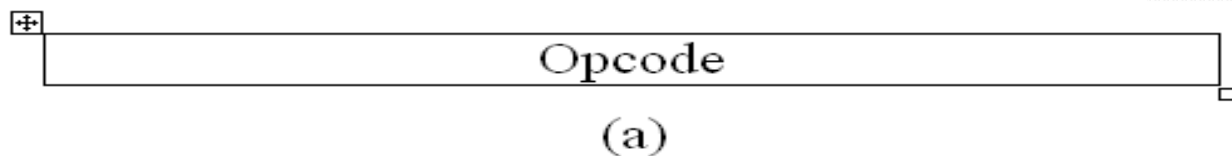
*Chu trình xử lý lệnh:*

- 6) Lệnh (chứa trong ô nhớ) được chuyển ra bus dữ liệu và tiếp theo được chuyển tiếp đến thanh ghi MBR;
- 7) MBR chuyển lệnh đến thanh ghi lệnh IR; <đọc lệnh xong>
- 8) IR chuyển lệnh vào bộ điều khiển CU;
- 9) CU giải mã lệnh và sinh các tín hiệu điều khiển cần thiết, yêu cầu các bộ phận chức năng của CPU, như ALU thực hiện lệnh;
- 10) Giá trị địa chỉ trong bộ đếm PC được tăng lên 1 đơn vị lệnh và nó trở đến địa chỉ của ô nhớ chứa lệnh tiếp theo;
- 11) Các bước từ 3-9 được lặp lại với tất cả các lệnh của chương trình.

# Chương 1 - Tổng quan về Kiến trúc máy tính

Consider the following pseudo assembly code for computing  $c = a + b$ . Assume that  $a$ ,  $b$ , and  $c$  are assigned to consecutive memory “words” (memory is generally addressed byte by byte and assume that a word is 4 bytes) and address for “ $a$ ” is  $0x0000ec00$ . Also, we have  $a = 22$ ,  $b = 158$ , and  $c = 0$  at the starting time. Assume that the first instruction of the code is stored in  $0x0000b128$ . Also, each instruction has the opcode in the first byte (most significant byte) and the remaining 3 bytes specify the corresponding address. The opcode for store is 1, load is 2, and add is 3.

$0x0000b128$  load  $a$   
 $0x0000b12c$  add  $b$   
 $0x0000b130$  store  $c$



- a. Show the memory addresses and contents for all the instructions and data involved. Use the format as follows to express your answer (but the following is not the answer). For all data, use hexadecimal representation.

addresses	contents
$0x00002104$	$0x00000001$
$0x00002108$	$0x00000002$

.....

.....

# Chương 1 - Tổng quan về Kiến trúc máy tính

Consider the following pseudo assembly code for computing  $c = a + b$ . Assume that  $a$ ,  $b$ , and  $c$  are assigned to consecutive memory “words” (memory is generally addressed byte by byte and assume that a word is 4 bytes) and address for “ $a$ ” is  $0x0000ec00$ . Also, we have  $a = 22$ ,  $b = 158$ , and  $c = 0$  at the starting time. Assume that the first instruction of the code is stored in  $0x0000b128$ . Also, each instruction has the opcode in the first byte (most significant byte) and the remaining 3 bytes specify the corresponding address. The opcode for store is 1, load is 2, and add is 3.

```
0x0000b128 load a
0x0000b12c add b
0x0000b130 store c
```

- b. Write the micro instructions for the code segment. Assume that current PC (program counter) is  $0x00001018$ . For each micro-instruction, also indicate the data that is transferred (if it is a memory access). For all data, use the hexadecimal representation. The following are the first two micro-instructions and the data transferred. Complete the rest.

Micro-instructions	data
PC $\rightarrow$ MAR	$0x0000b128$
M $\rightarrow$ MBR	$0x0200ec00$