



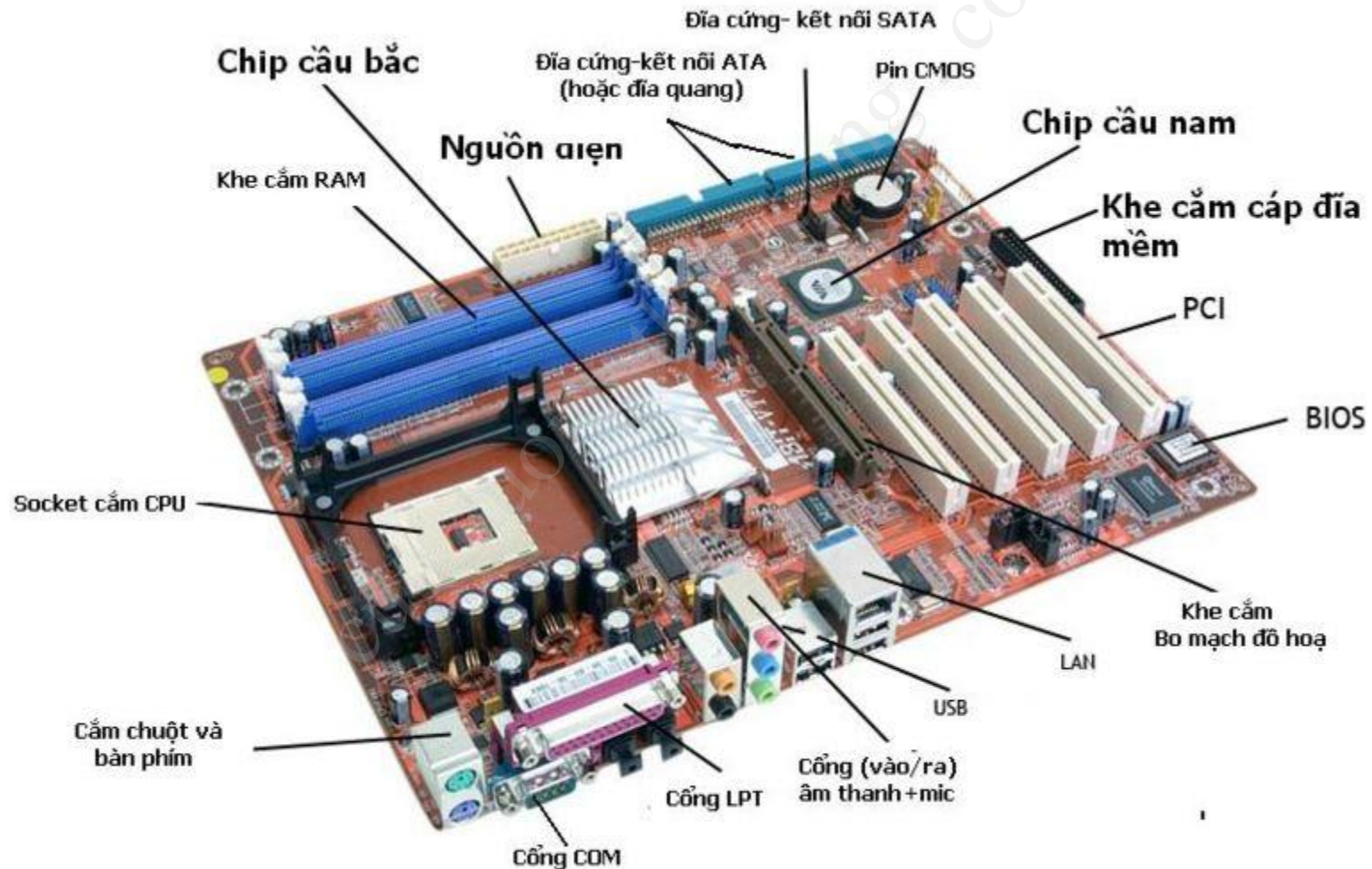
Chương 2

BỘ XỬ LÝ

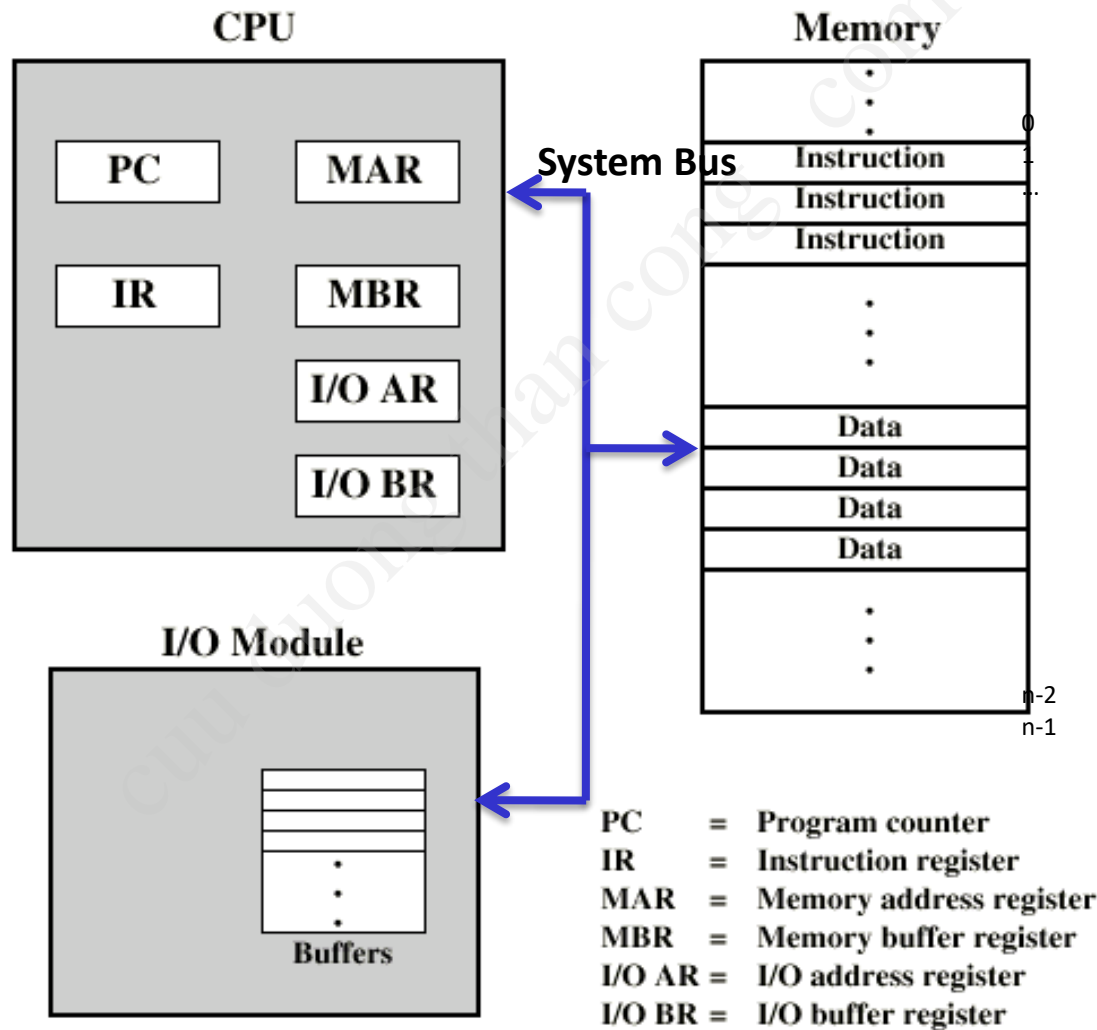
cuu duong than cong . com

1. Các thành phần chức năng của CPU.
2. Ngắt.
3. Thanh ghi.
4. Khối điều khiển.
5. Khối luận lý và số học
6. Bus hệ thống.
7. Lệnh máy tính.

Các thành phần chức năng của CPU

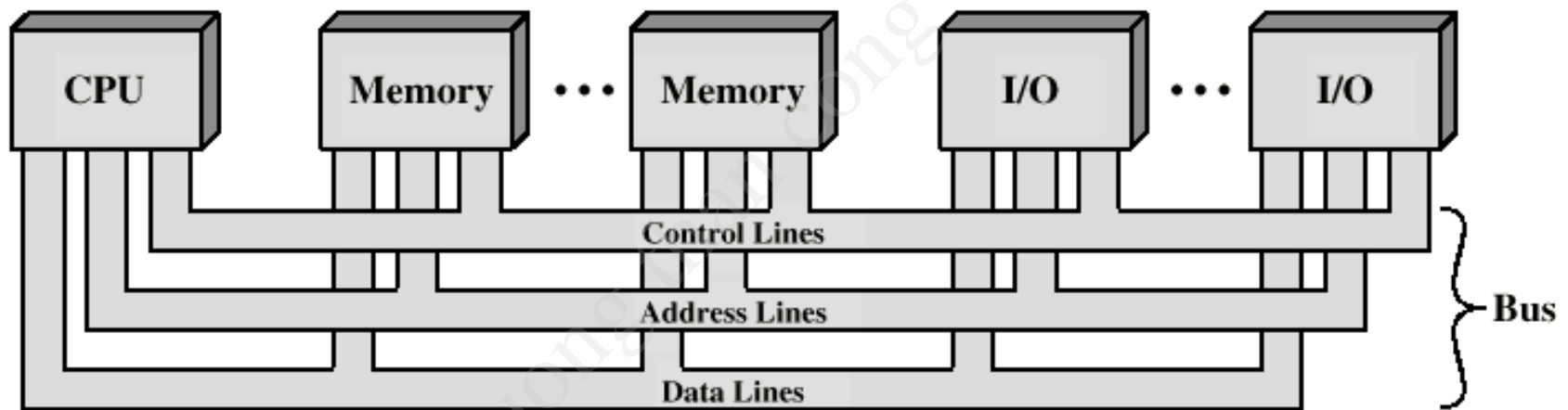


Các thành phần chức năng của CPU



Các thành phần chức năng của CPU

❑ Hệ thống Bus



❖ Data bus:

- Chuyển tải dữ liệu, lệnh.
- Độ rộng của Bus tương ứng khả năng chuyển tải dữ liệu của CPU: 8, 16, 32, 64 bit

Các thành phần chức năng của CPU

❑ Hệ thống Bus (tt):

❖ Address bus:

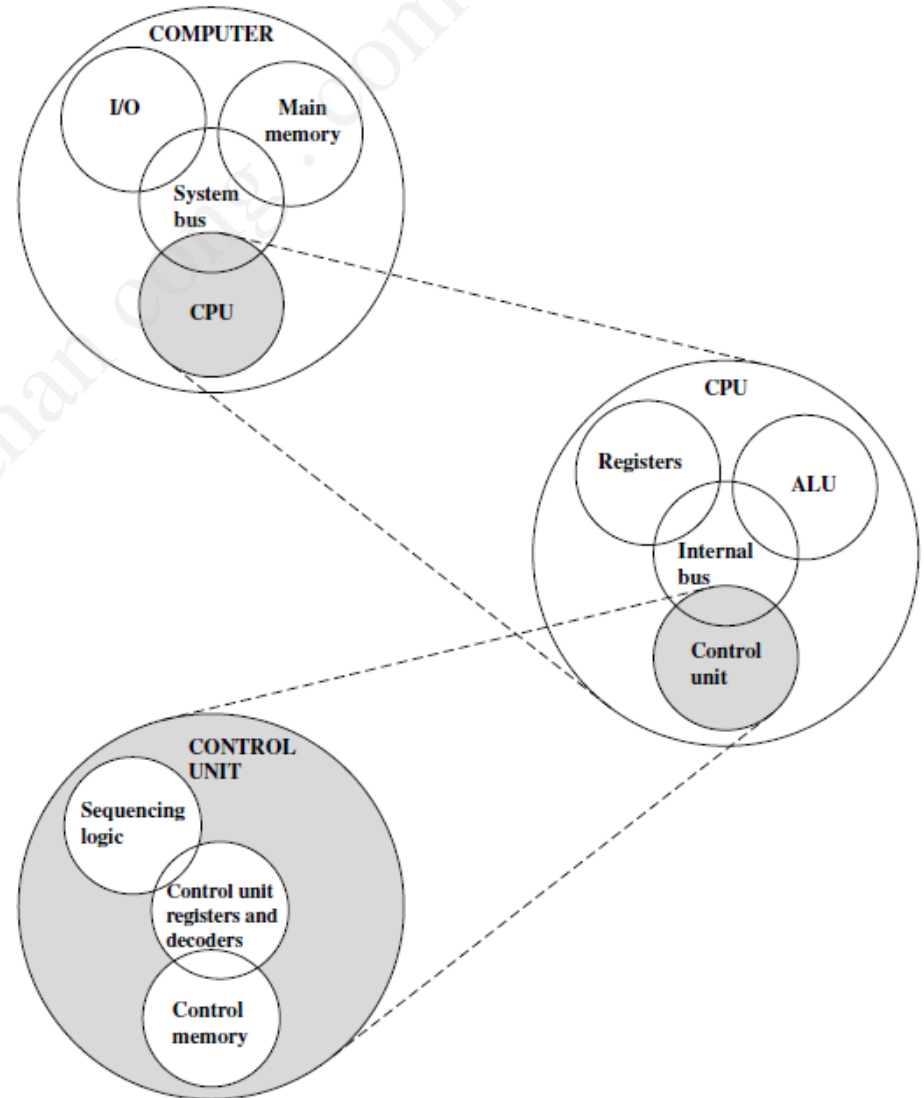
- Cung cấp địa chỉ nguồn và đích của dữ liệu cho CPU.
- Độ rộng Bus xác định khả năng nhớ của hệ thống. 8088 dùng 16 bit địa chỉ => không gian địa chỉ là 64K.

❖ Control bus:

- Điều khiển và định thời các thông tin:
 - ✓ Tín hiệu đọc/ghi bộ nhớ,
 - ✓ Yêu cầu ngắt,
 - ✓ Tín hiệu đồng hồ.

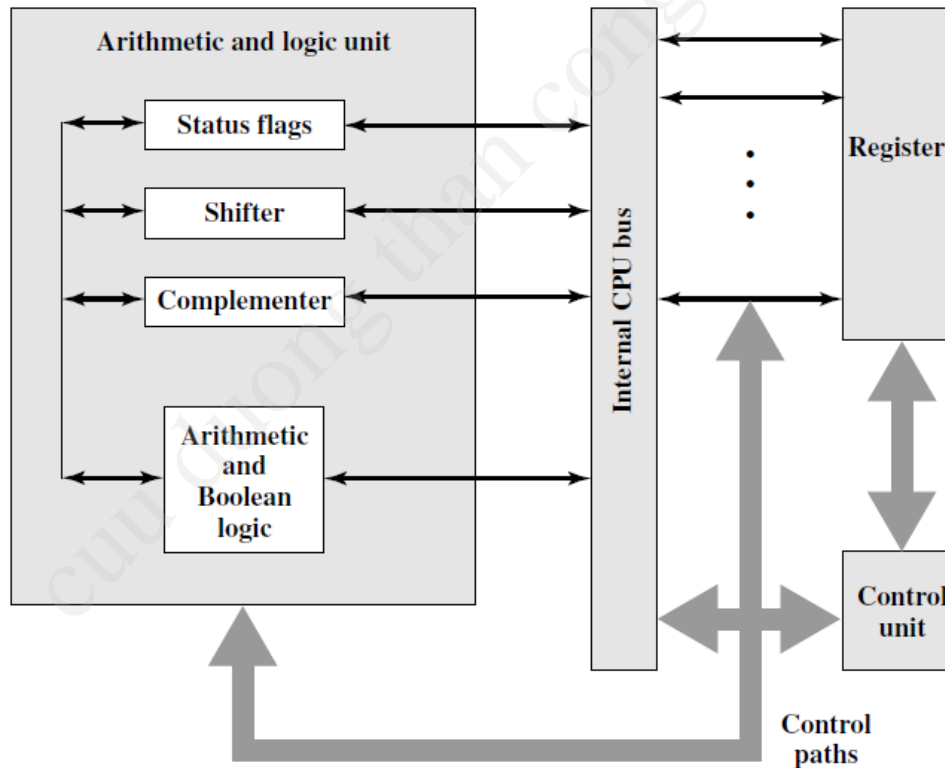
Các thành phần chức năng của CPU

❖ Cấu trúc mức tổng quát của CPU



Các thành phần chức năng của CPU

❖ Cấu trúc bên trong của CPU



Các thành phần chức năng của CPU

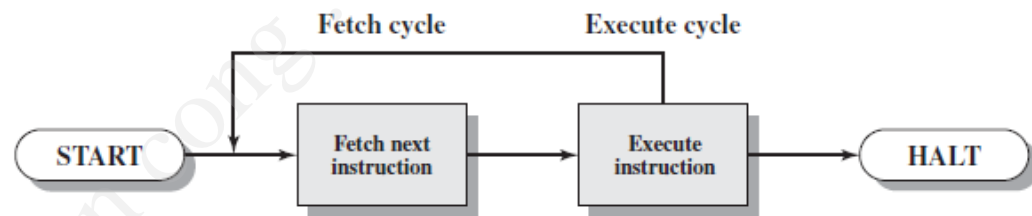
❖ Nhiệm vụ của CPU:

- Nhận lệnh (fetch instruction): đọc lệnh từ bộ nhớ.
- Giải mã lệnh (decode instruction): xác định tác vụ lệnh yêu cầu.
- Nhận dữ liệu (fetch data):
 - ✓ Nhận dữ liệu mới từ bộ nhớ hoặc các cổng I/O,
 - ✓ Xử lý dữ liệu (process data): thực hiện các phép toán số hoặc logic với các dữ liệu.
- Ghi dữ liệu (write data): ghi dữ liệu ra bộ nhớ hoặc các cổng I/O.

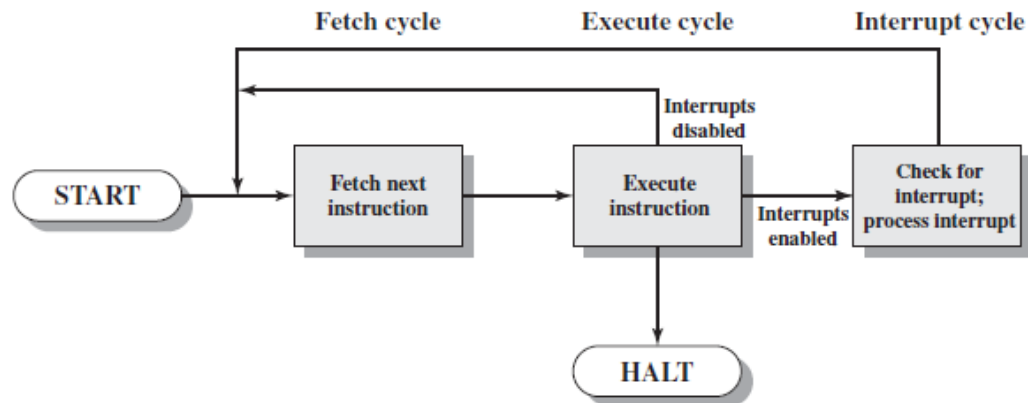
Các thành phần chức năng của CPU

❖ Chu kỳ lệnh:

- Nạp lệnh (**Fetch**):
Nạp lệnh → Giải mã lệnh → Nạp toán hạng
- Thực thi (**Execute**):
Thực thi → Cài toán hạng → Ngắt



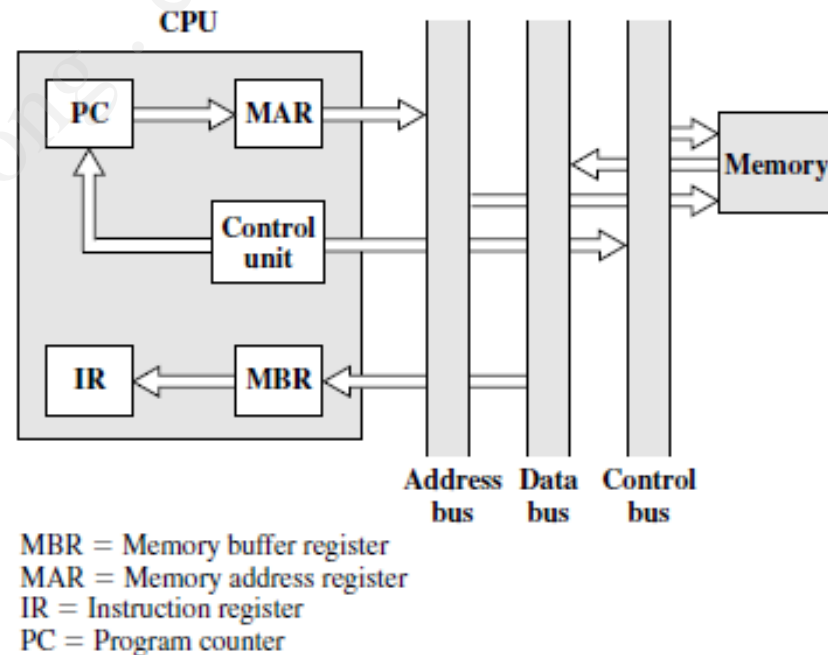
Chu kỳ lệnh cơ bản



Chu kỳ lệnh với ngắt

Các thành phần chức năng của CPU

- ❖ Nhận lệnh:
 - CPU đưa địa chỉ của lệnh cần nhận từ bộ đếm chương trình PC ra bus địa chỉ.
 - CPU phát tín hiệu điều khiển đọc bộ nhớ.
 - Lệnh từ bộ nhớ được đặt lên bus dữ liệu và được CPU copy vào thanh ghi lệnh IR.
 - CPU tăng nội dung PC để trở sang lệnh kế tiếp





Các thành phần chức năng của CPU

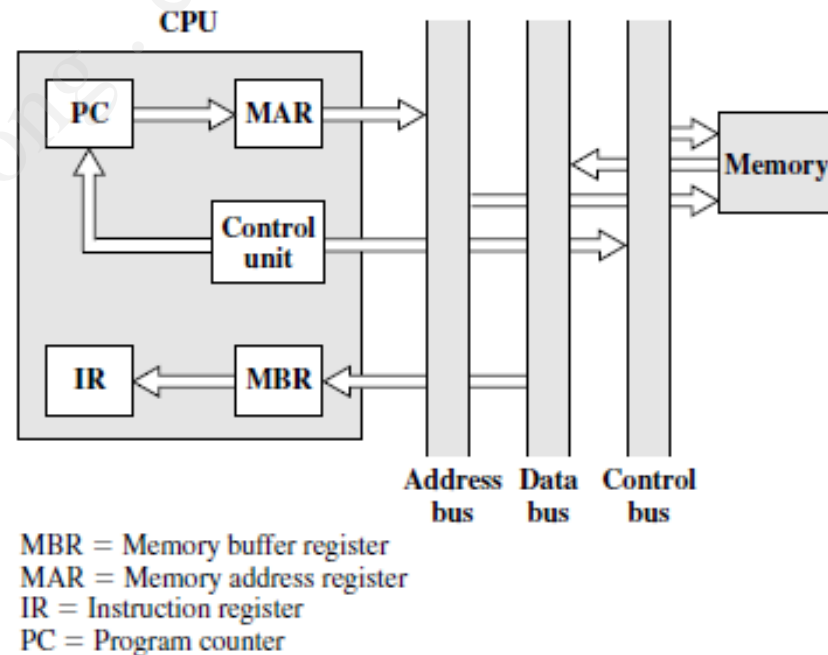
❖ Giải mã lệnh:

- Lệnh từ thanh ghi lệnh IR được đưa đến đơn vị điều khiển (**Control Unit**).
- Đơn vị điều khiển tiến hành giải mã lệnh để xác định tác vụ phải thực hiện.
- Giải mã lệnh xảy ra bên trong CPU

Các thành phần chức năng của CPU

❖ Nhận dữ liệu:

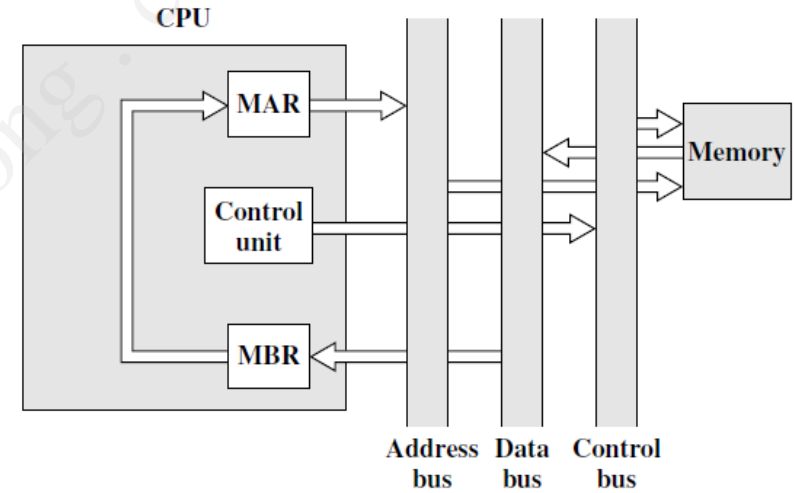
- CPU đưa địa chỉ của toán hạng ra bus địa chỉ
- CPU phát tín hiệu điều khiển đọc
- Toán hạng được đọc vào CPU
- Tương tự như nhận lệnh



Các thành phần chức năng của CPU

❖ Nhận dữ liệu gián tiếp (indirect):

- CPU đưa địa chỉ ra Address Bus, và phát tín hiệu điều khiển đọc.
- Nội dung ô nhớ (địa chỉ toán hạng – gián tiếp) được đọc vào CPU.
- CPU phát địa chỉ này ra bus địa chỉ để tìm toán hạng
- CPU phát tín hiệu điều khiển đọc, toán hạng được đọc vào CPU.



- MAR: Memory Address Register
- MBR: Memory Buffer Register

Các thành phần chức năng của CPU

❖ Thực thi lệnh:

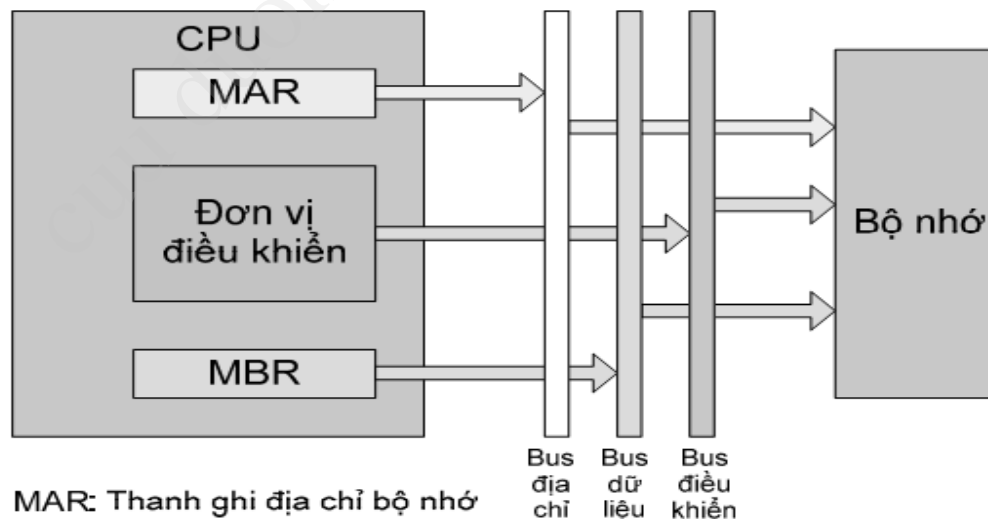
➤ Thực thi các tác vụ được mã hóa vào tập lệnh như:

- ✓ Đọc/Ghi bộ nhớ
- ✓ Nhập/Xuất
- ✓ Chuyển giữa các thanh ghi
- ✓ Thao tác số học/logic
- ✓ Chuyển điều khiển (rẽ nhánh)
- ✓ ...

Các thành phần chức năng của CPU

❖ Ghi toán hạng:

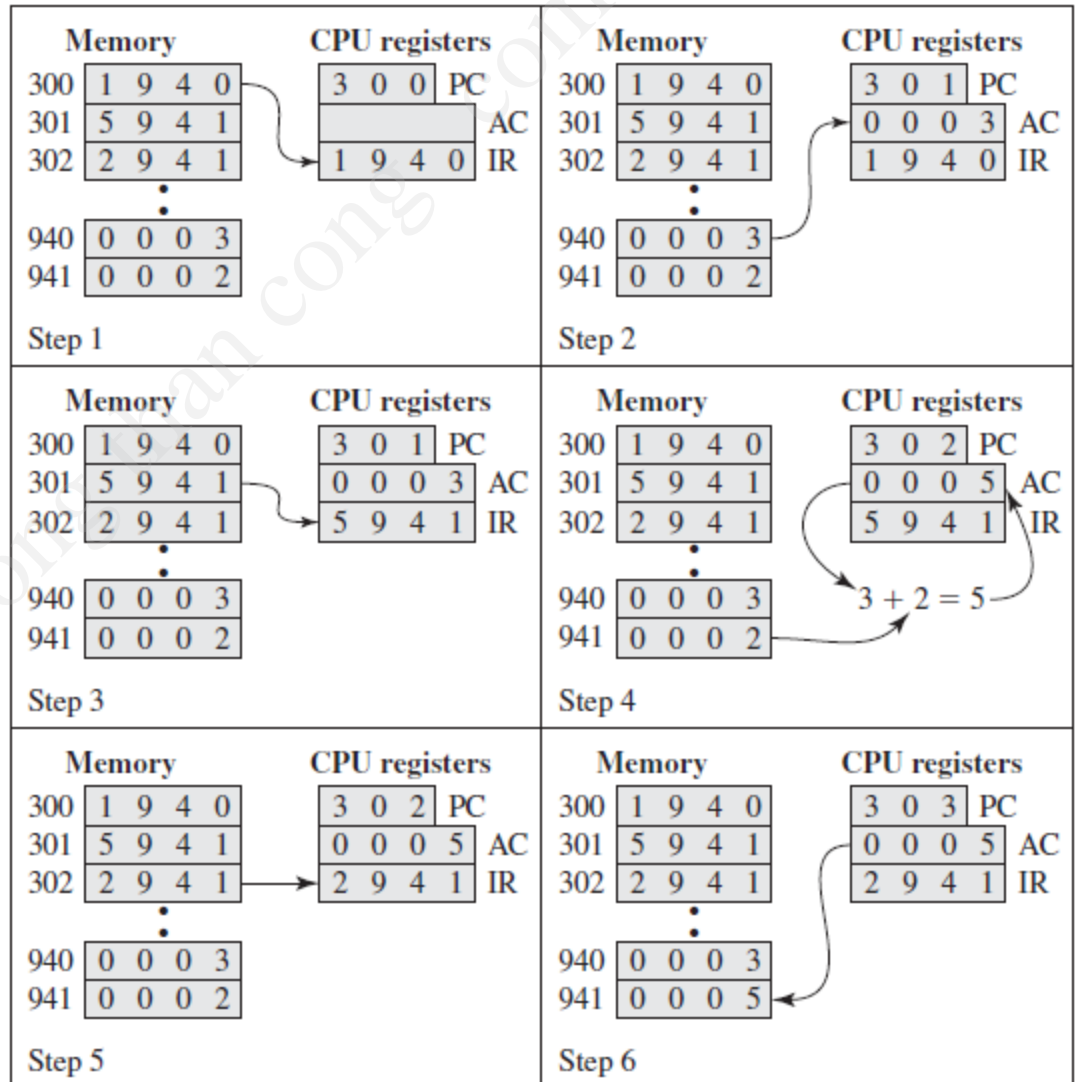
- CPU đưa địa chỉ ra bus địa chỉ.
- CPU đưa dữ liệu cần ghi ra bus dữ liệu.
- CPU phát tín hiệu điều khiển ghi.
- Dữ liệu trên bus dữ liệu được copy đến vị trí xác định.



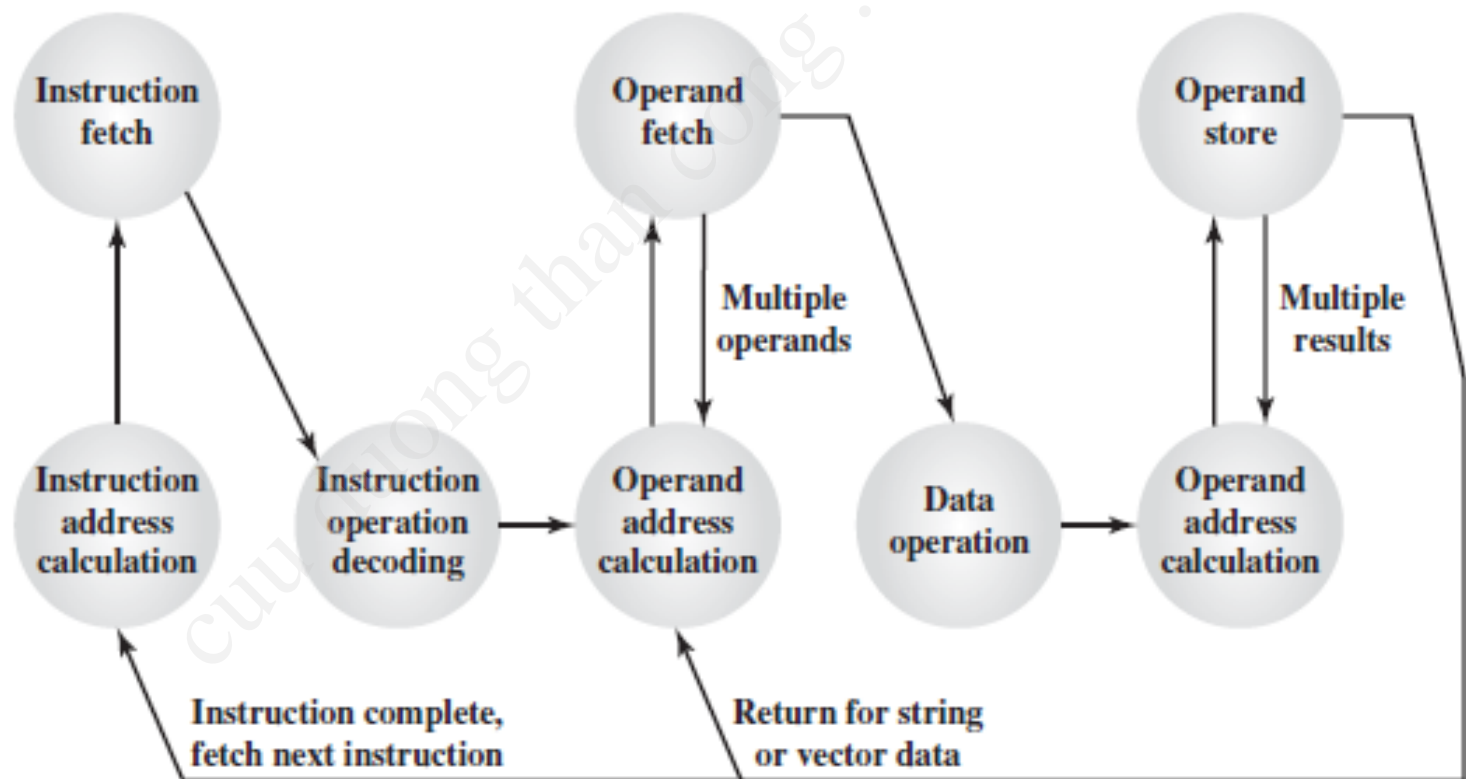
MAR: Thanh ghi địa chỉ bộ nhớ
MBR: Thanh ghi đệm bộ nhớ

Các thành phần chức năng của CPU

Ví dụ về thực thi chương trình



Các thành phần chức năng của CPU



Lưu đồ trạng thái chu kỳ lệnh

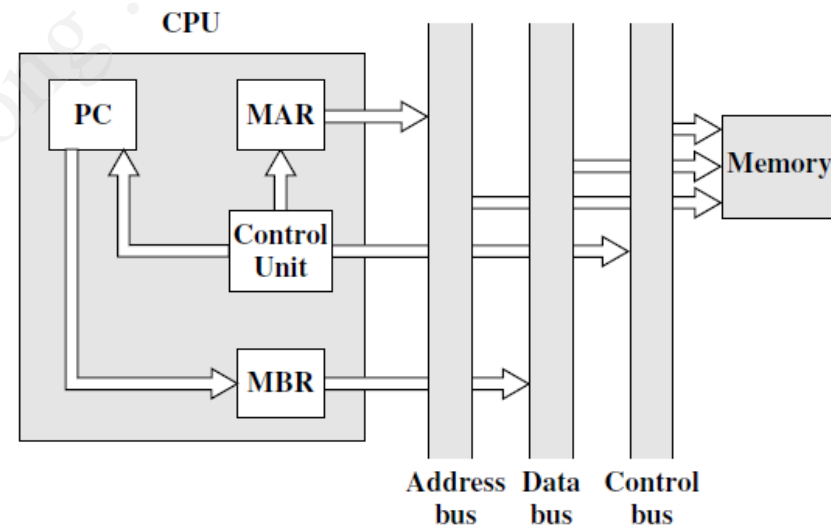


Ngắt (Interrupt)

- Là kỹ thuật cho phép các module khác (I/O) có thể ngắt ngang tiến trình xử lý bình thường
- Các loại ngắt:
 - ✓ Chương trình (**program**): overflow, division by zero
 - ✓ Định thời (**Timer**):
 - Được tạo bởi bộ định thời bên trong CPU
 - Được dùng ở chế độ độc quyền trong các hệ thống đa nhiệm
 - ✓ I/O: Từ I/O controller
 - ✓ Lỗi phần cứng: memory parity error

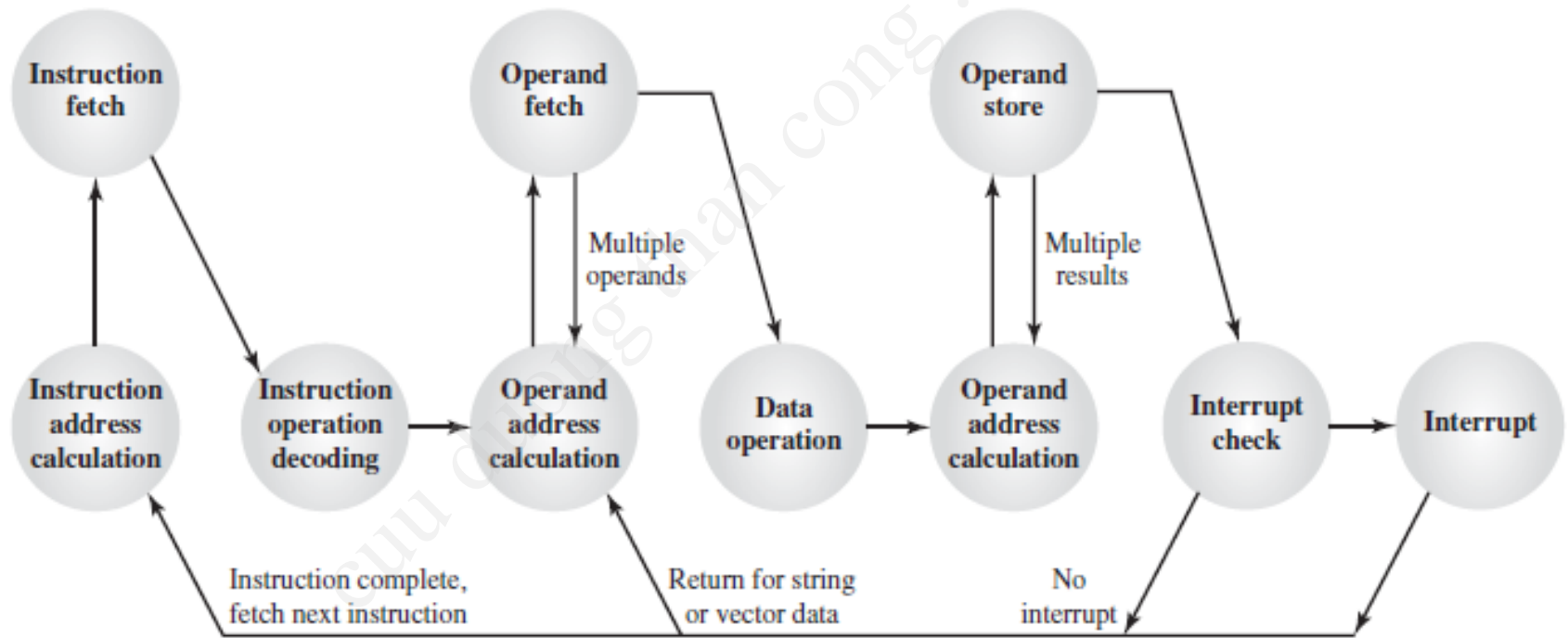
Ngắt (Interrupt)

- ❖ Chu kỳ lệnh có ngắt:
 - Nội dung của PC được đưa ra Data Bus.
 - CPU đưa địa chỉ (từ Stack Pointer) ra Address Bus.
 - CPU phát tín hiệu điều khiển ghi bộ nhớ.
 - Địa chỉ trả về trên Data Bus được ghi ra địa chỉ đã định (ở ngăn xếp)
 - Địa chỉ lệnh đầu tiên của chương trình con điều khiển ngắt được nạp vào PC



- MAR: Memory Address Register.
- MBR: Memory Buffer Register.
- PC: Program Counter

Ngắt (Interrupt)



Lưu đồ trạng thái trong chu kỳ lệnh có ngắt



Đa ngắt

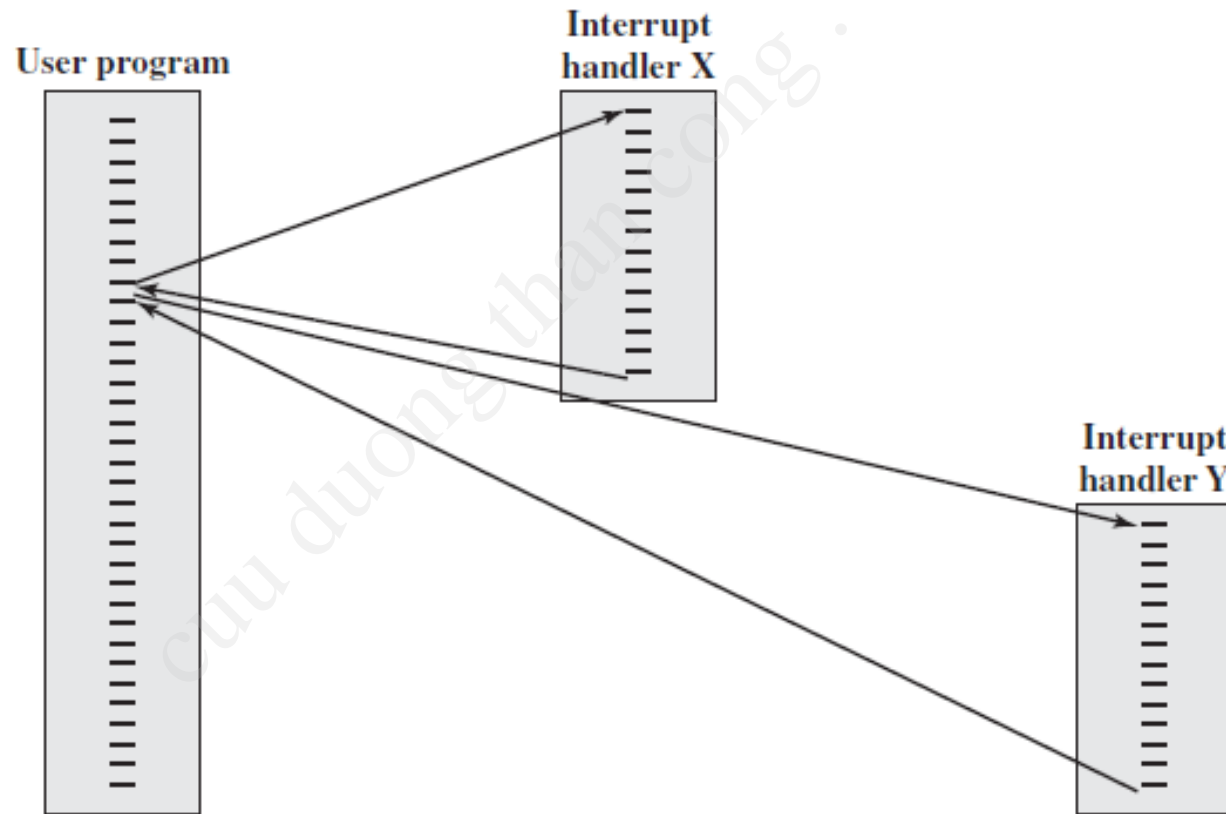
❖ Cấm ngắt

- CPU sẽ bỏ qua các ngắt mới, chỉ thi hành ngắt hiện tại.
- Các ngắt mới sẽ phải đợi và sẽ được kiểm tra sau khi ngắt hiện hành hoàn tất.
- Các ngắt được thi hành theo thứ tự mà chúng xảy ra.

❖ Độ ưu tiên ngắt

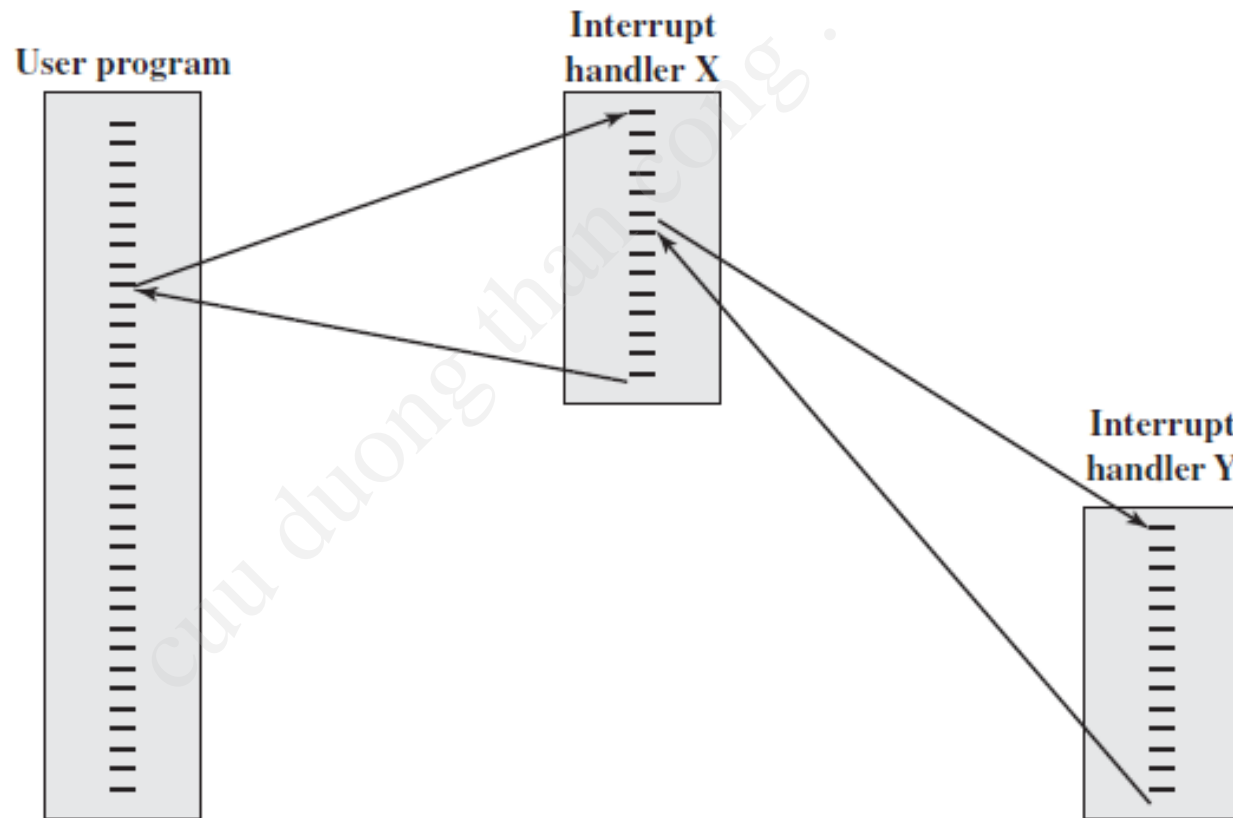
- Các ngắt có độ ưu tiên thấp có thể sẽ bị ngắt bởi các ngắt có độ ưu tiên cao hơn.
- Khi ngắt có độ ưu tiên cao hơn hoàn tất, CPU sẽ được trả về cho ngắt trước đó.

Đa ngắt



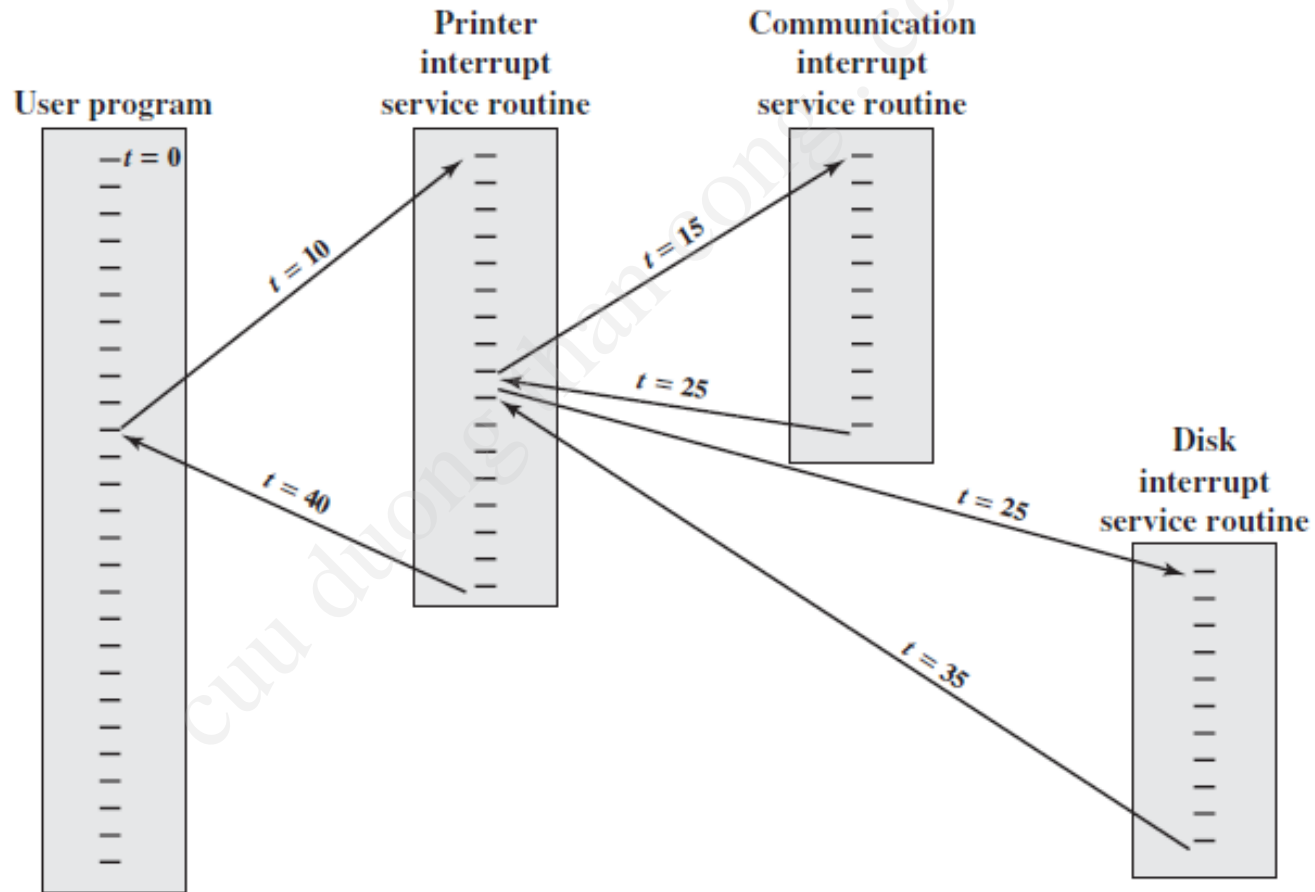
Đa ngắt tuần tự

Đa ngắt



Đa ngắt lồng nhau

Đa ngắt



Ví dụ về đa ngắt



Đường ống lệnh (Instruction Pipelining)

- Chia chu trình lệnh thành các công đoạn và cho phép thực hiện gộp lên nhau (như dây chuyền lắp ráp)
- Có 6 công đoạn:
 - ✓ Nhận lệnh (Fetch Instruction – **FI**)
 - ✓ Giải mã lệnh (Decode Instruction – **DI**)
 - ✓ Tính địa chỉ toán hạng (Calculate Operand Address – **CO**)
 - ✓ Nhận toán hạng (Fetch Operands – **FO**)
 - ✓ Thực hiện lệnh (Execute Instruction – **EI**)
 - ✓ Ghi toán hạng (Write Operands – **WO**)

Đường ống lệnh (Instruction Pipelining)

Time →

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Instruction 1	FI	DI	CO	FO	EI	WO								
Instruction 2		FI	DI	CO	FO	EI	WO							
Instruction 3			FI	DI	CO	FO	EI	WO						
Instruction 4				FI	DI	CO	FO	EI	WO					
Instruction 5					FI	DI	CO	FO	EI	WO				
Instruction 6						FI	DI	CO	FO	EI	WO			
Instruction 7							FI	DI	CO	FO	EI	WO		
Instruction 8								FI	DI	CO	FO	EI	WO	
Instruction 9									FI	DI	CO	FO	EI	WO

Biểu đồ thời gian của đường ống lệnh

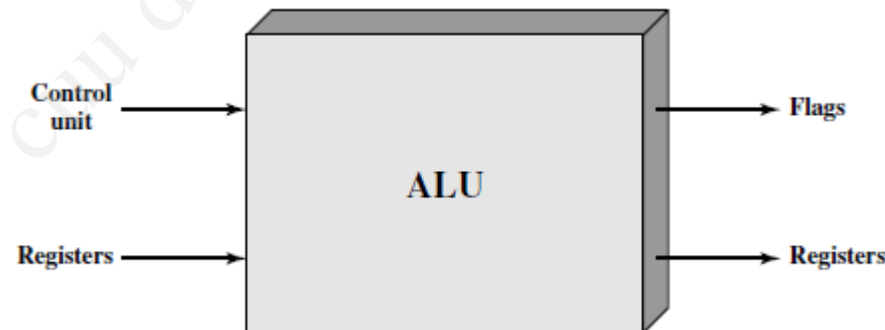


Đường ống lệnh (Instruction Pipelining)

- ❖ Các vấn đề với đường ống lệnh:
 - Xung đột cấu trúc: do nhiều công đoạn dùng chung một tài nguyên
 - Xung đột dữ liệu: lệnh sau sử dụng dữ liệu kết quả của lệnh trước
 - Xung đột điều khiển: do rẽ nhánh gây ra

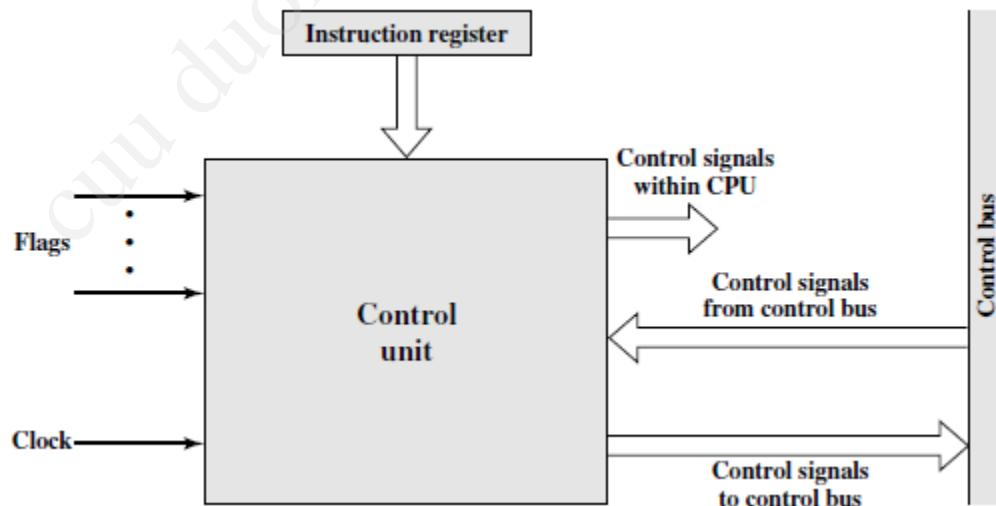
Khối luận lý và số học (Arithmetic and Logic Unit - ALU)

- Là một phần của máy tính, thực hiện các phép toán luận lý và số học.
- Số học: cộng, trừ, nhân, chia, tang, giảm, đảo dấu.
- Logic: AND, OR, XOR, NOT, phép dịch bit.



Đơn vị điều khiển (Control Unit)

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





Đơn vị điều khiển (Control Unit)

- ❖ **Clock**: Tín hiệu xung nhịp từ mạch tạo dao động.
- ❖ **Instruction register**: Mã lệnh từ thanh ghi lệnh đưa đến để giải mã.
- ❖ **Flags**: Các cờ từ thanh ghi cờ cho biết trạng thái của CPU và kết quả các phép toán từ khối ALU trước đó.
- ❖ **Control signals from control bus**: Các tín hiệu yêu cầu từ bus điều khiển.
- ❖ **Control signals within the processor**:
 - Điều khiển các thanh ghi
 - Điều khiển ALU
- ❖ **Control signals to control bus**:
 - Điều khiển bộ nhớ
 - Điều khiển các module I/O



Thanh ghi (Registers)

- Các thanh ghi nằm trong CPU.
- Chứa các thông tin tạm thời phục vụ cho hoạt động ở thời điểm hiện tại của CPU.
- Được coi là mức đầu tiên của hệ thống nhớ.
- Tùy thuộc vào bộ xử lý cụ thể.
- Số lượng thanh ghi nhiều -> tăng hiệu năng của CPU.
- Có hai loại thanh ghi:
 - ✓ Các thanh ghi lập trình được
 - ✓ Các thanh ghi không lập trình được



Thanh ghi (Registers)

- Thanh ghi địa chỉ: Quản lý địa chỉ của ngăn nhớ hay cổng vào/ra.
- Thanh ghi dữ liệu: Chứa các dữ liệu tạm thời.
- Thanh ghi đa năng: Có thể chứa địa chỉ hoặc dữ liệu.
- Thanh ghi điều khiển/trạng thái: Chứa các thông tin điều khiển và trạng thái của CPU.
- Thanh ghi lệnh: chứa lệnh đang được thực hiện.



Thanh ghi (Registers)

- ❖ Bộ đếm chương trình (Program Counter – **PC**):
 - Còn được gọi là con trỏ lệnh (Instruction Pointer - **IP**),
 - Giữ địa chỉ của lệnh tiếp theo sẽ được nhận vào,
 - Sau khi một lệnh được nhận vào, nội dung PC tự động tăng để trỏ sang lệnh kế tiếp.
- ❖ Con trỏ dữ liệu (Data Pointer - **DP**)
 - Chứa địa chỉ của ngăn nhớ dữ liệu mà CPU muốn truy nhập

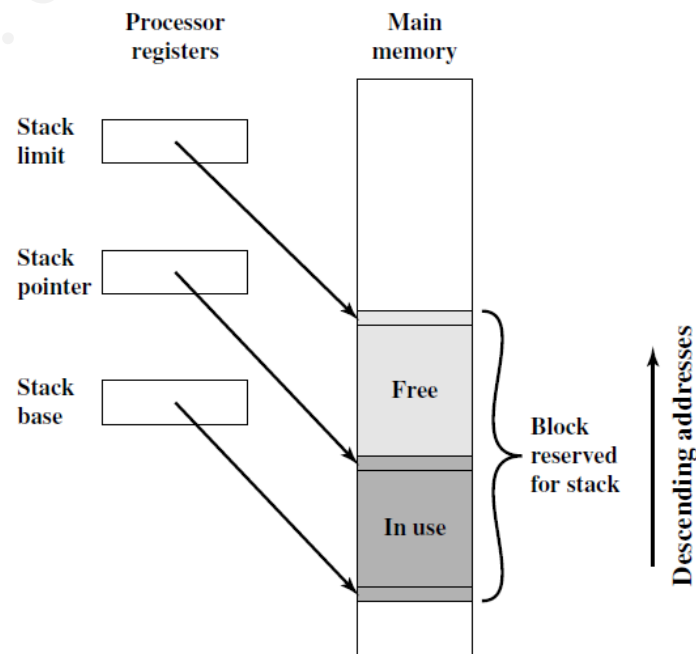


Thanh ghi (Registers)

- ❖ Ngăn xếp (**Stack**)
 - Là vùng nhớ có cấu trúc **LIFO** (Last In – First Out).
 - Thường dùng để phục vụ chương trình con.
 - Đáy ngăn xếp là một ngăn nhớ xác định.
 - Đỉnh ngăn xếp là thông tin nằm ở vị trí trên cùng trong ngăn xếp.
 - Đỉnh ngăn xếp có thể bị thay đổi.

Thanh ghi (Registers)

- ❖ Con trỏ ngăn xếp (Stack Pointer - **SP**):
 - Chứa địa chỉ đỉnh ngăn xếp,
 - Khi cất một thông tin vào ngăn xếp: Nội dung của SP giảm.
 - Khi lấy một thông tin ra: Nội dung của SP tăng.
 - Khi ngăn xếp rỗng SP trở vào đáy
- ❖ Thanh ghi cơ sở ngăn xếp (Stack Base): chứa địa chỉ vị trí đáy của ngăn xếp.
- ❖ Thanh ghi giới hạn ngăn xếp (Stack limit): Chứa địa chỉ vị trí tối đa của ngăn xếp.





Thanh ghi (Registers)

❖ Các thanh ghi dữ liệu:

- Chứa các dữ liệu tạm thời hoặc các kết quả trung gian.
- Cần có nhiều thanh ghi dữ liệu.
- Các thanh ghi số nguyên: 8, 16, 32, 64 bit.
- Các thanh ghi số dấu chấm động.



Thanh ghi (Registers)

- ❖ Thanh ghi trạng thái (**Status register**)
 - Còn gọi là thanh ghi cờ (Flag Register)
 - Chứa các thông tin trạng thái của CPU
 - Các cờ phép toán: báo hiệu trạng thái của kết quả phép toán
 - Các cờ điều khiển: biểu thị trạng thái điều khiển của CPU

Thanh ghi (Registers)

❖ Cờ phép toán

- Cờ Zero (cờ rỗng): được thiết lập lên 1 khi kết quả của phép toán bằng 0.
- Cờ Sign (cờ dấu): được thiết lập lên 1 khi kết quả phép toán nhỏ hơn 0
- Cờ Carry (cờ nhớ): được thiết lập lên 1 nếu phép toán có nhớ ra ngoài bit cao nhất → **cờ báo tràn với số không dấu.**
- Cờ Overflow (cờ tràn): được thiết lập lên 1 nếu cộng hai số nguyên cùng dấu mà kết quả có dấu ngược lại → **cờ báo tràn với số có dấu.**

Thanh ghi (Registers)

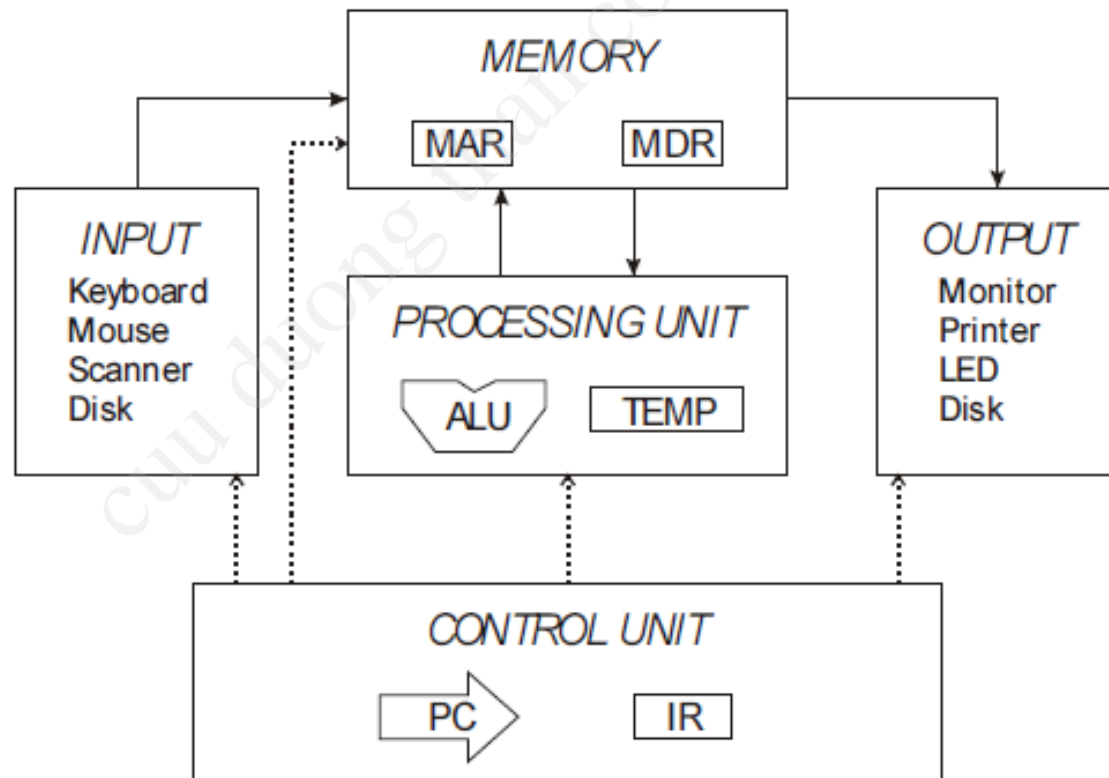
❖ Cờ điều khiển

➤ Cờ Interrupt (Cờ cho phép ngắt)

- ✓ Nếu $IF = 1 \rightarrow$ CPU ở trạng thái cho phép ngắt với tín hiệu yêu cầu ngắt từ bên ngoài gửi tới
- ✓ Nếu $IF = 0 \rightarrow$ CPU ở trạng thái cấm ngắt với tín hiệu yêu cầu ngắt từ bên ngoài gửi tới

Lệnh máy tính

❖ Mô hình máy tính Von Neumann:



Lệnh máy tính

1. Memory – Bộ nhớ:

- Một mảng $2^k \times m$ bit
- Địa chỉ: k bit
- Giá trị của ô nhớ: m bit
- Các hoạt động cơ bản:
 - ✓ LOAD: Đọc giá trị từ một ô nhớ
 - ✓ STORE: Viết giá trị đến một ô nhớ.

0000	
0001	
0010	
0011	00101101
0100	
0101	
0110	
	⋮
1101	10100010
1110	
1111	

Lệnh máy tính

1. Memory – Bộ nhớ:

❖ Giao tiếp với bộ nhớ:

➤ **MAR**: Memory Address Register

➤ **MDR**: Memory Data Register

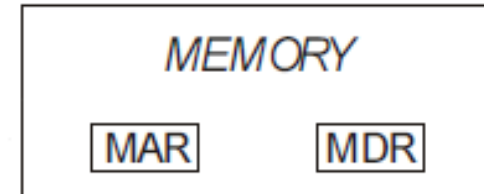
➤ Đọc từ một vị trí có địa chỉ “A”:

1. Ghi địa chỉ “A” vào MAR
2. Gửi tín hiệu điều khiển đọc đến bộ nhớ.

3. Đọc dữ liệu từ MDR

➤ Ghi giá trị “X” vào vị trí “A”:

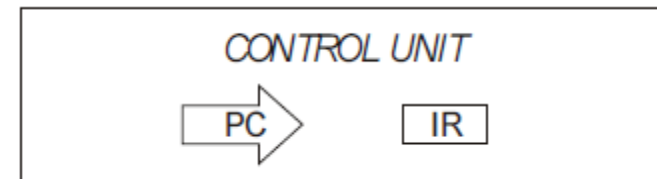
1. Ghi giá trị “X” vào MDR
2. Ghi địa chỉ “A” vào MAR
3. Gửi tín hiệu điều khiển ghi vào bộ nhớ



Lệnh máy tính

2. Control Unit – Đơn vị điều khiển:

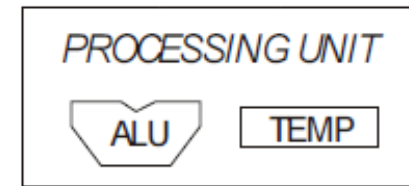
- **IR**: Instruction Register
 - ✓ Chứa lệnh hiện hành
- **PC**: Program Counter
 - ✓ Chứa địa chỉ của lệnh kết cần xử lý
- Control Unit đọc một lệnh từ bộ nhớ (trong PC)
- Dựa vào lệnh nhận được, khối này phát tín hiệu điều khiển để gọi thành phần tương ứng thực hiện tác vụ.
- Một lệnh có thể tốn một hoặc nhiều chu kỳ máy



Lệnh máy tính

3. Processing Unit – Đơn vị xử lý:

- Các đơn vị chức năng:
 - ✓ ALU (**Arithmetic and Logic Unit**),
 - ✓ Nhiều đơn vị chức năng như nhân, căn bậc 2, ...
 - ✓ LC-3 thực hiện được ADD, AND, NOT
- Thanh ghi (**Register**)
 - ✓ Dung lượng nhỏ, lưu trữ tạm thời,
 - ✓ Chứa toán hạng và kết quả thực hiện của các đơn vị chức năng,
 - ✓ LC-3 có 8 thanh ghi (R_0, \dots, R_7).
- Độ dài từ nhớ (**Word size**)
 - ✓ Là số bit ALU có thể xử lý trong một lệnh
 - ✓ LC-3 có độ dài từ nhớ là 16 bit.



Lệnh máy tính

4. Input/Output – Nhập/Xuất:

- Dùng để đưa dữ liệu vào bộ nhớ máy tính hoặc lấy dữ liệu ra từ bộ nhớ máy tính.
- Mỗi thiết bị có giao diện riêng, dạng thanh ghi **MAR** và **MDR** của bộ nhớ
 - ✓ LC-3 hỗ trợ keyboard (input) và màn hình (output),
 - ✓ Keyboard: thanh ghi dữ liệu (data register - **KBDR**); thanh ghi trạng thái (status register - **KBSR**)
 - ✓ Monitor: data register (**DDR**) và status register (**DSR**)
- Chương trình truy cập thiết bị thường sử dụng thông qua trình điều khiển thiết bị (**driver**)

INPUT
Keyboard
Mouse
Scanner
Disk

OUTPUT
Monitor
Printer
LED
Disk

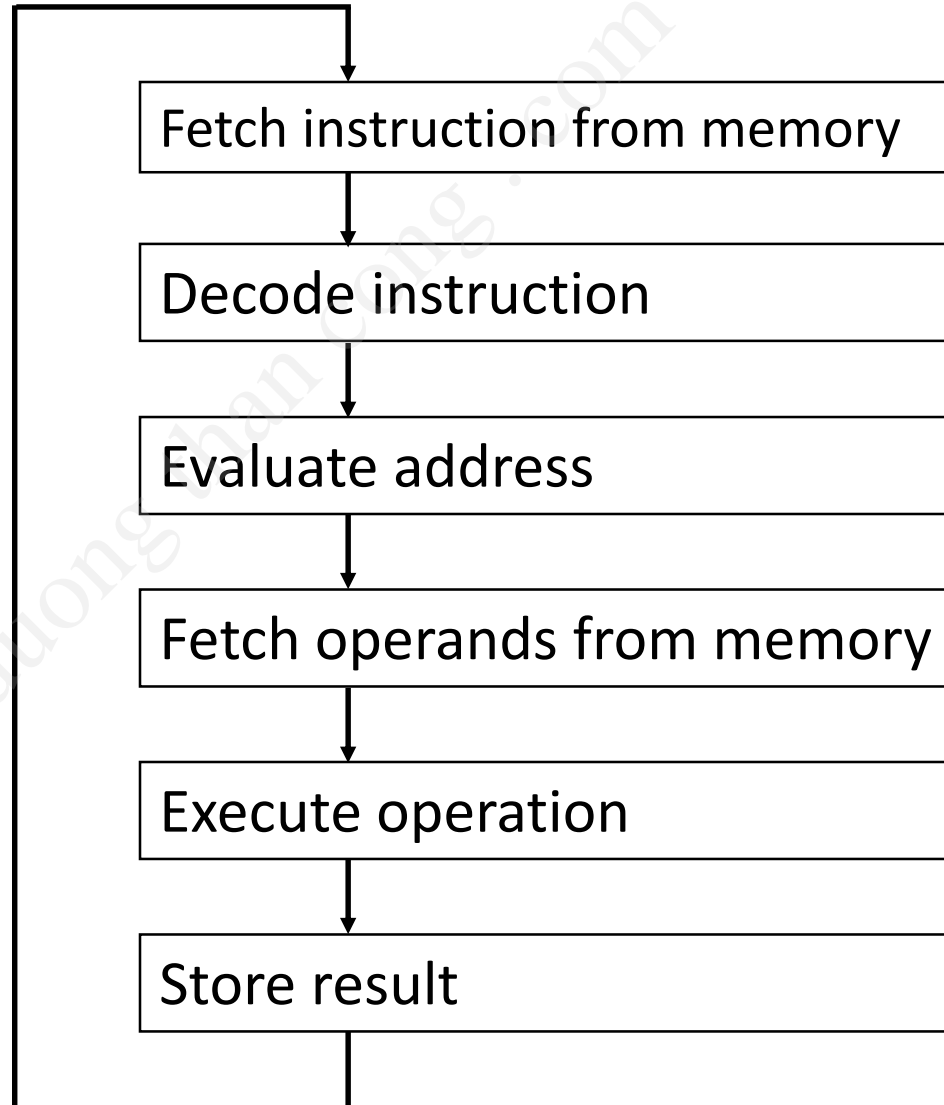


Lệnh máy tính

- Lệnh là một đơn vị cơ bản của công việc.
- Mỗi lệnh là một chuỗi số nhị phân mà bộ xử lý hiểu được để thực hiện một thao tác xác định.
- Control unit dịch lệnh, sinh ra chuỗi các tín hiệu điều khiển để thực hiện tác vụ.
- Thành phần của lệnh:
 - ✓ **Opcode**: tác vụ cần thực thi
 - ✓ **Operands**: dữ liệu/địa chỉ dùng cho tác vụ
- Các lệnh của máy tính và các định dạng của chúng gọi là kiến trúc tập lệnh (**Instruction Set Architecture – ISA**).
- Mỗi bộ xử lý có một tập lệnh xác định

Lệnh máy tính

❖ Xử lý lệnh:



Lệnh máy tính

LC-3 Instructions

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD+	0001				DR			SR1			0	00		SR2		
ADD+	0001				DR			SR1			1	imm5				
AND+	0101				DR			SR1			0	00		SR2		
AND+	0101				DR			SR1			1	imm5				
BR	0000				n	z	p	PCOffset9								
JMP	1100				000			Base R			000000					
JSR	0100				1	PCOffset11										
JSRR	0100				0	00		Base R			000000					
LD+	0010				DR			PCOffset9								
LDI+	1010				DR			PCOffset9								
LDR+	0110				DR			Base R			offset6					
LEA+	1110				DR			PCOffset9								
NOT+	1001				DR			SR			111111					
RET	1100				000			111			000000					
RTI	1000				000000000000											
ST	0011				SR			PCOffset9								
STI	1011				SR			PCOffset9								
STR	0111				SR			Base R			offset6					
TRAP	1111				0000			trapvect8								
reserved	1101															

Chế độ địa chỉ:

- Địa chỉ thanh ghi (**Register Addressing**): Toán hạng là 1 trong 8 thanh ghi
- Địa chỉ tức thời (**Immediate Addressing**): Toán hạng là giá trị chứa trong lệnh, không cần tham chiếu đến bộ nhớ
- Địa chỉ tương đối với PC (**PC-Relative**): Lấy nội dung ô nhớ ở địa chỉ tính từ thanh ghi PC.
- Địa chỉ gián tiếp (**Indirect**): Toán hạng là địa chỉ của con trỏ chỉ đến dữ liệu.
- Địa chỉ tương đối với thanh ghi cơ sở (**Base relative**): Toán hạng là độ dời từ nội dung của thanh ghi.

Lệnh máy tính

Lệnh	Mô tả chức năng
ADD	Phép toán '+'. Trị 0 tại bit 5 ứng với chế độ thanh ghi; 1 là chế độ địa chỉ trực tiếp
AND	Phép toán luận lý. Trị 0 tại bit 5 ứng với chế độ thanh ghi; 1 là chế độ địa chỉ trực tiếp
BR	Branch (If ...) – Rẽ nhánh đến một nhãn nếu ...
JMP	Jump – Nhảy đến địa chỉ là nội dung của thanh ghi.
JSR	Jump Subroutine – Đến vị trí chương trình con
JSRR	Jump Subroutine Register – Đến vị trí chương trình con có địa chỉ trong thanh ghi
LD	Load Data – Đọc dữ liệu tại ô nhớ tương đối so với thanh ghi PC.
LDI	Load Indirect – Đọc dữ liệu tại một địa chỉ gián tiếp
LDR	Load Register – Đọc dữ liệu tại địa chỉ là nội dung của thanh ghi + độ dời

Lệnh máy tính

Lệnh	Mô tả chức năng
LEA	Load Effective Address – Đọc dữ liệu tại địa chỉ PC + độ dời
NOT	Phép toán luận lý. Chỉ làm việc với thanh ghi.
RET	Return – Trở về từ chương trình con ($PC \leftarrow R7$)
RTI	Return from Interrupt – Trở về sau sự kiện ngắt
ST	Store Data – Viết dữ liệu vào địa địa chỉ PC + độ dời
STI	Store Indirect – Viết dữ liệu đến địa chỉ là một địa chỉ tương đối so với thanh ghi PC ($\text{mem}(\text{mem}(\text{PC} + \text{độ dời}))$)
STR	Store Register – Viết dữ liệu vào thanh ghi + độ dời
TRAP	Lời gọi hệ thống ($R7 \leftarrow PC$)

Lệnh máy tính

- LC-3 có lệnh 16 bit.
- Mỗi lệnh có 4 bit opcode [15, 14, 13, 12]
- Ví dụ:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD				Dst			Src1			0	0	0	Src2		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	1	1	0	0	1	0	0	0	0	1	1	0

“Cộng nội dung của thanh ghi R2 với thanh ghi R6, lưu trữ kết quả vào thanh ghi R6.”

Lệnh máy tính

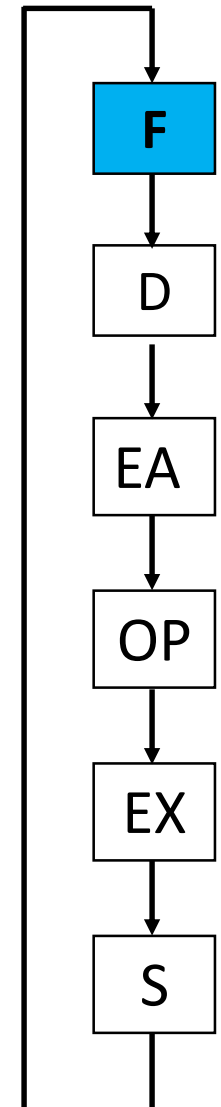
Xử lý lệnh:

❖ **FETCH.**

- Lấy lệnh kế tiếp từ PC đưa vào IR
 - ✓ Lấy nội dung của PC đưa vào MAR.
 - ✓ Gửi tín hiệu điều khiển đọc tới bộ nhớ.
 - ✓ Đọc nội dung của MDR và đưa vào IR
- Thanh ghi PC chỉ đến lệnh kế tiếp
 - ✓ $PC = PC + 2$

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD				Dst			Src1			0	0	0	Src2		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	1	1	0	0	1	0	0	0	0	1	1	0



Lệnh máy tính

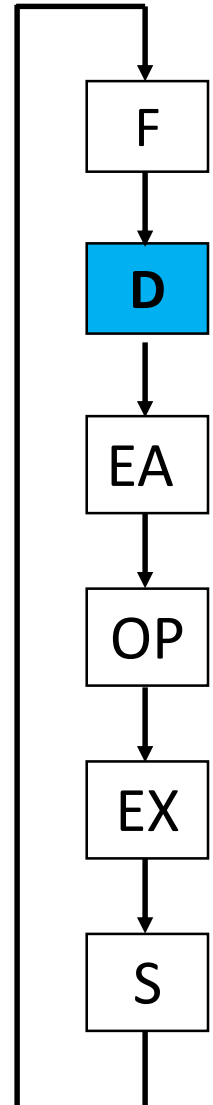
Xử lý lệnh:

❖ DECODE.

- Nhận dạng opcode.
 - ✓ 4 bit [15-12] của lệnh. ('0001')
- Xác định giá trị các toán hạng từ các bit còn lại (với lệnh **ADD**)
 - ✓ 3 bit [8-6] toán hạng nguồn 1 : 010
 - ✓ 3 bit [2-0] toán hạng nguồn 2 : 110
 - ✓ 3 bit [10-9] toán hạng đích : 110

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD				Dst			Src1			0	0	0	Src2		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	1	1	0	0	1	0	0	0	0	1	1	0



Lệnh máy tính

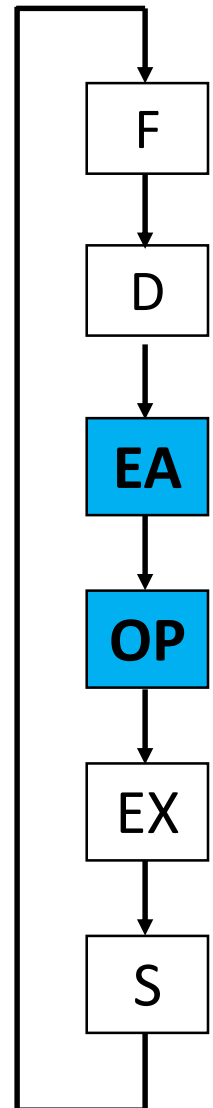
Xử lý lệnh:

❖ EVALUATE ADDRESS

- Trong những chỉ thị cần truy cập bộ nhớ, tính toán địa chỉ sử dụng cho việc truy cập bộ nhớ.

❖ FETCH OPERANDS

- Lấy các toán hạng nguồn để thi hành tác vụ.
- Ví dụ:
 - ✓ Lấy dữ liệu từ bộ nhớ (LDR)
 - ✓ Đọc dữ liệu từ thanh ghi



Lệnh máy tính

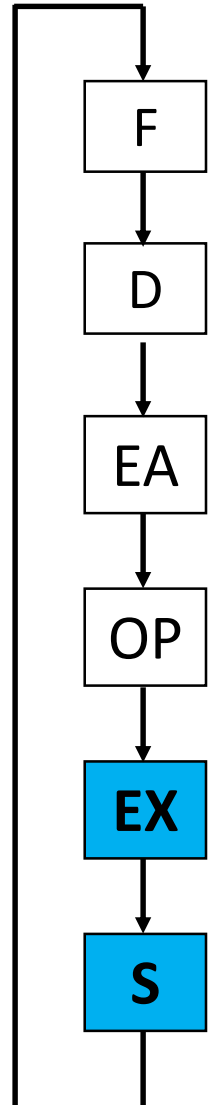
Xử lý lệnh:

❖ EXECUTE

- Lấy các toán hạng nguồn để thi hành tác vụ.
- Ví dụ:
 - ✓ Lấy dữ liệu từ bộ nhớ (LDR)
 - ✓ Đọc dữ liệu từ thanh ghi

❖ STORE.

- Lưu trữ kết quả ra bộ nhớ hoặc thanh ghi.



Lệnh máy tính

Ví dụ: Xác định nội dung thực hiện của chương trình sau.

```
.orig x3000
AND R0, R0, #0
LD R1, UNO
LD R2, DUE
INTEL ADD R0, R0, R2
      ADD R1, R1, #-2
      BRZ AMD
      BRNZP INTEL
AMD ST R0, TRE
      HALT
UNO .FILL X00DE
DUE .FILL X0003
TRE .FILL XFFFF
.end
```