

Chương 3: Khối xử lý trung tâm

Nội dung

3.1 Thành phần của hệ thống máy tính

3.2 Tổ chức của CPU

3.3 Thanh ghi

3.4 Khối điều khiển

3.5 Khối số học và logic

3.6 Phân tích mô hình máy tính đơn giản

Nội dung

3.1 Thành phần của hệ thống máy tính

3.2 Tổ chức của CPU

3.3 Thanh ghi

3.4 Khối điều khiển

3.5 Khối số học và logic

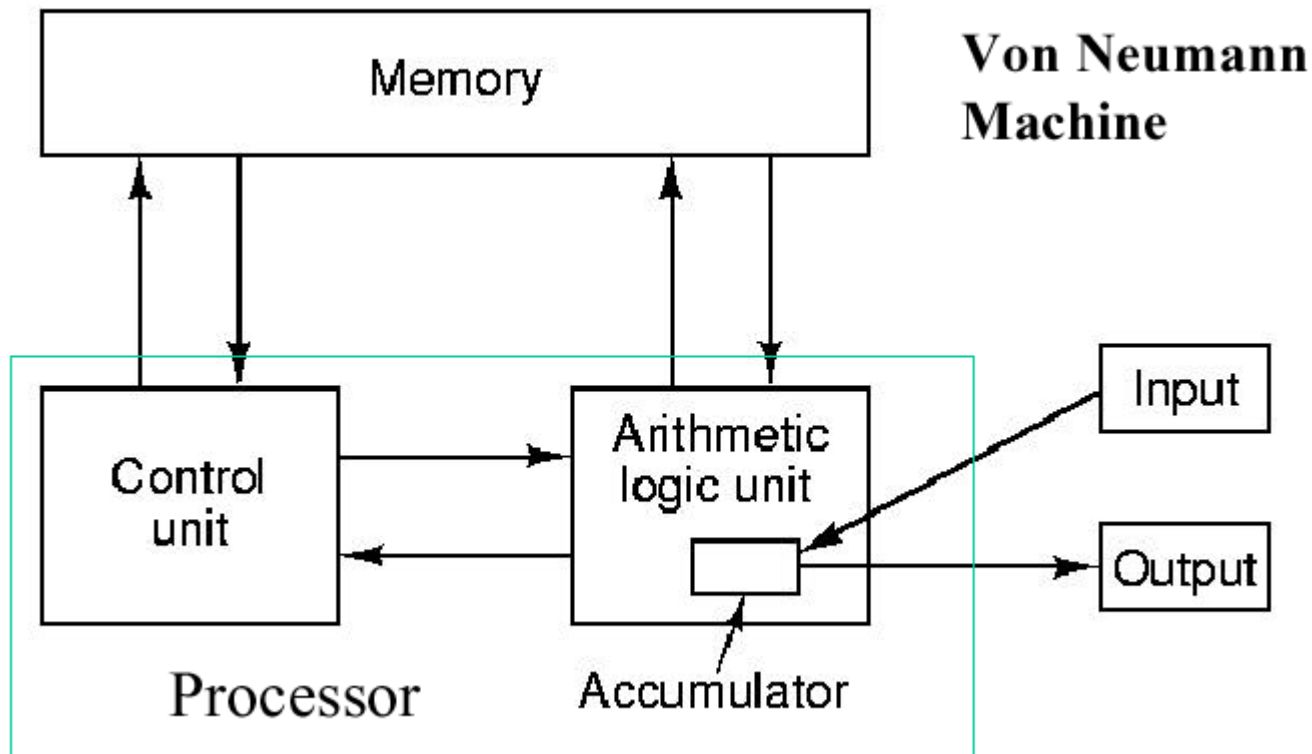
3.6 Phân tích mô hình máy tính đơn giản

3.1 Thành phần của hệ thống máy tính

- **Các thành phần của hệ thống máy tính:**
 - Bộ xử lý trung tâm (Central Processing Unit)
 - Bộ nhớ (Memory)
 - Hệ thống vào/ra (Input/Output System)
 - Liên kết hệ thống (System Interconnection)

3.1 Thành phần của hệ thống máy tính

- Mô hình máy tính theo kiến trúc Von-neumann



Nội dung

3.1 Thành phần của hệ thống máy tính

3.2 Tổ chức của CPU

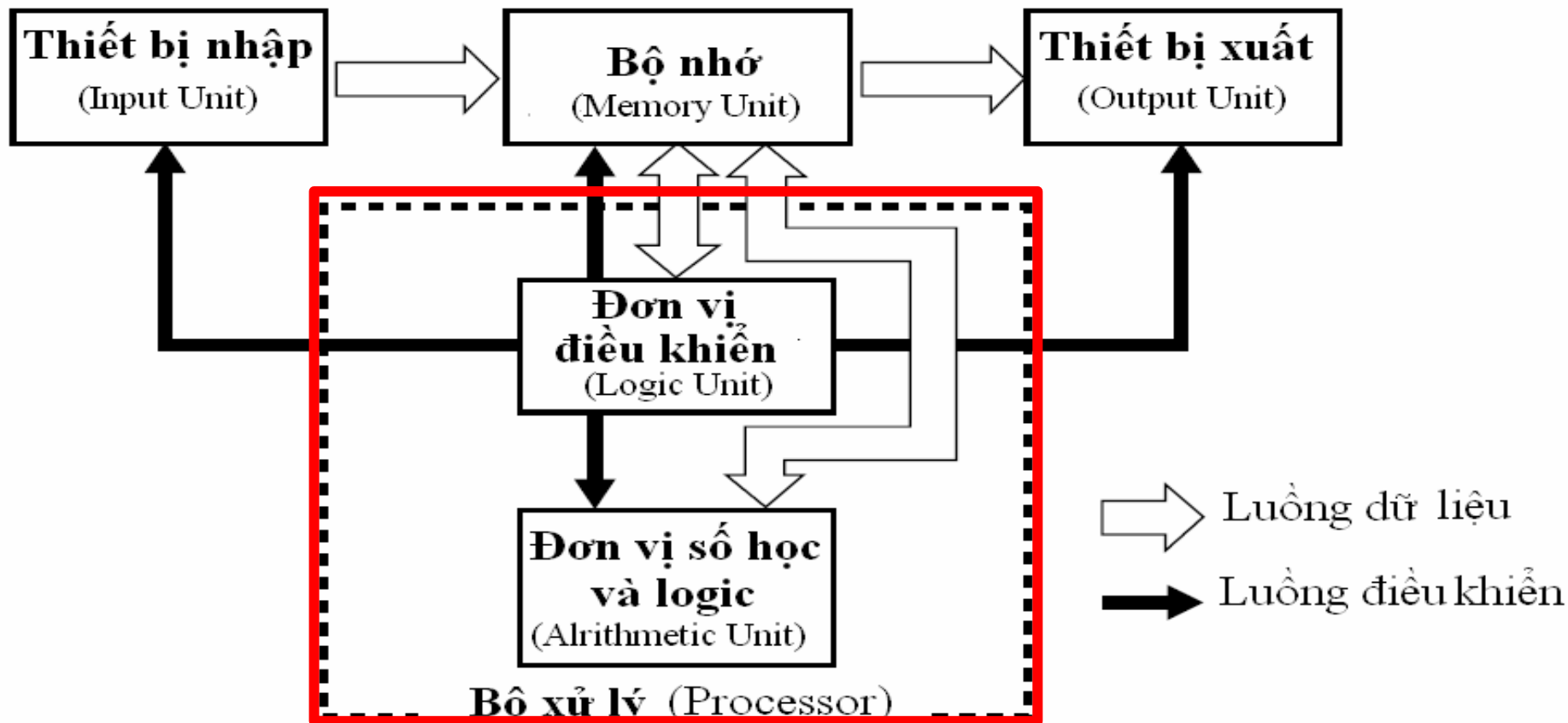
3.3 Thanh ghi

3.4 Khối điều khiển

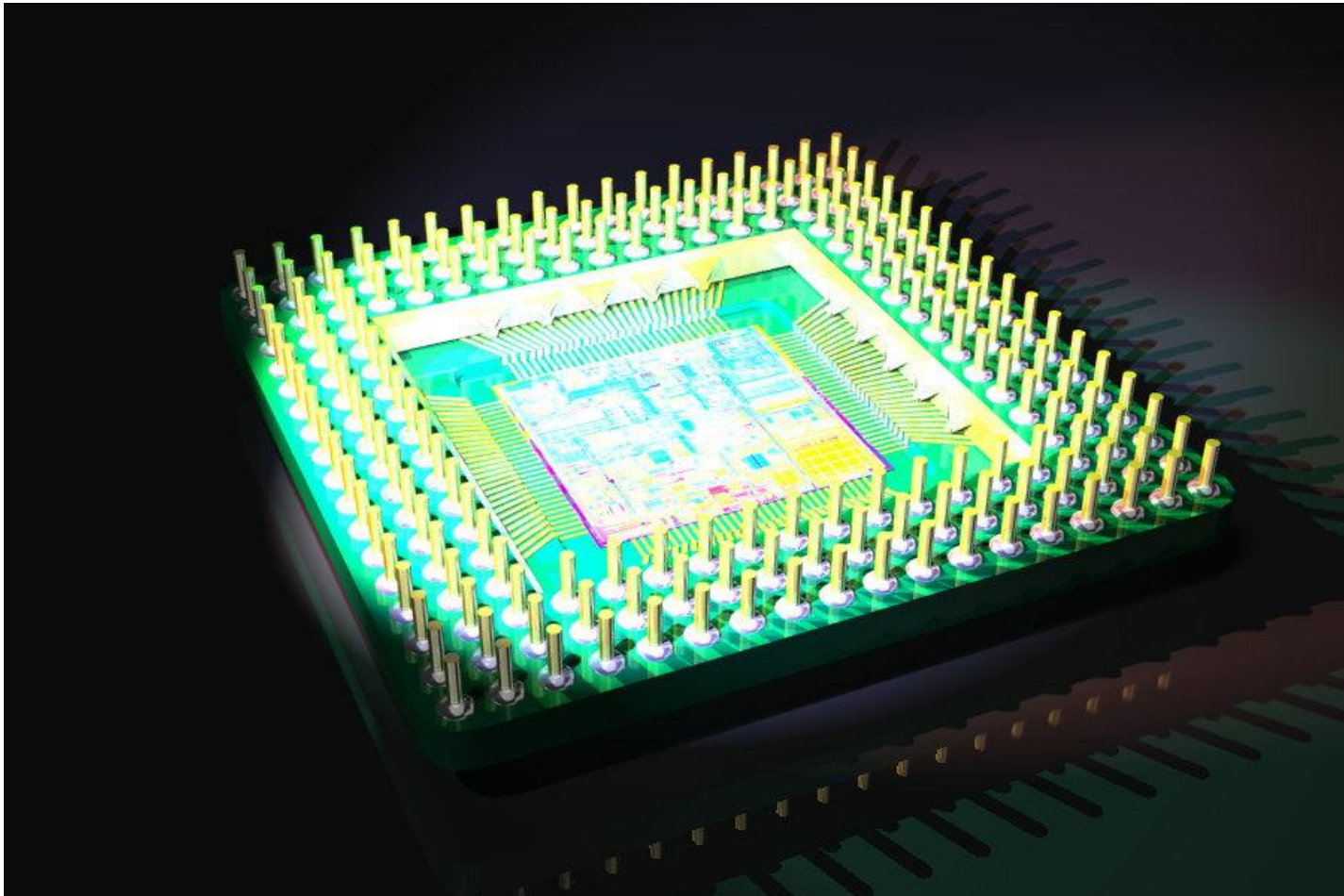
3.5 Khối số học và logic

3.6 Phân tích mô hình máy tính đơn giản

3.2 Bộ xử lý trung tâm CPU (Central Processing Unit)



3.2 Bộ xử lý trung tâm CPU (Central Processing Unit)



3.2 Bộ xử lý trung tâm CPU (Central Processing Unit)

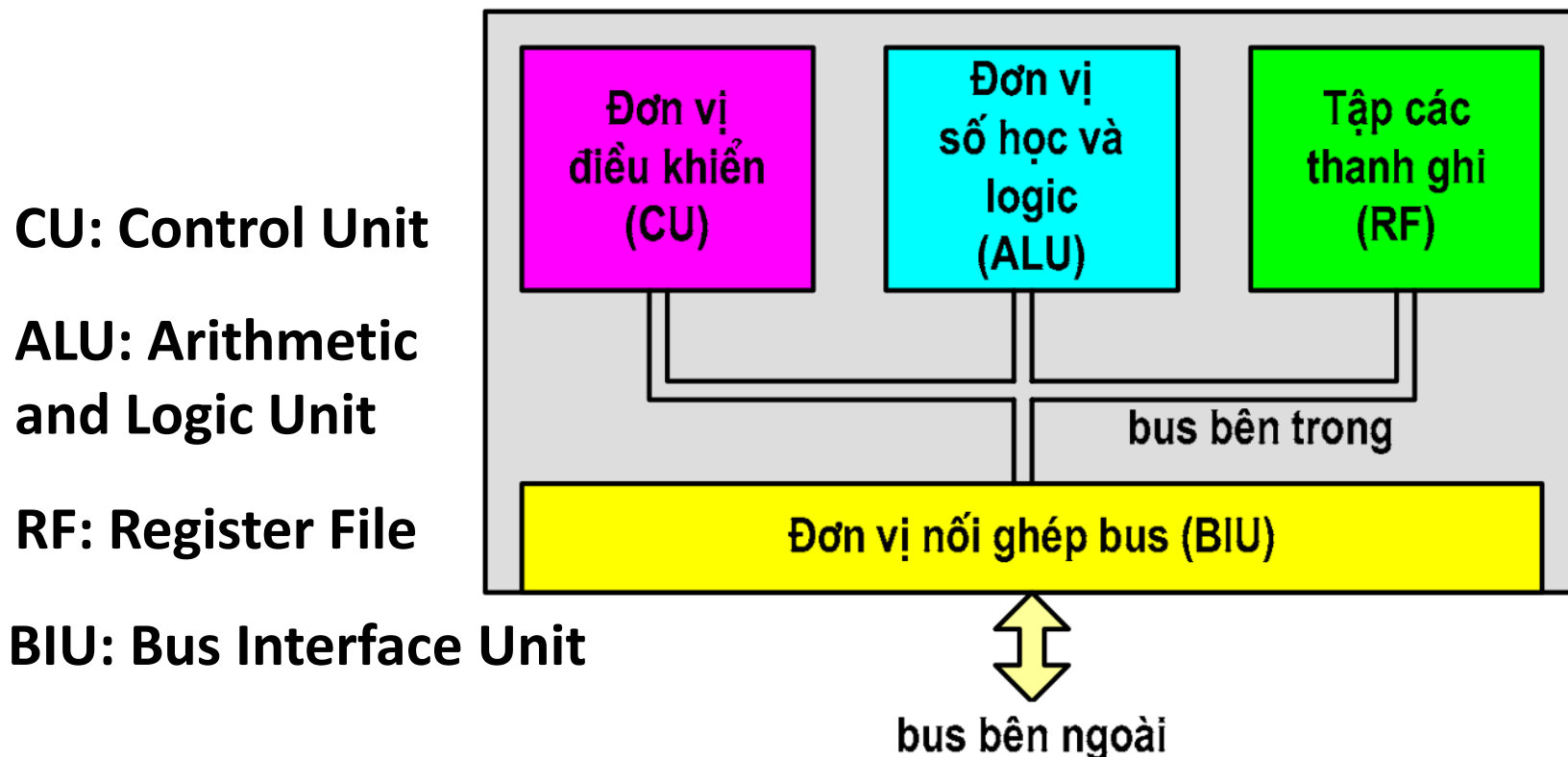
- Chức năng:
 - Điều khiển hoạt động của máy tính
 - Xử lý dữ liệu
- Nguyên tắc hoạt động cơ bản:
 - CPU hoạt động theo chương trình nằm trong bộ nhớ chính

3.2 Bộ xử lý trung tâm CPU (Central Processing Unit)

- Nhiệm vụ của CPU:
 - Nhận lệnh (**Fetch Instruction**): CPU đọc lệnh từ bộ nhớ
 - Giải mã lệnh (**Decode Instruction**): xác định thao tác mã 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 kết quả vào bộ nhớ hay thiết bị vào/ra

3.2 Bộ xử lý trung tâm CPU (Central Processing Unit)

- Cấu trúc cơ bản của CPU:



Bus ngoài chứa thông tin gì ?

3.2 Bộ xử lý trung tâm CPU (Central Processing Unit)

- **Đơn vị điều khiển** (*Control Unit - CU*): điều khiển hoạt động của máy tính theo chương trình định sẵn
- **Đơn vị số học và logic** (*Arithmetic and Logic Unit - ALU*): thực hiện các phép toán số học và logic
- **Tập thanh ghi** (*Register File - RF*): lưu trữ các thông tin tạm thời phục vụ cho hoạt động của CPU
- **Đơn vị phối ghép bus** (*Bus Interface Unit - BIU*): kết nối và trao đổi thông tin giữa bus bên trong (**Internal bus**) và bus bên ngoài (**External bus**)

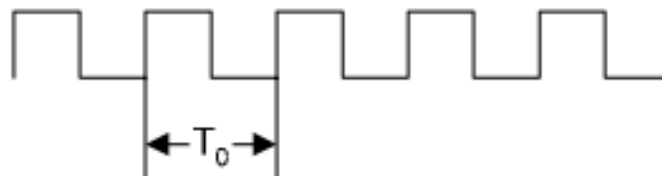
3.2 Bộ xử lý trung tâm CPU (Central Processing Unit)

- **Tốc độ của bộ xử lý:**
 - Số lệnh được thực hiện trong 1 giây: MIPS (Million of Instructions per Second)
 - Khó đánh giá chính xác
- **Tần số xung nhịp của bộ xử lý**
 - Bộ xử lý hoạt động theo một xung nhịp (clock) có tần số xác định
 - Tốc độ của bộ xử lý được đánh giá gián tiếp thông qua tần số xung nhịp

3.2 Bộ xử lý trung tâm CPU (Central Processing Unit)

- Tốc độ bộ xử lý:

- Dạng xung nhịp



- T_0 : chu kỳ xung nhịp
 - Tần số xung nhịp: $f_0 = 1/T_0$
 - Mỗi thao tác của bộ xử lý cần kT_0
 - T_0 càng nhỏ \rightarrow bộ xử lý chạy càng nhanh
 - Ví dụ: Máy tính dùng bộ xử lý 2GHz

Ta có $f_0 = 2\text{GHz} = 2 \times 10^9 \text{Hz}$

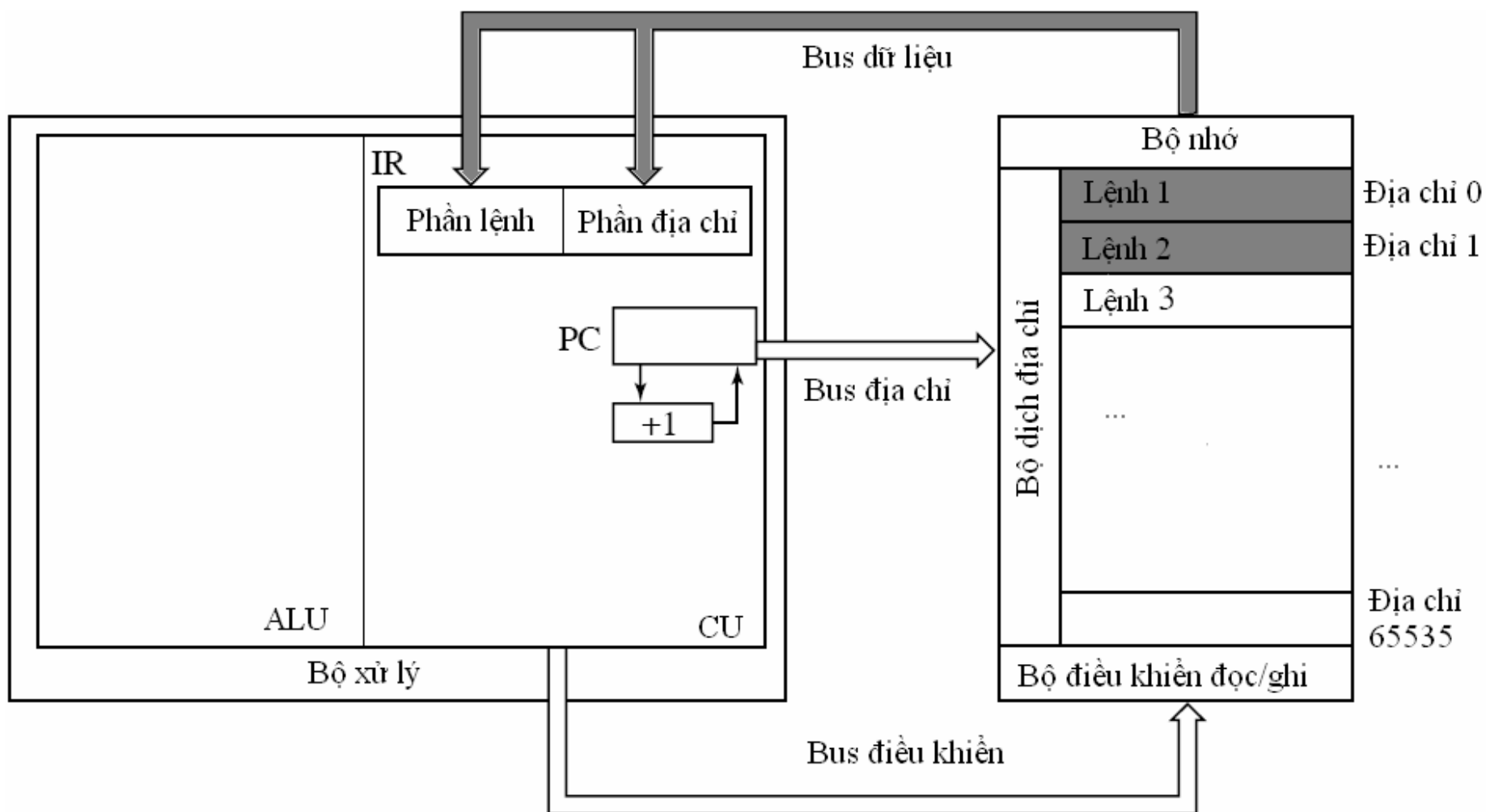
$\rightarrow T_0 = 1/f_0 = 1/(2 \times 10^9) = 0,5 \text{ ns.}$

3.2 Bộ xử lý trung tâm CPU (Central Processing Unit)

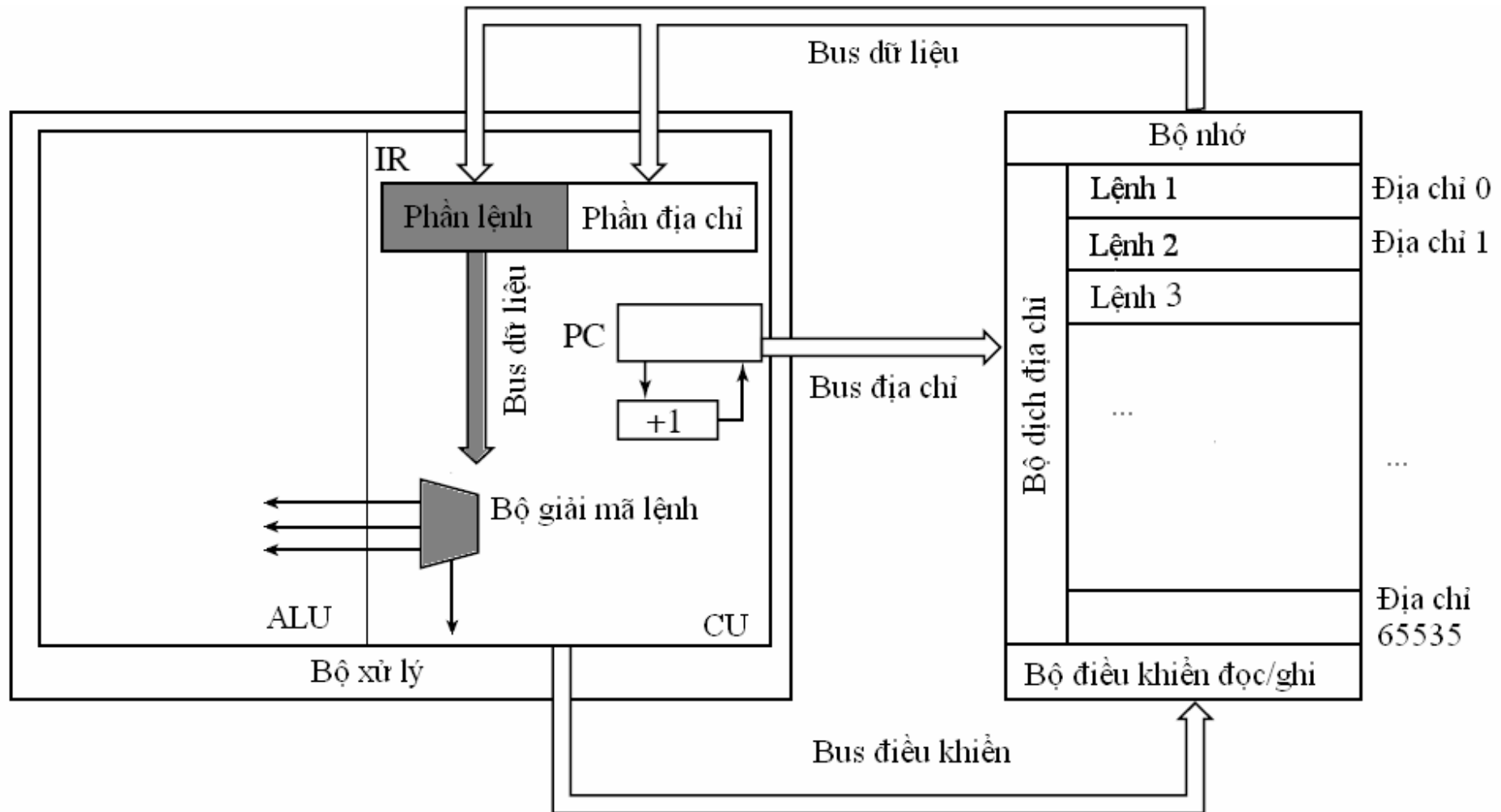
Sự thực thi các chỉ thị (Instruction)

- ❶ Lấy chỉ thị tiếp theo từ bộ nhớ đặt vào thanh ghi chỉ thị IR
- ❷ Thay đổi bộ đếm chương trình để trở tới chỉ thị tiếp theo
- ❸ Xác định kiểu của chỉ thị vừa lấy về
- ❹ Nếu chỉ thị sử dụng dữ liệu trong bộ nhớ thì xác định vị trí của dữ liệu
- ❺ Lấy dữ liệu về (nếu có), đặt vào các thanh ghi
- ❻ Thi hành chỉ thị
- ❼ Chứa kết quả vào nơi thích hợp

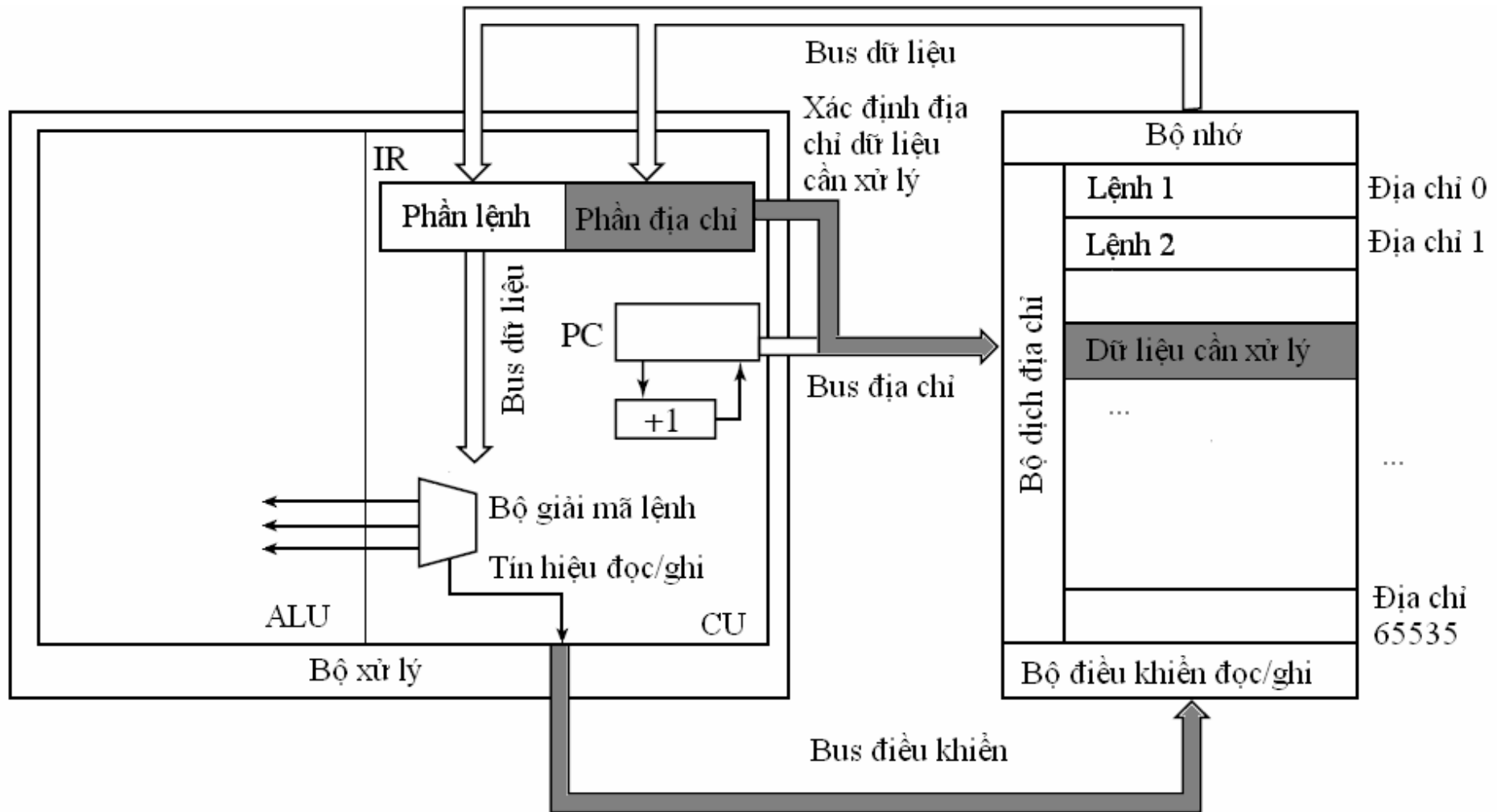
Thực thi các lệnh trong CPU



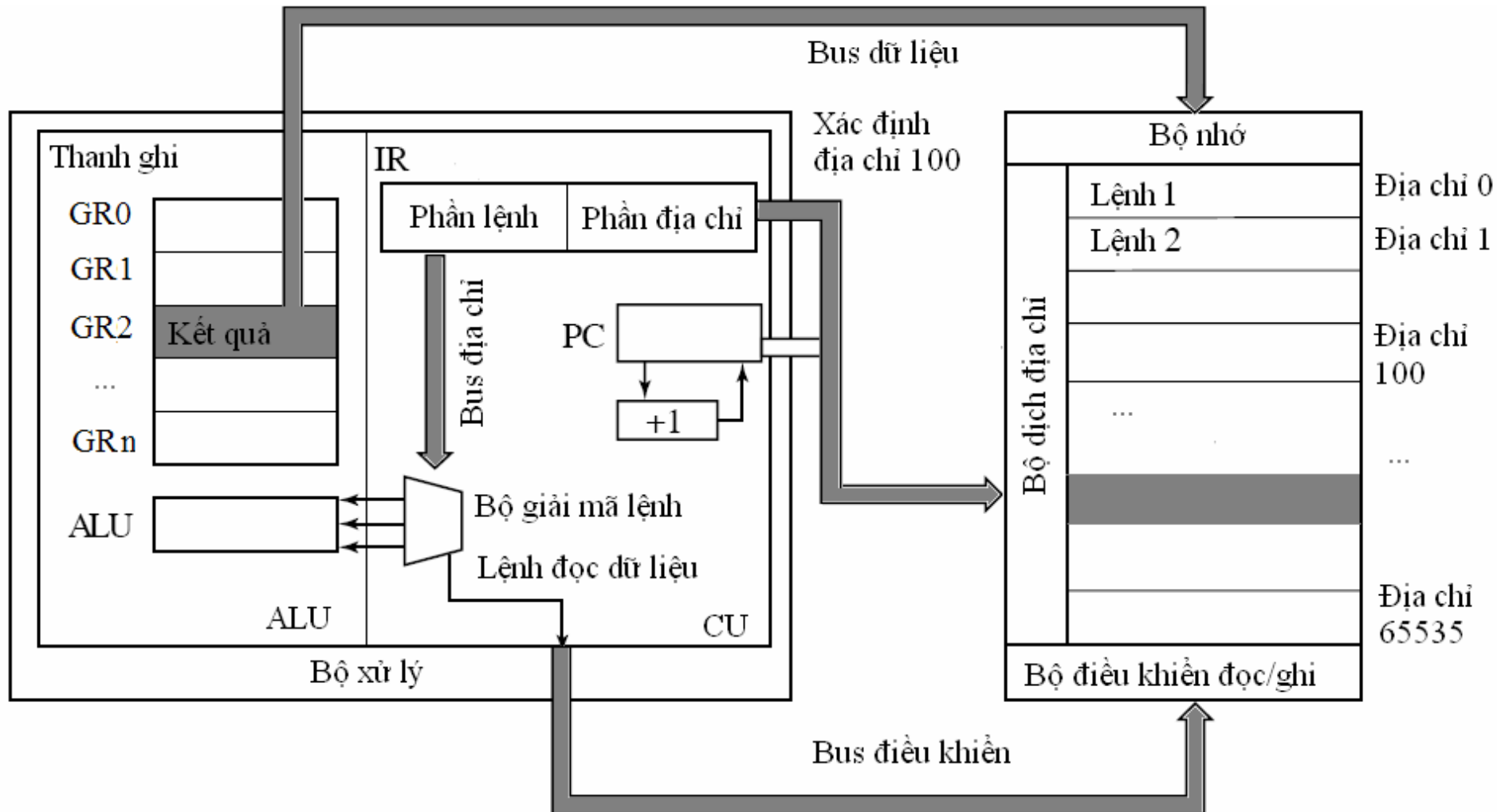
Thực thi các lệnh trong CPU



Thực thi các lệnh trong CPU



Thực thi các lệnh trong CPU



Thực thi các lệnh trong CPU

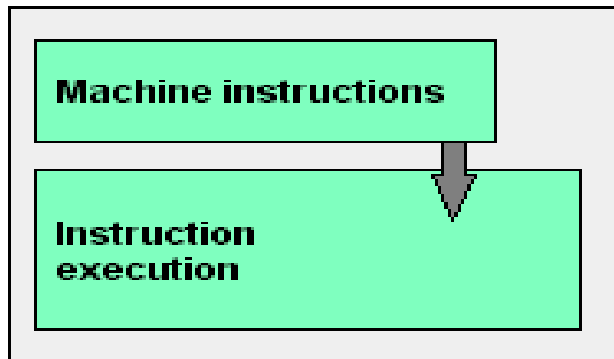
- **Tập chỉ thị (Instruction set):** là các chỉ thị mà người lập trình ở một mức có thể sử dụng
 - Kích thước: vài trăm – hàng triệu chỉ thị
 - Tập chỉ thị phức tạp chưa hẳn là tốt hơn tập chỉ thị đơn giản
 - Trình biên dịch ngôn ngữ lập trình bậc cao làm việc tốt hơn trên máy có tập chỉ thị đơn giản
- Có 2 loại máy: **CISC và RISC**
 - **CISC** – Complex Instruction Set Computer
 - **RISC** – Reduce Instruction Set Computer

So sánh CISC và RISC

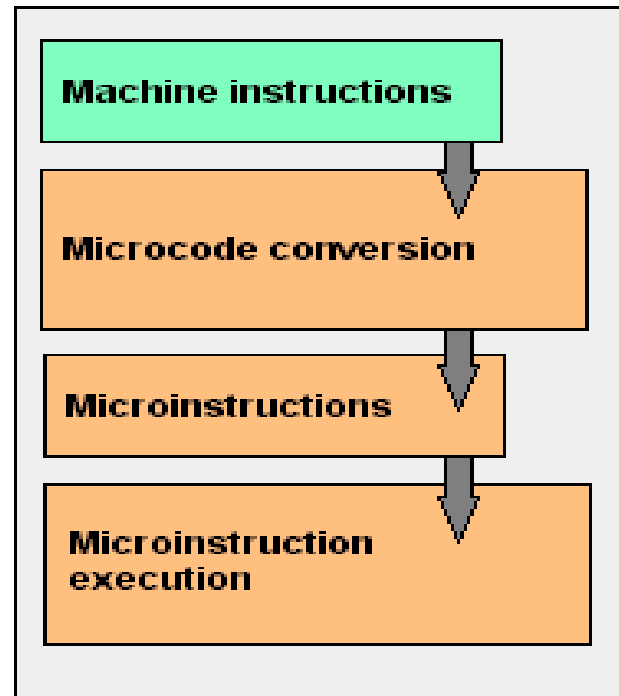
RISC	CISC
<ul style="list-style-type: none">• Một lệnh được thực thi bởi việc kết hợp một số chỉ lệnh đơn giản• Các chỉ lệnh được thực thi bởi phần cứng• Kích thước các chỉ lệnh và thời gian thực hiện các chỉ lệnh là bằng nhau	<ul style="list-style-type: none">• Các chỉ lệnh ở dạng phức hợp• Các chỉ lệnh được phân tích nhỏ thành vi chỉ lệnh để thực thi nhờ vi chương trình• Độ dài cũng như kích thước các chỉ lệnh thay đổi

So sánh CISC và RISC

RISC

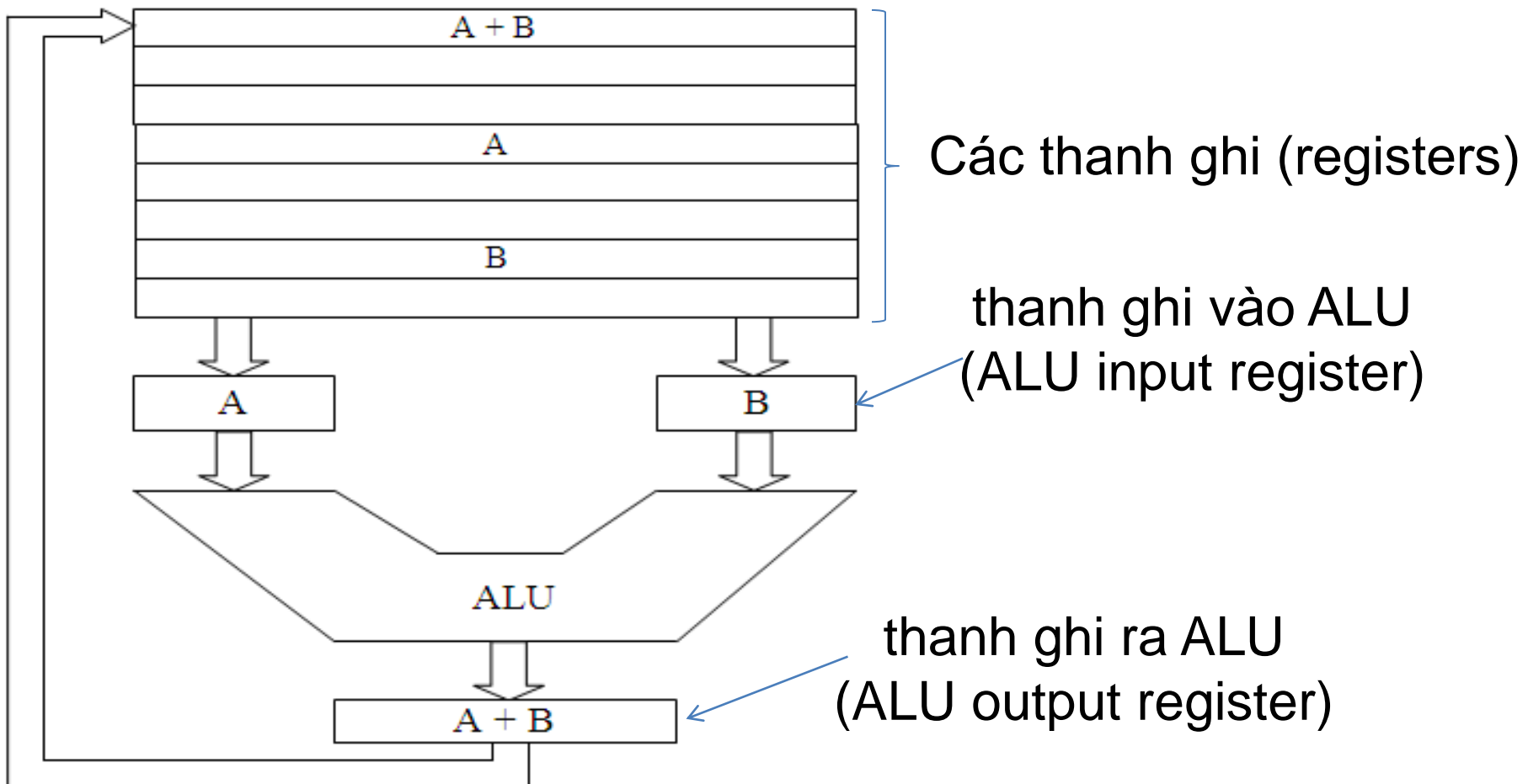


CISC

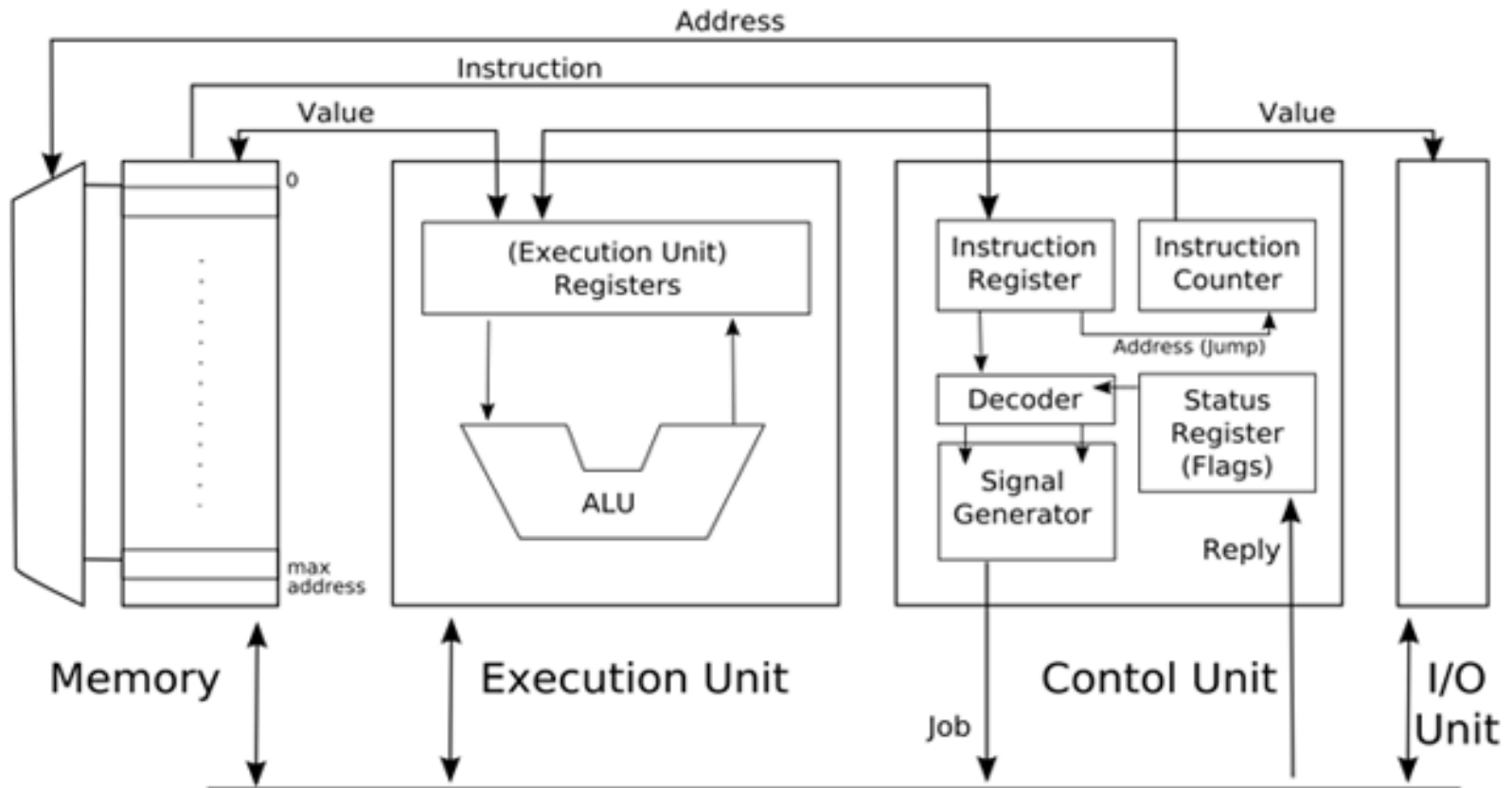


Tổ chức CPU

Sơ đồ đơn giản tổ chức CPU:



Tổ chức CPU



Tổ chức CPU

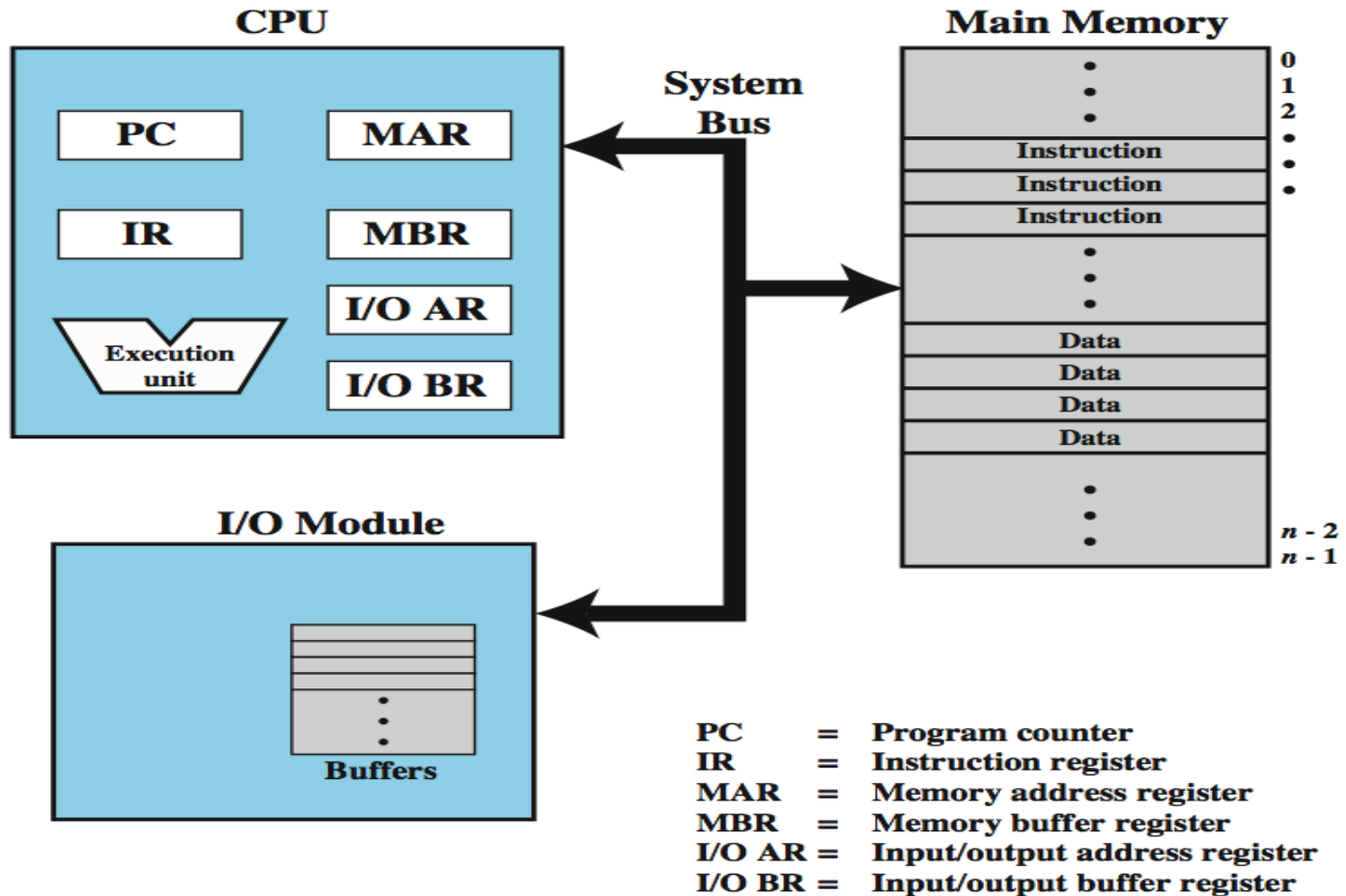


Figure 1.1 Computer Components: Top-Level View

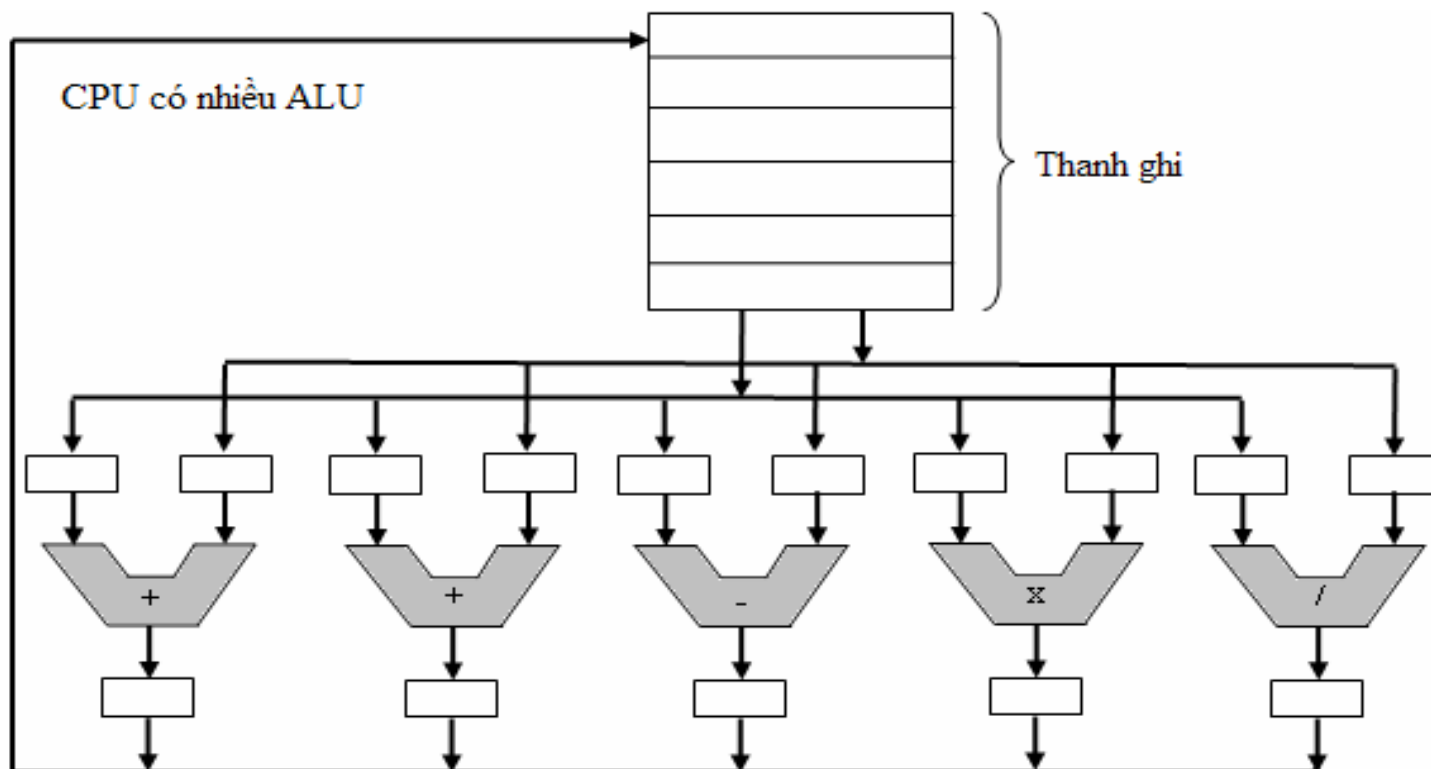
Các chế độ địa chỉ

- Là cách thức chỉ ra cho CPU lấy được dữ liệu, dựa vào vị trí của toán hạng có thể phân loại:
 - Chế độ địa chỉ thanh ghi – thanh ghi (Register-Register)
 - Chế độ địa chỉ thanh ghi – bộ nhớ (Register - Memory)
 - Chế độ địa chỉ bộ nhớ - bộ nhớ (Memory - Memory)

Thực hiện song song các lệnh

- Mục đích: nâng cao tốc độ hoạt động của máy tính, có 2 cách:
 - Nâng cao tốc độ hoạt động của phần cứng
 - Tổ chức hoạt thực hiện các lệnh song song
 - SISD (Single Instruction stream, Single Data stream)
 - SIMD (Single Instruction stream, Multiple Data stream)
 - MIMD (Multiple Instruction stream, Multiple Data stream)

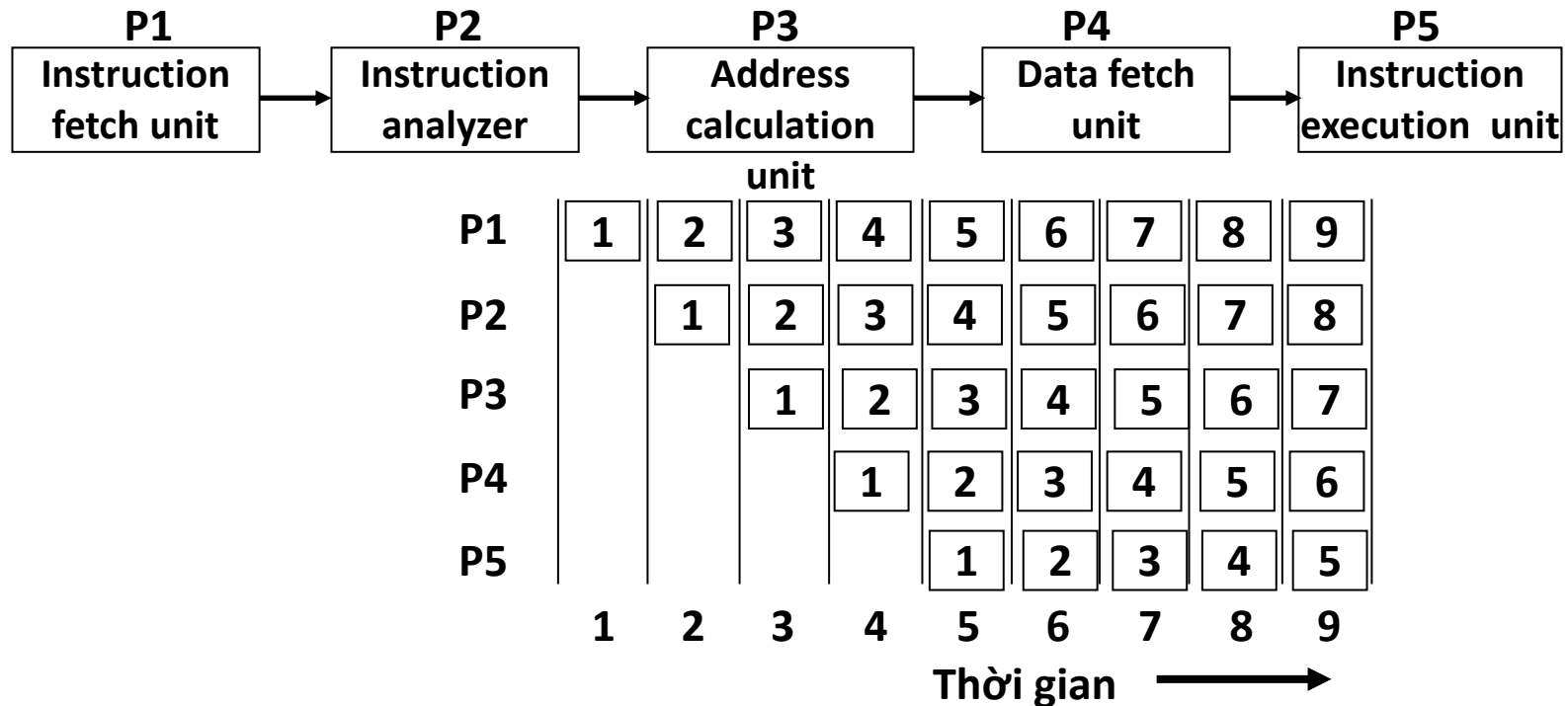
SISD – 1 đơn vị xử lý và 5 ALU



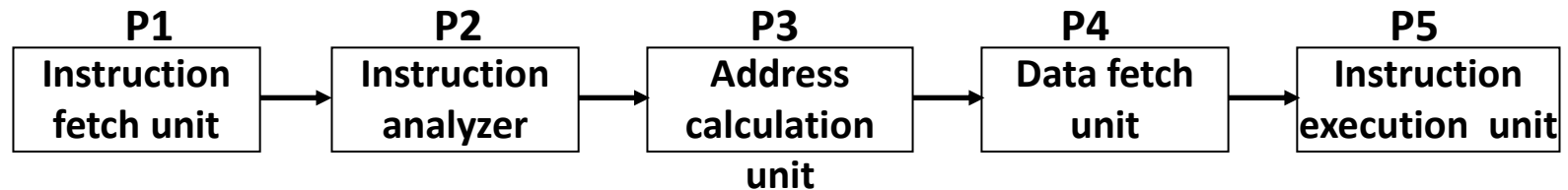
- Đơn vị điều khiển CU nhận về một lệnh rồi chuyển cho một trong các ALU thực hiện, đồng thời CU nhận về các lệnh tiếp theo, rồi chuyển cho một trong những ALU khác đang rỗi thực hiện, quá trình này chỉ dừng lại khi mọi ALU đã bận.
- Chiến lược làm việc này có hiệu quả khi thời gian thực hiện một lệnh dài hơn thời gian lấy lệnh

SISD – Cơ chế đường ống (Pipelining)

- Nguyên tắc: quá trình thực hiện lệnh được chia thành nhiều giai đoạn, mục đích tối ưu quá trình sử dụng phần cứng:

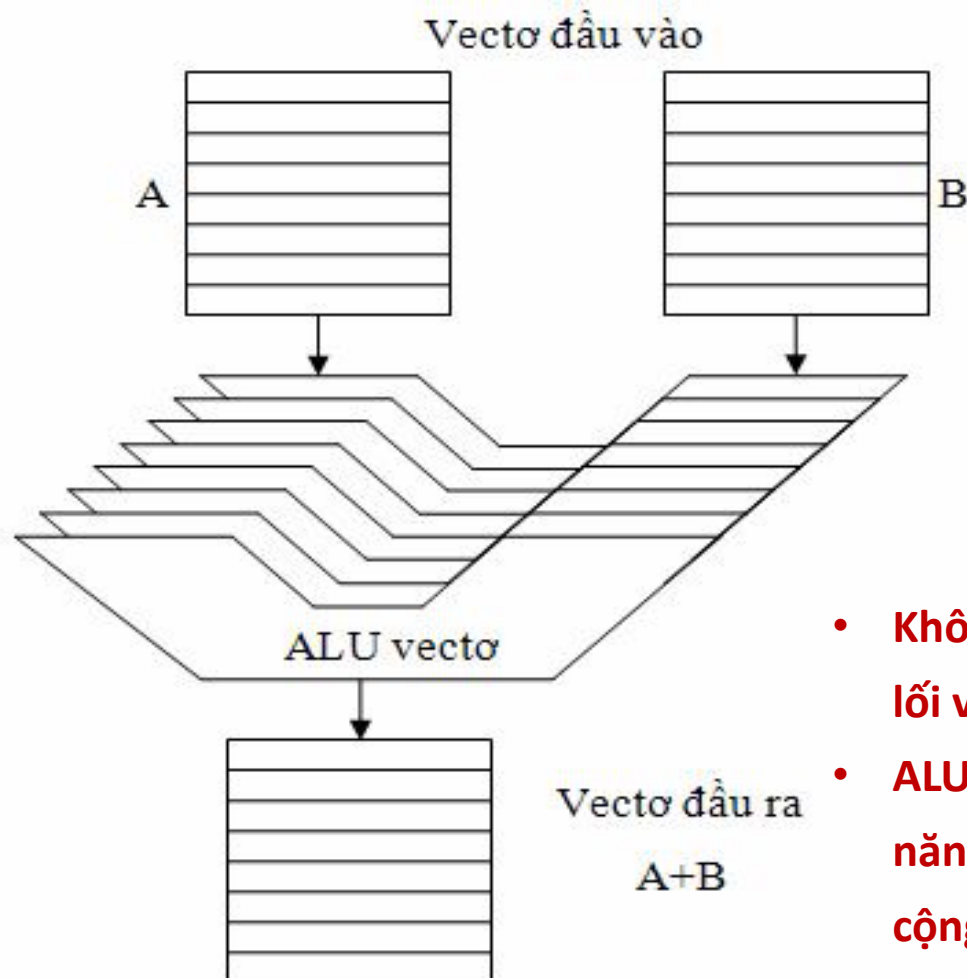


SISD – Cơ chế đường ống (Pipelining)



- + P1 nhận lệnh 1 từ bộ nhớ.
- + Lệnh 1 được chuyển cho P2 phân tích, P1 lấy lệnh 2.
- + Lệnh 1 chuyển cho P3 tính địa chỉ, P2 nhận lệnh 2 để phân tích, P1 nhận lệnh 3 từ bộ nhớ.
- + Lệnh 1 chuyển cho P4 để lấy dữ liệu, P3 nhận lệnh 2 để tính địa chỉ, P2 nhận lệnh 3 phân tích, P1 nhận lệnh 4.
- + Lệnh 1 chuyển cho P5 thực hiện, P4 nhận lệnh 2, P3 nhận lệnh 3, P2 nhận lệnh 4, P1 nhận lệnh 5

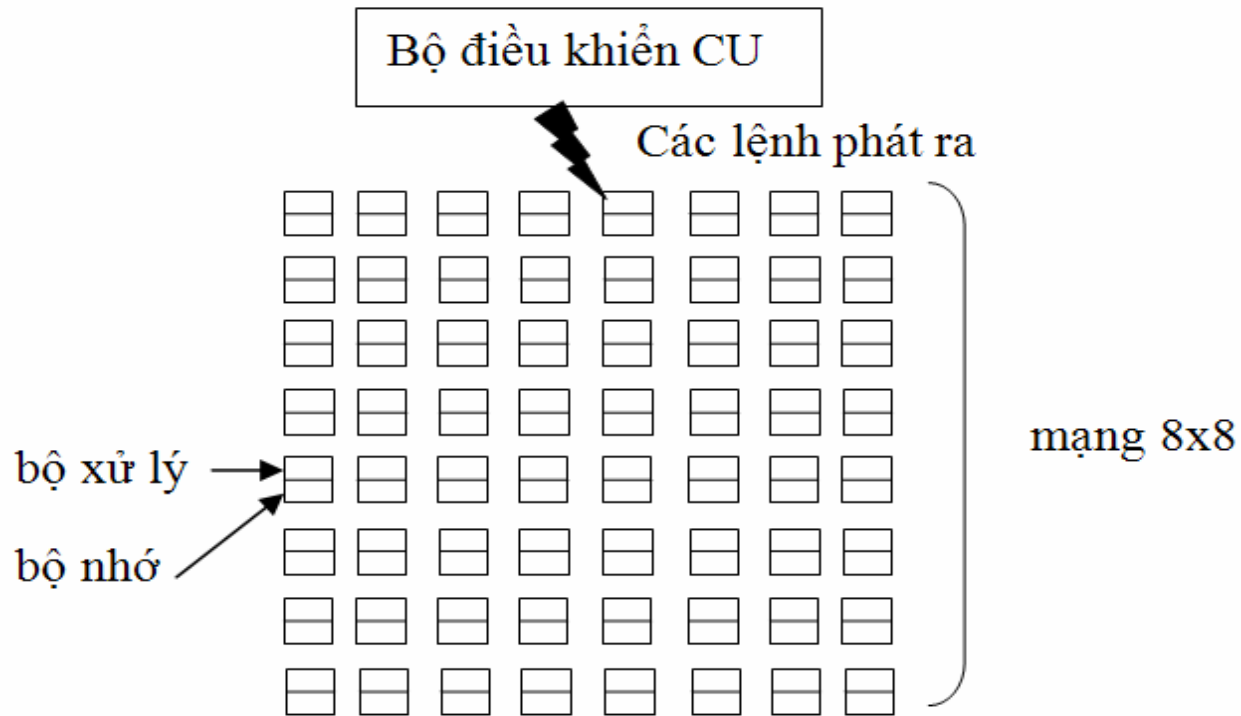
SIMD - Máy tính Vector



$$\begin{matrix} A \\ 4 \\ 6 \\ 3 \\ 2 \\ 7 \\ 8 \\ 3 \\ 8 \end{matrix} + \begin{matrix} B \\ 5 \\ 3 \\ 1 \\ 3 \\ 4 \\ 1 \\ 8 \\ 8 \end{matrix} = \begin{matrix} A+B \\ 9 \\ 9 \\ 4 \\ 5 \\ 11 \\ 9 \\ 11 \\ 16 \end{matrix}$$

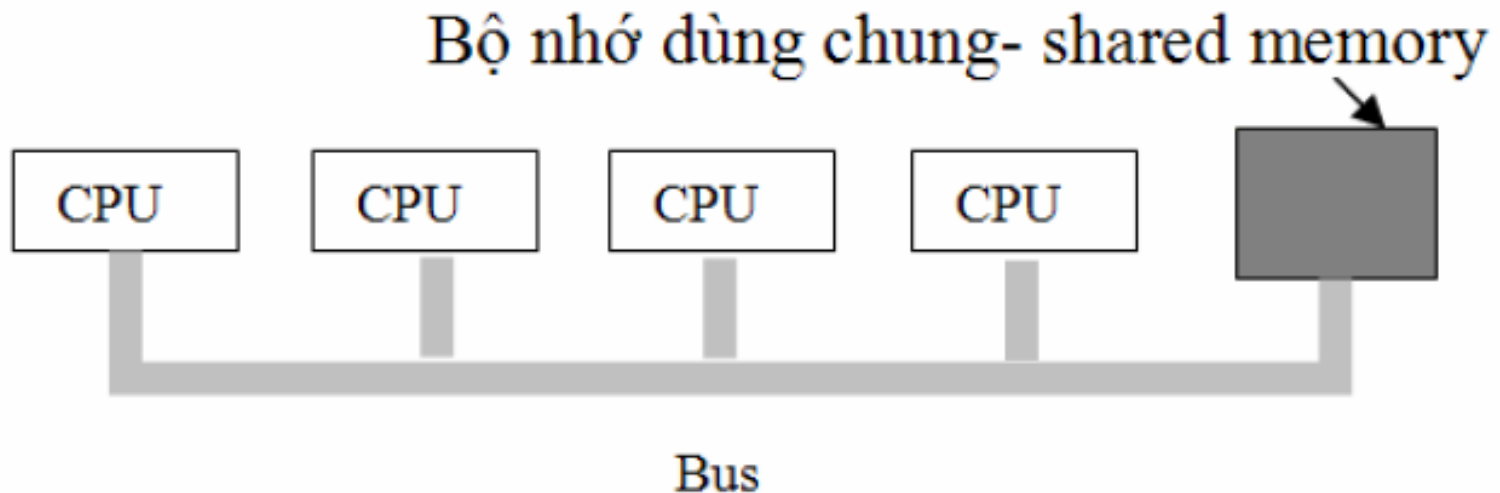
- Không phải chỉ có một biến đơn cho mỗi lối vào ALU mà có một vector n lối vào.
- ALU thực sự là một ALU vector, nó có khả năng thực hiện một phép toán như là phép cộng vector đối với 2 vector lối vào và đưa ra vector kết quả

SIMD: kiểu bộ xử lý mảng (Array processor)



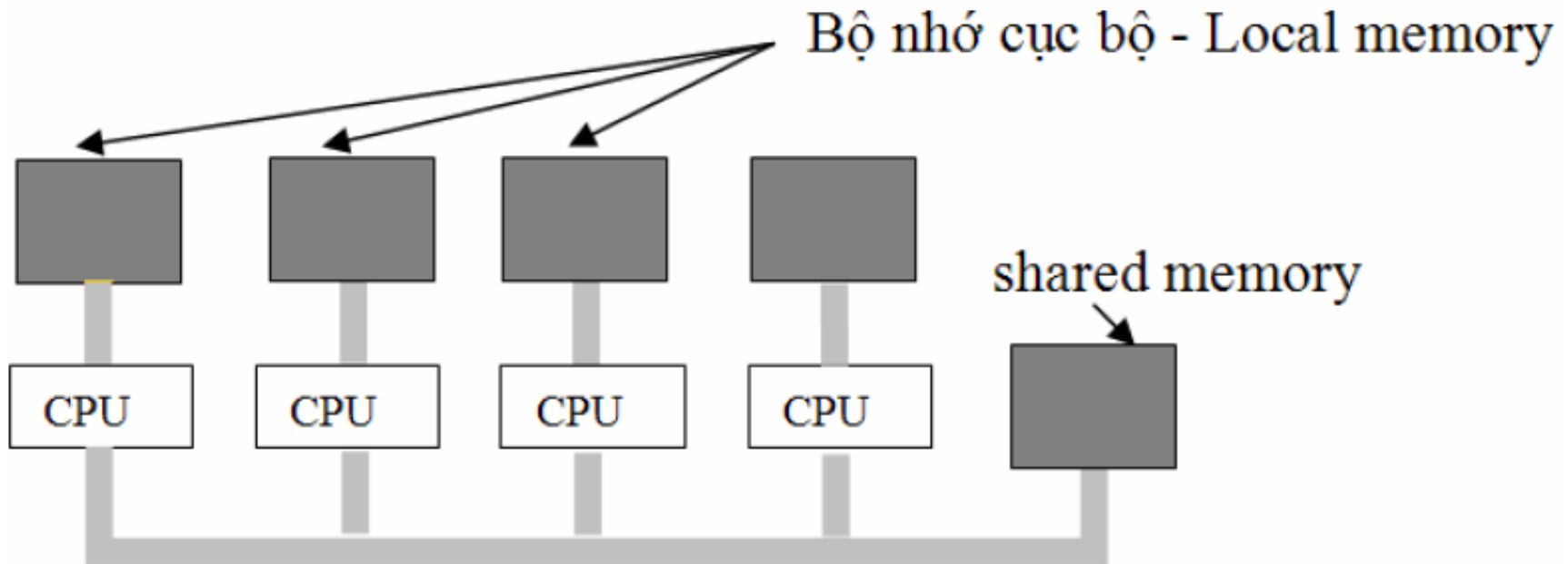
- Một mạng lưới các hình vuông các phần tử processor/memory
- Đơn vị CU phát ra các lệnh, các lệnh này được thực hiện từng bước bởi tất cả các bộ xử lý, mỗi bộ xử lý sử dụng dữ liệu riêng lấy từ bộ nhớ riêng của nó
- Bộ xử lý mảng thích hợp với các tính toán ma trận

MIMD - kiểu 1 bộ nhớ dùng chung



- Máy tính loại này có nhiều CPU, mỗi CPU thực hiện một chương trình khác nhau
- Các CPU có thể cùng chia sẻ một vùng nhớ chung (được gọi là shared memory).

MIMD - kiểu nhiều bộ nhớ



- Mỗi bộ xử lý được trang bị một bộ nhớ cục bộ riêng, các bộ xử lý khác không truy cập được.
- Bộ nhớ cục bộ chứa các đoạn mã chương trình hoặc các phần tử dữ liệu không được sử dụng chung.
- Các bộ xử lý khi truy cập bộ nhớ cục bộ riêng của mình không sử dụng bus chính dùng chung, nhờ vậy làm giảm rất nhiều việc truyền thông trên bus chính.

Nội dung

3.1 Thành phần của hệ thống máy tính

3.2 Tổ chức của CPU

3.3 Thanh ghi

3.4 Khối điều khiển

3.5 Khối số học và logic

3.6 Phân tích mô hình máy tính đơn giản

3.3 Tập các thanh ghi (Registers)

- Cấu trúc: là các FF mắc nối tiếp nhau, thanh ghi n bit chứa được n bit thông tin
- Chức năng và đặc điểm:
 - 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
 - Số lượng thanh ghi nhiều => tăng hiệu năng của CPU
 - Có 2 loại thanh ghi: **thanh ghi lập trình được** và **thanh ghi không lập trình được**

Phân loại các thanh ghi

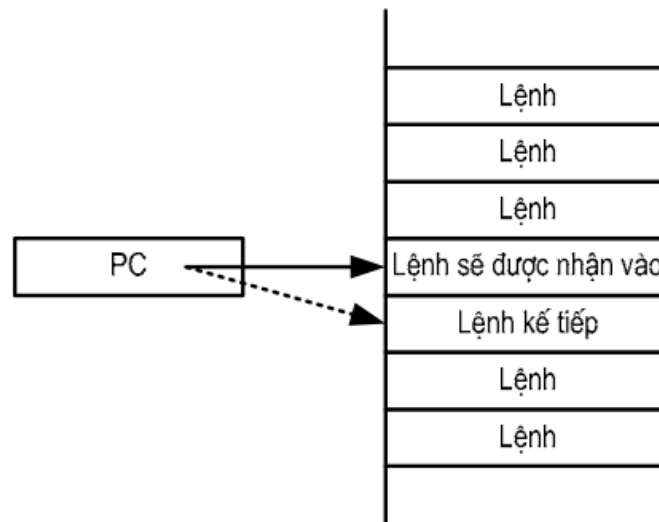
- Thanh ghi địa chỉ (Address Register): quản lý địa chỉ của ô nhớ hay địa chỉ của cổng vào/ra
- Thanh ghi dữ liệu (Data Reg): chứa tạm thời các dữ liệu
- Thanh ghi đa năng (Acc): có thể chứa địa chỉ hoặc dữ liệu
- Thanh ghi điều khiển/trạng thái (Control/ Status Reg): chứa các thông tin điều khiển và trạng thái của CPU
- Thanh ghi lệnh (Intruction Reg): chứa lệnh đang thực hiện

Một số thanh ghi điển hình

- Các thanh ghi địa chỉ
 - Bộ đếm chương trình PC (Program counter)
 - Con trỏ dữ liệu DP (Data Pointer)
 - Con trỏ ngăn xếp SP (Stack Pointer)
 - Thanh ghi cơ sở & thanh ghi chỉ số (Base Reg & Index Reg)
- Các thanh ghi dữ liệu
- Các thanh ghi trạng thái: Thanh ghi cờ

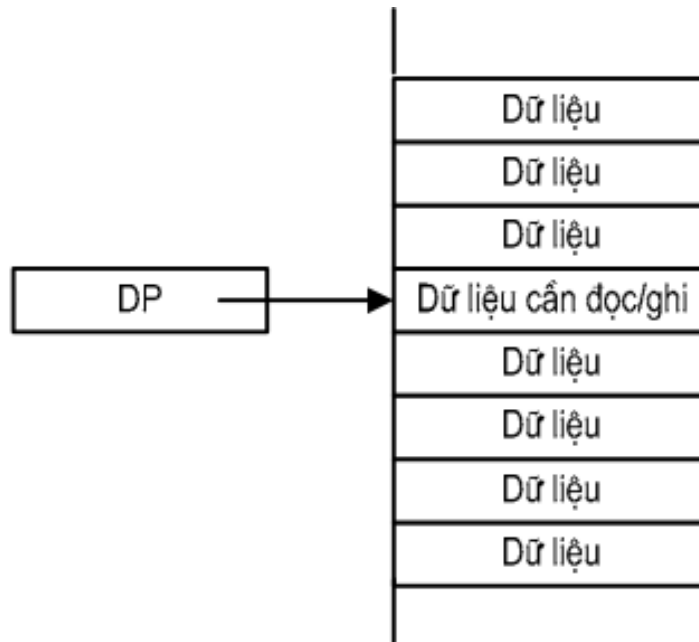
Bộ đếm chương trình PC

- Còn gọi là con trỏ lệnh IP (Instruction Pointer)
- Giữ địa chỉ của lệnh tiếp theo sẽ thực hiện, sau khi một lệnh được nhận vào, nội dung của PC tự động tăng để trỏ sang lệnh kế tiếp



Thanh ghi con trỏ dữ liệu

- Chứa địa chỉ của ô nhớ mà CPU muốn truy cập
- Thường có một số thanh ghi con trỏ dữ liệu



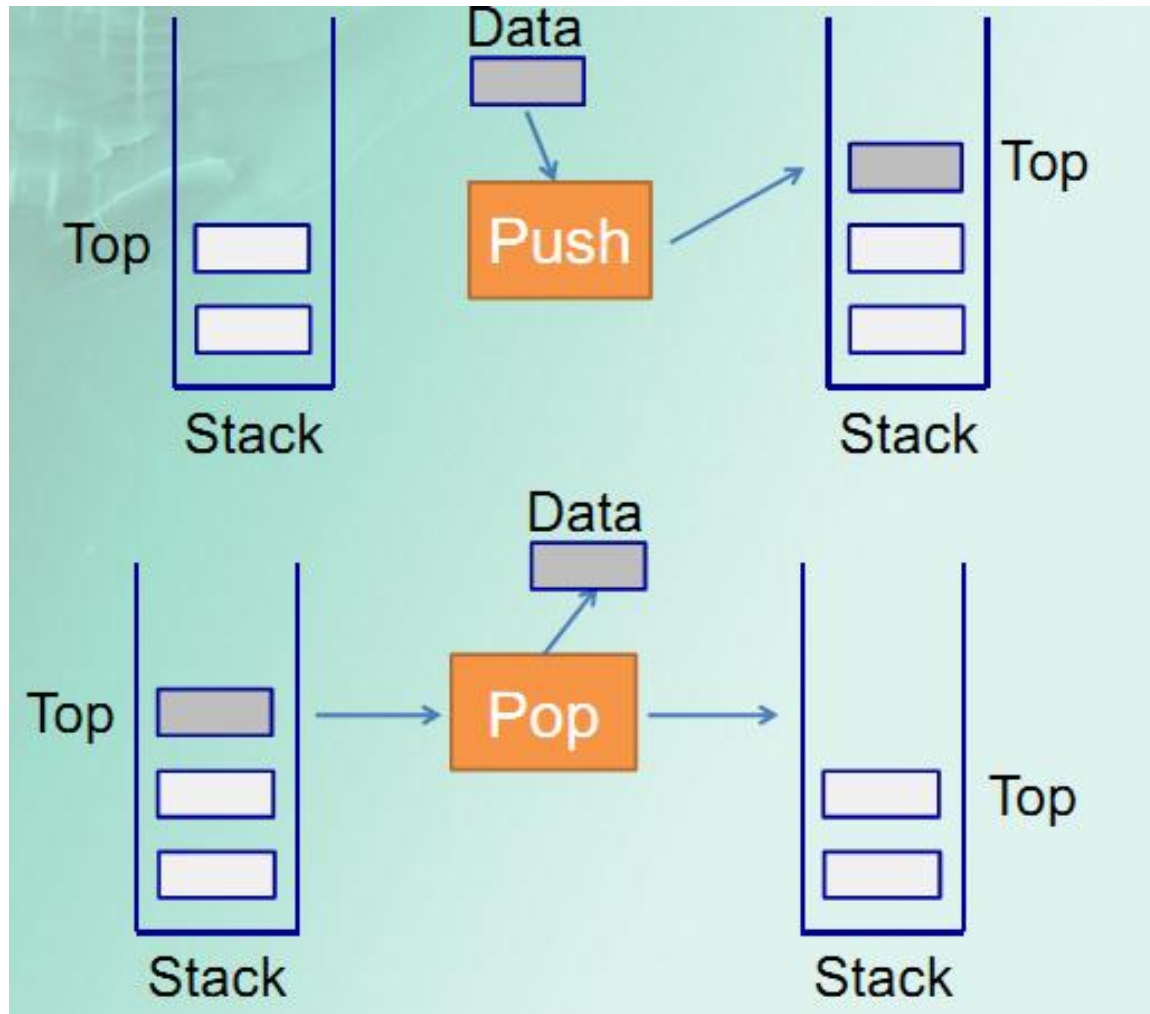
Ngăn xếp (Stack)

- Ngăn xếp là vùng nhớ có cấu trúc LIFO (Last In – First Out), thường dùng để phục vụ cho 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

Con trỏ ngăn xếp SP (Stack pointer)

- Chứa địa chỉ ở ngăn nhớ đỉnh ngăn xếp
- Khi cất một thông tin vào ngăn xếp:
 - Nội dung của SP tự động giảm
 - Thông tin được cất vào ngăn nhớ được trỏ bởi SP
- Khi lấy một thông tin ra khỏi ngăn xếp:
 - Thông tin được đọc từ ngăn nhớ được trỏ bởi SP
 - Nội dung của SP tự động tăng
- Khi ngăn xếp rỗng, SP trỏ vào đáy

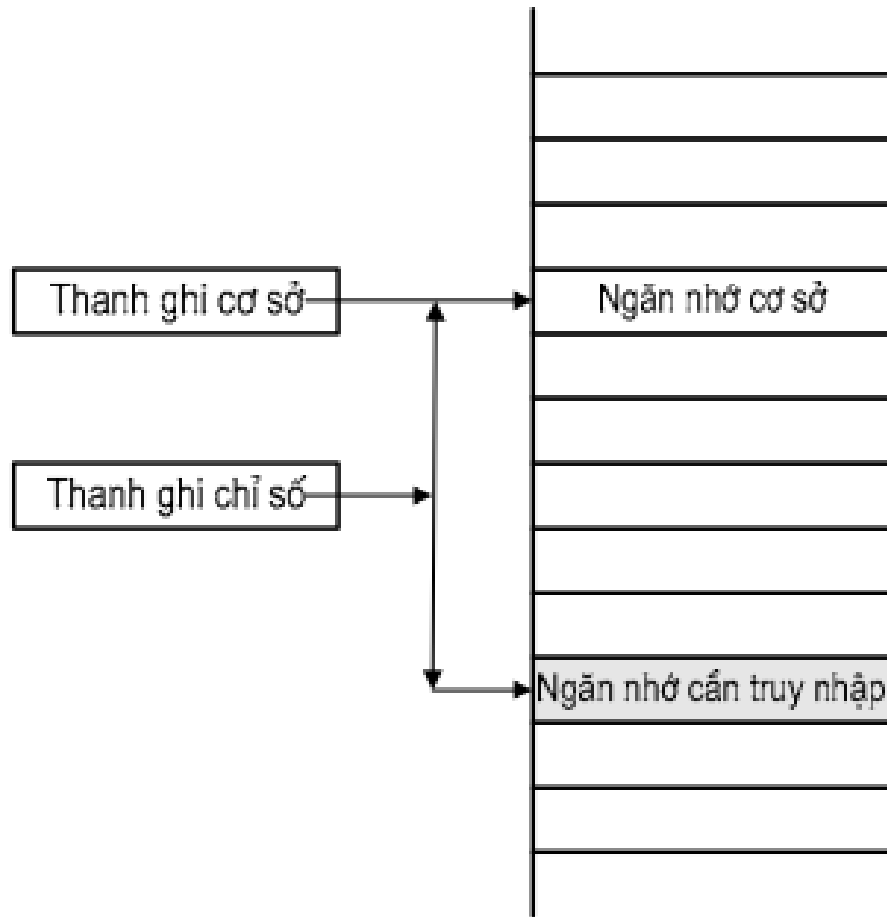
Ví dụ con trỏ ngăn xếp



Thanh ghi cơ sở và thanh ghi chỉ số

- **Thanh ghi cơ sở:** chứa địa chỉ của ô nhớ cơ sở (địa chỉ cơ sở)
- **Thanh ghi chỉ số:** chứa độ lệch địa chỉ giữa ô nhớ mà CPU cần truy cập so với ô nhớ cơ sở (chỉ số)
- **Địa chỉ của ô nhớ cần truy cập = địa chỉ cơ sở + chỉ số**

Ví dụ thanh ghi cơ sở và chỉ số



Các thanh ghi dữ liệu

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

Thanh ghi trạng thái (Status Reg)

- Còn gọi là thanh ghi cờ (Flag Reg)
- Chứa các thông tin trạng thái của CPU sau mỗi lệnh được thực hiện
 - 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
 - Ví dụ thanh ghi cờ:

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
S	Z	X	AC	X	P	X	CY

Tập thanh ghi của một số bộ xử lý

Data Registers	
D0	
D1	
D2	
D3	
D4	
D5	
D6	
D7	

Address Registers	
A0	
A1	
A2	
A3	
A4	
A5	
A6	
A7	
A7'	

Program Status	
Program Counter	
Status Register	

(a) MC68000

General Registers

AX	Accumulator
BX	Base
CX	Count
DX	Data

Pointer & Index

SP	Stack Pointer
BP	Base Pointer
SI	Source Index
DI	Dest Index

Segment

CS	Code
DS	Data
SS	Stack
ES	Extra

Program Status

Instr Ptr
Flags

(b) 8086

General Registers

EAX	AX
EBX	BX
ECX	CX
EDX	DX

ESP	SP
EBP	BP
ESI	SI
EDI	DI

Program Status

FLAGS Register
Instruction Pointer

(c) 80386 - Pentium II

Nội dung

3.1 Thành phần của hệ thống máy tính

3.2 Tổ chức của CPU

3.3 Thanh ghi

3.4 Khối điều khiển

3.5 Khối số học và logic

3.6 Phân tích mô hình máy tính đơn giản

3.4 Khối điều khiển (Control Unit - CU)

Chức năng:

- Điều khiển nhận lệnh từ bộ nhớ đưa vào thanh ghi lệnh
- Tăng nội dung PC để trở sang lệnh kế tiếp
- Giải mã lệnh đã được nhận 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 với các yêu cầu đó.

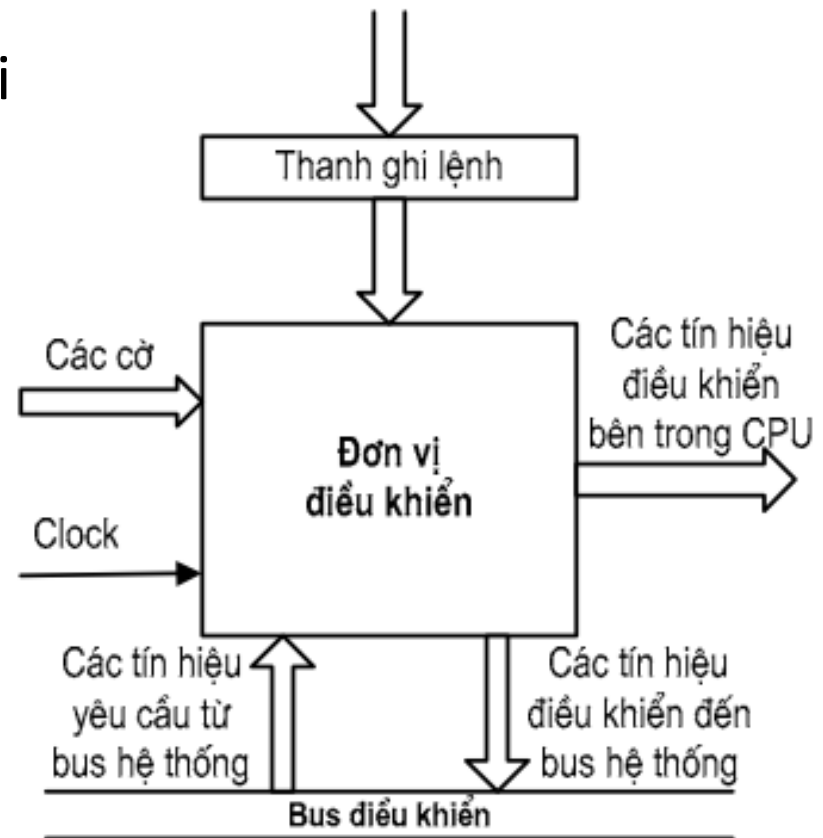
Mô hình kết nối đơn vị điều khiển

Vào:

- Clock: tín hiệu xung 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

Ra:

- Các tín hiệu điều khiển bên trong CPU (điều khiển Reg, ALU)
- Các tín hiệu điều khiển bên ngoài CPU (Mem, I/O)

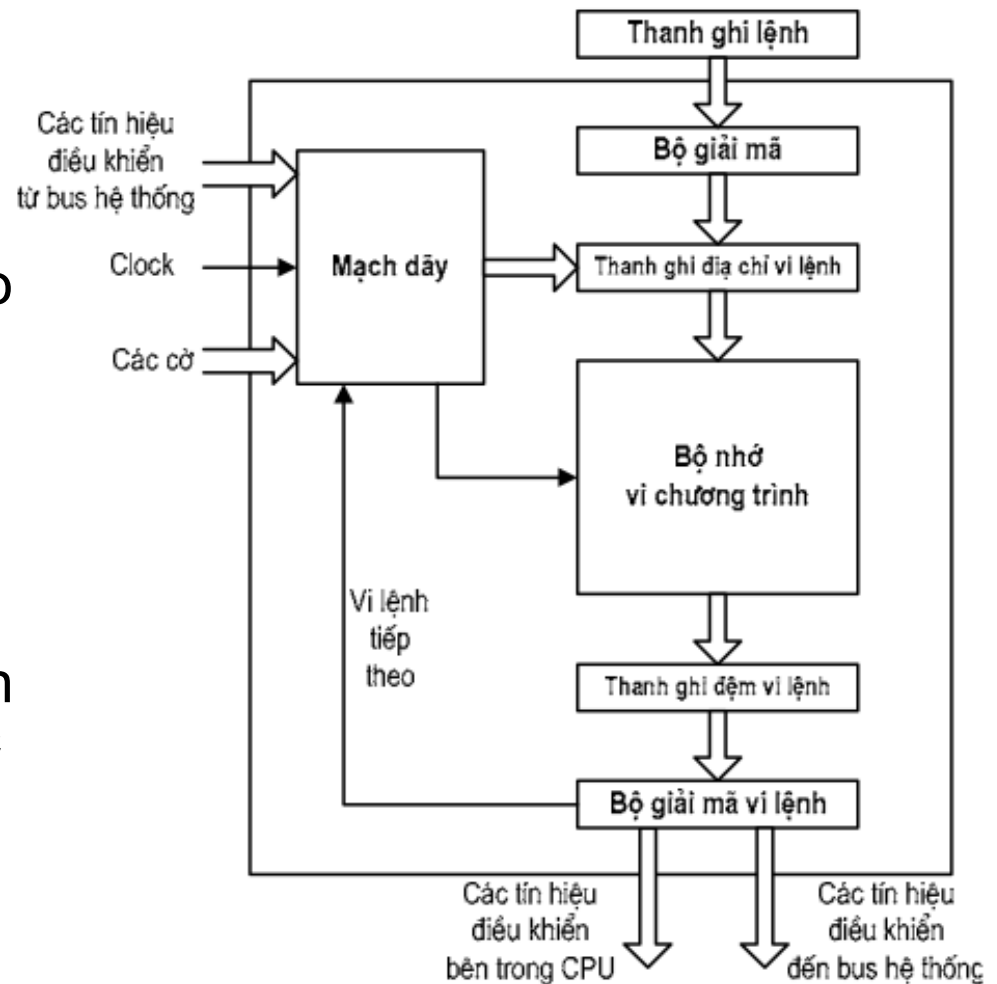


Các phương pháp thiết kế đơn vị điều khiển

- Đơn vị điều khiển vi chương trình (Microprogrammed Control Unit)
- Đơn vị điều khiển kết nối cứng (Hardwired Control Unit)

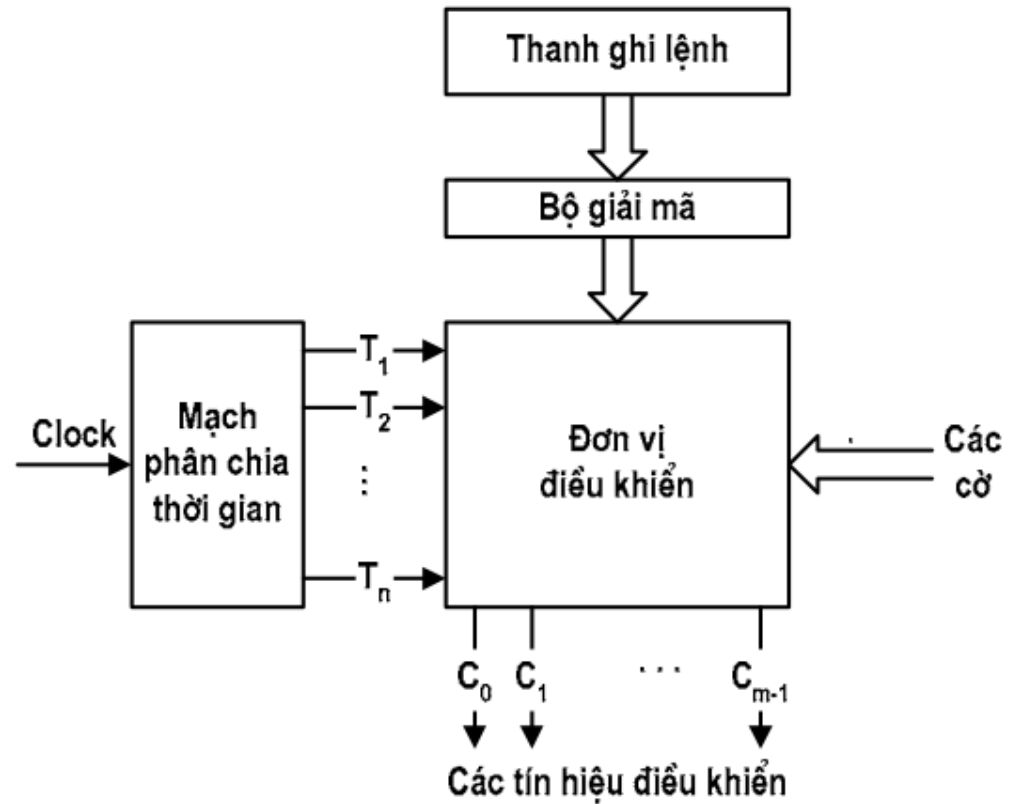
Đơn vị điều khiển vi chương trình

- Bộ nhớ vi chương trình (ROM) lưu trữ các vi chương trình (microprogram)
- Một vi chương trình bao gồm các vi lệnh (microinstruction)
- Mỗi vi lệnh mã hoá cho một vi thao tác (microoperation)
- Để hoàn thành một lệnh cần thực hiện một hoặc một vài vi chương trình
- Tốc độ chậm



Đơn vị điều khiển kết nối cứng

- Sử dụng mạch cứng để giải mã và tạo các tín hiệu điều khiển thực hiện lệnh
- Tốc độ nhanh
- Đơn vị điều khiển phức tạp



Nội dung

3.1 Thành phần của hệ thống máy tính

3.2 Tổ chức của CPU

3.3 Thanh ghi

3.4 Khối điều khiển

3.5 Khối số học và logic

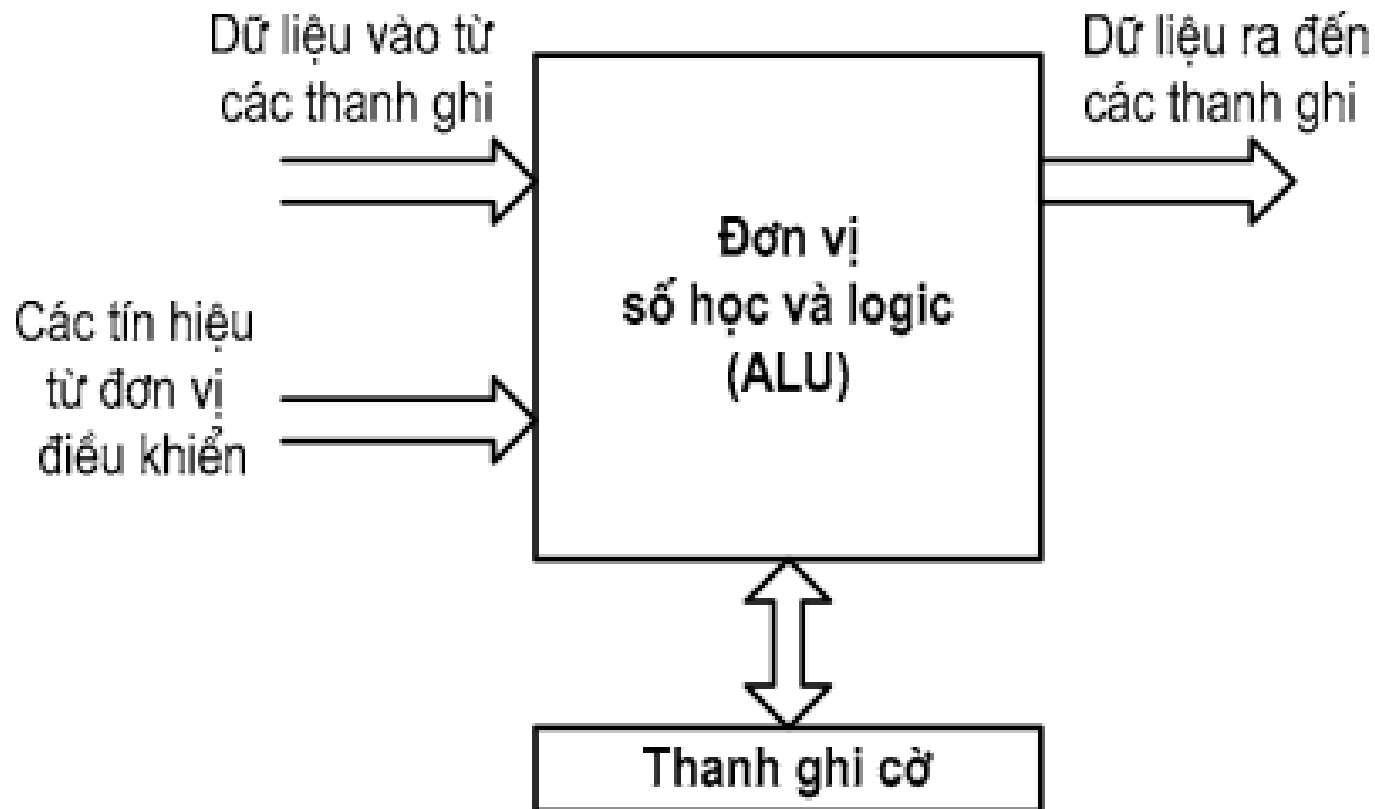
3.6 Phân tích mô hình máy tính đơn giản

3.5 Khối số học và logic (Arithmetic and Logic Unit - ALU)

Chức năng:

- Thực hiện các phép toán số học và phép toán logic
 - Số học: cộng, trừ, nhân, chia, tăng, giảm, đảo dấu
 - Logic: AND, OR, XOR, NOT, dịch bit

Mô hình kết nối ALU



Nội dung

3.1 Thành phần của hệ thống máy tính

3.2 Tổ chức của CPU

3.3 Thanh ghi

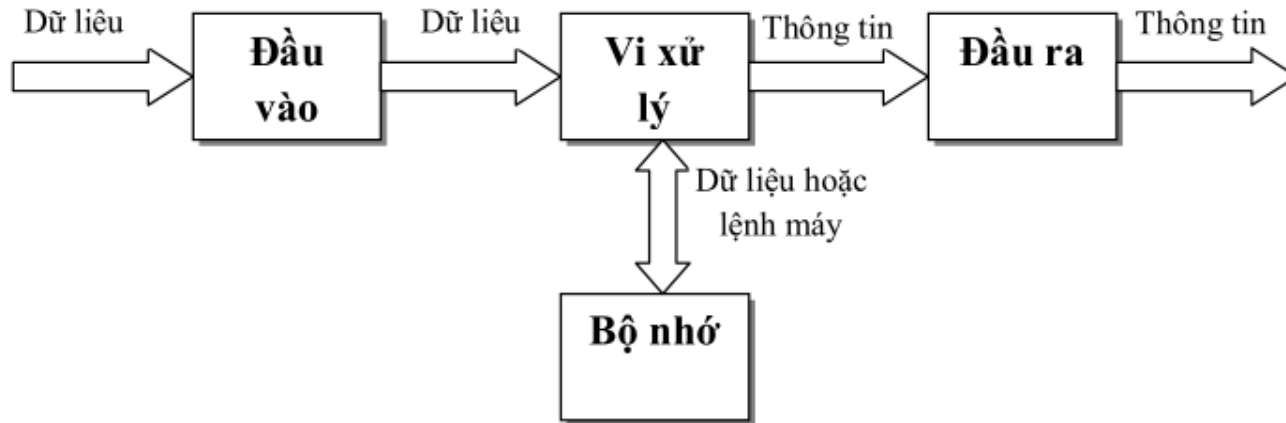
3.4 Khối điều khiển

3.5 Khối số học và logic

3.6 Phân tích mô hình máy tính đơn giản

3.6 Mô hình hệ thống máy tính đơn giản

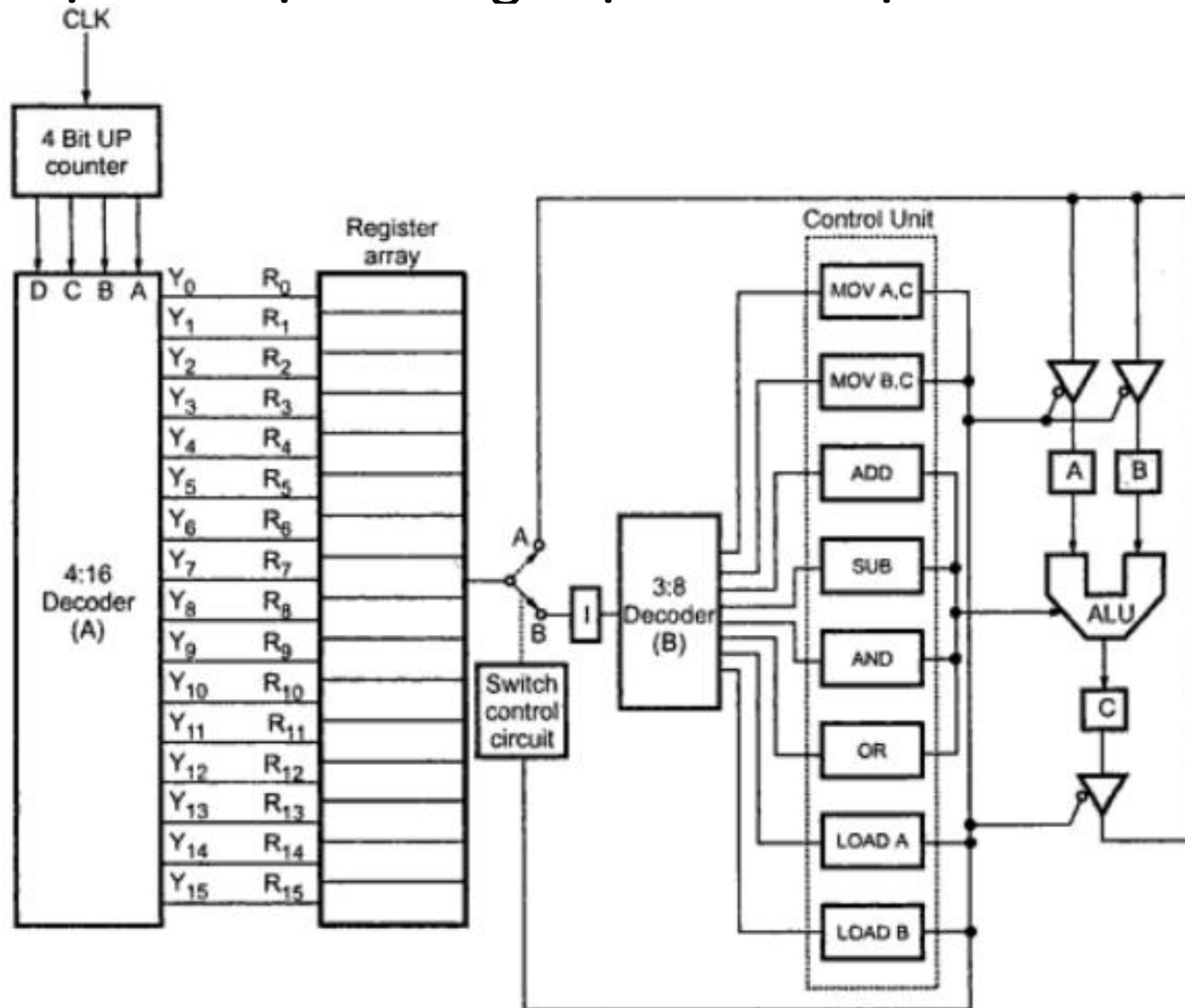
- Mô hình hệ thống máy tính đơn giản nhất:



- Hoạt động của mô hình hệ thống máy tính đơn giản:
 - Mô hình đơn giản là mạch số tuần tự: gồm **1 bộ đếm tăng 4 bit** (Counter), **bộ giải mã** (decoder), **các thanh ghi** (registers), **khối chuyển mạch** (Switch Unit), **khối điều khiển** (Control Unit), **khối số học và logic** (ALU-Arithmetic logic unit)

3.6 Mô hình hệ thống máy tính đơn giản

- Thực hiện câu lệnh bằng mạch tuần tự số:

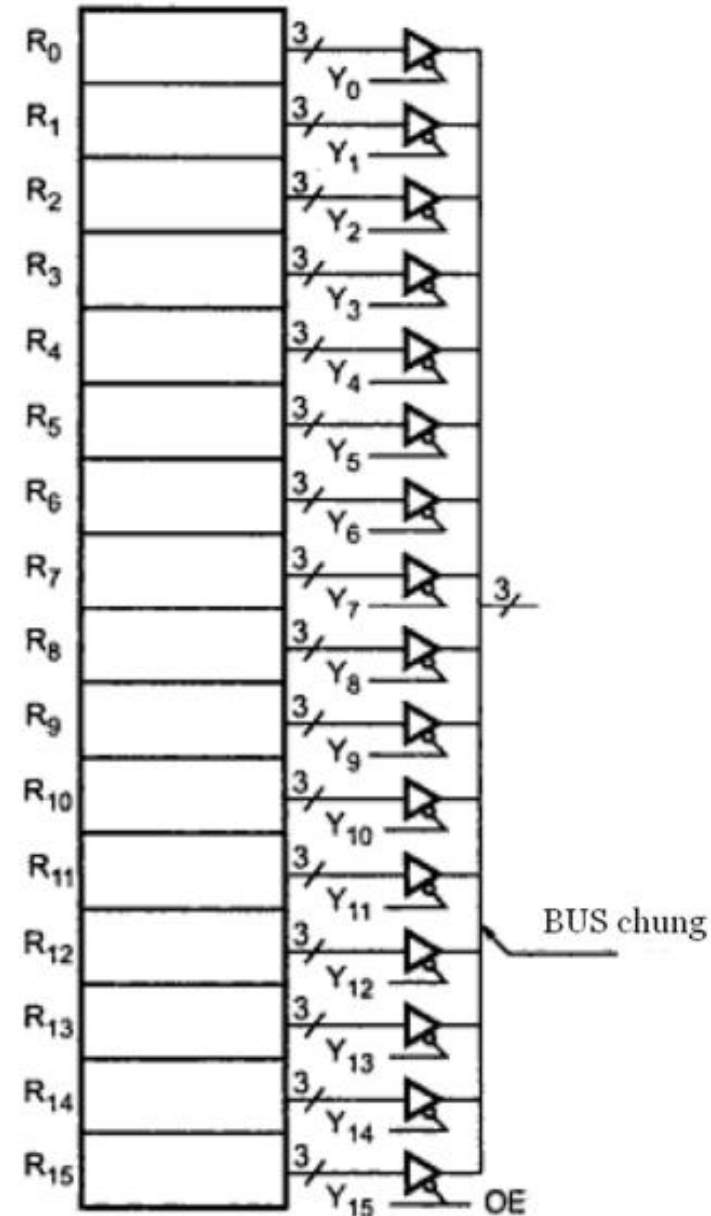


3.6 Mô hình hệ thống máy tính đơn giản

- Bộ đếm (Counter):
 - Đếm tăng
 - Bộ đếm 4 bit (đếm từ 0-15)
 - Điều khiển bởi xung clock (CLK)
 - Tín hiệu ra của bộ đếm là tín hiệu vào cho bộ giải mã A
- Bộ giải mã A (Decoder):
 - Là bộ giải mã 4:16, trạng thái tích cực đầu ra phụ thuộc tín hiệu vào
 - Đầu ra dùng để chọn một trong số các thanh ghi nằm trong mảng thanh ghi
 - Tại một thời điểm chỉ chọn một thanh ghi trong mảng thanh ghi
- Mảng thanh ghi (Register Array):

3.6 Mô hình hệ thống máy tính đơn giản

- Cấu trúc mảng thanh ghi
 - Gồm 16 thanh ghi 3 bit
 - Mỗi thanh ghi có thể chứa 1 dữ liệu 3 bit
 - Đầu ra được nối với Bus chung thông qua tín hiệu cho phép ra OE (Output Enable)
 - Đầu ra của bộ giải mã A được sử dụng để tích cực tín hiệu OE cho 16 thanh ghi. Tại một thời điểm chỉ có 1 đầu ra của bộ giải mã là tích cực, như vậy chỉ có dữ liệu từ 1 thanh ghi ra Bus chung



3.6 Mô hình hệ thống máy tính đơn giản

- Bus chung: được nối với thanh ghi A, thanh ghi B và thanh ghi I thông qua chuyển mạch
 - Chuyển mạch tại A: Bus nối với A or B, ngược lại nối với thanh ghi I
 - Chuyển mạch tại B: dữ liệu từ thanh ghi I được **bộ giải mã B (3:8)** dùng để chọn mạch điều khiển phù hợp trong khối điều khiển như bảng:

D ₂	D ₁	D ₀	Mạch điều khiển được chọn
0	0	0	MOV A,C
0	0	1	MOV B,C
0	1	0	ADD
0	1	1	SUB
1	0	0	AND
1	0	1	OR
1	1	0	LOAD A
1	1	1	LOAD B

- Khối điều khiển gồm 8 mạch điều khiển:
 - ADD: $A+B \rightarrow C$
 - SUB: $A-B \rightarrow C$
 - ...
 - LOAD A: mở cổng vào cho thanh ghi A, chuyển mạch về vị trí A; LOAD B: mở cổng vào cho thanh ghi B, chuyển mạch về vị trí A
- ALU: lấy dữ liệu từ thanh ghi A&B, thực hiện phép toán theo điều khiển, kết quả lưu ở thanh ghi C

3.6 Mô hình hệ thống máy tính đơn giản

- Ban đầu: Bộ đếm=0 -> Bộ giải mã chọn thanh ghi R0 -> dữ liệu trong R0 được đưa ra Bus chung
- Ban đầu thì chuyển mạch tại vị trí B -> dữ liệu từ Bus chung được đưa vào thanh ghi lệnh (I - Instruction) và được giải mã; căn cứ vào dữ liệu, khối điều khiển lựa chọn lệnh và ALU thực hiện lệnh đó.
- Dữ liệu trong thanh ghi lệnh (I) sẽ quyết định lệnh nào được thực hiện, mỗi lệnh có một mã lệnh riêng gọi là **Operation Code** hay viết tắt là **Opcode**. Các Opcode cho các lệnh như sau:

Lệnh tác động	Mã lệnh		
	D ₂	D ₁	D ₀
MOV A,C	0	0	0
MOV B,C	0	0	1
ADD	0	1	0
SUB	0	1	1
AND	1	0	0
OR	1	0	1
LOAD A	1	1	0
LOAD B	1	1	1

3.6 Mô hình hệ thống máy tính đơn giản

- Các lệnh cơ bản: **mov, add, sub, and, or, load**
- Chú ý:
 - Với lệnh **load A** và **load B** thì dữ liệu từ Bus chung được nạp trực tiếp vào thanh ghi A, B tương ứng, lệnh này cần chuyển mạch nằm ở vị trí A
 - Sau mỗi lần thực hiện lệnh bộ đếm lại tăng lên 1, do đó bộ giải mã lựa chọn thanh ghi tiếp theo, dữ liệu trong thanh ghi được chọn được đưa ra Bus chung và quá trình lại lặp lại
- VXL thực hiện các lệnh phức tạp bằng cách chia nhỏ thành các lệnh đơn giản và thực hiện các lệnh đơn giản một cách tuần tự
- Ví dụ: VXL thực hiện cộng 2 số; các bước:
 - Bước 1: Nạp số đầu tiên vào thanh ghi A
 - Bước 2: Nạp số thứ 2 vào thanh ghi B
 - Bước 3: Cộng 2 số

3.6 Mô hình hệ thống máy tính đơn giản

- Quá trình thực hiện như sau:

- Opcode lệnh **load A** trong R0
- Số thứ nhất nạp trong R1
- Opcode lệnh **load B** trong R2
- Số thứ 2 nạp trong R3
- Opcode lệnh **ADD** trong R4

R ₀	1	1	0
R ₁	1	0	0
R ₂	1	1	1
R ₃	0	1	0
R ₄	0	1	0
R ₅	—		
R ₆	—		

- Hoạt động:

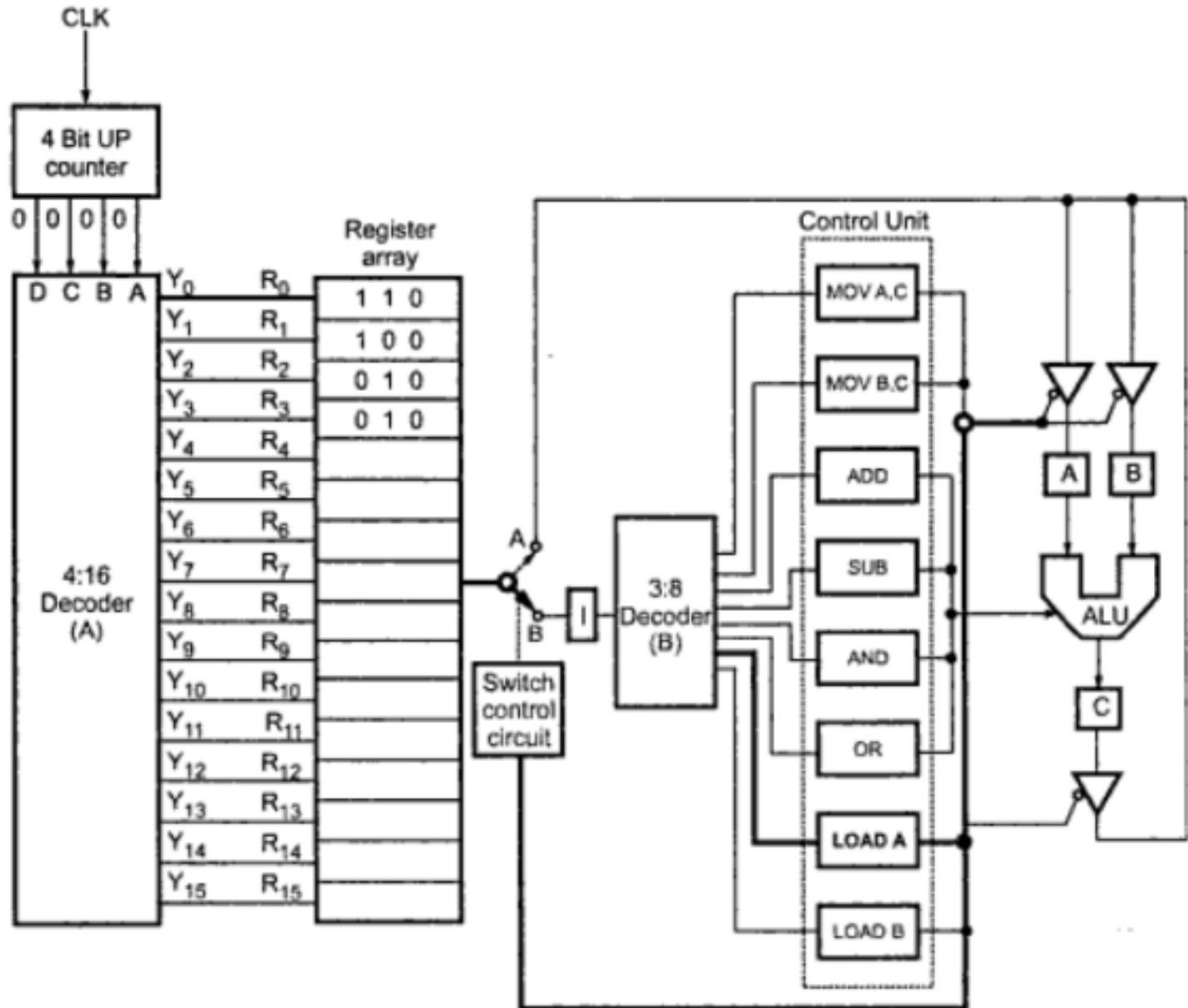
- B1: đầu ra bộ đếm là 0000, bộ giải mã A chọn R0 -> dữ liệu từ R0 là 110 đưa ra Bus chung -> thanh ghi I -> chọn lệnh **load A** -> mở cổng đầu vào thanh ghi A và chuyển công tắc từ vị trí B->A
- B2: bộ đếm 0001 ->R1 ->100 đưa vào bus chung và đưa trực tiếp vào A (do chuyển mạch vị trí A và đầu vào của A đã mở). Sau khi chuyển dữ liệu thì công tắc chuyển từ A->B

3.6 Mô hình hệ thống máy tính đơn giản

- B3: bộ đếm là 0010 -> chọn R2 -> 111 đưa vào bus chung -> I -> **load B** đồng thời chuyển công tắc từ B->A
 - B4: bộ đếm là 0011 -> chọn R3 -> 010 đưa vào B, chuyển mạch A->B
 - B5: bộ đếm là 0100 -> chọn R4 -> 010 đưa ra bus chung -> ADD và lưu kết quả vào thanh ghi C
- Nhận xét:
 - Sử dụng mạch tuần tự có thể thực hiện các phép tính số học và logic, việc này được thực hiện bằng cách đặt vào mảng thanh ghi R0-R15 các **opcode** và các giá trị phù hợp
 - Lệnh **load A, load B** cần 2 bước: mở cổng vào thanh ghi và nạp giá trị cho thanh ghi (giá trị này không phải là mã lệnh mà là toán hạng, một lệnh tổng quát bao gồm mã lệnh và toán hạng)

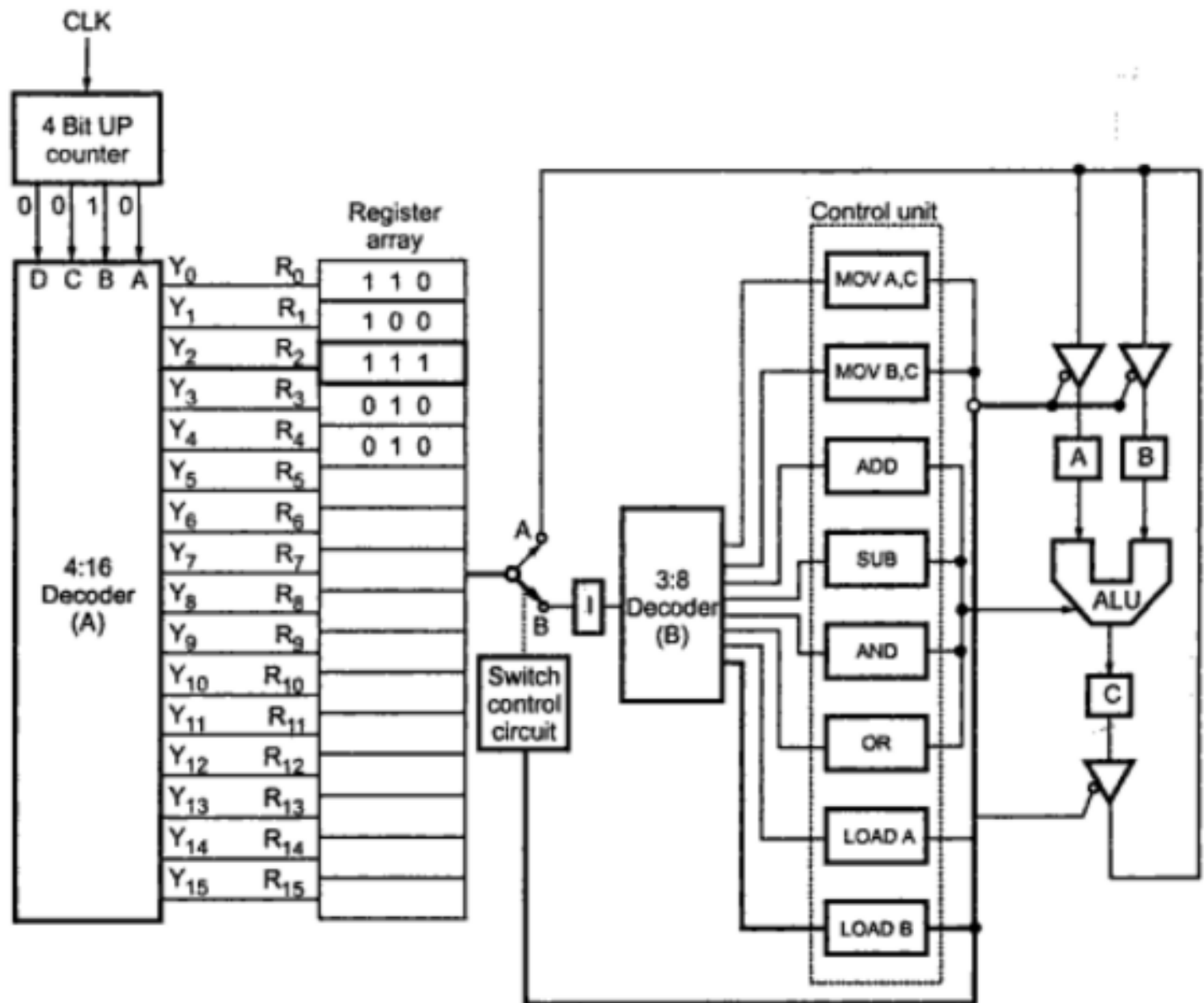
3.6 Mô hình hệ thống máy tính đơn giản

- Bước 1:**



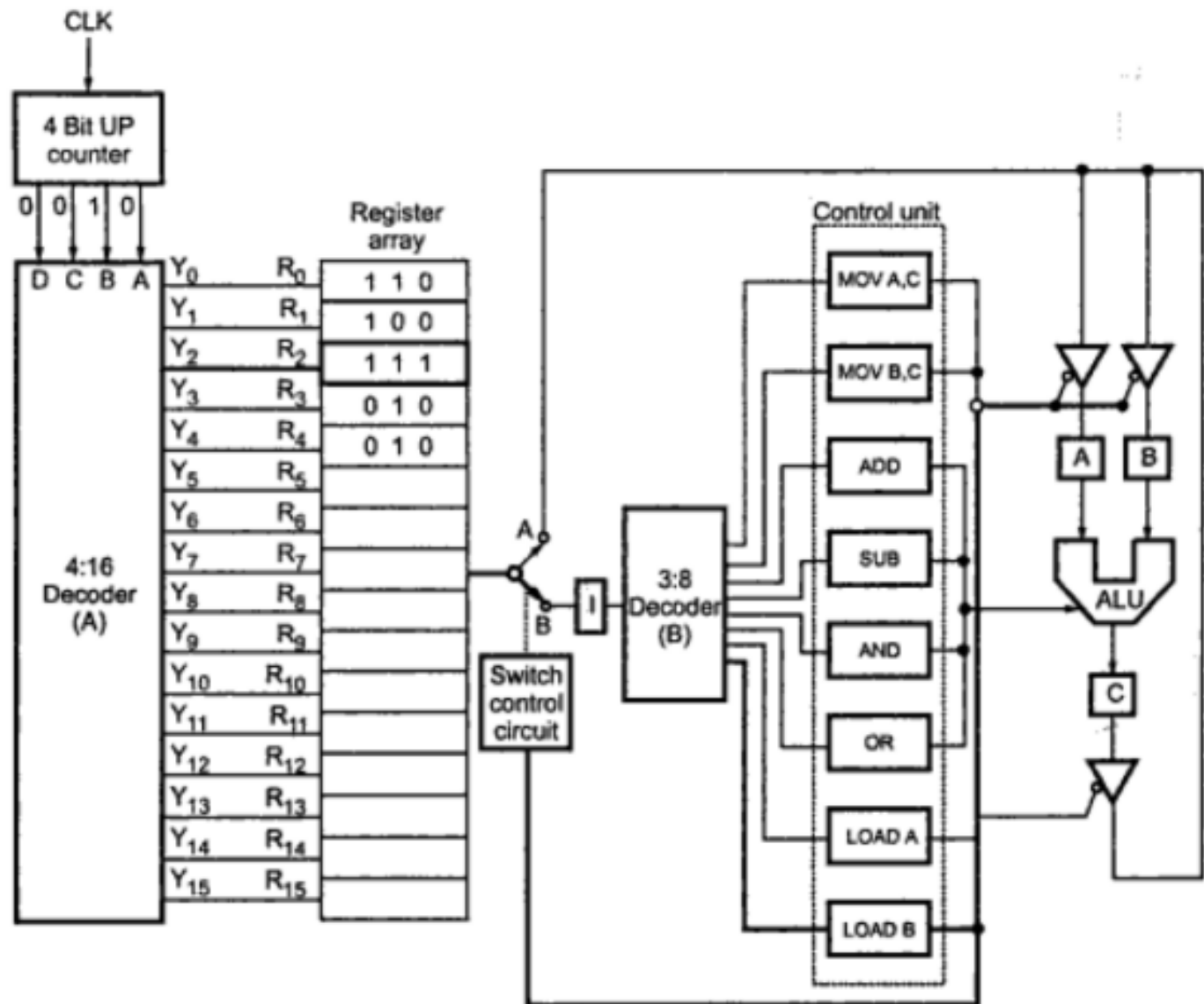
3.6 Mô hình hệ thống máy tính đơn giản

- Bước 2:**



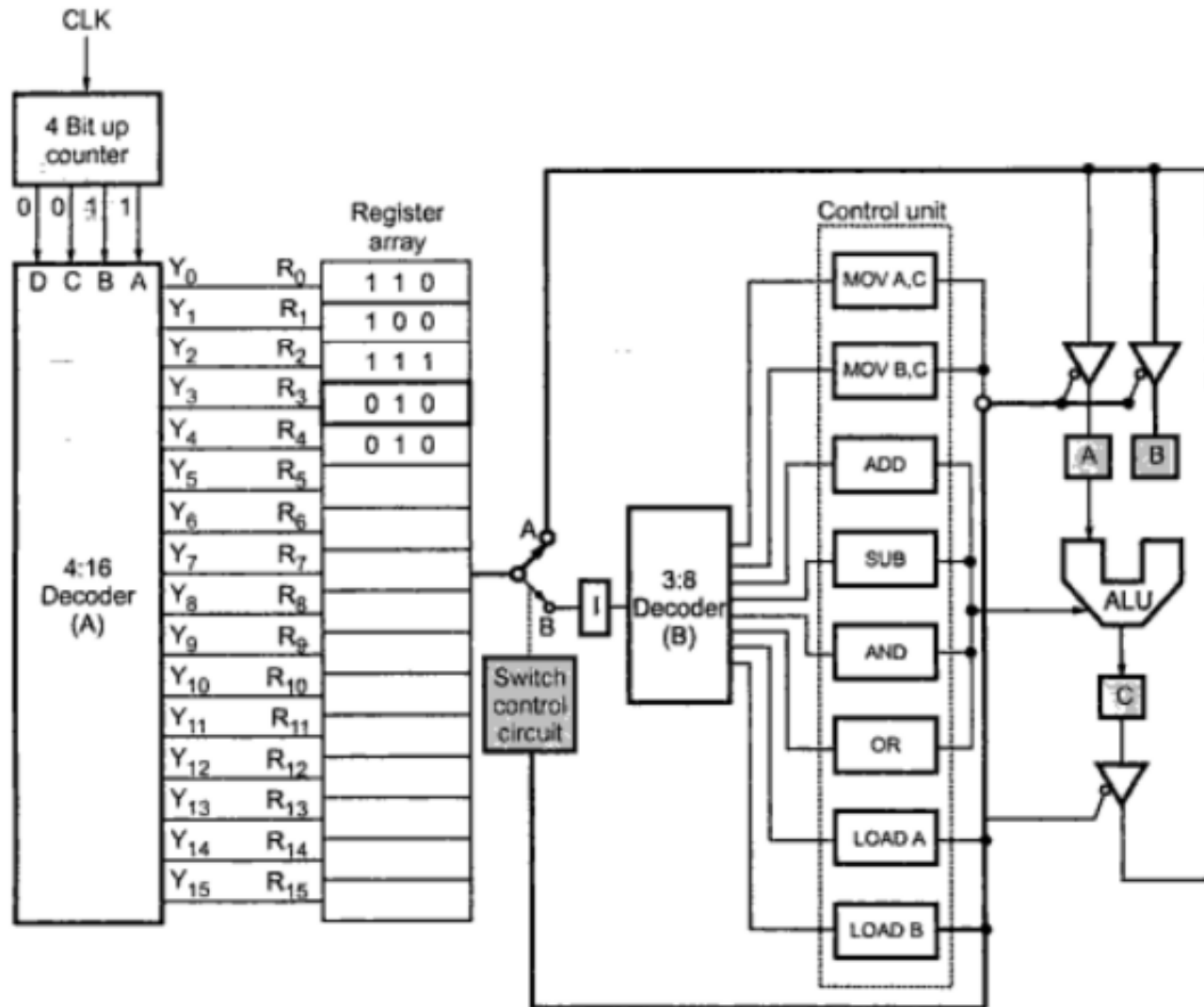
3.6 Mô hình hệ thống máy tính đơn giản

- Bước 3:**



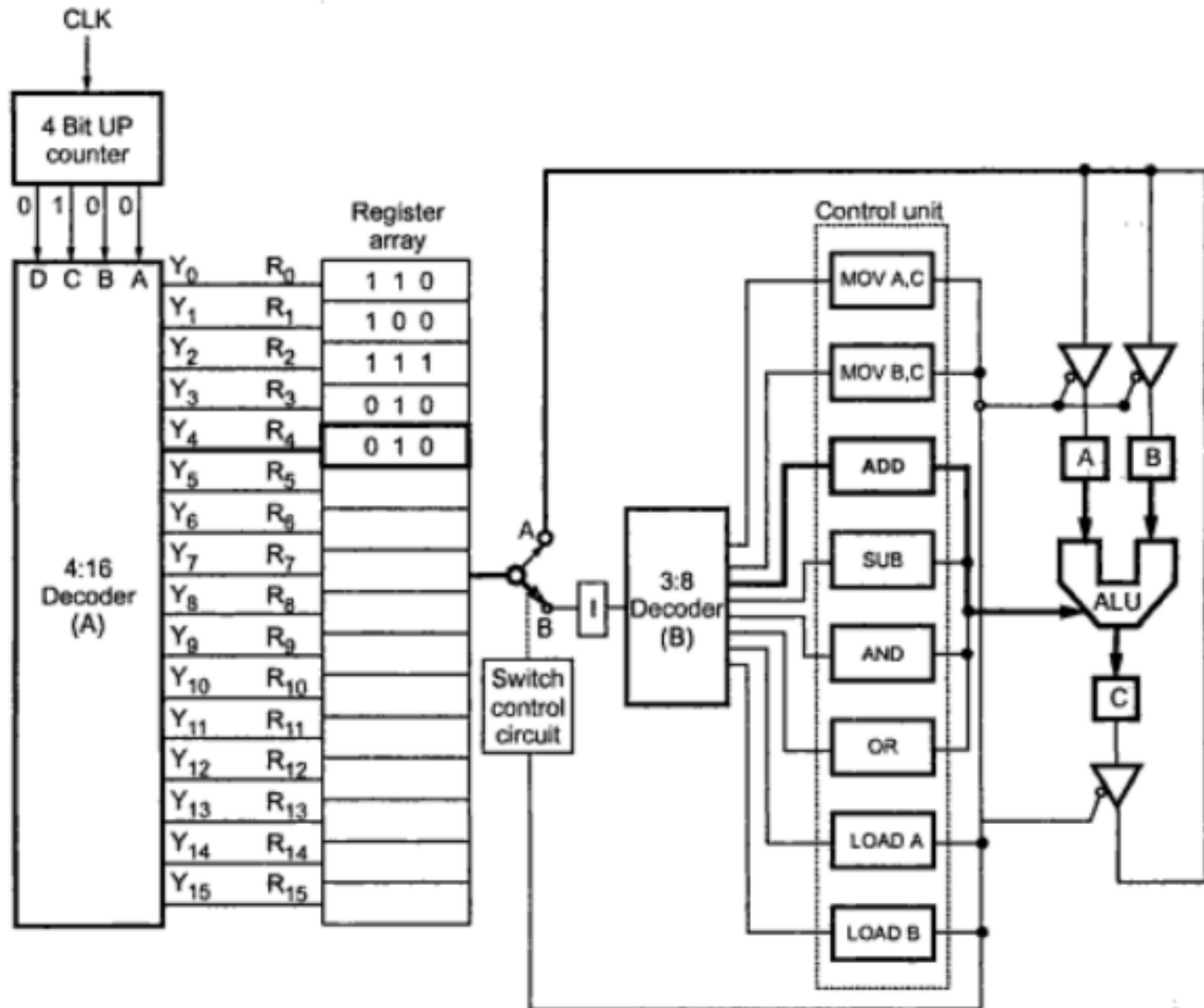
3.6 Mô hình hệ thống máy tính đơn giản

- Bước 4:**



3.6 Mô hình hệ thống máy tính đơn giản

- Bước 5:**



3.6 Mô hình hệ thống máy tính đơn giản

Một số ví dụ:

- VD1: phân tích quá trình thực hiện lệnh trừ (sub): 7 trừ 2

Mảng thanh ghi			Tác động
1	1	0	Nạp thanh ghi A
1	1	1	Với giá trị 7 (111)
1	1	1	Nạp thanh ghi B
0	1	0	Với giá trị 2 (010)
0	1	1	Giá trị trong thanh ghi A trừ giá trị trong thanh ghi B, kết quả đặt vào thanh ghi C

- VD2: Lệnh AND (3 and 5)

Mảng thanh ghi			Tác động
1	1	0	Nạp thanh ghi A
0	1	1	Với giá trị 3 (011)
1	1	1	Nạp thanh ghi B
1	0	1	Với giá trị 5 (101)
1	0	0	Giá trị trong thanh ghi A AND giá trị trong thanh ghi B, kết quả đặt vào thanh ghi C

3.6 Mô hình hệ thống máy tính đơn giản

Một số ví dụ:

- VD3: thực hiện phép toán: $5+1-3$

Mảng thanh ghi			Tác động
1	1	0	Nạp thanh ghi A
1	0	1	Với giá trị 5 (101)
1	1	1	Nạp thanh ghi B
0	0	1	Với giá trị 1 (001)
0	1	0	Giá trị trong thanh ghi A cộng giá trị trong thanh ghi B, kết quả đặt vào thanh ghi C
0	0	0	Copy nội dung thanh ghi C vào thanh ghi A
1	1	1	Nạp vào thanh ghi B
0	1	1	Với giá trị 3 (011)
0	1	1	Giá trị trong thanh ghi A trừ giá trị trong thanh ghi B, kết quả đặt vào thanh ghi C

3.6 Mô hình hệ thống máy tính đơn giản

Một số ví dụ:

- VD4: thực hiện phép toán: 4 AND 3 OR 1

Mảng thanh ghi			Tác động
1	1	0	Nạp thanh ghi A
1	0	0	Với giá trị 4 (100)
1	1	1	Nạp thanh ghi B
0	1	1	Với giá trị 1 (011)
1	0	0	Giá trị trong thanh ghi A AND giá trị trong thanh ghi B, kết quả đặt vào thanh ghi C
0	0	0	Copy nội dung thanh ghi C vào thanh ghi A
1	1	1	Nạp vào thanh ghi B
0	1	1	Với giá trị 3 (011)
1	0	1	Giá trị trong thanh ghi A OR giá trị trong thanh ghi B, kết quả đặt vào thanh ghi C