

# CTT009

## Thao tác dữ liệu

Lê Thị Nhân  
[lt Nhan@fit.hcmus.edu.vn](mailto:lt Nhan@fit.hcmus.edu.vn)



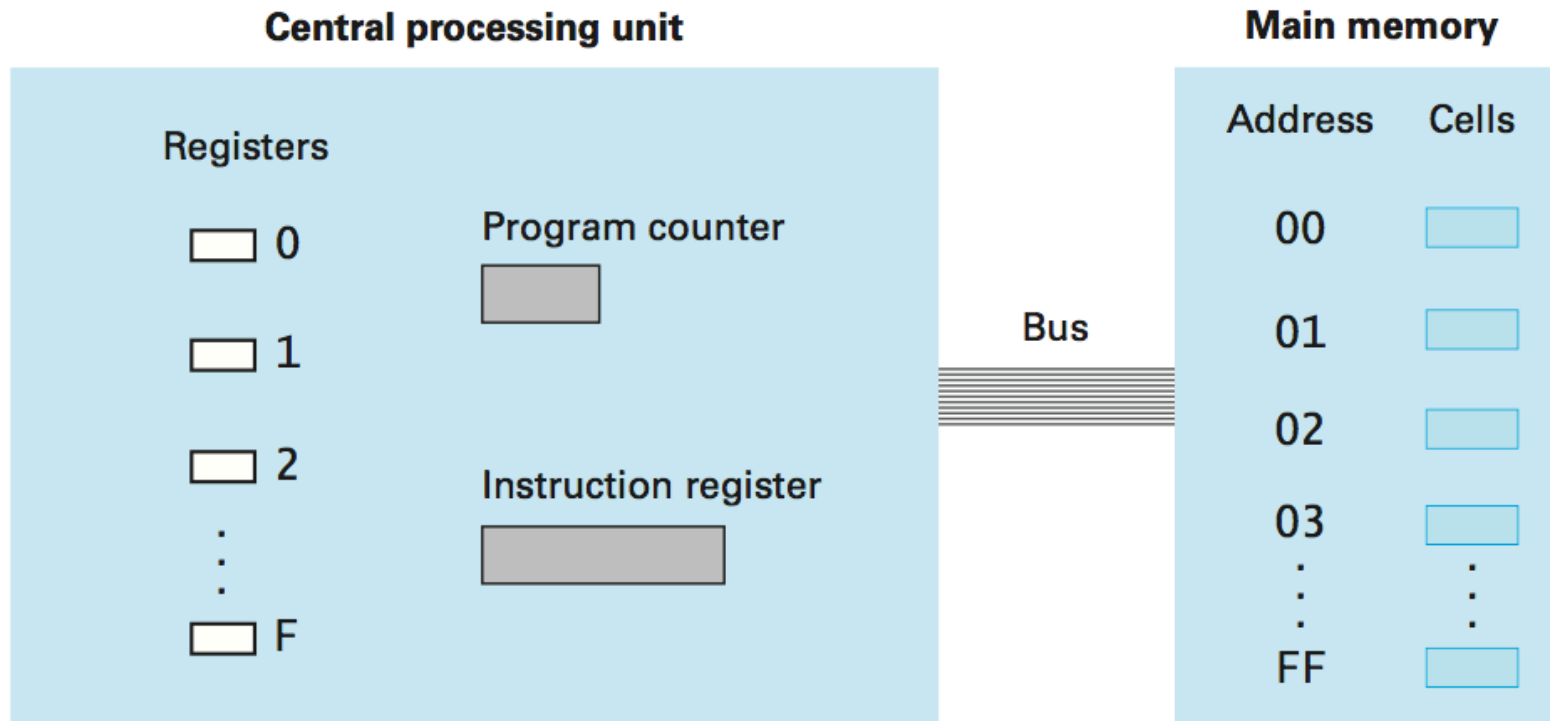
KHOA CÔNG NGHỆ THÔNG TIN  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

# Nội dung

- ☐ Nhắc lại
- ☐ Thực thi chương trình
- ☐ Lệnh arithmetic/logic
- ☐ Giao tiếp với các thiết bị khác



# Kiến trúc máy



Nguồn: Computer Science - An Overview, 12e

# Tập mã lệnh đơn giản

Op-code	Operand	Description
1	RXY	LOAD reg. R from cell XY
2	RXY	LOAD reg. R with XY
3	RXY	STORE reg. R at XY
4	0RS	MOVE R to S
5	RST	ADD S and T into R (2's comp.)
6	RST	ADD S and T into R (floating pt.)
7	RST	OR S and T into R
8	RST	AND S and T into R
9	RST	XOR S and T into R
A	R0X	ROTATE reg. R X times
B	RXY	JUMP to XY if R = reg. 0
C	0	HALT

# Khái niệm

- ☐ Chương trình lưu trữ (stored program) là một dạng đặc biệt dữ liệu
  - ☐ Chuỗi các câu lệnh được mã hóa dưới dạng chuỗi bits
  - ☐ Được lưu trong bộ nhớ
  - ☐ CPU có thể trích ra những lệnh này và đem đi thực thi
  
- ☐ Quá trình thực thi diễn ra như thế nào?



# THỰC THI CHƯƠNG TRÌNH

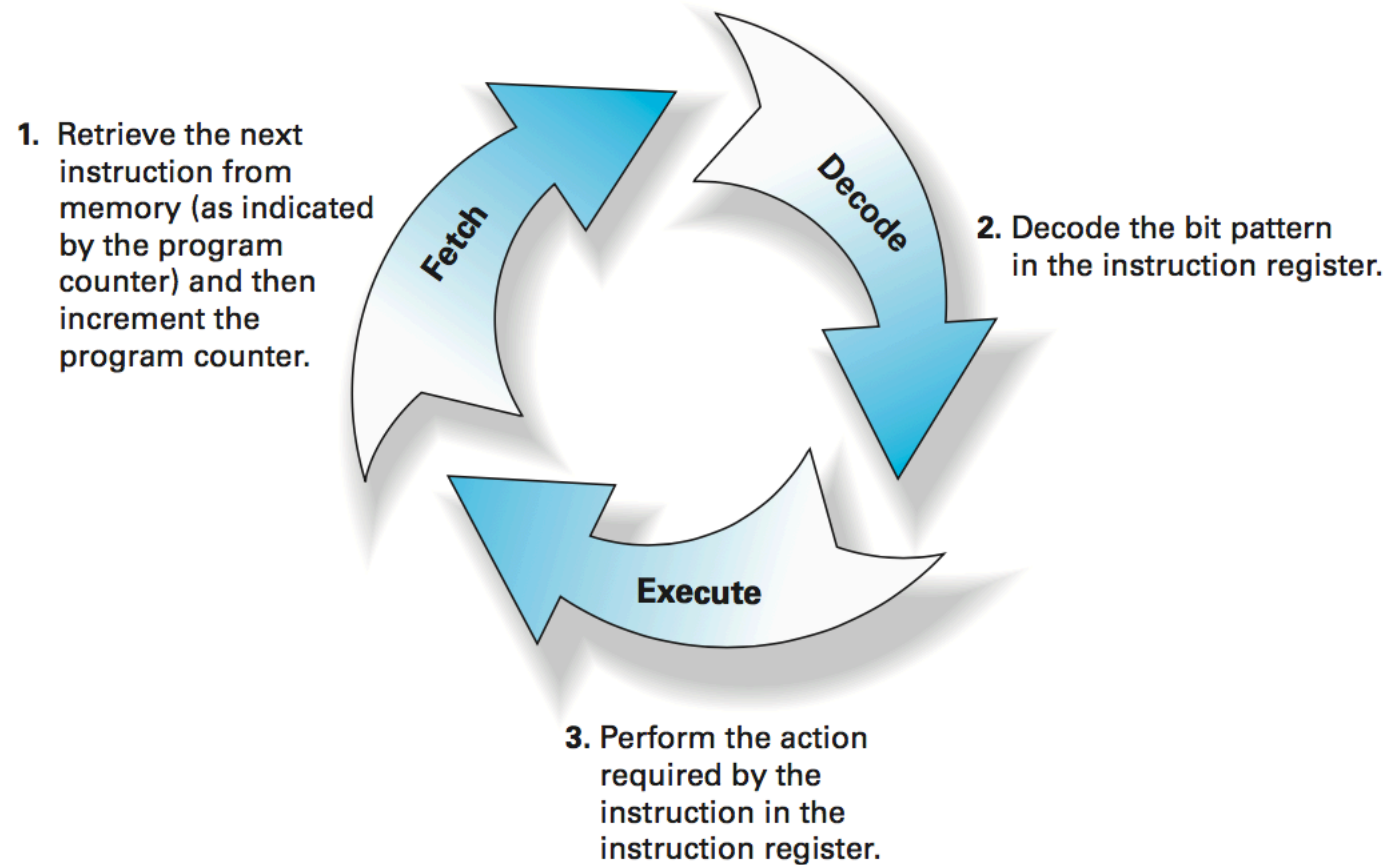


# Thực thi chương trình

- Hai thanh ghi mục đích đặc biệt
  - Chỉ lệnh (instruction register)
    - Chứa câu lệnh được thực thi
  - Bộ đếm chương trình (program counter)
    - Chứa địa chỉ của câu lệnh kế tiếp
    - Cách mà máy tính theo dõi nó đang ở đâu trong 1 chương trình



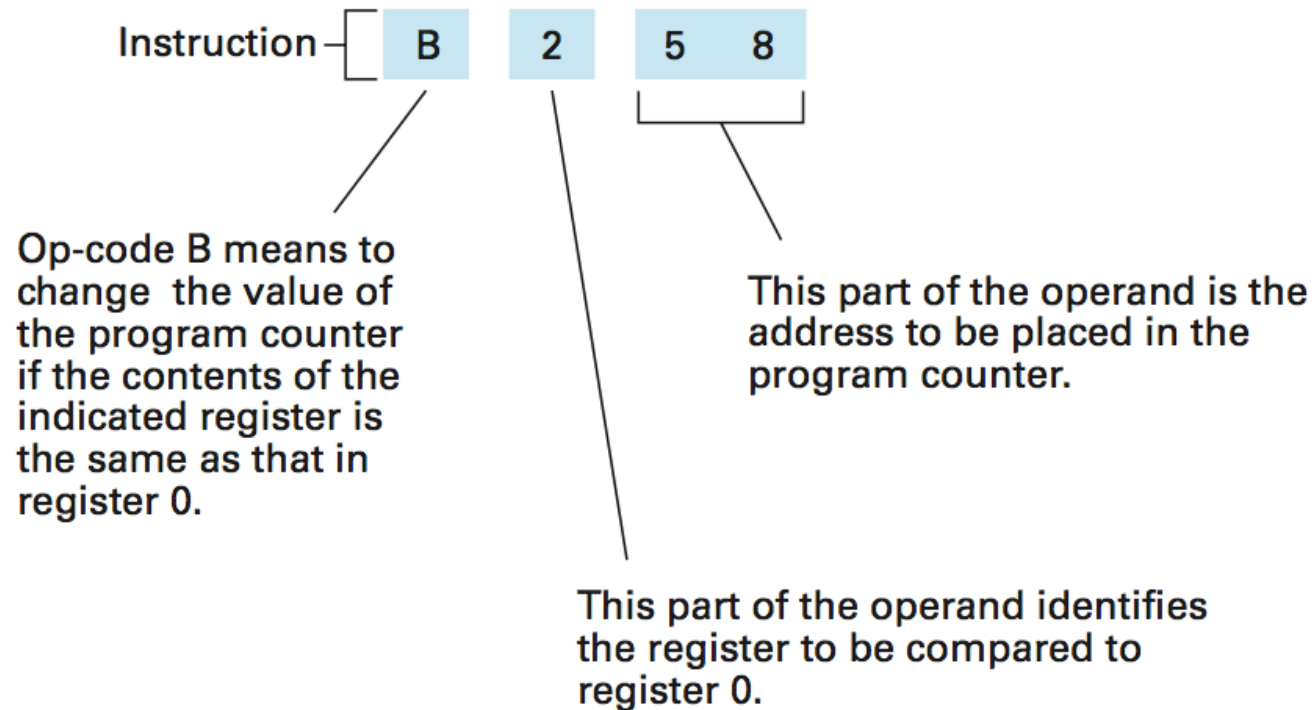
# Chu kỳ máy



*Nguồn: Computer Science - An Overview, 12e*



# Giải mã lệnh B258

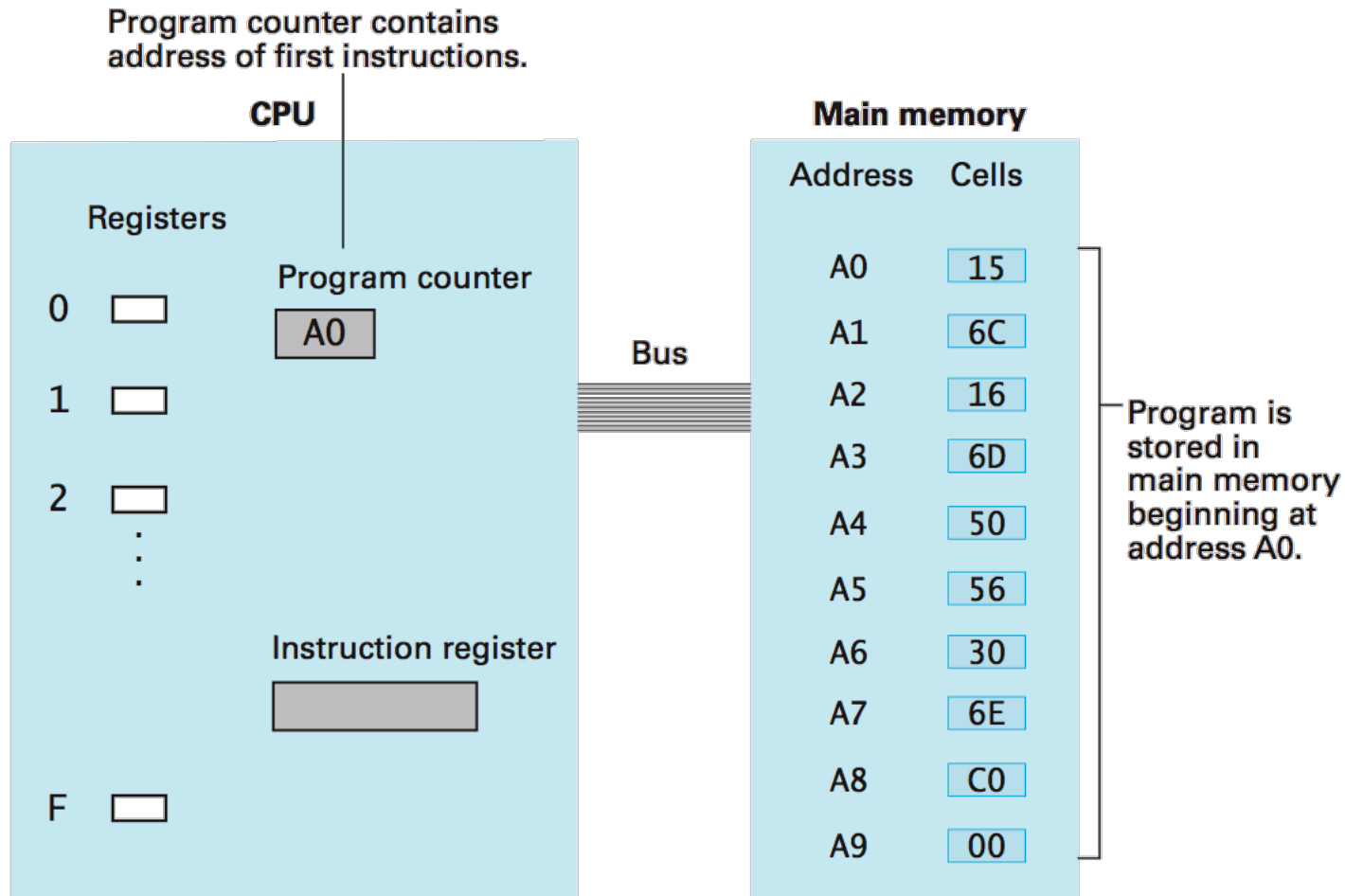


# Nhắc lại ví dụ

Encoded instructions	Translation
<b>156C</b>	Load register 5 with the bit pattern found in the memory cell at address 6C.
<b>166D</b>	Load register 6 with the bit pattern found in the memory cell at address 6D.
<b>5056</b>	Add the contents of register 5 and 6 as though they were two's complement representation and leave the result in register 0.
<b>306E</b>	Store the contents of register 0 in the memory cell at address 6E.
<b>C000</b>	Halt.

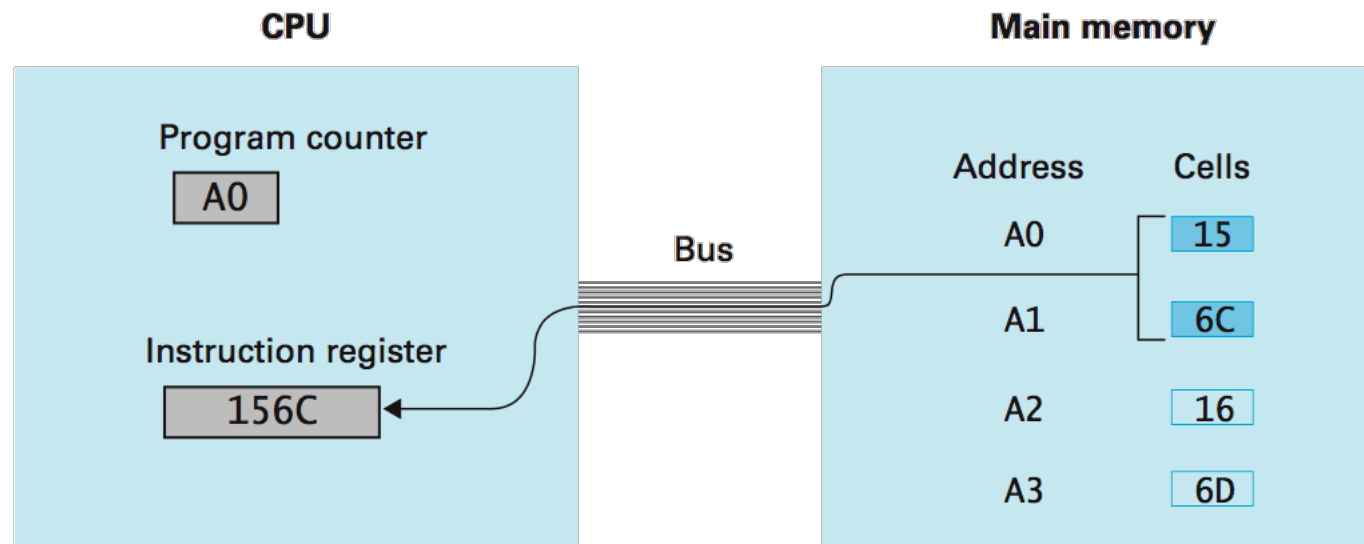
*Nguồn: Computer Science - An Overview, 12e*

# Chương trình sẵn sàng để thực thi



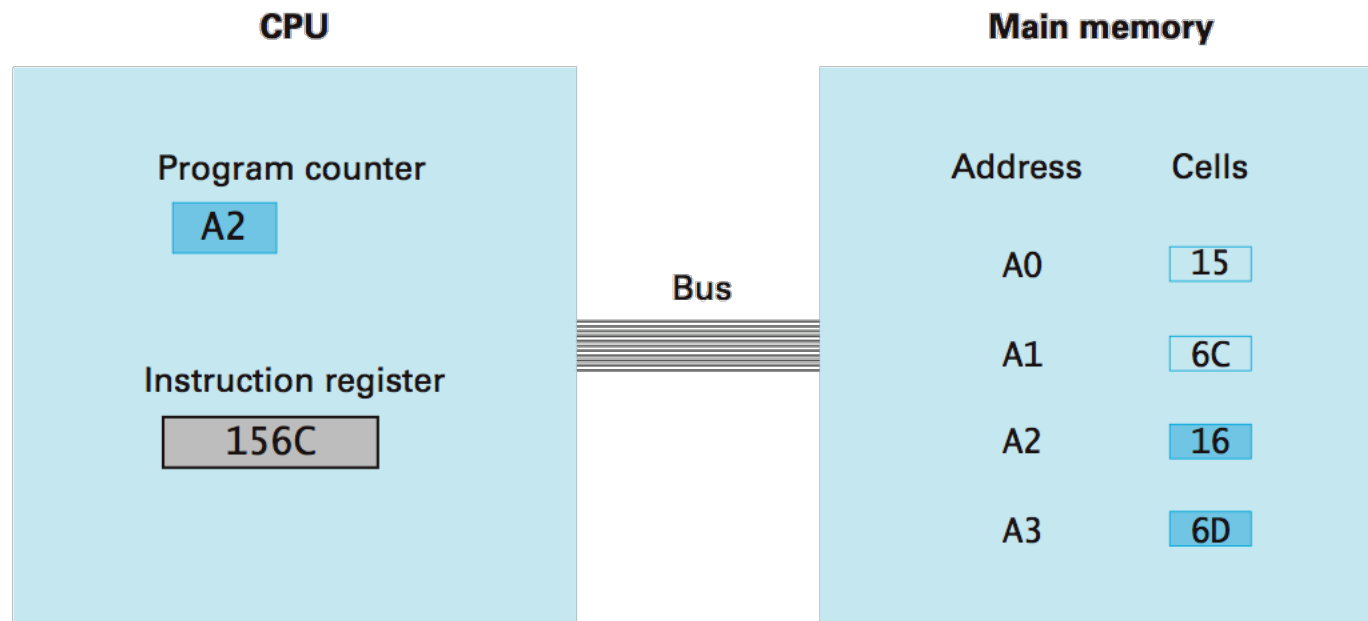
Nguồn: Computer Science - An Overview, 12e

# Fetch



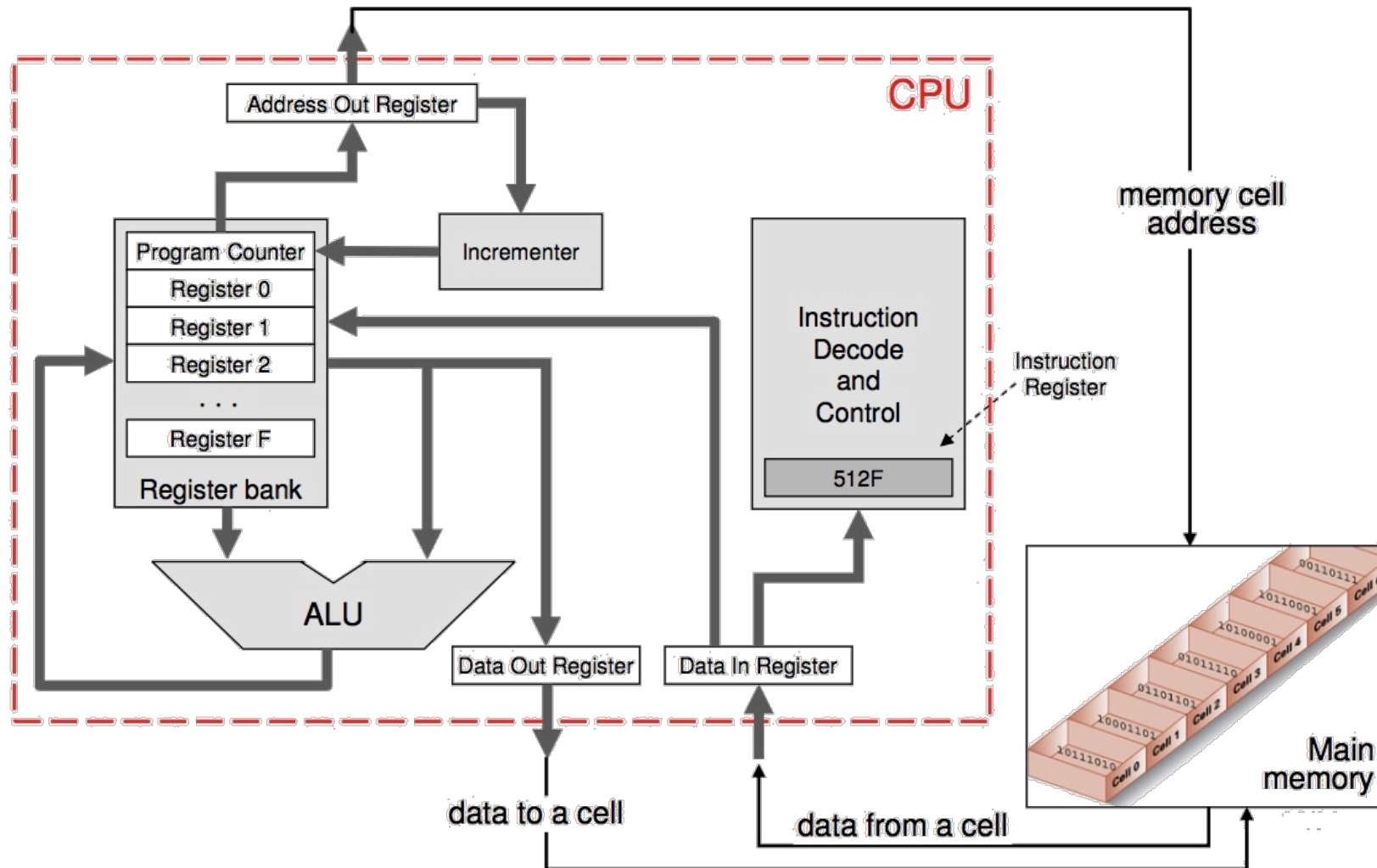
- a. At the beginning of the fetch step the instruction starting at address A0 is retrieved from memory and placed in the instruction register.

# Fetch



**b.** Then the program counter is incremented so that it points to the next instruction.

# Kiến trúc của CPU thực tế



Source: Chun-Jen Tsai, ics12, National Chiao Tung University

# LỆNH ARITHMETIC/LOGIC



# Lệnh arithmetic/logic

- Logic – AND, OR, XOR
  - Masking
- Xoay (rotate) và dịch (shift)
  - Circular shift (rotation)
  - Logical shift
  - Arithmetic shift
- Arithmetic – cộng, trừ, nhân, chia
  - Sự chính xác tùy thuộc vào các giá trị được mã hóa như thế nào (two's complement vs floating-point)



# Masking

$$\begin{array}{r} 00001111 \\ \text{AND } 10101010 \\ \hline 00001010 \end{array}$$

# Masking

$$\begin{array}{r} 11110000 \\ \text{OR } \boxed{1010} \boxed{1010} \\ \hline 11111010 \end{array}$$

# Masking

$$\begin{array}{r} 11111111 \\ \text{XOR } 10101010 \\ \hline 01010101 \end{array}$$

# Circular shift

0 1 1 0 0 1 0 1      The original bit pattern

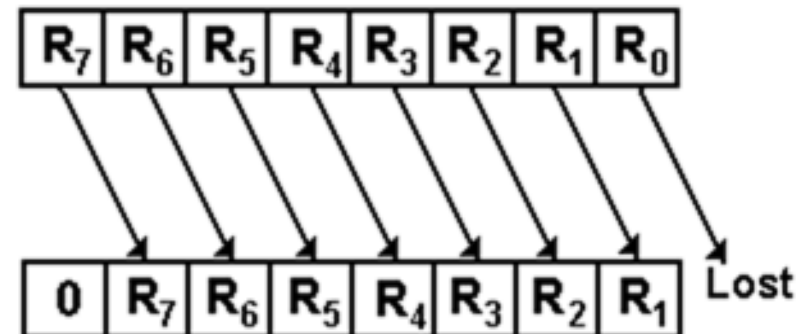
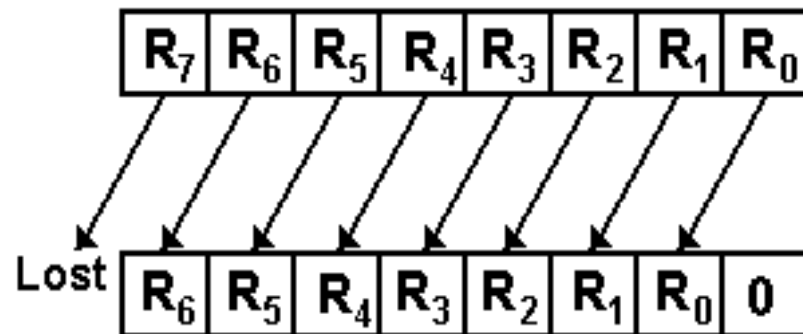
0 1 1 0 0 1 0

The bits move one position to the right. The rightmost bit "falls off" the end and is placed in the hole at the other end.

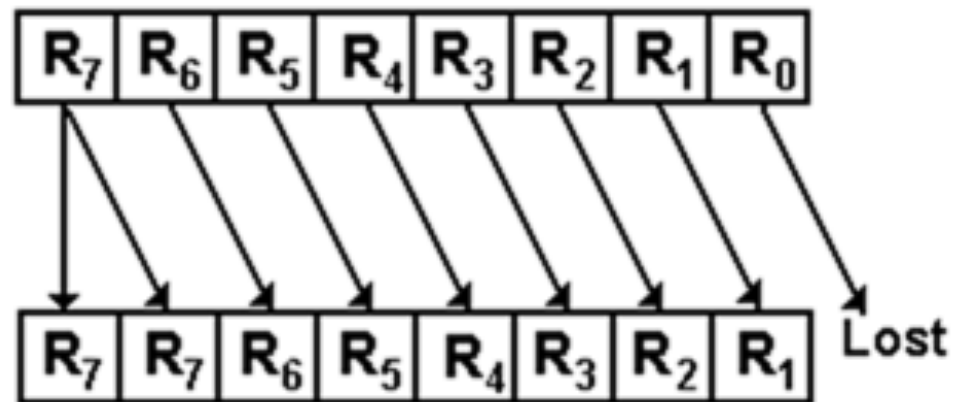
1 0 1 1 0 0 1 0

The final bit pattern

# Logical shift



# Arithmetic shift



# **GIAO TIẾP VỚI CÁC THIẾT BỊ KHÁC**

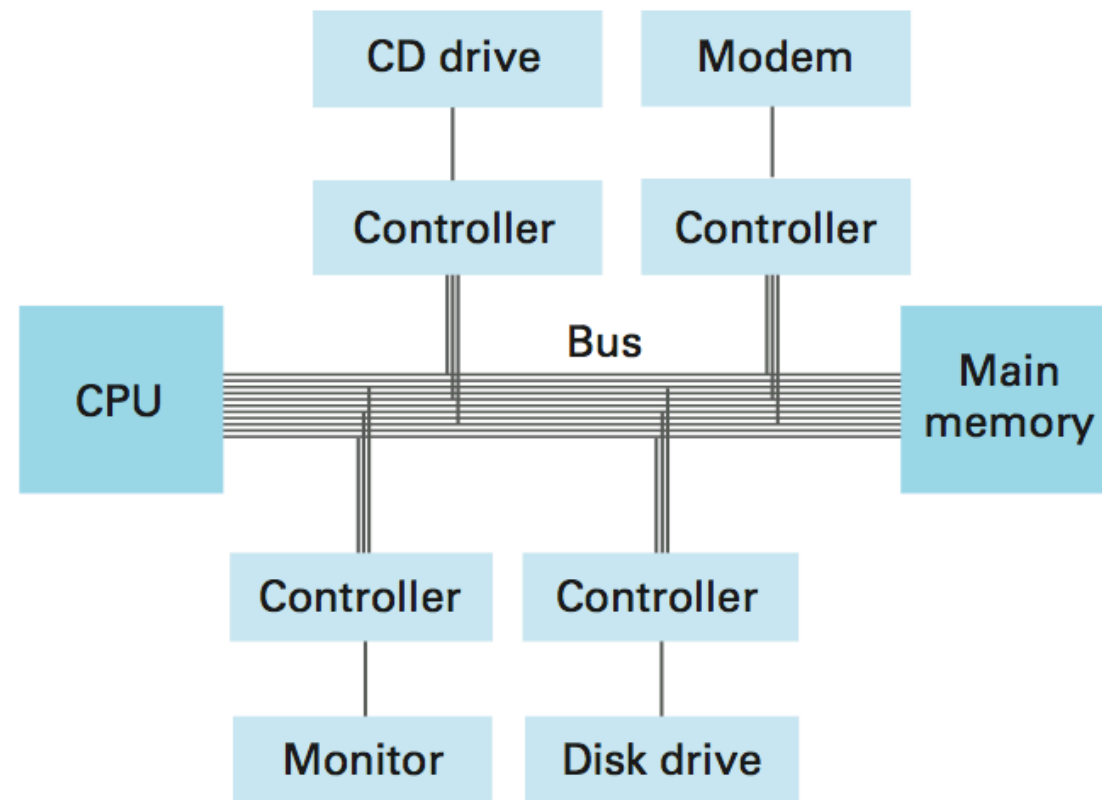


# Các thiết bị khác

- ☐ Bộ điều khiển (controller)
  - ☐ Bộ trung gian xử lý thông tin liên lạc giữa máy tính và thiết bị
    - Bộ điều khiển chuyên biệt cho từng thiết bị
    - Bộ điều khiển mục đích chung (USB, WireFire-Apple hay IEEE 1394)
- ☐ Cổng (port)
  - ☐ Nơi để 1 thiết bị kết nối với máy tính
- ☐ Memory-mapped I/O
  - ☐ CPU giao tiếp với thiết bị ngoại vi như thể với ô nhớ

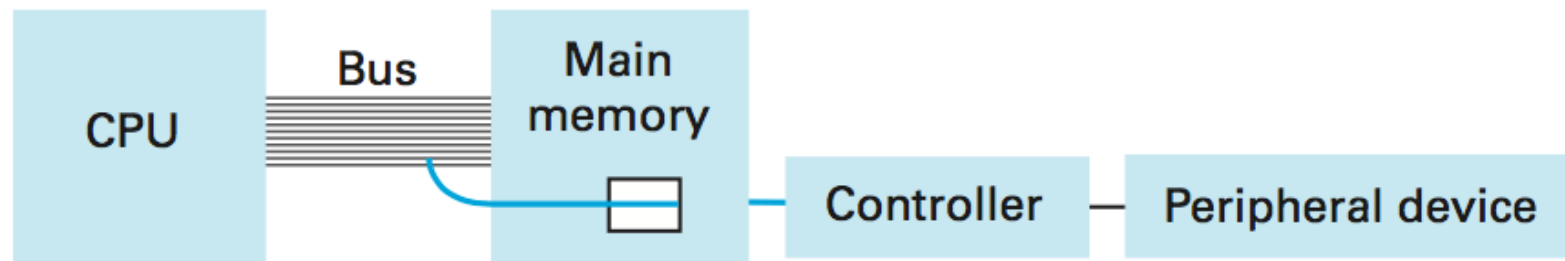


# Bộ điều khiển



Nguồn: Computer Science - An Overview, 12e

# Memory-mapped I/O



*Nguồn: Computer Science - An Overview, 12e*

# Các khái niệm liên quan

- Direct memory access (DMA)
  - Truy xuất bộ nhớ chính bởi 1 bộ điều khiển liên kết với đường bus
- Von Neumann Bottleneck
  - Tốc độ đường bus không đủ ảnh hưởng tới hiệu năng
- Bắt tay (handshaking)
  - Quá trình điều phối việc chuyển đổi dữ liệu giữa các thành phần



# Loại giao tiếp

- ☐ Song song (parallel communication)
  - ☐ Nhiều đường truyền giao tiếp truyền đồng thời các bits
  
- ☐ Tuần tự (serial communication)
  - ☐ Các bits được truyền theo thứ tự qua 1 đường truyền giao tiếp



# Tốc độ truyền dữ liệu

## □ Đơn vị đo

- Bps: Bits per second
- Kbps: Kilo-bps (1,000 bps)
- Mbps: Mega-bps (1,000,000 bps)
- Gbps: Giga-bps (1,000,000,000 bps)

## □ Băng thông (bandwidth)

- Tốc độ tối đa



# KIẾN TRÚC KHÁC



# Kiến trúc khác

## □ Công nghệ gia tăng thông lượng

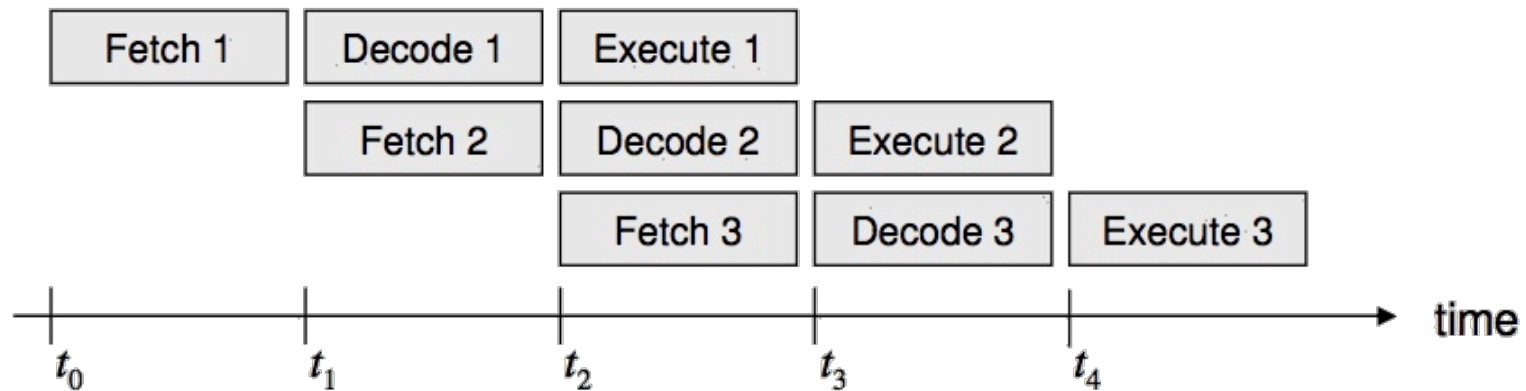
### □ Pipelining

- Chồng các bước của chu kỳ máy

### □ Xử lý song song (parallel processing)

- Sử dụng nhiều bộ vi xử lý đồng thời
- SISD (Single-Instruction Single-Data)
  - Không xử lý song song, 1 bộ vi xử lý thực thi 1 lệnh tại 1 thời điểm
- MIMD (Multiple-Instructions Multiple-Data)
  - Nhiều lệnh khác nhau được thực hiện tại 1 thời điểm thao tác trên nhiều dữ liệu khác nhau
- SIMD (Single-Instruction Multiple-Data)
  - Một lệnh được thực hiện trên nhiều dữ liệu khác nhau

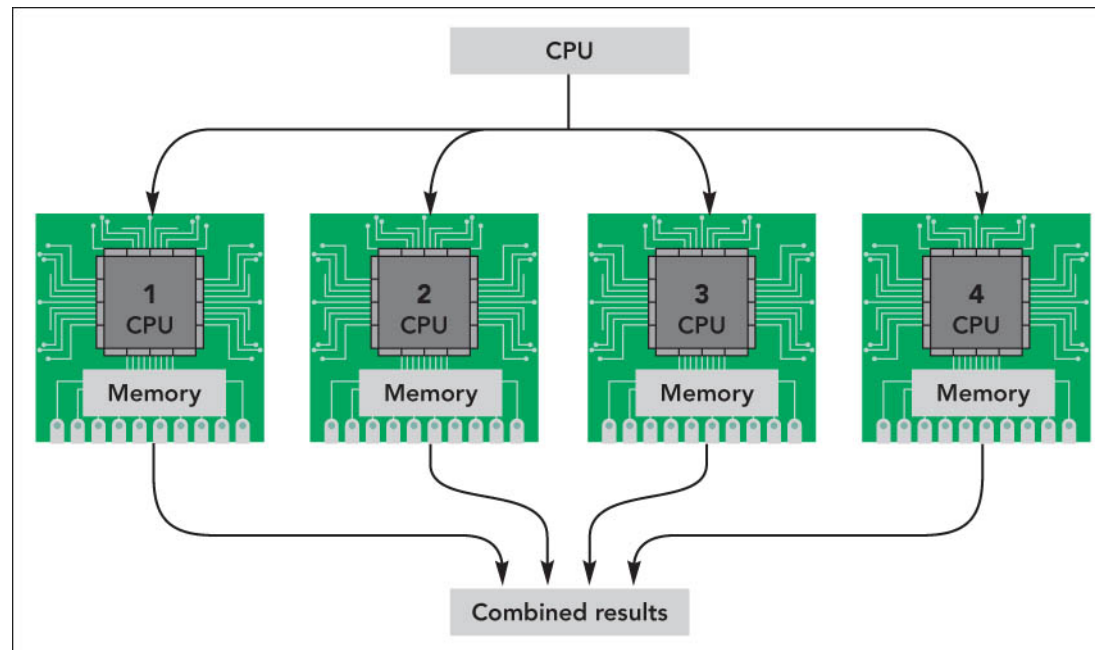
# Pipelining



Source: Chun-Jen Tsai, ics12, National Chiao Tung University



# Parallel processing





# TÓM TẮT





# Bài giảng hôm nay

- ☐ Chu kỳ máy
- ☐ Masking
- ☐ Logical & arithmetic shifts
- ☐ Các thiết bị khác
- ☐ Pipelining



# Bài giảng lần tới

- ☐ Hệ điều hành (chapter 3)
  - ☐ Lịch sử
  - ☐ Kiến trúc



