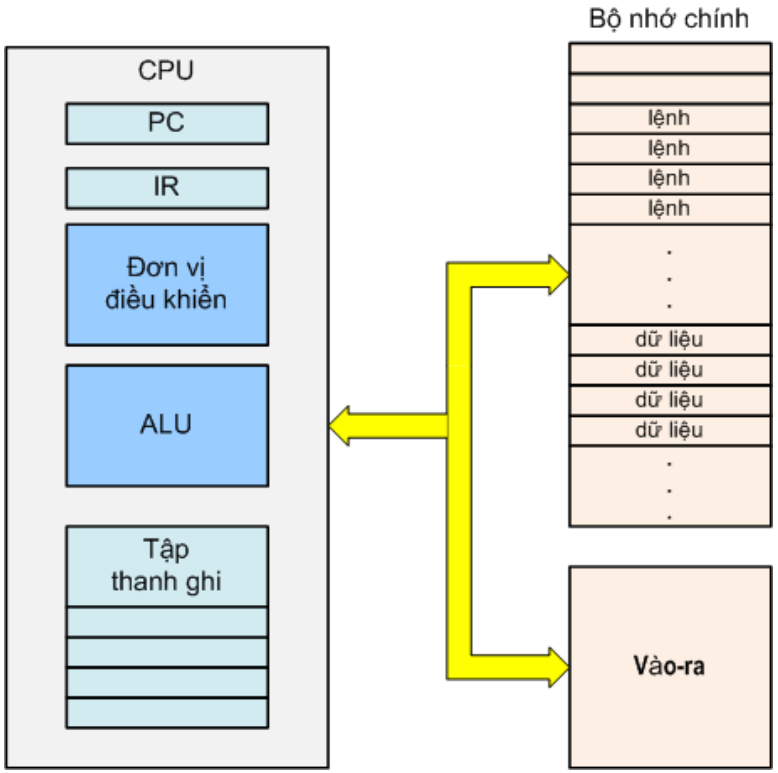
Chương 3: Tập lệnh máy tính

Chương 3: Nội dung chính

- Giới thiệu về tập lệnh
- Khuôn dạng và các thành phần của lệnh
- Các dạng toán hạng lệnh
- Các chế độ địa chỉ
- Một số dạng lệnh thông dụng
- Cơ chế ống lệnh

Giới thiệu chung

Mô hình lập trình của máy tính



PC: Program Counter

IR: Instruction Register

Giới thiệu chung về tập lệnh

- Mỗi bộ xử lý có một tập lệnh xác định
- Tập lệnh thường có hàng chục đến hàng trăm lệnh
- Mỗi lệnh là một chuỗi số nhị phân mà bộ xử lý hiểu được để thực hiện một thao tác xác định.
- Các lệnh được mô tả bằng các ký hiệu gợi nhớ dạng text: chính là các lệnh của hợp ngữ (assembly language)

Giới thiệu chung

- Lệnh máy tính là một từ nhị phân (binary word) mà thực hiện một nhiệm vụ cụ thể:
  - Lệnh được lưu trong bộ nhớ
  - Lệnh được đọc từ bộ nhớ vào CPU để giải mã và thực hiện
  - Mỗi lệnh có chức năng riêng của nó
- Tập lệnh gồm nhiều lệnh, có thể được chia thành các nhóm theo chức năng:
  - Chuyển dữ liệu (data movement)
  - Tính toán (computational)
  - Điều kiện và rẽ nhánh (conditioning & branching)
  - Các lệnh khác …

Giới thiệu chung

- Quá trình thực hiện/ chạy lệnh được chia thành các pha hay giai đoạn (stage). Mỗi lệnh có thể được thực hiện theo 4 giai đoạn:
  - Đọc lệnh IF(Instruction Fetch): lệnh được đọc từ bộ nhớ vào CPU
  - Giải mã lệnh ID(Instruction Decode): CPU giải mã lệnh
  - Chạy lệnh IE(Instruction Execution): CPU thực hiện lệnh
  - Ghi WB(Write Back): kết quả lệnh (nếu có) được ghi vào thanh ghi hoặc bộ nhớ

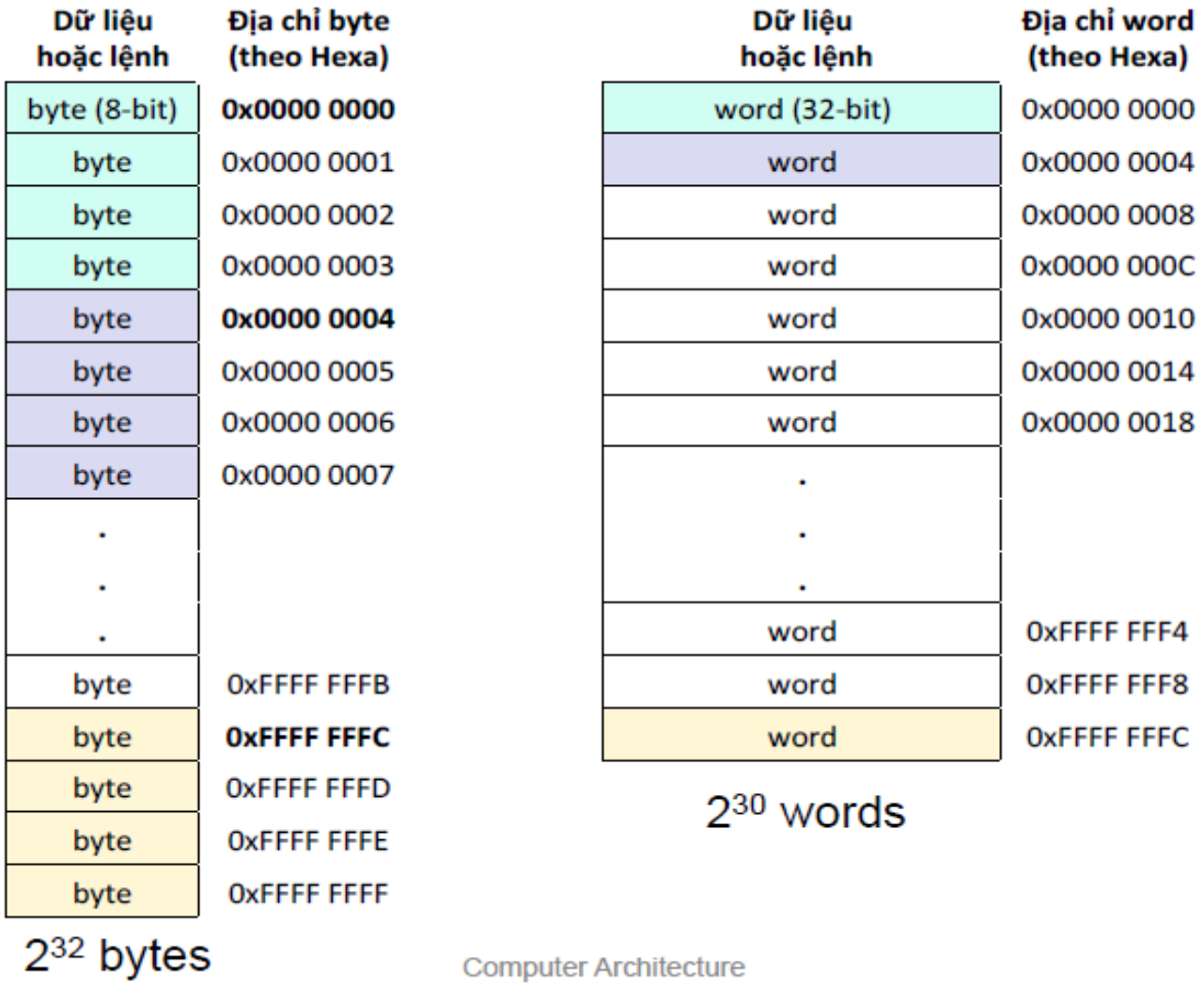
Chu kỳ thực hiện lệnh

- Chu kỳ thực hiện lệnh (instruction execution cycle) là khoảng thời gian mà CPU thực hiện xong một lệnh
  - Một chu kỳ thực hiện lệnh gồm một số giai đoạn thực hiện lệnh
  - Một giai đoạn thực hiện lệnh có thể gồm một số chu kỳ máy
  - Một chu kỳ máy có thể gồm một số chu kỳ đồng hồ

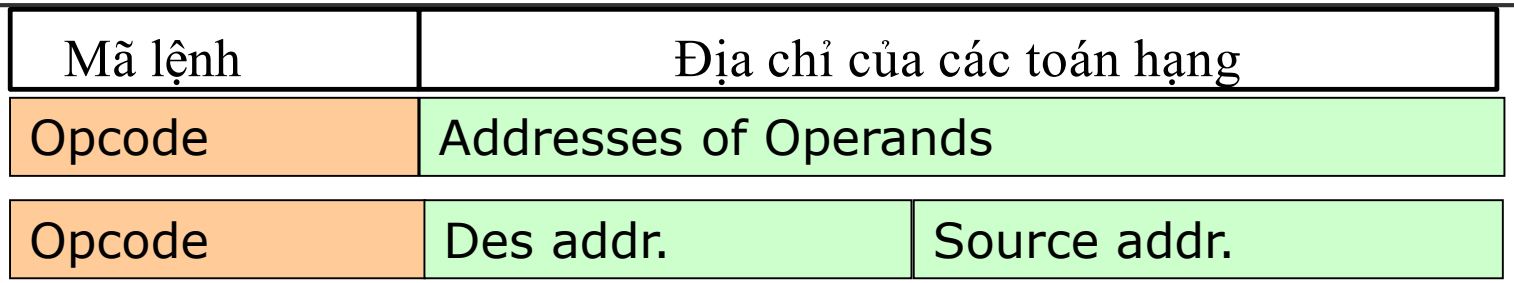
Chu kỳ thực hiện lệnh

- Một chu kỳ thực hiện lệnh có thể gồm các thành phần sau:
  - Chu kỳ đọc lệnh
  - Chu kỳ đọc bộ nhớ (memory read)
  - Chu kỳ ghi bộ nhớ (memory write)
  - Chu kỳ đọc thiết bị ngoại vi (I/O read)
  - Chu kỳ ghi thiết bị ngoại vi (I/O write)
  - Chu kỳ bus rỗi (bus idle)

Địa chỉ byte nhớ và word nhớ



Các thành phần của lệnh máy



- Mã lệnh (operation code □ opcode): mã hóa cho thao tác mà bộ xử lý phải thực hiện
- Địa chỉ toán hạng: chỉ ra nơi chứa các toán hạng mà thao tác sẽ tác động:
  - Toán hạng nguồn (source operand): dữ liệu vào của thao tác
  - Toán hạng đích (destination operand): dữ liệu ra của thao tác

Chế độ địa chỉ toán hạng

- Toán hạng của lệnh có thể là:
  - Một giá trị cụ thể nằm ngay trong lệnh
  - Nội dung của thanh ghi
  - Nội dung của ngăn nhớ hoặc cổng vào-ra
- Phương pháp định địa chỉ (addressing modes) là cách thức địa chỉ hóa trong trường địa chỉ của lệnh để xác định nơi chứa toán hạng

Các chế độ địa chỉ

- Chế độ địa chỉ là cách thức CPU tổ chức các toán hạng
  - Chế độ địa chỉ cho phép CPU kiểm tra dạng và tìm các toán hạng của lệnh
- Một số chế độ địa chỉ tiêu biểu:
  - Chế độ địa chỉ tức thì (Immediate)
  - Chế độ địa chỉ trực tiếp (Direct)
  - Chế độ địa chỉ gián tiếp qua thanh ghi (Register Indirect)
  - Chế độ địa chỉ gián tiếp qua bộ nhớ (Memory Indirect)
  - Chế độ địa chỉ chỉ số (Indexed)
  - Chế độ địa chỉ tương đối (Relative)

Chế độ địa chỉ tức thì

- Giá trị của toán hạng nguồn có sẵn trong lệnh (hằng số)
- Toán hạng đích có thể là thanh ghi hoặc một vị trí bộ nhớ
- Ví dụ:

LOAD R_1, #1000; 1000 → R_1

giá trị 1000 được tải vào thanh ghi R1

LOAD B, #500; 500 → M[B]

Giá trị 500 được tải vào vị trí B trong bộ nhớ

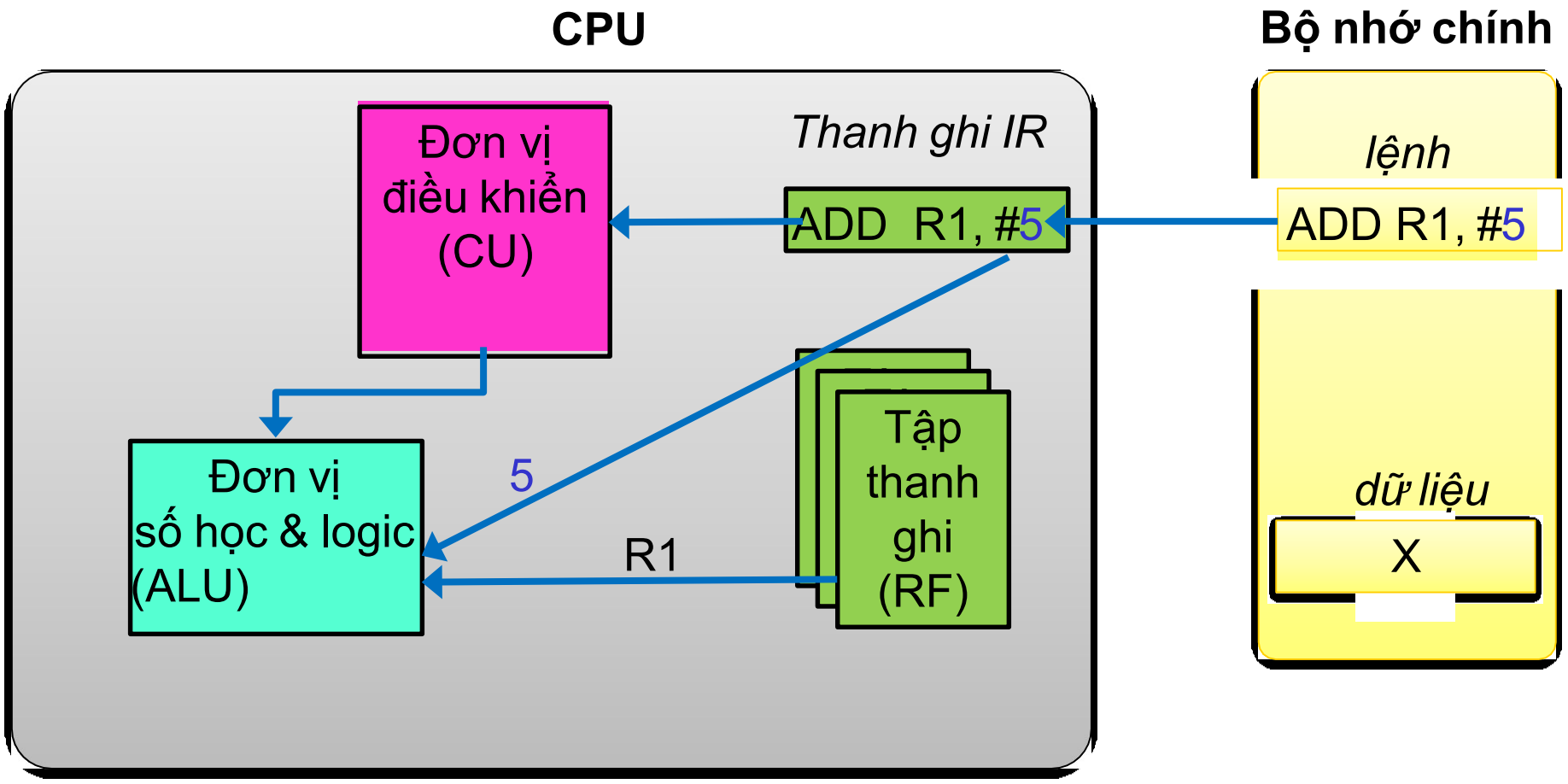
Chế độ địa chỉ tức thì

- Toán hạng là hằng số nằm ngay trong lệnh
- Chỉ có thể là toán hạng nguồn
- Ví dụ:

```
ADD   R1, #5      // R1 =  R1+5
```

- Không tham chiếu bộ nhớ
- Truy nhập toán hạng rất nhanh
- Dải giá trị của toán hạng bị hạn chế

Chế độ địa chỉ tức thì



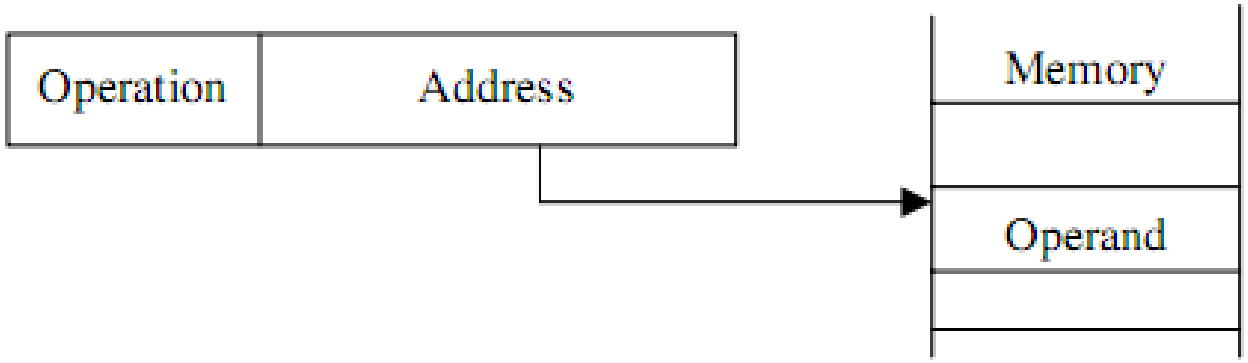
Chế độ địa chỉ trực tiếp/ tuyệt đối

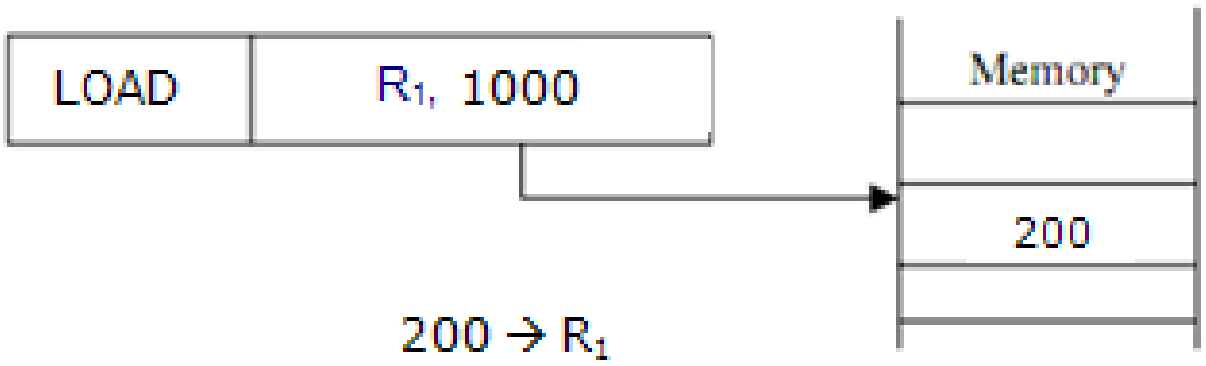
- Một toán hạng là địa chỉ của một vị trí trong bộ nhớ chứa dữ liệu
- Toán hạng kia là thanh ghi hoặc 1 địa chỉ ô nhớ
- Ví dụ:

LOAD R_1, 1000; M[1000] → R_1

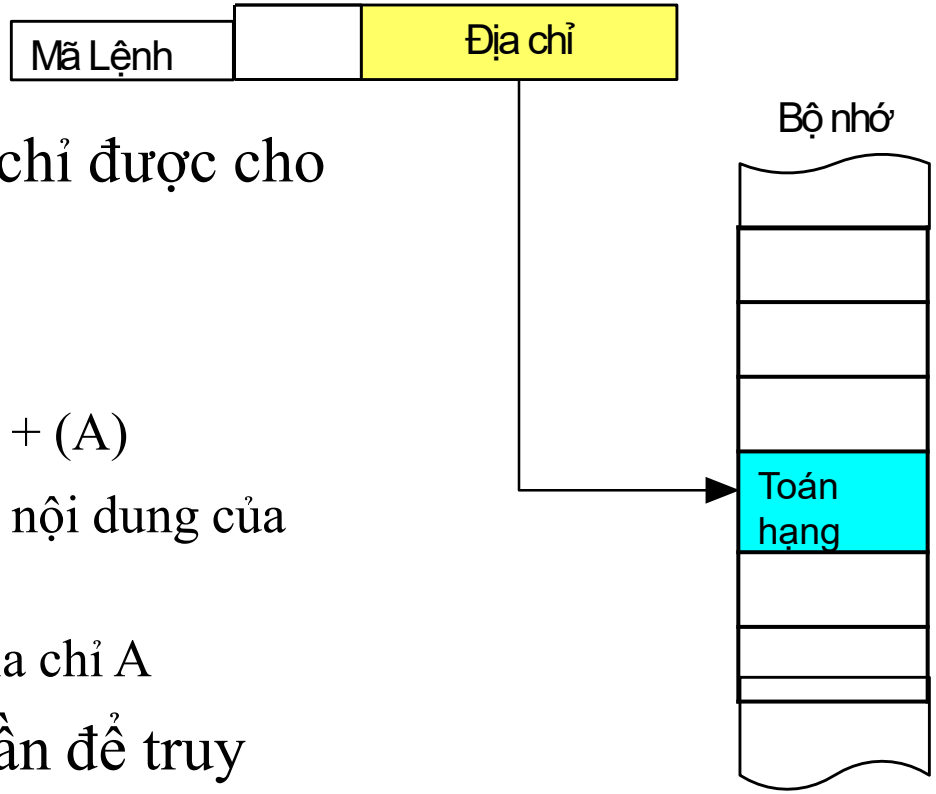
giá trị lưu trong vị trí 1000 ở bộ nhớ được tải vào thanh ghiR1

Chế độ địa chỉ trực tiếp/ tuyệt đối





Chế độ địa chỉ trực tiếp

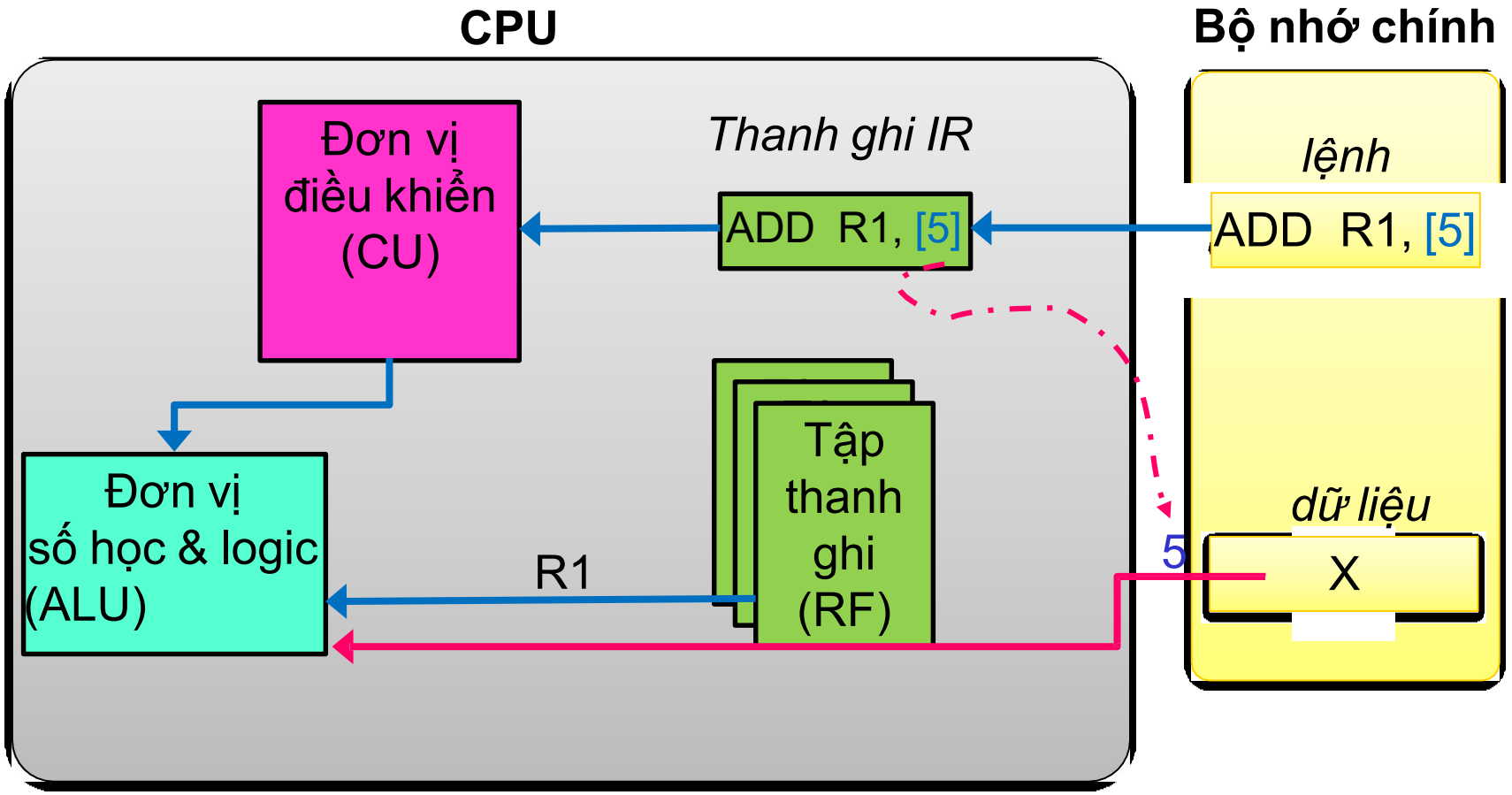


- Toán hạng là ngăn nhớ có địa chỉ được cho trực tiếp trong lệnh
- Ví dụ:

  ADD R1, A #R1 = R1 + (A)

  - Cộng nội dung thanh ghi R1 với nội dung của ngăn nhớ có địa chỉ là A
  - Tìm toán hạng trong bộ nhớ ở địa chỉ A
- CPU tham chiếu bộ nhớ một lần để truy nhập dữ liệu

Chế độ địa chỉ trực tiếp



Chế độ địa chỉ gián tiếp

- Một thanh ghi hoặc một vị trí trong bộ nhớ được sử dụng để lưu địa chỉ của toán hạng
  - Gián tiếp thanh ghi:

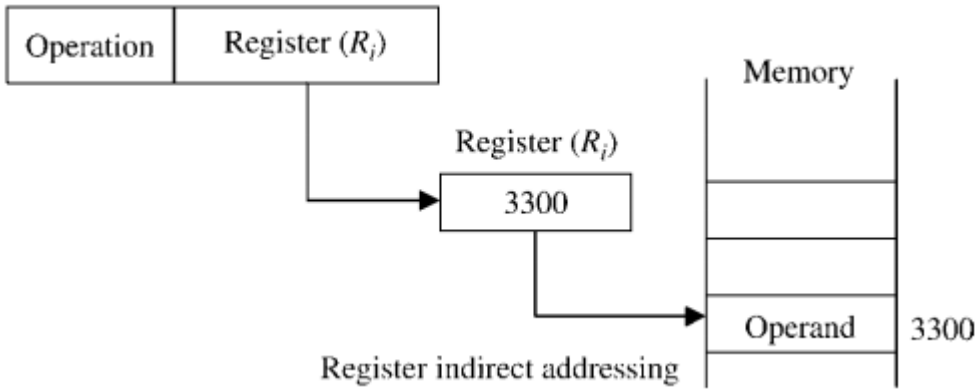
    LOAD R_j ,(R_i); M[R_i] → R_j

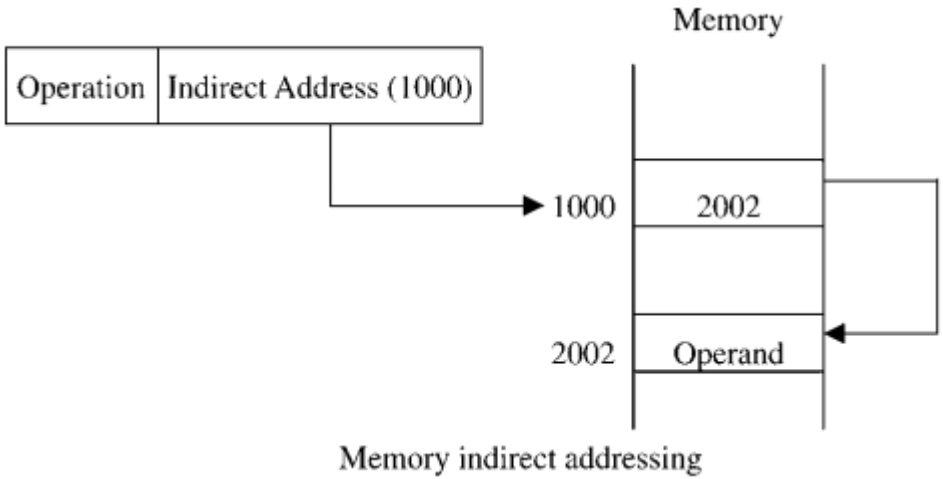
    Tải giá trị tại vị trí bộ nhớ có địa chỉ được lưu trong Ri vào thanh ghi Rj
  - Gián tiếp bộ nhớ:

    LOAD R_i , (1000); M[M[1000]] → R_i

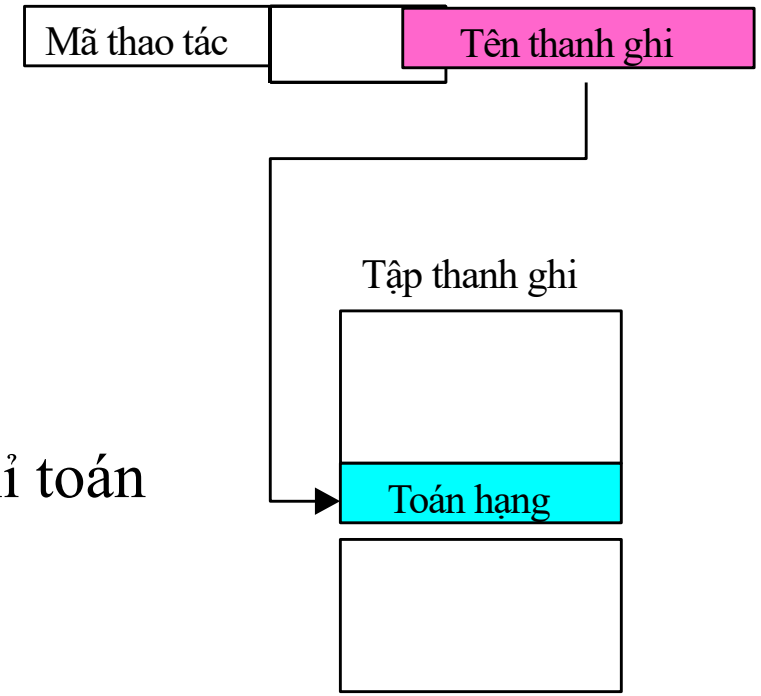
    Giá trị của vị trí bộ nhớ có địa chỉ được lưu tại vị trí 1000 vào Ri

Chế độ địa chỉ gián tiếp





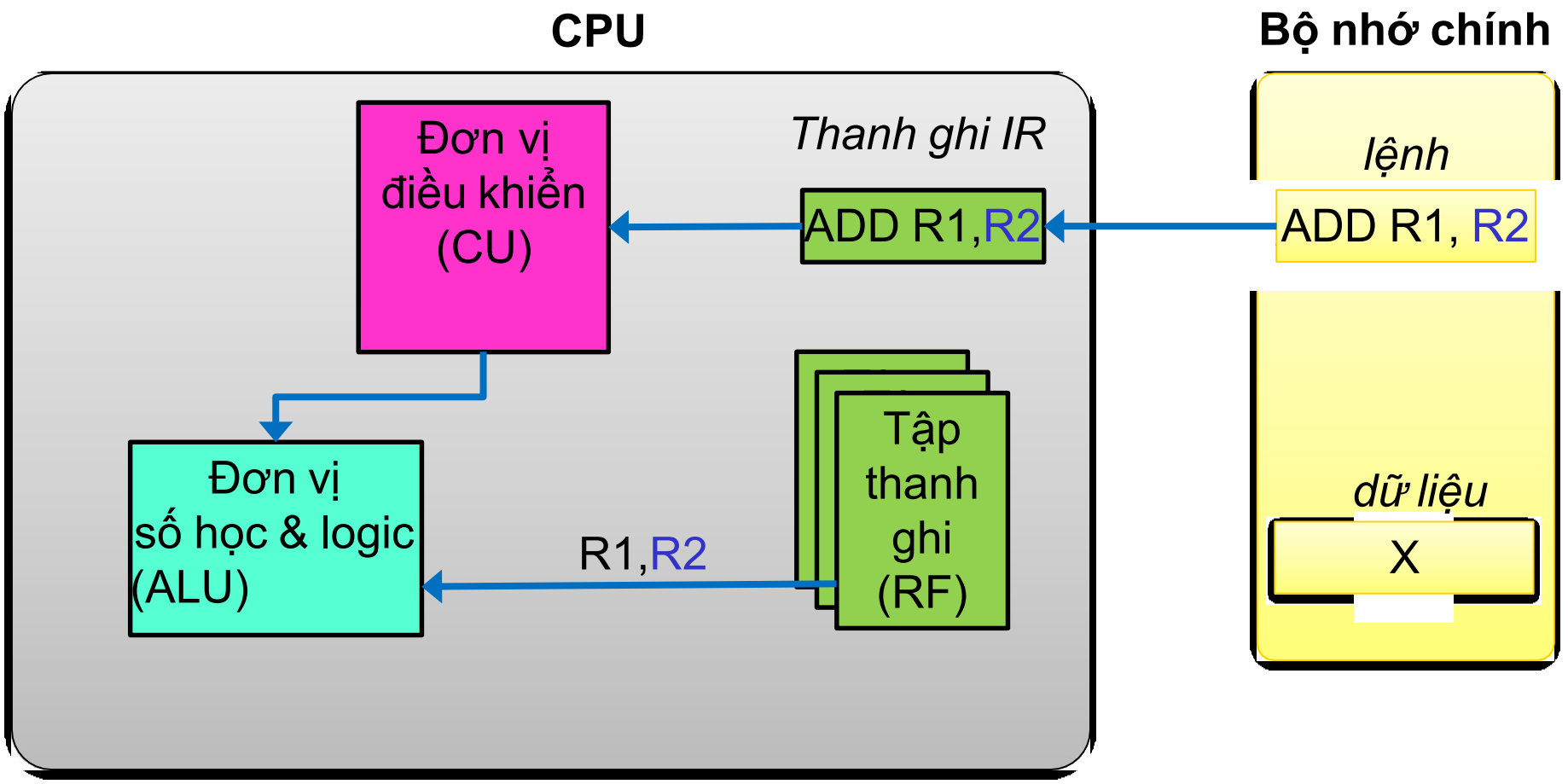
Định địa chỉ thanh ghi



- Toán hạng nằm trong thanh ghi có tên được chỉ ra trong lệnh
- Ví dụ:

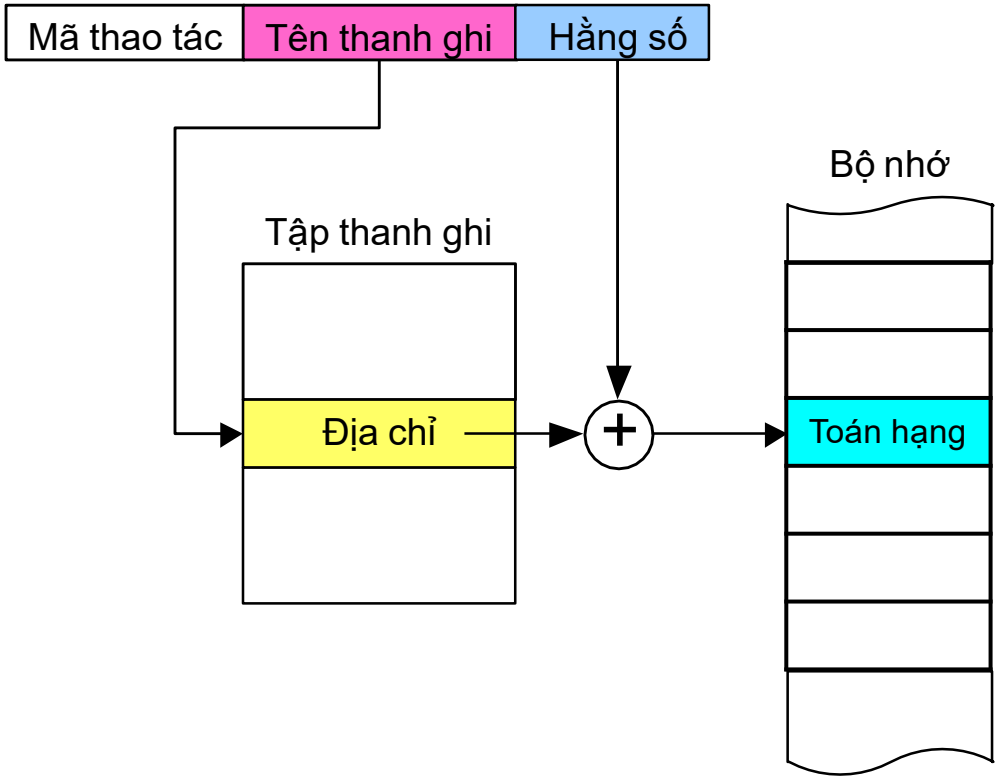
  ADD R1, R2 # R1= R1+R2
- Số lượng thanh ghi ít : Trường địa chỉ toán hạng chỉ cần ít bit
- Không tham chiếu bộ nhớ
- Truy nhập toán hạng nhanh
- Tăng số lượng thanh ghi : hiệu quả hơn

Chế độ địa chỉ thanh ghi

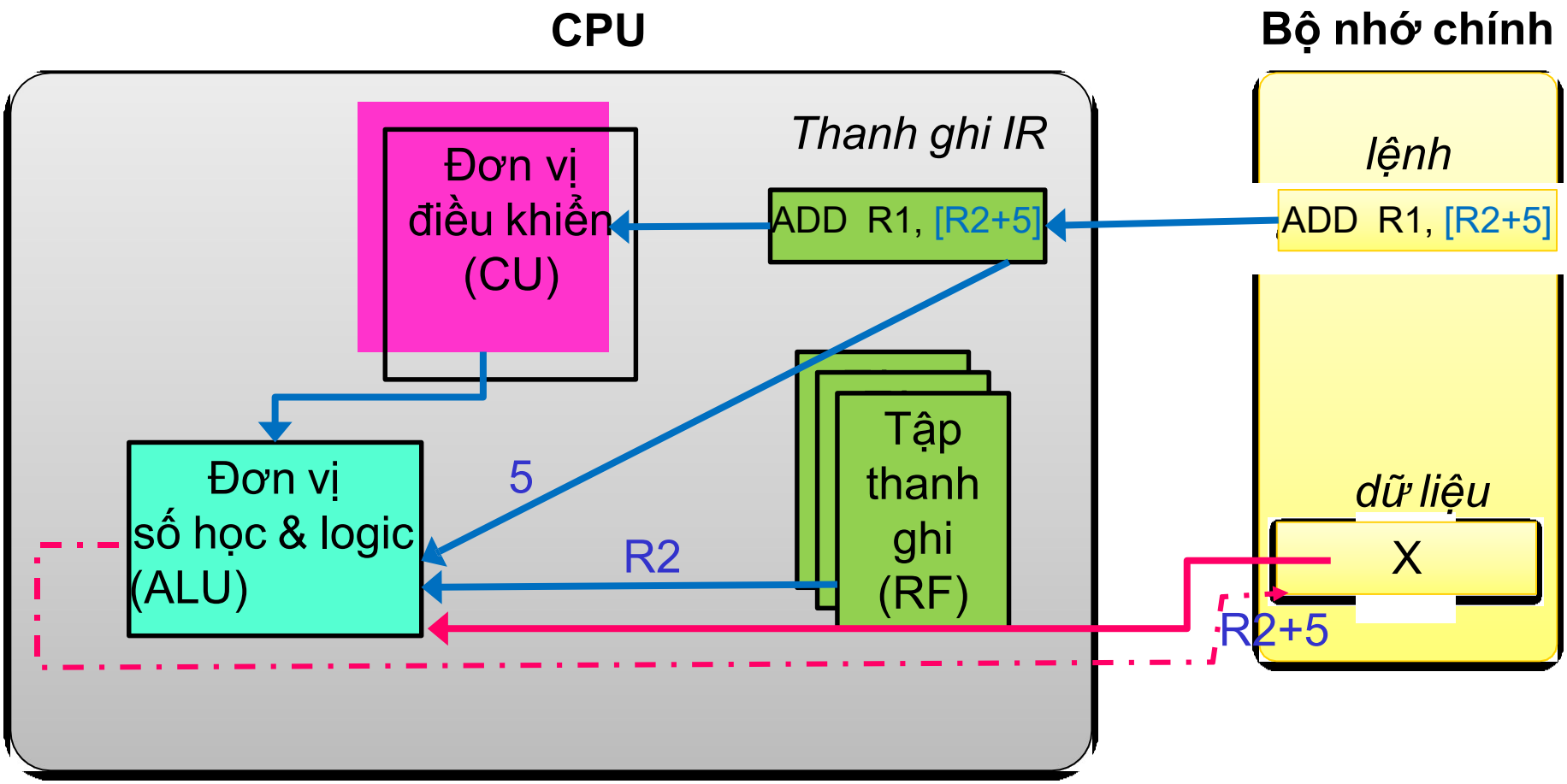


Chế độ địa chỉ dịch chuyển

- Để xác định toán hạng, Trường địa chỉ chứa hai thành phần:
  - Tên thanh ghi
  - Hằng số (offset)
- Địa chỉ của toán hạng = nội dung thanh ghi + hằng số
- Thanh ghi có thể được ngầm định



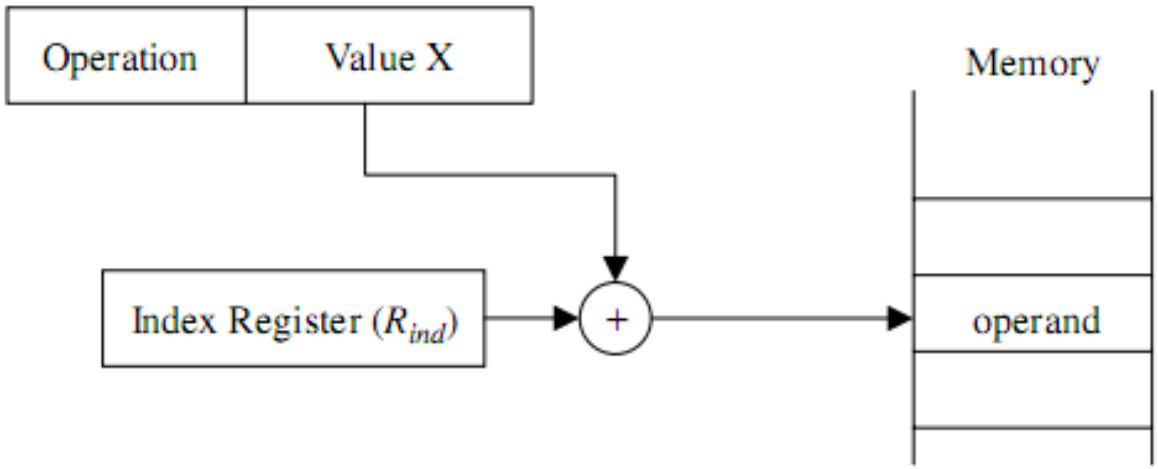
Chế độ địa chỉ dịch chuyển



Chế độ địa chỉ chỉ số

- Địa chỉ của toán hạng có được bằng cách cộng thêm hằng số vào nội dung của một thanh ghi, là thanh ghi chỉ số
- Ví dụ

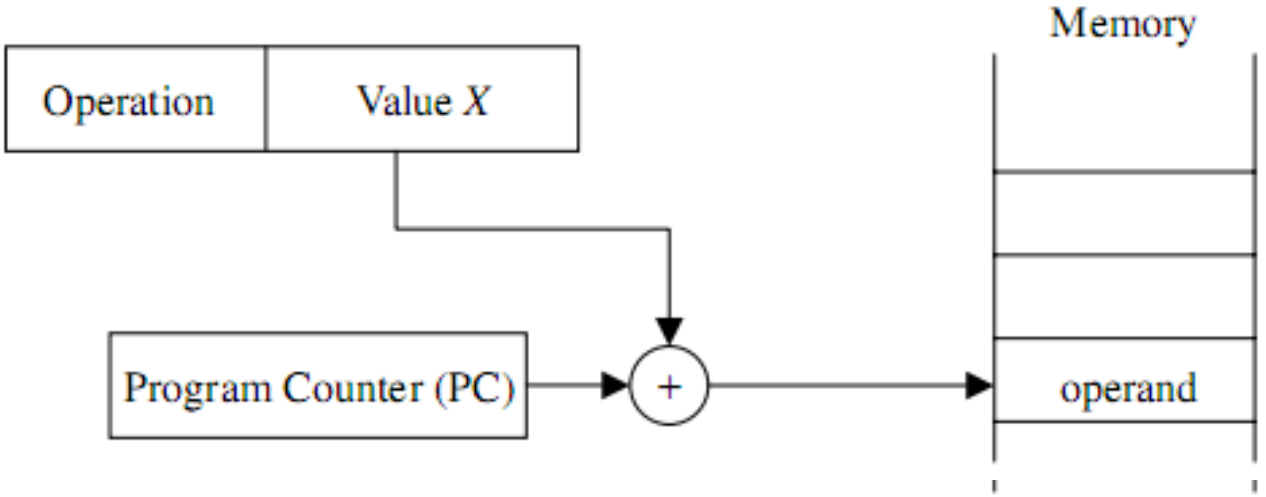
  LOAD R_i, $X(R_{ind})$; $M[X+R_{ind}] \rightarrow R_i$



Chế độ địa chỉ tương đối

- Địa chỉ của toán hạng có được bằng cách cộng thêm hằng số vào nội dung của một thanh ghi, là thanh ghi con đếm chương trình PC
- Ví dụ

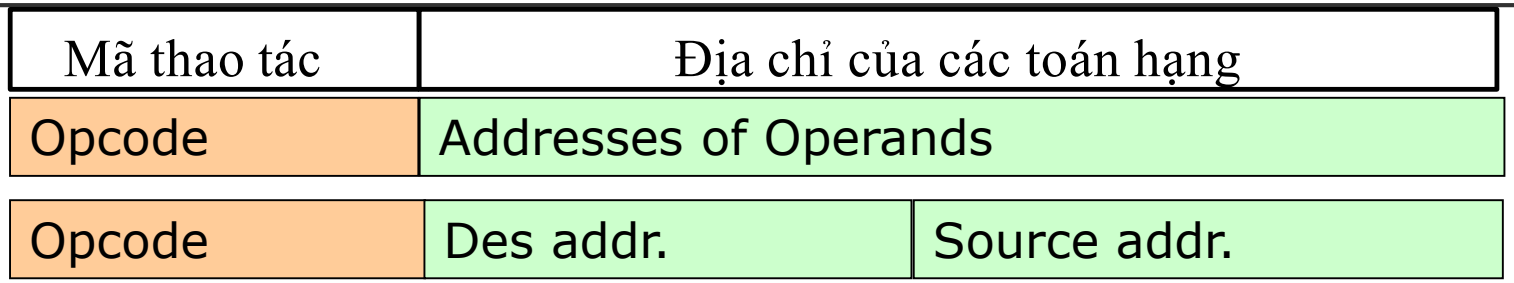
  LOAD R_i, X(PC); M[X+PC] → R_i



Tổng kết các chế độ địa chỉ

Chế độ địa chỉ	Ý nghĩa	Ví dụ	Thực hiện
Tức thì	Giá trị của toán hạng được chứa trong lệnh	LOAD Ri, #1000	Ri ← 1000
Trực tiếp	Địa chỉ của toán hạng được chứa trong lệnh	LOAD Ri, 1000	Ri ← M[1000]
Gián tiếp thanh ghi	Giá trị của thanh ghi trong lệnh là địa chỉ bô nhớ chứa toán hạng	LOAD Ri, (Rj)	Ri ← M[Rj]
Gián tiếp bộ nhớ	Địa chỉ bộ nhớ trong lệnh chứa địa chỉ bộ nhớ của toán hạng	LOAD Ri, (1000)	Ri ← M[M[1000]]
Chỉ số	Địa chỉ của toán hạng là tổng của hằng số (trong lệnh) và giá trị của một thanh ghi chỉ số	LOAD Ri, X(Rind)	Ri ← M[X+ Rind]
Tương đối	Địa chỉ của toán hạng là tổng của hằng số và giá trị của thanh ghi con đếm chương trình	LOAD Ri, X(PC)	Ri ← M[X+ PC]

Các thành phần của lệnh máy



- Mã thao tác (operation code - opcode): mã hóa cho thao tác mà bộ xử lý phải thực hiện
- Địa chỉ toán hạng: chỉ ra nơi chứa các toán hạng mà thao tác sẽ tác động
  - Toán hạng nguồn (source operand): dữ liệu vào của thao tác
  - Toán hạng đích (destination operand): dữ liệu ra của thao tác

Toán hạng 3 địa chỉ

- Khuôn dạng:
  - opcode addr1, addr2, addr3
  - Mỗi địa chỉ addr1, addr2, addr3: tham chiếu tới một ô nhớ hoặc 1 thanh ghi
- Ví dụ
  1. ADD R_1, R_2, R_3; $R_2 + R_3 \rightarrow R_1$

     R_2 cộng R_3 sau đó kết quả đưa vào R_1

     R_i là các thanh ghi CPU
  2. ADD A, B, C; M[B]+M[C] → M[A]

     A, B, C là các vị trí trong bộ nhớ

Toán hạng 2 địa chỉ

- Khuôn dạng:
  - opcode addr1, addr2
  - Mỗi địa chỉ addr1, addr2: tham chiếu tới 1 thanh ghi hoặc 1 vị trí trong bộ nhớ
- Ví dụ
  1. ADD R_1, R_2; $R_1 + R_2 \rightarrow R_1$

     R_1 cộng R_2 sau đó kết quả đưa vào R_1

     R_i là các thanh ghi CPU
  2. ADD A, B; M[A]+M[B] → M[A]

     A, B là các vị trí trong bộ nhớ

Toán hạng 1 địa chỉ

- Khuôn dạng:
  - opcode addr
  - addr: tham chiếu tới 1 thanh ghi hoặc 1 vị trí trong bộ nhớ
  - Khuôn dạng này sử dụng R_{acc} (thanh ghi tích lũy) mặc định cho địa chỉ thứ 2
- Ví dụ
  1. ADD R_1; $R_1 + R_{acc} \rightarrow R_{acc}$
     R_1 cộng R_{acc} sau đó kết quả đưa vào R_{acc}
     R_i là các thanh ghi CPU
  2. ADD A; $M[A]+R_{acc} \rightarrow R_{acc}$
     A là vị trí trong bộ nhớ

Toán hạng 1.5 địa chỉ

- Khuôn dạng:
  - opcode addr1, addr2
  - Một địa chỉ tham chiếu tới 1 ô nhớ và địa chỉ còn lại tham chiếu tới 1 thanh ghi
  - Là dạng hỗn hợp giữa các toán hạng thanh ghi và vị trí bộ nhớ
- Ví dụ
  1. ADD R_1, B; M[B] + R_1 → R_1

Toán hạng 0 địa chỉ

- Được thực hiện trong các lệnh mà thực hiện các thao tác ngăn xếp: push & pop
  - Ví dụ:

push a

push b

add pop c

có nghĩa là : c = a+b

Một số dạng lệnh thông dụng

- Các lệnh vận chuyển dữ liệu
- Các lệnh số học và logic
- Các lệnh điều khiển chương trình
- Các lệnh vào/ ra

Lệnh vận chuyển dữ liệu

- Chuyển dữ liệu giữa các phần của máy tính
  - Giữa các thanh ghi trong CPU

    MOVE Ri, Rj ; Rj -> Ri
  - Giữa thanh ghi CPU và một vị trí trong bộ nhớ

    MOVE Rj,1000; M[1000] -> Rj
  - Giữa các vị trí trong bộ nhớ

    MOVE 1000, (Rj) ; M[Rj] -> M[1000]

Một số lệnh vận chuyển dữ liệu thông dụng

- MOVE: chuyển dữ liệu giữa thanh ghi – thanh ghi, ô nhớ - thanh ghi, ô nhớ - ô nhớ
- LOAD: nạp nội dung 1 ô nhớ vào 1 thanh ghi
- STORE: lưu nội dung 1 thanh ghi ra 1 ô nhớ
- PUSH: đẩy dữ liệu vào ngăn xếp
- POP: lấy dữ liệu ra khỏi ngăn xếp

Lệnh số học và logic

- Thực hiện các thao tác số học và logic giữa các thanh ghi và nội dung ô nhớ
- Ví dụ:

ADD R1, R2, R3; R2 + R3 -> R1

SUBSTRACT R1, R2, R3; R2 – R3 -> R1

Các lệnh tính toán số học thông dụng

- ADD: cộng 2 toán hạng
- SUBSTRACT: trừ 2 toán hạng
- MULTIPLY: nhân 2 toán hạng
- DIVIDE: chia số học
- INCREMENT: tăng 1
- DECREMENT: giảm 1

Các lệnh logic thông dụng

- NOT: phủ định
- AND: và
- OR: hoặc
- XOR: hoặc loại trừ
- COMPARE: so sánh
- SHIFT: dịch
- ROTATE: quay

Lệnh điều khiển/ tuần tự

- Được dùng để thay đổi trình tự các lệnh được thực hiện:
  - Các lệnh rẽ nhánh (nhẩy) có điều kiện (conditional branching/ jump)
  - Các lệnh rẽ nhánh (nhẩy) không điều kiện (unconditional branching/ jump)
  - CALL và RETURN: lệnh gọi thực hiện và trở về từ chương trình con
- Đặc tính chung của các lệnh này là quá trình thực hiện lệnh của chúng làm thay đổi giá trị PC
- Sử dụng các cờ ALU để xác định các điều kiện

Một số lệnh điều khiển thông dụng

- BRANCH – IF – CONDITION: chuyển đến thực hiện lệnh ở địa chỉ mới nếu điều kiện là đúng
- JUMP: chuyển đến thực hiện lệnh ở địa chỉ mới
- CALL: chuyển đến thực hiện chương trình con
- RETURN: trở về (từ chương trình con) thực hiện tiếp chương trình gọi

Một số lệnh điều khiển thông dụng

```
    LOAD  R1, #100
LAP:
    ADD R0, (R2)
    DECREMENT  R1
    BRANCH_IF >0 LAP
```

Các lệnh vào/ ra

- Được dùng để truyền dữ liệu giữa máy tính và các thiết bị ngoại vi
- Các thiết bị ngoại vi giao tiếp với máy tính thông qua các cổng. Mỗi cổng có một địa chỉ dành riêng
- Hai lệnh I/O cơ bản được sử dụng là các lệnh INPUT và OUTPUT
  - Lệnh INPUT được dùng để chuyển dữ liệu từ thiết bị ngoại vi vào tới bộ vi xử lý
  - Lệnh OUTPUT dùng để chuyển dữ liệu từ VXL ra thiết bị đầu ra

Các ví dụ

```
    CLEAR R0;
    MOVE R1, #100;
    CLEAR R2;
LAP:
    ADD R0, 1000(R2);
    INCREMENT R2;
    DECREMENT R1;
    BRANCH_IF>0  LAP;
    STORE 2000, R0;
```

Các ví dụ

CLEAR R0;	R0← 0
MOVE R1, #100;	R1← 100
CLEAR R2;	R2← 0
LAP:	
ADD R0, 1000(R2);	R0←R0+ M[R2+1000]
INCREMENT R2;	R2← R2+1
DECREMENT R1;	R1← R1-1
BRANCH_IF>0 LAP;	go to LAP if R1>0
STORE 2000, R0;	M[2000]← R0

BÀI TẬP

1. Cho đoạn lệnh sau:

   ADD R2, (R0);

   SUBSTRACT R2, (R1);

   MOVE 500(R0), R2;

   LOAD R2, #5000;

   STORE 100(R2), R0;

Biết R0=1500, R1=4500, R2=1000, M[1500]=3000, M[4500]=500

a. Chỉ rõ chế độ địa chỉ của từng lệnh

b. Hãy chỉ ra giá trị của thanh ghi và tại vị trí trong bộ nhớ qua mỗi lệnh thực hiện.

BÀI TẬP

2. Cho đoạn lệnh sau:

```
        MOVE R0, #100;
        CLEAR R1;
        CLEAR R2;
LAP:
        ADD R1, 2000(R2);
        ADD R2, #2;
        DECREMENT R0;
        BRANCH_IF>0 LAP;
        STORE 3000, R1;
```

a. Hãy giải thích ý nghĩa của từng lệnh
b. Chỉ ra chế độ địa chỉ của từng lệnh (đối với các lệnh có 2 toán hạng)
c. Đoạn lệnh trên thực hiện công việc gì?

BÀI TẬP

- Cho một mảng gồm 10 số, được lưu trữ liên tiếp nhau trong bộ nhớ, bắt đầu từ vị trí ô nhớ 1000. Viết đoạn chương trình tính tổng các số dương trong mảng đó và lưu kết quả vào ô nhớ 2000.

CISC và RISC

- CISC: Complex Instruction Set Computer:
  - Máy tính với tập lệnh phức tạp
  - Các bộ xử lý truyền thống: Intel x86, Motorola 680x0
- RISC: Reduced Instruction Set Computer:
  - Máy tính với tập lệnh thu gọn
  - SunSPARC, Power PC, MIPS, ARM ...
  - RISC đối nghịch với CISC
  - Kiến trúc tập lệnh tiên tiến