



TRƯỜNG ĐẠI HỌC THỦY LỢI  
KHOA CÔNG NGHỆ THÔNG TIN



# KIẾN TRÚC MÁY TÍNH

Giảng viên: TS. Đoàn Thị Quế

Bộ môn: Mạng và an toàn thông tin

# Chương 10. Tập lệnh: Các chế độ định địa chỉ và định dạng lệnh

- 10.1 Các chế độ địa chỉ
- 10.2 Định dạng lệnh

# 10.1 Các chế độ địa chỉ

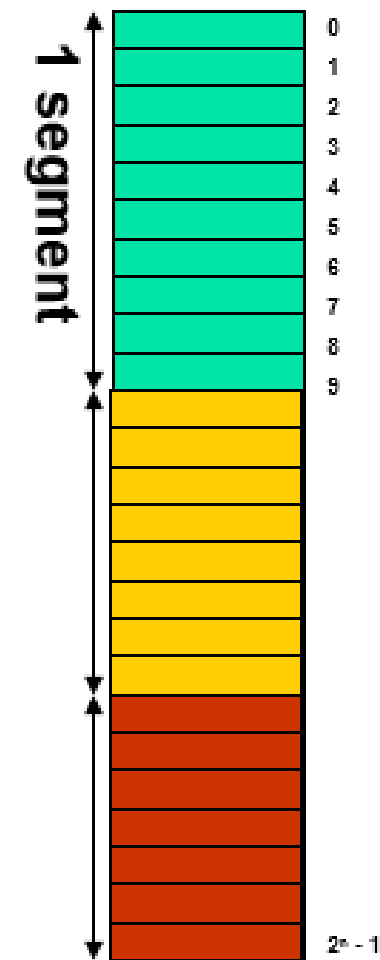
1. Giới thiệu về chế độ định địa chỉ
2. Các chế độ định địa chỉ phổ biến

# 1. Giới thiệu về chế độ định địa chỉ

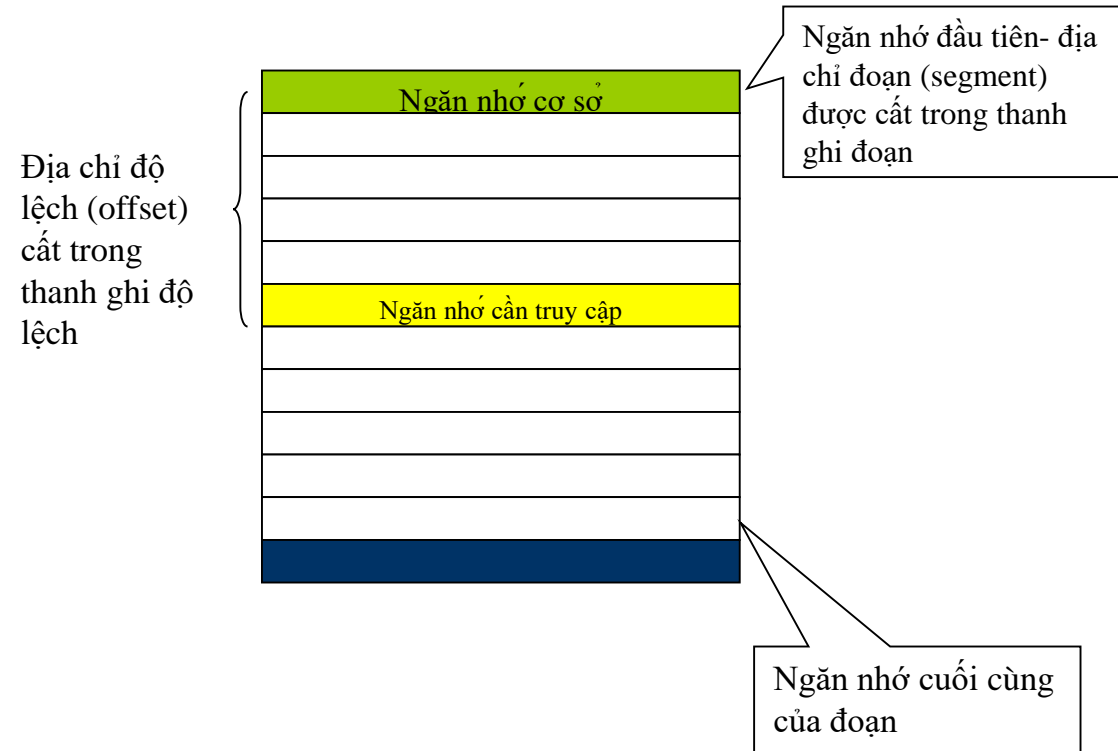
- Chế độ định địa chỉ: là cách thức địa chỉ hóa trong trường địa chỉ của lệnh để xác định nơi chứa toán hạng
- Toán hạng của lệnh có thể đặt ở đâu?
  - Một giá trị cụ thể nằm ngay trong lệnh (chỉ có với toán hạng nguồn)
  - Đặt ở thanh ghi → phải cho biết tên của thanh ghi
  - Đặt ở ngăn nhớ → phải cho biết địa chỉ của ngăn nhớ
  - ...

# Đánh địa chỉ cho bộ nhớ

- Địa chỉ vật lý của ngăn nhớ:
  - Bộ nhớ gồm một dãy các ngăn nhớ được đánh địa chỉ theo thứ tự tăng dần bắt đầu từ 0 tới  $2^n - 1$  (địa chỉ vật lý)
  - Số ngăn nhớ lớn  $\rightarrow$  không gian địa chỉ bộ nhớ lớn  $\rightarrow$  khó quản lý
- Địa chỉ logic của ngăn nhớ :
  - Bộ nhớ được chia thành các đoạn (Segment)



# Đánh địa chỉ cho bộ nhớ (tiếp)



Địa chỉ logic của ngăn nhớ cần truy cập:

**Địa chỉ đoạn (Segment) : Địa chỉ lệch (Offset)**

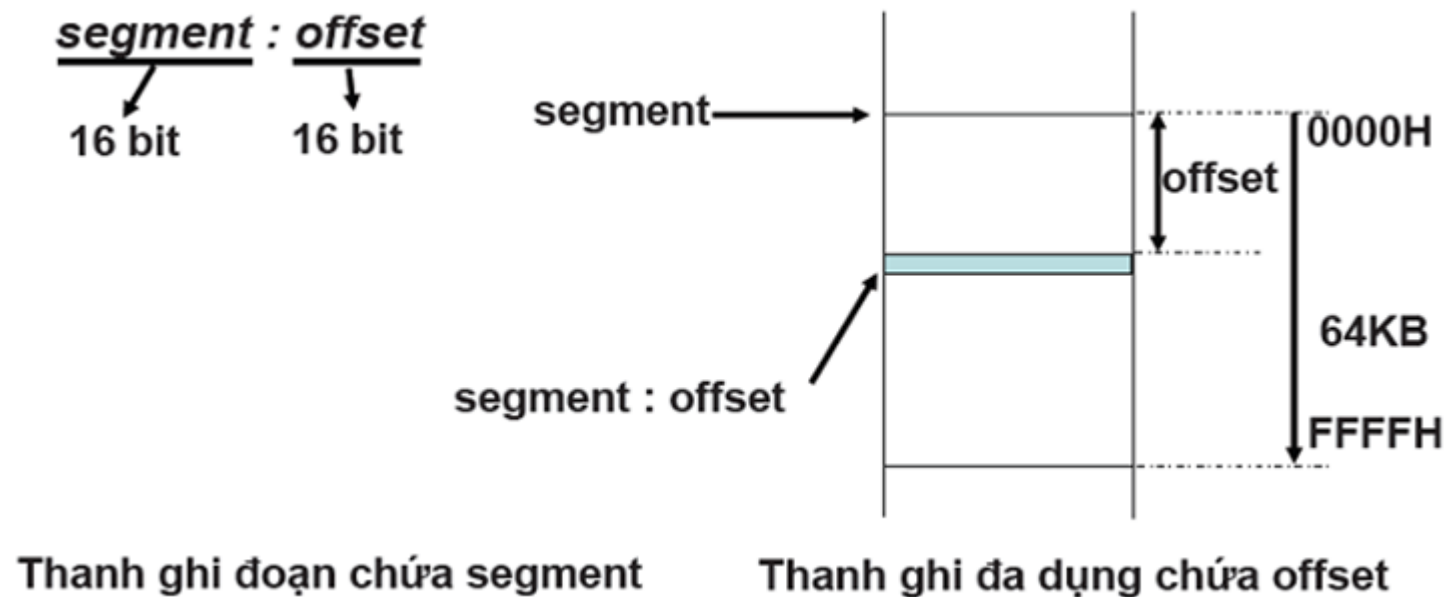
# Đánh địa chỉ cho bộ nhớ (tiếp)

## Nhận xét:

- Chương trình (các lệnh CPU) luôn dùng địa chỉ logic
- Khi truy xuất bộ nhớ, CPU sẽ chuyển đổi địa chỉ logic sang địa chỉ vật lý để truy xuất bộ nhớ

# Bộ nhớ CPU Intel 8086

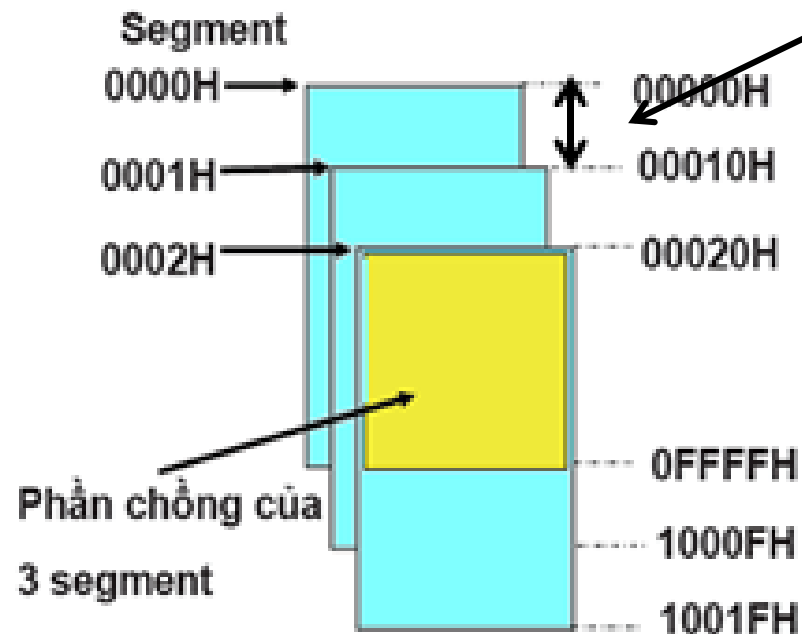
- Không gian địa chỉ vật lý (A19-A0):  $00000_H \rightarrow FFFFF_H$  (1MB)
- Địa chỉ logic gồm hai thành phần:





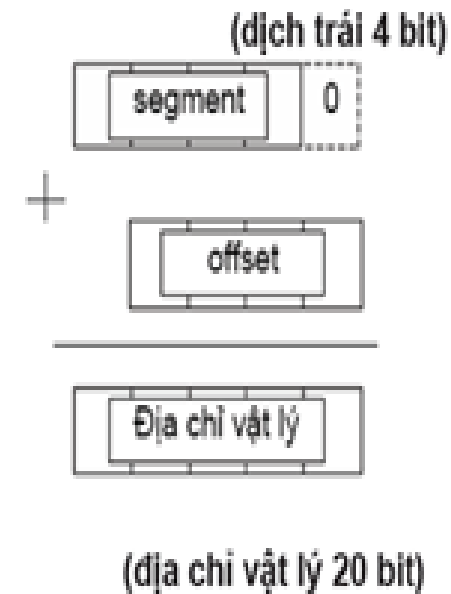
# Bộ nhớ CPU Intel 8086 (tiếp)

Địa chỉ vật lý = (segment \* 16) + offset



Khoảng cách giữa hai đoạn

$$\frac{2^{20}}{2^{16}} \text{B} = 16\text{B}$$



# Ví dụ 1

Cặp thanh ghi CS:IP chứa địa chỉ logic

a.  $F0F0_H : F0F0_H$

b.  $FFFF_H : 0000_H$

Hãy tính địa chỉ vật lý của các địa chỉ logic trên?

	a/	b/
	$F0F00_H$ + <u><math>F0F0_H</math></u>	$FFFF0_H$ + <u><math>0000_H</math></u>
Kết quả	$FFFF0_H$	$FFFF0_H$

## Ví dụ 2

Tìm địa chỉ vật lý từ các địa chỉ logic sau:

a.  $12A4_H : 0022_H$

b.  $11CD_H : 0D92_H$

	a/	b/
	$12A40_H$ + <u><math>0022_H</math></u>	$11CD0_H$ + <u><math>0D92_H</math></u>
<b>Kết quả</b>	<b><math>12A62_H</math></b>	<b><math>12A62_H</math></b>

# Nhận xét

- Một địa chỉ vật lý có thể được xác định thông qua nhiều địa chỉ logic khác nhau
- Nhưng với mỗi địa chỉ logic duy nhất chỉ xác định được 1 địa chỉ vật lý

# Ngầm định thanh ghi đoạn

- Một ngăn nhớ được xác định bởi 2 địa chỉ:
  - Địa chỉ ngăn nhớ = Địa chỉ đoạn: Địa chỉ lệch
- Ngầm định thanh ghi đoạn
  - Ngăn nhớ = [Tên thanh ghi lệch]

# Ví dụ thanh ghi ngầm định của VXL 8086

- $[BP] = SS:BP$
- $[SP] = SS:SP$
- $[SI] = DS:SI$
- $[BX] = DS:BX$
- $[DI] = DS:DI$
- $[IP] = CS:IP$

## 2. Các chế độ định địa chỉ phổ biến

- a. Tức thì
- b. Trực tiếp
- c. Gián tiếp
- d. Thanh ghi
- e. Gián tiếp thanh ghi
- f. Dịch chuyển
- g. Ngăn xếp

## Các chế độ định địa chỉ phổ biến

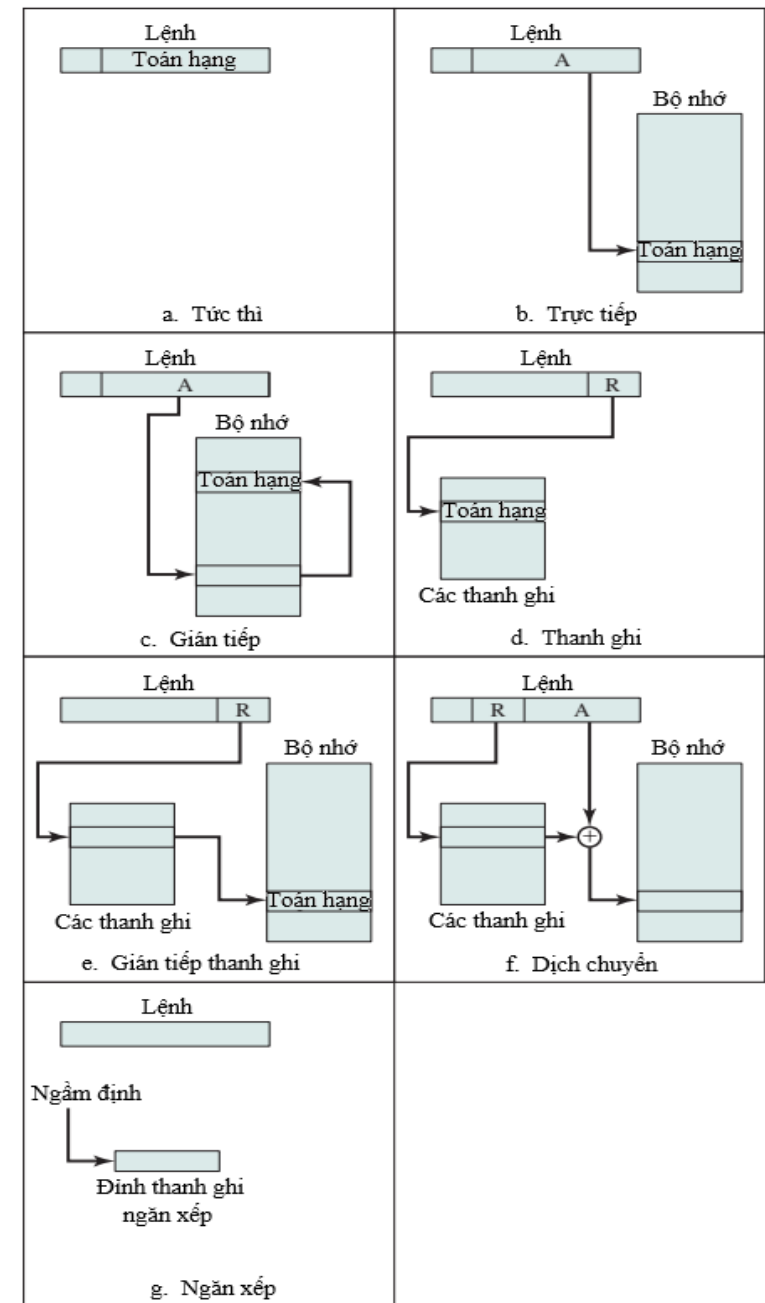
Chế độ	Thuật toán	Ưu điểm	Nhược điểm
Tức thì	Toán hạng = A	Không cần tham chiếu bộ nhớ	Hạn chế về giá trị của toán hạng
Trực tiếp	EA = A	Đơn giản	Không gian địa chỉ hạn chế
Gián tiếp	EA = (A)	Không gian địa chỉ lớn	Tham chiếu bộ nhớ nhiều lần
Thanh ghi	EA = R	Không cần tham chiếu bộ nhớ	Không gian địa chỉ hạn chế
Gián tiếp thanh ghi	EA = A + (R)	Linh hoạt	Phức tạp
Ngăn xếp	EA = đỉnh ngăn xếp	Không cần tham chiếu bộ nhớ	Khả năng ứng dụng ít

A = nội dung trường địa chỉ trong lệnh

R = nội dung trường địa chỉ trong lệnh tham chiếu đến một thanh ghi

EA = địa chỉ hiệu dụng của một vị trí chứa toán hạng được tham chiếu

(X) = nội dung của vị trí bộ nhớ X hoặc thanh ghi X





# a. Định địa chỉ tức thì

- Dạng đơn giản nhất của định địa chỉ
- Toán hạng được đặt trong trường địa chỉ của lệnh

Toán hạng = A

- Chế độ này có thể được sử dụng để định nghĩa và sử dụng các hằng số và thiết lập các giá trị ban đầu của biến
  - Các số thường được lưu trữ dưới dạng số bù hai
  - Bit ngoài cùng bên trái của trường toán hạng được sử dụng như bit dấu
- Ưu điểm:
  - Chỉ cần truy xuất bộ nhớ một lần (để lấy lệnh), do vậy tiết kiệm một chu kỳ cache hoặc bộ nhớ trong chu kỳ lệnh.
- Nhược điểm:
  - Kích thước của toán hạng bị giới hạn bởi kích thước của trường địa chỉ vì thông thường kích thước của trường này nhỏ hơn kích thước từ



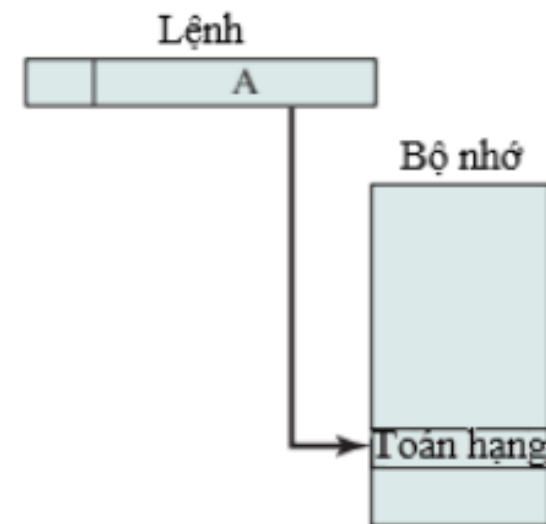
Ví dụ:

ADD R1, 5 ; R1 ← R1 + 5

Trong đó, 5 là một toán hạng nguồn được tham chiếu trực tiếp trong câu lệnh: địa chỉ tức thì

## b. Định địa chỉ trực tiếp

- Toán hạng đặt trong bộ nhớ
- Trường địa chỉ chứa địa chỉ hiệu dụng của toán hạng
  - Địa chỉ hiệu dụng (Effective address (EA)) = trường địa chỉ (Address field (A))
- Phổ biến trong các thế hệ máy tính trước đây
- Tham chiếu bộ nhớ một lần để lấy toán hạng
- Hạn chế: chỉ cung cấp một không gian địa chỉ hạn chế



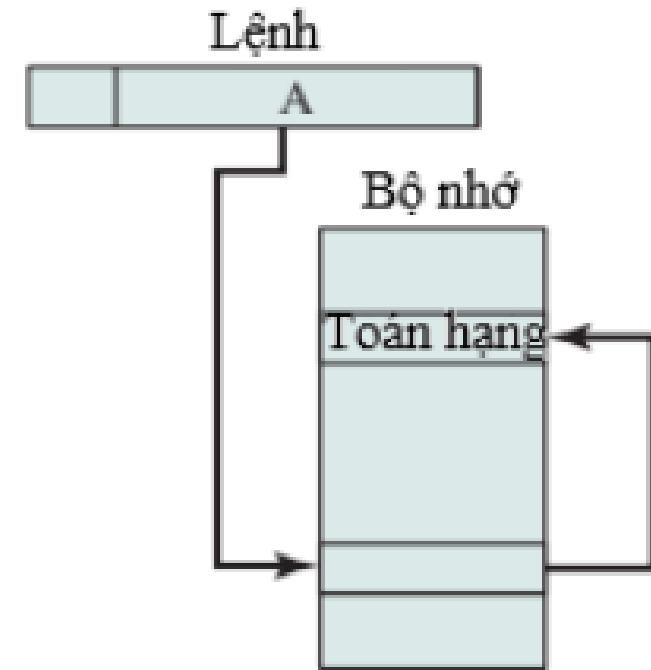
Ví dụ:

`ADD R1, A` ;  $R1 \leftarrow R1 + (A)$

Trong đó, A là địa chỉ một vị trí bộ nhớ, (A) là nội dung của vị trí đó. CPU tham chiếu đến toán hạng có địa chỉ A trong bộ nhớ.

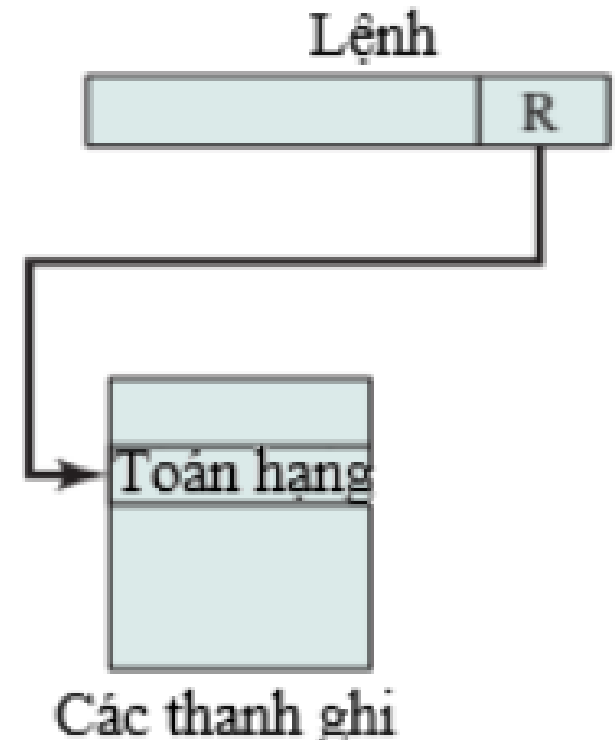
## c. Định địa chỉ gián tiếp

- Toán hạng đặt trong một từ nhớ có địa chỉ để trong một từ nhớ khác trong bộ nhớ
- Tham chiếu đến địa chỉ của một từ trong bộ nhớ chứa địa chỉ đầy đủ của toán hạng
  - $EA = (A)$
  - Dấu ngoặc đơn được hiểu như là *nội dung của vị trí A*
- Ưu điểm:
  - Với một từ có kích thước N cho phép một không gian địa chỉ là  $2^N$
- Nhược điểm:
  - Thực thi câu lệnh đòi hỏi hai lần tham chiếu bộ nhớ để truy xuất toán hạng
    - Một để lấy ra địa chỉ, hai là để lấy ra giá trị của nó
- Một biến thể hiếm gặp của địa chỉ gián tiếp là địa chỉ gián tiếp nhiều cấp hoặc nhiều tầng
  - $EA = ( \dots (A) \dots )$
  - Nhược điểm là cần ba hoặc nhiều hơn tham chiếu bộ nhớ để truy xuất toán hạng



## d. Định địa chỉ thanh ghi

- Toán hạng nằm trong thanh ghi
- Trường địa chỉ dùng để tham chiếu thanh ghi
  - $EA = R$
- Ưu điểm:
  - Chỉ cần một trường địa chỉ nhỏ trong lệnh (do số lượng thanh ghi ít)
  - Không cần tham chiếu bộ nhớ
- Nhược điểm:
  - Không gian địa chỉ giới hạn



Ví dụ:

ADD R1, A ;  $R1 \leftarrow R1 + (A)$

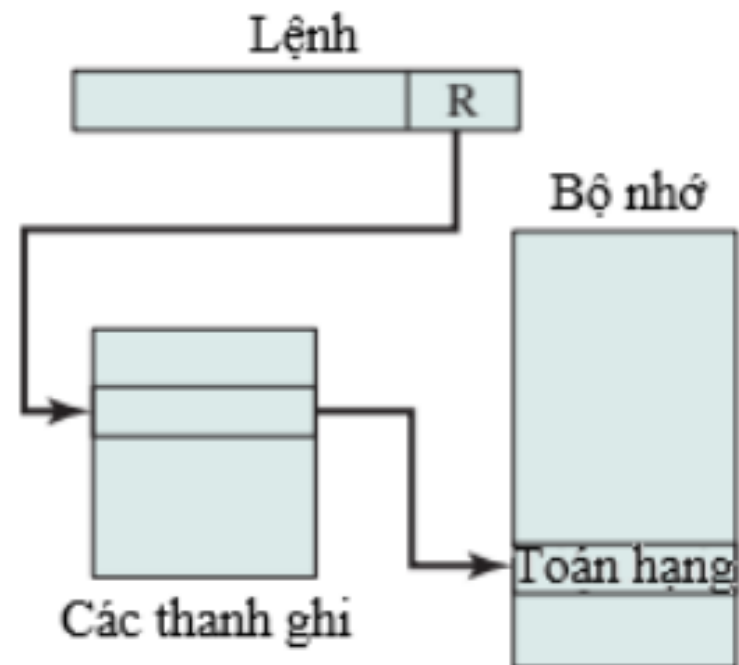
Trong đó, R1 là một thanh ghi trong bộ xử lý, câu lệnh trên tham chiếu giá trị (A) trong bộ nhớ cộng với giá trị lưu trữ trong thanh ghi R1, kết quả được ghi vào R1.

■ Ví dụ:

- ADD AL, DL ; Cộng nội DL với nội dung AL
- MOV BX, DX ; Copy nội dung DX vào BX
- MOV AL, BL ; Copy nội dung BL vào AL
- MOV AL, BX ; không hợp lệ vì các thanh ghi có kích thước khác nhau
- MOV ES, DS ; không hợp lệ (segment to segment)
- MOV CS, AX ; không hợp lệ vì CS không được dùng làm thanh ghi đích

## e. Định địa chỉ gián tiếp thanh ghi

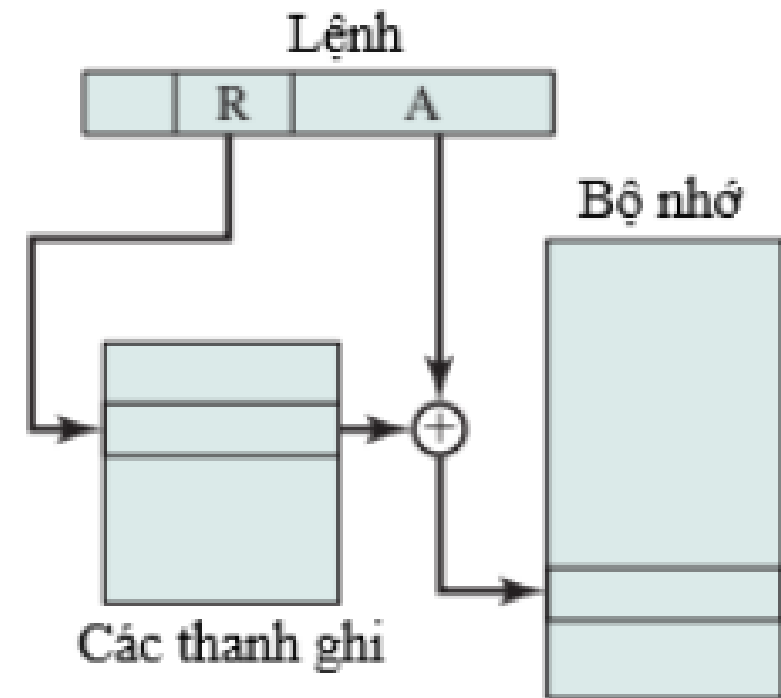
- Tương tự như địa chỉ gián tiếp
  - Sự khác biệt duy nhất là trường địa chỉ tham chiếu đến thanh ghi
  - $EA = (R)$
- Không gian địa chỉ lớn hơn (trường địa chỉ tham chiếu đến vị trí chứa địa chỉ có độ dài bằng một từ )
- Tham chiếu bộ nhớ ít hơn định địa chỉ gián tiếp



- Ví dụ:
  - `MOV AL, [BX]` ; Copy nội dung ô nhớ có địa chỉ DS:BX vào AL
  - `MOV [DI], AX` ; copy nội dung của AX vào 2 ô nhớ liên tiếp DS: DI và DS: (DI +1)

# f. Định địa chỉ dịch chuyển - Displacement Addressing

- Kết hợp chế độ định địa chỉ trực tiếp và định địa chỉ gián tiếp thanh ghi
  - $EA = A + (R)$
- Yêu cầu lệnh phải có hai trường địa chỉ, ít nhất một trong hai phải có giá trị cụ thể
  - Một giá trị trong một trường địa chỉ (giá trị = A) được sử dụng trực tiếp
  - Một trường địa chỉ khác tham chiếu đến thanh ghi trong đó nội dung của nó được cộng với A để tạo ra địa chỉ hiệu dụng
  - Ví dụ:  
`MOV AL, [BP]+5` ; copy nội dung của ô nhớ SS:BP+5 vào thanh ghi AL
- Có ba cách sử dụng phổ biến nhất của chế độ địa chỉ dịch chuyển:
  - Định địa chỉ tương đối
  - Định địa chỉ thanh ghi cơ sở
  - Định địa chỉ chỉ mục





# Định địa chỉ tương đối

- Thanh ghi được tham chiếu ngầm là thanh ghi PC (program counter)
  - Địa chỉ lệnh tiếp theo được cộng vào trường địa chỉ để tạo ra EA
  - Thông thường trường địa chỉ được coi là dữ liệu dạng số bù 2 của hoạt động này
  - Do đó, địa chỉ hiệu dụng của toán hạng dịch chuyển một đoạn so với địa chỉ của lệnh.
- Địa chỉ tương đối khai thác tính cục bộ của bộ nhớ:
  - Nếu hầu hết các toán hạng được tham chiếu gần với lệnh đang thực hiện, thì việc sử dụng địa chỉ tương đối sẽ giúp giảm bớt số lượng bit địa chỉ cần thiết trong lệnh.

# Định địa chỉ thanh ghi cơ sở

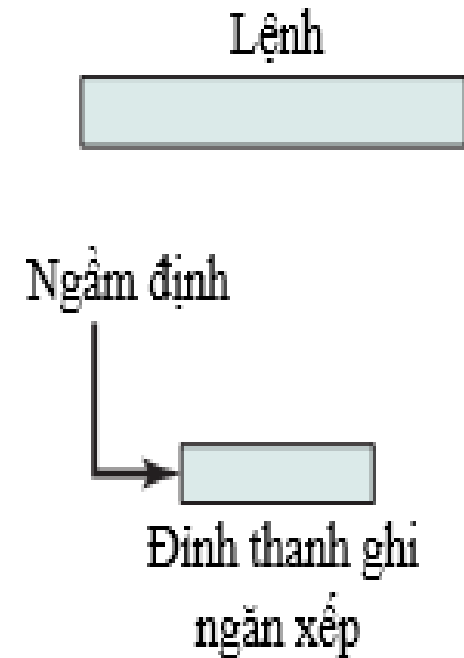
- Thanh ghi được tham chiếu chứa một địa chỉ bộ nhớ chính và trường địa chỉ chứa một giá trị dịch chuyển so với địa chỉ này
- Thanh ghi được tham chiếu có thể được chỉ rõ trong lệnh hoặc ngầm định.
- Khai thác tính cục bộ của tham chiếu bộ nhớ
- Chế độ địa chỉ này giúp thực hiện một kỹ thuật phân đoạn bộ nhớ chính
  - Trong một số trường hợp, một thanh ghi đoạn ngầm định sẽ chứa địa chỉ đoạn
  - Một số trường hợp khác, người lập trình có thể lựa chọn một thanh ghi chứa địa chỉ cơ sở đoạn và lệnh phải tham chiếu đến thanh ghi này.

# Định địa chỉ chỉ số - Indexed Addressing

- Trường địa chỉ tham chiếu đến một địa chỉ bộ nhớ chính và đến một thanh ghi chứa một giá trị dịch chuyển dương từ địa chỉ đó.
- Phương pháp này tính toán EA giống như với định địa chỉ thanh ghi cơ sở
- Một ứng dụng quan trọng: cung cấp một cơ chế hiệu quả để thực hiện các hoạt động lặp
- Autoindexing
  - Tự động tăng hoặc giảm thanh ghi chỉ mục sau mỗi tham chiếu đến nó
  - $EA = A + (R)$
  - $(R) = (R) + 1$
- Postindexing
  - Chế độ địa chỉ chỉ số được thực hiện sau chế độ địa chỉ gián tiếp
  - $EA = (A) + (R)$
- Preindexing
  - Chế độ địa chỉ chỉ số được thực hiện trước chế độ địa chỉ gián tiếp
  - $EA = (A + (R))$

# Định địa chỉ ngăn xếp

- Một ngăn xếp là một mảng liên tiếp các ô nhớ: phần tử được đưa vào sau sẽ được lấy ra trước
- Thanh ghi SP (stack pointer – con trỏ ngăn xếp) chứa địa chỉ đỉnh ngăn xếp (trỏ vào đỉnh ngăn xếp)
  - Do đó, chế độ địa chỉ ngăn xếp thực chất là chế độ địa chỉ gián tiếp thanh ghi



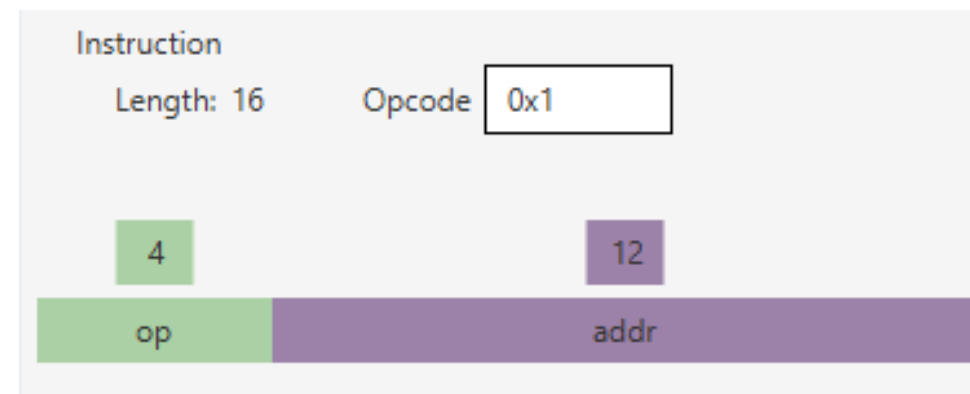
# Chương 10. Tập lệnh: Các chế độ định địa chỉ và định dạng lệnh

- 10.1 Các chế độ địa chỉ
- **10.2 Định dạng lệnh**

## 10.2 Định dạng lệnh

- Một định dạng lệnh định nghĩa bố cục của các bit của lệnh, được chia dưới dạng các trường.
- Định dạng lệnh bao gồm các trường:
  - Mã lệnh (Opcode)
  - Không có hoặc gồm nhiều toán hạng:
    - Mỗi toán hạng sẽ được tham chiếu bằng một trong các chế độ địa chỉ
    - Chế độ địa chỉ được sử dụng phải được chỉ ra trong lệnh hoặc phải được quy ước theo một cách nào đó.
- Với mỗi tập lệnh, có thể có nhiều định dạng được sử dụng
  - Ví dụ: máy giả thuyết CPUSIM

read	0011 0000 0000 0000
jmpn Done	1011 0000 0000 1010



# Một số vấn đề chính trong việc thiết kế tập lệnh

- Kích thước lệnh
- Phân bổ bit
- Các lệnh có độ dài thay đổi

# Kích thước lệnh

- Kích thước lệnh phụ thuộc vào:
  - Kích thước bộ nhớ
  - Tổ chức bộ nhớ
  - Cấu trúc bus
  - Sự phức tạp của bộ vi xử lý
  - Tốc độ bộ vi xử lý
- Kích thước lệnh phải bằng hoặc là bội của đơn vị truyền
  - Ví dụ: bus dữ liệu có kích thước 16b, kích thước lệnh có thể là 16b hoặc 32b tùy kiến trúc
- Kích thước lệnh phải là bội số của chiều dài ký tự (thường là 8 bit) và chiều dài của dạng biểu diễn dấu chấm tĩnh.



# Phân bổ bit

- Việc phân bổ các bit (phân chia các trường) trong một từ lệnh phụ thuộc vào các yếu tố sau:
  - **Số lượng chế độ địa chỉ:** Một số lệnh có thể ngầm định chế độ địa chỉ. Các trường hợp khác, cần phải chỉ rõ chế độ địa chỉ được sử dụng thông qua một hoặc nhiều bit chế độ.
  - **Số lượng toán hạng:** Lệnh càng ít toán hạng, chương trình dài hơn, phức tạp hơn. Các định dạng lệnh ngày nay thường bao gồm hai toán hạng.
  - **Số lượng thanh ghi:** Càng sử dụng nhiều thanh ghi để lưu trữ toán hạng thì cần càng ít bit địa chỉ.
  - **Số lượng tập thanh ghi:** Các thanh ghi được chia thành các nhóm, ví dụ nhóm cho lưu trữ dữ liệu riêng và cho lưu trữ địa chỉ trong chế độ địa chỉ dịch chuyển. Việc sử dụng nhiều nhóm riêng biệt đòi hỏi ít bit hơn để sử dụng trong lệnh.
  - **Dải địa chỉ (không gian có thể đánh địa chỉ được)**
  - **Chế độ đánh địa chỉ bộ nhớ**
- Nếu muốn có đồng thời:
  - Kích thước lệnh hợp lý
  - Khả năng đánh địa chỉ hợp lý
  - Số lượng lớn opcodes

ta có thể sử dụng opcode có kích thước thay đổi

# Định dạng lệnh PDP-8

Các lệnh tham chiếu bộ nhớ

Opcode	D/I	Z/C	Dịch chuyển
0	2	3	4 5 11

Các lệnh vào/ra

1	1	0	Thiết bị	Opcode
0	1	2	3 8	9 11

Các vi lệnh nhóm 1

Các lệnh tham chiếu thanh ghi

1	1	1	0	CLA	CLL	CMA	CML	RAR	RAL	BSW	IAC
0	1	2	3	4	5	6	7	8	9	10	11

Các vi lệnh nhóm 2

1	1	1	0	CLA	SMA	SZA	SNL	RSS	OSR	HLT	0
0	1	2	3	4	5	6	7	8	9	10	11

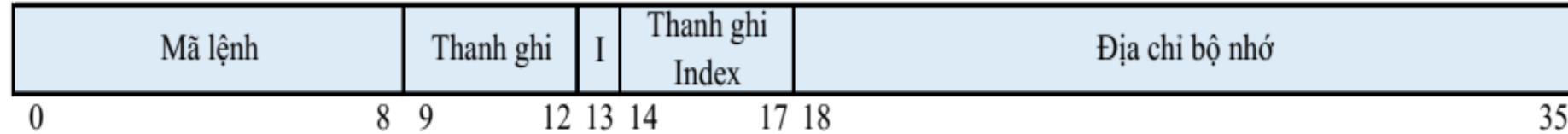
Các vi lệnh nhóm 3

1	1	1	0	CLA	MQA	0	MQL	0	0	0	1
0	1	2	3	4	5	6	7	8	9	10	11

D/I = Địa chỉ trực tiếp/gián tiếp  
 Z/C = Page 0 hoặc page hiện tại  
 CLA = Xóa thanh ghi AC  
 CLL = Xóa liên kết  
 CMA = Thực thi AC  
 CML = Thực thi liên kết  
 RAR = Dịch vòng AC sang phải  
 RAL = Dịch vòng AC sang trái  
 BSW = Đảo byte

IAC = Tăng AC  
 SMA = Bỏ qua nếu AC âm  
 SZA = Bỏ qua nếu AC bằng không  
 SNL = Bỏ qua nếu liên kết khác không  
 RSS = Reverse Skip Sense  
 OSR = Phép Hoặc với thanh ghi Switch  
 HLT = Dừng  
 MQA = Nạp số nhân, thương vào AC  
 MQL = Tải số nhân, thương

# Định dạng lệnh PDP-10

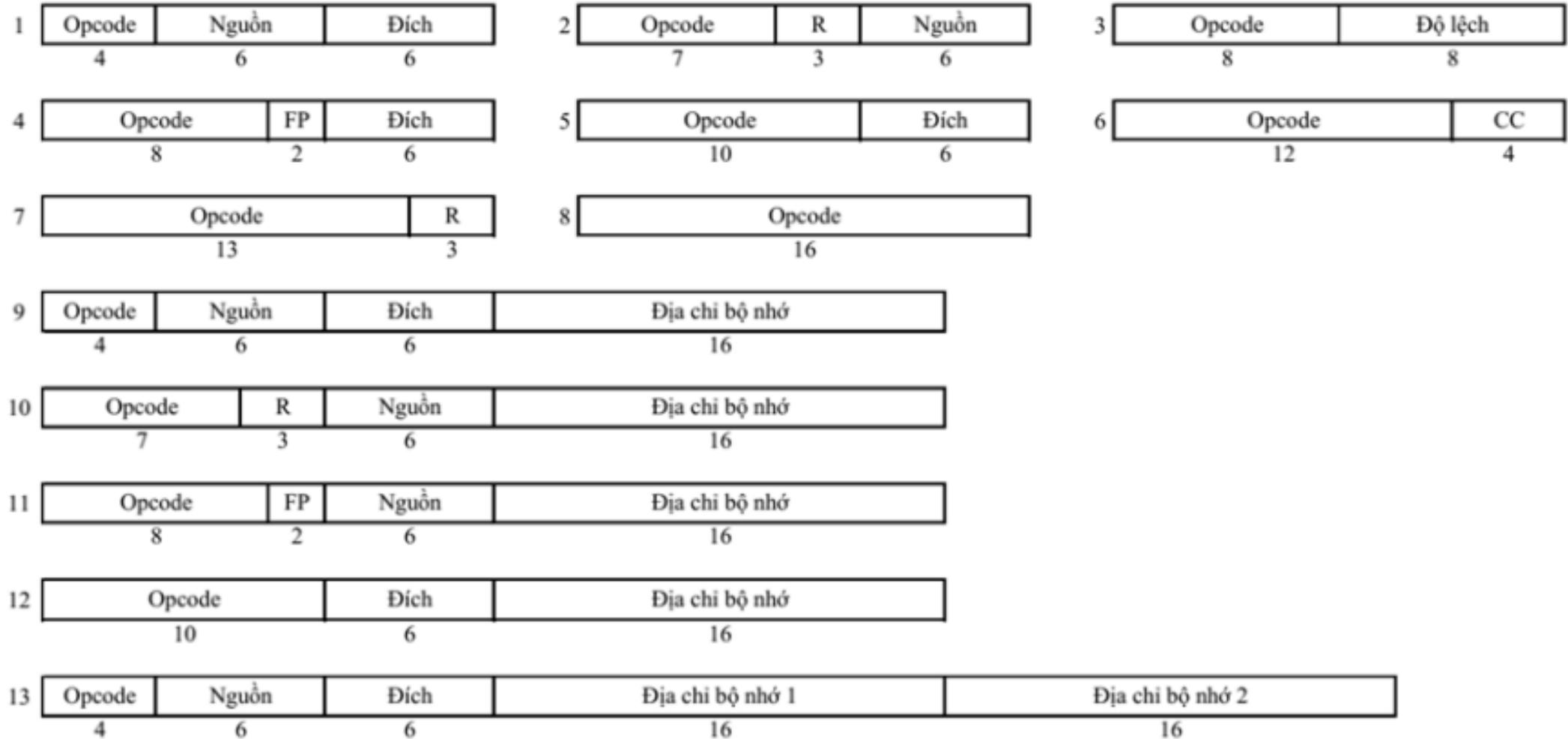


I = bit gián tiếp

# Các lệnh có độ dài thay đổi

- Tập lệnh có độ dài thay đổi: gồm các lệnh có kích thước khác nhau
  - Có thể có nhiều opcode hơn và độ dài opcode khác nhau.
  - Các chế độ địa chỉ sẽ linh hoạt hơn và có thể kết hợp giữa việc tham chiếu thanh ghi và bộ nhớ cùng với các chế độ địa chỉ.
- Ưu điểm: Có nhiều biến thể hiệu quả và gọn gàng.
- Nhược điểm: Tăng sự phức tạp của bộ xử lý
- Lưu ý: Việc sử dụng các lệnh có độ dài thay đổi vẫn phải hướng đến việc độ dài lệnh bằng bội số của kích thước word vì bộ xử lý sẽ không biết độ dài của lệnh tiếp theo được truy xuất.

# Tập lệnh của PDP-11



Các số ở dưới các trường biểu thị số bit của trường tương ứng

Nguồn, Đích: toán hạng nguồn và toán hạng đích. Các trường này gồm 3-bit xác định chế độ địa chỉ và 3-bit xác định một thanh ghi

FP chỉ ra một trong bốn thanh ghi dấu chấm động

R chỉ ra một trong các thanh ghi đa năng

CC là trường mã điều kiện

# Tổng kết

## Chương 10

### Tập lệnh: Các chế độ định địa chỉ và định dạng lệnh

- Các chế độ định địa chỉ
  - Định địa chỉ tức thì
  - Định địa chỉ trực tiếp
  - Định địa chỉ gián tiếp
  - Định địa chỉ thanh ghi
  - Định địa chỉ gián tiếp thanh ghi
  - Định địa chỉ dịch chuyển
  - Định địa chỉ ngăn xếp
- Định dạng lệnh
  - Kích thước lệnh
  - Phân bố các bit
  - Các lệnh có độ dài thay đổi

# Câu hỏi ôn tập

1. Các chế độ định địa chỉ nào không cần truy xuất toán hạng từ bộ nhớ?
2. Các chế độ định địa chỉ nào dành cho các toán hạng nằm trong ngăn nhớ?
3. Chế độ địa chỉ nào cần hai lần truy xuất bộ nhớ để lấy toán hạng?
4. Cấu trúc của mã lệnh bao gồm những trường cơ bản nào? Nếu ý nghĩa của từng trường
5. Kích thước từ lệnh phụ thuộc vào những yếu tố nào?

Hình ảnh và nội dung trong bài giảng này tham khảo từ cuốn sách và slide bài giảng “Computer Organization and Architecture”, 10th Edition, của tác giả William Stallings.