



HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÀI GIẢNG MÔN

KIẾN TRÚC MÁY TÍNH

**CHƯƠNG 3 – BỘ VI XỬ LÝ
INTEL 8086/8088**

Giảng viên:

TS. Nguyễn Văn Thủy

Điện thoại/E-mail:

thuynv@ptit.edu.vn

Bộ môn:

Khoa học máy tính - Khoa CNTT1

NỘI DUNG

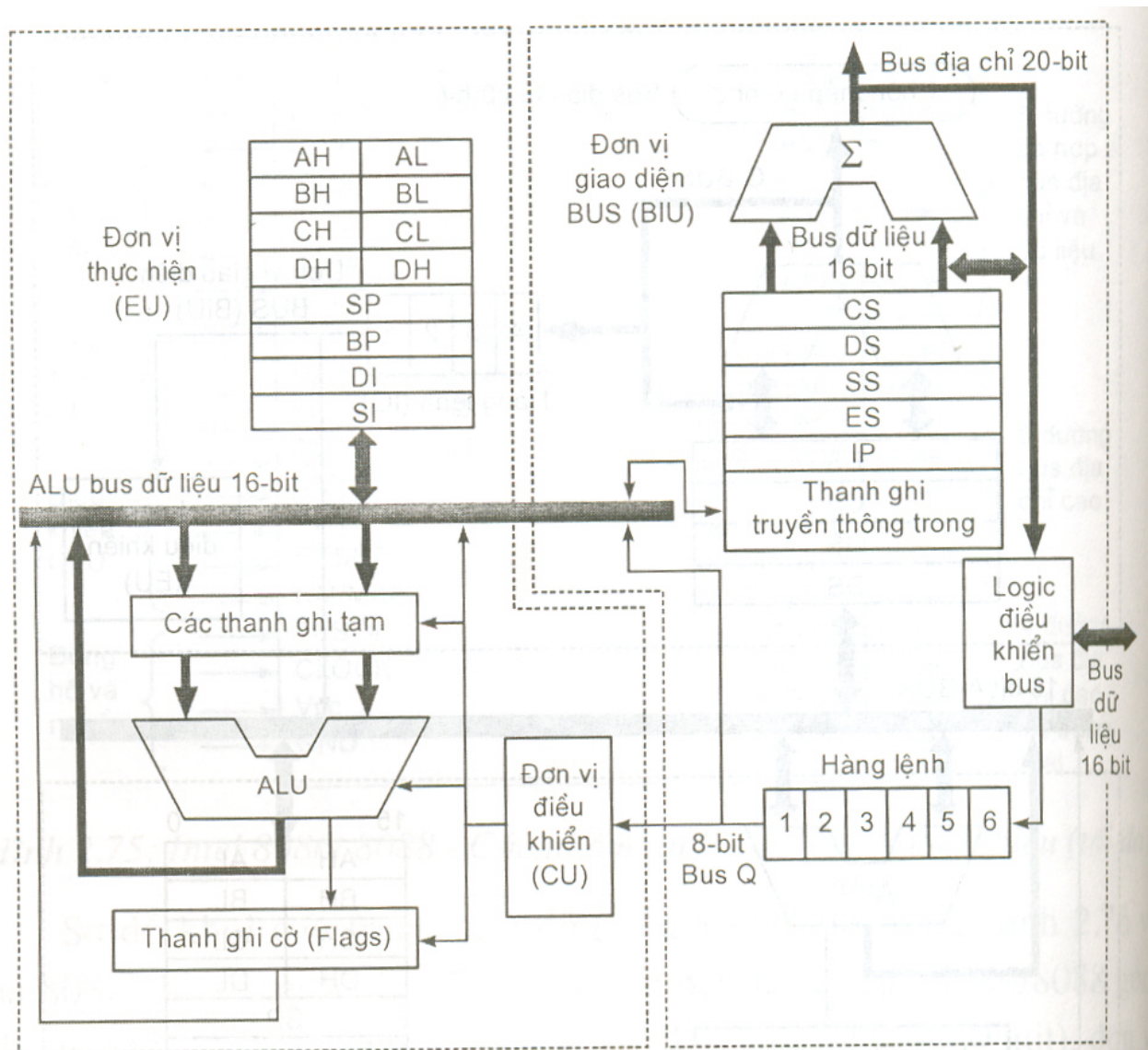
A. Kiến trúc bên trong của 8086/8088

1. Sơ đồ khối
2. Các đơn vị chức năng của 8088/8086
3. Các thanh ghi của 8086/8088
4. Phân đoạn bộ nhớ trong 8086/8088

B. Tập lệnh của 8088/8086

5. Khái niệm về lệnh và cách mã hoá lệnh
6. Các chế độ địa chỉ của vi xử lý 8086/8088
7. Phân loại tập lệnh của vi xử lý
8. Mô tả tập lệnh của 8086/8088

1. Sơ đồ khối vi xử lý 8086/8088



2. Các đơn vị chức năng của 8088/8086

❖ Đơn vị giao tiếp bus BIU (Bus Interface Unit)

- Điều khiển bus hệ thống: đưa địa chỉ ra bus và trao đổi dữ liệu với bus
 - Đưa ra địa chỉ
 - Đọc mã lệnh từ bộ nhớ
 - Đọc/ghi dữ liệu từ/vào bộ nhớ hoặc cổng vào/ra
- Các khối:
 - Bộ cộng để tính địa chỉ
 - 4 thanh ghi đoạn 16-bit: CS, DS, SS, ES
 - Bộ đếm chương trình/con trỏ lệnh 16-bit (PC/IP)
 - Hàng đợi lệnh IQ (4 bytes trong 8088 và 6 bytes trong 8086)
 - Logic điều khiển bus

2. Các đơn vị chức năng của 8088/8086

❖ Đơn vị thực hiện EU (Execution Unit)

- Chức năng: EU nhận lệnh & dữ liệu từ BIU để xử lý. Kết quả xử lý lệnh được chuyển ra bộ nhớ hoặc thiết bị I/O thông qua BIU.
- Các khối:
 - ALU
 - CU
 - 8 thanh ghi 16-bit: AX, BX, CX, DX, SP, BP, SI, DI
 - Thanh ghi cờ FR

❖ Bus trong (Internal Bus): liên kết BIU và EU

- 16-bit A-BUS trong 8088
- 16-bit ALU-BUS trong 8086

3. Các thanh ghi của 8086/8088

❖ Các thanh ghi đa năng:

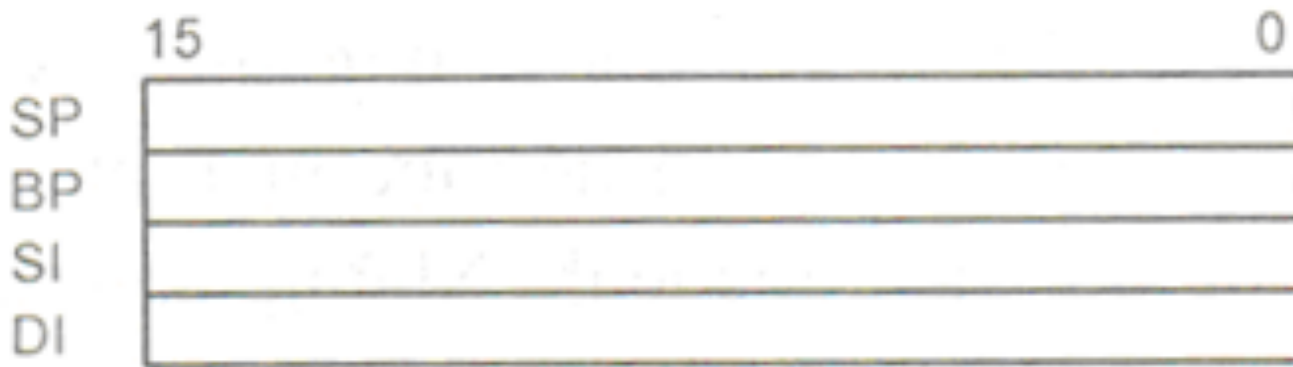
- 4 thanh ghi 16 bits:
 - AX: Thanh ghi tổng, thường dùng để lưu kết quả
 - BX: Thanh ghi cơ sở, thường dùng chứa địa chỉ ô nhớ
 - CX: Thanh ghi đếm, thường dùng làm con đếm cho các lệnh lặp
 - DX: Thanh ghi dữ liệu
- Hoặc 8 thanh ghi 8 bits: AH, AL, BH, BL, CH, CL, DH, DL

	15	7	0
AX	AH		AL
BX	BH		BL
CX	CH		CL
DX	DH		DL

3. Các thanh ghi của 8086/8088

❖ Các thanh ghi con trỏ và chỉ số:

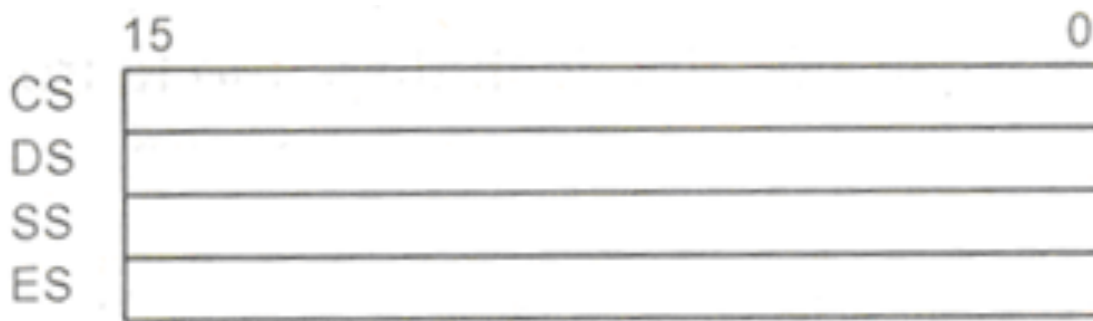
- SP (Stack Pointer): con trỏ ngăn xếp. SP luôn chứa địa chỉ đỉnh ngăn xếp
- BP (Base Pointer): Con trỏ cơ sở - sử dụng với đoạn ngăn xếp
- SI (Source Index): Thanh ghi chỉ số nguồn. SI thường dùng chứa địa chỉ ô nhớ nguồn trong các thao tác chuyển dữ liệu
- DI (Destination Index): Thanh ghi chỉ số đích. DI thường dùng chứa địa chỉ ô nhớ đích trong các thao tác chuyển dữ liệu



3. Các thanh ghi của 8086/8088

❖ Các thanh ghi đoạn:

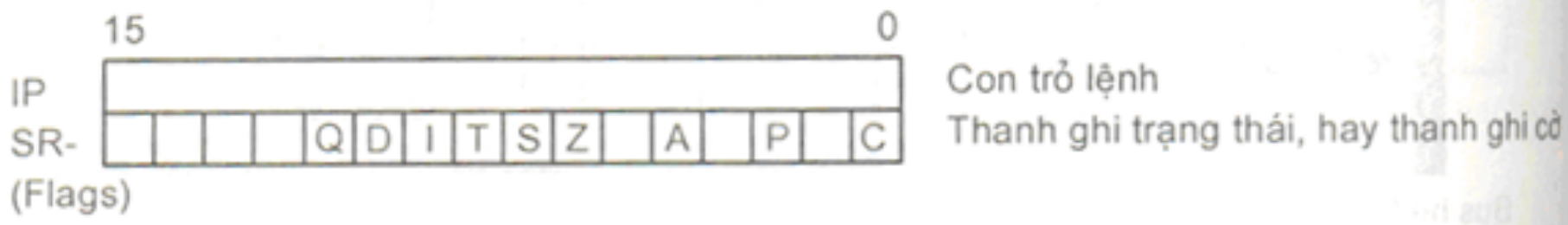
- CS (Code Segment): Thanh ghi đoạn mã. CS chứa địa chỉ bắt đầu đoạn mã
- DS (Data Segment): Thanh ghi đoạn dữ liệu. DS chứa địa chỉ bắt đầu đoạn dữ liệu
- SS (Stack Segment): Thanh ghi đoạn ngăn xếp. SS chứa địa chỉ bắt đầu đoạn ngăn xếp
- ES (Extra Segment): Thanh ghi đoạn dữ liệu mở rộng. ES chứa địa chỉ bắt đầu đoạn dữ liệu mở rộng.



3. Các thanh ghi của 8086/8088

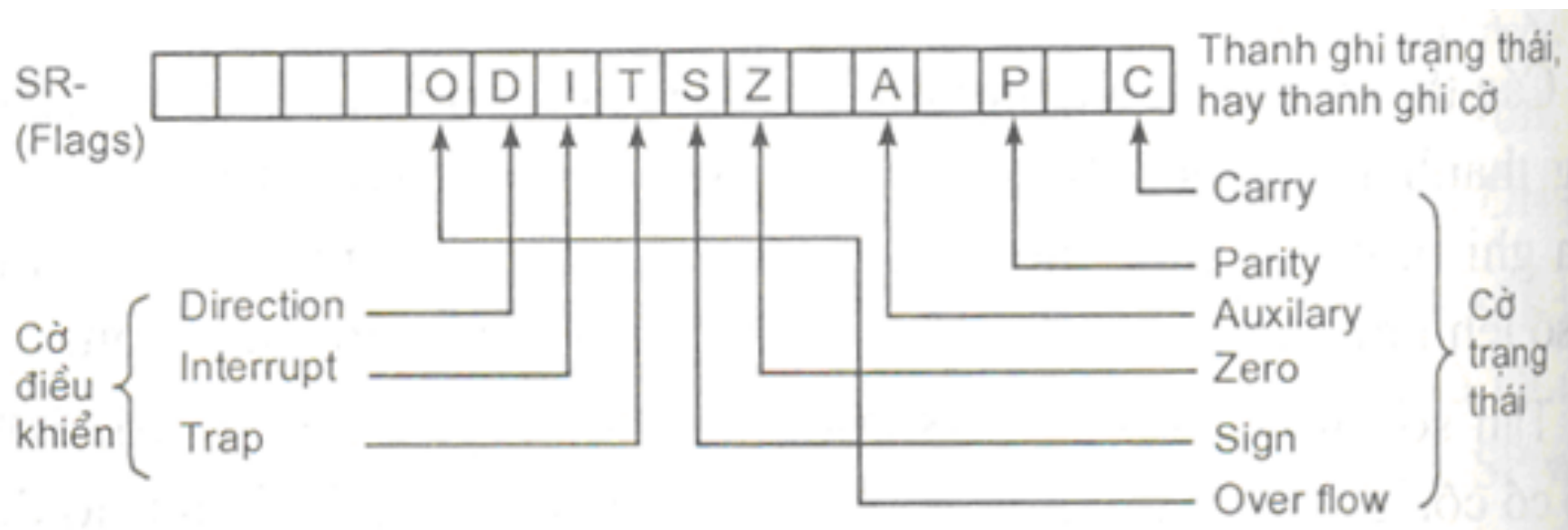
❖ Con trở lệnh và thanh ghi cờ:

- IP (Instruction Pointer): Con trỏ lệnh (còn gọi là bộ đếm chương trình PC). IP luôn chứa địa chỉ của lệnh tiếp theo sẽ được thực hiện;
- FR (Flag Register) hoặc SR (Status Register): Thanh ghi cờ hoặc thanh ghi trạng thái.
 - Cờ trạng thái: Các bit của FR lưu các trạng thái của kết quả phép toán ALU thực hiện
 - Cờ điều khiển: trạng thái của tín hiệu điều khiển.



3. Các thanh ghi của 8086/8088

❖ Các bit của thanh ghi cờ:



3. Các thanh ghi của 8086/8088

❖ Các cờ trạng thái:

- C (Carry): cờ nhớ. $C=1 \rightarrow$ có nhớ; $C=0 \rightarrow$ không nhớ
- A (Auxiliary): cờ nhớ phụ. $A=1 \rightarrow$ có nhớ phụ; $A=0 \rightarrow$ không nhớ phụ
- P (Parity): cờ chẵn lẻ. $P=1$ khi tổng số bit 1 trong kết quả là lẻ, $P=0$ khi tổng số bit 1 trong kết quả là chẵn
- O (Overflow): cờ tràn. $O=1$ khi kết quả bị tràn
- Z (Zero): cờ zero. $Z=1$ khi kết quả bằng 0; ngược lại $Z=0$
- S (Sign): cờ dấu. $S=1$ khi kết quả âm; $S=0$ khi kết quả không âm

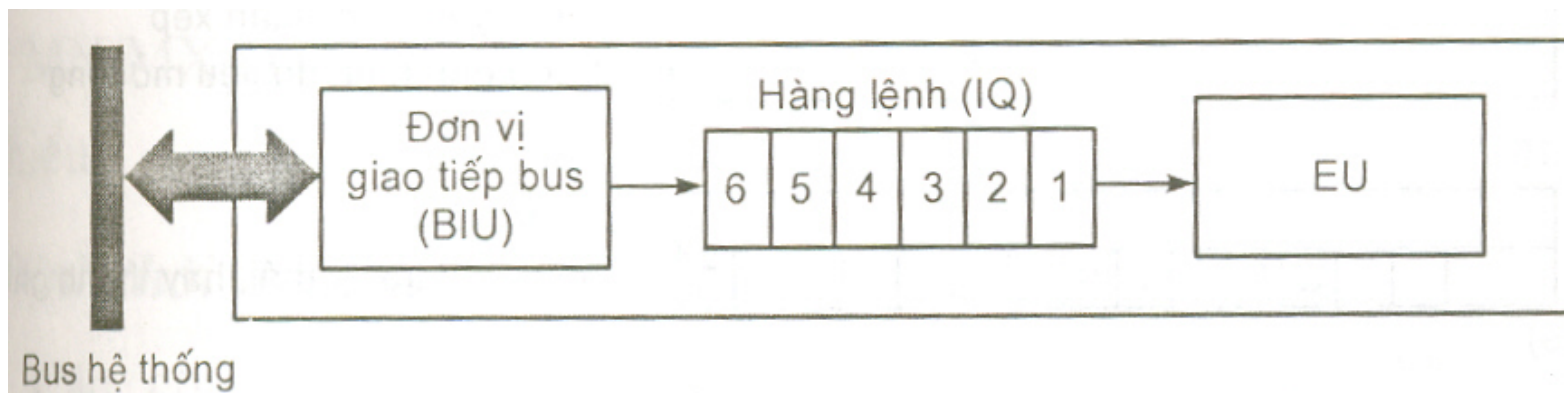
❖ Các cờ điều khiển:

- D (Direction): cờ hướng, chỉ hướng tăng giảm địa chỉ với các lệnh chuyển dữ liệu. $D=0 \rightarrow$ địa chỉ tăng. $D=1 \rightarrow$ địa chỉ giảm.
- T (Trap/Trace): cờ bẫy/lần vết, được dùng khi gỡ rối chương trình. $T=1 \rightarrow$ CPU ở chế độ chạy từng lệnh
- I (Interrupt): cờ ngắt. $I=1 \rightarrow$ cho phép ngắt; $I=0 \rightarrow$ cấm ngắt

Hàng đợi lệnh IQ

❖ Hàng đợi lệnh IQ (Instruction Queue):

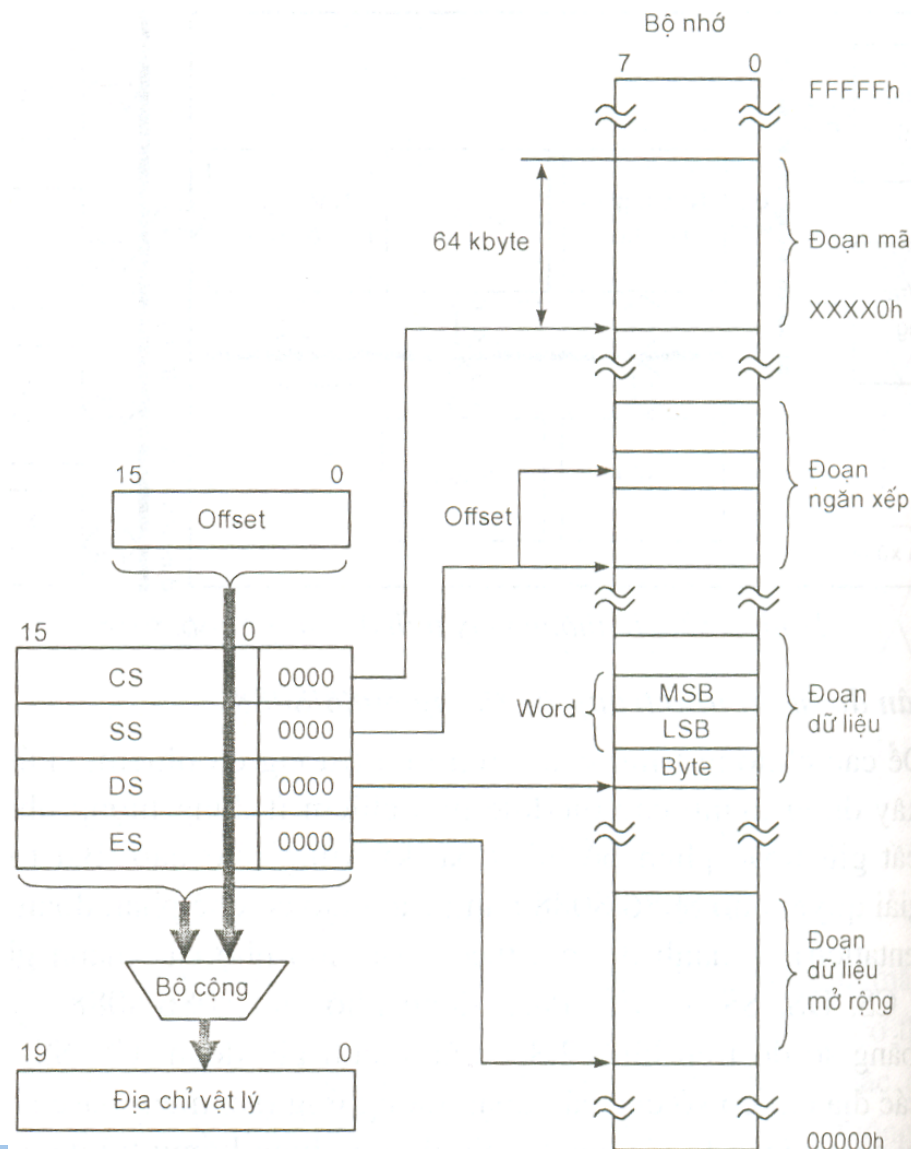
- Chứa lệnh đọc từ bộ nhớ cho EU thực hiện.
- Trong 8088, IQ có 4 bytes, còn trong 8086, IQ có 6 bytes.
- IQ là một thành phần quan trọng của cơ chế ống lệnh giúp tăng tốc độ xử lý lệnh.



4. Phân đoạn bộ nhớ trong 8086/8088

❖ VXL 8088/8086 sử dụng 20 bit để địa chỉ hoá bộ nhớ:

- Tổng dung lượng tối đa có thể địa chỉ hoá của bộ nhớ là $2^{20} = 1\text{MB}$;
- Địa chỉ được đánh từ 00000h đến FFFFFh.



4. Phân đoạn bộ nhớ trong 8086/8088

- ❖ Bộ nhớ được chia thành các đoạn (segment):
 - Các thanh ghi đoạn (CS, DS, SS, ES) trỏ đến địa chỉ bắt đầu của các đoạn
 - Vị trí của ô nhớ trong đoạn được xác định bằng địa chỉ lệch Offset: 0000h-FFFFh
 - Địa chỉ logic đầy đủ của một ô nhớ là Segment:Offset
- ❖ Địa chỉ vật lý 20-bit của một ô nhớ được xác định bằng phép cộng giữa địa chỉ đoạn 16-bit được dịch trái 4 bit (nhân với 16) và địa chỉ lệch 16-bit.
 - VD: CS:IP chỉ ra địa chỉ lệnh sắp thực hiện trong đoạn mã. Nếu CS=F000h và IP=FFF0h thì:
 - $CS:IP \sim F000h \times 16 + FFF0h = F0000h + FFF0h = FFFF0h$

5. Khái niệm về lệnh và cách mã hoá lệnh

❖ Lệnh (instruction) là gì?

- Là một từ nhị phân
- Lệnh được lưu trữ trong bộ nhớ
- Lệnh được nạp vào CPU để thực hiện
- Mỗi lệnh có một nhiệm vụ cụ thể
- Các nhóm lệnh thông dụng: vận chuyển dữ liệu, điều khiển chương trình, tính toán, vv.

❖ Các pha (phase) chính thực hiện lệnh:

- Đọc lệnh (IF: Instruction Fetch)
- Giải mã lệnh (ID: Instruction Decode)
- Thực hiện lệnh (EX: Instruction Execution)

5. Khái niệm về lệnh và cách mã hoá lệnh

❖ Chu kỳ lệnh (instruction cycle)

- Là khoảng thời gian CPU thực hiện xong 1 lệnh
- Mỗi pha của lệnh gồm một số chu kỳ máy
- Mỗi chu kỳ máy gồm một số chu kỳ nhịp đồng hồ
- Một CK lệnh có thể gồm:
 - Chu kỳ đọc lệnh
 - Chu kỳ đọc bộ nhớ (dữ liệu)
 - Chu kỳ ghi bộ nhớ (dữ liệu)
 - Chu kỳ đọc I/O (dữ liệu)
 - Chu kỳ ghi I/O (dữ liệu)
 - Chu kỳ chấp nhận ngắt
 - Bus rỗi

5. Khái niệm về lệnh và cách mã hoá lệnh

❖ Dạng lệnh

- Dạng tổng quát của lệnh: 2 thành phần: mã lệnh và địa chỉ của các toán hạng
- Độ dài của từ lệnh: 8, 16, 24, 32 và 64 bit.
- Lệnh của 8086/8088 có thể có độ dài 1-6 byte

Opcode	Operands
--------	----------

Mã lệnh

Các toán hạng

Mã lệnh	Đích, Gốc
---------	-----------

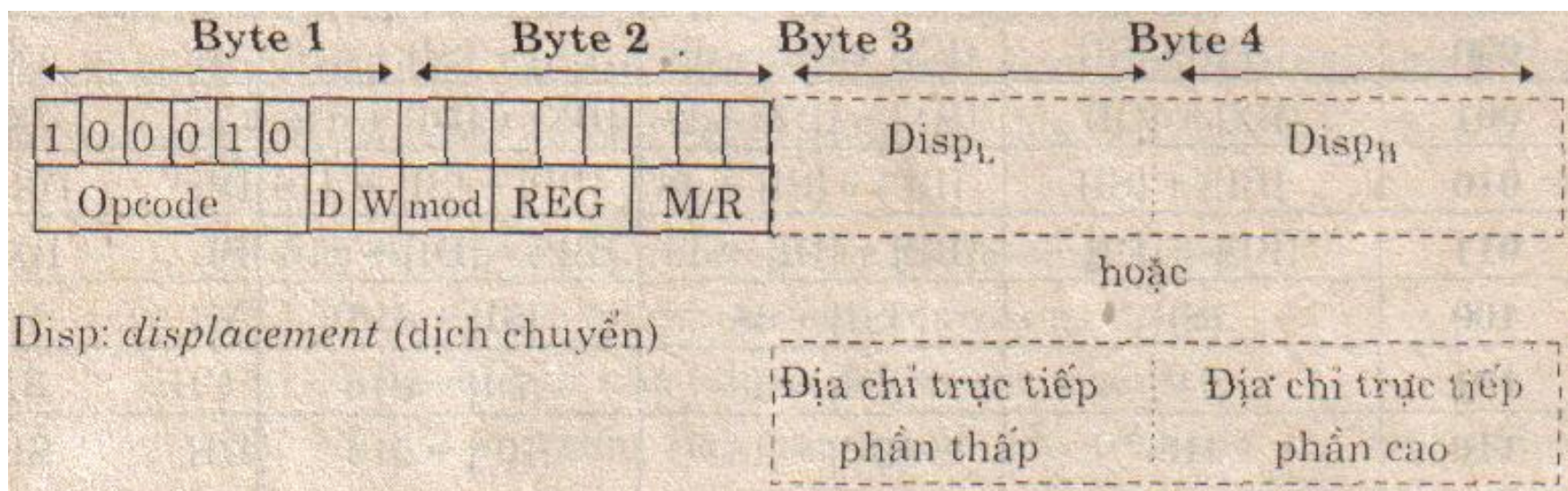
MOV	AX, 100
-----	---------

$AX \leftarrow 100$

5. Khái niệm về lệnh và cách mã hoá lệnh

❖ Mã hoá lệnh

- Opcode: mã lệnh gồm 6 bit; Mã lệnh của MOV là 100010
- D: bit hướng, chỉ hướng vận chuyển dữ liệu; D=1: dữ liệu đi đến thanh ghi cho bởi 3 bit REG; D=0: dữ liệu đi ra từ thanh ghi cho bởi 3 bit REG;
- W: bit chỉ độ rộng toán hạng; W=0: toán hạng 1 byte; W=1: toán hạng 2 bytes



5. Khái niệm về lệnh và cách mã hoá lệnh

❖ Mã hoá lệnh

- REG: 3 bit là mã của thanh ghi toán hạng theo hướng chuyển dữ liệu D:
 - Nếu D=1, REG biểu diễn toán hạng Đích
 - Nếu D=0, REG biểu diễn toán hạng Gốc

Các thanh ghi đoạn	Mã thanh ghi
CS	01
DS	11
ES	00
SS	10

Các thanh ghi		Mã thanh ghi
W=1	W=0	
AX	AL	000
BX	BL	011
CX	CL	001
DX	DL	010
SP	AH	100
DI	BH	111
BP	CH	101
SI	DH	110

5. Khái niệm về lệnh và cách mã hoá lệnh

❖ Mã hoá lệnh

- MOD (2 bit) và R/M (3 bit): MOD và R/M kết hợp với nhau để biểu diễn các chế độ địa chỉ của 8086/8088
- Disp_L: khoảng dịch chuyển phần thấp
- Disp_H: khoảng dịch chuyển phần cao.

MOD R/M	00	01	10	11	
				W=0	W=1
000	[BX]+[SI]	[BX]+[SI]+d8	[BX]+[SI]+d16	AL	AX
001	[BX]+[DI]	[BX]+[DI]+d8	[BX]+[DI]+d16	CL	CX
010	[BP]+[SI]	[BP]+[SI]+d8	[BP]+[SI]+d16	DL	DX
011	[BP]+[DI]	[BP]+[DI]+d8	[BP]+[DI]+d16	BL	BX
100	[SI]	[SI]+d8	[SI]+d16	AH	SP
101	[DI]	[DI]+d8	[DI]+d16	CH	BP
110	d16	[BP]+d8	[BP]+d16	DH	SI
111	[BX]	[BX]+d8	[BX]+d16	BH	DI

← Các chế độ bộ nhớ →

← Các chế độ thanh ghi →

Ghi chú:

- d8: khoảng dịch chuyển, 8 bit
- d16: khoảng dịch chuyển, 16 bit

6. Các chế độ địa chỉ của 8086/8088

- ❖ Chế độ địa chỉ (Addressing Mode) là cách CPU tổ chức các toán hạng của lệnh;
- ❖ Một bộ vi xử lý có thể có nhiều chế độ địa chỉ. Vi xử lý 8086/8088 có 7 chế độ địa chỉ:
 1. Chế độ địa chỉ thanh ghi (Register Addressing Mode)
 2. Chế độ địa chỉ tức thì (Immediate Addressing Mode)
 3. Chế độ địa chỉ trực tiếp (Direct Addressing Mode)
 4. Chế độ địa chỉ gián tiếp qua thanh ghi (Register Indirect Addressing Mode)
 5. Chế độ địa chỉ tương đối cơ sở (Based Plus Displacement Addressing Mode)
 6. Chế độ địa chỉ tương đối chỉ số (Indexed Plus Displacement Addressing Mode)
 7. Chế độ địa chỉ tương đối chỉ số cơ sở (Based Indexed Plus Displacement Addressing Mode)

6. Các chế độ địa chỉ của 8086/8088

❖ Chế độ địa chỉ thanh ghi:

- Sử dụng các thanh ghi bên trong cpu như là các toán hạng để chứa dữ liệu cần thao tác.
- Cả toán hạng gốc và đích đều là các thanh ghi
- VD:

`mov bx, dx; bx \leftarrow dx`

`mov ds, ax; ds \leftarrow ax`

`add al, dl; al \leftarrow al + dl`

6. Các chế độ địa chỉ của 8086/8088

❖ Chế độ địa chỉ tức thì:

- Toán hạng đích là một thanh ghi hay một ô nhớ
- Toán hạng gốc là một hằng số
- VD:

`mov cl, 200; cl \leftarrow 200`

`mov ax, 0ff0h; ax \leftarrow 0ff0h`

`mov ds, ax`

`mox [bx], 200; chuyển 200 vào ô nhớ có địa chỉ là DS:BX`

6. Các chế độ địa chỉ của 8086/8088

❖ Chế độ địa chỉ trực tiếp:

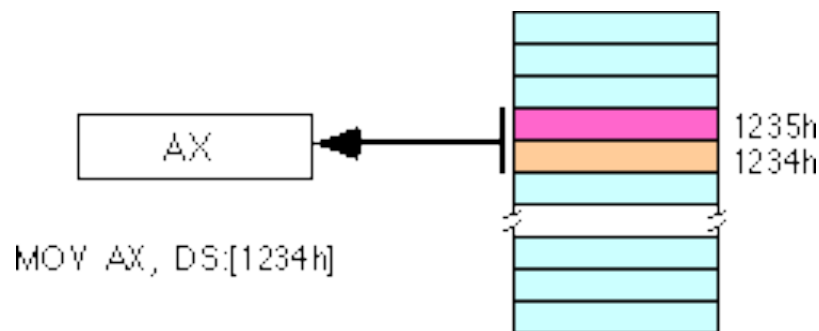
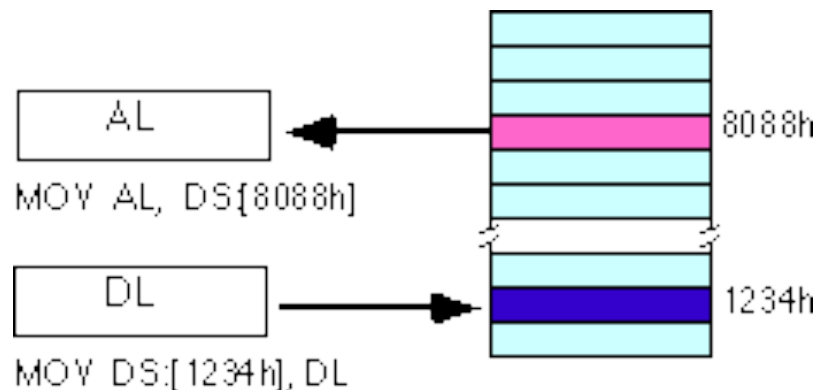
- Một toán hạng là một hằng biểu diễn địa chỉ lệch (offset) của ô nhớ
- Toán hạng còn lại có thể là thanh ghi (không được là ô nhớ)
- VD:

MOV AL, [8088H]

MOV [1234H], DL

MOV AX, [1234H]

DS là thanh ghi đoạn ngầm định trong chế độ địa chỉ trực tiếp.



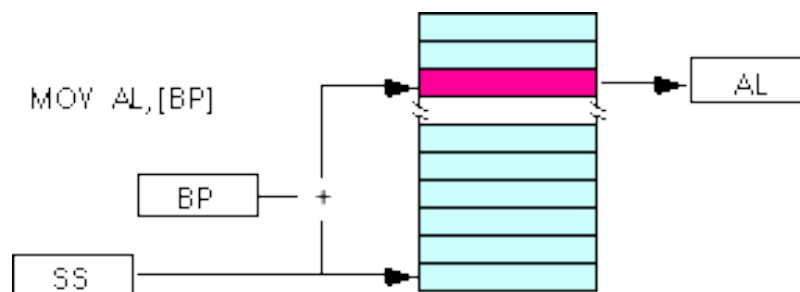
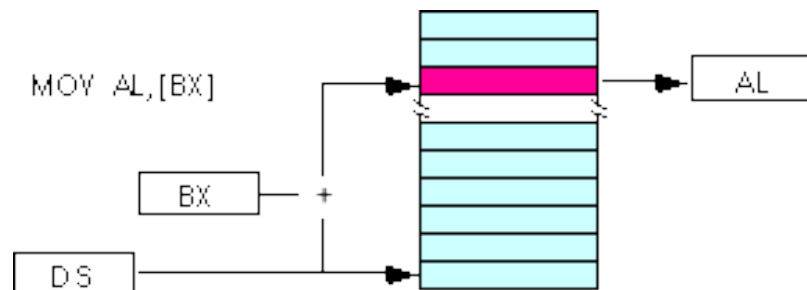
6. Các chế độ địa chỉ của 8086/8088

❖ Chế độ địa chỉ gián tiếp qua thanh ghi:

- Một toán hạng là một thanh ghi chứa địa chỉ lệch của ô nhớ
- Toán hạng còn lại có thể là thanh ghi hoặc hằng (không được là ô nhớ)
- VD:

`MOV AL, [BX]; AL ← [DS:BX]`

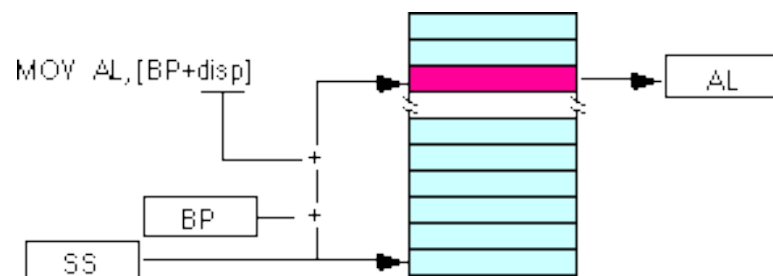
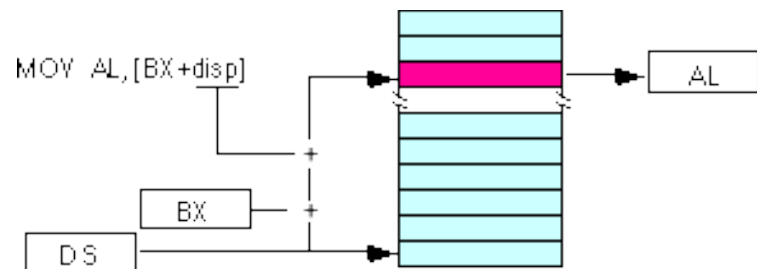
`MOV AL, [BP]; AL ← [SS:BP]`



6. Các chế độ địa chỉ của 8086/8088

❖ Chế độ địa chỉ tương đối cơ sở:

- Một toán hạng là đ/c của ô nhớ.
 - Đ/c của ô nhớ được tạo bởi việc sử dụng thanh ghi cơ sở như BX (đoạn DS) hoặc BP (đoạn SS) và một hằng số.
 - Hằng số trong địa chỉ tương đối cơ sở biểu diễn các giá trị dịch chuyển (displacement) được dùng để tính địa chỉ hiệu dụng của các toán hạng trong các vùng nhớ DS và SS.
- Toán hạng còn lại có thể là thanh ghi (không được là ô nhớ)
- VD:
`MOV AL, [BX+100]; AL ← [DS: BX+100]`
`MOV AL, [BP+200]; AL ← [SS: BP+200]`



6. Các chế độ địa chỉ của 8086/8088

❖ Chế độ địa chỉ tương đối chỉ số:

- Một toán hạng là đ/c của ô nhớ.
 - Đ/c của ô nhớ được tạo bởi việc sử dụng thanh ghi cơ sở SI hoặc DI và một hằng số.
 - Hằng số trong địa chỉ tương đối cơ sở biểu diễn các giá trị dịch chuyển (displacement) được dùng để tính địa chỉ hiệu dụng của các toán hạng trong các vùng nhớ DS.
- Toán hạng còn lại có thể là thanh ghi (ko được là ô nhớ)
- VD:
MOV AL, [SI+100]; AL \leftarrow [DS: SI+100]
MOV AL, [DI+200]; AL \leftarrow [DS: DI+200]

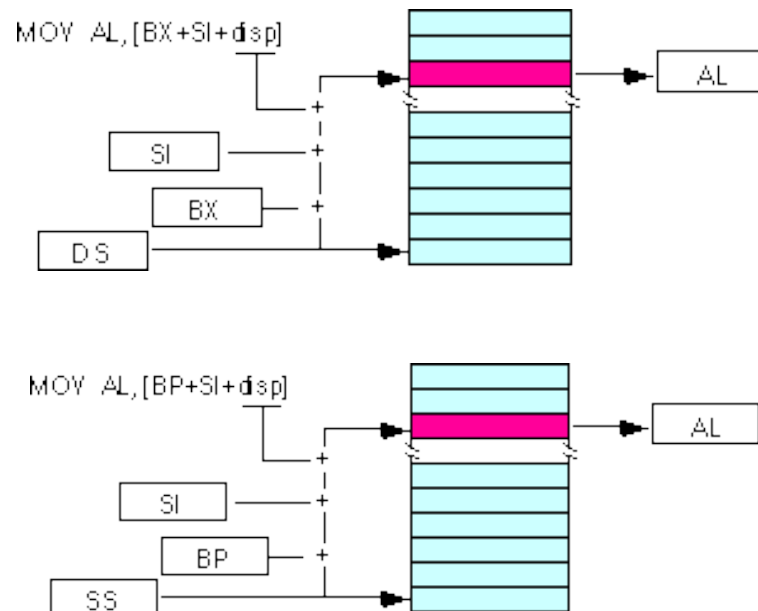
6. Các chế độ địa chỉ của 8086/8088

❖ Chế độ địa chỉ tương đối chỉ số cơ sở:

- Một toán hạng là đ/c của ô nhớ.
 - Đ/c của ô nhớ được tạo bởi việc sử dụng các thanh ghi BX+SI/DI (đoạn DS) hoặc BP+SI/DI (đoạn SS) và một hằng số.
 - Hằng số trong địa chỉ tương đối cơ sở biểu diễn các giá trị dịch chuyển (displacement) được dùng để tính địa chỉ hiệu dụng của các toán hạng trong các vùng nhớ DS và SS.
- Toán hạng còn lại có thể là thanh ghi (không được là ô nhớ)

❖ VD:

```
MOV AL, [BX+SI+100]; AL ← [DS:BX+SI+100]  
MOV AL, [BP+DI+200]; AL ← [SS:BP+DI+200]
```



Ánh xạ ngầm định trong các chế độ địa chỉ

Chế độ địa chỉ	Toán hạng	Đoạn ngầm định
Thanh ghi	Reg	
Tức thì	Data	
Trực tiếp	[offset]	DS
Gián tiếp qua thanh ghi	[BX]	DS
	[BP]	SS
	[SI]	DS
	[DI]	DS
Tương đối cơ sở	[BX] + Disp	DS
	[BP] + Disp	SS
Tương đối chỉ số	[SI] + Disp	DS
	[DI] + Disp	DS
Tương đối chỉ số cơ sở	[BX] + [SI] + Disp	DS
	[BX] + [DI] + Disp	DS
	[BP] + [SI] + Disp	SS
	[BP] + [DI] + Disp	SS

Ánh xạ ngầm định giữa thanh ghi đoạn và lịch

- ❖ Quan hệ ngầm định giữa các thanh ghi đoạn và các thanh ghi lịch:

Thanh ghi đoạn	CS	DS	ES	SS
Thanh ghi lịch	IP	SI, DI, BX	DI	SP, BP

- ❖ Địa chỉ ngầm định:

MOV AL, [BX]; $AL \leftarrow [DS:BX]$

MOV [SI+300], AH; $[DS:SI+300] \leftarrow AH$

- ❖ Địa chỉ tường minh (đầy đủ):

MOV AL, ES:[BX]; $AL \leftarrow [ES:BX]$

MOV SS:[SI+300], AH; $[SS:SI+300] \leftarrow AH$

7. Phân loại tập lệnh của vi xử lý

- ❖ Tập lệnh phức hợp (CISC) và tập lệnh giảm thiểu (RISC)
 - CISC (Complex Instruction Set Computers)
 - Hỗ trợ tập lệnh phong phú -> giảm lượng mã chương trình
 - Tập lệnh lớn -> khó tối ưu hoá cho chương trình dịch
 - Các lệnh có độ dài và thời gian thực hiện khác nhau -> giảm hiệu năng của cơ chế ống lệnh (pipeline)
 - RISC (Reduced Instruction Set Computers)
 - Tập lệnh tối thiểu: số lượng lệnh, các chế độ đ/c khuôn dạng lệnh và thời gian thực hiện
 - Tăng được hiệu năng của cơ chế ống lệnh (pipeline)
 - Dễ tối ưu hoá trong chương trình dịch
 - Chương trình thường dài, cần nhiều bộ nhớ và tăng thời gian truy cập bộ nhớ

7. Phân loại tập lệnh của vi xử lý

- ❖ Phân loại tập lệnh của vi xử lý họ CISC
 - Vận chuyển DL
 - Số học nguyên và logic
 - Dịch và quay
 - Chuyển điều khiển
 - Xử lý bit
 - Điều khiển hệ thống
 - Thao tác dấu phẩy động
 - Các lệnh của các đơn vị chức năng đặc biệt

8. Mô tả tập lệnh của 8086/8088

- ❖ Các lệnh vận chuyển dữ liệu: vận chuyển dữ liệu giữa:
 - thanh ghi – thanh ghi;
 - thanh ghi–ô nhớ;
 - thanh ghi – thiết bị vào ra.
- ❖ Các lệnh:
 - MOV
 - LODSB, LODSW, STOSB, STOSW
 - MOVSB, MOVSW
 - IN, OUT
- ❖ Các lệnh vận chuyển dữ liệu không ảnh hưởng đến các cờ trạng thái của thanh ghi cờ

8. Mô tả tập lệnh của 8086/8088

❖ Lệnh MOV:

- Dạng lệnh: MOV Đích, Gốc; Đích \leftarrow Gốc
- Ý nghĩa: chuyển (sao chép) dữ liệu từ Gốc sang Đích
- Lưu ý: hai toán hạng Đích và Gốc phải tương thích về kích cỡ
- Ví dụ: MOV AL, 100; AL \leftarrow 100
 MOV [BX], AH; [DS:BX] \leftarrow AH
 MOV DS, AX; DS \leftarrow AX

8. Mô tả tập lệnh của 8086/8088

8086/8088 Instruction MOV

ASL Code	mov destination , source		
Machine Encoding	Cases		Machine Code Structure
	Copy Memory or Register to/from Register		100010 <u>d</u> <u>w</u> <u>mod</u> <u>reg</u> <u>r/m</u> [offset(Lower 8 bits) [offset(Upper 8 bits)]]
	Copy constant (16 bit or 8 bit) to Memory/Register		1100011 <u>w</u> <u>mod</u> 000 <u>r/m</u> [data(Lower 8 bits) [data(Upper 8 bits)]]
	Copy constant (16 bit or 8 bit) to Register		1011 <u>w</u> <u>reg</u> [data(Lower 8 bits) [data(Upper 8 bits)]]
	Copy (16 bit or 8 bit) Memory content to Accumulator		1010000 <u>w</u> [address(Lower 8 bits) [address(Upper 8 bits)]]
	Copy Accumulator (16 bit or 8 bit) to Memory		1010001 <u>w</u> [address(Lower 8 bits) [address(Upper 8 bits)]]
	Copy Memory or Register to Segment register		10001110 <u>mod</u> 0 <u>sr</u> <u>r/m</u> [offset(Lower 8 bits) offset(Upper 8 bits)]
	Copy Segment Register to Memory or Register		10001100 <u>mod</u> 0 <u>sr</u> <u>r/m</u> [offset(Lower 8 bits) offset(Upper 8 bits)]
Flags	Not Affected		
Description	MOV copies 16 bit (or 8 bit) content of SOURCE to DESTINATION of equal size.		
Examples	ASL	Machine Code	Comment
	mov bx,ax	1000 1001 1101 1000 (89D8h)	; AX -> BX
	mov ax,sp	1000 1001 1110 0000 (89E0h)	; SP -> AX (copying pointer to pinter is prohibited)
	mov al,cl	1000 1000 1100 1000 (88C8h)	; CL -> AL
	mov cx,1000h	1011 1000 0000 0000 0001 0000 (b80010h)	; 1000h -> CX
	mov [bx],ax	1000 1001 0000 0111 (8907h)	; AX -> Memory[BX]
	mov dx,[bp+2]	1000 1011 0101 0110 0000 0010 (8B5602h)	; Memory[BP+2] -> DX
	mov [b800h],ax	1010 0011 0000 0000 1011 1000 (a300b8h)	; AX -> [B800h]
	mov ss,[b800h]	1000 1110 0001 0110 1011 1000 (8E1600B8h)	; [B800h] -> SS

8. Mô tả tập lệnh của 8086/8088

❖ Lệnh LODSB, LODSW:

- **Dạng lệnh:**

LODSB;	AL \leftarrow [DS: SI]
	SI \leftarrow SI \pm 1
LODSW;	AX \leftarrow [DS: SI]
	SI \leftarrow SI \pm 2
- **Ý nghĩa:** Nạp nội dung ô nhớ có địa chỉ chứa trong SI thuộc đoạn DS vào thanh ghi AL/AX và tăng hoặc giảm nội dung của SI. Nếu DF = 0 \rightarrow tăng, DF = 1 \rightarrow giảm.
- **Ví dụ:**

MOV SI, 1000;	SI \leftarrow 1000
MOV [DS:SI], 200;	[DS:SI] \leftarrow 200
CLD;	DF \leftarrow 0
LODSB;	AL \leftarrow 200; SI \leftarrow SI + 1

8. Mô tả tập lệnh của 8086/8088

❖ Lệnh STOSB, STOSW:

- **Dạng lệnh:**

STOSB;	[ES: DI] ← AL
	DI ← DI ± 1
STOSW;	[ES: DI] ← AX
	DI ← DI ± 2
- **Ý nghĩa:** Lưu nội dung thanh ghi AL/AX vào ô nhớ có địa chỉ chứa trong DI thuộc đoạn ES và tăng hoặc giảm nội dung của DI. Nếu DF = 0 → tăng, DF = 1 → giảm.
- **Ví dụ:**

MOV DI, 1000;	DI ← 1000
MOV AL, 200;	AL ← 200
CLD;	DF ← 0
STOSB;	[ES:DI] ← AL ; DI ← DI + 1

8. Mô tả tập lệnh của 8086/8088

❖ Lệnh MOVSB, MOVSW:

- **Dạng lệnh:**

MOVS _B ;	$[ES:DI] \leftarrow [DS:SI]$
	$SI \leftarrow SI \pm 1; \quad DI \leftarrow DI \pm 1$
MOVS _W ;	$[ES:DI] \leftarrow [DS:SI]$
	$SI \leftarrow SI \pm 2; \quad DI \leftarrow DI \pm 2$
- **Ý nghĩa:** Chuyển nội dung ô nhớ tại địa chỉ DS:SI vào ô nhớ có địa chỉ ES:DI và tăng hoặc giảm nội dung của SI và DI. Nếu $DF = 0 \rightarrow$ tăng, $DF = 1 \rightarrow$ giảm.
- **Ví dụ:**

MOV SI, 1000;	$SI \leftarrow 1000$
MOV DI, 2000;	$DI \leftarrow 2000$
CLD;	$DF \leftarrow 0$
MOVS _B ;	$[ES:DI] \leftarrow [DS:SI]$

8. Tập lệnh - Các lệnh vận chuyển dữ liệu

❖ Lệnh IN:

- Dạng lệnh: IN <thanh ghi>, <địa chỉ cổng vào>
- Ý nghĩa: đọc dữ liệu từ <địa chỉ cổng vào> lưu vào <thanh ghi>. Có thể dùng giá trị số trực tiếp trong lệnh nếu <địa chỉ cổng vào> nằm trong khoảng 00-FFh; Nếu <địa chỉ cổng vào> lớn hơn FFh, địa chỉ cổng cần được lưu vào thanh ghi DX.
- Ví dụ: IN AL, 0F8H; AL \leftarrow (0F8h)
 MOV DX, 02F8H
 IN AL, DX; AL \leftarrow (DX)

8. Tập lệnh - Các lệnh vận chuyển dữ liệu

❖ Lệnh OUT:

- Dạng lệnh: OUT <địa chỉ cổng ra>, <Gốc>
- Ý nghĩa: Lưu dữ liệu từ Gốc ra <địa chỉ cổng ra>. Có thể dùng giá trị số trực tiếp trong lệnh nếu <địa chỉ cổng ra> nằm trong khoảng 00-FFh; Nếu <địa chỉ cổng ra> lớn hơn FFh, địa chỉ cổng cần được lưu vào thanh ghi DX.
- Ví dụ: OUT 0F8H, AL; (0F8h) \leftarrow AL
 MOV DX, 02F8H
 OUT DX, AL; (DX) \leftarrow AL

8. Tập lệnh - Các lệnh số học

- ❖ Là các lệnh thực hiện các phép toán số học: cộng (ADD), trừ (SUB), nhân (MUL) và chia (DIV);
- ❖ Lệnh ADD – cộng các số nguyên:
 - Dạng lệnh: $\text{ADD } \langle \text{Đích} \rangle, \langle \text{Gốc} \rangle; \text{Đích} \leftarrow \text{Đích} + \text{Gốc}$
 - Ý nghĩa: Lấy Gốc cộng với Đích, kết quả lưu vào Đích
 - Lệnh ADD ảnh hưởng đến các cờ: C, Z, S, P, O, A
 - Ví dụ:
 $\text{ADD AX, BX; } \text{AX} \leftarrow \text{AX} + \text{BX}$
 $\text{ADD AL, 10; } \text{AL} \leftarrow \text{AL} + 10$
 $\text{ADD [BX], AL; [DS:BX]} \leftarrow [\text{DS:BX}] + \text{AL}$

8. Tập lệnh - Các lệnh số học

❖ Lệnh SUB – trừ các số nguyên:

- Dạng lệnh: SUB <Đích>, <Gốc>; Đích \leftarrow Đích - Gốc
- Ý nghĩa: Lấy Đích trừ Gốc, kết quả lưu vào Đích
- Lệnh SUB ảnh hưởng đến các cờ: C, Z, S, P, O, A
- Ví dụ:
 SUB AX, BX; AX \leftarrow AX - BX
 SUB AL, 10; AL \leftarrow AL - 10
 SUB [BX], AL; [DS:BX] \leftarrow [DS:BX] - AL

8. Tập lệnh - Các lệnh số học

❖ Lệnh MUL – nhân các số nguyên:

- Dạng lệnh: MUL <Gốc>;
- Gốc phải là một thanh ghi hoặc địa chỉ ô nhớ
- Ý nghĩa:
 - Nếu Gốc là 8 bit: $AX \leftarrow AL * \text{Gốc}$
 - Nếu Gốc là 16 bit: $DXAX \leftarrow AX * \text{Gốc}$
- Lệnh MUL ảnh hưởng đến các cờ: Z, S, P
- Ví dụ: tính $10 * 30$

MOV AL, 10;	$AL \leftarrow 10$
MOV BL, 30;	$BL \leftarrow 30$
MUL BL;	$AX \leftarrow AL * BL$

8. Tập lệnh - Các lệnh số học

❖ Lệnh DIV – chia các số nguyên:

- Dạng lệnh: DIV <Gốc>;
- Gốc phải là một thanh ghi hoặc địa chỉ ô nhớ
- Ý nghĩa:
 - Nếu Gốc là 8 bit: AX : Gốc; AL chứa thương và AH chứa phần dư
 - Nếu Gốc là 16 bit: DXAX : Gốc; AX chứa thương và DX chứa phần dư
- Lệnh DIV ảnh hưởng đến các cờ: Z, S, P
- Ví dụ: tính $100 : 30$
 MOV AX, 100; AL \leftarrow 100
 MOV BL, 30; BL \leftarrow 30
 DIV BL; AX : BL; AL = 3, AH = 10

8. Tập lệnh - Các lệnh logic

❖ Các lệnh logic: NOT (phủ định), AND (và), OR (hoặc) và XOR (hoặc loại trừ). Bảng giá trị của các phép toán logic:

X	Y	NOT	AND	OR	XOR
0	0	1	0	0	0
0	1	1	0	1	1
1	0	0	0	1	1
1	1	0	1	1	0

8. Tập lệnh - Các lệnh logic

❖ Lệnh NOT

- Dạng: NOT <Đích>
- Ý nghĩa: Đảo các bit của toán hạng Đích
- Lệnh NOT ảnh hưởng đến các cờ: Z, S, P
- VD:

MOV AL, 80H;

80H = 1000 0000B

NOT AL;

7FH = 0111 1111B

8. Tập lệnh - Các lệnh logic

❖ Lệnh AND

- Dạng: AND <Đích>, <Gốc>
- Ý nghĩa: Nhân các cặp bit của 2 toán hạng Đích, Gốc, kết quả chuyển vào Đích
- Lệnh AND ảnh hưởng đến các cờ: Z, S, P
- VD: AND có thể được dùng để xoá một hoặc một số bit

Xoá bit thứ 3 của thanh ghi AL (0-7)

AND AL, F7H; F7H = 1111 0111B

Xoá 4 bit phần cao của thanh ghi AL (0-7)

AND AL, 0FH; 0FH = 0000 1111B

8. Tập lệnh - Các lệnh logic

❖ Lệnh OR

- Dạng: OR <Đích>, <Gốc>
- Ý nghĩa: Cộng các cặp bit của 2 toán hạng Đích, Gốc, kết quả chuyển vào Đích
- Lệnh OR ảnh hưởng đến các cờ: Z, S, P
- VD: OR có thể được dùng để lập một hoặc một số bit lập bit thứ 3 của thanh ghi AL (0-7)

OR AL, 08H; 08H = 0000 1000B

lập bit thứ 7 của thanh ghi AL (0-7)

OR AL, 80H; 80H = 1000 0000B

8. Tập lệnh - Các lệnh logic

❖ Lệnh XOR

- Dạng: XOR <Đích>, <Gốc>
- Ý nghĩa: Cộng đảo các cặp bit của 2 toán hạng Đích, Gốc, kết quả chuyển vào Đích
- Lệnh XOR ảnh hưởng đến các cờ: Z, S, P
- VD: Dùng XOR để xoá nội dung của thanh ghi/ô nhớ xoá thanh ghi AL

XOR AL, AL; AL \leftarrow 0

xoá thanh ghi BX

XOR BX, BX; BX \leftarrow 0

8. Tập lệnh - Các lệnh dịch và quay

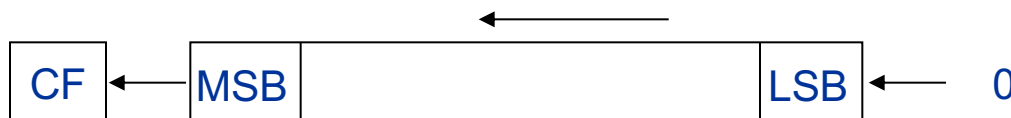
- ❖ Gồm các lệnh:
 - Dịch trái: SHL (Shift Left)
 - Dịch phải: SHR (Shift Right)
 - Quay trái: ROL (Rotate Left)
 - Quay phải: ROR (Rotate Right)
- ❖ Các lệnh dịch thường được dùng để thay cho phép nhân (dịch trái) và thay cho phép chia (dịch phải)
- ❖ Các lệnh dịch và quay còn có thể được sử dụng khi cần xử lý từng bit.

8. Tập lệnh - Các lệnh dịch và quay

❖ Lệnh dịch trái SHL

- Dạng: SHL <Đích>, 1
SHL <Đích>, CL
- Ý nghĩa: Dịch trái một bit hoặc dịch trái số bit lưu trong thanh ghi CL nếu số bit cần dịch lớn hơn 1.
 - MSB (Most Significant Bit) chuyển sang cờ nhớ CF
 - 0 được điền vào LSB (Least Significant Bit)
 - Các bit giữa MSB và LSB được dịch sang trái 1 bit
- VD:

MOV AL, 08H;	0000 1000B (8)
SHL AL, 1;	0001 0000B (16)
MOV CL, 2	
SHL AL, CL;	0100 0000B (64)

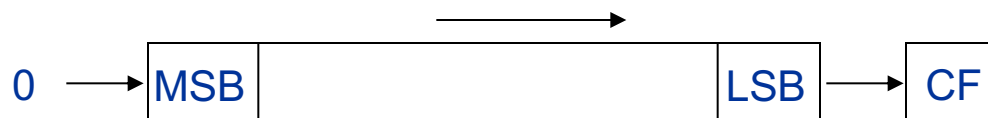


8. Tập lệnh - Các lệnh dịch và quay

❖ Lệnh dịch phải SHR

- Dạng: SHR <Đích>, 1
 SHR <Đích>, CL
- Ý nghĩa: Dịch phải một bit hoặc dịch phải số bit lưu trong thanh ghi CL nếu số bit cần dịch lớn hơn 1.
 - LSB (Least Significant Bit) chuyển sang cờ nhớ CF
 - 0 được điền vào MSB (Most Significant Bit)
 - Các bit giữa MSB và LSB được dịch sang phải 1 bit
- VD:

```
MOV AL, 80H;           1000 0000B (128)
SHR AL, 1;             0100 0000B (64)
MOV CL, 2
SHR AL, CL;            0001 0000B (16)
```

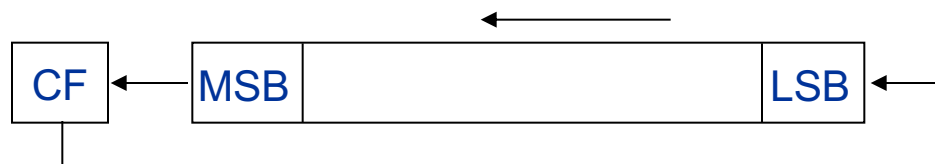


8. Tập lệnh - Các lệnh dịch và quay

❖ Lệnh quay trái ROL

- Dạng: ROL <Đích>, 1
ROL <Đích>, CL
- Ý nghĩa: Quay trái một bit hoặc quay trái số bit lưu trong thanh ghi CL nếu số bit cần quay lớn hơn 1.
 - MSB (Most Significant Bit) chuyển sang cờ nhớ CF
 - MSB được chuyển đến LSB (Least Significant Bit)
 - Các bit giữa MSB và LSB được dịch sang trái 1 bit
- VD:

```
MOV AL, 88H;      1000 1000B
ROL AL, 1;         0001 0001B
MOV CL, 2
ROL AL, CL;        0100 0100B
```

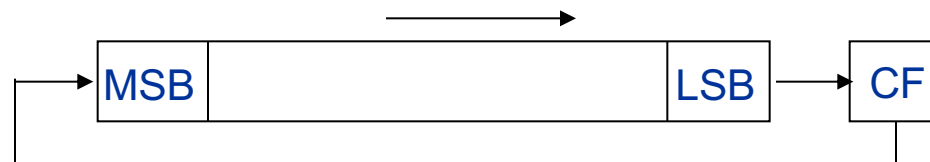


8. Tập lệnh - Các lệnh dịch và quay

❖ Lệnh quay phải ROR

- Dạng: ROR <Đích>, 1
ROR <Đích>, CL
- Ý nghĩa: Quay phải một bit hoặc quay phải số bit lưu trong thanh ghi CL nếu số bit cần quay lớn hơn 1.
 - LSB (Least Significant Bit) chuyển sang cờ nhớ CF
 - LSB được chuyển đến MSB (Most Significant Bit)
 - Các bit giữa MSB và LSB được dịch sang phải 1 bit
- VD:

```
MOV AL, 88H;      1000 1000B
ROR AL, 1;         0100 0100B
MOV CL, 2
ROR AL, CL;        0001 0001B
```



8. Tập lệnh - Các lệnh chuyển điều khiển

- ❖ Các lệnh chuyển điều khiển (program flow control instructions) là các lệnh làm thay đổi trật tự thực hiện chương trình;
- ❖ Gồm các lệnh:
 - Lệnh nhảy không điều kiện JMP
 - Lệnh nhảy có điều kiện JE, JZ, JNE, JNZ, JL, JLE, JG, JGE, ...
 - Lệnh lặp LOOP, LOOPE, LOOPZ
 - Lệnh gọi thực hiện chương trình con CALL
 - Lệnh trở về từ chương trình con RET

8. Tập lệnh - Các lệnh chuyển điều khiển

❖ Lệnh nhảy không điều kiện JMP

- Dạng lệnh: JMP <nhãn>
- Ý nghĩa: chuyển đến thực hiện lệnh nằm ngay sau <nhãn>
- <nhãn> là một tên được đặt trước một lệnh, phân cách bằng dấu hai chấm (:). Khoảng nhảy của JMP có thể là ngắn ($-128 \div +127$), gần ($-32768 \div +32767$) và xa (sử dụng địa chỉ đầy đủ CS:IP).
- VD:

START:

ADD AX, BX

SUB BX, 1

.....

JMP START ; chuyển đến thực hiện lệnh nằm sau nhãn START

8. Tập lệnh - Các lệnh chuyển điều khiển

❖ Lệnh nhảy có điều kiện JE, JZ, JNE, JNZ, JL, JG

- Dạng lệnh:

JE <nhãn> : nhảy nếu bằng nhau hoặc kết quả bằng 0

JZ <nhãn> : nhảy nếu bằng nhau hoặc kết quả bằng 0

JNE <nhãn> : nhảy nếu không bằng nhau hoặc kết quả khác 0

JNZ <nhãn> : nhảy nếu không bằng nhau hoặc kết quả khác 0

JL <nhãn> : nhảy nếu bé hơn

JLE <nhãn> : nhảy nếu bé hơn hoặc bằng

JG <nhãn> : nhảy nếu lớn hơn

JGE <nhãn> : nhảy nếu lớn hơn hoặc bằng

- Khoảng nhảy của các lệnh nhảy có điều kiện là ngắn ($-128 \div +127$).

8. Tập lệnh - Các lệnh chuyển điều khiển

❖ Lệnh nhảy có điều kiện JE, JZ, JNE, JNZ, JL, JG

- VD: viết đoạn chương trình tính tổng các số từ 1-20

MOV AX, 0 ; AX chứa tổng

MOV BX, 20 ; đặt giá trị cho biến đếm BX

START:

ADD AX, BX ; cộng dồn

SUB BX, 1 ; giảm biến đếm

JZ STOP ; dừng nếu BX = 0

JMP START ; quay lại vòng lặp tiếp

STOP:

8. Tập lệnh - Các lệnh chuyển điều khiển

❖ Lệnh lặp LOOP

- Dạng lệnh: LOOP <nhãn>
- Ý nghĩa: chuyển đến thực hiện lệnh nằm ngay sau <nhãn> nếu giá trị trong thanh ghi CX khác 0. Tự động giảm giá trị của CX 1 đơn vị khi thực hiện.

- VD: viết đoạn chương trình tính tổng các số từ 1-20

```
MOV AX, 0      ; AX chứa tổng
MOV CX, 20     ; đặt giá trị cho biến đếm CX
```

START:

```
ADD AX, CX     ; cộng dồn
LOOP START     ; kiểm tra CX, nếu CX=0 → dừng
               ; nếu CX khác 0: CX ← CX-1 và quay lại
               ; bắt đầu vòng lặp mới từ vị trí của START
```

8. Tập lệnh - Các lệnh chuyển điều khiển

❖ Lệnh CALL và RET

- Dạng lệnh:
 - CALL <tên chương trình con>: gọi thực hiện chương trình con
 - RET : trở về từ chương trình con; thường đặt ở cuối chương trình con

- VD:

CALL GIAITHUA ; gọi thực hiện chương trình con GIAITHUA

.....

; phần mã của chương trình con

PROC GIAITHUA ; bắt đầu mã CT con

.....

RET ; trở về chương trình gọi

GIAITHUA ENDP ; kết thúc mã CT con

8. Tập lệnh - Các lệnh xử lý bit

- ❖ Gồm nhóm các lệnh xử lý một số bit (D, C, I) của thanh ghi cờ FR;
- ❖ Các lệnh lập cờ (đặt bit cờ bằng 1)
 - STD: lập cờ hướng D
 - STC: lập cờ nhớ C
 - STI: lập cờ ngắt I
- ❖ Các lệnh xóa cờ (đặt bit cờ bằng 0)
 - CLD: xóa cờ hướng D
 - CLC: xóa cờ nhớ C
 - CLI: xóa cờ ngắt I

8. Tập lệnh - Các lệnh điều khiển hệ thống

❖ Gồm 2 lệnh:

- Lệnh NOP (No Operation):
 - NOP không thực hiện nhiệm vụ cụ thể, chỉ tiêu tốn thời gian bằng 1 chu kỳ lệnh
- Lệnh HLT (Halt)
 - HLT dừng việc thực hiện chương trình

8. Tập lệnh – Một số lệnh khác

❖ Lệnh tăng INC

- Dạng: INC <Đích> ; $\text{Đích} \leftarrow \text{Đích} + 1$

❖ Lệnh giảm DEC

- Dạng: DEC <Đích> ; $\text{Đích} \leftarrow \text{Đích} - 1$

❖ Lệnh so sánh CMP

- Dạng: CMP <Đích>, <Gốc>
- Ý nghĩa: Tính toán Đích - Gốc, kết quả chỉ dùng cập nhật các bit cờ trạng thái, không lưu vào Đích:

Trường hợp	C	Z	S
Đích > Gốc	0	0	0
Đích = Gốc	0	1	0
Đích < Gốc	1	0	1

8. Tập lệnh – Một số lệnh khác

❖ Lệnh PUSH – đẩy dữ liệu vào ngăn xếp

- Dạng: PUSH <Gốc>
- Ý nghĩa: Nạp Gốc vào đỉnh ngăn xếp; Gốc phải là toán hạng 2 bytes. Diễn giải:

$SP \leftarrow SP + 2$; tăng con trỏ ngăn xếp SP

$\{SP\} \leftarrow \text{Gốc}$; nạp dữ liệu vào ngăn xếp

- VD: PUSH AX

❖ Lệnh POP – lấy dữ liệu ra khỏi ngăn xếp

- Dạng: POP <Đích>
- Ý nghĩa: Lấy dữ liệu từ đỉnh ngăn xếp lưu vào Đích; Đích phải là toán hạng 2 bytes. Diễn giải:

$\text{Đích} \leftarrow \{SP\}$; lấy dữ liệu ra khỏi ngăn xếp

$SP \leftarrow SP - 2$; giảm con trỏ ngăn xếp SP

- VD: POP BX

8. Tập lệnh – Một số lệnh khác

❖ Lệnh NEG – đảo dấu giá trị của toán hạng

- Dạng: NEG <Đích>
- Ý nghĩa: Đảo dấu giá trị lưu trong Đích
- VD: MOV AX, 1000; $AX \leftarrow 1000$
 NEG AX; $AX \leftarrow -(AX) = -1000$

❖ Lệnh XCHG – Tráo đổi giá trị hai toán hạng

- Dạng: XCHG <Operand1>, <Operand2>
- Ý nghĩa: Tráo đổi giá trị hai toán hạng <Operand1> và <Operand2>
- VD: MOV BX, 100
 MOV AX, 200
 XCHG AX, BX; $AX \leftarrow 100, BX \leftarrow 200$

8. Tập lệnh – Một số lệnh khác

- ❖ Lệnh REP – lặp việc thực hiện các lệnh MOVSB, MOVSW, LODSB, LODSW, STOSB, STOSW một số lần – số lần lưu trong thanh ghi CX.
 - Dạng: REP <Lệnh cần lặp>
 - Ý nghĩa: Lặp CX lần việc thực hiện một lệnh khác
 - VD:
 - MOV SI, 1000; Đặt địa chỉ nguồn
 - MOV DI, 2000; Đặt địa chỉ đích
 - MOV CX, 10; Đặt số lần lặp cho REP
 - REP MOVSB; Thực hiện MOVSB 10 lần: chuyển nội dung 10 ô nhớ bắt đầu từ DS:SI sang 10 ô nhớ bắt đầu từ ES:DI

8. Tập lệnh – Một số lệnh khác

❖ Lệnh INT – Triệu gọi dịch vụ ngắt

- Dạng: INT <Số hiệu ngắt>
- Ý nghĩa: Gọi thực hiện chương trình con phục vụ ngắt tương ứng với <Số hiệu ngắt>
- VD: MOV AH, 4Ch; Nạp hàm 4Ch
 INT 21h; Gọi ngắt DOS số 21h