

# **Lập trình hợp ngữ 8086/88 và các chế độ địa chỉ**

# 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 gọi nhớ → chính là các lệnh của hợp ngữ

# 1. Nhắc lại về tập thanh ghi

- Thanh ghi đa năng
  - Thanh ghi AX: Lưu kết quả tính toán
  - Thanh ghi BX: Thường dùng để lưu địa chỉ cơ sở
  - Thanh ghi CX: Lưu số lần lặp trong lệnh LOOP
  - Thanh ghi DX: Kết hợp với thanh ghi AX để lưu kết quả của phép nhân hoặc chia số 16 bit. Ngoài ra còn dùng để lưu địa chỉ 16 bit của cổng vào ra

# 1. Nhắc lại về thanh ghi

- Thanh ghi đoạn
  - Thanh ghi CS: Để ghi địa chỉ mã lệnh
  - Thanh ghi DS: Chứa địa chỉ dữ liệu
  - Thanh ghi SS: Chứa địa chỉ ngăn xếp
  - Thanh ghi ES: Thanh ghi phụ, thường dùng để chứa địa chỉ chuỗi

# 1. Nhắc lại về thanh ghi

- Thanh ghi con trỏ và chỉ số
  - IP (PC): Thanh ghi trỏ tới lệnh tiếp theo
  - BP: Thanh ghi trỏ về dữ liệu bộ nhớ
  - SP: Trỏ tới đỉnh ngăn xếp
  - SI: Thanh ghi chỉ số nguồn, trỏ vào dữ liệu, kết hợp với thanh ghi DS để chỉ địa chỉ của dữ liệu là DS:SI
  - DI: Thanh ghi chỉ số đích, trỏ vào dữ liệu, kết hợp với ES để có ES:DI

# Khai báo dữ liệu

- Khai báo biến:

A1 DB 1	; A1 dài 1 byte, A1=1
A2 DB ?	; A2 dài 1 byte và không có giá trị
A3 DW 1	; A3 dài 2 byte và A3=1
A4 DD ?	; A4 dài 4 byte, không có giá trị ban đầu
A5 DB 1,2,3	; Mảng 3 phần tử
A6 DB 10 DUP(?)	; Mảng 10 phần tử, không gán giá trị ban đầu
A7 DB 'A'	; Biến kiểu ký tự

- Khai báo hằng số:

Muoi EQU 10  
BaA EQU 'AAA'

## 2. Tập lệnh 8086

### Dạng lệnh

<Mã gọi nhớ> <Toán hạng đích>[,<Toán hạng nguồn>]

- **Mã gọi nhớ:** Giúp người sử dụng biết hoạt động của lệnh
- **Toán hạng đích:**
  - Giữ kết quả (nếu có yêu cầu) sau khi thi hành lệnh.
  - Có thể là thanh ghi hay bộ nhớ.
- **Toán hạng nguồn:** có thể là thanh ghi, bộ nhớ hay một số tức thời

# Các nhóm lệnh

- Chuyển dữ liệu
- Xử lý số học với số nguyên
- Xử lý logic
- Điều khiển vào-ra
- Chuyển điều khiển (rẽ nhánh)
- Điều khiển hệ thống
- Xử lý số dấu phẩy động
- Xử lý các dữ liệu chuyên dụng



## 2.1 Nhóm lệnh chuyển dữ liệu

- Nhóm lệnh chuyển dữ liệu đa dụng
- Nhóm lệnh chuyển địa chỉ
- Nhóm lệnh chuyển cờ

# Nhóm lệnh chuyển dữ liệu đa dụng

- **MOVE** Copy dữ liệu từ nguồn đến đích
- **LOAD** Nạp dữ liệu từ bộ nhớ đến bộ xử lý
- **STORE** Cát dữ liệu từ bộ xử lý đến bộ nhớ
- **EXCHANGE** Trao đổi nội dung của nguồn và đích
- **CLEAR** Chuyển các bit 0 vào toán hạng đích
- **SET** Chuyển các bit 1 vào toán hạng đích
- **PUSH** Cát nội dung toán hạng nguồn vào ngăn xếp
- **POP** Lấy nội dung đỉnh ngăn xếp đưa đến toán hạng đích

# Lệnh MOV

## MOV đích, nguồn

Đích	Nguồn	Ví dụ	Giải thích
1 Bộ nhớ	Thanh ghi	<b>MOV DS:100H, AX</b>	- chuyển nội dung trong AX vào vị trí nhớ 100H.
2 Thanh ghi	Bộ nhớ	<b>MOV AX, MEM1</b>	-Chuyển nội dung trong vị trí nhớ do nhãn MEM1 chỉ ra vào thanh ghi AX.
3 Thanh ghi	Thanh ghi	<b>MOV AX, BX</b>	-- Chuyển nội dung trong BX vào thanh ghi AX.
4 Thanh ghi	Tức thời	<b>MOV AX, 0FFFFH</b>	- Chuyển giá trị hằng số FFFFH vào thanh ghi AX

# PUSH/POP

- **Lệnh PUSH:** Cất dữ liệu vào ngăn xếp
  - Viết lệnh: **PUSH nguồn**
  - Mô tả:  $SP \quad SP - 2$   
Nguồn {SP}.
  - Ví dụ:
    - `PUSH BX;` Cất BX vào ngăn xếp tại vị trí do SP chỉ ra
    - `PUSH Table[BX]` ;Cất 2 byte của vùng dữ liệu DS có địa chỉ đầu tại `Table+BX`

# Ví dụ lệnh PUSH/POP

- **Lệnh POP:** Lấy dữ liệu từ ngăn xếp
  - Viết lệnh: **POP đích**
  - Mô tả: **Đích {SP}.**  
**SP SP + 2**
  - Ví dụ:
    - POP BX; lấy 2 byte từ đỉnh ngăn xếp  
đưa vào BX
    - POP Table[BX] ;lấy 2 byte từ đỉnh ngăn xếp  
;đưa vào DS có địa chỉ đầu  
;tại Table+BX

# Nhóm lệnh chuyển địa chỉ

- Lệnh

**LEA reg, mem**

chuyển địa chỉ offset của toán hạng bộ nhớ vào thanh ghi reg16

# Nhóm lệnh chuyển cờ

- Lệnh LAHF (Load AH from flag)
- Lệnh SAHF (Store AH into flag)

# Nhóm lệnh chuyển dữ liệu qua cổng

- **INPUT** Copy dữ liệu từ một cổng xác định đến đích
- **OUTPUT** Copy dữ liệu từ nguồn đến một cổng xác định



# Lệnh IN/OUT

- **Lệnh IN:** Đọc dữ liệu từ cổng vào thanh ghi Acc
  - Viết lệnh: **IN Acc, Port**
  - Port: từ 00h .. FFh
  - Nếu Acc là AL: dữ liệu 8 bit được đưa vào từ cổng port
  - Nếu Acc là AX: Dữ liệu 16 bit được đưa vào từ cổng port, port+1
  - Cách khác: **IN Acc, DX**
    - DX: Được gán giá trị cổng (0000h .. FFFFh) từ trước

# Lệnh IN/OUT

- **Lệnh OUT:** Đưa dữ liệu từ thanh ghi Acc ra cổng
  - Viết lệnh: **OUT Port, Acc**
  - Port: từ 00h .. FFh
  - Nếu Acc là AL: dữ liệu 8 bit được đưa ra cổng
  - Nếu Acc là AX: Dữ liệu 16 bit được đưa ra cổng port, port+1
  - Cách khác: **OUT DX, Acc**
    - DX: Được gán giá trị cổng (0000h .. FFFFh) từ trước

# Nhóm lệnh số học

- ADD Cộng hai toán hạng
- SUBTRACT Trừ hai toán hạng
- MULTIPLY Nhân hai toán hạng
- DIVIDE Chia hai toán hạng
- ABSOLUTE Lấy trị tuyệt đối toán hạng
- NEGATE Đổi dấu toán hạng (lấy bù 2)
- INCREMENT Tăng toán hạng thêm 1
- DECREMENT Giảm toán hạng đi 1
- COMPARE Trừ hai toán hạng để lập cờ

# Lệnh ADD/SUB

- Viết lệnh:  
**ADD** **đích, nguồn**; đích đích + nguồn  
**SUB** **đích, nguồn**; đích đích - nguồn
- Đích và nguồn phải có cùng kích thước
- Bảng loại địa chỉ khác nhau của đích và nguồn

<b>Đích (nơi đến)</b>	<b>Nguồn (gốc)</b>
Thanh ghi	Thanh ghi
Thanh ghi	Bộ nhớ
Bộ nhớ	Thanh ghi
Bộ nhớ	Tức thời(hằng số)
Thanh ghi	Tức thời (hằng số)

# Ví dụ

- Viết đoạn chương trình cộng 5h với 3h dùng hai thanh ghi AL và BL

MOV AL, 05H ; AL 05H

MOV BL, 03H ; BL 03H

ADD AL, BL ; AL 05H+03H =08H

MOV DS:100H, AL ; Di chuyển kết quả từ AL vào vị trí nhớ DS:100H.

# Lệnh MUL

- Viếtlệnh:  
**MUL** số nhân nguồn
- Mô tả
  - Nhân 2 số không dấu
  - Toán hạng nguồn 8 bit: AX AL\*số nhân nguồn
  - Toán hạng nguồn 16 bit: DXAX AX\*số nhân nguồn

# MUL: Ví dụ

- Ví dụ1: Viết đoạn chương trình nhân 5h với 3h dùng thanh ghi CL

MOV AL, 05H ; AL 5H (số bị nhân)

MOV CL, 03H ; CL 3H (số nhân)

MUL CL ; AX FH (kết quả)

MOV MEM1, AX ; chuyển kết quả(0FH)

; từ AX vào vị trí nhớ có nhãn

MEM1

# MUL: Ví dụ

- Ví dụ 2: Viết đoạn chương trình nhân 3A62H với 2B14H.

MOV AX, 3A62H ; AX 3A62H

MOV CX, 2B14H ; CX 2B14H

MUL CX ; DXAX tích =

289C63A8H



# Lệnh DIV

- Viết lệnh:  
**DIV số chia nguồn**
- Mô tả
  - Chia 2 số không dấu
  - **Toán hạng nguồn 8 bit:**
    - AL AX/số chia nguồn,
    - AH số dư AX/số chia nguồn
  - **Toán hạng nguồn 16 bit:**
    - AX DXAX/số chia nguồn
    - DX số dư DXAX/số chia nguồn

# DIV: Ví dụ

- Ví dụ 1: Viết đoạn chương trình chia 6h cho 3h, dùng thanh ghi CL

`MOV AX, 0006H` ; AX 6H

`MOV CL, 03H` ; CL 3H

`DIV CL` ; AH:AL 0H (số dư), 2H (thương số)

- Chú ý: 6H được đưa vào thanh 0006H để lấp đầy toàn bộ thanh ghi AX. Như vậy các byte trọng số cao của AX sẽ bị xoá để tránh bị lỗi.

# DIV: Ví dụ

- Ví dụ 2: Viết đoạn chương trình chia 1A034H cho 1002H, dùng thanh ghi BX

```
MOV AX, A034H ; AX  A034H
MOV DX, 0001H ; DX  0001H
MOV BX, 1002H ; BX  1002H
DIV BX        ; DXAX  00H (số dư)1AH
               ; (thương số)
```

# Nhóm lệnh logic

AND	Thực hiện phép AND hai toán hạng
OR	Thực hiện phép OR hai toán hạng
XOR	Thực hiện phép XOR hai toán hạng
NOT	Đảo bit của toán hạng (lấy bù 1)
TEST	Thực hiện phép AND hai toán hạng để lập cờ
SHIFT	Dịch trái (phải) toán hạng
ROTATE	Quay trái (phải) toán hạng

# Lệnh NOT

- Viết lệnh:

**NOT** **Đích** ; Đích  $\overline{\text{Đích}}$

- Ví dụ: Xác định kết quả của đoạn chương trình sau:

```
MOV BL, 00110011B
```

```
NOT BL
```

```
MOV MEM1, BL
```

# Lệnh AND/OR/XOR

- Viết lệnh:

**AND**   đích, nguồn

**OR**   đích, nguồn

**XOR**   đích, nguồn

- Ví dụ:

**AND BL, 0FH** ; che 4 bit cao của BL.

**OR BL, 30H**   ; Thiết lập bit b4 và b5 bằng 1.

**XOR BL,BL**   ; Xoá các bit của BL

# Nhóm lệnh xử lý chuỗi

- Lệnh LODS

**LODSB** ;     AL ← DS:SI, SI ← SI+1

**LODSW** ;     AX ← [DS:SI+1;DS:SI], SI ← SI+2

- Lệnh STOS

**STOSB** ;     ES:DI ← AL , DI ← DI+1

**STOSW** ;     [ES:DI+1, ES:DI] ← AX , DI ← DI+2

- MOVS: Chuyển nội dung tại địa chỉ trỏ bởi DS:SI đến địa chỉ trỏ bởi ES:DI. Giá trị của SI và DI thay đổi giống trường hợp trên
- CMPS: So sánh byte hay từ tại vị trí DS:SI và ES:DI. Giá trị của SI và DI thay đổi giống trường hợp trên

# Nhóm lệnh chuyển điều khiển

- Lệnh nhảy không điều kiện
- Lệnh nhảy có điều kiện
- Lệnh so sánh
- Vòng lặp



# Lệnh nhảy không điều kiện

- Lệnh

**JMP label**

Chuyển chương trình từ vị trí này sang vị trí khác

# Lệnh nhảy có điều kiện

- **JA label**: Nhảy nếu lớn hơn
- **JB label**: Nhảy nếu nhỏ hơn
- **JE label**: Nhảy nếu bằng

Ví dụ:

```
cmp al,13  
JE  dungnhap
```

# Lệnh so sánh

- Lệnh

**CMP** left,right

Dùng để so sánh nội dung 2 toán hạng

Ví dụ:

MOV AX,A

CMP AX,B

JA label1

JB label2

JMP label3

# Ngắt 21h của 8086

- Hàm 01h: Nhập 1 ký tự từ bàn phím và hiện ký tự ra màn hình, AL chứa ký tự nhập

**MOV Ah,01h**

**INT 21h**

- Hàm 02h: Xuất 1 ký tự trong thanh ghi DL ra màn hình.

**MOV AH,02h**

**MOV DL,'A'**

**INT 21h**

# Ngắt 21h của 8086

- Hàm 09h: Xuất một chuỗi ký tự ra màn hình, địa chỉ chuỗi lưu trong DS:DX, chuỗi kết thúc bằng ký tự \$
- Hàm 0Ah: Nhập một chuỗi ký tự từ bàn phím đến khi Enter. Địa chỉ chuỗi trong DS:DX  
Chuỗi có dạng:  
trường 0: Số byte tối đa cần đọc  
trường 1: Số byte đã đọc  
trường 2: Lưu chuỗi

```
.DATA  
Msg DB 'Hello$'  
...  
MOV AH,09h  
LEA DX,Msg  
INT 21h
```

```
.DATA  
Msg      DB 101 ; Đọc tối đa  
          100 ký tự  
          DB ?  
          DB 101 DUP(?)  
...  
MOV AH,0Ah  
LEA DX,Msg  
INT 21h
```

# Khung chương trình hợp ngữ

```
.MODEL    Kiểu kích thước bộ nhớ ; Khai báo quy mô sử dụng
bộ nhớ
.STACK    Kích thước                ; Khai báo dung lượng
đoạn stack
.DATA                                           ; Khai báo đoạn dữ
liệu
        msg DB 'Hello$'
.CODE                                           ; Khai báo đoạn mã
main PROC
...
main ENDP
END main
```

# Ví dụ

- Viết chương trình hiển thị dòng chữ “Hello”

```
.MODEL Small
```

```
.STACK 100h
```

```
.DATA
```

```
msg DB 'Hello$'
```

```
.CODE
```

```
main PROC
```

```
MOV AX,@DATA ; Khởi động thanh ghi DS
```

```
MOV DS,AX
```

```
LEA DX,msg ; chuyển địa chỉ vào DX
```

```
MOV AH,09h ; In thông điệp ra màn hình
```

```
INT 21h
```

```
mov AH,01h ; dừng màn hình, đợi ký tự
```

```
int 21h ; nhập
```

```
main ENDP
```

```
END main
```

# Bài tập

Bài 1. Nhập vào 1 ký tự và hiển thị ra màn hình

Bài 2. Nhập vào một chuỗi và hiển thị chuỗi này ra màn hình

Bài 3. Đọc hiểu bài “Nhập một chuỗi ký tự và chuyển chữ thường thành chữ hoa” trong tài liệu

Bài 4. Lập trình thực hiện các công việc sau:

- Hiển thị thông báo : 'Hãy gõ vào một chữ cái hoa: '
- Vào một chữ cái hoa
- Xuống dòng, về đầu dòng
- Hiển thị thông báo : 'Chữ cái thường là: '
- Hiển thị chữ cái thường tương ứng
- Xuống dòng về đầu dòng
- Hiển thị thông báo : 'Mã ASCII của chữ cái là:'
- Hiện thị mã ASCII của chữ cái.
- Thoát về DOS.



# 3.Các phương pháp định địa chỉ

Khái niệm về định địa chỉ (addressing)

- 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 là cách thức địa chỉ hóa trong trường địa chỉ của lệnh để xác định toán hạng

# Các phương pháp định địa chỉ thông dụng

- Định địa chỉ tức thì
- Định địa chỉ thanh ghi
- Định địa chỉ trực tiếp
- Định địa chỉ gián tiếp qua thanh ghi
- Định địa chỉ dịch chuyển

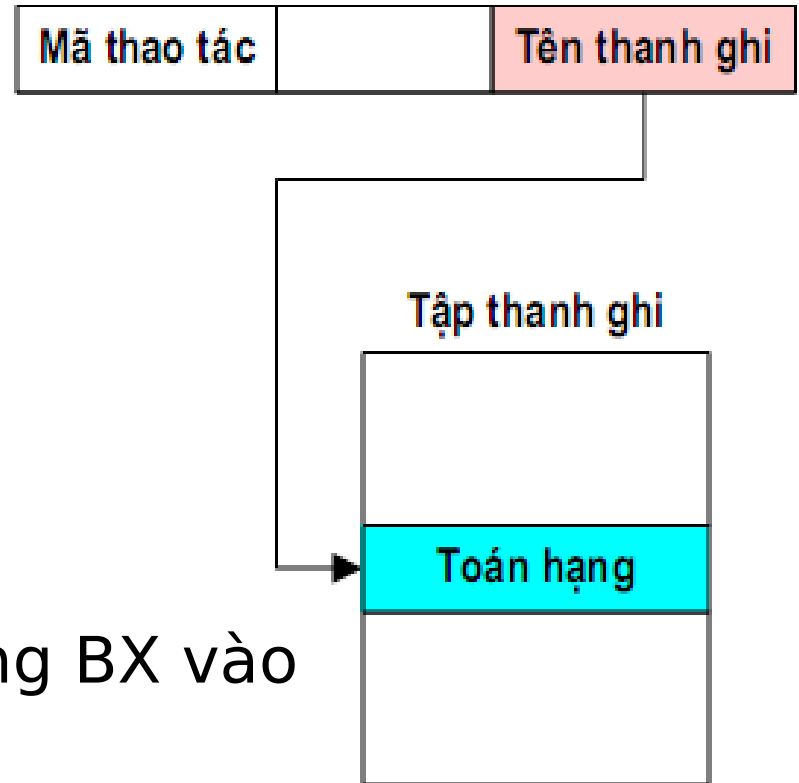
# Định địa chỉ tức thì

Mã thao tác		Toán hạng
-------------	--	-----------

- Toán hạng đích: Thanh ghi, ô nhớ
- Toán hạng nguồn: Hằng số

# Định địa chỉ thanh ghi

- Sử dụng các thanh ghi như là toán hạng để chứa dữ liệu cần để thao tác



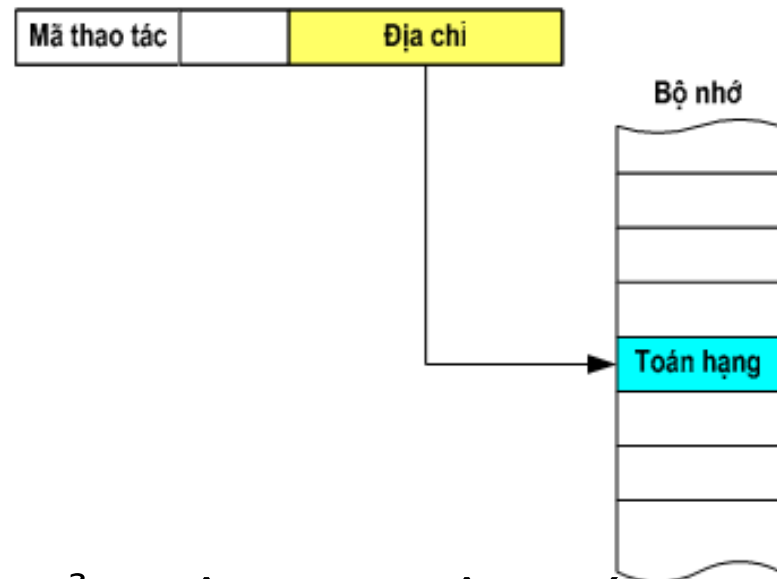
MOV AX, BX ; chuyển nội dung BX vào AX.

ADD DS, DL ; cộng nội dung AL và DL ,

; kết quả giữ trong AL.

# Định địa chỉ trực tiếp

- Toán hạng: Địa chỉ ô nhớ để chứa dữ liệu
- Toán hạng kia: Thanh ghi



MOV AL, DS:1234H ; chuyển nội dung ô nhớ  
;DS:1234H vào AL.

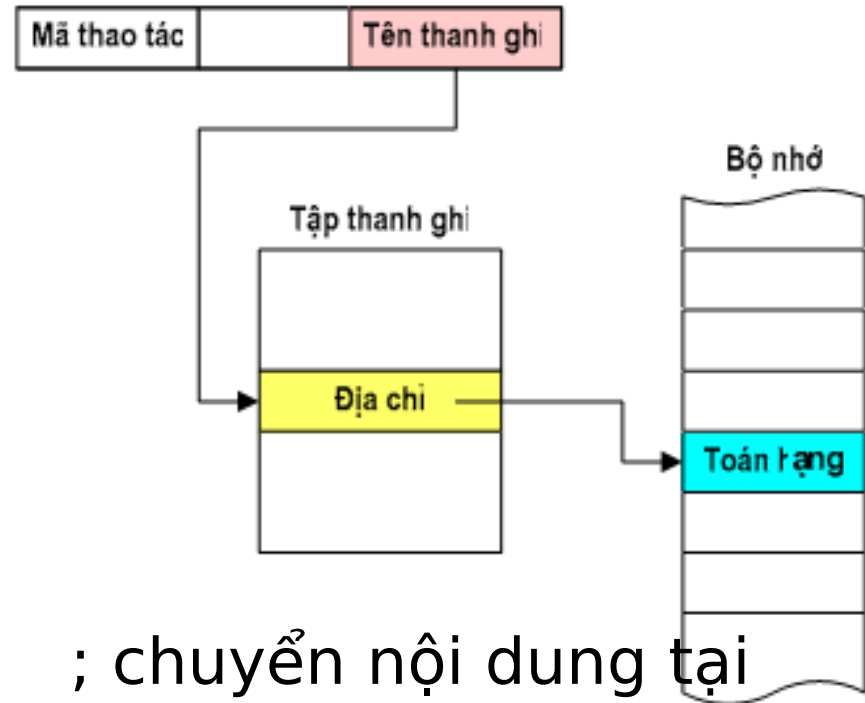
MOV DS:4321H, CX ; chuyển nội dung CX vào 2 vị trí  
nh

; liên tiếp là DS:4321 và

DS:4322.

# Định địa chỉ gián tiếp qua thanh ghi

- Toán hạng: Thanh ghi chứa địa chỉ ô nhớ của dữ liệu
- Toán hạng kia: Thanh ghi

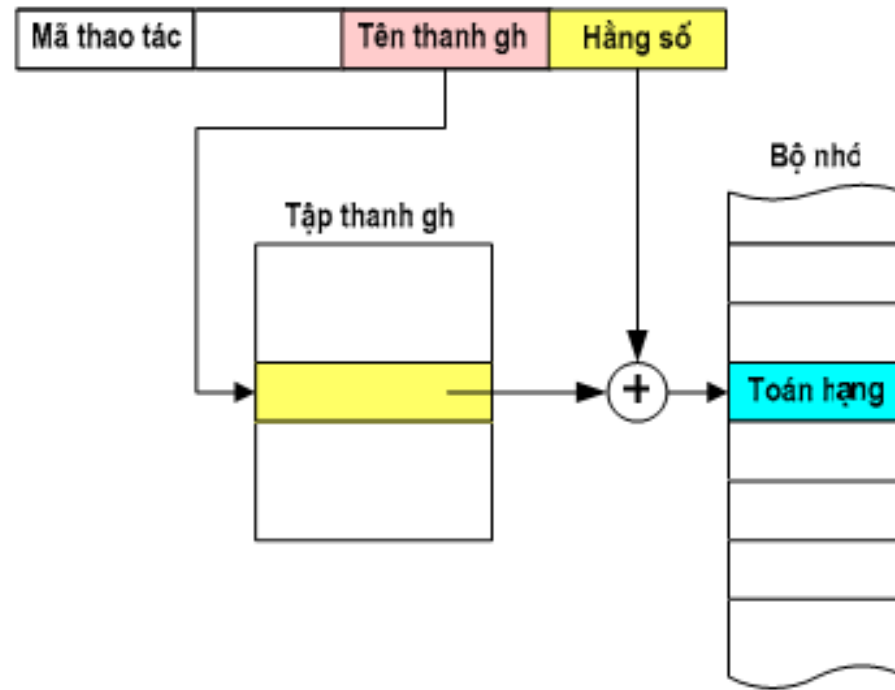


ô      MOV AL, DS:BX      ; chuyển nội dung tại

                                 ; nhớ DS:BX vào AL.

MOV DS:SI, CL ; chuyển nội dung CL vào  
                 ; ô nhớ DS:SI.

# Định địa chỉ tương đối



- Để xác định toán hạng gồm 2 thành phần
  - Tên thanh ghi và hằng số
  - Địa chỉ toán hạng = Nội dung thanh ghi + hằng số

# Ví dụ: Định địa chỉ tương đối

## Tương đối chỉ số

MOV AL, DS:[SI+10] ; chuyển nội dung ô nhớ DS:  
(SI+10)  
; vào AL.

## Tương đối cơ sở

MOV CX, DS:[BX + 10] ; chuyển nội dung 2 ô nhớ liên  
tiếp có  
; địa chỉ DS:(BX+10) và DS:  
(BX+11)  
;vào CX.