Chương 6 : Toán tử – Toán hạng - các phép định địa chỉ – Tập lệnh

- Hiểu cách dùng toán tử trong ASM.
- Nắm được tập lệnh của CPU 8086/8088.
- Biết cách định địa chỉ thông qua toán hạng.
- Biết vận dụng các kỹ năng trên vào vấn đề cần giải quyết và hiện thực thành chương trình.

Toán tử

Điểm quan trọng phân biệt giữa toán tử và lệnh là :

- Toán tử điều khiển sự tính toán các trị hằng xác định lúc dịch.
- Lệnh điều khiển sự tính toán các trị không xác định được cho đến khi CT thực hiện.

Ex : toán tử + điều khiển phép cộng khi dịch.

Lệnh cộng ADD điều khiển phép cộng khi chương trình thực hiện.

Toán tử số học

Toán tử	Cú pháp	Công dụng
+	+ expression	Dương
-	- expression	Âm
*	exp1*exp2	Nhân
/	exp1/exp2	Chia
MOD	exp1 mod exp2	Phần dư
+	exp1 + exp2	Cộng
-	exp1 - exp2	Trừ
SHL	exp shl n	Dịch exp sang trái n bit
SHR	exp shr n	Dịch exp sang phải n bit

Toán tử logic

Not	Not expression
And	Exp1 and exp2
Or	Exp1 or exp2
Xor	Exp1 xor exp2

Ex : MOV AH , 8 OR 4 AND 2

MOV AL, NOT (20 XOR 0011100B)

Toán Tử Quan Hệ

- **So sánh 2 biểu thức và cho trị là true (-1) nếu điều kiện của toán tử thỏa, ngược lại là false.**

EQ	Exp1 EQ exp2	True nếu Exp1 = exp2
NE	Exp1 NE exp2	True nếu Exp1 <> exp2
LT	Exp1 LT exp2	True nếu Exp1 < exp2
LE	Exp1 LE exp2	True nếu Exp1 <= exp2
GT	Exp1 GT exp2	True nếu Exp1 > exp2
GE	Exp1 GE exp2	True nếu Exp1 >= exp2

ĐỘ ƯU TIÊN TOÁN TỬ

TOÁN TỬ	MÔ TẢ
( )	Dấu ngoặc
+ , -	Dấu dương , âm
* / MOD	Nhân , chia, Modulus
+ , -	Cộng, trừ

Toán tử SEG

- Cú pháp :
  SEG expression
- Cho địa chỉ đoạn của biểu thức expression.
- Expression có thể là biến | nhãn | tên segment hay toán hạng bộ nhớ khác.

Toán tử OFFSET

- Cú pháp :
  OFFSET expression
- Cho địa chỉ OFFSET của biểu thức expression.
- Expression có thể là biến | nhãn | tên segment hay toán hạng trực tiếp bộ nhớ khác.

Ex : nạp địa chỉ segment và offset của biến table vào DS :AX

TABLE DB ?

MOV AX, SEG TABLE

MOV DS, AX

MOV DX, OFFSET Table

TOÁN TỬ $

- **Cho địa chỉ của OFFSET của phát biểu chứa toán tử $.**
- **Thường được dùng để tính chiều dài chuỗi.**

TOÁN TỬ PTR

Cú pháp : type PTR expression

- **Cho phép thay đổi dạng của expression**
- **nếu expr là 1 biến | toán hạng bộ nhớ thì type có thể là byte , word hay dword.**
- **Nếu expr là 1 nhãn thì type có thể là near hay far.**

Ex : mov ax, word ptr var1 ; var1 là toán hạng kiểu Word

mov bl , byte ptr var2 ; var2 là toán hạng kiểu byte

Toán hạng (Operand)

Các toán hạng chỉ ra nơi chứa dữ liệu cho 1 lệnh , chỉ thị.

Hầu hết các lệnh Assembly đều có đối số là 1 hoặc 2 toán hạng
Có 1 số lệnh chỉ có 1 toán hạng như RET, CLC.

Với các lệnh 2 toán hạng thì toán hạng thứ 2 là toán hạng nguồn (source) – chứa dữ liệu hoặc địa chỉ của dữ liệu.

Toán hạng (Operand)

- **Toán hạng đích giữ kết quả (nếu có yêu cầu) sau khi thi hành lệnh.**
- **Toán hạng đích có thể là thanh ghi hay Bộ nhớ.**

Toán hạng nguồn có thể là thanh ghi, bộ nhớ hay 1 giá trị tức thời .
Toán hạng số tức thời có thể là số trong các hệ đếm khác nhau và được viết theo qui định sau :

Số hệ 2 : xxxxxxxxB (x là bit nhị phân)
Số hệ 10 : xxxxxD hay xxxxx (x là 1 số hệ 10)
Số hệ 16 : xxxxH và bắt đầu bằng số (x là 1 số hệ 16)

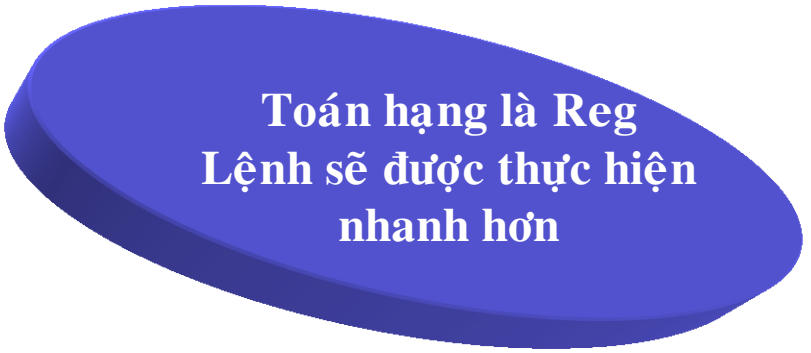
Cơ chế định vị địa chỉ

- Cách xác định dữ liệu trong lệnh được gọi là cơ chế định vị địa chỉ (addressing mode) ➔ chỉ ra nơi cất dữ liệu

- Cơ chế này chia làm 3 loại : định vị bằng thanh ghi, bằng giá trị tức thời và bằng bộ nhớ.

MOV AL, BL ; định vị bằng thanh ghi

INC BX ; định vị bằng t



ĐỊNH VỊ TỨC THỜI

- Toán hạng tức thời là dữ liệu 8 bit hay 16 bit nằm ngay trong câu lệnh.
- Dữ liệu xử lý được lưu ngay trong lệnh

  Ex : MOV CL, 61h ; ← toán hạng tức thời

  Mã máy của lệnh trên là **B161h**

ĐỊNH VỊ THANH GHI

Giá trị của toán hạng được truy xuất nằm ngay trong thanh ghi của CPU.

Ex : MOV AX,BX ; chuyển nội dung của thanh ghi BX vào thanh ghi AX

ĐỊNH VỊ BỘ NHỚ

- **Định vị gián tiếp thanh ghi :**
  địa chỉ toán hạng không chứa trực tiếp trong lệnh mà gián tiếp thông qua một thanh ghi.

Ex : SUB DX, [BX] ;

Khác với lệnh SUB DX, BX

Trong chế độ này, địa chỉ Offset của ô nhớ chứa nộI dung của toán hạng nằm trong các thanh ghi BX,BP,SI,DI.

Địa chỉ segment ngầm định chứa trong DS nếu dùng BX,SI,DI

Địa chỉ segment ngầm định chứa trong ES nếu dùng BP

Định vị gián tiếp thanh ghi :

EX1 : **MOV AX, [SI]**

Nạp nội dung của ô nhớ mà địa chỉ Offset lưu trong SI và địa chỉ đoạn lưu trong DS vào AX.

EX2 : **MOV AX, [BP]**

Nạp nội dung của ô nhớ mà địa chỉ Offset lưu trong BP và địa chỉ đoạn lưu trong ES vào AX.

ĐỊNH VỊ TRỰC TIẾP

Địa chỉ Offset của ô nhớ chứa dữ liệu toán hạng nằm trực tiếp trong câu lệnh còn địa chỉ segment ngầm định chứa trong DS.

Ex : MOV BX, [1234]

Nạp nội dung ô nhớ có địa chỉ DS:1234 ➔ BX

ĐỊNH VỊ CƠ SỞ

Địa chỉ Offset của toán hạng được tính là tổng của nội dung thanh ghi BX hoặc BP và 1 độ dịch.

Độ dịch là 1 số nguyên âm hoặc dương. Địa chỉ đoạn là đoạn hiện tại.

ĐỊA CHỈ HIỆU DỤNG

Toán hạng bộ nhớ dùng trong tập lệnh vi xử lý 86 sử dụng phương pháp định địa chỉ tổng hợp được gọi là địa chỉ hiệu dụng.

Địa chỉ hiệu dụng là tổ hợp của 3 nhóm sau đặt trong dấu [ ].

Nhóm thanh ghi chỉ số : SI , DI
Nhóm thanh ghi nền : BX, BP
Địa chỉ trực tiếp : số 16 bit

Các thanh ghi trong cùng 1 nhóm không được xuất hiện trong cùng 1 địa chỉ hiệu dụng.

ĐỊA CHỈ HIỆU DỤNG

Địa chỉ hiệu dụng hợp lệ :

[1000h] [SI], [DI] , [BX] , [BP]

[SI+BX], [SI+BP] , [DI+BX] , [DI+BP] , [SI+1000h], [DI+100h]

[SI] [BX] [1000h], [SI+BP+1000h] , [DI+BX][1000h], [DI+1000h]+[BP]

Địa chỉ hiệu dụng không hợp lệ :

[70000], [AX] , [SI+DI+1000h], [BX] [BP]

ĐỊA CHỈ HIỆU DỤNG (tt)

- Địa chỉ hiệu dụng chính là phần offset của địa chỉ luận lý bộ nhớ.
- Segment của địa chỉ hiệu dụng được mặc định như sau :
  nếu không sử dụng BP trong địa chỉ hiệu dụng thì mặc định theo DS.
  nếu có sử dụng BP trong địa chỉ hiệu dụng thì mặc định theo ES.

Địa chỉ hiệu dụng (tt)

Để thuận tiện trong vấn đề giải thích lệnh, ta qui ước sau :

Dữ liệu 8 bit bộ nhớ : [địa chỉ]

Dữ liệu 16 bit bộ nhớ : [địa chỉ +1, địa chỉ]

Để xác định rõ hoạt động của bộ nhớ , ta phải dùng thêm toán tử PTR như sau :

8 bit : BYTE PTR [1000H]

Tham khảo 1 byte bộ nhớ ở địa chỉ 1000h

16 bit : WORD PTR [1000H]

Tham khảo 2 byte bộ nhớ liên tiếp ở địa chỉ 1000h và 1001h

Ex : Tính tổng 1 array có 5 phần tử

```
MOV BX, OFFSET LIST
MOV AX, 0
MOV AL, [BX]
ADD AL , [BX+1]
ADD AL , [BX+2]
ADD AL , [BX+3]
ADD AL , [BX+4]
MOV  SUM , AX
.....
LIST  DB  10h, 20h, 40h, 2h, 5h
SUM  DW  0
```

Cách thực hiện :

Lấy địa chỉ của List vào BX

Dựa vào BX để xác định các phần tử của array.

Khi tính tổng xong, đưa tổng vào biến SUM.

CHẠY CT này bằng DEBUG

Ex : Tính tổng 1 array có 5 phần tử

```
-A 100
MOV  BX, 0120
MOV AX, 0
MOV AL, [BX]
ADD AL , [BX+1]
ADD AL , [BX+2]
ADD AL , [BX+3]
ADD AL , [BX+4]
MOV  [0125], AX
-A 120
DB 10, 20, 40, 2, 5
DW  0
```

Tập lệnh

Lệnh MOV :
Ý nghĩa : copy giá trị từ toán hạng nguồn → toán hạng đích
Cú pháp : MOV dest , source
Yêu cầu : Dest và source cùng kiểu
Dạng lệnh :
MOV reg , reg
MOV mem , reg
MOV reg, mem
MOV reg16, segreg
MOV segreg, reg16
MOV reg, immed
MOV mem, immed
MOV mem16, segreg
MOV segreg, mem16

Minh hoạ lệnh MOV

```
MOV AX, CX
MOV DL, BH
MOV [SI+1000h], BP ; [SI+1000h, SI+1001h] ← BP
MOV DX, [1000h]  ; DX ← [1000h, 1001h]
MOV DI, 12h
MOV AL, 12h
MOV  BYTE  PTR [1000h], 12h
MOV  WORD  PTR [2000h] , 1200h
MOV  [BX] , DS
MOV SS, [2000h]
```

Chú ý

- Lệnh MOV không làm ảnh hưởng đến cờ.
- Không thể chuyển dữ liệu trực tiếp giữa 2 toán hạng bộ nhớ với nhau, muốn chuyển phải dùng thanh ghi trung gian.
- Không thể chuyển 1 giá trị tức thời vào thanh ghi đoạn, muốn chuyển phải dùng thanh ghi trung gian.
- Không thể chuyển trực tiếp giữa 2 thanh ghi đoạn

Minh họa lệnh MOV

Ex1 : Cho table là 1 mảng gồm 10 phần tử dạng byte

Table DB 3,5,6,9,10, 29,30,46,45,90

Truy xuất phần tử đầu , phần tử thứ 2 và thứ 5 của mảng:

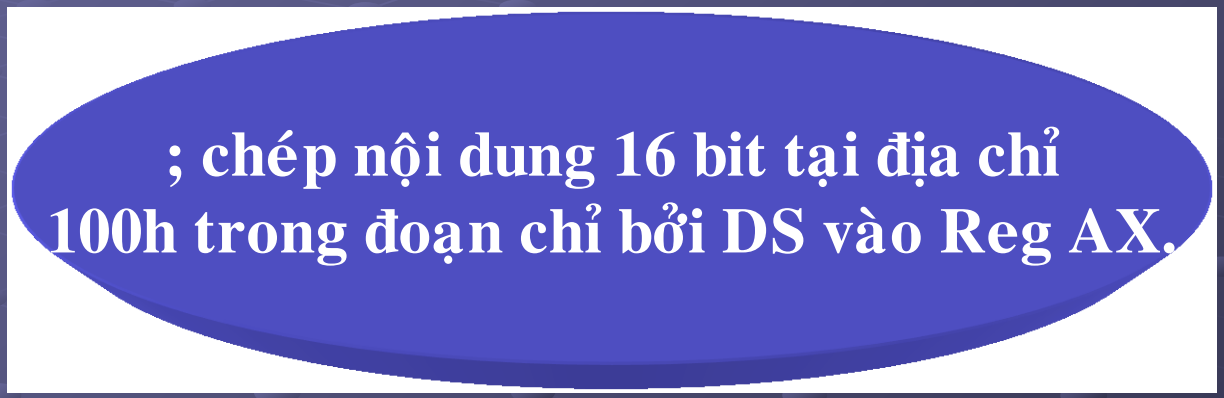
MOV AL, TABLE hay MOV AL, TABLE[0]

MOV AL, TABLE+1 hay MOV AL, TABLE[1]

MOV AL, TABLE+4 hay MOV AL, TABLE[4]

Minh họa lệnh MOV

Ex2 : MOV AX, DS : [100h]



Ex3 : MOV AX, [1

Áp dụng

Viết chương trình chuyển nội dung vùng nhớ bắt đầu tại địa chỉ 70 sang vùng nhớ có địa chỉ bắt đầu là 1000h. Biết chiều mỗi vùng nhớ là 9 bytes và dữ liệu đang khảo sát trong đoạn được chỉ bởi D

Cho vùng nhớ MEM có chiều dài 9 bytes gồm các ký tự 'abcdefghi' trong đoạn chỉ bởi DS.
Viết chương trình đảo ngược vùng nhớ MEM.

Lệnh LEA (Load Effective Address)

Cú pháp : LEA REG | MEM
ý nghĩa : nạp địa chỉ Offset vào thanh ghi để khởi động Reg.
Ex : MOV DX, OFFSET MES
Tương đương với LEA DX, MES

Ex : LEA BX, [1000h] ; BX ⟵ 1000h

LEA SI, [DI][BX][2000h] ; SI ⟵ DI + BX + 2000h

Lệnh XCHG (XCHANGE)

Cú pháp : XCHG DEST , SOURCE
ý nghĩa : hoán chuyển nội dung 2 Reg, Reg và ô nhớ
Yêu cầu :
2 toán hạng phải cùng kiểu
2 toán hạng không thể là 2 biến bộ nhớ. Muốn hoán đổi trị của 2 biến phải dùng Reg trung gian.

Ex : XCHG AH, BL

MOV VAR1, VAR2 ; không hợp lệ, phải dùng Reg tạm

Lệnh PUSH

Cú pháp : PUSH REG16
PUSH MEM16
PUSH SEGREG
Đẩy toán hạng nguồn 16 bit vào STACK

Ex : PUSH DI ; [SS :SP+1, SS :SP]⟵ DI

Ex : PUSH CS ; [SS :SP+1, SS :SP]⟵ CS

Lệnh POP

Cú pháp : POP REG16
POP MEM16
POP SEGREG
Lấy dữ liệu từ đỉnh STACK vào toán hạng đích.

Ex : POP AX ; AX ⟵ [SS :SP+1, SS :SP]

Ex : POP [BX+1] ; [BX+2, BX+1] ⟵ [SS :SP+1, SS :SP]

Lệnh IN

Cú pháp : IN ACCUM, IMMED8
IN ACCUM, DX
nhập dữ liệu từ cổng xuất nhập vào thanh ghi tích luỹ AL hay AX. Trường hợp AX sẽ nhập byte thấp trước, byte cao sau.

Ex : IN AL ,61h

IN AX, 40h

Ex : MOV DX, 378H

IN AL, DX

Dạng lệnh có Reg DX dùng Để cho cổng có địa chỉ 16 bit

SUMMARY

- Dùng DEBUG để hợp dịch và chạy chương trình sau : Chép 3 số nguyên kiểu Word ở địa chỉ 0120h vào địa chỉ 0130h.
- Cho biết giá trị của AX sau khi các lệnh sau được thực thi :

```
MOV AX, ARRAY1
 INC   AX
 ADD  AH, 1
 SUB AX, ARRAY1
 ……
ARRAY1 DW  10h, 20h
```

SUMMARY

- Giả sử biến VAL1 ở địa chỉ offset 0120h và PTR1 ở địa chỉ 0122h. Cho biết giá trị của các thanh ghi AX, BX khi mỗi lệnh sau được thực thi :

```
.CODE
  MOV AX, @DATA
MOV DS, AX
MOV AX, 0
MOV AL, BYTE  PTR  VAL1   ; AX = ?
MOV BX, PTR1              ; BX = ?
XCHG AX, BX               ; BX = ?
SUB AL,2                  ; AX = ?
MOV AX, PTR2              ; AX = ?
.DATA
   VAL1  DW 3Ah
   PTR1 DW  VAL1
   PTR2  DW  PTR1
```

Cho biết giá trị của các thanh ghi ở bên phải, khi mỗi lệnh của đoạn chương trình sau được thực thi. Giả sử FIRST ở offset 0H

```
MOV AL, BYTE  PTR  FIRST+1                    ; AL =
MOV  BX, WORD  PTR  SECOND+2                  ; BX =
MOV DX, OFFSET  FIRST + 2                     ; DX  =
MOV AX, 4C00H
INT 21H
.......
FIRST  DW  1234h
SECOND  DW  16385
THIRD  DB  10,20,30,40
```

Bài tập Lập trình

Bài 1 : Viết chương trình nhập 1 ký tự.
Hiển thị ký tự đứng trước và ký tự đứng sau ký tự đã nhập theo thứ tự mã ASCII.
Kết quả có dạng :
Nhập một ký tự : B
Ký tự đứng trước : A
Ký tự đứng sau : C

Bài 2 : Viết chương trình nhập 2 ký tự và hiển thị ký tự thứ 3 có mã ASCII là tổng của mã 2 ký tự đã nhập.
Kết quả có dạng :