**LAB 2: CÁC KIỂU ĐỊNH VỊ VÀ THANH GHI CỜ**

**I. Lý thuyết**

1. **Giả sử AX = 0500h, BX = 1000h, SI = 1500h, DI = 2000h, [DS:1000] = 0100h, [DS:1500] = 0150h, [DS:2000] = 0200h, [DS:2500] = 0250h, [DS:3000] = 3000h, và biến BETA là biến từ nằm ở địa chỉ ô 1000h. Trong các lệnh sau đây, nếu hợp lệ, hãy cho biết địa chỉ ô của toán hạng nguồn hoặc thanh ghi và kết quả lưu trong toán hạng đích. Nếu lệnh không hợp lệ, giải thích tại sao ?**
2. MOV DL, SI ; Không hợp lệ (DL chứa 8 bit, SI chứa 16 bit)
3. MOV DI, [DI] ; DI 0200h
4. ADD AX, [SI] ; AX 0650h
5. SUB BX, [DI] ; BX 0E00h
6. LEA BX, BETA[BX] ; BX 1100h
7. ADD [SI], [DI] ; Không hợp lệ (2 toán hạng đều là ô nhớ)
8. ADD BH, [BL] ; BH 10h
9. ADD AH, [SI] ; AH 0155h
10. MOV AX, [BX+DI+BETA] ; AX 3000h
11. **Giả sử DI = 2000h, [DS:2000] = 0200h. Hãy cho biết địa chỉ ô của toán hạng nguồn và kết quả lưu trong toán hạng đích sau khi thực hiện lệnh   
    MOV DI, [DI]**

* Địa chỉ nguồn: [DS:2000h]
* Sau khi thực hiện lệnh, DI = 0200h

1. **Giả sử AX = 0500h, SI = 1500h, [DS:1500] = 0150h. Hãy cho biết địa chỉ ô của toán hạng nguồn và kết quả lưu trong toán hạng đích của lệnh ADD AX, [SI]**

* Địa chỉ nguồn: [DS:1500]
* Sau khi thực hiện lệnh, AX = 0150h

1. **Giả sử BX = 1000h, [DS:2000] = 0200h và BETA là biến từ nằm ở địa chỉ ô 1000h. Hãy cho biết địa chỉ ô của toán hạng nguồn và kết quả lưu trong toán hạng đích sau khi thực hiện lệnh LEA BX, BETA[BX]**

* Địa chỉ nguồn: [DS:2000]
* Sau khi thực hiện lệnh, BX = 0200h

1. **Với khai báo: A DB 1, 2, 3. Hãy cho biết trị của toán hạng đích khi thực hiện lệnh MOV AH, BYTE PTR A**

* Giá trị của toán hạng đích: 1

1. **Với khai báo B DW 4, 5, 6. Hãy cho biết trị của toán hạng đích khi thực hiện lệnh MOV AX, WORD PTR B**

* Giá trị của toán hạng đích: 4

**II. Bài tập**

***Thực hiện các bài tập sau và kiểm tra lại bằng cách Debug chương trình***

1. **Cho biết nội dung của toán hạng đích và trị các cờ CF, SF, ZF, PF và OF sau khi thực hiện các lệnh sau:**
2. **ADD AX, BX ;Với AX = 7FFFh, BX = 1**

AX = 0111 1111 1111 1111b

BX = 0000 0000 0000 0001b

ADD AX, BX ; AX = 1000 0000 0000 0000b

* CF = 0, vì không có mượn từ bit MSB
* SF = 1, vì kết quả là 1000 0000 0000 0000b (bit MSB là 1)
* ZF = 0, vì kết quả là một số khác 0
* PF = 1, vì byte thấp của AX có tổng số bit 1 là số chẵn (0)
* OF = 1, vì kết quả 1000 0000 0000 0000b khác dấu

1. **SUB AL, BL ;Với AL = 1, BL = FFh**

AL = 0000 0001b

BL = 1111 1111b

SUB AL, BL ;AL = 0000 0010b

* CF = 1, vì có mượn từ bit MSB
* SF = 0, vì kết quả là 0000 0010b (bit MSB là 0)
* ZF = 0, vì kết quả là một số khác 0
* PF = 0, vì byte thấp của AL có tổng số bit 1 là số lẻ (1)
* OF = 0, vì kết quả 0000 0010b cùng dấu

1. **DEC AL ; Với AL = 0**

DEC AL ;AL = 1111 1111b

* CF = 0, vì không có mượn từ bit MSB
* SF = 1, vì kết quả là 1111 1111b (bit MSB là 1)
* ZF = 0, vì kết quả là một số khác 0
* PF = 1, vì byte thấp của AL có tổng số bit 1 là số chẵn (8)
* OF = 0, vì kết quả 1111 1111b cùng dấu

1. **NEG AL ;Với AL = 7Fh**

AL = 0111 1111b

NEG AL ;AL = 1000 0001b

* CF = 1, vì có mượn từ bit MSB
* SF = 1, vì kết quả là 1000 0001b (bit MSB là 1)
* ZF = 0, vì kết quả là một số khác 0
* PF = 1, vì byte thấp của AL có tổng số bit 1 là số chẵn (2)
* OF = 0, vì kết quả 1000 0001b cùng dấu

1. **XCHG AX, BX ; Với AX = 1ABCh, BX = 712Ah**

AX = 0001 1010 1011 1100b

BX = 0111 0001 0010 1010b

XCHG AX, BX ; AX = 0111 0001 0010 1010b

Các cờ không bị ảnh hưởng bởi lệnh XCHG

1. **ADD AL, BL ;Với Al = 80h, BL = FFh**

AL = 1000 0000b

BL = 1111 1111b

ADD AL, BL ; AL = 0111 1111b

* CF = 1, vì có mượn từ bit MSB
* SF = 0, vì kết quả là 0111 1111b (bit MSB là 0)
* ZF = 0, vì kết quả là một số khác 0
* PF = 0, vì byte thấp của AL có tổng số bit 1 là một số lẻ (7)
* OF = 1, vì kết quả 0111 1111b khác dấu

1. **SUB AX, BX ;Với AX = 0, BX = 8000h**

AX = 0000 0000 0000 0000b

BX = 1000 0000 0000 0000b

SUB AX, BX ;AX = 1000 0000 0000 0000b

* CF = 1, vì có mượn từ bit MSB
* SF = 1, vì kết quả là 1000 0000 0000 0000b (bit MSB là 1)
* ZF = 0, vì kết quả là một số khác 0
* PF = 1, vì byte thấp của AL có tổng số bit 1 là một số chẵn (0)
* OF = 1, vì kết quả 1000 0000 0000 0000b khác dấu

1. **NEG AX ;Với AX = 1**

AX = 0000 0000 0000 0001b

NEG AX ;AX = 1111 1111 1111 1111b

* CF = 1, vì có mượn từ bit MSB
* SF = 1, vì kết quả là 1111 1111 1111 1111b (bit MSB là 1)
* ZF = 0, vì kết quả là một số khác 0
* PF = 1, vì byte thấp của AX có tổng số bit 1 là một số chẵn (8)
* OF = 0, vì kết quả 1111 1111 1111 1111b cùng dấu

1. **Cho biết kết quả sau khi thực hiện lệnh ADD AX, BX và trạng thái cờ tràn (có hoặc không dấu) có xảy ra không với:**

a) AX = 512Ch, BX = 4185h

AX = 512Ch + 4185h = 92B1h

**Cờ tràn:**

**CF (Carry Flag):** 0 (không có carry ra ngoài bit cao nhất)

**OF (Overflow Flag):** 1 ( có tràn dấu)

A screenshot of a computer

Description automatically generated

b) AX = FE12h, BX = 1ACBh

**Kết quả:** AX = FE12h + 1ACBh = 18ADDh

**Cờ tràn:**

**CF (Carry Flag):** 1 (có carry ra ngoài bit cao nhất, kết quả vượt quá 16 bit, chỉ giữ lại 8DDh)

**OF (Overflow Flag):** 0 (không có tràn dấu, kết quả là số dương)

A screenshot of a computer

Description automatically generated

c) AX = E1E4h, BX = DAB3h

**Kết quả:** AX = E1E4h + DAB3h = 1BC97h

**Cờ tràn:**

**CF (Carry Flag):** 1 (có carry ra ngoài bit cao nhất, chỉ giữ lại BC97h)

**OF (Overflow Flag):** 0 (không tràn dấu)

A screenshot of a computer

Description automatically generated

d) AX = 7132h, BX = 7000h

**Kết quả:** AX = 7132h + 7000h = E132h

**Cờ tràn:**

**CF (Carry Flag):** 0 (không có carry ra ngoài bit cao nhất)

**OF (Overflow Flag):** 1 (có tràn dấu, kết quả là số âm mặc dù cả hai giá trị ban đầu đều là số dương)

A screenshot of a computer

Description automatically generated

e) AX = 6389h, BX = 1176h

**Kết quả:** AX = 6389h + 1176h = 74FFh

**Cờ tràn:**

**CF (Carry Flag):** 0 (không có carry ra ngoài bit cao nhất)

**OF (Overflow Flag):** 0 (không có tràn dấu, kết quả là số dương)

A screenshot of a computer

Description automatically generated

**3) Cho biết kết quả sau khi thực hiện lệnh SUB AX, BX và trạng thái cờ tràn (có hoặc không dấu) có xảy ra không với:**

a) AX = 2143h, BX = 1986h

**Kết quả:** AX = 2143h - 1986h = 07BDh

**Cờ tràn:**

**CF (Carry Flag):** 0 (không có mượn)

**OF (Overflow Flag):** 0 (không có tràn dấu, kết quả là số dương)

A screenshot of a computer

Description automatically generated

b) AX = 81FEh, BX = 1986h

**Kết quả:** AX = 81FEh - 1986h = 6887h

**Cờ tràn:**

**CF (Carry Flag):** 0 (không có mượn)

**OF (Overflow Flag):** 1 (có tràn dấu, vì số âm trừ đi số dương cho kết quả là số dương)

A screenshot of a computer

Description automatically generated

c) AX = 19BCh, BX = 81FEh

**Kết quả:** AX = 19BCh - 81FEh = 97BEh (do mượn 1 từ bit cao nhất)

**Cờ tràn:**

**CF (Carry Flag):** 1 (có mượn)

**OF (Overflow Flag):** 1 (có tràn dấu, vì số dương trừ số âm cho kết quả là số âm)

A screenshot of a computer

Description automatically generated

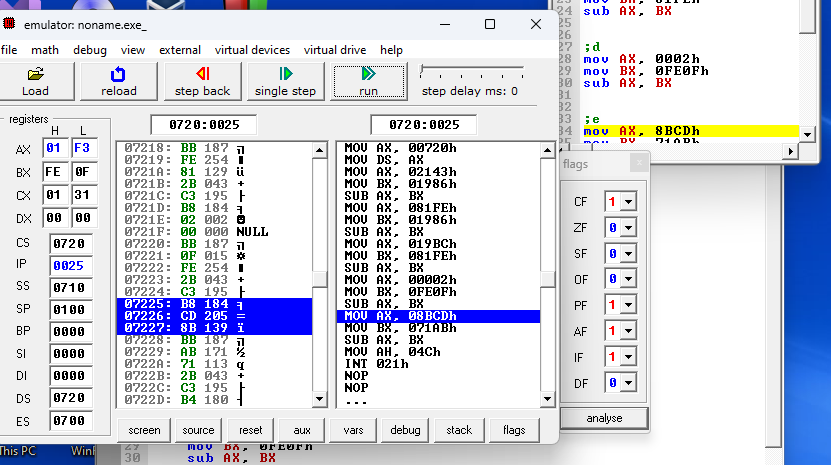
d) AX = 0002h, BX = FE0Fh

**Kết quả:** AX = 0002h - FE0Fh = 0203h (do mượn 1 từ bit cao nhất)

**Cờ tràn:**

**CF (Carry Flag):** 1 (có mượn)

**OF (Overflow Flag):** 0 (không có tràn dấu, kết quả là số dương)



e) AX = 8BCDh, BX = 71Abh

**Kết quả:** AX = 8BCDh - 71ABh = 1A22h

**Cờ tràn:**

**CF (Carry Flag):** 0 (không có mượn)

**OF (Overflow Flag):** 1 (có tràn dấu, vì số âm trừ số dương cho kết quả là số dương)

A screenshot of a computer program

Description automatically generated