① Subtraction of two 16bit numbers

A = AH AL B = BH BL

AH: high byte, AL: low byte
BH: low byte, BH: high byte

AL + BL = 16bits

A − B → AH AL − BH BL = CH CL

1) First do AL − BL subtraction

if borrow is needed (Borrow | Carry Flag in status register is 0 if borrow is needed)

2) Second perform AH − BH

Subtraction operation.

(2)

Ex8

A = 0x1206

B = 0x0814

AHAL
BHBL
= CHCL

Write a program which finds A-B result. display the result at PORTB

Sln:

```
list p=16f84A
include "p16f84A.INC"
clrf PORTB;
bsf STATUS,RP0;
clrf TRISB;
bcf STATUS,RP0;

AL   equ 0x0C;
AH   equ 0x0D;
BL   equ 0x0E;
BH   equ 0x0F;

movlw 0x06;
movwf AL;
movlw 0x12;
movwf AH;
movlw 0x14;
movwf 0x08;
```

} load the numbers to register locations

(3)

Subtraction_operation

```
movf  BL,W ;  W ← BL
Subwf AL,F ;  (AL) = (AL) - W
                          (BL)
btfss STATUS,C ; check carry flag for 0
decf  AH,F ; if C=0  AH = AH-1
                     due to borrow

movf  BH,W ;  (W) ← BH
Subwf AH,W ;  (AH) = (AH) - (W)
                            (BH)
```

display-result

```
movf  AL,W ;
movwf PORTB ;   → show CL byte of result
call delay_two_sec ;   → delay subroutine
movf  AH,W                available in
movwf PORTB ;             header file
loop  goto loop           (assume)
```

---

display port can be controlled by RAL of PORTA for this program PORTA at input at the beginning and at display port write

```
display_result
    movf  AL,W
    movwf PORTB
```

(4)

```
test_RAL    btfsc   PORTA,RAL;
            goto    test_RAL
            movf    AH, W
            movf    PORTB;
loop        goto    loop;
            end
```

low byte of result is showed and for the display of high byte of result RAL press is worked