National University
Of Computer and Emerging Sciences

EE2003 – Computer Organization and Assembly Language (Sp'24) Mar 2024

Assignment: 01, **Weight:** 3.0, Due Date: 2 Jun, **CLO:** 1

Teacher: Dr Muhammad Usman Abbasi

Name: Tazmeen Afroz **Roll No:** P22-9252 **Section:** BAI-4A

Note:

Plagiarism will be marked zero straight away to all parties involved.

Question 2 requires an explanation, which should be kept short and to the point.

Late submissions are not allowed.

Question No. 1:

Write a program in assembly language for each of the below separately that sets the following flags.

(write four programs i.e. One for each part)

a) Zero Flag

b) Carry Flag

c) Parity Flag

d) Auxiliary Flag

a) Zero Flag

```
[org 0x0100]
mov al, [num1]          ; move the value at memory location 'num1' (which is 255) to al
mov bl, [num1+1]        ; move the value at memory location 'num1+1' (which is 1) to bl
add al, bl               ; add bl (1) to al (255), result is 256 in 9-bit (1 0000 0000),
                         ; lower 8 bits are 0 (0000 0000), setting al to 0,
                         ; zero flag (ZF) is set because the result is 0
mov ax, 0x4c00
int 0x21
num1 db 255,1 ; define num1 as 255 and the next byte as 1
```

b) Carry Flag

```
[org 0x0100]
mov al, [num1]          ; move the value at memory location 'num1' (which is 2) to al
mov bl, [num1 + 1]      ; move the value at memory location 'num1 + 1' (which is 8) to bl
sub al, bl              ; subtract bl (8) from al (2), result is -6 in 2's complement (1111 1010)
                        ; carry flag is set because 2 < 8, indicating a borrow
mov ax, 0x4c00
int 0x21
num1 : db 2,8           ; define num1 as 2 and the next byte as 8
```

c) Parity Flag

```
[org 0x0100]
mov al, [num1]          ; move the value at memory location 'num1' (which is 2) to al
mov bl, [num2]          ; move the value at memory location 'num2' (which is 3) to bl
add al, bl              ; add bl (3) to al (2), result is 5
                        ; 5 in binary is 101, which has an even number of set bits,
                        ; so the parity flag (PF) is set to 1
mov [result], al        ; store the result (5) in memory location 'result'
mov ax, 0x4c00
int 0x21
num1 db 2               ; define num1 as 2
num2 db 3               ; define num2 as 3
result db 0             ; initialize result as 0
```

d) Auxiliary Flag

```
[org 0x0100]
mov ax, [num1]        ; move the value at memory location 'num1' (which is 15) to ax
mov bx, [num2]        ; move the value at memory location 'num2' (which is 5) to bx
add ax, bx            ; add bx (5) to ax (15), result is 20
                      ; Auxiliary flag (AF) is set if there is a carry from bit 3 to bit 4
                      ; 15 in binary is 0000 1111
                      ; 5 in binary is 0000 0101
                      ; Adding these results in 0001 0100
                      ; When adding the lower nibbles:
                      ; 1111 (lower nibble of 15)
                      ; + 0101 (lower nibble of 5)
                      ; ----------
                      ; 1 0100 (5 bits result, with a carry from bit 3 to bit 4)
                      ; The carry from the 4th bit (bit 3) to the 5th bit (bit 4) sets the AF
mov [result], ax      ; store the result (20) in memory location 'result'
mov ax, 0x4c00
int 0x21
num1 dw 15 ; define num1 as 15
num2 dw 5 ; define num2 as 5
result dw 0 ; initialize result as 0
```

Question No. 2:

What will be the size of the following assembly language program in bytes? Explain your answer

using ".lst" file of this code.

```
[org 0x0100]
mov ax, 5
mov bx, 10
add ax, bx
mov bx, 15
add ax, bx
mov ax, 0x4c00
int 0x21
```

.lst File

```
 1 [org 0x0100]
 2 00000000 B80500                mov ax, 5           ; 3 bytes (instruction: B8 05 00)
 3 00000003 BB0A00                mov bx, 10          ; 3 bytes (instruction: BB 0A 00)
 4 00000006 01D8                  add ax, bx          ; 2 bytes (instruction: 01 D8)
 5 00000008 BB0F00                mov bx, 15          ; 3 bytes (instruction: BB 0F 00)
 6 0000000B 01D8                  add ax, bx          ; 2 bytes (instruction: 01 D8)
 7 0000000D B8004C                mov ax, 0x4c00      ; 3 bytes (instruction: B8 00 4C)
 8 00000010 CD21                  int 0x21            ; 2 bytes (instruction: CD 21)
```

Now, summing up the sizes of each instruction:

3 + 3 + 2 + 3 + 2 + 3 + 2 = 18 bytes

Therefore, the size of the assembly language program is 18 bytes.

Question No. 3:

Calculate the physical memory address generated by the following segment-offset pairs:

1DDD:0436

1234:7920

74F0:2123

0000:6727

FFFF:4336

1080:0100

1DDD:0436 = 1DDD0 + 00436 = **1E206**

1234:7920 = 12340 + 07920 = **19C60**

74F0:2123 = 74F00 + 02123 = **77023**

0000:6727 = 00000 + 06727 = **06727**

FFFF:4336 = FFFF0 + 04336 = **104326**

1080:0100 = 10800 + 00100 = **10900**