University of Central Punjab

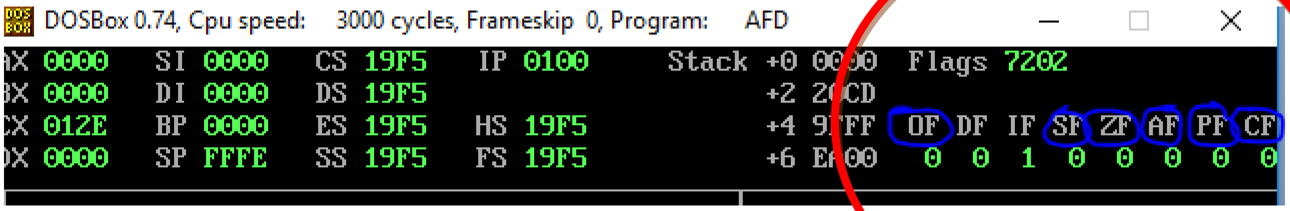
(Incorporated by Ordinance No. XXIV of 2002 promulgated by Government of the Punjab)

FACULTY OF INFORMATION TECHNOLOGY

Computer Organization and Assembly Language

Lab 4	
Topic	1. Mov instruction 2. Add,sub 3. Memory Addressing modes 4. Flag register

PART 1



```
DOSBox 0.74, Cpu speed:    3000 cycles, Frameskip  0, Program:    AFD
AX 0000   SI 0000   CS 19F5   IP 0100      Stack +0 0000  Flags 7202
BX 0000   DI 0000   DS 19F5                      +2 20CD
CX 012E   BP 0000   ES 19F5   HS 19F5            +4 9FFF  OF DF IF SF ZF AF PF CF
DX 0000   SP FFFE   SS 19F5   FS 19F5            +6 EA00   0  0  1  0  0  0  0  0
```

1. Carry Flag
2. Auxiliary Carry
3. Zero Flag
4. Sign Flag
5. Overflow Flag

Note: Flag register is updated only when any arithmetic or logical operation is performed otherwise it will maintain its state.

C	Carry	When two 16bit numbers are added the answer can be 17 bits long or when two 8bit numbers are added the answer can be 9 bits long. This extra bit that won't fit in the target register is placed in the carry flag where it can be used and tested.
A	Auxiliary Carry	A number in base 16 is called a hex number and can be represented by 4 bits. The collection of 4 bits is called a nibble. During addition or subtraction if a carry goes from one nibble to the next this flag is set. Carry flag is for the carry from the whole addition while auxiliary carry is the carry from the first nibble to the second.
Z	Zero Flag	The Zero flag is set if the last mathematical or logical instruction has produced a zero in its destination.
O	Overflow Flag	The overflow flag is set during signed arithmetic, e.g. addition or subtraction, when the sign of the destination changes unexpectedly. The actual process sets the overflow flag whenever the carry into the MSB is different from the carry out of the MSB
S	Sign Flag	A signed number is represented in its two's complement form in the computer. The most significant bit (MSB) of a negative number in this representation is 1 and for a positive number it is zero. The sign bit of the last mathematical or logical operation's destination is copied into the sign flag.

1. **Sample instructions which set multiple flags at the same time. Execute them one by one and verify the behaviour of these flags.**

		CF	AF	ZF	OF	SF
	Mov al,200 Add al,57	☑	☑			
	Mov bx,0x00AB Mov cx,0x0012 Sub cx,bx	☑	☑			☑
	Mov bx,0x00AB Mov cx,0x0012 Sub cl,bl	☑	☑			
	Mov al,0xff Add al,1	☑	☑	☑		
	Mov al,0x8f Add al,0x87	☑	☑		☑	
	mov cx,-1 mov dx,1 add cx,dx	☑	☑	☑		
	Mov ah,40 Mov al,40 Sub al,ah			☑		
	Mov al,0x7f Add al,0x77		☑		☑	☑
	mov al,0x7f mov bl,0x7f add al,bl		☑		☑	☑
	mov al,0xA3 mov bl,0xC3 add al,bl	☑			☑	

Q2: Execute the following sample code and observe the values of CF,AF,ZF,SF and OF at every instruction.

```
[org 0x100]
mov ax,0
mov bx,0
mov cx,0
mov dx,0

mov bx,my_array1
mov si,0
mov di,my_array2

mov al,[bx+si]                    ;base+index register indirect addressing mode
add [di],al                       ;index register indirect addressing mode

add si,1
add di,1
mov al,[bx+si]
add [di],al

add si,1
add di,1
mov al,[bx+si]
add [di],al

add si,1
add di,1
mov al,[bx+si]
add [di],al

add si,1
add di,1
mov al,[bx+si]
add [di],al


mov ax,0x4c00
int 21h
my_array1: db 150,120,0x6,-4,20
my_array2: db 250,34,4,5,6
```