


Philadelphia University Faculty of Engineering Department of Computer Engineering		Date:- 6/1/2014 Allowed time:- 60 minutes
Microprocessors Second Exam		
Student Name: -ID: -		

Q1). Mark the correct answer of the followings. 5 marks

1- Assume ARR is an array of double words which of the following will increment the third element in the array

- a). inc DWORD PTR [ARR+2] b). inc DWORD PTR [ARR+ 4]
c). inc DWORD PTR [ARR+6] **d). inc DWORD PTR [ARR+8]**

2- the value of eax register after executing the following code is

```
MOV eax,0A2FEh  
ADD eax,2
```

- a). 0A2FFh b).0A200h **c).0A300h** d). 0A3FEh

3- the values of Sign flag and Overflow flag after executing the following code are

```
MOV AX,8000h  
ADD AX,1
```

- a). SF=0,OF=0 b). SF=0,OF=1
c). SF=1,OF=0 d).SF=1,OF=1

4- LOOPNZ instruction repeats when.

- a). ecx=0,ZF=0 b).ecx=0,ZF=1 **c).ecx≠0,ZF=0** d).ecx≠0,ZF=1

5- to clear a specific bit(s) in a number we can use.

- a.)AND** b.)OR c.)NOT d.) XOR

Q2). Write the values of the Carry, Sign, Zero and Overflow flags after executing the following instructions 6 marks

MOV AX,7FF0h

ADD AL,10h CF=1 SF=0 ZF=1 OF=0

ADD AH,1 CF=0 SF=1 ZF=0 OF=0

ADD AX,2 CF=0 SF=1 ZF=0 OF=0

Q3). Write the instructions that perform the following operations 3 marks

1- Jump to label L1 if the unsigned integer in BX is greater than or equal the integer in CX

CMP BX,CX

JAE L1

2- Clear bits 3 and 4 in AL register then if the result is Zero jump to L2, otherwise jump to L3

AND AL,E7

JZ L2

JMP L3

3- Clear the lower half of DX register and do not change the upper half.

AND DX,F0

Q4). Write a program that define a 2 arrays of 10 integers for each, initialize one of them by number of your choice. Your program should add 1 to the absolute value of the elements in the first array and store the results in the second array. 6 marks

```
.data
Array1    WORD    4,5,-3,8,9,-1,10,60,50,-20 ; OR ANY NUMBERS
Array2    WORD    10 DUP(?)

.code
PROC      MAIN
MOV       ESI , 0
MOV       ECX , LENGTHOF Array1
TOP:      MOV       AX , Array1[ESI]
          CMP       AX , 0
          JGE      POSITIVE
          NEG       AX
POSITIVE: ADD       AX,1
          MOV       Array2[ESI],AX
          ADD       ESI,2
          LOOP     TOP
          EXIT
MAIN      ENDP
END       MAIN
```