FAST- National University of Computer & Emerging Sciences, Karachi.
Department of Computer Science,
Mid Term I Examinations, Fall 201.
25th September, 2019, 1:00 pm – 2:00 pm

Course Code: EE 213	**Course Name:** Computer Organization and Assembly Language
Instructors: Muhammad Danish Khan, Dr. Nouman M Durrani and Zail-ul-Hassan	
Student's Roll No:	**Section:**

Instructions:

- Except your Roll No and Section, DO NOT SOLVE anything on this paper.
- Return the question paper.
- Read each question completely before answering it. There are **3 questions on 2 pages.**
- In case of any ambiguity, you may make assumption. But your assumption should not contradict any statement in the question paper.
- All the answers must be solved according to the SEQUENCE given in the question paper, otherwise points will be deducted.
- This paper is subjective.
- Where asked for values, only provide the **hex-decimal** values.
- Problems needing iterations should be coded using iterative instructions. No points will be awarded otherwise.

Time Allowed: 60 minutes. **Maximum Points**: 30 points

==========

Q No. 1 Briefly answer each of the following: [6 x 2 = 12 points]

(i) Explain why memory access takes more machine cycles than register access?

(ii) Discuss the instruction execution cycle of the following instruction:

```
00920080    MOV EBX, OFFSET X1    ; X1 is a Data Label
00920080    …
```

(iii) How address, data, and control busses are used during the instruction execution?

(iv) After a program has been loaded into memory, how does it begin execution?

(v) Discuss why overflow flag is set when two positive operands are added and their sum is negative.

(vi) How a data label identifies location of a variable? Explain with examples.

(vii) A label placed just before an instruction implies the instruction's address. How code labels are used as targets of jumping and looping instructions?

(viii) Give one example instruction for each of the following addressing modes:

a. Indirect addressing mode

b. Base indexed

Q No. 2 (i) Give the contents of the status flags C, O, S and Z and the content of destination register after the execution of each of the following sequence of instructions:

```
x1  word 8F7AH, 7AF8H                                          [02 points]
x2  sword FFFFH

a.    MOV   AX, x1
      ADD   AX, [x1+2]          O = ____   C = ___   S = ____ Z = ____
b.    MOV   ESI, OFFSET x2
      Mov   BX, [ESI]
      SUB   BX, 1               O = ____   C = ___   S = ____ Z = ____
```

(ii) Consider the following data definition directives: [2 + 4 = 6 points]

```
.DATA

     . . .  (some declarations not shown)

     X1    BYTE  41h,42h,43h,17h, 0A9, 3 DUP(?)
           WORD  2 DUP(12h,13h,14h)
     X2    DWORD 1234h,5678h
     X3    QWORD 0A987654321h
```

a) Assign proper physical addresses to each byte of the word array stored in the above data segment (Assume DS= 2AA0h and the first element of X1 is at offset 23E4h).

b) Consider the above data segment. Give the content of the destination register after the execution of the following instructions:

```
MOV   EAX, DWORD PTR [X1+1]
ADD   AH, TYPE var1
XCHG  AH, AL                        ; EAX: ________________

MOV ESI, OFFSET var1 + 0Ch
MOV EAX, LENGTHOF var2
ADD EAX, [ESI+4]                    ; EAX: ________________
```

Q. No. 3 (i) Write assembly language code that directly exchanges respective elements of two byte sized arrays X1 and X2 having 10 elements each. Your code should not use a third array. [05 points]

(ii) Write an assembly language program that declares three integer arrays containing 100 elements each. Consider *bArray* as a byte array, *wArray* as a word array and *dArray* as a double word array. Let dArray holds the sum of the respective elements of *bArray* and *wArray* as follows:

dArray[i] = bArray[i] + wArray[i] [05 points]

STAY BRIGHT