# FAST- National University of Computer & Emerging Sciences, Karachi.
Department of Computer Science,
Mid Term I Examinations, Fall 2018.
3rd October, 2018, 1:00 pm – 2:00 pm

| | |
|---|---|
| **Course Code:** EE 213 | **Course Name:** Computer Organization and Assembly Language |
| **Instructors:** Nadeem Kafi Khan, Dr. Nouman M Durrani and Muhammad Danish Khan | |
| **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?

Answer: *because Register are located within the CPU, no addressing is involved hence no need to fetch data through busses, access delay is minimal which is not the case when accessing memory.*

(ii) What are the basic steps in the instruction execution cycle?

Answer: *Instruction Fetch, Decode, Operands Fetching, Execution, Write Results*

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

Answer: *The ALU starts executing the instructions in a specified order in the code segment.*

(iv) How do you differentiate a data label from a code label?

Answer: *Data labels are names of variables, pointing somewhere in data segment whereas code label specifies some point in code part*.

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

Answer: *They are used to access memory (data operands or instructions). Address bus generates a 20 bit address and control bus facilitate reading or writing control. Data is transferred to/from memory on data bus*

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

a. Register indirect

Answer: ***MOV EAX, [EBX]****; EBX contains a valid non-zero memory address*

b. Base indexed

Answer: ***MOV EAX, array[ESI]***

Q No. 2

(i) Give the contents of the status flags C, S and Z and the content of destination register after the execution of each of the following sequence of instructions:

[02 points]

```
x1  sword 8F7AH, 7AF8H
x2  word 0000H
a.     MOV   AX, x1
       ADD   AX, [x1+2]          C = 1 S = 0 Z = 0
b.     MOV   ESI, OFFSET x2
       MOV   BX, [ESI]
       SUB   BX, 1               C = 1 S = 1 Z = 0
```

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

```
.DATA
     . . .  (some declarations not shown)
     var1  BYTE  2 DUP(41h,42h,43h),3 DUP(?)
           WORD  2 DUP(12h,13h,14h)
     var2  DWORD 1234h,5678h
     var3  QWORD 0A987654321h
```

a) Assign proper physical addresses to each byte of the word array stored in the above data segment (Assume DS= 2CAEh and the first element of var1 at offset 2366h).

| | | | | | |
|---|---|---|---|---|---|
| **236Fh** | 12 | **2373h** | 14 | **2377h** | 13 |
| **2370h** | 00 | **2374h** | 00 | **2378h** | 00 |
| **2371h** | 13 | **2375h** | 12 | **2379h** | 14 |
| **2372h** | 00 | **2376h** | 00 | **237Ah** | 00 |

If calculating Physical Address first; it will be

DS ⇐ 0 (Implied Zero) + Offset

2CAE ⇐ 0
= 2CAE0 + 2366
= 2EE46

| | | | | | |
|---|---|---|---|---|---|
| **2EE50h** | 12 | **2EE54h** | 14 | **2EE58h** | 13 |
| **2EE51h** | 00 | **2EE55h** | 00 | **2EE59h** | 00 |
| **2EE52h** | 13 | **2EE56h** | 12 | **2EE5Ah** | 14 |
| **2EE53h** | 00 | **2EE57h** | 00 | **2EE5Bh** | 00 |

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

```
MOV   EAX, DWORD PTR [var3+2]
ADD   AH, TYPE var1
XCHG  AH, AL                       ; EAX: 00A96588h
```

```
MOV ESI, OFFSET var1 + 0Ch
MOV EAX, LENGTHOF var2
ADD EAX, [ESI+4]                ; EAX: 14001302h
```

Q. No. 3 (i) Write assembly language code that directly exchanges respective elements of two word sized arrays X1 and X2 having 20 elements each. Your code should not use a third array. [05 points]

```
MOV ECX, LENGTHOF X1
MOV ESI, OFFSET X1
MOV EDI, OFFSET X2
L1:   MOV AX, WORD PTR [ESI]
      MOV DX, WORD PTR [EDI]
      MOV [ESI],DX
      MOV [EDI], AX
      ADD ESI,2
      ADD EDI,2
LOOP L1
```

(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. Also, define another array xArray, where each element holds the sum of the respective elements of *bArray*, *wArray*, and *dArray*, as follows:

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

```
MOV ECX, LENGTHOF xArray - 1
MOV ESI,0
L1:   MOVZX AX, bArray[ESI*TYPE bArray]
      ADD   AX, wArray[ESI*TYPE wArray]

      MOVZX EDX, AX
      ADD   EDX, dArray[ESI*TYPE dArray]

      MOV   xArray[ESI*TYPE xArray], EDX
      INC   ESI
LOOP L1
```

**STAY BRIGHT**