FAST- National University of Computer & Emerging Sciences, Karachi
School of Computing,
Assignment # 01 [CLO: 01]
Computer Organization and Assembly Language

Date : 10-09-2025

Submission deadline: 17-09-2025

Instructions for Submission:

1. Soft copy only
2. You are required to Submit Assignment in softcopy on Google classroom.
3. Strictly follow the deadline

Q1: (a) Draw a memory map of the 8086 real-address mode 1 MB memory space. Clearly label the Code Segment (CS), Data Segment (DS), Stack Segment (SS), and Extra Segment (ES). Explain how a physical address (linear address) is calculated from a segment:offset pair.

b) Compute the physical address (in hex) for each segment:offset pair below. Indicate whether wrap-around occurs. Show your work.
(i) 1234:5678
(ii) FFFF:000F
(iii) F000:FFF0
(iv) F800:9000

Q2 Explain the role of the following x86 registers. For each register give at least one example of a typical use-case:

- General purpose: EAX, EBX, ECX, EDX
- Index/base/stack/instruction: ESI, EDI, ESP, EBP, EIP

Q3 Store the 32-bit hexadecimal number 0x1234ABCD into memory starting at address 4000h.

- Write the contents of individual addresses if the system is **little-endian**.
- Write the contents of individual addresses if the system is **big-endian**.

Q4 For each sequence below, give the contents of the status flags CF (Carry Flag), SF (Sign Flag), and the contents of the destination register after executing the instructions. Show intermediate results.

```
MOV AX, 8F7AH          ; Ax = ?, CF = ?, SF = ?
ADD AX, 7AF8H          ; Ax = ?, CF = ?, SF = ?
MOV BX, 0FA77H         ; Bx = ?, CF = ?, SF = ?
INC BX                 ; Bx = ?, CF = ?, SF = ?
```

Q5(i) Define symbolic constants:

```
Seconds EQU 60
Minutes EQU 60
Hours   EQU 24
Days    EQU 7
```

(ii) Calculate the number of seconds in a week using registers (show instruction sequence).

(iii) Save the result in memory and display it.

Q6 Write an assembly program to rotate the elements of a **word** array (size = 10) one position to the left. The first element should move to the end of the array. Use loops and addressing modes only (no high-level constructs). Include:

- Data declarations
- Code with comments explaining each step
- An example initial array and the array contents after rotation

Q7 A sensor sends a 32-bit reading 0xA1B2C3D4 to your x86 system, starting at memory address 3000h. Assuming the system is little-endian, answer:
(a) Show how this value will be stored in memory (byte-by-byte).
(b) If the system were big-endian, how would the memory layout change?
(c) How can you read this value into EAX using a single MOV instruction?

Q8 Explain why general-purpose registers (EAX, EBX, ECX, EDX) can be split into smaller registers (e.g., AX, AH, AL). Give a situation where using a smaller register is more efficient than using a full 32-bit register.

Q9 Write a short assembly program that calculates the expression

```
A = (A + B) – (C + D)
```

using registers. Assign integer values to EAX, EBX, ECX, EDX and show the steps and final result.

Q10 (Short Answer Questions)

1. What is meant by a one-to-many and one-to-one relationship? Explain with examples.
2. Explain why memory access takes more machine cycles than register access.
3. Briefly explain what is meant by Machine-dependent and Machine-independent languages. Give examples.
4. Briefly explain the difference between Computer Organization and Computer Architecture.
5. Which flag is set when the result of an unsigned arithmetic operation is too large to fit into the destination?
6. How is a source file different from a listing file?
7. Why does reading from memory usually take longer than reading from registers?
8. Differentiate between Real Mode and Protected Mode in x86 processors. Why can’t modern operating systems (e.g., Windows) run entirely in Real Mode?
9. Which components are essential for the CPU to communicate with memory, and how does the bus system help in this process?