



Capital University of Science and Technology

Department of Computer Science

CS2523 – Computer Organization and Assembly Language

ASSIGNMENT NO. 3: Write, Assemble, Debug, and Execute Assembly Code, Instruction Groups

CLO: 2. Describe how the basic units of the Intel 8088 architecture work together to represent Integer Numbers, Floating Numbers and register representation inside the microprocessor. [C2- Understanding]

CLO: 3. Implement assembly programs of intermediate complexity using the intel 8088 architecture. The student should also be able to convert intermediate complexity program in high level language into assembly code. [C3- Applying]

Semester: Summer 22

Max Marks: 10

Instructor: Ms. Tayyaba Zaheer

Assigned Date: August 29, 2022

Due Date: September 01, 2022

Name:

Reg. No.

Guidelines:

You are required to submit the **screenshots of code and output of the program (where required) and concepts in your own words i.e. must be hand written** in the assignment file (word or pdf – pictures attached must be readable and in portrait mode) as **courseCode_studentReg#_studentName** via Microsoft Teams.

Important Note:

- 1) Must not copy from other students, so do it all yourself.
- 2) Assignment should be hand written.

Description:

Emu8086 is an 8086-microprocessor emulator and disassembler. Emu8086 permits to assemble, emulate and debug 8086 programs (16bit/DOS).

Tasks: **[Hint: you can take help from lectures]**

Task#1: Assembly Language Programming: Write down the states of RAM and registers, in line by line Debugging Mode, in the following given scenarios. **(06 marks)**

Question#1:

[org 0x100]

mov ax, 3

add ax, 1

mov ax, 2

mov bx, 3

sub bx, ax

add ax, bx

mov ax, 0x4c00

int 0x21

Solution:

1. Ax = 3
2. Ax = 4
3. Ax = 2
4. Ax = 2 and Bx = 3
5. Ax=2 and Bx=1
6. Ax=3 and Bx=1
7. Ax=4C00 and Bx=1
8. Ax=0 and Bx=0

Question#2:

[org 0x100]

mov ax, 2

mov bx, 1

sub ax, bx

add ax, bx

add ax, bx

mov ax, 4

mov cx, 4

mov ax, 0x4c00

int 0x21

Solution:

1. Ax = 2
2. Ax = 2 and Bx = 1
3. Ax = 1 and Bx=1
4. Ax = 2 and Bx = 1
5. Ax=3 and Bx=1
6. Ax=4 and Bx=1
7. Ax=4, Bx=1 and Cx=4
8. Ax=4C00, Bx=1 and Cx=4

9. Ax=0, Bx=0, and Cx=0

Question#3:

[org 0x100]

mov ax, 2

mov cx, 4

mov dx, 1

add cx, dx

add ax, dx

sub cx, dx

add dx, ax

mov bx, 8

mov ax, 0x4c00

int 0x21

Solution:

1. Ax = 2
2. Ax = 2 and Cx = 4
3. Ax = 2, Cx = 4 and Dx=1
4. Ax = 2, Cx = 5 and Dx=1
5. Ax = 3, Cx = 5 and Dx=1
6. Ax = 3, Cx = 4 and Dx=1
7. Ax = 3, Cx = 4 and Dx=4
8. Ax = 3, Bx=8, Cx = 4 and Dx=4
9. Ax = 4C00, Bx=8, Cx = 4 and Dx=4
10. Ax = 0, Bx=0, Cx = 0 and Dx=0

Task#2: Instruction Groups:

(04 marks)

Question: Explain and differentiate the four instruction groups of the assembly language in which instructions have been categorized i.e.

1. Data Movement Instructions
2. Arithmetic/Logic Instructions
3. Program Control Group Instructions
4. Special Group Instructions

Solution: Lecture "8IntroToAssemblyLanguageInstructionGroups"