Capital University of Science and Technology

Department of Computer Science

CS2523 – Computer Organization and Assembly Language

ASSIGNMENT NO. 4: Registers, Segmented Memory Model, Logical to Physical Address Calculation

CLO: 1. Define concepts in the design of microprocessor as state machine and designing its data path and its controller. [C1- Remembering]

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: September 02, 2022 **Due Date:** September 06, 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#_studenName** 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: Segmented Memory Model and Registers: (04 marks)

Question#1: What are the General Purpose Registers in Assembly Language?

Solution: Lecture “6BusesIntroToAssemblyLanguageGeneralPurposeRegisters”

Question#2: What are the Special Purpose Registers in Assembly Language?

Solution: Lecture “7IntroToAssemblyLanguageRegisterFlagAndIP”

Question#3: What are the Segment Registers?

Solution: Lecture “12SegmentedMemoryModelSegmentRegistersOffsetRegisters”

Question#4: Why we need Offset Registers?

Solution: Lecture “12SegmentedMemoryModelSegmentRegistersOffsetRegisters”

Task#2: Logical to Physical Address Calculation: (06 marks)

Question: Calculate the physical memory address generated by the following segment offset pairs (both are hexadecimal values).

a) 0000:0100

16-bit Segment Address: 0000
16-bit Offset Address: 0100

Solution:

Step1: Converting these to 20-bit addresses
20-bit Segment Address: 00000
20-bit Offset Address: 00100
Step2: Performing hexadecimal addition:
20-bit Physical address (in hexadecimal): 00000 + 00100 = 00100
20-bit Physical address (in binary): 0000 0000 0001 0000 0000
[Each hexadecimal digit can be represented in its 4-bit equivalent binary (as shown above)]

b) DAD1:2345

16-bit Segment Address: DAD1
16-bit Offset Address: 2345

Solution:

Step1: Converting these to 20-bit addresses
20-bit Segment Address: DAD10
20-bit Offset Address: 02345
Step2: Performing hexadecimal addition:
20-bit Physical address (in hexadecimal): DAD10 + 02345 = DD055
20-bit Physical address (in binary): 1101 1101 0000 0101 0101

c) FFFF:5432

16-bit Segment Address: FFFF
16-bit Offset Address: 5432

Solution:

Step1: Converting these to 20-bit addresses
20-bit Segment Address: FFFF0
20-bit Offset Address: 05432
Step2: Performing hexadecimal addition:
20-bit Physical address (in hexadecimal): FFFF0 + 05432 = 105422 [As it is more than 20-bits, so wrap it around, that is, discard the most significant hexadecimal digit]
20-bit Physical address (in binary): 0000 0101 0100 0010 0010