Capital University of Science and Technology

Department of Computer Science

CS2523 – Computer Organization and Assembly Language

ASSIGNMENT NO. 5: Illegal Instructions in Assembly Language, 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 07, 2022 **Due Date:** September 11, 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: 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) 0010:0000

16-bit Segment Address: 0010
16-bit Offset Address: 0000

Solution:

Step1: Converting these to 20-bit addresses
20-bit Segment Address: 00100
20-bit Offset Address: 00000

Step2: Performing hexadecimal addition:
20-bit Physical address (in hexadecimal): 00100 + 00000 = 00100
20-bit Physical address (in binary): 0000 0000 0001 0000 0000

b) 5432:FFFF

16-bit Segment Address: 5432
16-bit Offset Address: FFFF
Solution:
Step1: Converting these to 20-bit addresses
20-bit Segment Address: 54320
20-bit Offset Address: 0FFFF
Step2: Performing hexadecimal addition:
20-bit Physical address (in hexadecimal): 54320 + 0FFFF = 6431F
20-bit Physical address (in binary): 0110 0100 0011 0001 1111

c) FEFF:4241

16-bit Segment Address: FEFF
16-bit Offset Address: 4241
Solution:
Step1: Converting these to 20-bit addresses
20-bit Segment Address: FEFF0
20-bit Offset Address: 04241
Step2: Performing hexadecimal addition:
20-bit Physical address (in hexadecimal): FEFF0 + 04241 = 103231 [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 0011 0010 0011 0001

Task#2: Illegal Instructions in Assembly Language: **(04 marks)**
Question: Identify the problems in the following instructions and correct them by replacing them with one or two instruction having the same effect.

i. mov [05], [24]

Solution:
Memory to memory move is illegal in Intel architecture. The correct instructions for this operation could be

```
mov ax, [24]
mov [05],ax
```

ii. mov bx, al

Solution:
Size mismatch i.e. bx is 16-bit register and al is 8-bit register. The correct statement could be

```
mov bx, ax
```

or

```
mov bl, al
```

iii. mov ax, [si+di+100]

Solution:
Addition of two index registers in one memory access is disallowed. We can do this operation by addition of index and base register.

```
mov ax,[si+ bx+100]
```

iv. mov bx,[bs+bp+200]

Solution:

Addition of two base registers in one memory access is disallowed. We can do this operation by addition of index and base register.
mov ax,[bp+ si+100]