

COEN 311 Section W (Computer Organization and Software)

Assignment 3

Due Monday Feb 28, 2022

8086 Instructions and Simple Assembly programs

Submission instructions:

- You have to demonstrate your work for each question
- Only one **pdf** file is acceptable. Include the statement that you are submitting your original work.
- Submit through Moodle.
- Emails are not accepted. In case you face technical issue with Moodle, you must send a screenshot of the issue and email your assignment **before** the deadline.

Question 1) Assume that the state of the 8086 registers and memory just prior to the execution of each instruction is given as below:

(AX) = 0010 H
(BX) = 0020 H
(CX) = 0030 H
(DX) = 0040 H
(SI) = 0100 H
(DI) = 0200 H
(CF) = 1
(DS:100H) = 10 H
(DS:101H) = 00 H
(DS:120H) = FF H
(DS:121H) = FF H
(DS:130H) = 08 H
(DS:131H) = 00 H
(DS:150H) = 02 H
(DS:151H) = 00 H
(DS:200H) = 30 H
(DS:201H) = 00 H
(DS:210H) = 40 H
(DS:211H) = 00 H
(DS:220H) = 30 H
(DS:221H) = 00 H

Note: (DS:100H) = 10 H ; this means that the data written in the address that is obtained by combining DS register (as base address) and 100H (offset) , is 10H.

Show what result is produced in the destination operand in each of the following cases.

No need to show the flags. (Assume instructions are independent): **(32 points)**

- ADD AX, 00FF H
- ADC SI, AX
- INC BYTE PTR [0100H]
- SUB DL, BL

- e) SBB DL, [0200H]
- f) MUL DX
- g) IMUL BYTE PTR [SI]
- h) IDIV BX

NOTE: BYTE PTR [SI] is a pointer directive which assists the compiler and tells it about the size of the data which needs to be retrieved from the memory. So, in this case an 8 bit data is taken from the memory that is in an address that must be calculated using SI register.

Question 2) Write an assembly program implementing following function, where the variables x and f are positive 16-bit and 32-bit memory references, respectively. **(34 points)**

$$f = [(x-4)^2 \div x] + 8$$

Consider only the integer part of the division in the function f above.

Question 3) Convert the following C code into Intel 8086 assembly program, which reads a 32-bit number stored in memory and checks if it is a prime number or not. **(34 points)**

```
int  number;
bool isPrime = true;
for(int i=2; i <= (number / 2 ); i++ )
{
    if ((number % i ) == 0 )    // check remainder
    {
        isPrime = false;      // % is modulus
        break;
    }
}
```