Green University of Bangladesh

Department of Computer Science and Engineering(CSE)

Faculty of Sciences and Engineering

Semester: (Spring, Year: 2021), B.Sc. in CSE (Day)

LAB REPORT NO : 02

Course Title: Microprocessors and Microcontrollers Lab

Course Code: CSE 304 **Section:** PC-DD

Lab Experiment Name: Introduction of understanding advanced 8086 I/O instruction.

Student Details

Name	Id
Md. Romzan Alom	201902144

Lab Date: 25.10. 2021

Submission Date: 01.11.2021

Course Teacher's Name: Rusmita Halim Chaity

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status	
Marks: …………………………………	Signature:.....................
Comments:...	Date:.............................

TITLE OF THE LAB EXPERIMENT: Introduction of understanding advanced 8086 I/O instruction.

OBJECTIVES/AIM:

We learn some op code. They are ADD, SUB, MUL, DIV, MOV, INT 21H, R DB ? , LEA DX, R2 and MOV AH,9 and AAA. We can use that op-code to take user input and we can perform various operations on this input and show the result as output.

PROCEDURE / ANALYSIS / DESIGN:

Instruction MOV AX, BX is used moves data BX to AX and data is stored in AX.

Instruction DIV BX is used division. Here, AX is divided by BX and Quotient is stored in AL and Remainder is stored in AH.

Instruction MUL BX is used multiplication. Here, BX and AX are multiplied and this data is stored in AX.

Instruction ADD AX, BX is used addition. Here, add to BX and AX and this data is stored in AX.

Instruction SUB AX, BX is used subtraction. Here, BX and AX are subtracted and this data is stored in AX.

Instruction SUB AX, BX is used subtraction. Here, BX and AX are subtracted and this data is stored in AX.

Instruction MOV AH, 1 and INT 21H are used for user input and it stored AX.

Instruction MOV DX, AX and MOV AH, 2 and INT 21H are used for show output. Here, DX store AX data and compiler understand DX data then it show DX data as output.

Instruction LEA DX, R and MOV AH, 9 and INT 21H are used for call variable and show output the variable data.

Instruction MOV AH,2 and MOV DL,0DH and INT 21H and MOV DL,0AH and INT 21H are used for newline.

Instruction V DB 0AH, 0DH, “The result is: $" are used for newline and take a message. Here, 0AH, 0DH, is used newline.

Problem-01

Write an assembly code that will take three decimal numbers from user and print the summation as output. Like a=1, b=2, c=1 then (a + b + c) * 2 = ?

Pseudo-code:

```
.MODEL SMALL

.STACK 100H

.DATA



 N1 DB ?

 N2 DB ?

 N3 DB ?

 Sum DB ?


 R1 DB 0AH,0DH,"Enter the 1st number: $"

 R2 DB 0AH,0DH,"Enter the 2nd number: $"

 R3 DB 0AH,0DH,"Enter the 3rd number: $"

 R4 DB 0AH,0DH,"The Summation is: $"


.CODE

         MAIN PROC
```

```
MOV AX,DATA

MOV DS,AX


LEA DX,R1

MOV AH,9

INT 21H


MOV AH,1

INT 21H

SUB AL,30H



MOV CL,2

MUL CL

MOV N1,AL


LEA DX,R2

MOV AH,9

INT 21H


MOV AH,1

INT 21H

SUB AL,30H
```

```
MOV N2,AL

MUL CL

ADD AL,N1

MOV Sum,AL

LEA DX,R3
MOV AH,9
INT 21H

MOV AH,1
INT 21H
SUB AL,30H
MOV N3,AL

MUL CL

ADD AL,Sum
MOV Sum,AL

MOV AH,0
```

```
AAA
AAA

ADD AH,30H
ADD AL,30H

MOV CX,AX

LEA DX,R4
MOV AH,9
INT 21H

MOV AH,2
MOV DL,CH
INT 21H
MOV AH,2
MOV DL,CL
INT 21H

MOV AH,4CH
INT 21H

          MAIN ENDP
```

```
END MAIN
```

Problem-02

Write an assembly code to solve this corresponding expression. Take a single value as input and solve the expression and the output will also be single value (input*2+2).

Pseudo-code:

```
.MODEL SMALL
.STACK 100H
.DATA

 R1 DB "Enter your number: $"
 R2 DB 0AH,0DH,"The result is: $"


.CODE
  MAIN PROC

MOV AX,DATA
MOV DS,AX

LEA DX,R1
MOV AH,9
```

```
INT 21H



MOV AH,1

INT 21H

SUB AL,30H


MOV BL,2


MUL BL


ADD AL,BL


MOV BL,AL



MOV AX,DATA

MOV DS,AX


LEA DX,R2

MOV AH,9

INT 21H
```

```
MOV DL,BL

ADD DL,48

MOV AH,2


INT 21H


MOV AH,4CH

INT 21H


       MAIN ENDP

      END MAIN
```

Problem-03

Write an assembly code that will take five decimal numbers from user (single digit) and print the summation as output in also in single digit.

Pseudo-code:

```
.MODEL SMALL

.STACK 100H

.DATA
```

```
 R1 DB "Enter your five number to add: $"
 R2 DB 0AH,0DH,"The result is: $"


.CODE
  MAIN PROC

MOV AX,DATA
MOV DS,AX

LEA DX,R1
MOV AH,9
INT 21H


MOV AH,1 ;1
INT 21H
SUB AL,30H
MOV BL,AL

MOV AH,2
MOV DL,0DH
INT 21H
```

```
MOV DL,0AH

INT 21H


MOV AH,1   ;2

INT 21H

SUB AL,30H

ADD AL,BL


MOV BL,AL


MOV AH,2

MOV DL,0DH

INT 21H

MOV DL,0AH

INT 21H



MOV AH,1   ;3

INT 21H

SUB AL,30H

ADD AL,BL


MOV BL,AL
MOV AH,2
MOV DL,0DH
INT 21H
MOV DL,0AH
INT 21H

MOV AH,1 ;4
INT 21H
SUB AL,30H
ADD AL,BL

MOV BL,AL

MOV AH,2
MOV DL,0DH
INT 21H
MOV DL,0AH
INT 21H

MOV AH,1 ;5
INT 21H
SUB AL,30H
```

```
ADD AL,BL


MOV BL,AL


MOV AH,2

MOV DL,0DH

INT 21H

MOV DL,0AH

INT 21H



LEA DX,R2

MOV AH,9

INT 21H


MOV AH,2

MOV DL,BL

ADD DL,30H

INT 21H


; MOV AH,4CH

;INT 21H

            MAIN ENDP
```

END MAIN

Calculation:

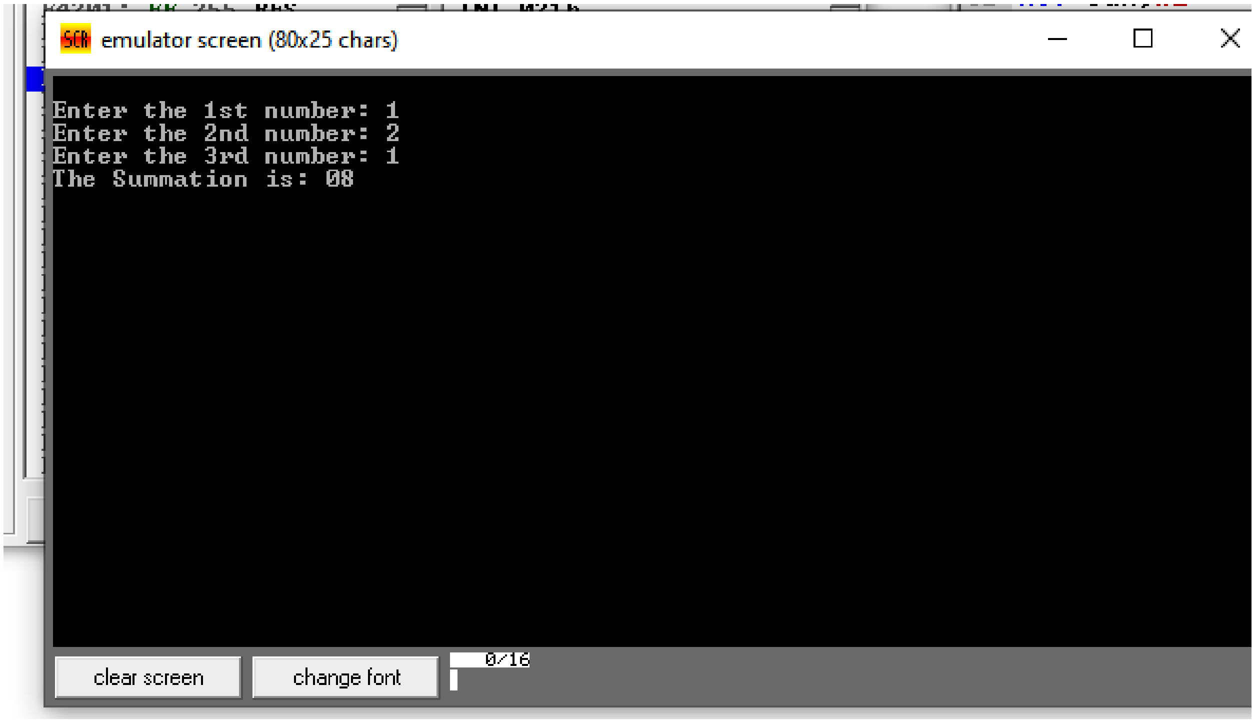
From problem 1, we will try to find the summation of three numbers by using a law. This law is Area = (a * b * c) * 2 .where a, b and c are the user input.

From problem 2, we will try to solve an corresponding expression. This expression is = (input * 2) + 2 .where input is the user input.

From problem 3, we will try to find the summation of five numbers. Here, summation = (a+ b + c + d + e). Where a, b, c, d and e are the user input.

TEST RESULT / OUTPUT:

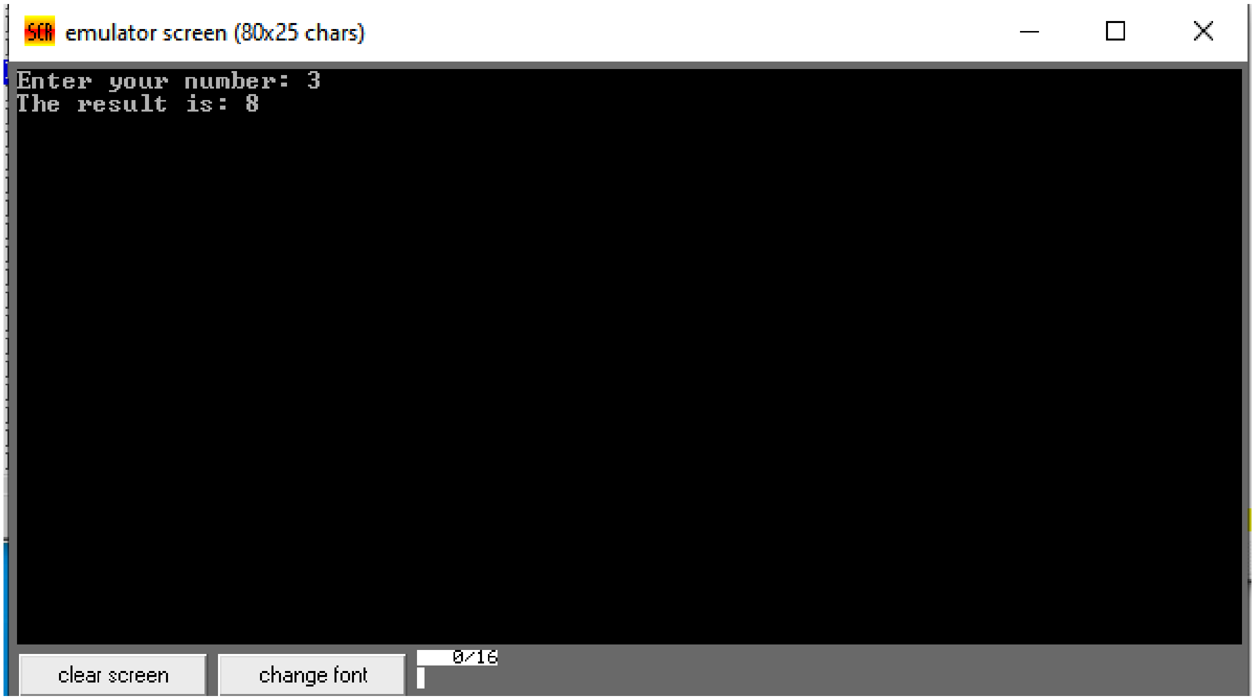
Problem-1 output:



```
Enter the 1st number: 1
Enter the 2nd number: 2
Enter the 3rd number: 1
The Summation is: 08
```

This output is 08 that is decimal value. But it should have been (1+2+1)*2 = 4 * 2= 8
It is only allow for single digit output. This reason 8 is 100% right outputs.

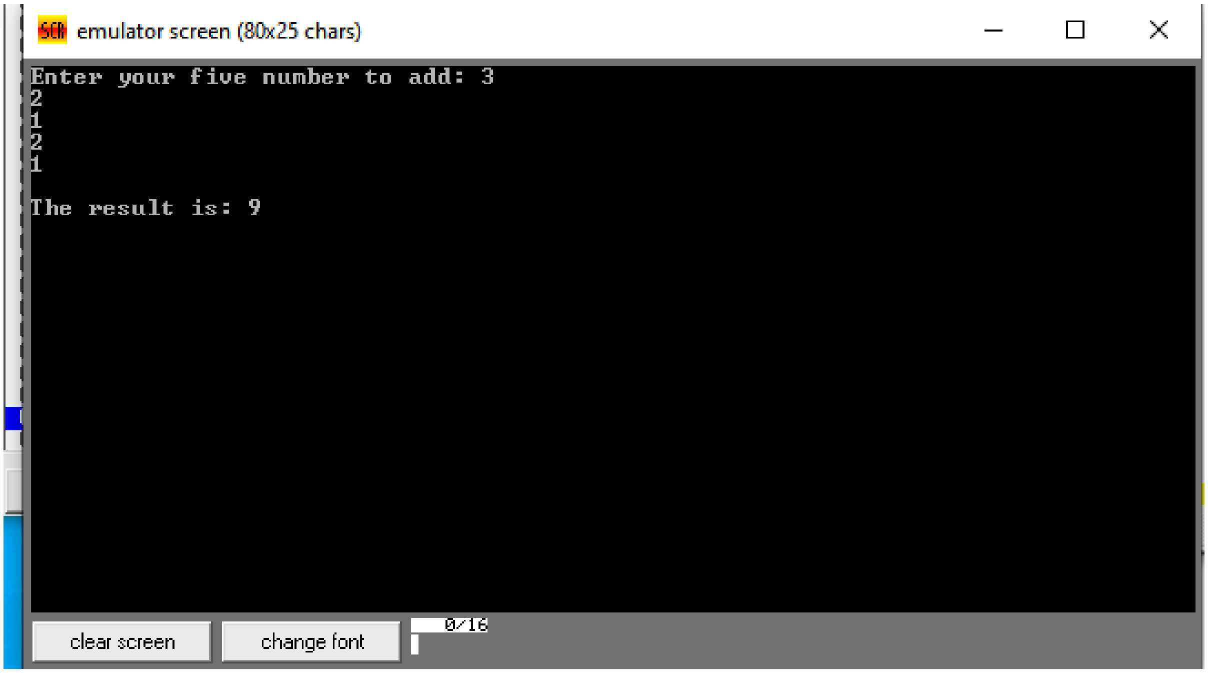
Problem-2 output:



```
Enter your number: 3
The result is: 8
```

This output is 8 and it should have been = (3 * 2 +2) = 6 + 2 = 8
It is only allow for single digit output. This reason 8 is 100% right outputs.

Problem-3 output:



```
Enter your five number to add: 3
2
1
2
1

The result is: 9
```

This output is 9 that is decimal value it should have been = (3+2+1+2+1) = 9

It is only allow for single digit output. This reason 9 is 100% right outputs.

ANALYSIS AND DISCUSSION:

1. Due to Covid-19 situation, we can't do this experiment directly. So, it is completely based on software.

2. Since, it is done with Software. So it may have some Software and Mechanical errors.

3. To emulate those codes I am facing so many problems for hexadecimal number.

4. From those three problems, those are only allowed for single digit output. If the output is above 10 then those output will valid otherwise not.

5. From problem-1, we can use AAA but our class we would not read it. That is really confusing.

6. To multiple user input and other value, we face some problem.

7. From our compile show only char value of hexadecimal number. That why we face some many problem to add two values.

8. Network problem. Cause of the ups and downs of the internet we could not attend the class properly. That why I saw the class recording but this video was very bad quality. This reason I have so many confusion.

SUMMARY:

From those three problems, we can take so many decimal numbers from user as input and print the summation or perform various operations on this input and show the result as output very easily which will help the user to change the program's output. Those are very important to complete this course.