**PRACTICAL # 03**

**OBJECT:**

Addition and subtraction arithmetic operations in assembly.

**THEORY:**

The arithmetic operations are the basic operations the CPU performs. In assembly, there are keywords for these arithmetic operations.

**Addition operation**

**Add a, b**

The above instruction adds operands **a** and **b** together and stores the result in operand **a.** The operand **a** acts both as a source 1 and destination, the operand **b** acts as source 2 only.

The operand **a** can be a **register** or **memory.**

The operand **b** can be a **register**, **memory** or an **immediate** value (constant).

The operands **a** and **b** cannot both be **memory** **locations**, this is a feature in the 8086 organization defined in the 8086 **Instruction Set Architecture**.

Another instruction for adding 1 to the current value is the opcode **inc.** inc is similar to the ++ operator in c/c++. It adds 1 to the current value.

**Subtraction operation**

**sub a, b**

The above instruction subtracts **b** from **a** and stores result in the operand **a.**

Rest of the rules are the same as for **add** operation**.**

Another instruction for subtracting 1 from the current value is the opcode **dec.** decis similar to the -- operator in c/c++. It subtracts 1 from the current value.

**Program:**

The program assumes two constant values in the registers and performs the addition operation. Finally outputs the result.

.model small

.stack 100h

.data

.code

main proc

; add two numbers

mov bh, 5

mov bl, 4

add bl, bh ; add 5 + 4 => bl = bl + bh

; get ASCII value of the resultant number

add bl, 48 ; since the ASCII value of 0 is 48, so add it to the result

; output character (the resulting number )

mov dl, bl

mov ah, 2

int 21h

mov ah, 4ch ; return control to DOS/OS

int 21h

main endp

end main

The DL register contains the value of the character to be output. The interrupt number for character output is filled in register AH. Finally the interrupt 21H is called to execute the operation.

**INC opcode**

The code for incrementing 1 to the current value is as

MOV DL, 52

INC DL

**ACTIVITIES**

**Activity 1**

Write the above program in text editor, assemble and link the program to generate executable file. Run the exe file and observe the output.

**Activity 2**

Write a similar program to perform subtraction operation. And generate exe file to observe output.

**Activity 3**

Write a similar program to perform subtraction operation, but take the single digit numbers input from user. Generate exe file to observe output.

**Activity 4**

Write the above program in text editor but using **inc** operator to add 1 to the current value, assemble and link the program to generate executable file. Run the exe file and observe the output.

**REVIEW QUESTIONS**

1. What is the instruction format for the addition and subtraction operations in assembly?
2. What will be the result if 5 is subtracted from 3 or 3 – 5?
3. What is the difference between **add** and **inc** opcodes?

1. Why do we need to add 48 in the number before printing it as in instruction add bl, 48?