KHULNA UNIVERSITY OF ENGINEERING AND TECHNOLOGY, KUET

SESSIONAL REPORT

Course No: CSE 2204

Department of: Computer Science and Engineering

Experiment No: **01**

Name of the Experiment: Developing a program that perform Addition, subtraction, multiplication and addition in assembly language

Remarks

Date of Performance: 07.04.21

Date of Submission: 12.04.21

Name: Rifat Arefin

Roll: 1807117

Year: 2nd

Semester: 2nd

No. of experiment : 1

Name of experiement : Developing a program that performs addition, subtraction, multiplication and dividation operation over two numbers in assembly language.

Objective :

1. To add two numbers
2. To subtract one number from another
3. To know how to multiply two numbers
4. To divide one number by another

Introduction :

To perform addition between two operand we have to use "ADD" Onstruction. The syntex of ADD instruction is -

ADD destination | source

Here source ~~and~~ would be a register, memory location and immediate and destination would be register and memory location but the source and destination can't be memory location in the same instruction.

For subtraction we should use "SUB" ~~instr~~ Operator in replace of "ADD" which was used for addition.

Point to be mentioned that the source and destination should be either word or byte but their size should be same.

For multiplication, we use MUL operator and there is only one operand and that is source. The destination will be either AX register or AL register.

Three types of multiplication can be performed

1. Byte by Byte.
2. Word by Word
3. Byte by Word

1 Byte by Byte:

For this type, we have to assign a value in the AL register and then multiply a register or memory which contains 8-bit data using MUL operator. The multiplication result should be in AX register.

2 Word by Word:

We have to assign a value in AX register as destination and a memory or register containing 16-bit word data. The result should be stored in DX-AX register

3. Byte by Word:

In this case the destination should be a byte data and other part of the word register should be assigned ~~as~~ with zero (0). Here the result should be stored in DX- AX register

For performing division, we have to use DIV operator.

There are 4 types of division:

1 Byte by Byte ; where the quotient should be stored in AL register and the reminder should be in AH register.

2 Word by Word: where the quotient stored in AX and reminder in DX register

3- Word by Byte; Here the quoitent is stored inside AL and the reminder is stored in AH register

4. Double Word by Word: Where the quoitent is stored in AX and reminder in DX register.

If we devide by byte, we have to take aL register as destination and if we take word type denominator then we have to take AX register as destination. We can use register or memory as source.

Apparatus Required: Emu 8086, Laptop.

Methodology:

+ Addition:

```
org 100h
mov ax, 0020h  ; ax reg. is initialized with 20h
mov bx, 0030h  ; bx register is initialized with 10h
add ax, bx     ; adding the values of two
               register.
```

output: ax : 0050 h (result of addition)

2. Subtraction:

Code:

```
org 100h

mov ax, 2000h ; ax is initialized with
                 2000h

mov, bx, 1000h; bx is initialized with
                 1000h

sub ax, bx    ; subtracting values from
                 ax by bx
```

Output:

```
ax : 1000h (result of subtraction)
```

3. Multiplication:

Code:

```
org 100h

mov al, 03h ; initialize al with 03h

mov bl, 02h; initialize bl with 02h

mul bl      ; multiplication ~~with~~ of al with
              bl register's values.
```

Output:

```
al: 0006h (result of multiplication)
```

4 Division:

Code:

```
org 100h
mov al, 06h ; initialize al with 06h
mov bl, 02h ; initialize bl with 02h
div bl; divide the value of al with the value of
        bl
```

Output:

```
ah : 00h (reminder)
al : 03h (quoitent)
```

Result and Discussion:

The experiment was performed perfectly and we got result according to our theory for addition, subtraction, multiplication and division. We tried to ensure our experiment to be perfect using various values and every time we got expected result. So eventually we can clarify that this experiment was perfect.

Conclusion: From this experiment, we learnt how to ~~additi~~ perform addition, subtraction, multiplication and division operation ~~adhi~~ in assembly language which are very important and necessory operation and basic operations too for upcoming experiments.

References:

1. Microprocessor and Interfacing - by D.V. Hall

2 emu 8086 / documentation / index . html