Lab Manual for Computer Organization and Assembly Language
Lab-4
Arithmetic Operations – Part 2

Contents

1. Introduction

The four basic arithmetic operations are addition, subtraction, multiplication and division. Apart from this we will cover increment and decrement operator as well in this lab.

1.1 Addition

Addition is the most basic operation of arithmetic. In its simplest form, addition combines two quantities into a single quantity, or *sum*. For example, say you have a group of 2 boxes and another group of 3 boxes. If you combine both groups together, you now have one group of 5 boxes. To represent this idea in mathematical terms:

$$2+3=5$$

1.2 Subtraction

Subtraction is the opposite of addition. Instead of adding quantities together, we are removing one quantity from another to find the *difference* between the two. Continuing the previous example, say you start with a group of 5 boxes. If you then remove 3 boxes from that group, you are left with 2 boxes. In mathematical terms:

$$5-3=2$$

1.2.1 Multiplication

Multiplication also combines multiple quantities into a single quantity, called the *product*. In fact, multiplication can be thought of as a consolidation of many additions. Specifically, the product of x and y is the result of x added together y times. For example, one way of counting four groups of two boxes is to add the groups together:

$$2+2+2+2=8$$

However, another way to count the boxes is to multiply the quantities:

$$2 \cdot 4 = 8$$

Note that both methods give you the same result—8—but in many cases, particularly when you have large quantities or many groups, multiplying can be much faster.

1.2.2 Division

Division is the inverse of multiplication. Rather than multiplying quantities together to result in a larger value, you are splitting a quantity into a smaller value, called the *quotient*. Again, to return to the box example, splitting up a group of 8 boxes into 4 equal groups results in 4 groups of 2 boxes:

$$8 \div 4 = 2$$

2. The Basic Arithmetic Properties

2.1 Commutative Property

The commutative property describes equations in which the *order* of the numbers involved does not affect the result. Addition and multiplication are commutative operations:

- 2+3=3+2=52+3=3+2=5
- 5·2=2·5=105·2=2·5=10

Subtraction and division, however, are not commutative.

2.2 Associative Property

The associative property describes equations in which the *grouping* of the numbers involved does not affect the result. As with the commutative property, addition and multiplication are associative operations:

- (2+3)+6=2+(3+6)=11(2+3)+6=2+(3+6)=11
- (4·1)·2=4·(1·2)=8(4·1)·2=4·(1·2)=8

Once again, subtraction and division are not associative.

2.3 Distributive Property

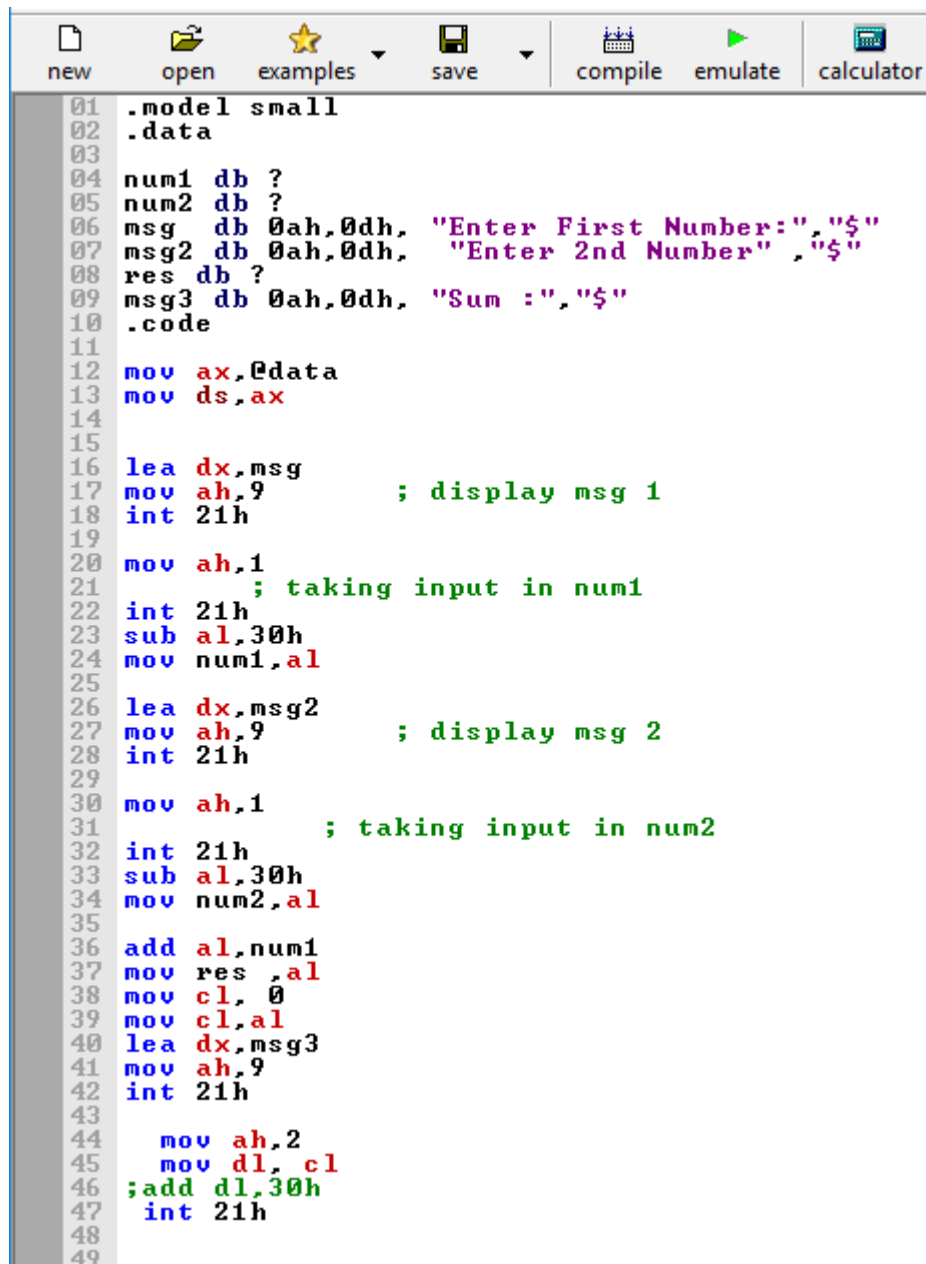The distributive property can be used when the sum of two quantities is then multiplied by a third quantity.

- (2+4)·3=2·3+4·3=18(2+4)·3=2·3+4·3=18

3. Objective

- To get basic understanding of arithmetic operation in Assembly Language.
- Arithmetic properties
- To get an understanding of number representation in ascii format.
  .

4. Walkthrough Task

```
new    open  examples  save      compile  emulate  calculator

01  .model small
02  .data
03
04  num1 db ?
05  num2 db ?
06  msg  db 0ah,0dh,  "Enter First Number:","$"
07  msg2 db 0ah,0dh,  "Enter 2nd Number" ,"$"
08  res db ?
09  msg3 db 0ah,0dh, "Sum :","$"
10  .code
11
12  mov ax,@data
13  mov ds,ax
14
15
16  lea dx,msg
17  mov ah,9        ; display msg 1
18  int 21h
19
20  mov ah,1
21              ; taking input in num1
22  int 21h
23  sub al,30h
24  mov num1,al
25
26  lea dx,msg2
27  mov ah,9        ; display msg 2
28  int 21h
29
30  mov ah,1
31                ; taking input in num2
32  int 21h
33  sub al,30h
34  mov num2,al
35
36  add al,num1
37  mov res ,al
38  mov cl, 0
39  mov cl,al
40  lea dx,msg3
41  mov ah,9
42  int 21h
43
44     mov ah,2
45     mov dl, cl
46  ;add dl,30h
47   int 21h
48
49
```

The above program shows the input of two numbers and add them and store the result in res variable

5. Procedure& Tools

Procedure and tools installation discussed in first lab (Lab01).

5.1 Tools
- Installing Microprocessor Intel 8086 emulator.

6. Practice Tasks

This section will provide more practice exercises which you need to finish during the lab. You need to finish the tasks in the required time. When you finish them, put these tasks in the following folder:
\\fs\assignments$\

6.1 Practice Task 1   [Expected time = 15mins]

Write an assembly program to solve the following equation
Res=A+B+(C+D)+E+(F+G)
Whereas the value of parameters is as follow
- A=1
- B=2
- C=3
- D=4
- E=5
- G=6

At the end the result of the equation should be stored in **Res** variable.

6.2 Practice Task 2   [Expected time = 15mins]

Write an assembly program to solve the following equation
Res=A-B-(C-D)-E-(F-G)
Whereas the value of parameters is as follow
- A=6
- B=5
- C=4
- D=3
- E=2
- G=1

At the end the result of the equation should be stored in **Res** variable.

6.3 Practice Task 3   [Expected time = 15mins]

Write an assembly program to solve the following equation
Res=A+B-(C+D)-E+(F-G)
Whereas the value of parameters is as follow
- A=3
- B=2
- C=2
- D=1
- E=1
- G=3

At the end the result of the equation should be stored in **Res** variable.

6.4 Practice Task 4   **[Expected time = 15mins]**

Write an assembly program to solve the following equation
Res=A-B-(C+D)+E-(F-G)
Whereas the value of parameters is as follow
- A=4
- B=1
- C=3
- D=2
- E=4
- G=3

At the end the result of the equation should be stored in Res variable.

6.5 Out comes

After completing this lab, student will be able to compile and run basic Assembly arithmetic operation with their associative properties.

7. Evaluation Task (Unseen)   [Expected time = 30mins for tasks]

The lab instructor will give you unseen task depending upon the progress of the class.

8. Evaluation criteria

The evaluation criteria for this lab will be based on the completion of the following tasks. Each task is assigned the marks percentage which will be evaluated by the instructor in the lab whether the student has finished the complete/partial task(s).

Table 3: Evaluation of the Lab

Sr. No.	Task No	Description	Marks
1	4	Problem Modeling	20
2	6	Procedures and Tools	10
3	7	Practice tasks and Testing	35
4	8	Evaluation Tasks (Unseen)	20
5		Comments	5
6		Good Programming Practices	10

9. Further Reading

This section provides the references to further polish your skills.

10. Slides

The slides and reading material can be accessed from the folder of the class instructor available at \\fs\lectures$