

Dr. D. Y. Patil Unitech Society  
Dr. D. Y. Patil Science & Computer Science College,  
Akurdi, Pune - 411 044

**DEPARTMENT OF ELECTRONICS**

Name : Patil Sudeej Shantaram.			REMARKS
Class : SYBSc [computer science]		Roll No. : 85      Batch :	
Experiment No. :		Performed Date :      /      /20	
Title of Experiment : Arithmetic and logical problem.			

Aim :- Arithmetic and logical problem.

1. String Addition.
2. Largest and smallest number in the string.
3. Hexadecimal to decimal.
4. decimal to Hexadecimal.

Software :- Keil.

Procedure to write and debug program.

- To enter a new program.
1. Enter Keil software.
  2. Go to project, then select new project and then save the project by giving the name to it.
  3. Select the device Atmel or Philips then select device no. AT89C51RD2 OR ATLU51
  4. Go to File and select new File, type the program and save as filename.asm.
  5. At project workspace create source group and add the particular file to source group by right clicking and selecting the option add to group.
  6. To build a link select and build option.

- To debug the program :-

1. Go to debug and select start and stop option.
2. Then run the program by selecting debug and run.
3. The contents of all registers used in the program is seen on the project workspace.

- To create Hex Files :-

1. Select project then option for target, in that select output, select "create Hex Files", OK, then rebuild the target.
2. To observe go to File select open, open the particular file with ".lst" extension.
3. This gives the information about address lines, opcodes of the instructions.

- Algorithm : Addition of multiple byte numerical string

1. Start.
2. Initialize two strings and memory location where the result is to be stored.
3. Clear carry, index and counter.
4. Read a byte from first string.
5. Read a byte from second string.
6. Add the two bytes along with carry if generated.
7. Store the result in memory location.
8. Go for the next memory location to store the result.
9. Increment index.
10. Decrement count.
11. If count not equal to zero, go to step 5.



12. else, store final carry in memory location.

13. stop.

#### Algorithm: Largest.

1. start.

2. define an Array 'A'.

3. initialize index and counter.

4. Get  $A_i$  in register.

5. compare  $A_i$  and  $A_{i+1}$ .

6. Is  $A_i \leq A_{i+1}$ , if yes, go to step 10.

7. If no, swap the bytes.

8. Increment the index.

9. Decrement counter.

10. If counter not equal to zero then go to step 5.

11. If equal to zero, store the largest number in the register.

12. stop.

#### Algorithm smallest :-

1. start.

2. define an array 'A'.

3. initialize index 'I' and counter.

4. Get  $A_i$  in register.

5. compare  $A_i$  and  $A_{i+1}$ .

6. Is  $A_i \geq A_{i+1}$ , if yes, go to step 10.

7. If no, swap the bytes.

8. Increment the index.

9. Decrement counter.

10. If counter not equal to zero then go to step 5.

11. if equal to zero, store the largest number in the register

12. stop.

• Algorithm: Hexadecimal to decimal.

1. start.

2. initialize a hexadecimal number.

3. divide the hexadecimal number by 100.

4. store the quotient as hundreds place.

5. divide the remainder by 10.

6. store the quotient of the above division as tens place.

7. store the remainder as units place.

8. stop.

• Algorithm: decimal to Hexadecimal.

1. start.

2. initialize decimal number.

3. divide decimal number by 16.

4. rotate the quotient left by 4 times.

5. logically OR the remainder with above result.

6. store the result as hexadecimal.

7. stop.

2) program for addition of two strings :-

count equ 04H

org 0200H

str1: db 0FFH, 0FFH, 0FFH, 0FFH

org 0220H

str2: db 0FFH, 0FFH, 0FFH, 0FFH

org 0000H



```

    clr C
    mov R3, #00H
    mov R1, #50H ; memory location to save the result
    clr A.
    mov R0, #count
above: mov dptr, #str1
    mov A, R3
    movc A, @A+dptr
    mov B, A
    mov dptr, #str2,
    mov R3,
    movc A, @A+dptr
    addc, A, B
    mov @R1, A
    jnc R1
    inc R3
    djnz R0, above
    inc exit
    mov @R1, #01H
exit: - nop
    end

```

2) program to find largest number

```

count equ 05H
org 0200H
num: db 45H, 20H, 56H, 10H, 15H
    org 000H
    clr A
    mov R3, #00H

```

```

mov R2, #count
dec R2
mov dptr, #num
mov A, R3
movc A, @A+dptr
mov B, A
loop: inc R3
mov A, R3
movc A, @A+dptr
jne A, B, chk
jmp ok
chk: jc ok
mov B, A
ok: djnz R2, loop
mov R1, B
end

```

Note :- for smallest the program is same as above instead of jc ok, here it will be jnc ok.

• result : 1. Hexadecimal to decimal :

Hex value A = 804, R1 = 0x01

value: R2 = 0x02

R3 = 0x08

2. decimal to Hexadecimal :

decimal value A = 128 H

Hex value = R2 = 0x80



3. program for largest number :-

array - 45H, 20H, 50H, 10H, 15H

R1 = 0X56H

4. string addition :

R1 = 0X56H

C = 0200H

C = 0X200 = 01 02 03 04

C = 0X220H

C = 0X220 :- 05 06 07 08

D = 50H

D = 0X50 :- 06 08 0A 0C

• conclusion :-

we have studied assembly language program  
execute the program verify the result we have used  
keil version to execute our program.