



Department of Computer Science Engineering Data Science

Academic Year: 2022-23
Class / Branch: S.E.D.S.

Semester: IV
Subject: Microprocessor Lab

Experiment No. 4

1. **Aim:** Write an Assembly Language Program to find LCM of two numbers.
2. **Software used:** tasm,tlink ,td,dosemu
3. **Theory :-**

3.1 Stack Operation Instructions in 8086:

It is called a stack, because you "stack" things on it. The philosophy is that you retrieve (pop) things in the opposite order of storing (push) them. Items are placed on the stack using the PUSH instruction and removed from the stack using the POP instruction. When an item is pushed onto the stack, the processor decrements the SP register, then writes the item at the new top of stack. When an item is popped off the stack, the processor reads the item from the top of stack, then increments the SP register. In this manner, the stack grows **down** in memory (towards lesser addresses) when items are pushed on the stack and shrinks **up** (towards greater addresses) when the items are popped from the stack.

In the 8086, the stack pointer is SS:SP, which is a 16 bit pointer into a 20 bit address space. It, at any point of time, points to the last item pushed on the stack. If the stack is empty, it points to the highest address of the stack plus one.

a. **PUSH and POP instruction:** These instructions are used to copy a word on top of the stack or remove the word from top of the stack in the register specified. The following table gives an example of the PUSH and the POP operation. The operand in both (PUSH and POP) instructions can be a general purpose register, segment register(except CS) or a memory location.

3.2 Syntax:

1. PUSH Operand

eg. PUSH BX

Copies the BH at SP-1 and BL at SP-2. Thus after the complete execution of PUSH instruction SP is decremented by 2, this new value (SP-2) is the new top of stack.

2. POP Operand

eg. POP CX

Copies byte from the top of stack in CL and sets SP to SP+1, copies the byte from this location to CH and sets the SP to SP+1. Thus after the complete execution of POP instruction SP is increments by 2, this new value (SP+2) is the new top of stack.

4. Program:

```
print macro msg
    lea dx,msg
    mov ah,09h
    int 21h
endm

read macro n,j1,j2
    mov cx,0ah
j1: mov ah,01h
    int 21h
    cmp al,0dh
    je j2
    sub al,30h
    mov bl,al
    mov ax,n
    mul cx
    xor bh,bh
    add ax,bx
    mov n,ax
    jmp j1
j2 :nop
endm
.model small
.stack 100h
.data
    msg1 db 10,13,'Enter the 1st number: $'
    msg2 db 10,13,'Enter the 2nd number: $'
    msg3 db 10,13,'The LCM= $'
    data1 dw 0
    data2 dw 0
```

```

    dat1 dw 0
    dat2 dw 0
.code
main proc
    mov ax,@data
    mov ds,ax

    print msg1
    ;reading 1st multidigit number
    read data1,jump1,jump2

    print msg2
    ;reading 2nd multidigit number
    read data2,jump3,jump4


    ;copy the data1 and data2 to dat1 & dat2
    mov bx,data1
    mov dat1,bx

    mov cx,data2
    mov dat2,cx

    ;Algorithm for finding lcm
    ;if(dat1=dat2) then finish, lcm=dat1 or dat2
    ;elseif(dat1<dat2) then dat1=dat1+data1
    ;else dat2=dat2+data2
    ;repeat

loop1:mov ax,dat1
    cmp ax,dat2
    je jump5
    jc jump6

    mov ax,dat2
    add ax,cx
    mov dat2,ax
    jmp loop1

jump6:mov ax,dat1
    add ax,bx
    mov dat1,ax
    jmp loop1

    ;printing LCM
jump5:mov bx,0ah
    xor cx,cx

    ;push into stack
p1:xor dx,dx
    div bx
    push dx
    inc cx
    cmp ax,00h

```

```

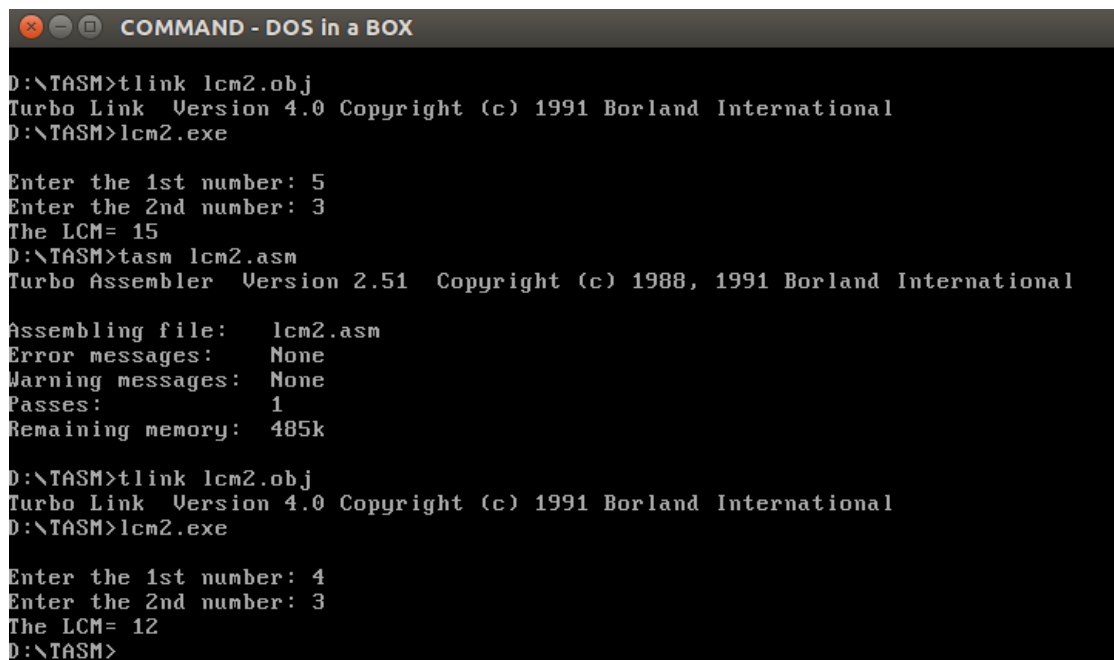
jne p1

print msg3
;pop from stack
display:pop dx
add dl,30h
mov ah,02h
int 21h
loop display
mov ah,4ch
int 21h

main endp
end

```

Output:



```

COMMAND - DOS in a BOX

D:\TASM>tlink lcm2.obj
Turbo Link Version 4.0 Copyright (c) 1991 Borland International
D:\TASM>lcm2.exe

Enter the 1st number: 5
Enter the 2nd number: 3
The LCM= 15
D:\TASM>tasm lcm2.asm
Turbo Assembler Version 2.51 Copyright (c) 1988, 1991 Borland International

Assembling file:   lcm2.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  485k

D:\TASM>tlink lcm2.obj
Turbo Link Version 4.0 Copyright (c) 1991 Borland International
D:\TASM>lcm2.exe

Enter the 1st number: 4
Enter the 2nd number: 3
The LCM= 12
D:\TASM>

```

5. Conclusion:

Exercise: Write an Assembly Language Program to find GCD of two numbers.