

# Assignment 09 | MFP

## CE-092

Assignment submission for Microprocessor Fundamentals and Programming subject week 9.

[nevilparmar24@gmail.com](mailto:nevilparmar24@gmail.com)

---

### Task 1:

Write an assembly language to find the LCM of two numbers.

**Passing params using registers.**

Code:

```
data segment
    num1 dw 0016
    num2 dw 0024
    gcd dw ?
    lcm dw ?
data ends

code segment
    assume ds:data,cs:code
start:
    mov ax,data
    mov ds,ax
    mov ax,num1      ;NUM1 is stored in ax
    mov bx,num2      ; NUM2 is stored in bx
    call calculateLCM ; calling the calculateLCM
procedure by passing the parameters in register
    mov lcm,AX
```

```

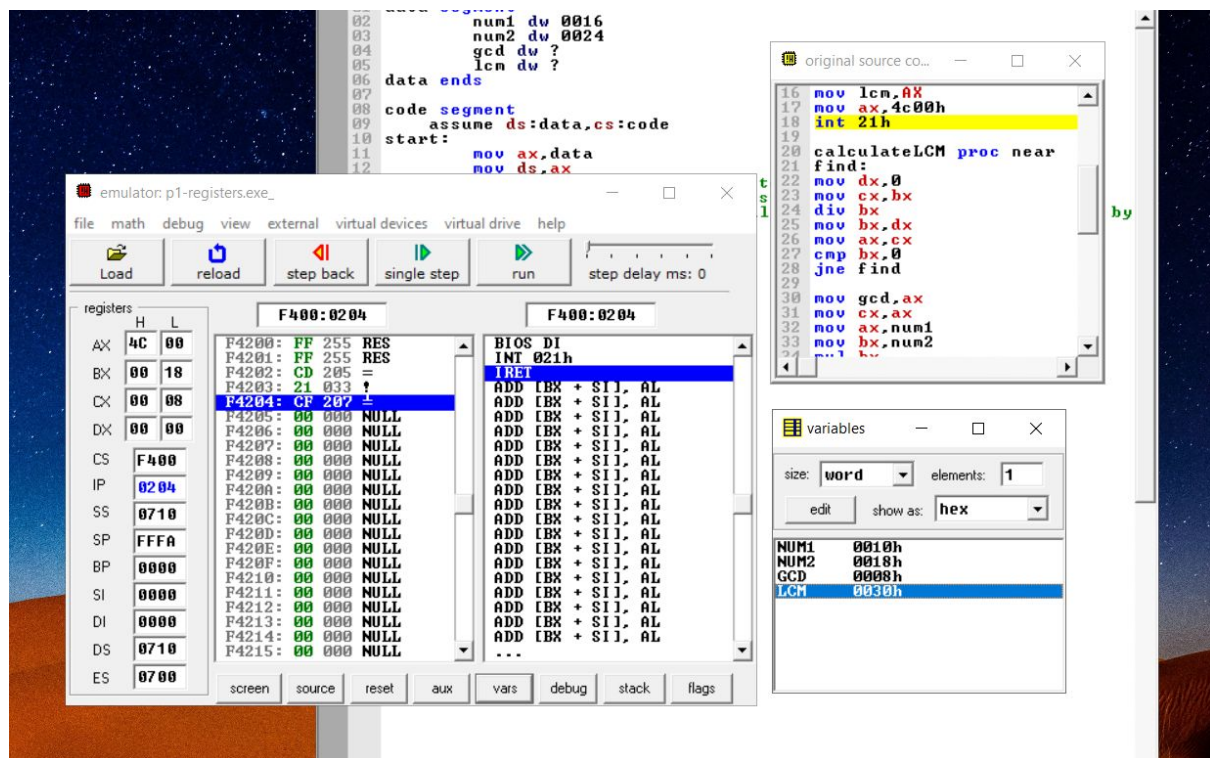
        mov ax,4c00h
        int 21h

calculateLCM proc near
find:
        mov dx,0
        mov cx,bx
        div bx
        mov bx,dx
        mov ax,cx
        cmp bx,0
        jne find

        mov gcd,ax
        mov cx,ax
        mov ax,num1
        mov bx,num2
        mul bx
        div cx
        RET
calculateLCM endp
code ends
end start

```

Output:



Passing params using pointers.

Code:

```
data segment
    num1 dw 0016
    num2 dw 0024
    gcd dw ?
    lcm dw ?
data ends

code segment
    assume ds:data, cs:code
start:
    mov ax, data
    mov ds, ax
    mov SI, offset num1 ; pass the offset of
```

```

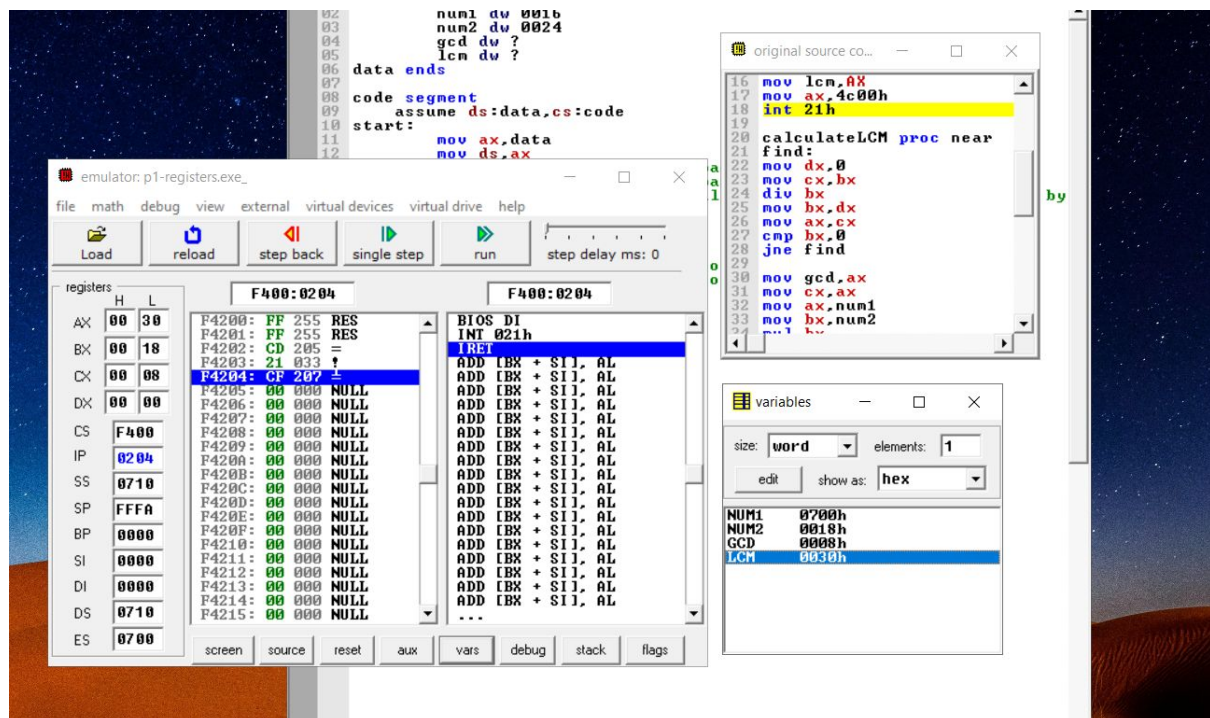
num1 in SI
    mov DI,offset num2      ; pass the offset of
num2 in DI
    call calculateLCM      ; calling the calculateLCM
procedure by passing the parameters as a pointer
    mov ax,4c00h
    int 21h

calculateLCM proc near
    mov ax,[si] ; ax points to num1 in the memory
    mov bx,[di] ; bx points to num2 in the memory
find:
    mov dx,0
    mov cx,bx
    div bx
    mov bx,dx
    mov ax,cx
    cmp bx,0
    jne find

    mov gcd,ax
    mov cx,ax
    mov ax,[si]
    mov bx,[di]
    mul bx
    div cx
    mov lcm, ax
    RET
calculateLCM endp
code ends
end start

```

Output:



## Passing params using stack.

Code:

```
data segment
    num1 dw 0016
    num2 dw 0024
    gcd dw ?
    lcm dw ?
data ends

code segment
    assume ds:data, cs:code
start:
    mov ax, data
    mov ds, ax
```

```

        ; push the necessary params to the stack to
        acces them inside the procedure instr
        push num1      ; offset SP + 10
        push num2      ; offset SP + 8
        push offset gcd      ; offset SP + 6
        push offset lcm      ; offset SP + 4

        call calculateLCM      ; calling the calculateLCM
        procedure by passing the parameters as a pointer
        mov ax,4c00h
        int 21h

```

```

calculateLCM proc near

```

```

    push bp
    mov bp, sp
    mov ax, [bp + 10]
    mov bx, [bp + 8]

```

```

find:

```

```

    mov dx,0
    mov cx,bx
    div bx
    mov bx,dx
    mov ax,cx
    cmp bx,0
    jne find

```

```

    mov gcd, ax ; store the gcd
    mov cx,ax
    mov ax, [bp + 10]
    mov bx, [bp + 8]
    mul bx
    div cx

```

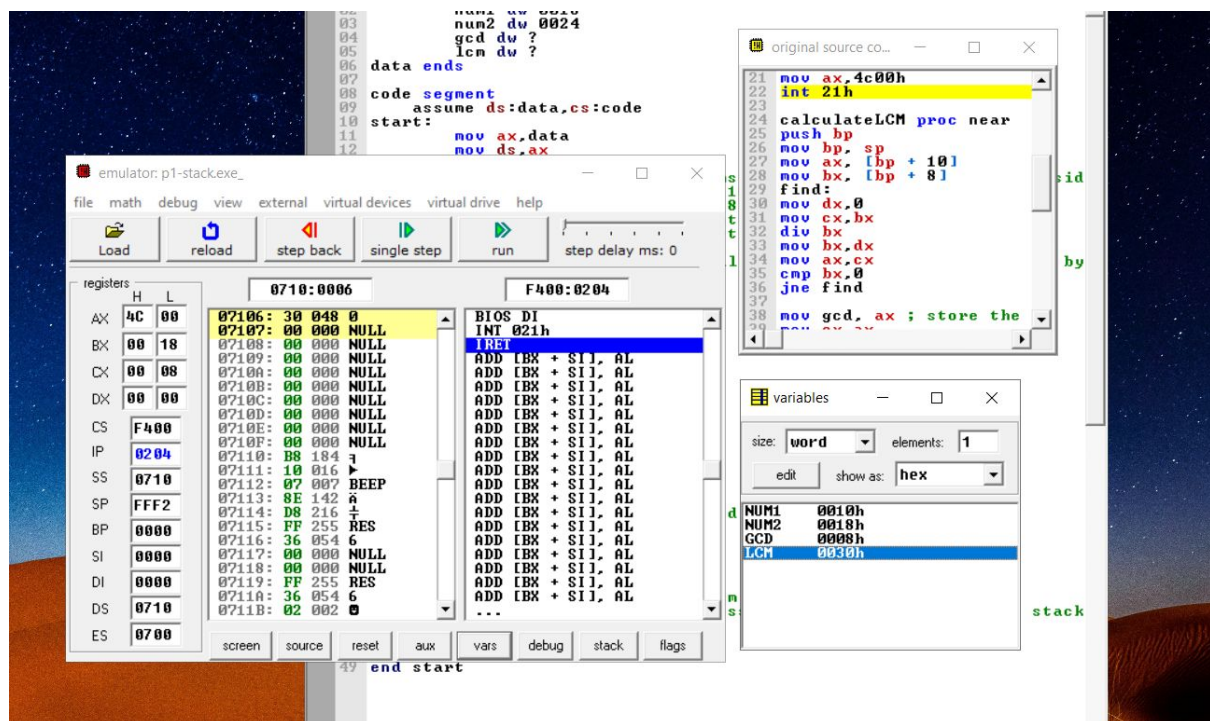
```

        mov lcm, ax ; store the lcm
        pop bp ; restore the address of BP from the top
of the stack

        RET
calculateLCM endp
code ends
end start

```

Output:



## Task 2:

Write an assembly language to divide a 32 bit number by 16 bit number. The program should handle the quotient size exceeds 16 bits.

### Code:

```
data segment
    num1 dd 45124579H
    num2 dw 0002h
    quotient dd ?
    remainder dw ?
data ends

code segment
    assume cs:code,ds:data
    mov ax,data
    mov ds,ax
    call division
    int 03

division proc near
    lea si,num1
    lea di,quotient
    mov ax,[si+2]
    mov bx,num2
    mov dx,0h
    div bx
    mov [di+2],ax
    mov ax,[si]
    div bx
    mov remainder,dx
    mov [di],ax
```



```

        ret
division endp
code ends
end

```

## Output:

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
C:\DEBUG125>debug .\p2final.exe
-g=0010
Unexpected breakpoint interrupt
AX=22BC BX=0002 CX=0039 DX=0001 SP=0000 BP=0000 SI=0000 DI=0006
DS=0744 ES=0734 SS=0743 CS=0744 IP=0019 NU UP EI PL ZR NA PE NC
0744:0019 BE0000          MOV     SI,0000
-d
0744:0000  79 45 12 45 02 00 BC 22-89 22 01 00 00 00 00 00 yE.E..."..."
0744:0010  B8 44 07 8E D8 E8 01 00-CC BE 00 00 BF 06 00 8B .D.....
0744:0020  44 02 8B 1E 04 00 BA 00-00 F7 F3 89 45 02 8B 04 D.....E...
0744:0030  F7 F3 89 16 0A 00 89 05-C3 26 80 0E 01 00 04 26 .....&.....&
0744:0040  A1 16 00 99 26 A3 0A 00-26 89 16 0C 00 26 88 16 ....&...&...&..
0744:0050  24 00 33 C0 26 A3 14 00-26 A3 16 00 26 80 26 05 $.3.&...&...&..
0744:0060  00 FC 26 80 0E 05 00 01-E8 5E 3B E8 43 3B E8 82 ..&.....^;.C;..
0744:0070  3B E9 DD 04 BF 3C 01 57-8B D8 E8 87 EE 26 C7 06 ;....<.W.....&..

```

Nevil Parmar  
CE-092

<https://nevilparmar.me>