**EXPERIMENT – 2**

**Program -1**

**Aim:** Store the data byte 32H into memory location 4000H.

**Code:**

; You may customize this and other start-up templates;

; The location of this template is c:\emu8086\inc\0\_com\_template.txt

org 100h

mov bx,0400h

mov ds,bx ; cannot move address to DS so made use of accumulator

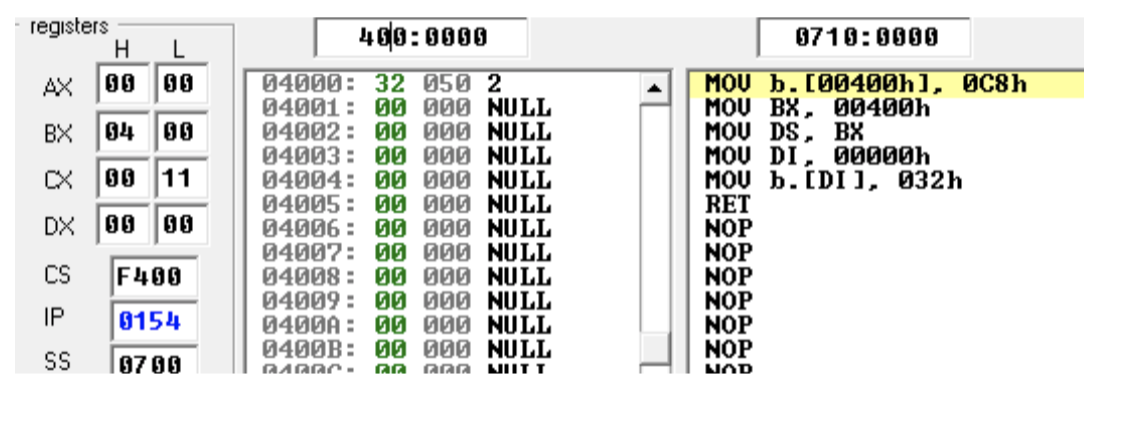
mov di,0000h ; 10h\*base+offset

mov [di],32h

ret

**Input/ Output:**

Input: BX: 400h

Output: 4000: 32h

**EXPERIMENT – 2**

**Program -2**

**Aim:** Exchange the contents of memory locations 2000H and 4000H

**Code:**

; You may customize this and other start-up templates;

; The location of this template is c:\emu8086\inc\0\_com\_template.txt

org 100h

; add your code here

mov ax,0200h ; base address 2000 stored in AX

mov ds,ax

mov di,0000h

mov [di] ; 19h; memory location [2000h] will have 19h

mov si,2000h ; 4000h location

mov [si] ; 26h 4000h memory location will have 26h

mov bx, [di] ; bx will have 19h

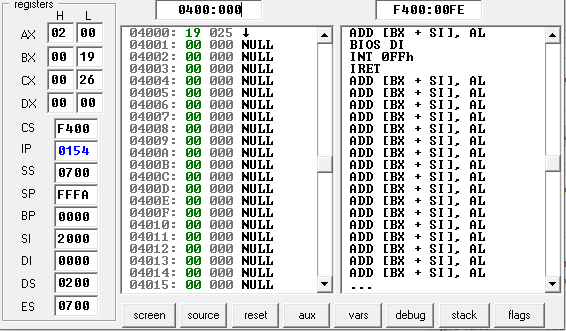
mov cx, [si] ; cx will have 26h

mov [di], cx ; now exchange will take place and cx will be move to DI

mov [si], bx; 19h in bx will be moved to SI

ret

**Input/ Output:**

Input: BX: 19h, CX:26h Output: CX:19h, BX:26h

**EXPERIMENT – 2**

**Program -3**

**Aim:** Convert the below given C Program into Assembly Language.

main()

{ intl,m,n,o,p;

l=m+n-o+p; }

**Code:**

; You may customize this and other start-up templates;

; The location of this template is c:\emu8086\inc\0\_com\_template.txt

org 100h

; add your code here

; here the four variables are taken in form of immediate data and the operation are performed using arithmetic operation

org 100h

mov ax,0200h

mov ds,ax

mov [001h],10 ; memory location 2001h will have 10

mov [002h],11 ; memory location 2001h will have 11

mov [003h],14 ; memory location 2001h will have 14

mov [004h],115 ; memory location 2001h will have 115

add ax,[001h] ; value in AX and 10 will be added

add ax,[002h] ; value of AX will be added with 11

sub ax,[003h] ; value of AX is subtracted with 14

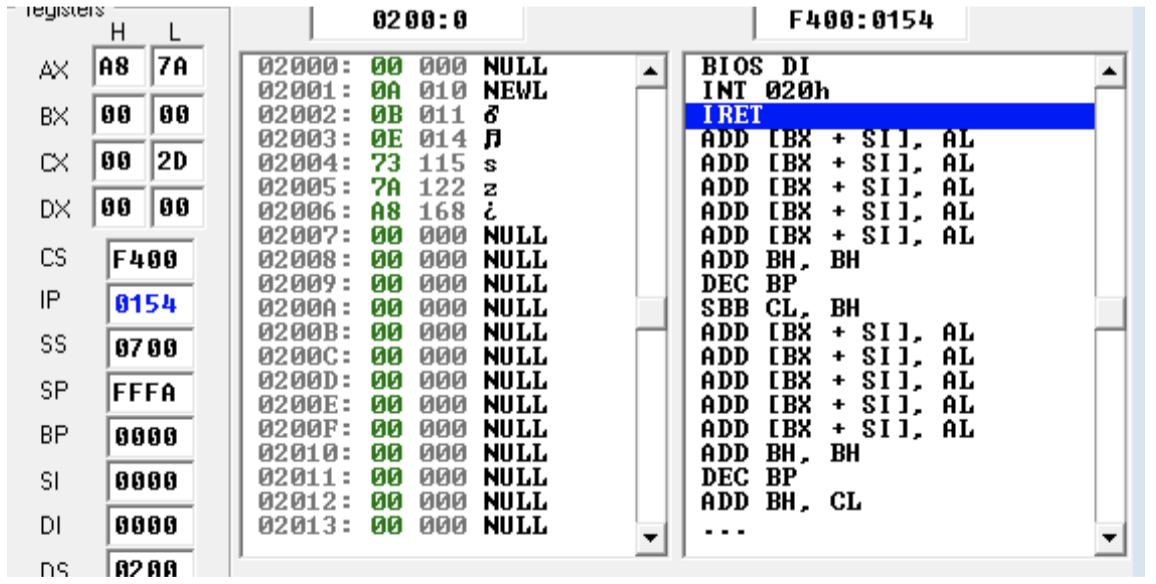
add ax,[004h] ; value of AX is added to 115

mov [005h],ax ; the final result is stored in 2005h memory location

ret

**Input/ Output:**

Input: [2001h]:10, [2002h]:11, [2003h]:14, [2004h]:115

Output: [2005h]: 7A

**EXPERIMENT – 2**

**Program -4**

**Aim:** Subtract the contents of memory location 4001H from the memory location 2000H

and place the result in memory location 4002H.

**Code:**

; You may customize this and other start-up templates;

; The location of this template is c:\emu8086\inc\0\_com\_template.txt

org 100h

; add your code here

MOV AX, 0200H ; base address is given to accumulator AX of 16 bits

MOV DS, AX ; data segment gets the base address of the AX

MOV DI, 0000H ; offset is 0 so the effective address (EA) is 2000H

MOV [DI], 9H ; 9h i.e. hexadecimal is move to destination index

MOV BX, [DI] ; and BX gets value of DX

; same as above instruction the later will be followed

MOV DI, 2001H

MOV [DI], 5H

MOV CX, [DI]

SUB BX, CX ; subtracts source from destination and answer is stored in destination

MOV DI, 2002H ; to store answer the offset is being assigned

MOV [DI], BX

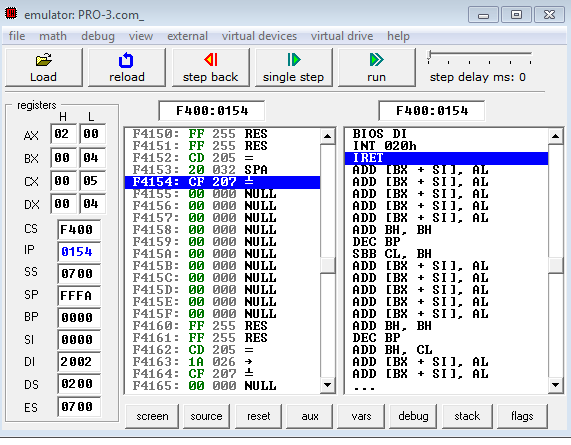
MOV DX, [DI] ; DX stores the result

ret

**Input / Output:**

Input: BX: 9H, CX: 4H

Output: DX: 5H



**EXPERIMENT – 2**

**Program -5**

**Aim:** Add the 16-bit number in memory locations 4000H and 4001H to the 16-bit number

in memory locations 4002H and 4003H. The most significant eight bits of the two numbers to be

added are in memory locations 4001H and 4003H. Store the result in memory locations 4004H

and 4005H with the most significant byte in memory location 4005H.

**Code:**

; You may customize this and other start-up templates;

; The location of this template is c:\emu8086\inc\0\_com\_template.txt

org 100h

; same as above code the AX is given base address and from offset assigned the value is being stored accordingly.

MOV AX, 0400H

MOV DS, AX

MOV DI, 0000H

MOV [DI],2211H ; here MSB will be stored at 40001H and LSB at 4000H

MOV BX, [DI]

MOV DI, 0002H

MOV [DI],4433H ; here MSB will be stored at 4003H and LSB at 4002H

MOV CX, [DI]

ADD BX, CX ; adds source and destination and result is stored in destination for 16bits

MOV DI, 0004H

MOV [DI], BX

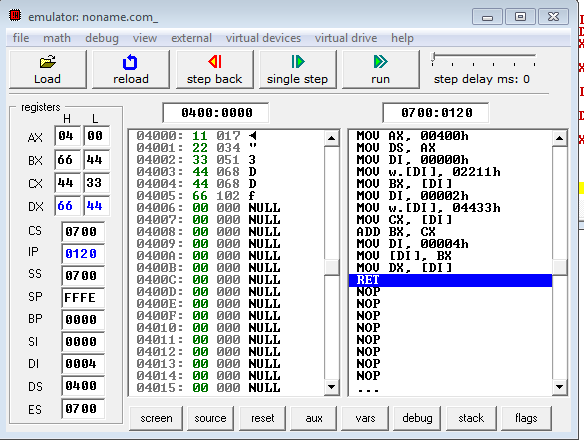
MOV DX, [DI] ; result is stored in DX and on address (LSB) 4004H and (MSB) 40005H, as it deals with 16bits

ret

**Input / Output:**

Input: AX: 2211H, BX: 4433H

Output: 6644H



**EXPERIMENT – 2**

**Program -6**

**Aim:** Subtract the 16-bit number in memory locations 4002H and 4003H from the 16-bit

number in memory locations 4000H and 4001H. The most significant eight bits of the two

numbers are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in Memory location 4005H.

**Code:**

; You may customize this and other start-up templates;

; The location of this template is c:\emu8086\inc\0\_com\_template.txt

org 100h

; same as above code the AX is given base address and from offset assigned the value is being stored accordingly.

; add your code here

MOV AX, 0400H

MOV DS, AX

MOV DI, 0000H

MOV [DI],2211H

MOV BX, [DI]

MOV DI, 0002H

MOV [DI],4433H

MOV CX, [DI]

SUB CX, BX ; BX is subtracted from CX (16bits) and the value is stored similar to the previous program, MSB at higher level address and LSB at lower.

MOV DI, 0004H

MOV [DI], CX

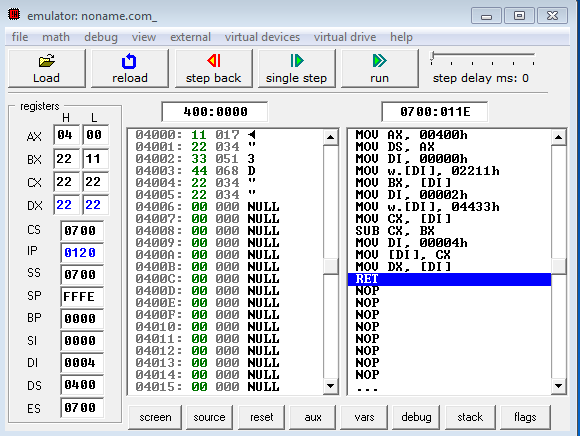
MOV DX, [DI]

Ret

**Input / Output:**

Input: AX: 2211H, BX: 4433H

Output: 2222H



**EXPERIMENT – 2**

**Program -7**

**Aim:** Add Two 32-bit numbers stored in consecutive memory locations and store the

result in memory locations starting from 7000H

**Code:**

; You may customize this and other start-up templates;

; The location of this template is c:\emu8086\inc\0\_com\_template.txt

org 100h

; for 32 bits the operations between two numbers will occur between the 16bit LSB’s of each number and then another operation will occur on 16bits MSB including carry flag.

MOV AX, 0400H

MOV DS, AX

MOV DI, 0000H

MOV [DI],7744H ; LSB of first number is assigned to DI and later to BX

MOV BX, [DI] ; from 4000H and 40001H LSB of first number will be stored

MOV DI, 0004H

MOV [DI],4433H ; LSB of second number is assigned to DI and later to CX

MOV CX, [DI] ; from 4004H and 4005H LSB of second number will be stored

ADD BX, CX

MOV DI, 2000H ; the result of first LSB addition will be stored at 6000H and 6001H

MOV [DI], BX

MOV DX, [DI] ; result of LSB stored in DX

MOV DI, 0002H

MOV [DI],2211H ; the MSB of first number is stored from 4002H to 4003H

MOV BX, [DI]

MOV DI, 0006H

MOV [DI],3322H ; the MSB of second number is stored from 4006H to 4007H

MOV CX, [DI]

ADC BX, CX ; add with carry operation is the summation of source destination and the carry flag generated by the LSB’s addition

MOV DI, 2002H ; result is stored at 6002H and 6003H of MSB’s addition with carry

MOV [DI], BX

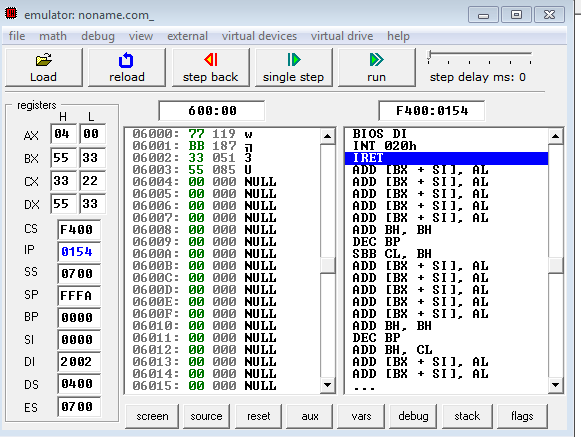
MOV DX, [DI]

ret

**Input/ Output:**

Input: BX: 22117744 and CX: 33224433

Output: 5533BB77



**EXPERIMENT – 2**

**Program -8**

**Aim:** Subtract Two 32-bit numbers stored in consecutive memory locations and store the

result in memory locations starting from 7000H

**Code:**

; You may customize this and other start-up templates;

; The location of this template is c:\emu8086\inc\0\_com\_template.txt

org 100h

; here the subtraction of two LSB’s take place and then for the MSB’s.

MOV AX, 0400H

MOV DS, AX

MOV DI, 0000H

MOV [DI],8899H ; LSB of first number is stored on 4000H and 4001H

MOV BX, [DI]

MOV DI, 0004H

MOV [DI],9933H ; LSB of second number is stored on 4004H and 4005H

MOV CX, [DI]

SUB BX, CX

MOV DI, 2000H ; result of LSB is stored at 6000H and 60001H

MOV [DI], BX

MOV DX, [DI]

MOV DI, 0002H ; MSB of first number is stored from 4002H to 4003H

MOV [DI],5566H

MOV BX, [DI]

MOV DI, 0006H ; MSB of second number is stored from 4006H to 4007H

MOV [DI],1144H

MOV CX, [DI]

SBB BX, CX ; Subtract with borrow

MOV DI, 2002H ; result of MSB is stored in 6002H to 6003H

MOV [DI], BX

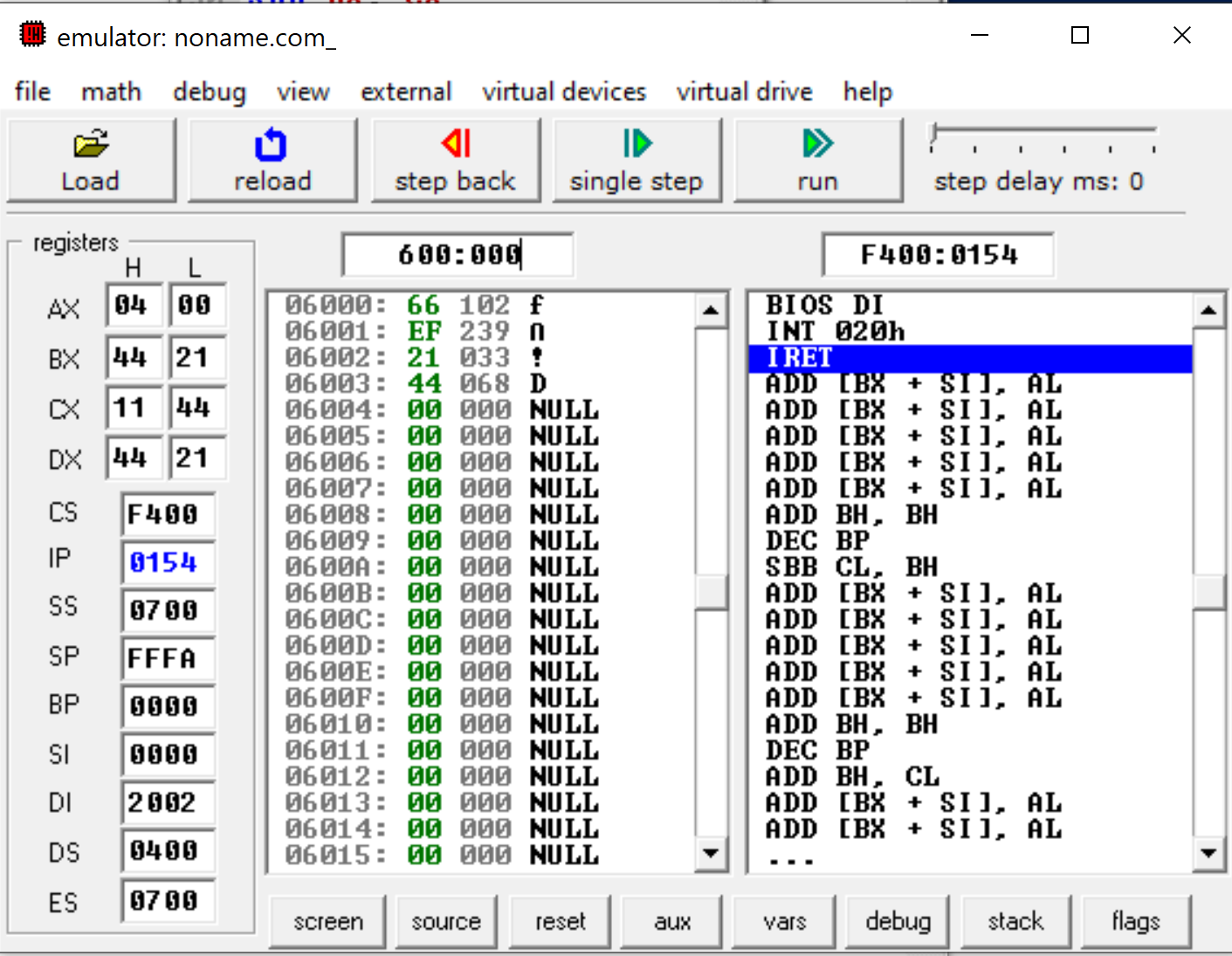
MOV DX, [DI]

ret

**Input/ Output:**

Input: BX: 55668899 and CX: 11449933

Output: 4421EF66



**EXPERIMENT – 2**

**Program -9**

**Aim:** Write an assembly language program to convert temperature in F to C.

C=(F-32) \*5/9

**Code:**

; You may customize this and other start-up templates;

; The location of this template is c:\emu8086\inc\0\_com\_template.txt

org 100h

; C=(F-32) \*5/9

mov ax,0400h

mov ds,ax

mov di,000h

mov ax, 104H ; move 104 in AX

mov [di],ax

sub ax,32 ; subtract 104-32 and result will be saved in AX

mov bx,5

mul bx ; multiply AX and BX

mov bx,9

div bx ; multiply AX and BX

mov di,0002h

mov [di], ax

ret

**Input/ Output:**

Input: AX: 104H

Output: AX: 28H

