**Computer Organization and Assembly Language**

|  |  |
| --- | --- |
| **Lab 05** | |
| **Topic** | * Arithmetic & Logical instructions * Selective bit setting/clearing/complimenting * Shifting and Rotations variations |

***Part 1***

**Example 1:**

**Let the binary of a number (0XABCD) is 10101011** **11001101.**

1. **Set the fourth bit.**

10101011 11011101

mov ax,0xABCD

or ax,0000000000010000b

ret

1. **Clear the L.S.B.**

1010101111001100b

mov ax,0xABCD

and ax,1111111111111110b

ret

1. **Invert the M.S.B.**

00101011 11001101

mov ax,0xABCD

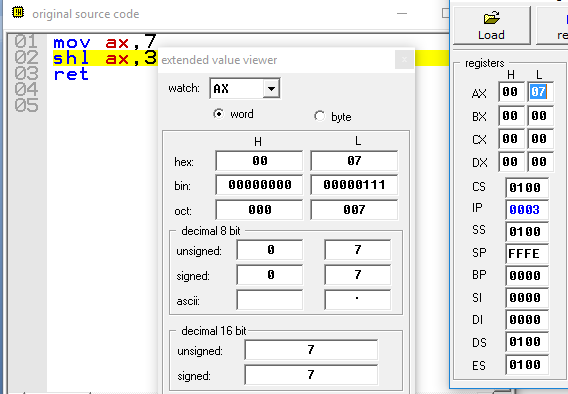
xor ax,1000000000000000b

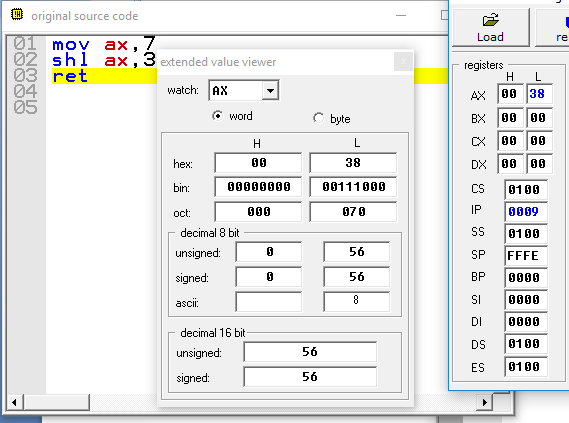
ret

***Note: logical operations are bitwise operations.***

**Example 2:**

**Multiply the number by 8 using shift operator.**

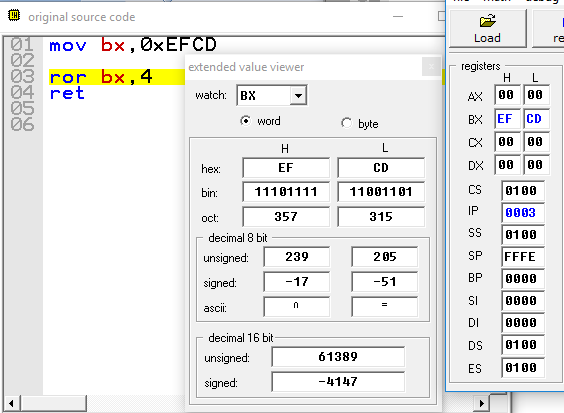
**Let the number is 7.**

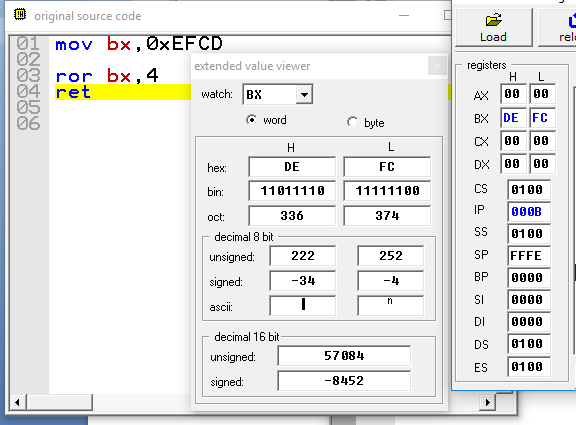
****

**Example 3:**

**Rotate 4 times the value in register bx.**

**Let BX=0xEFCD**

****

****

**PART 2**

**Problem #1:**

**Let Ax=0x56AB;**

Set3rd, 8th, 10th, 13th and 15thbit of ax.

Ans:

mov ax,0x56AB

or ax,1010010100001000b

then invert 3rd, 8th, 10th, 13th and 15thbit of ax.

Ans:

mov ax,0x56AB

xor ax,1010010100001000b

then clear 3rd, 8th, 10th, 13th and 15thbit of ax.

finally clear the ax.

Ans:

mov ax,0x56AB

and ax,0101101011110111b

**Problem #2:**

Write an assembly language program that will check whether the given numbers are Even or Odd. If number is even replace the value with 0 else if the number is odd replace the value with 1.

Let values are (byte size)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Value | 0xAB | 0x02 | 0xDF | 0xEF | 0x55 | 0xAA | 0x19 | 0x26 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Value | 1 | ? | ? | ? | ? | ? | ? | 0 |

Ans:

jmp start

arr db 0xAB,0x02,0xDF,0xEF,0x55,0xAA,0x19,0x26

mov cl,8

mov bx,0

start:

label1:

lea di,[arr+bx]

mov al,[di]

and al,00000001b

mov [di],al

inc bx

dec cl

cmp cl,8

jnz label1

ret

**Problem #3:**

Write an assembly language program that will count the number of occurrences of 0x5 and save the result in variable “total”.

**Without Using CMP and SUB instructions.**

**Array1** (byte size)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Value | 0x5 | 0x9 | 0x5 | 0x4 | 0xA | 0xB | 0x5 | 0x9 |
|  |  |  |  |  |  |  |  |  |

**Ans:**

**JMP ST:**

**value db 0X5,0X9,0X5,0X4,0XA,0XB,0X5,0X9**

**total db ?**

**ST:**

**mov cx,08h**

**mov si,00h**

**c:**

**mov Bl,[value[si]]**

**xor bl,05h**

**jz tal:**

**inc si**

**loop c:**

**jmp exit:**

**tal:**

**inc si**

**add dl,01h**

**loop c:**

**exit:**

**mov total,dl**

**ret**

**Problem #4:**

Write an assembly language program to multiply two variables using logical operators.

Let var1=0x7;

Var2=0x9;

Ax=var1\*var2;

**Problem #5:**

Write an assembly language program to count number of ones in a value of word size. Use logical operators.

Binary of F37E is : 1111 0011 0111 1110

For-example

Number dw: F37E h

Count db: 12

**Problem #6:**

Write an assembly language program to flipnthbit in a number of word size. As described in example.

Binary of F37E is : 1111 0011 0111 1110

For-example

Number dw: F37E h

Position db 5 (5th bit starting from first bit)

After Execution

Number dw: F35E h

**Problem #7:**

Write an assembly language program to count the number of ones in a higher byte of register ax. Let value in ax

**(Hint: You can use ROR,JMP,TEST,LOOP,JNZ)**

**Before execution**

AX=EF56

Counter =0

**After execution**

AX=EF56

Counter =7

Answer:

jmp st:

num dw 0xEF56

counter db ?

st:

mov ax,num

mov cx,8

c:

ror ah,1

jc odd:

mov dl,0h

loop c:

jmp exit:

odd:

mov dl,1h

add dh,dl

loop c:

exit:

mov counter,dh

ret