**Instructor Name:** Dr Usman Abbasi

**Student Name:** Zeeshan Zafar

**Roll no:** p200569

**Department:** BS(CS)

**Section:**3D

**Date**: Nov 17,2021

**Report:** 07

**. 1-Bitwise Logical operations .**

**The four Basic operations:**

1-AND

2-OR

3-XOR

4-NOT

1-AND (and):

*In and when our both bits are 1 then our result will be also 1*

*It will takes two operands*

**Syntax:**

and oper1,oper2

our result will be stored in oper2

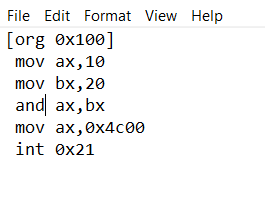
**Example:**

10=1010 ,20=10100

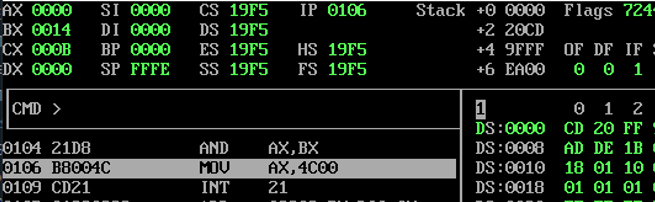
*Result will be*

Ax=0000000

**Code:**



**Output:**



**2- OR operation**

*The OR operator returns 1, if the matching bits from either or both operands are one. It returns 0, if both the bits are zero otherwise 1.*

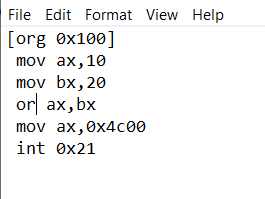
*Example:*

Al=1010

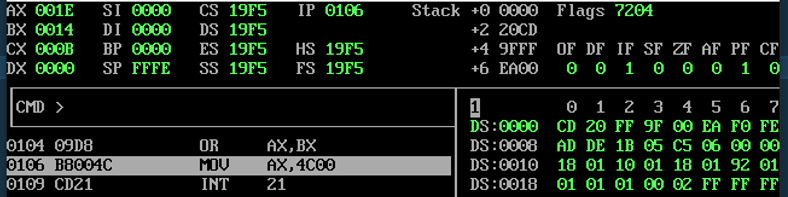
Bl=10100 ; by or both operands result will be

Ax=00011110

**Code:**



**Output:**



**3**-**XOR operand:**

*Actually XOR operation sets the resultant bit to 1, if and only if the bits from the operands are not same. If the bits from the operands are same, the resultant bit will be cleared to 0.*

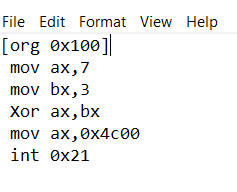
*For Example:*

*If 7=0111 and 3=0011*

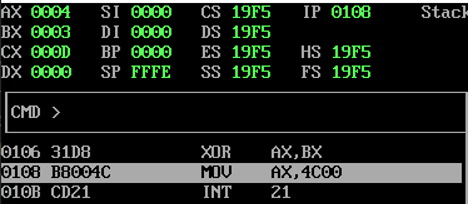
**Result**

XOR = 0100

**Code:**



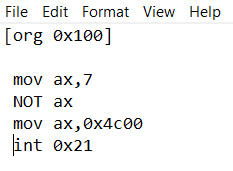
**Output:**



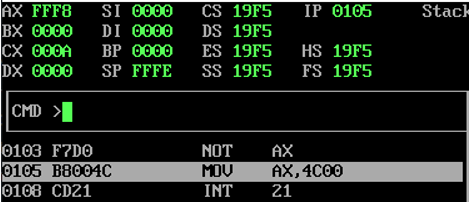
**4-NOT operand:**

*This operation reverse the bits in an operand. The operand could be either in a register or in the memory*

Code:



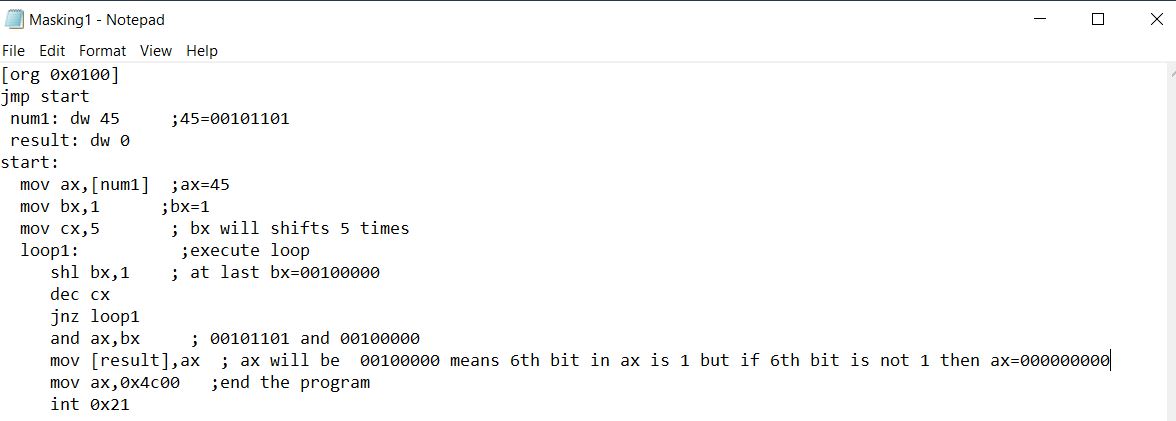
Output:



**Bit Manipulation .**

**First - Find ith Bit**

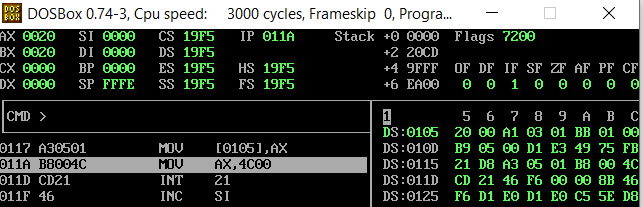
**Code:**



**Explanation:**

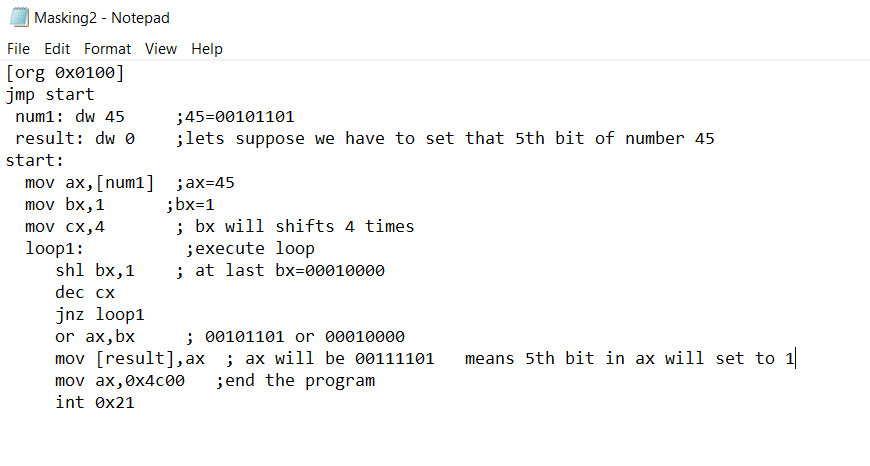
*In this code we will find that 6th bit of number 45 is 1 or not.So we will shift bx(bx=1) 5 times after this we will multiply it with ax if the 6th bit is one ax will remain same other wise it will be zero .*

**OutPut:**



**2 – Set the ith Bit:**

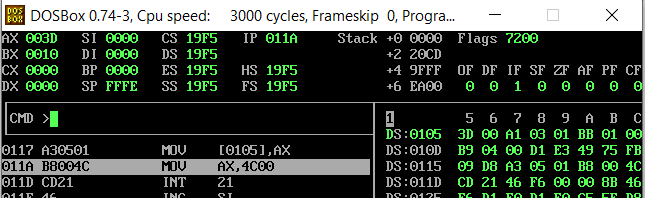
**Code:**



**Explanation:**

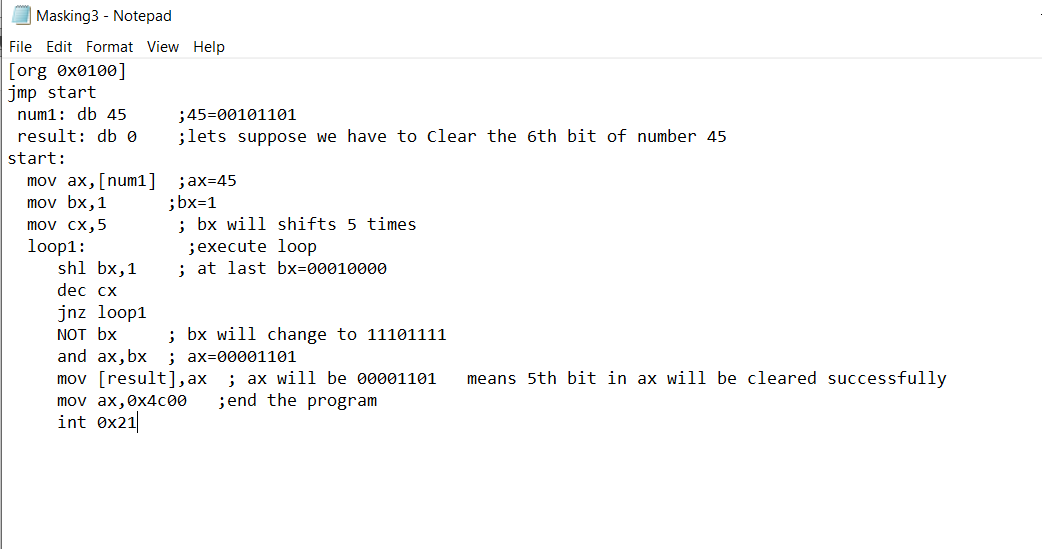
*In this code we will set 5th bit of number 45 to 1.So we will shift bx(bx=1) 4 times after this we will take add it with ax and finally bit will be set and we move this value to result*

**OutPut**



**3 – Clear the ith Bit:**

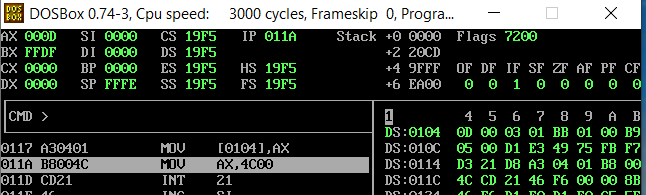
**Code:**



**Explanation:**

*In this code we will clear that 6th bit of number 45.So we will shift bx(bx=1) 5 times after this we will take inverse of bx and then multiply it with ax and the 6th bit will be cleared and become 0 at last move ax to result variable.*

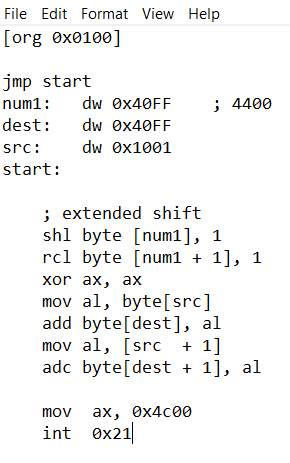
**OutPut:**



**Extended operations .**

**1-Extended Addition:**

**Code:**

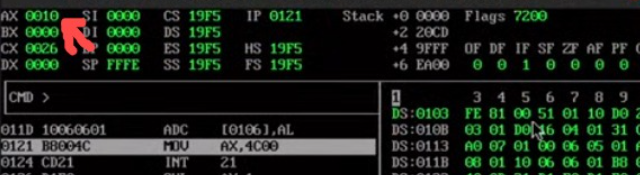


**Explanation:**

*In this First we will adds lower bits of destination and source and then we add higher bits of destination and source with carry we have taken destination and source as word and we will take the higher part of source register and add into the higher part of destination register and carry*

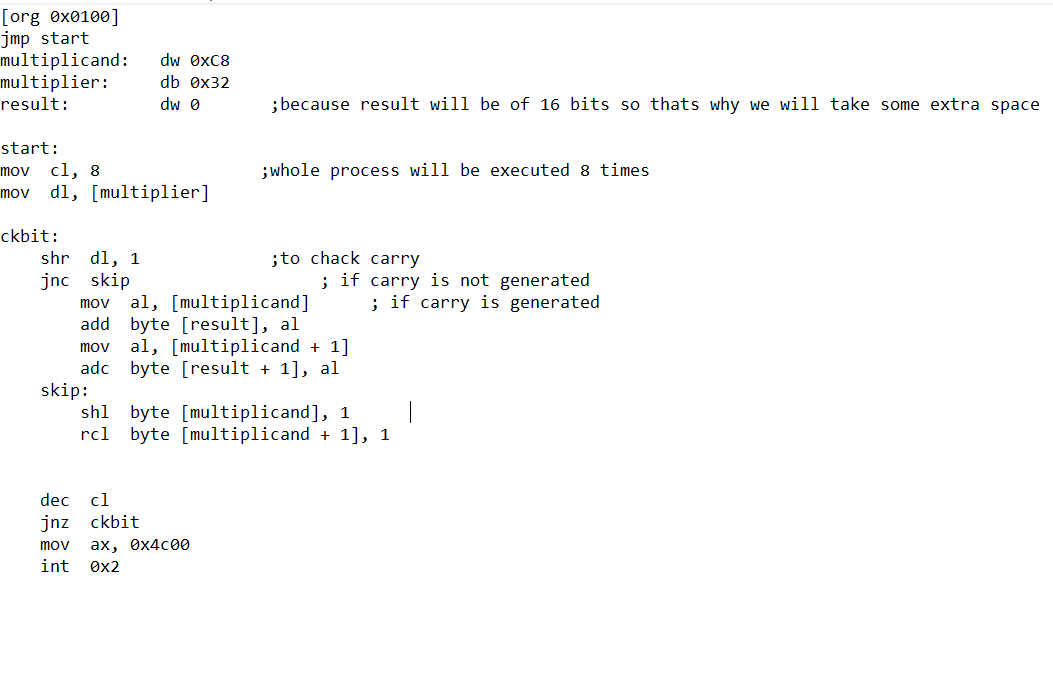
*also added during in it by using a keyword adc(add with carry). Similarly we will take the lower part of source register and add into the lower part of destination register this is how extended addition will be performed.*

**OutPut:**



**2-Extended Multiplication:**

**Code:**



***Explanation:***

*In extended Multiplication we have 8bit number but we will take 16bit registor because when the shifting is performed we have to save carries in future , we have taken 8 bit multiplier and result is defined for 16bit number because there will be also a possibility to produce 16 bit output we will only use shift left and rotate through carry left to do multipication.*

***OutPut:***



-End