Name:	Vedika Pawar
Roll No:	46
Class/Sem:	SE/IV
Experiment No.:	2A
Title:	Program to perform multiplication without using MUL instruction
Date of Performance:	
Date of Submission:	
Marks:	
Sign of Faculty:	

Aim: Program for multiplication without using the multiplication instruction.

Theory:

In the multiplication program, we multiply the two numbers without using the direct instructions MUL. Here we can successive addition methods to get the product of two numbers. For that, in one register we will take multiplicand so that we can add multiplicand itself till the multiplier stored in another register becomes zero.

ORG 100H:

It is a compiler directive. It tells the compiler how to handle source code. It tells the compiler that the executable file will be loaded at the offset of 100H (256 bytes.)

INT 21H:

The instruction INT 21H transfers control to the operating system, to a subprogram that handles I/O operations.

MUL: MUL Source.

This instruction multiplies an unsigned byte from some source times an unsigned byte in the AL register or an unsigned word from some source times an unsigned word in the AX register.

Source: Register, Memory Location.

When a byte is multiplied by the content of AL, the result (product) is put in AX. A 16-bit destination is required because the result of multiplying an 8-bit number by an 8-bit number can be as large as 16-bits. The MSB of the result is put in AH and the LSB of the result is put in AL.

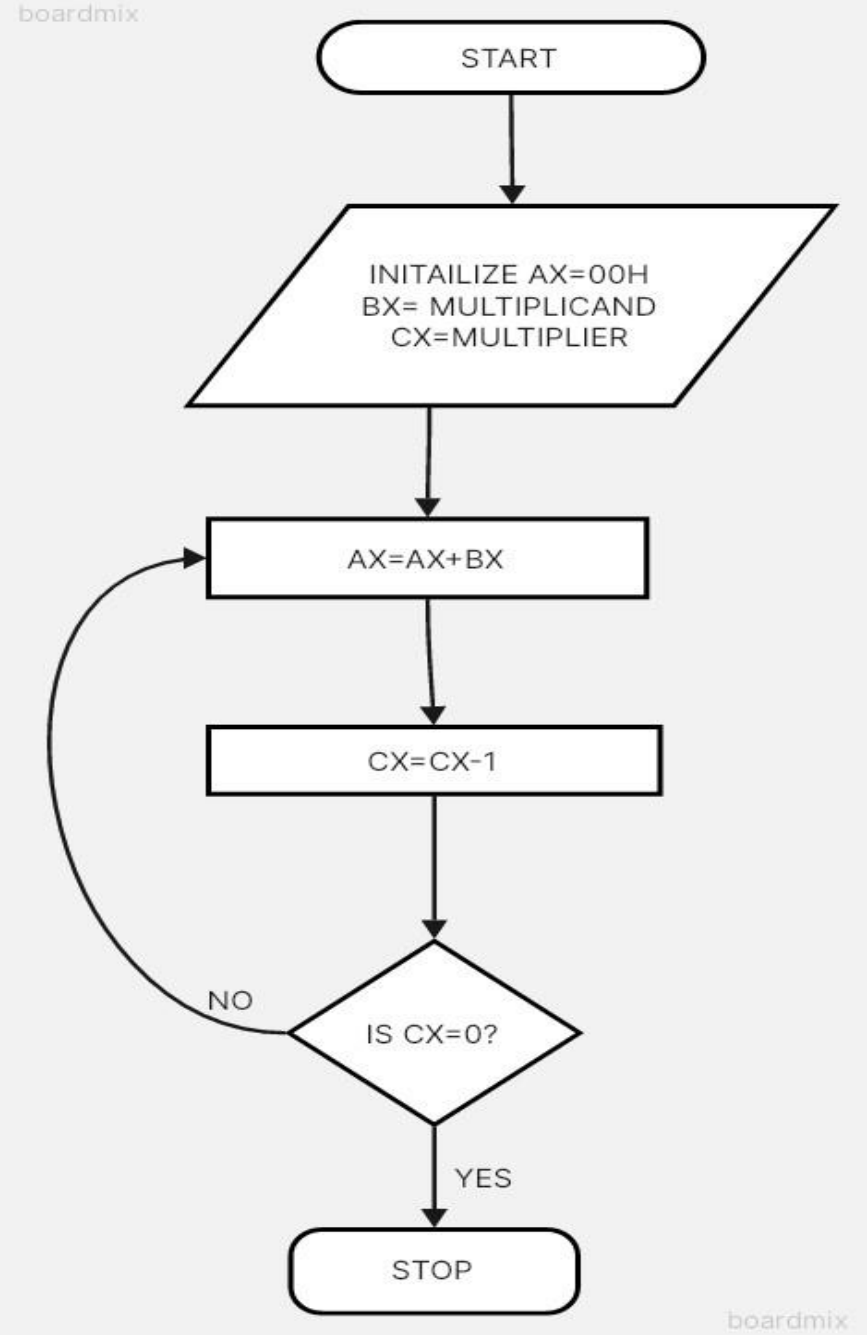
When a word is multiplied by the contents of AX, the product can be as large as 32 bits. The MSB of the result is put in the DX register and the LSB of the result is put in the AX register.

MUL BH; multiply AL with BH; result in AX.

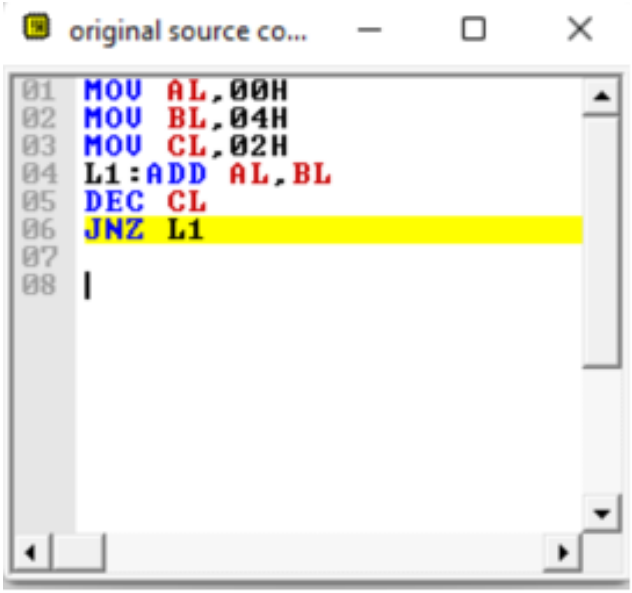
Algorithm:

1. Start.
2. Set AX=00H, BX= Multiplicand, CX=Multiplier 3 Add the content of AX and BX.
4. Decrement content of CX.
5. Repeat steps 3 and 4 till CX=0.
6. Stop.

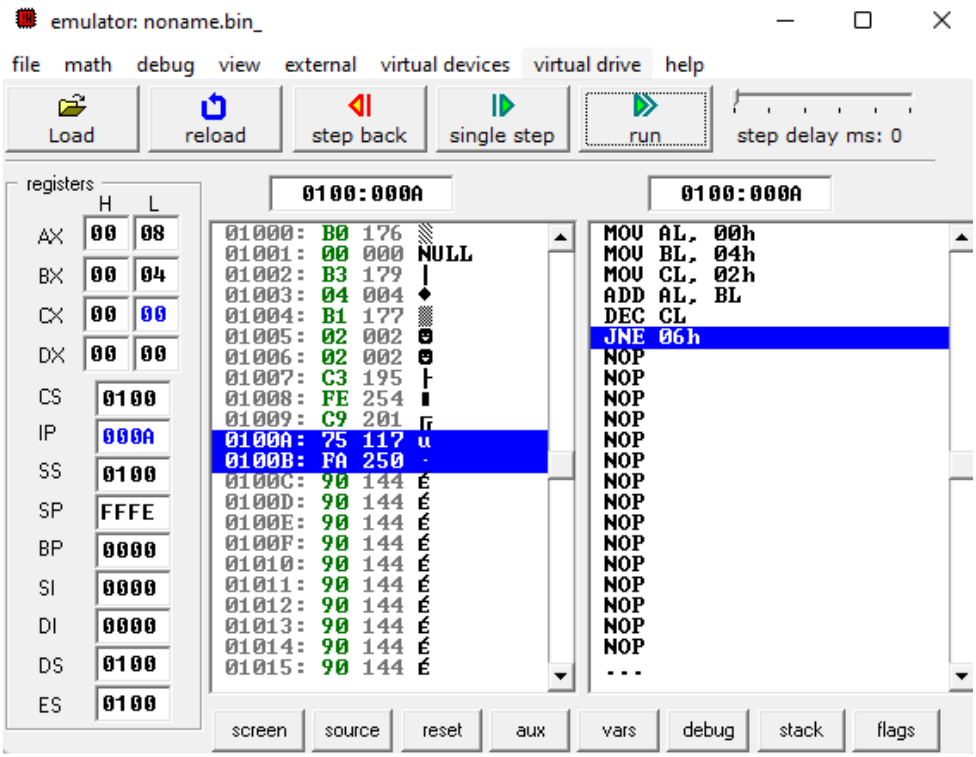
Flowchart:



Assembly Code:



Output:



Conclusion:

1. Explain data transfer instructions.

Ans. Data transfer instructions are essential in computer architecture for moving data between memory, registers, and I/O devices. Here's an overview of these instructions:

a) Load (MOV): Copies data from memory or an I/O device into a register, enabling further processing.
b) Store (MOV): Transfers data from a register to memory or an I/O device, preserving it for later use.
c) Push and Pop: Pushes data onto or pops data off the stack, facilitating function calls and parameter passing.
d) Load Immediate (LXI): Loads an immediate value directly into a register, useful for initializing variables or constants.
e) Exchange (XCHG): Swaps the contents of two registers, aiding in data manipulation or register management.
f) Input and Output (IN, OUT): Reads data from an input port or sends data to an output port, facilitating communication with peripherals.
g) Move String (MOVS): Transfers a block of data from one memory location to another, commonly used in string manipulation operations.

These instructions enable efficient data movement and manipulation, fundamental to program execution in a computer system.

2. Explain Arithmetic instructions.

Ans. Arithmetic instructions are fundamental operations in computer architecture for manipulating numerical data:

a) Addition (ADD): Adds the content of a specified memory location or register to the content of the accumulator, storing the result in the accumulator.
b) Subtraction (SUB): Subtracts the content of a specified memory location or register from the content of the accumulator, storing the result in the accumulator.
c) Increment (INC): Increases the content of a specified memory location or register by one.
d) Decrement (DEC): Decreases the content of a specified memory location or register by one.
e) Multiply (MUL): Multiplies the content of the accumulator by a specified memory location or register, storing the result in the accumulator.
f) Divide (DIV): Divides the content of the accumulator by a specified memory location or register, storing the quotient in the accumulator and the remainder in another specified register.
g) Increment and Decrement (INR, DCR): Instructions to increment or decrement the content of a specified register or memory location by one, respectively.

These instructions are essential for performing mathematical operations in programs, making them indispensable for virtually all computing tasks.