Name: Amber Khurshid

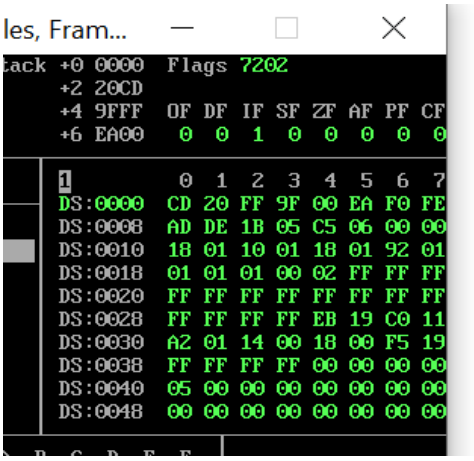
Section: BAI-4A

Roll No: 22P-9295

COAL Lab Task 2

Memory Representation M1:

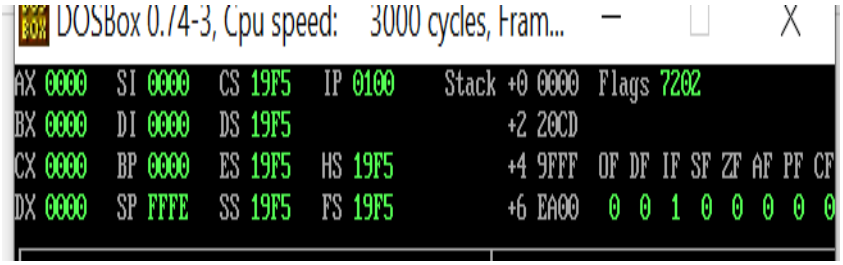
M1 refers to the organization of memory in a computer system, here rows represent different areas of memory.



Memory Representation M2:

M2 represents organization of memory column-wise.

Different types of Registers:



Following are some general-purpose registers that are commonly used for various operations:

AX register(accumulator register):

AX is the primary register used for arithmetic and logic operations. It can be divided into two 8-bit registers: AH (high byte) and AL (low byte).
It is commonly used for storing the result of mathematical operations.

BX register(Base Address Register):

It is often used as a base register for memory addressing.

Similar to AX register, it can be divided into two 8-bit registers: BH (high byte) and BL (low byte).

CX register(Count Register):

CX is used in loop control and string manipulation instructions.

Like AX and BX, it can be divided into two 8-bit registers: CH (high byte) and CL (low byte).

DX (Data Register):

It is used for I/O operations and storing the result of certain instructions and is divided into two 8-bit registers: DH (high byte) and DL (low byte).

SP (Stack Pointer):

It is the stack pointer register that points to the top of the stack. It is used in managing the stack, including pushing and popping values.

IP (Instruction Pointer):

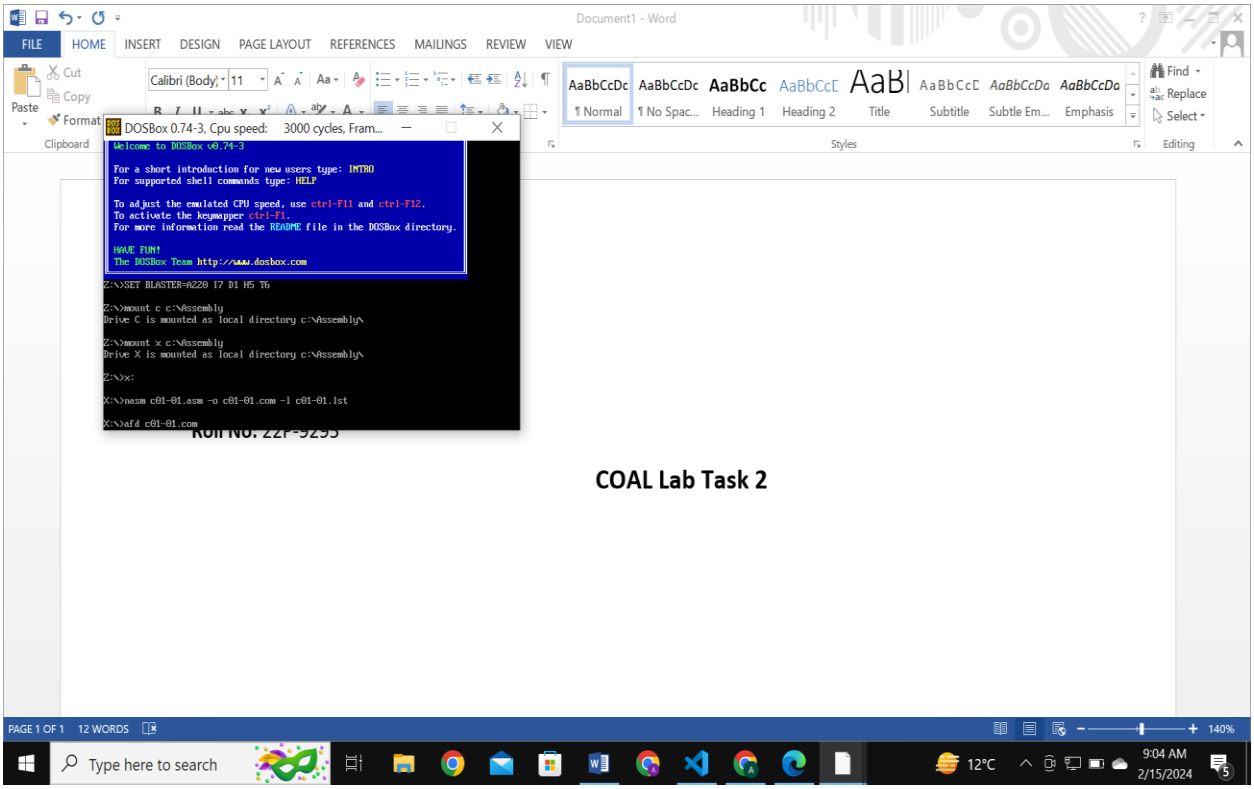
It is the instruction pointer register which holds the memory address of the next instruction to be executed. It also determines the flow of control in the program.

FLAGS:

The FLAGS register, also known as the status register, contains various status flags that reflect the outcome of arithmetic and logical operations. Some common flags include zero flag (ZF), carry flag (CF), and overflow flag (OF). These flags are used to control conditional branching and to check the status of the last operation.

Step 1

Mount the file using the following commands.



*****[org 0x0100]**

The org command tells the program where to load itself into in memory .

Step 2

In this step we run the command.

mov ax, 5

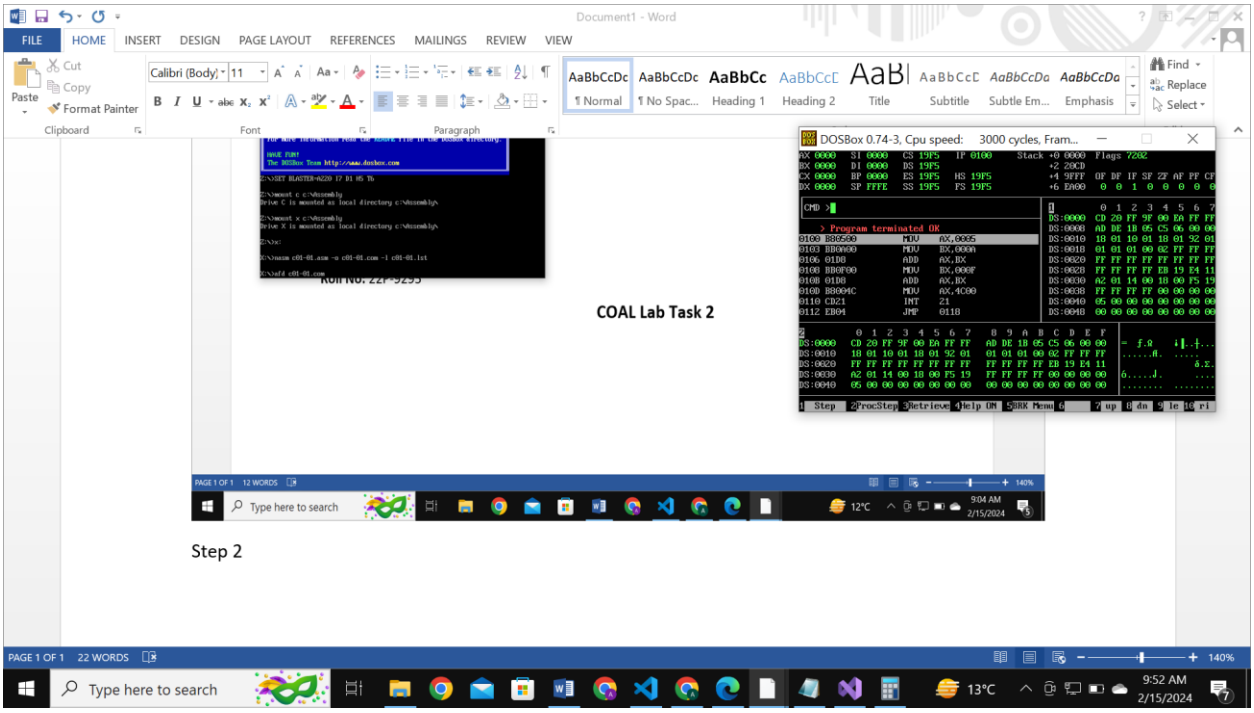
In assembly language it is given as:

0100 B80500

0100 shows the memory address of the instruction.

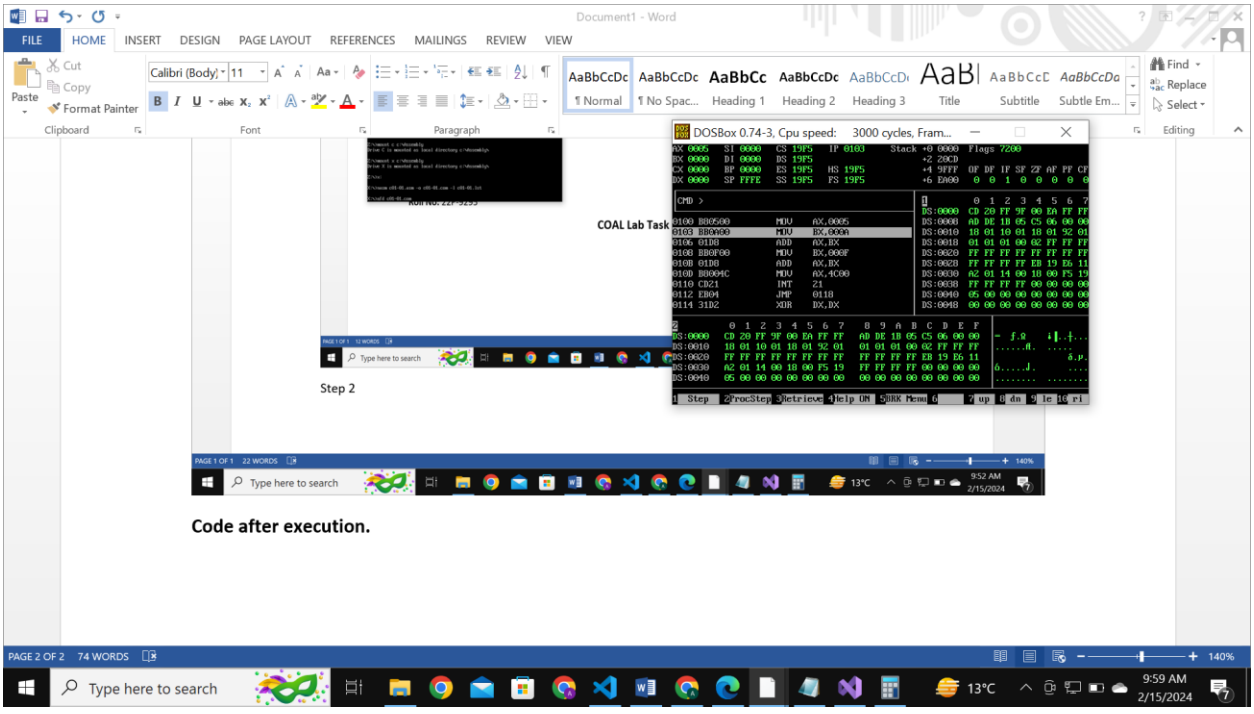
B80500 here **B8** is opcode for **mov** instruction and **0500** is the value to be moved in **ax** register.

Code before execution.



Code after execution.

Here we can see in the top left AX register has got the value 5.



Step 3:

In this step we run the command.

mov bx, 10

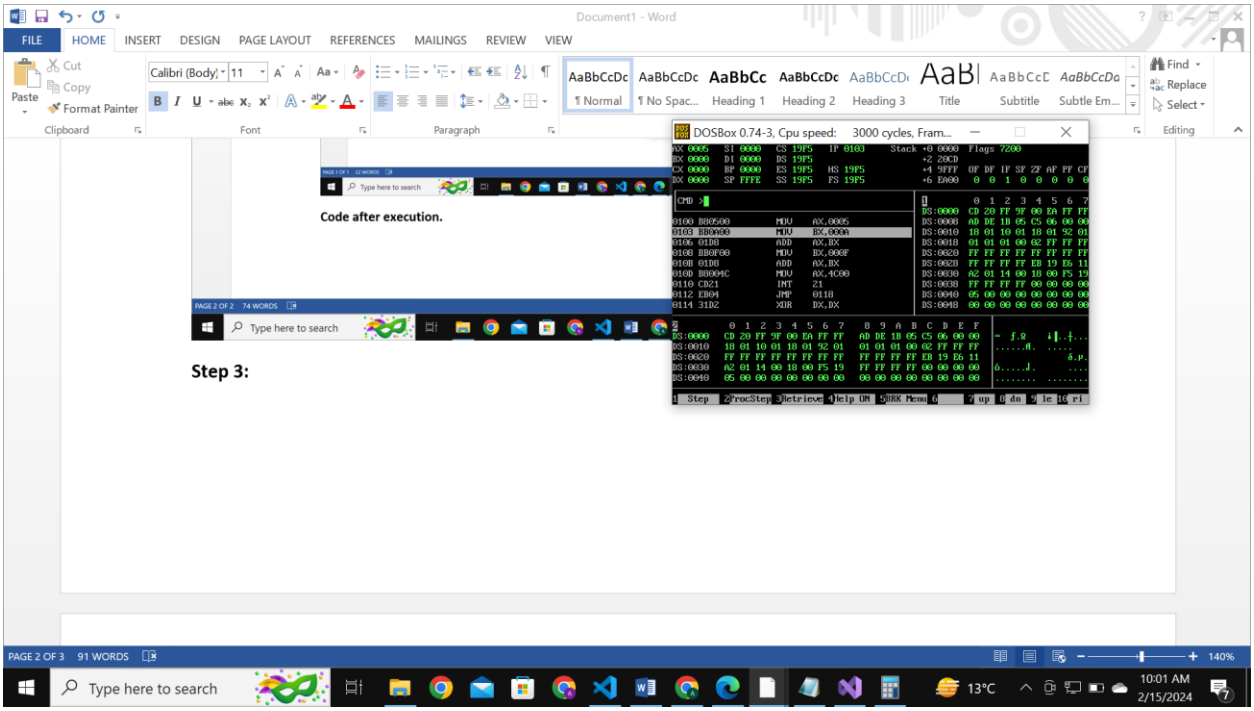
In assembly language it is given as:

0103 BB0A00

0103 shows the memory address of the instruction.

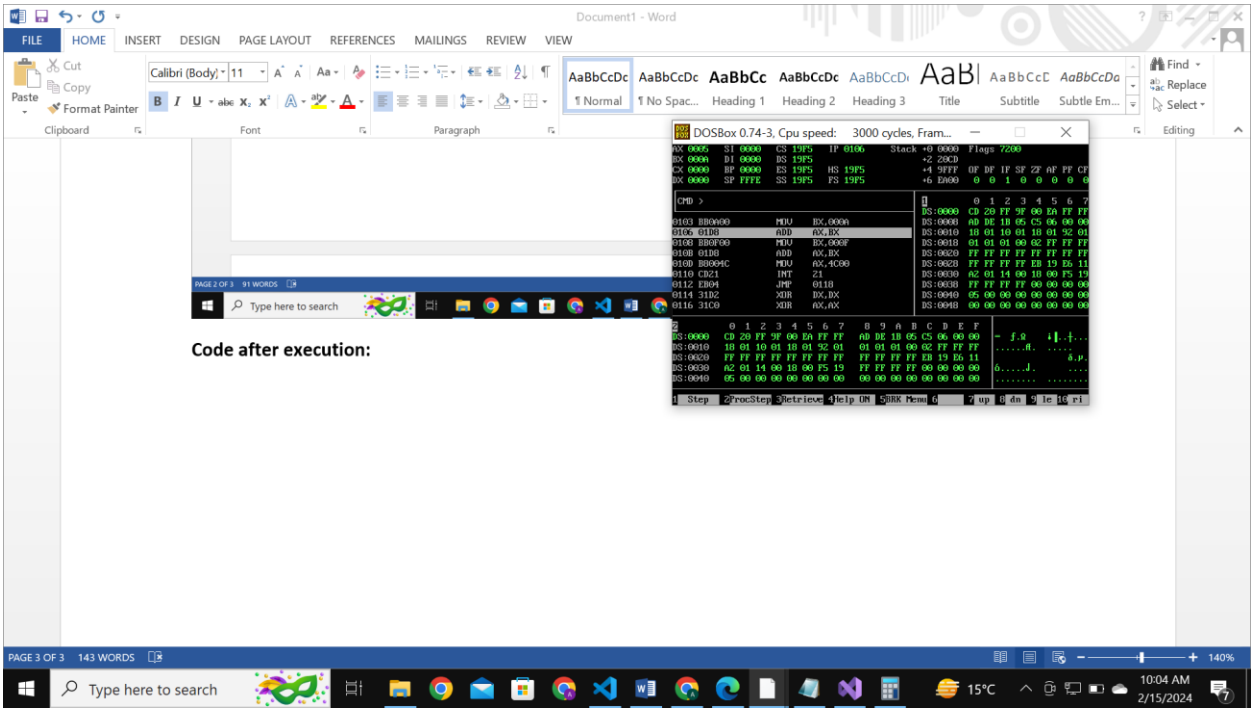
BB0A00 here **BB** is opcode for **mov** instruction and **0A00** is the value to be moved in **ax** register.

Code before execution.



Code after execution:

Here we can see in the top left BX register has got the value 5.



Step 4:

In this step we run the command.

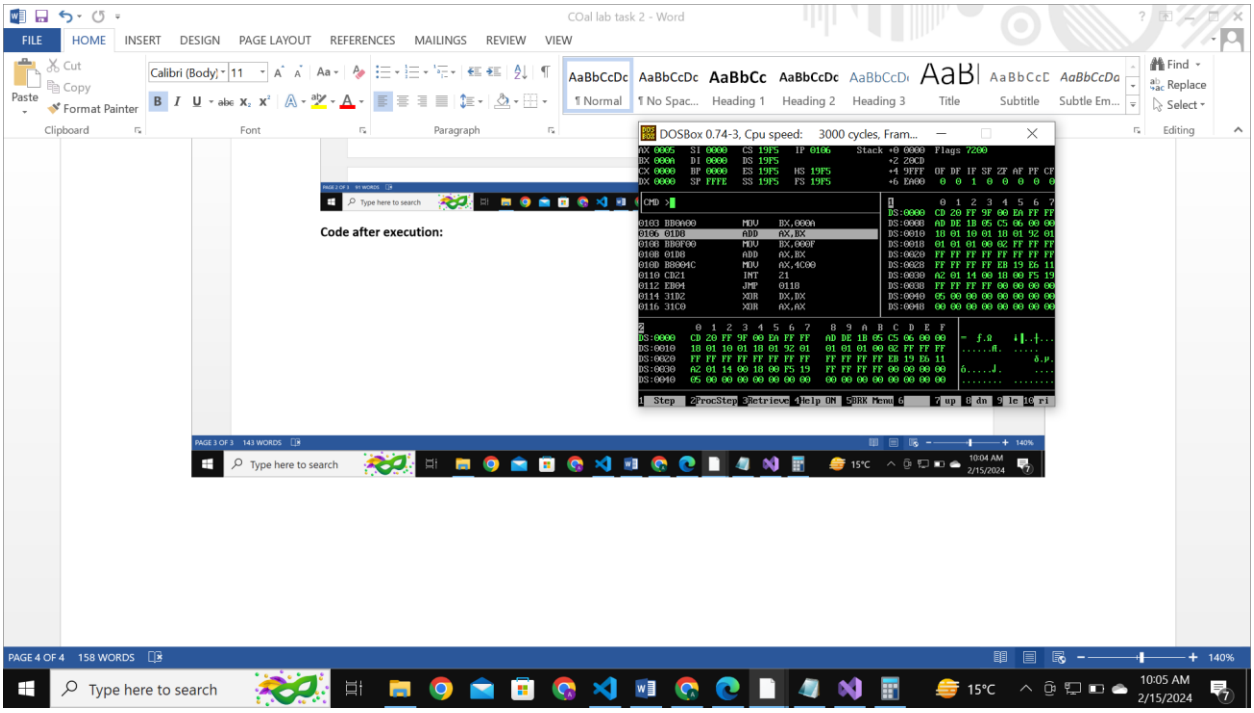
add ax,bx

In assembly language it is given as:

0106 01D8

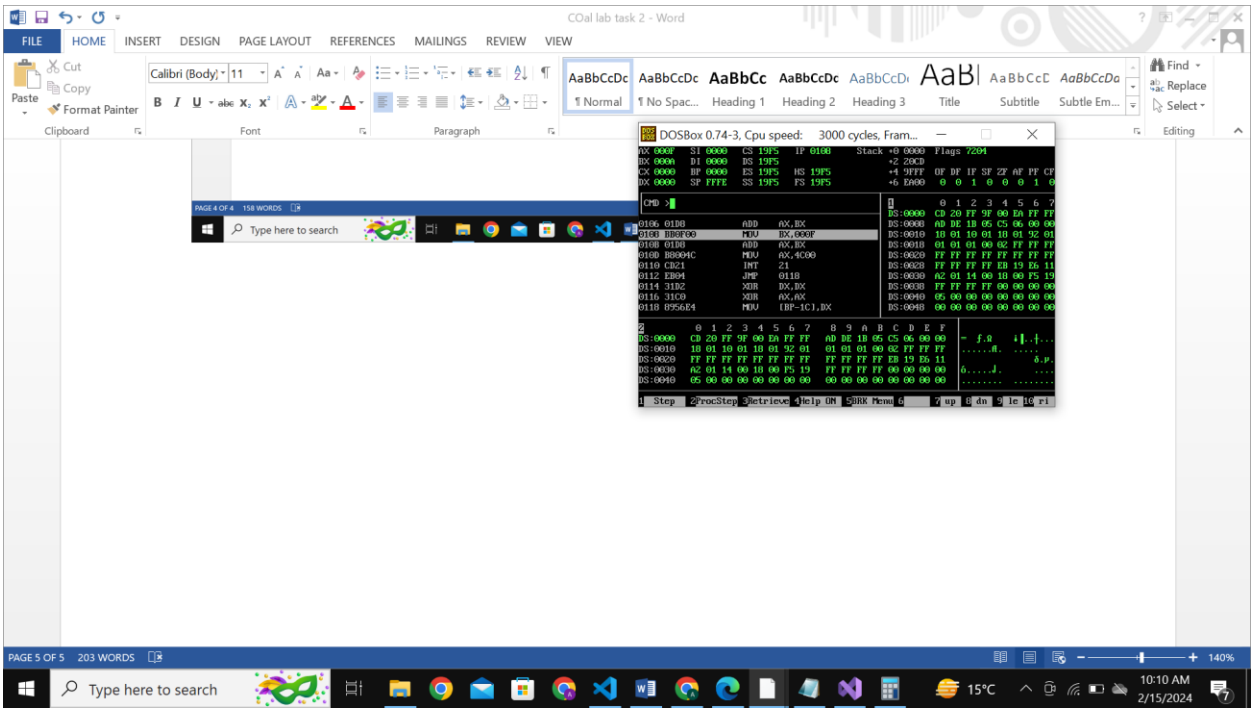
0106 shows the memory address of the instruction.

01D8 here **01** is opcode for **add** instruction and **D8** is the operands **ax** and **bx** registers.



After execution:

Here the value of ax (5) and **bx**(10) are added together and the result in stored in ax register.



Step 5:

In this step we run the command.

mov bx, 15

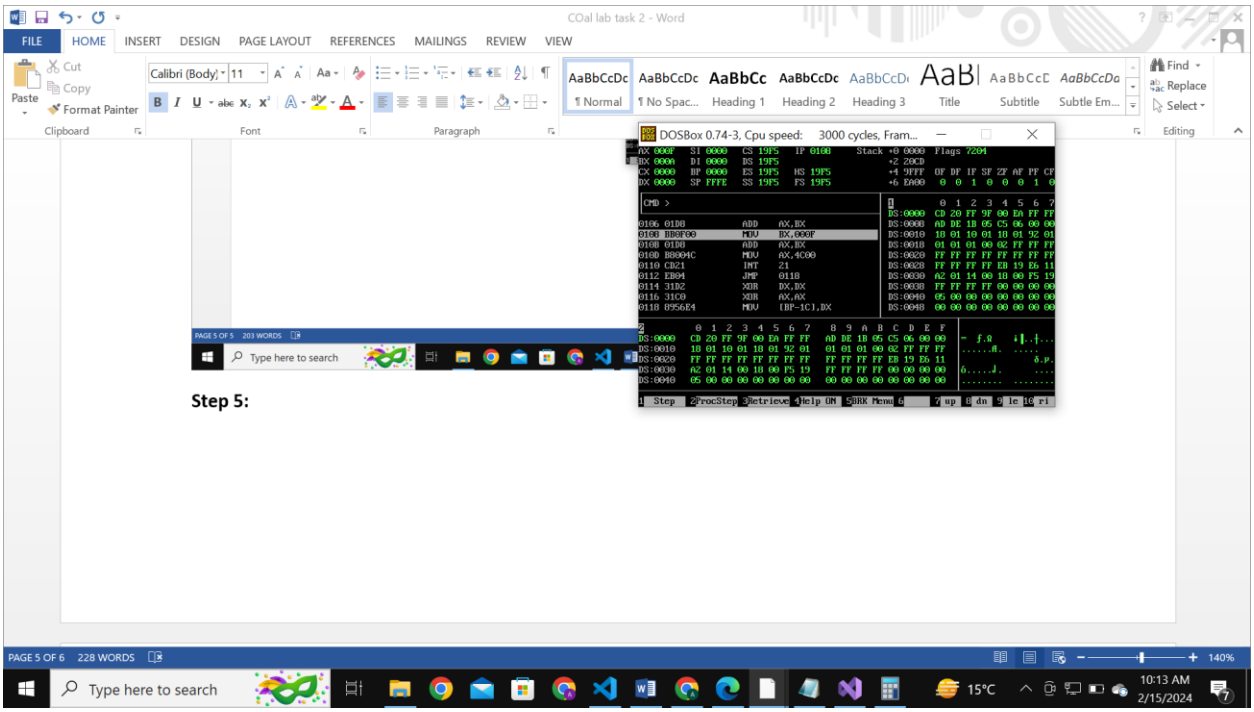
In assembly language it is given as:

0108 BB0F00

0108 shows the memory address of the instruction.

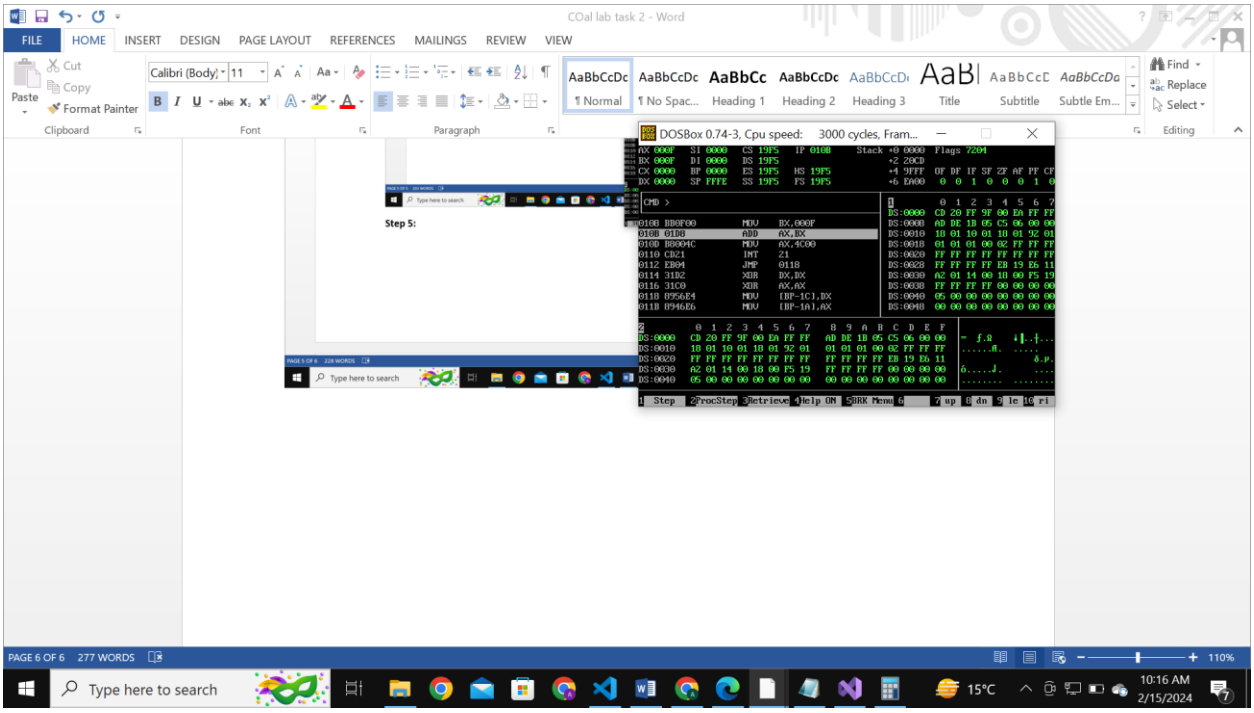
BB0A00 here **BB** is opcode for **mov** instruction and **0F00** is the value to be moved in **bx** register.

Code before execution.



After execution:

Here we can see in the top left BX register has got the value 15.



Step 6:

In this step we run the command.

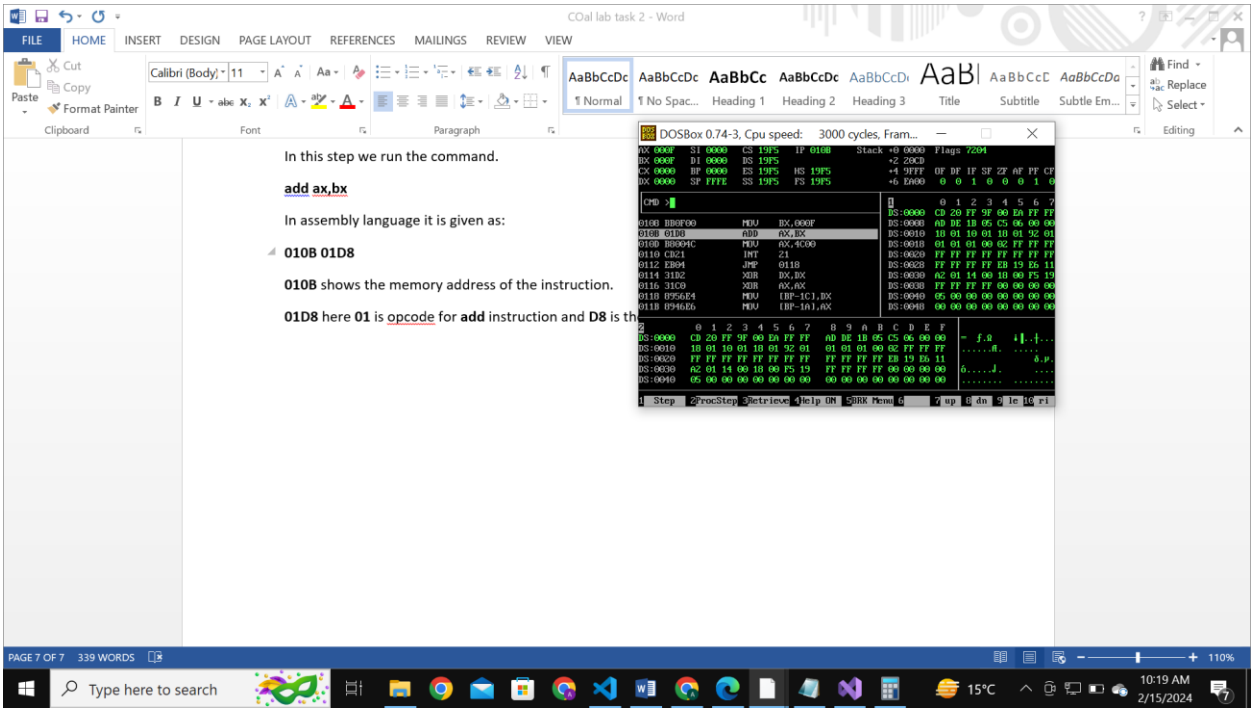
add ax,bx

In assembly language it is given as:

010B 01D8

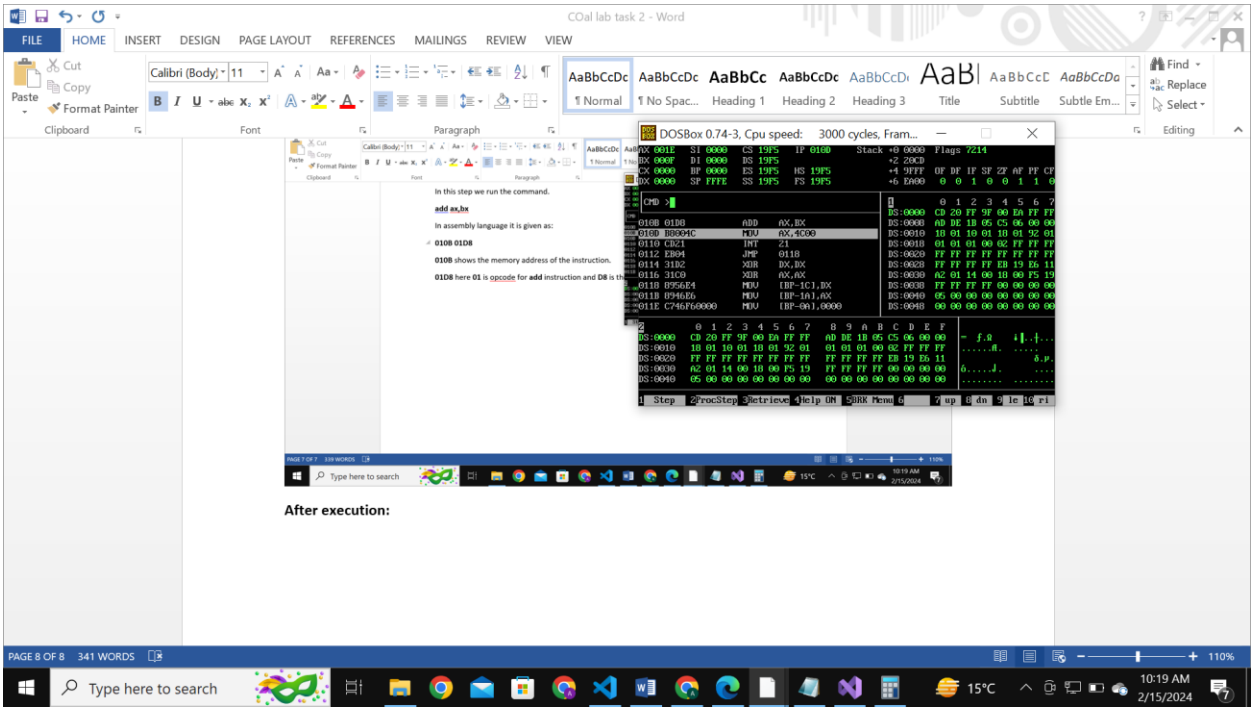
010B shows the memory address of the instruction.

01D8 here **01** is opcode for **add** instruction and **D8** is the operands **ax** and **bx** registers.



After execution:

Here we can see in the top left AX register has got the value 30.



Step 7:

In this step we run the command.

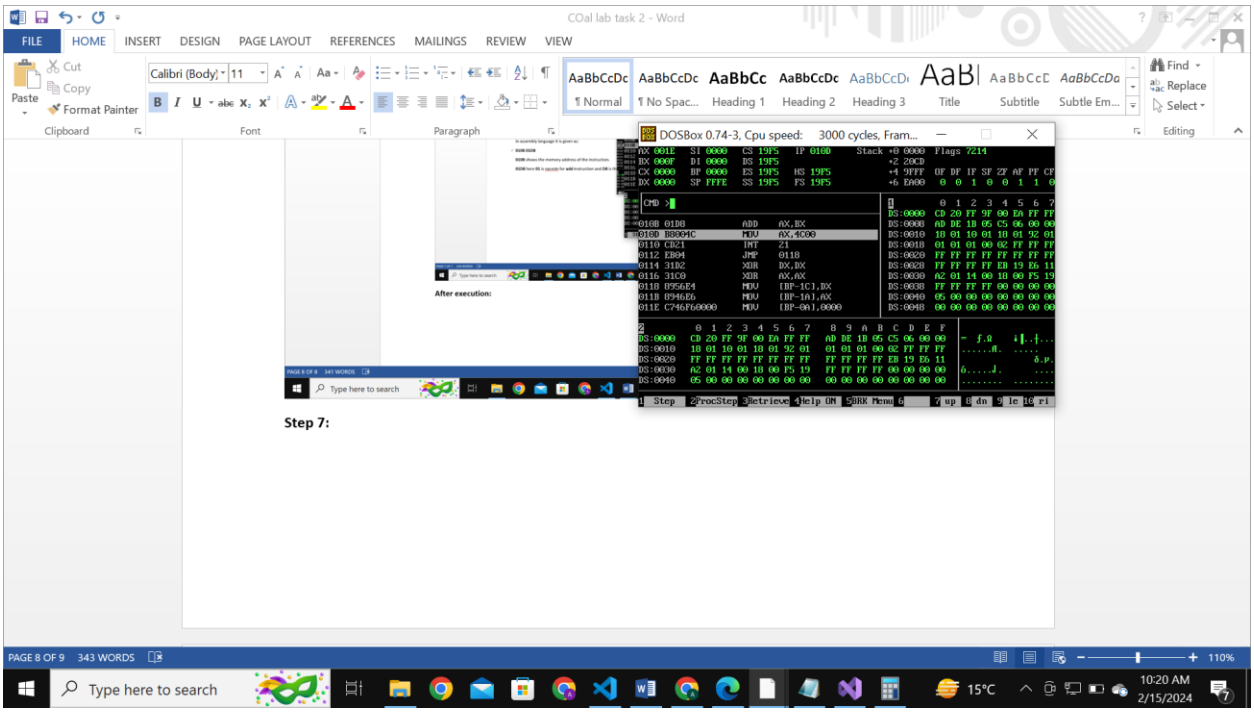
mov ax, 0x4c00

In assembly language it is given as:

010D B8004C

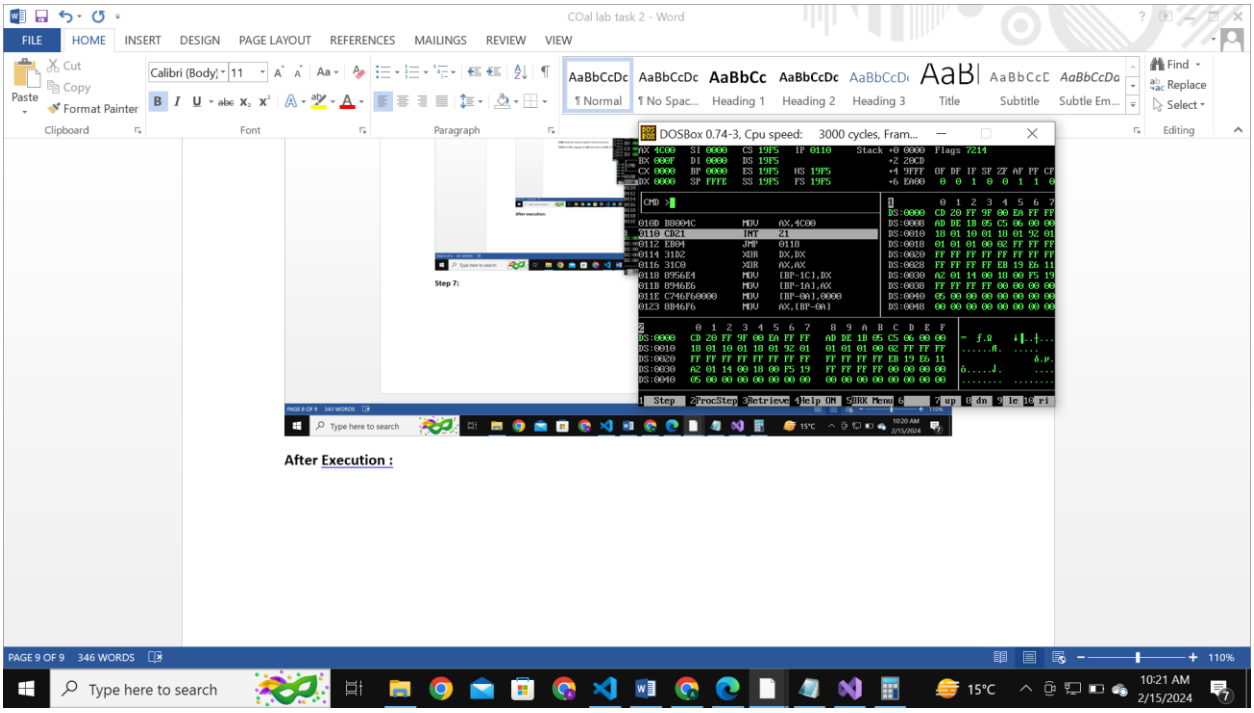
010D shows the memory address of the instruction.

B8004C here **B8** is opcode for **mov** instruction and **004C** is the value to be moved in **ax** register**.**



After Execution :

Here we can see in the top left AX register has got the value **4C00**.



Step 8:

In this step we run the command.

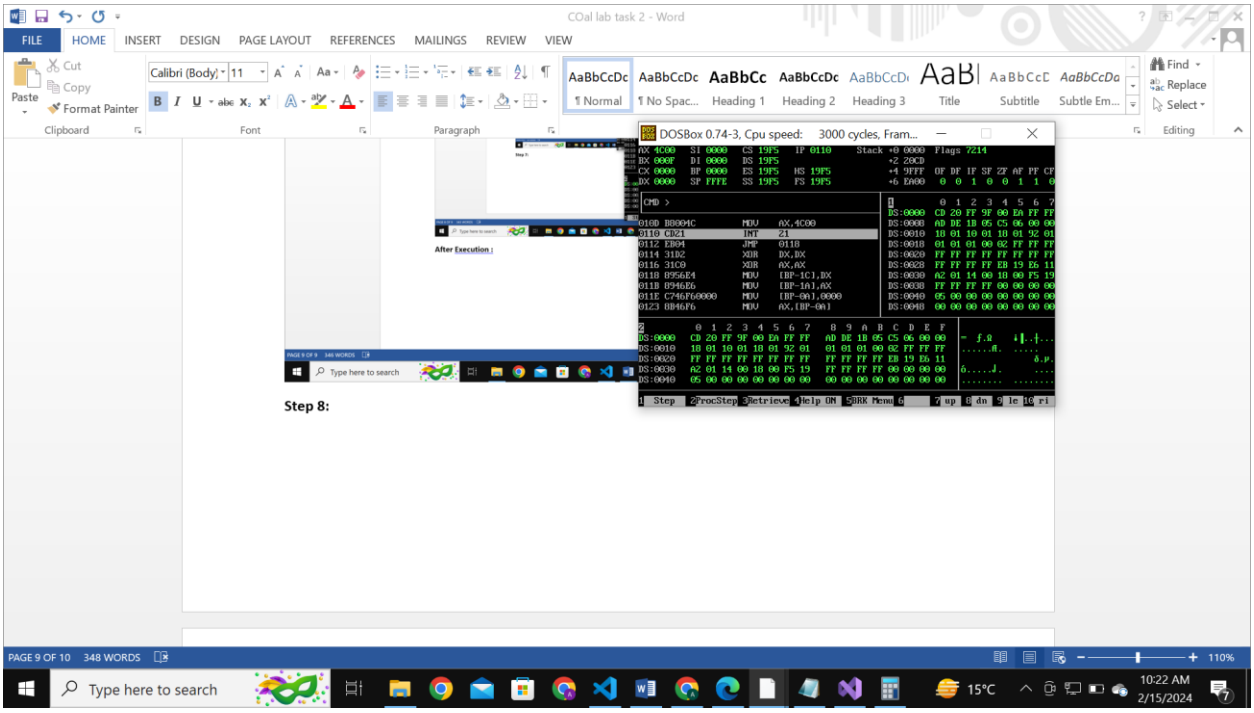
int 0x21

In assembly language it is given as:

0110 CD21

0110 shows the memory address of the instruction.

CD21 here **B8** is opcode for **int (interrupt)** instruction and **21** is the interrupt service routine (ISR) number. Int instruction is a way for programs to interact with the operating system and ask for specific services.



After Execution

Finally the program terminates after **int** instruction.