Experiment No. 02

2.1 Experiment Name

Arithmetic operations (addition, subtraction, division, and multiplication) using memory location

2.2 Objectives

- To understand the structure of Assembly Language programming
- To learn about the microprocessor emulator "Emu 8086" and its operation
- To know how to addition, subtraction, multiplication, division bit-size number using memory location at “Emu8086” using Assembly language
- To learn how to implement program in “MDA 8086” Trainer Board and interconnect it with “Emu 8086”

2.3 Theory

Arithmetic operations are a branch of mathematics that deals with the study of numbers and the numerical operations that are used in all other branches of mathematics. Assembly-language programming is used to implement it. A low-level programming language for microprocessors and other programmable devices is assembly language. A microprocessor executes a set of instructions written in machine language, directing the processor what to do. A 16-bit microprocessor trainer board, the MDA-8086 Kit. It is made up of various units. Arithmetic instructions are classified into four types: addition, subtraction, multiplication, and division.

- The ADD instruction modifies the contents of another register (destination) or memory address by adding immediate data or the contents of a memory location specified in the instruction. The outcome is saved in the destination operand.
- The SUB instruction modifies the contents of another register (destination) or memory address by subtracting immediate data or the contents of a memory location specified in the instruction.
- The MUL operation is used to multiply two unsigned numbers, while the IMUL procedure is used to multiply two signed numbers. Based on the number of bits, multiplication is split into three types.
- The DIV operation is used for division of two unsigned numbers or operands.

2.4 Apparatus

- Emu 8086 - Microprocessor Emulator

2.5 Emulator Code & Output

- **Addition**

```
DTA DW 5,2,1,9,8
MOV SI, OFFSET DTA
MOV AX, [SI]
MOV BX, [SI+6H]
ADD AX, BX
DAA
```

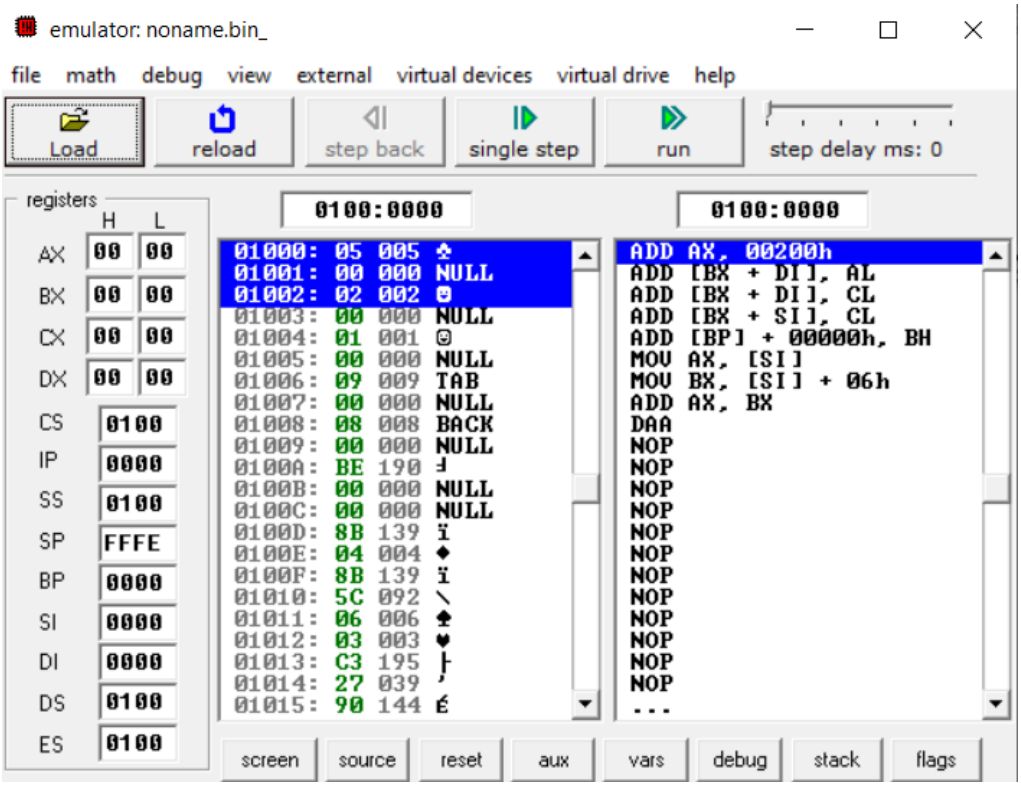

Fig 2.1: Emulator output for Assembly Language Program (After Step 0)

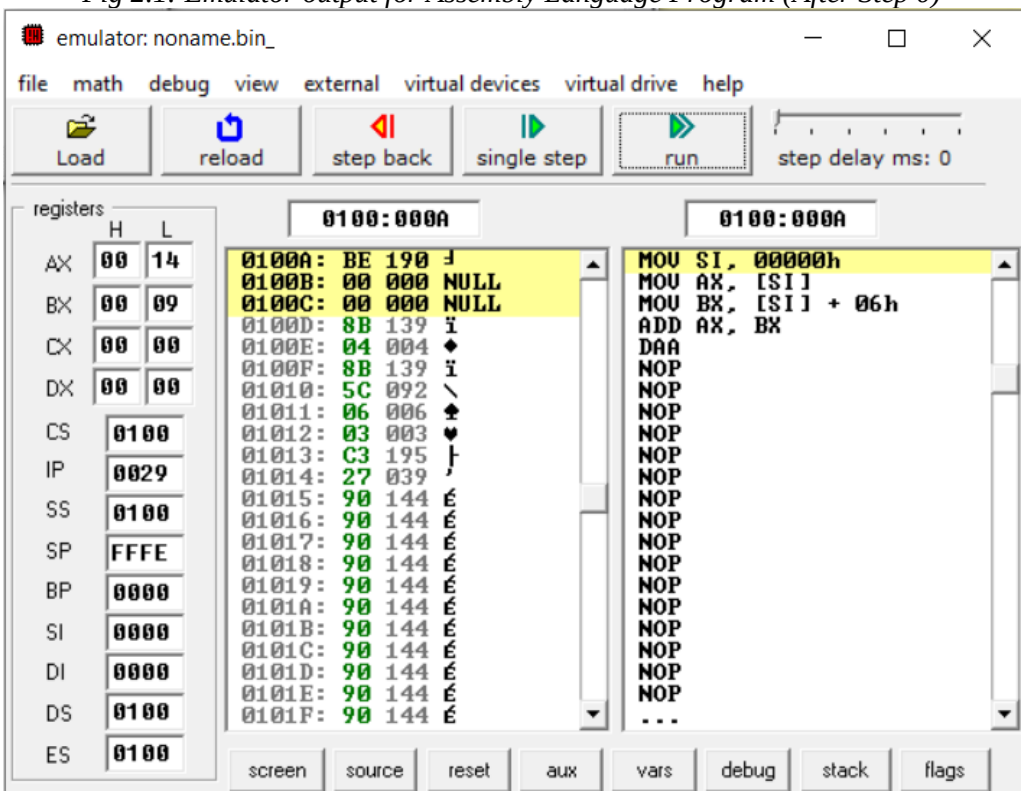

Fig 2.2: Emulator output for Assembly Language Program (After Step 1)

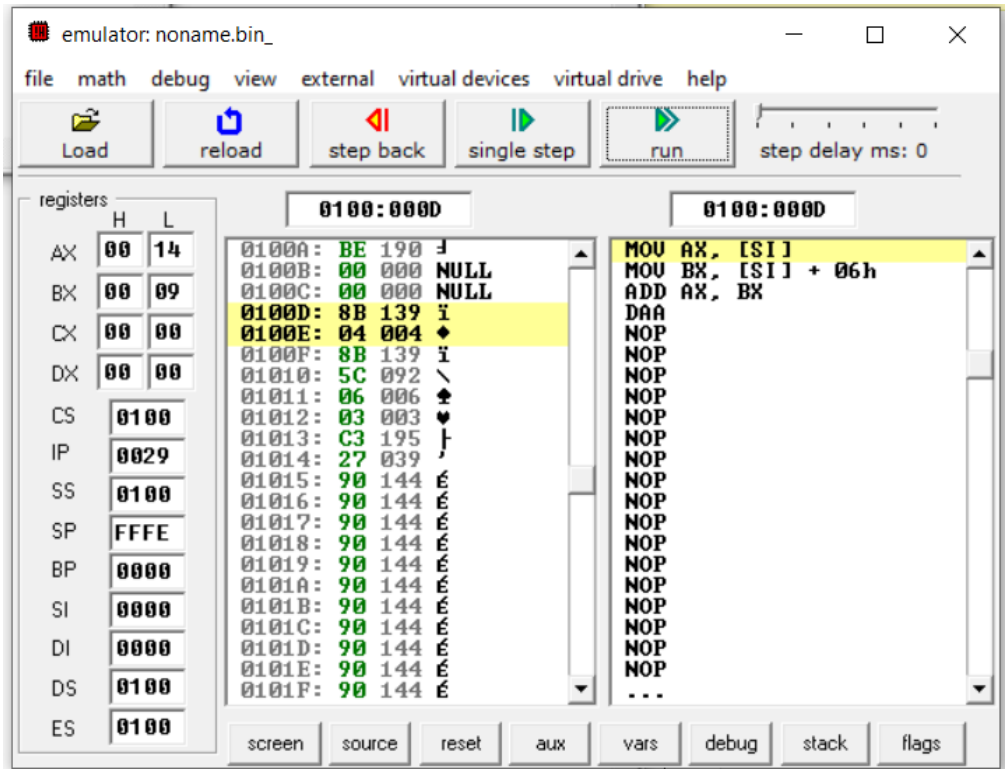

Fig 2.3: Emulator output for Assembly Language Program (After Step 2)

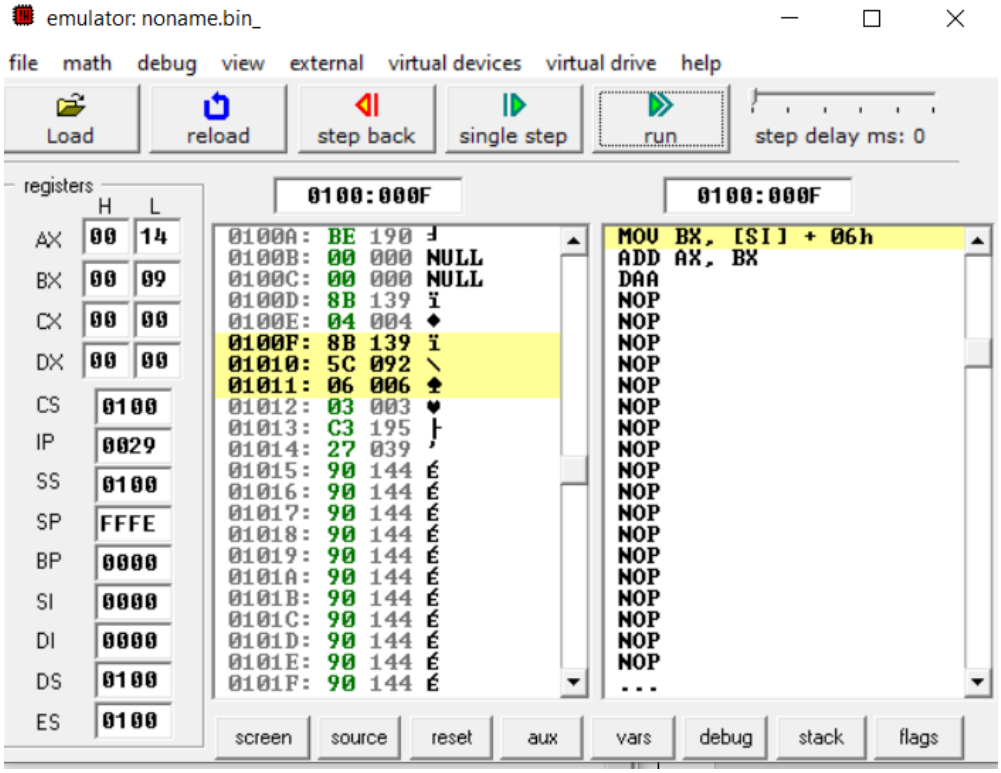

Fig 2.4: Emulator output for Assembly Language Program (After Step-3)

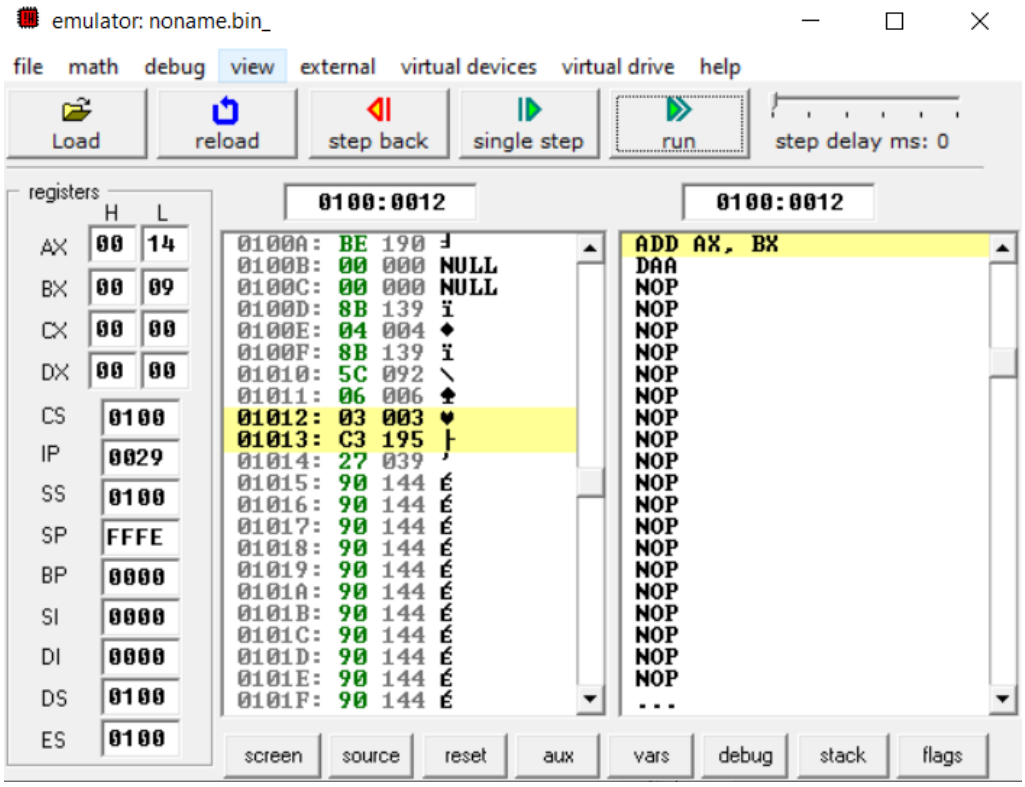

Fig 2.5: Emulator output for Assembly Language Program (After Step-4)

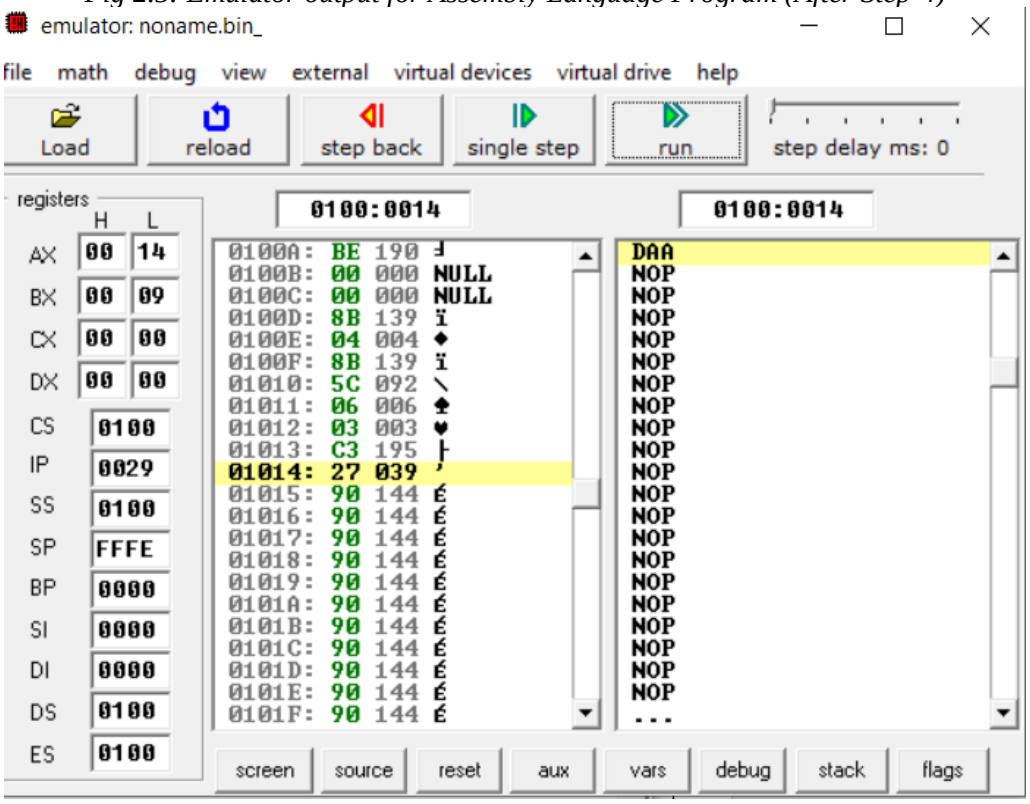

Fig 2.6: Emulator output for Assembly Language Program (After Step-5)

- **Subtraction**

DTA DW 5,2,1,9,8
MOV SI, OFFSET DTA
MOV AX, [SI+6H]

MOV BX, [SI+2H]
ADD AX, BX
DAA

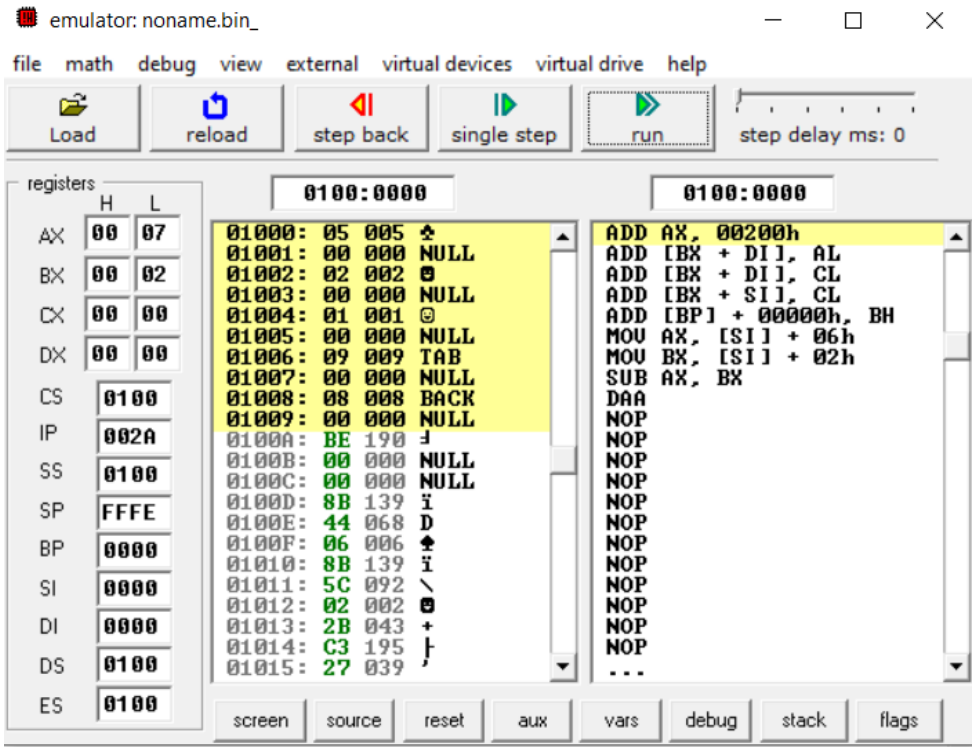

Fig 2.7: Emulator output for Assembly Language Program (After Step 0)

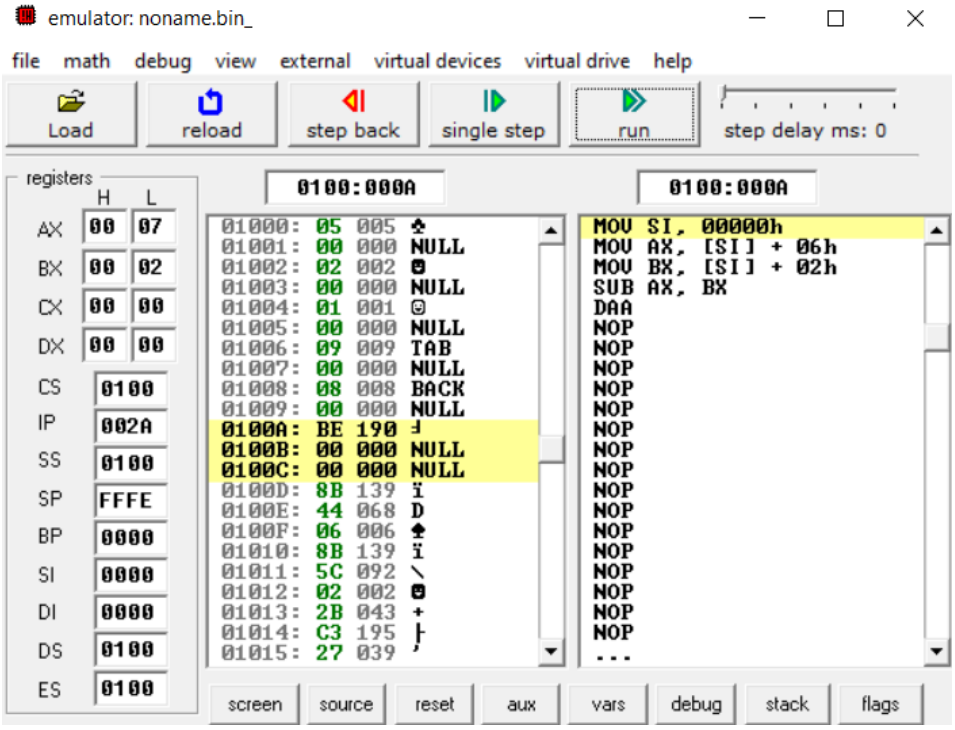

Fig 2.8: Emulator output for Assembly Language Program (After Step 1)

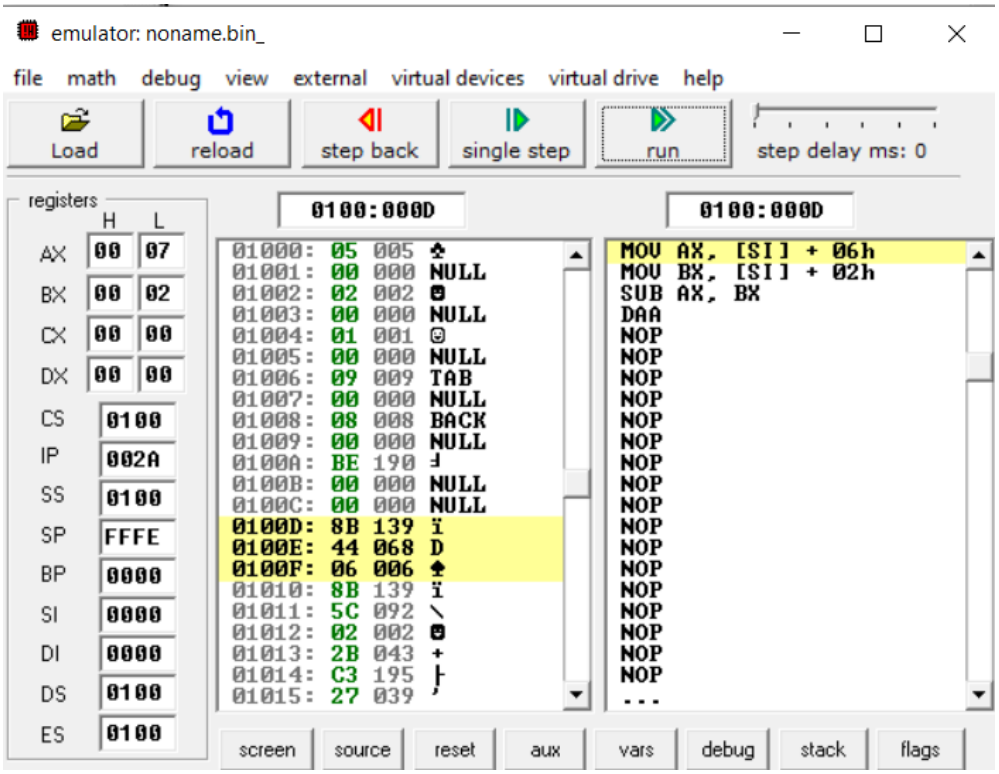

Fig 2.9: Emulator output for Assembly Language Program (After Step 2)

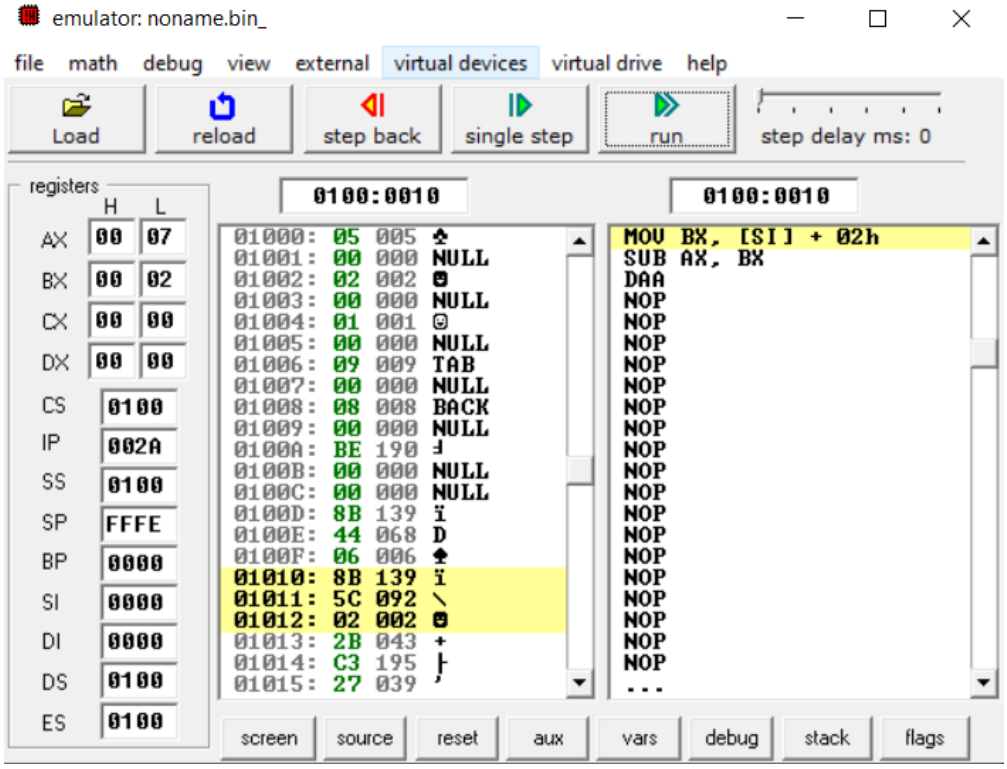

Fig 2.10: Emulator output for Assembly Language Program (After Step 3)

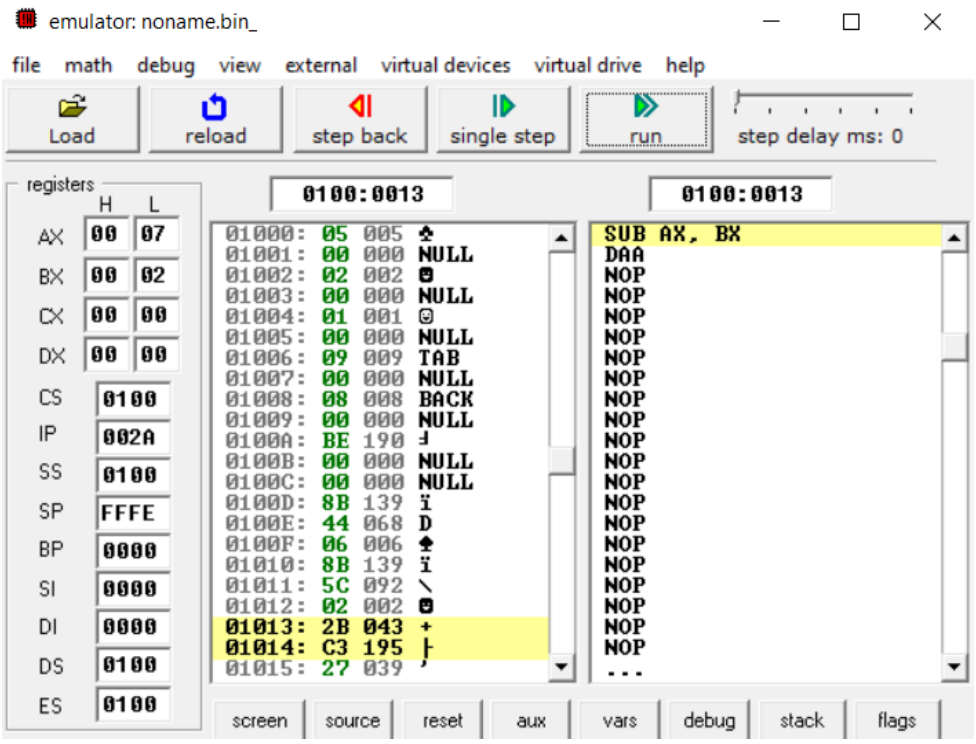

Fig 2.11: Emulator output for Assembly Language Program (After Step 4)

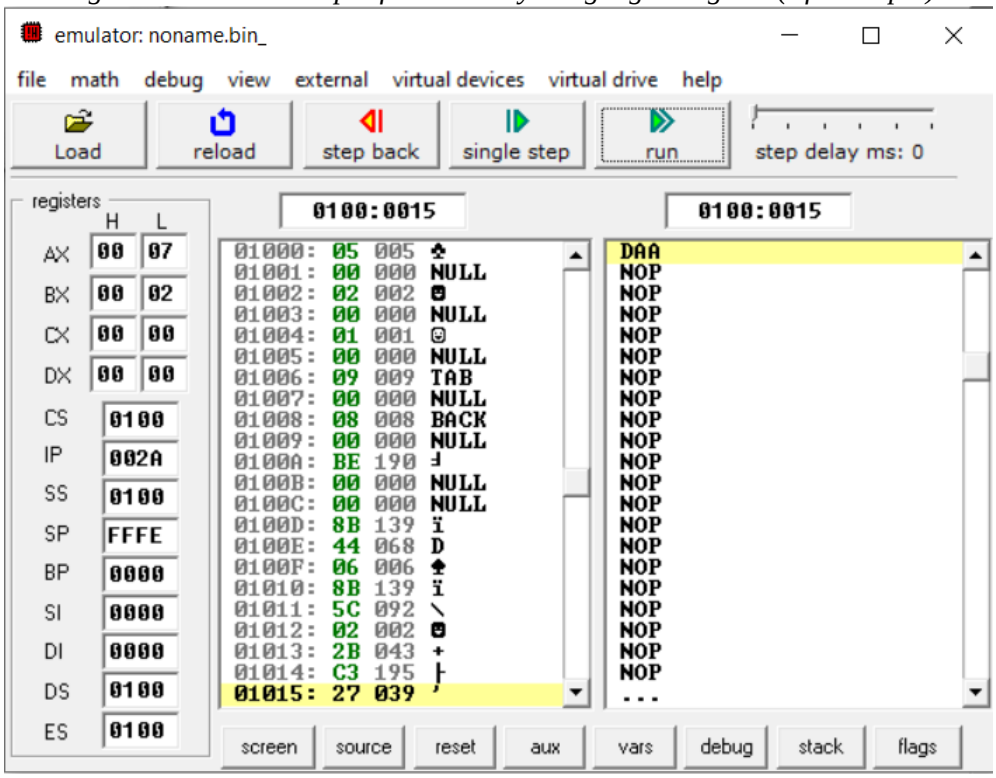

Fig 2.12: Emulator output for Assembly Language Program (After Step 5)

- **Multiplication**

```
MOV AL, 4
MOV SI, 0120H
MOV [SI], 3
MUL [SI]
```

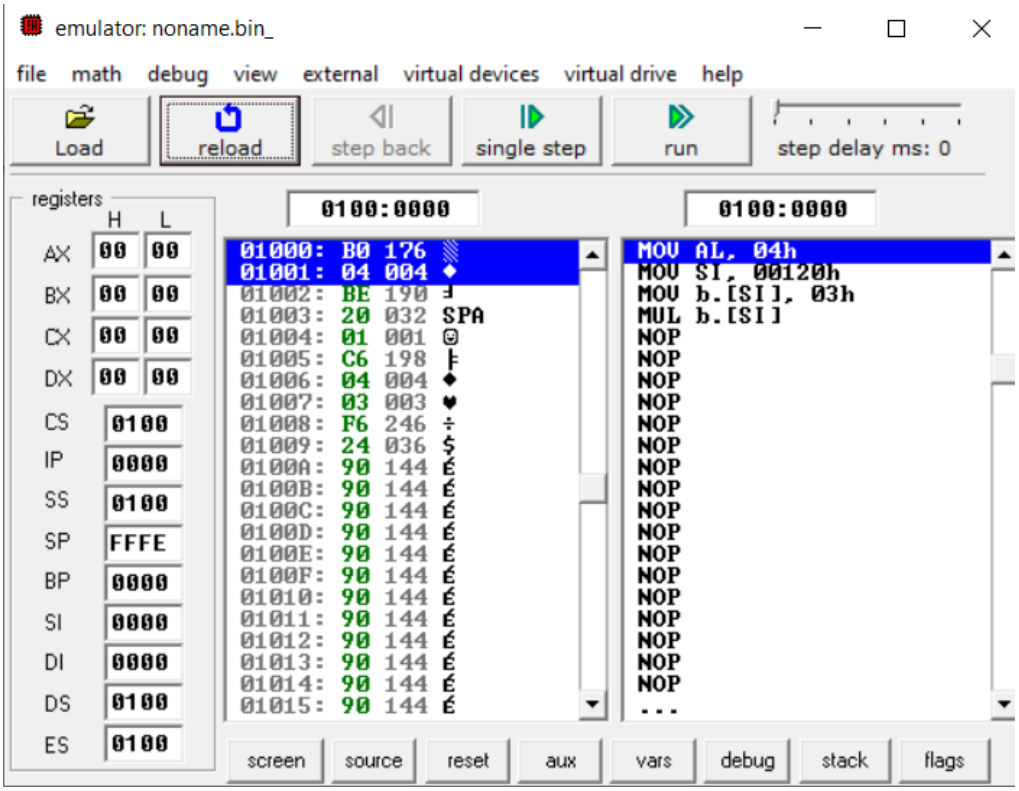

Fig 2.13: Emulator output for Assembly Language Program (After Step 0)

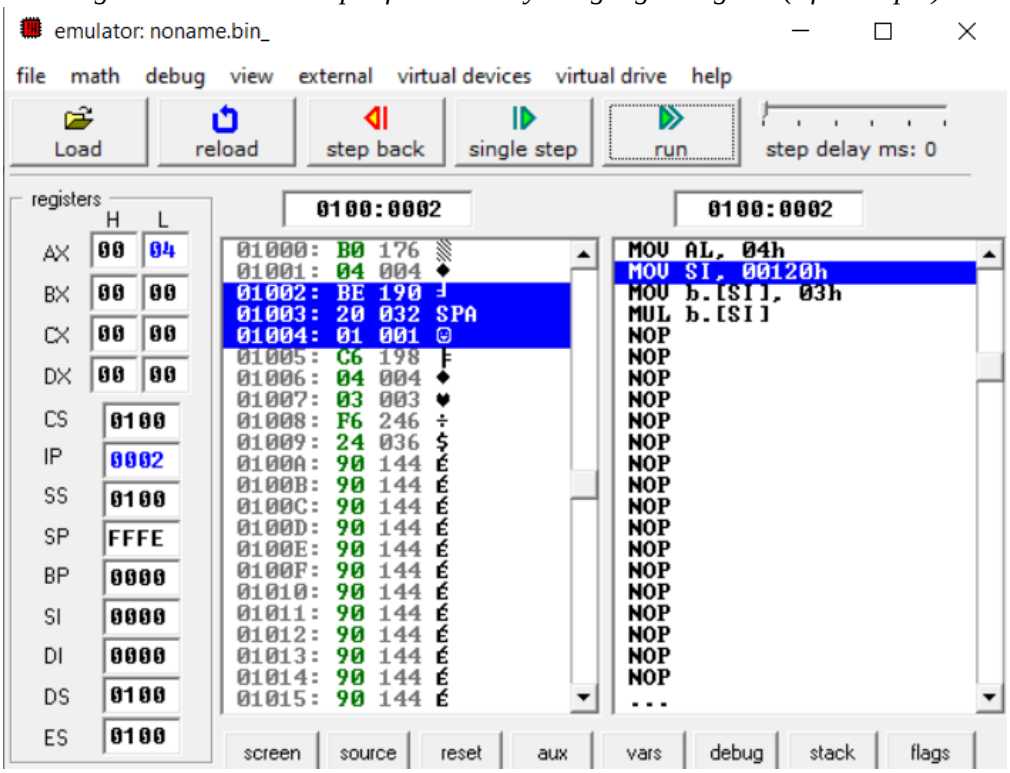

Fig 2.14: Emulator output for Assembly Language Program (After Step 1)

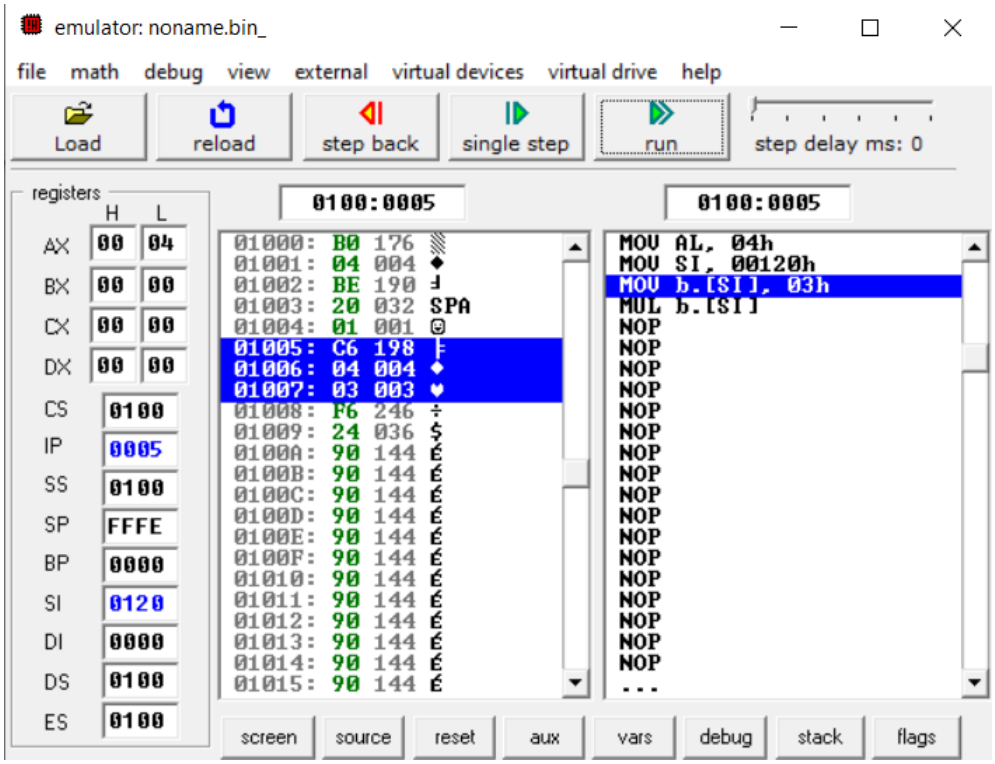

Fig 2.15: Emulator output for Assembly Language Program (After Step 2)

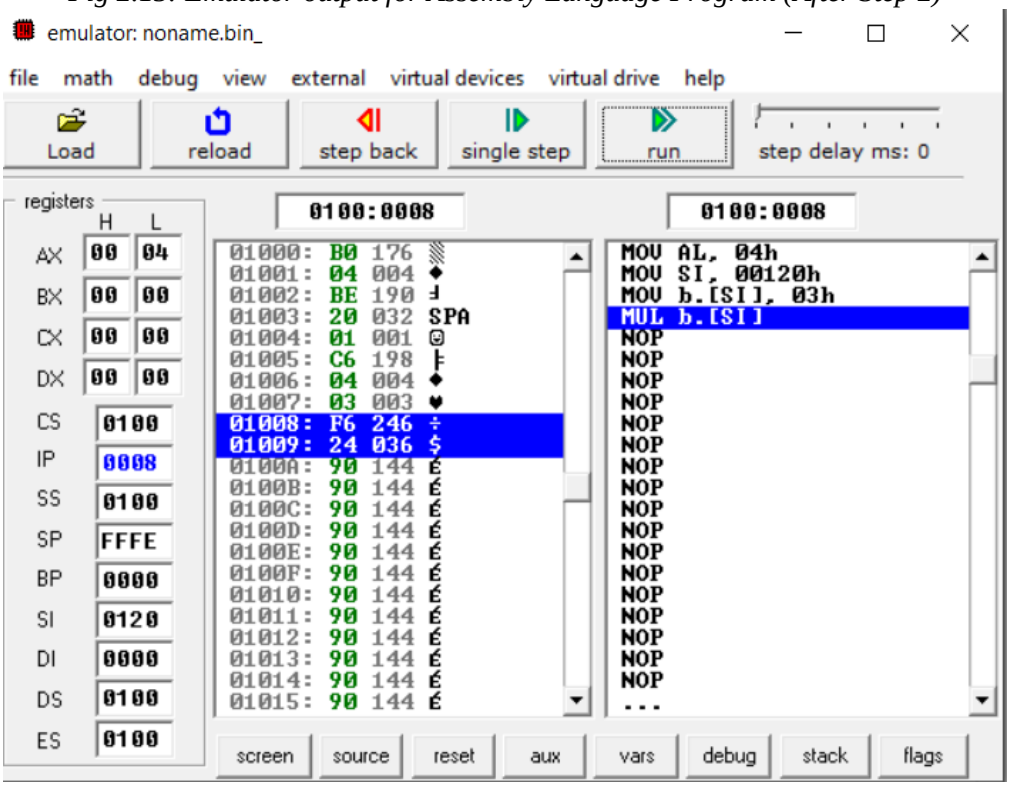

Fig 2.16: Emulator output for Assembly Language Program (After Step 3)

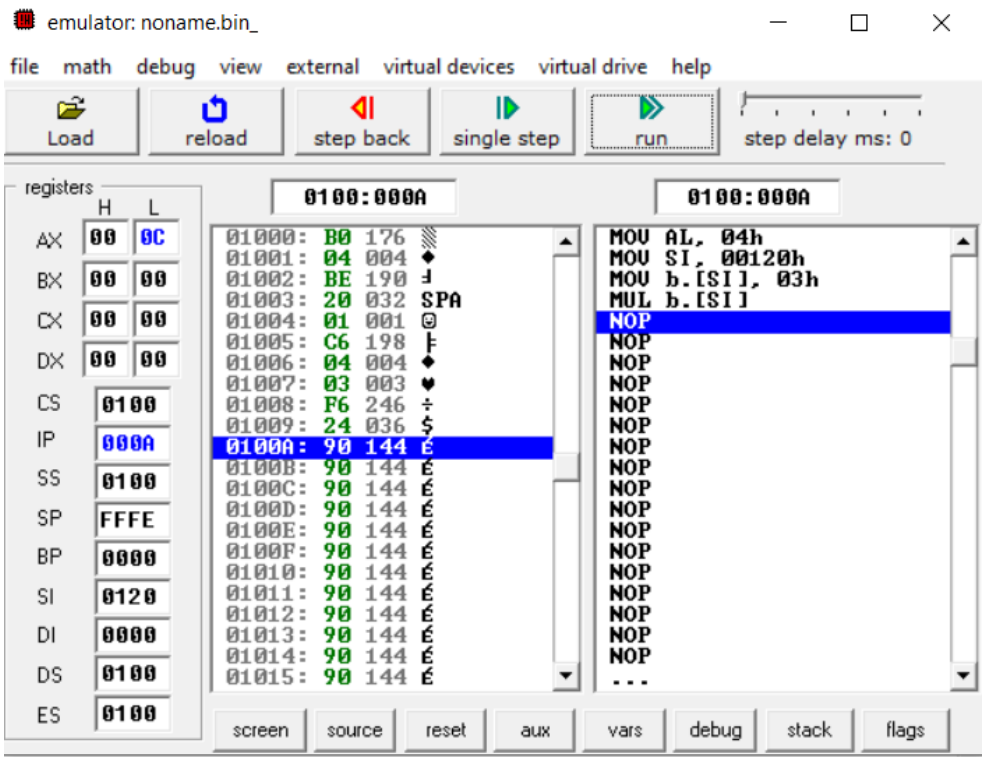

Fig 2.17: Emulator output for Assembly Language Program (After Step 4)

- **Division**

MOV AL, 4
MOV SI, 0120H
MOV [SI], 3
DIV [SI]

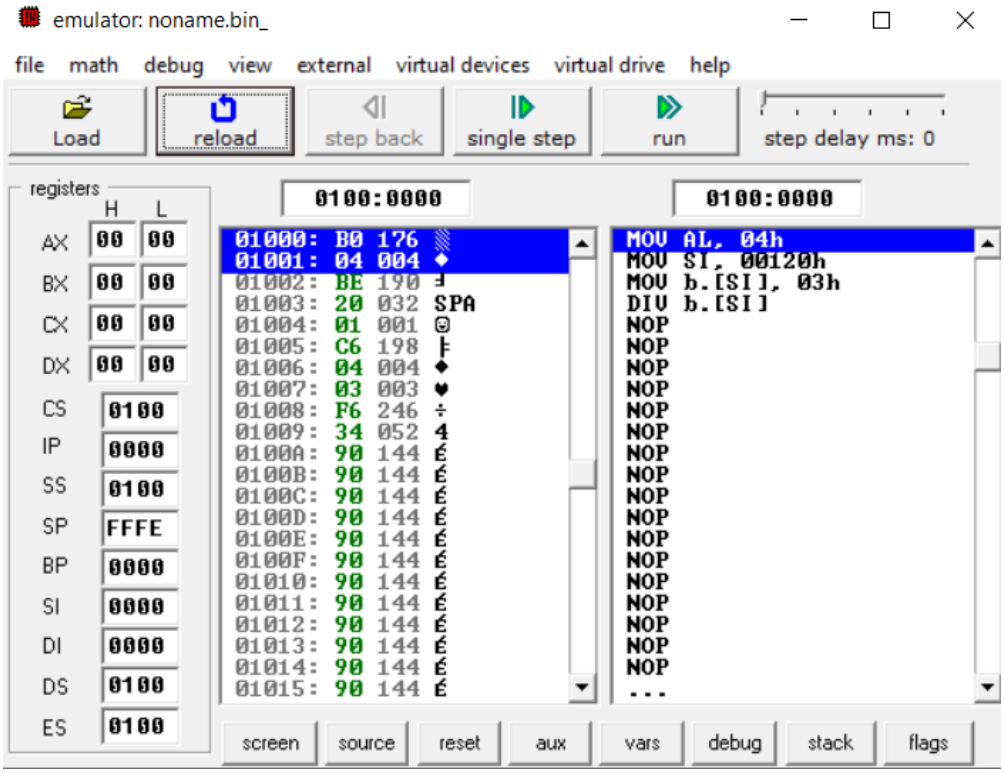

Fig 2.18: Emulator output for Assembly Language Program (After Step 0)

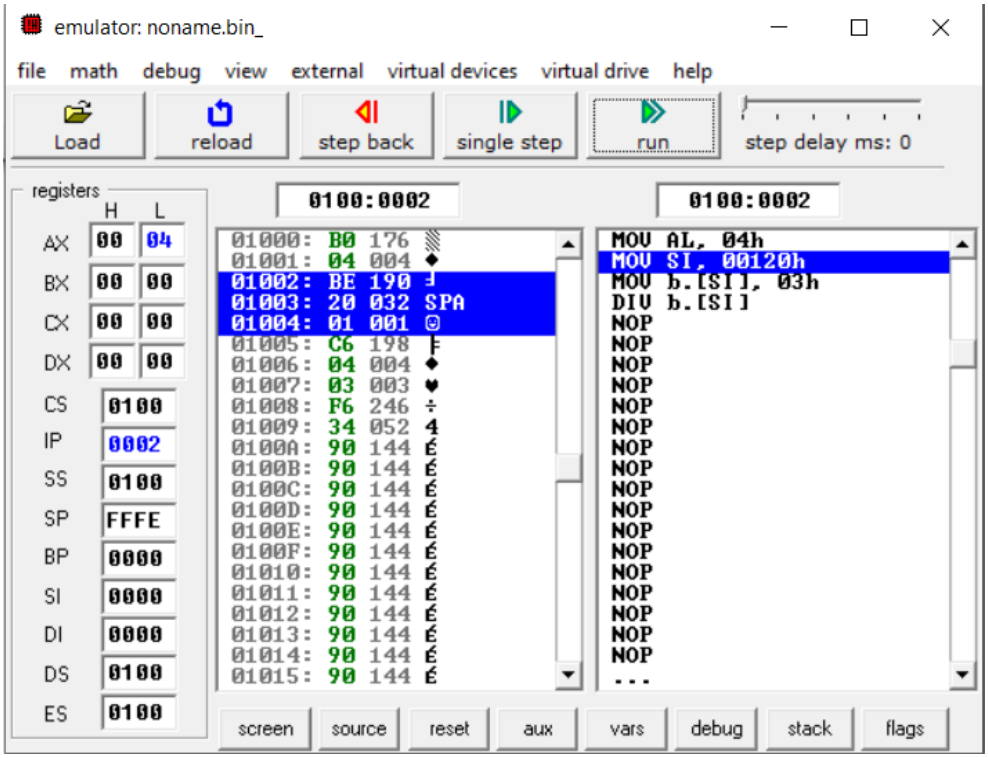

Fig 2.19: Emulator output for Assembly Language Program (After Step 1)

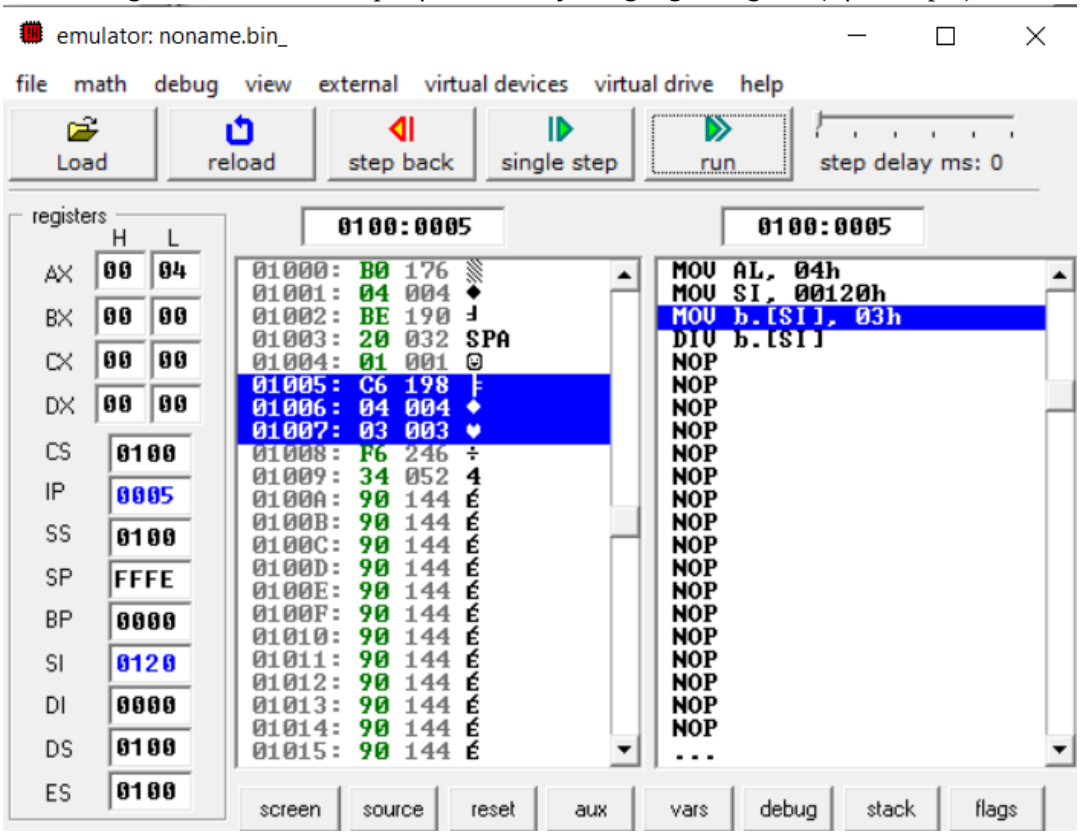

Fig 2.20: Emulator output for Assembly Language Program (After Step 2)

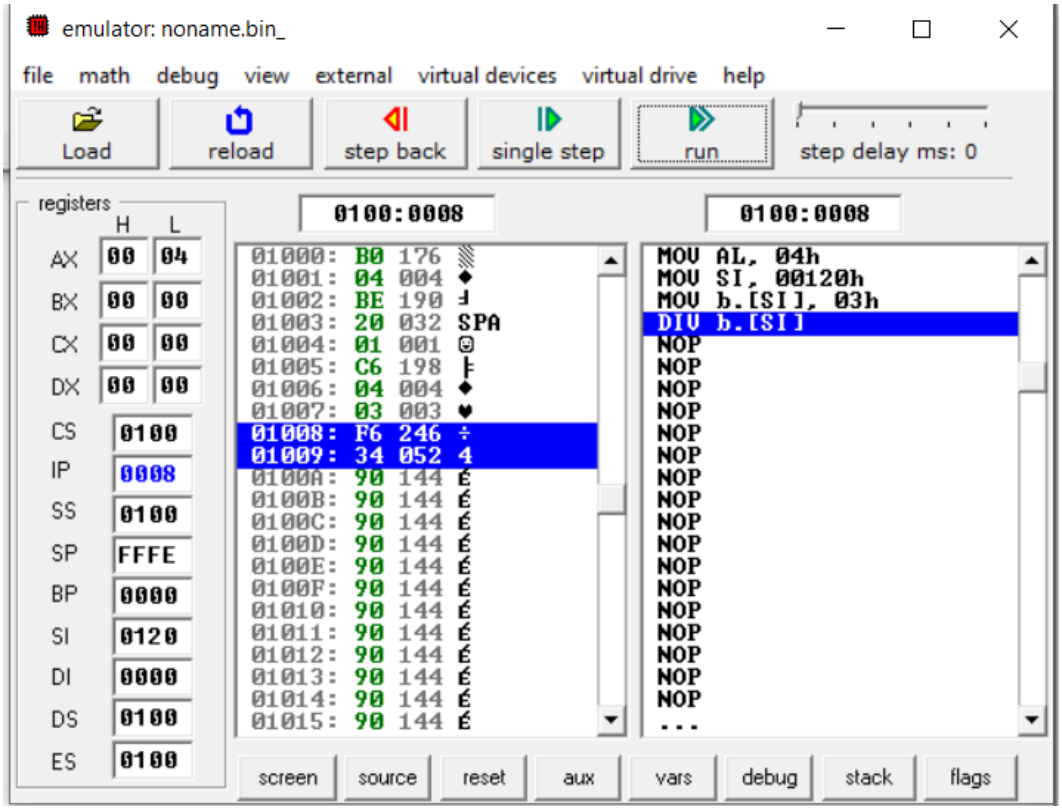

Fig 2.21: Emulator output for Assembly Language Program (After Step 3)

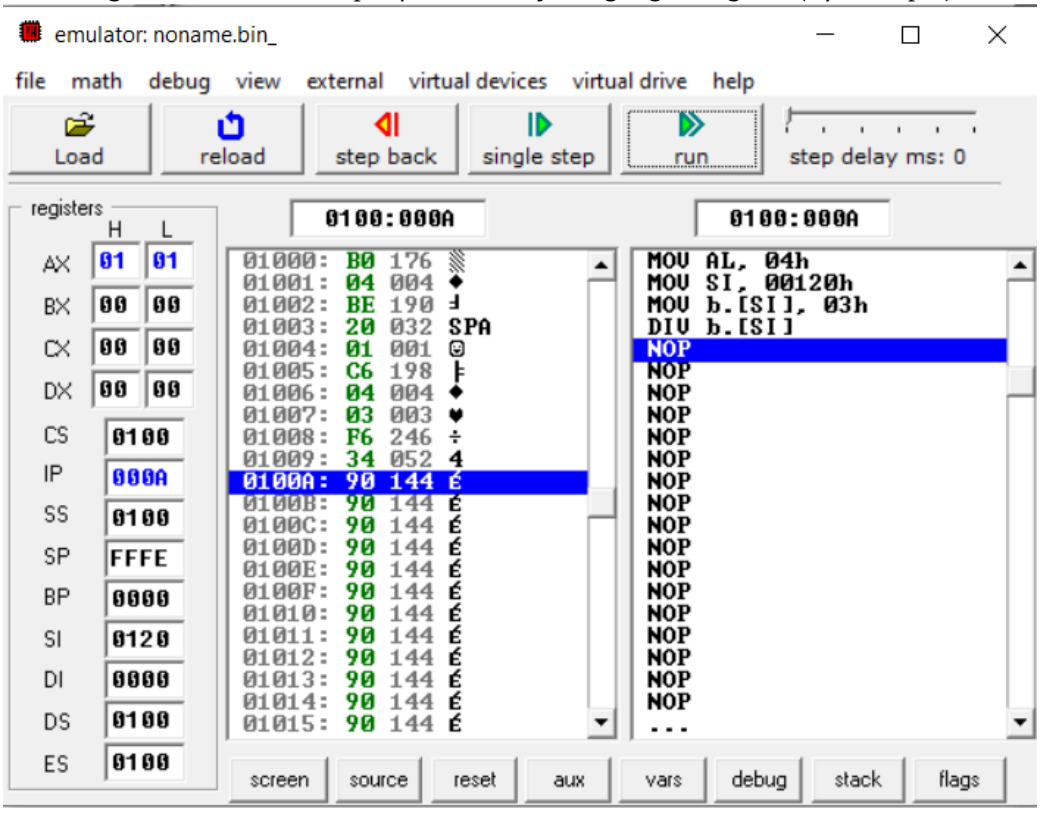

Fig 2.22: Emulator output for Assembly Language Program (After Step 4)

2.6 Discussion & Conclusion

In this experiment, the MDA-8086 kit used includes a central processor unit and other components. During this experiment, we became acquainted with the MDA-8086 microprocessor kit and the EMU program.

The assembly language is written on EMU 8086 to conduct arithmetic operations such as addition, subtraction, division, and multiplication, in various locations in our memory. This is due to the fact that we included the carry from the prior data transaction. MOV instructions were used to maneuver information from a memory location to a register. Then, by running the imitator step by step, the specified output was obtained. Thus, the offset was observed because of the RAM access. The experiment's aims are met, and the results are satisfactory. Finally, the experiment can be described as productive and successful.