**Lab 8: Simulate Subtract Operation**

# Learning Objectives:

* Understand Two’s Complement for subtraction.
* Perform subtraction using bitwise operations.
* Analyze step-by-step execution of subtraction in assembly.
* Apply low-level programming for arithmetic operations.

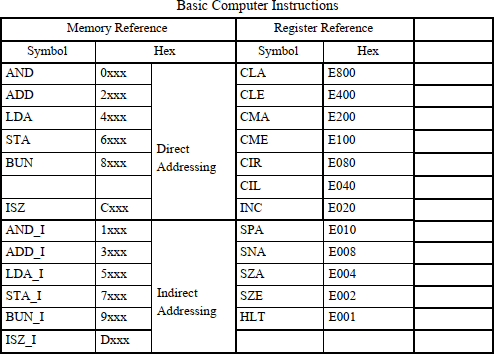
# Essential Tools in Our Lab:

* **Computer System:** The main machine
* **CPU Sim**: CPU Sim is a tool used for simulating simple CPU architectures, helping students understand processor design, instruction execution, and debugging.

**Program**

|  |  |
| --- | --- |
| START: |  |
| INP | ; Take first input from the user and store it in the accumulator |
| STA NUM | ; Store the value from the accumulator into memory location NUM |
| INP | ; Take second input from the user and store it in the accumulator |
| CMA | ; Complement (invert) all bits of the second input (Two's complement preparation for subtraction) |
| INC | ; Add 1 to the complemented value (Completing Two's complement to get negative of the second input) |
| ADD NUM | ; Add the stored first input (NUM) with the negated second input (Effectively performing subtraction: First Input - Second Input) |
| OUT | ; Output the result of the subtraction operation |
| HLT | ; Halt execution |
|  |  |
| NUM: .data 1 0 | ; Memory location to store the first input, initially set to 0 |

# Basic Machine Instructions

****

**Define Microinstructions for CMA and INC**

Since we previously created microinstructions for addition, we will follow a similar approach but add **CMA and INC** instructions.

1. **Navigate to the Microinstructions Section** in your simulator/tool.

# Define CMA (Complement Accumulator) Microinstruction:

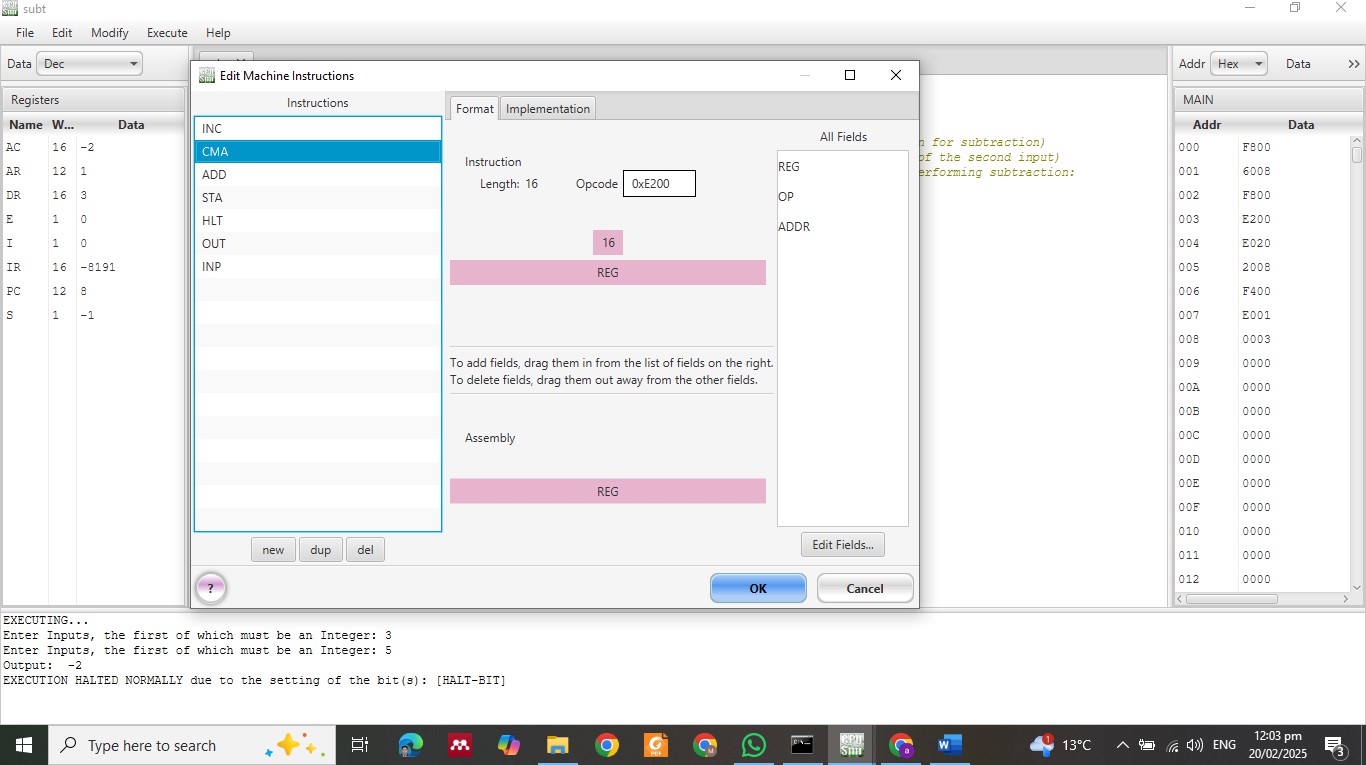
* + This instruction will perform **bitwise complement** (logical NOT) on the Accumulator (AC).
  + **Implementation:** AC ← AC'

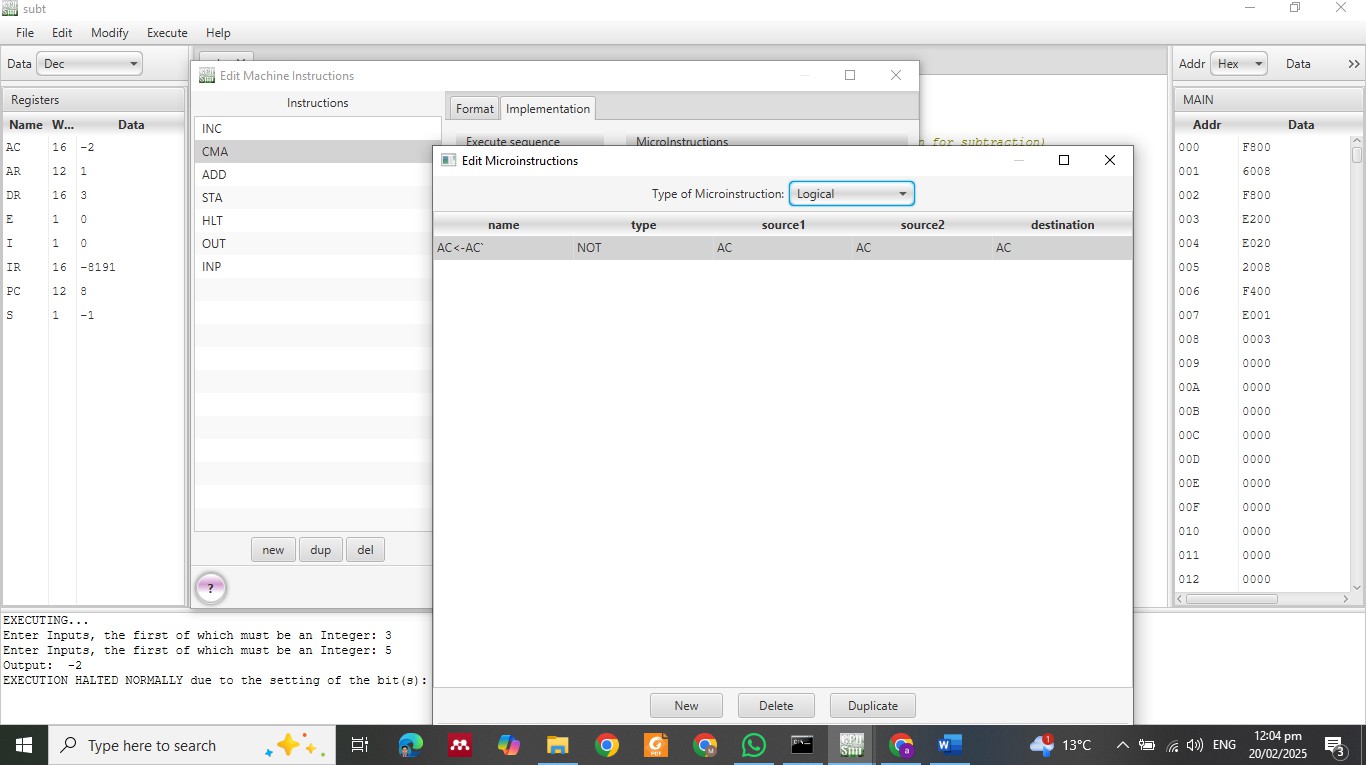
# Define INC (Increment Accumulator) Microinstruction:

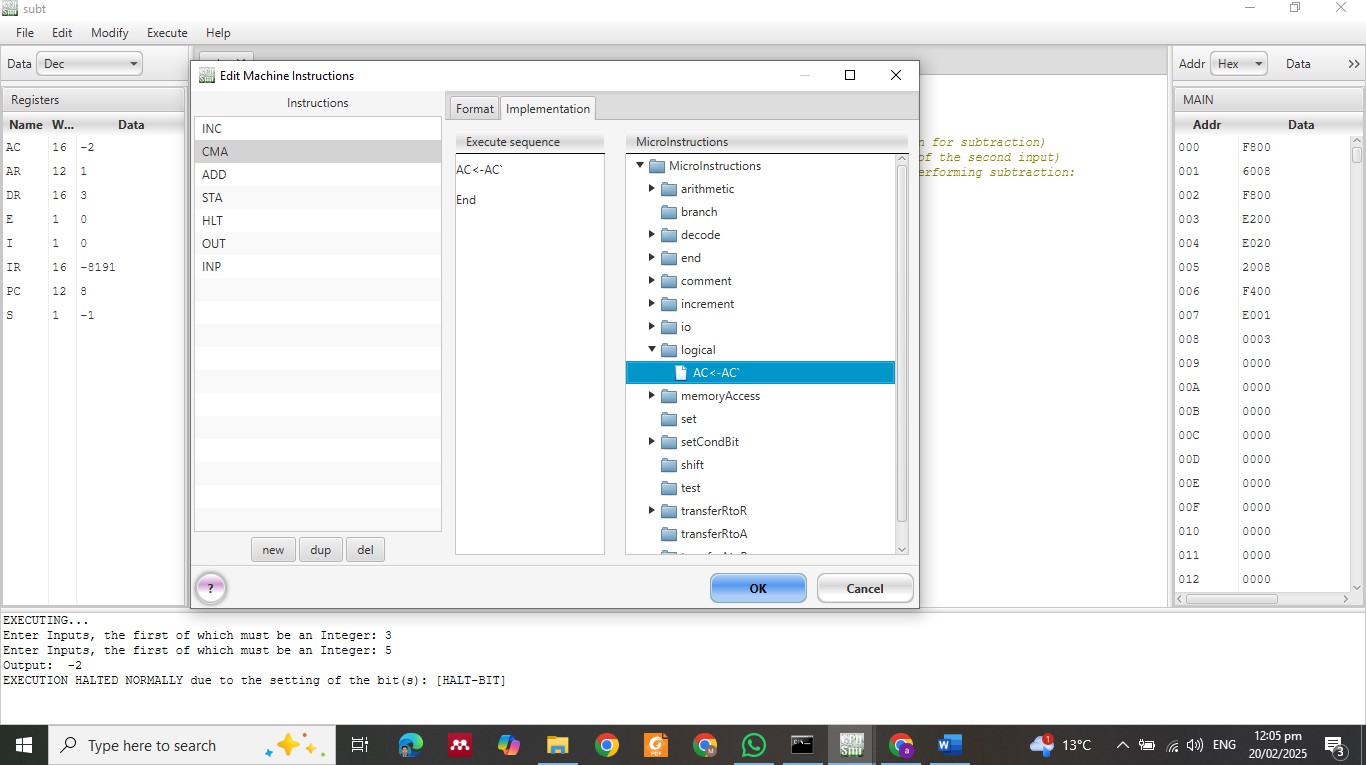
* + This instruction increments the value in the Accumulator by 1.
  + **Implementation:** AC ← AC + 1

# Add CMA Instruction to the Machine Instruction Set

* Set the **Opcode** for CMA as **E200**.
* Set it as a **Register Instruction**.
* Go to **Implementation** and select **Logical Instruction** with the operation **AC ← AC'**.
* Define a **Sequence Instruction** for execution.

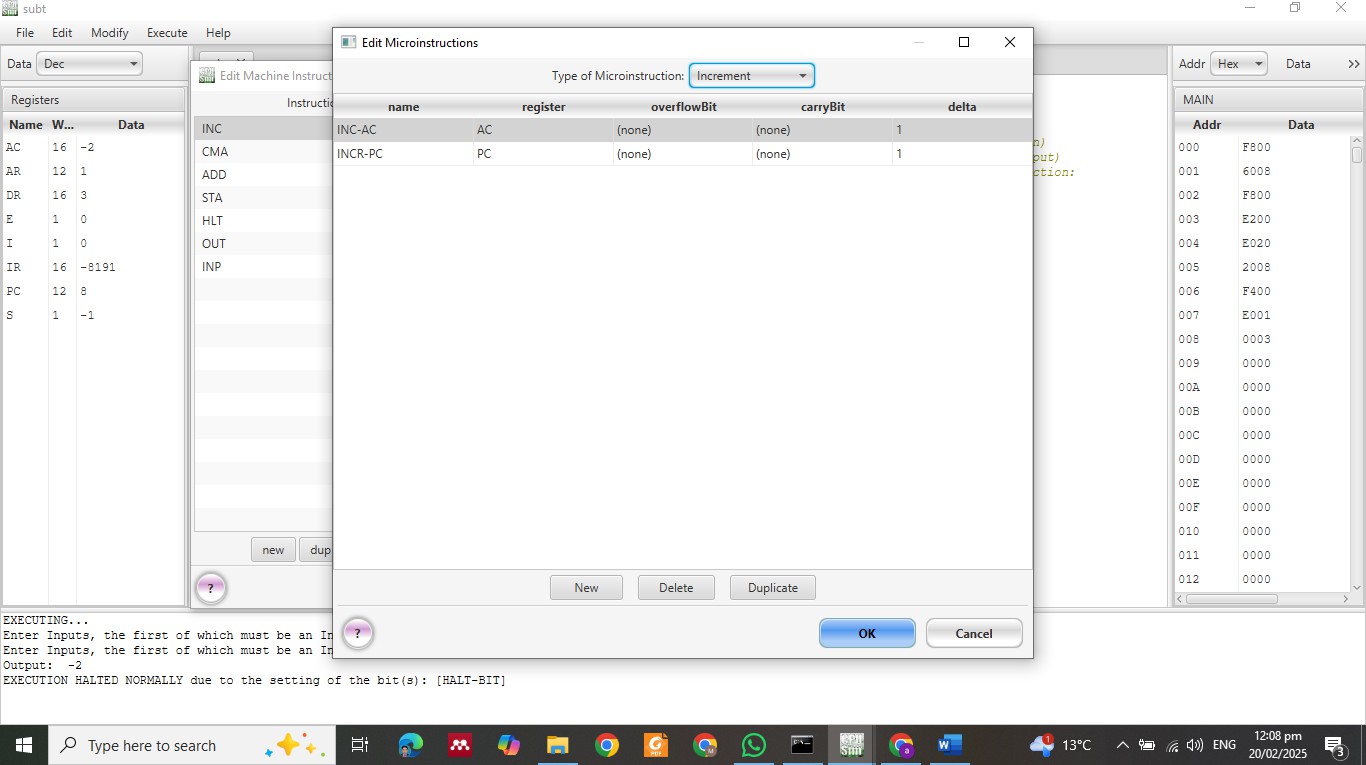
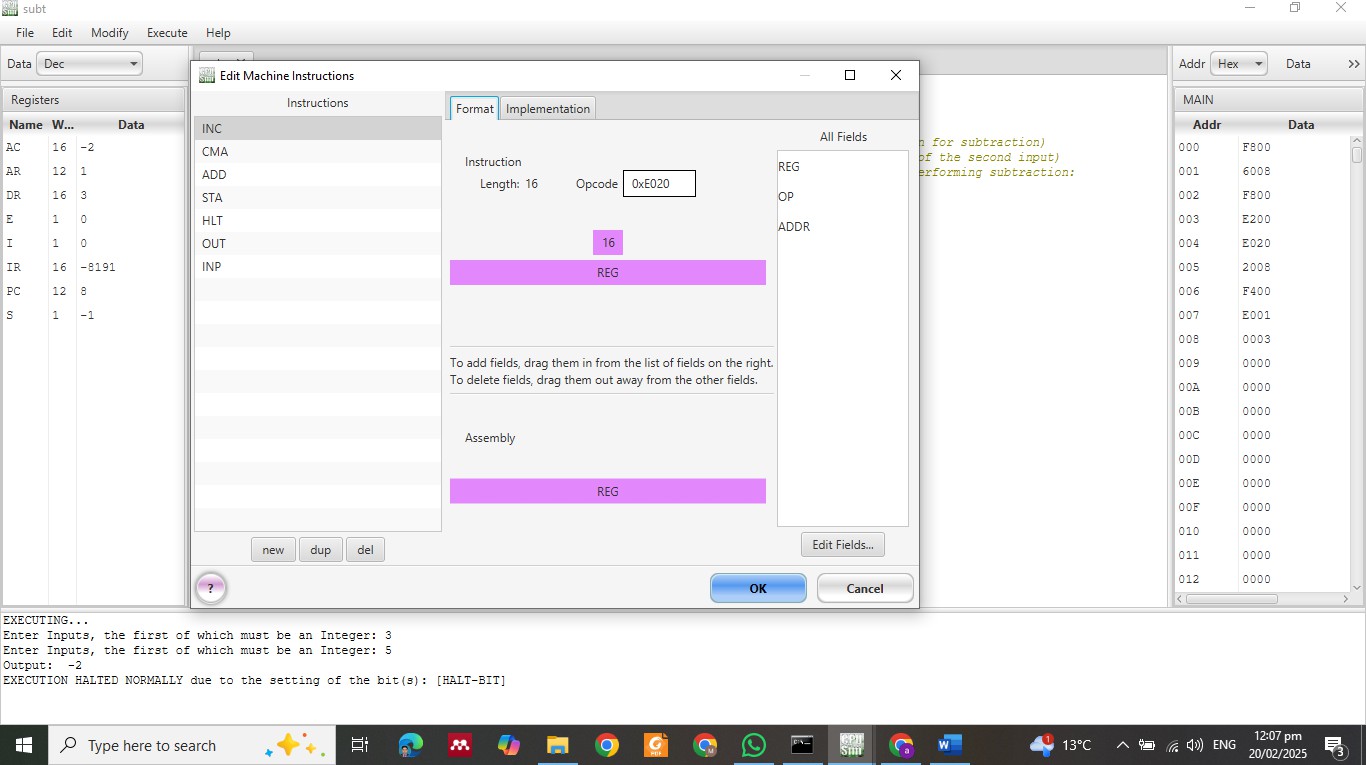


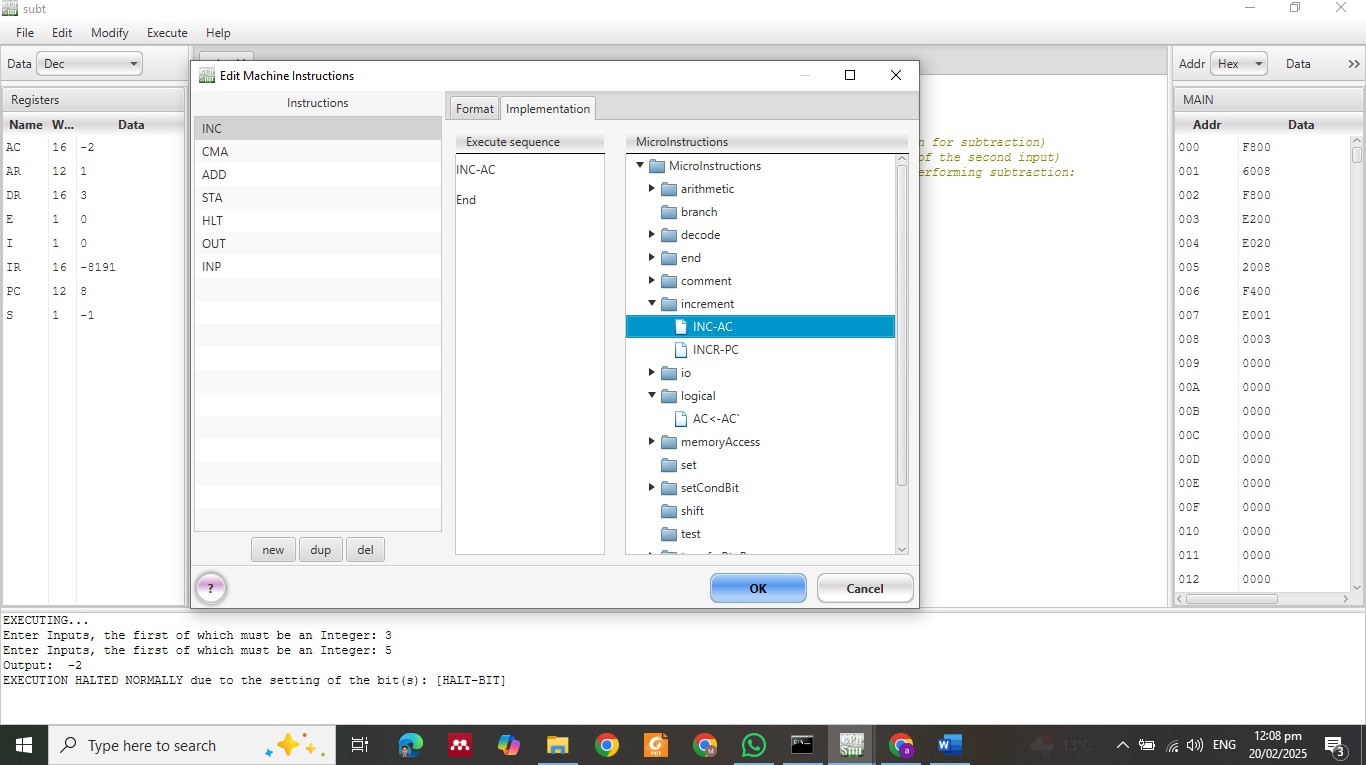


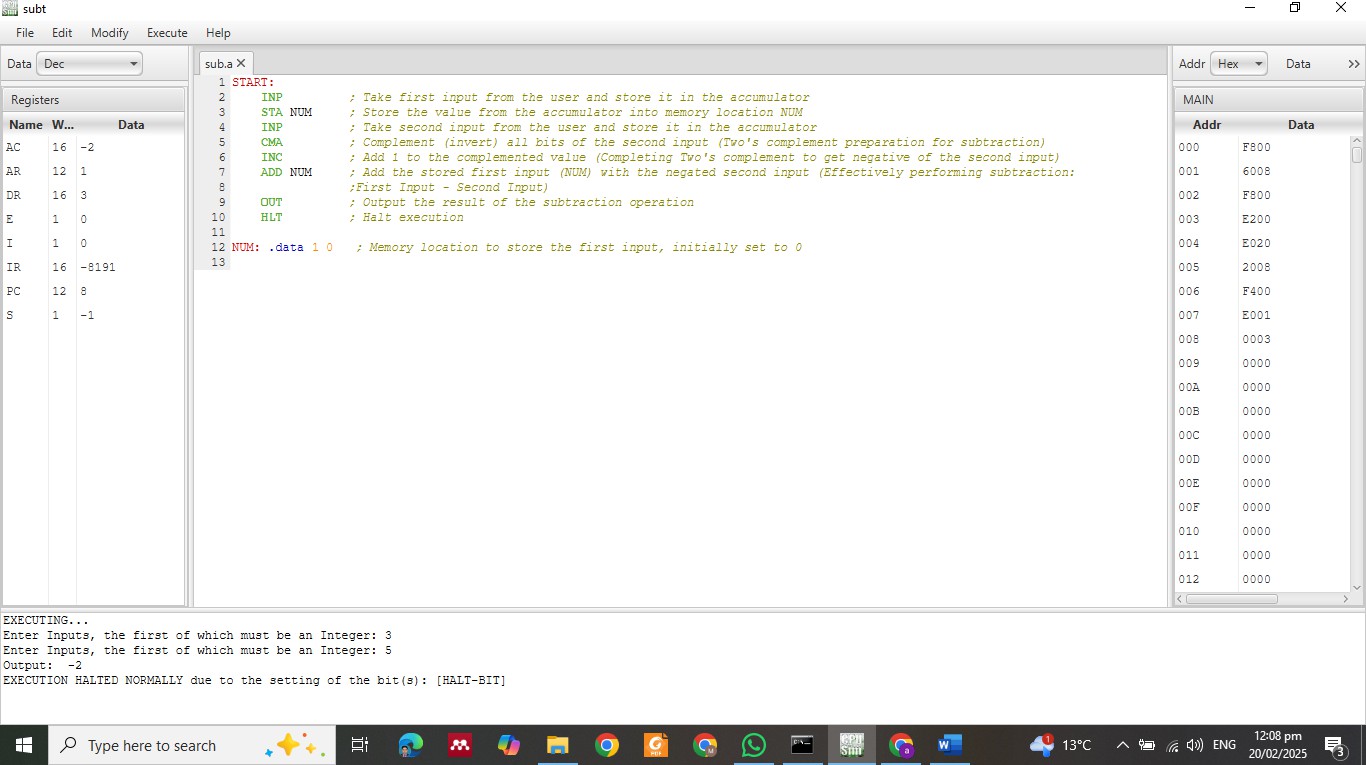


# Add INC Instruction to the Machine Instruction Set

* Set the **Opcode** for INC as **E020**.
* Set it as a **Register Instruction**.
* Select **Increment Instruction** with operation **AC ← AC + 1**.
* Define a **Sequence Instruction** for execution.







**LAB TASKS**

**Task 1:**

Accept **three numbers** as input and subtract them.

### **Task 2:**

### **Subtract Two Numbers Stored in Memory (Predefined Values).**

**Task 3:**

**Store the result of subtraction in a memory location.**