Lab 11: Simulate ISZ Operation

Learning Objective:

* Understand the Increment and Skip if Zero (ISZ) operation.
* Define and implement ISZ microinstructions in CPU Sim.
* Observe how ISZ increments a value and skips the next instruction if zero.
* Apply ISZ in loop counters and conditional execution.

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.

ISZ (Increment and Skip if Zero) Operation

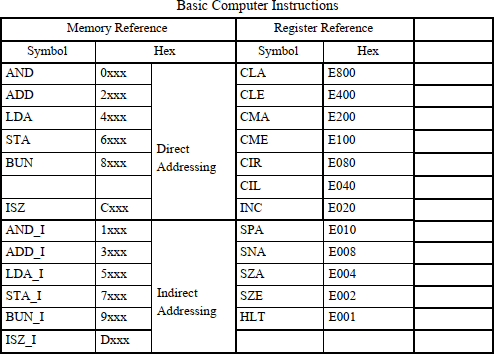
The **ISZ (Increment and Skip if Zero)** instruction increments the value stored at a specified memory location. If the result after incrementing is **zero**, the next instruction in the program is **skipped**.

**Program**

|  |  |
| --- | --- |
| ISZ 009 | ; Increment the value at memory location 009.  ; If the result becomes zero after incrementing, skip the next instruction. |
| OUT | ; Output the value in the accumulator (Skipped if ISZ resulted in zero). |
| HLT | ; Halt execution. |

The **ISZ (Increment and Skip if Zero)** instruction increments the value stored at a specific memory address. If the result becomes zero, it skips the next instruction. Below is the **step- by-step breakdown** of how the ISZ microinstruction executes in a **basic CPU architecture**.

Basic Machine Instructions

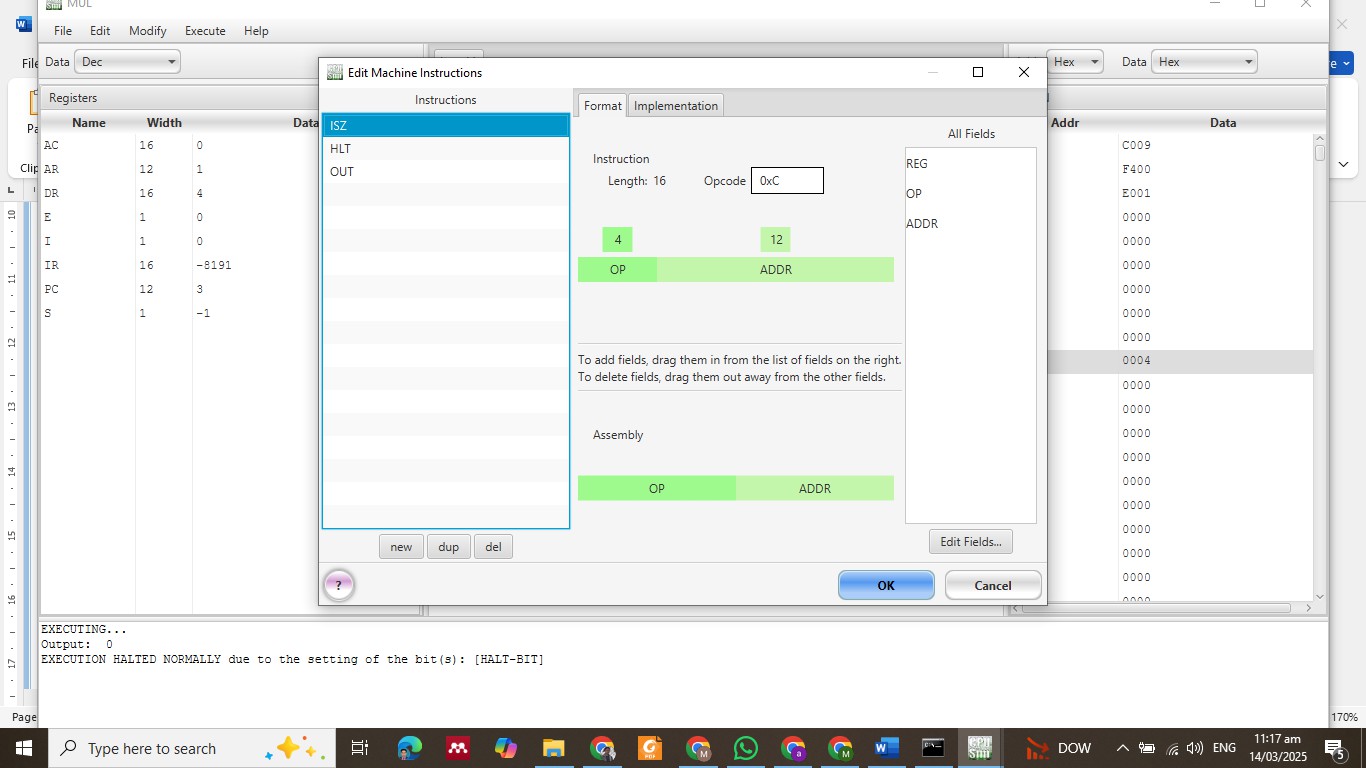


Step-by-Step Execution:

1. **ISZ 009 (Increment and Skip if Zero)**
   * **Reads the value stored at memory location 009.**
   * **Increments it by 1.**
   * **If the new value becomes 0, it skips the next instruction (OUT).**
   * **Otherwise, it proceeds to the next instruction.**
2. **OUT (Output the value in AC)**
   * **If ISZ does not skip, this instruction executes.**
   * **Outputs the value in the accumulator (AC) to the display/output device.**
3. **HLT (Halt Execution)**
   * **Stops program execution.**

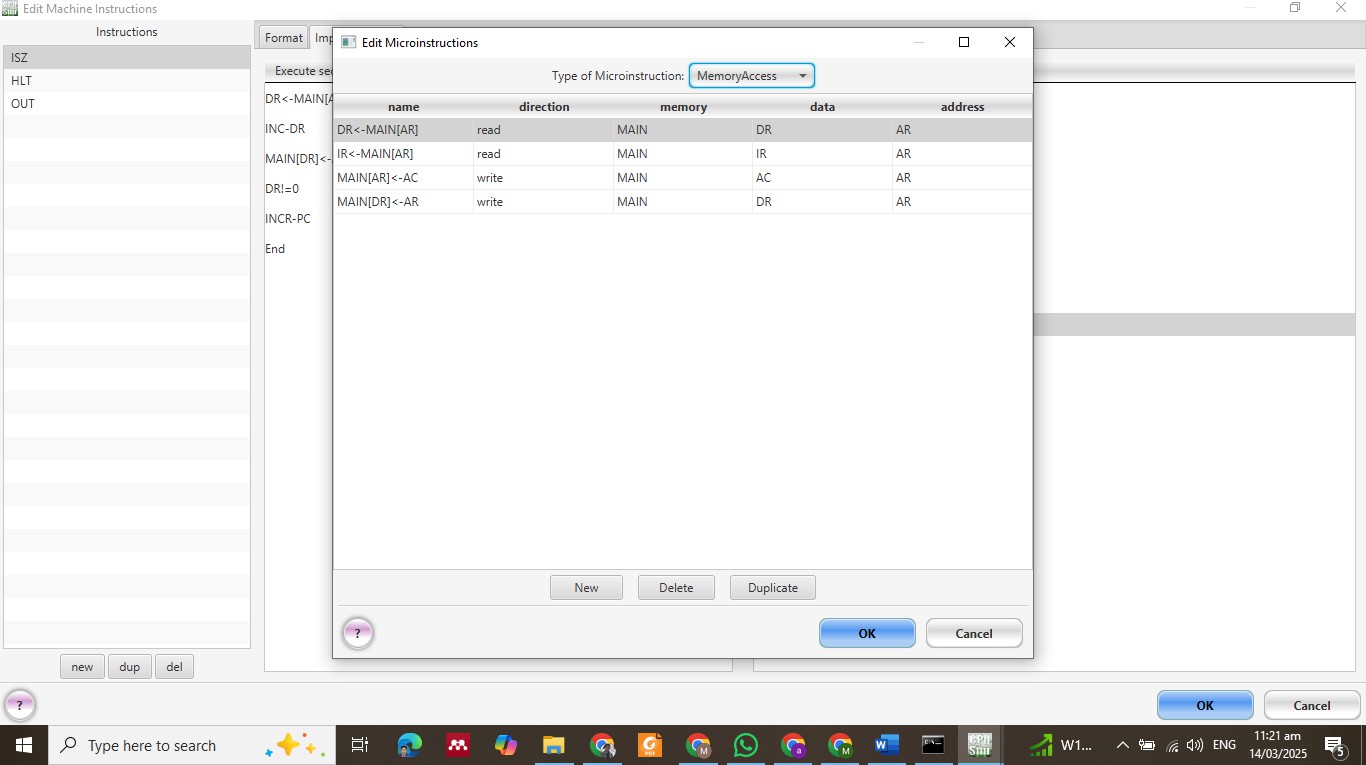
Step 1: Navigate to the Microinstructions Section

* **Open your simulator/tool and access the Microinstructions Section.**

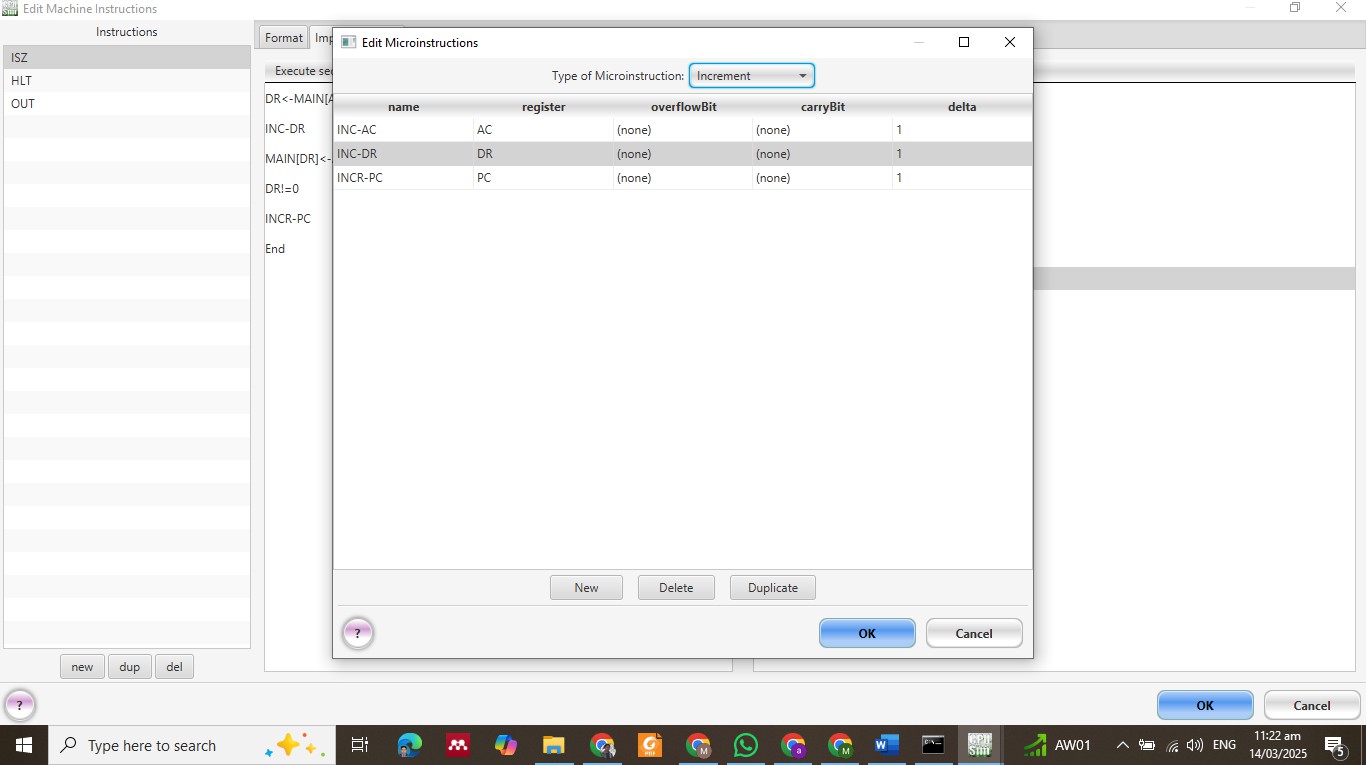


Step 2: Define ISZ Microinstruction

* **Fetch the value from memory (Address Register - AR) into Data Register (DR):**

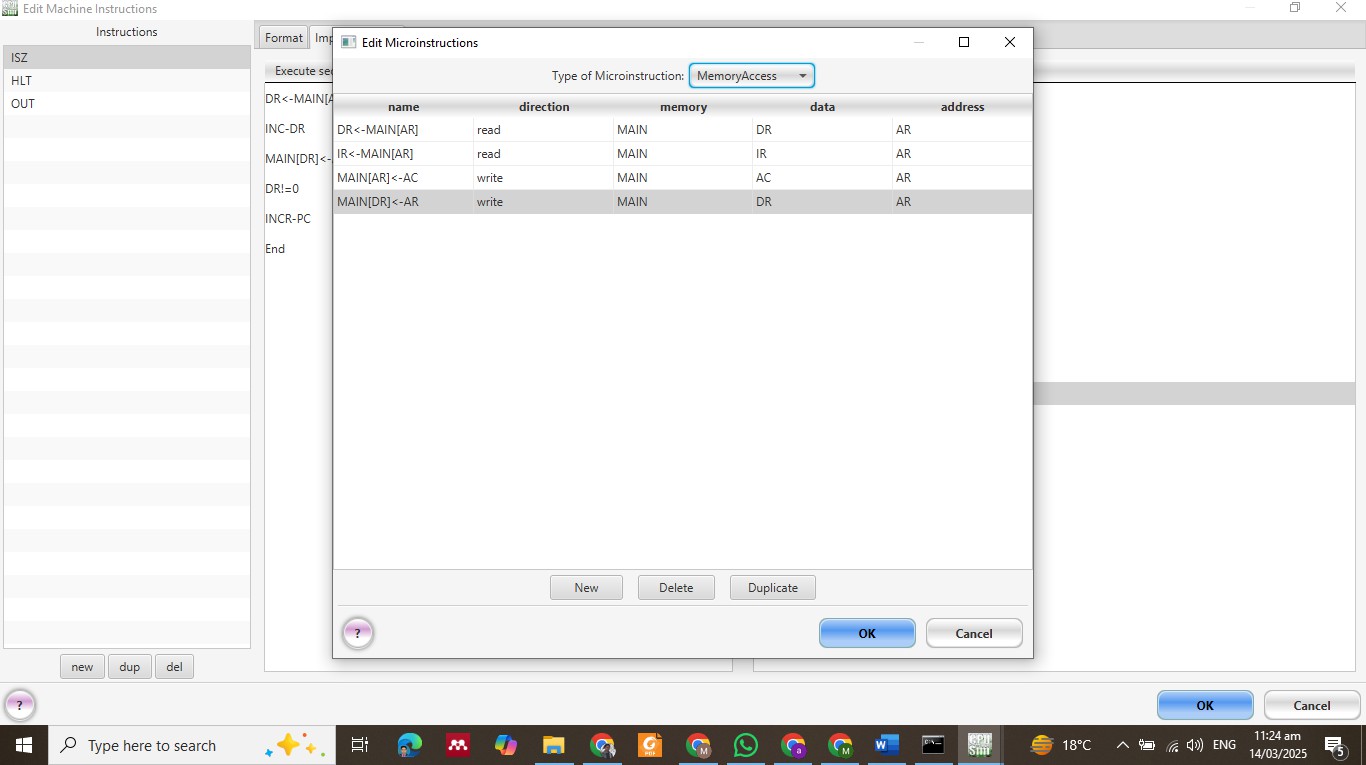
***DR ← MAIN[AR] ; Load the value from memory address AR into DR***

* **Increment the value in DR:**

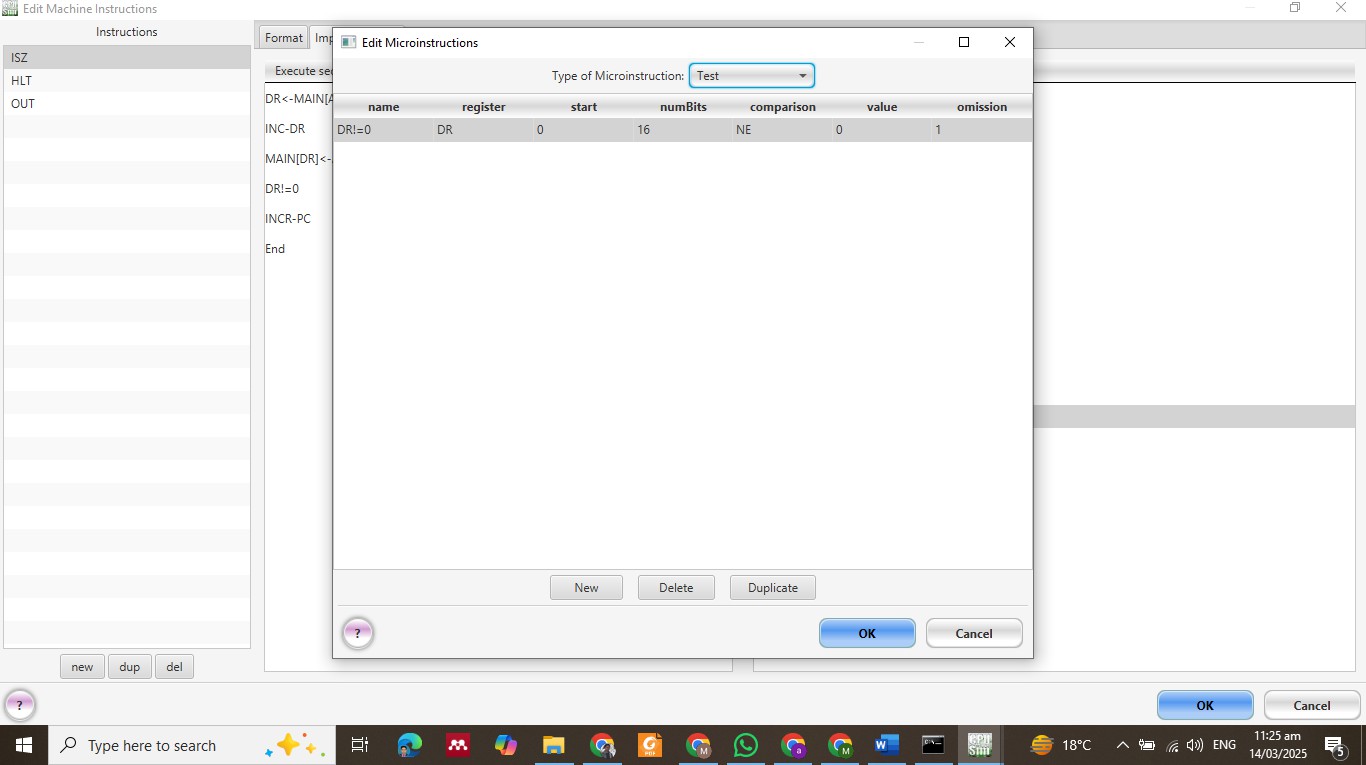
***INC-DR ; Increment DR by 1***

* **Store the updated value back into memory:**

***MAIN[AR] ← DR ; Store the new value back into the same memory address***

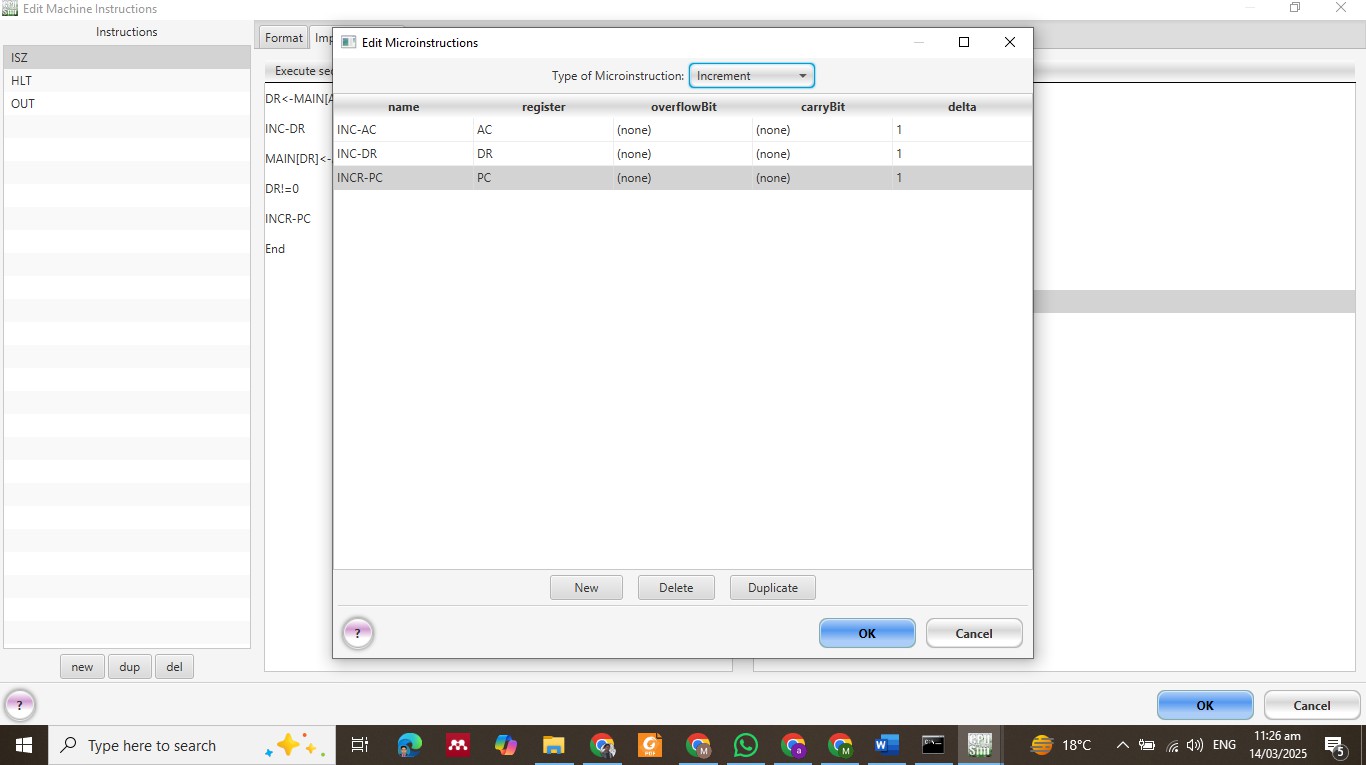
**

* **Check if the new value in DR is zero:**

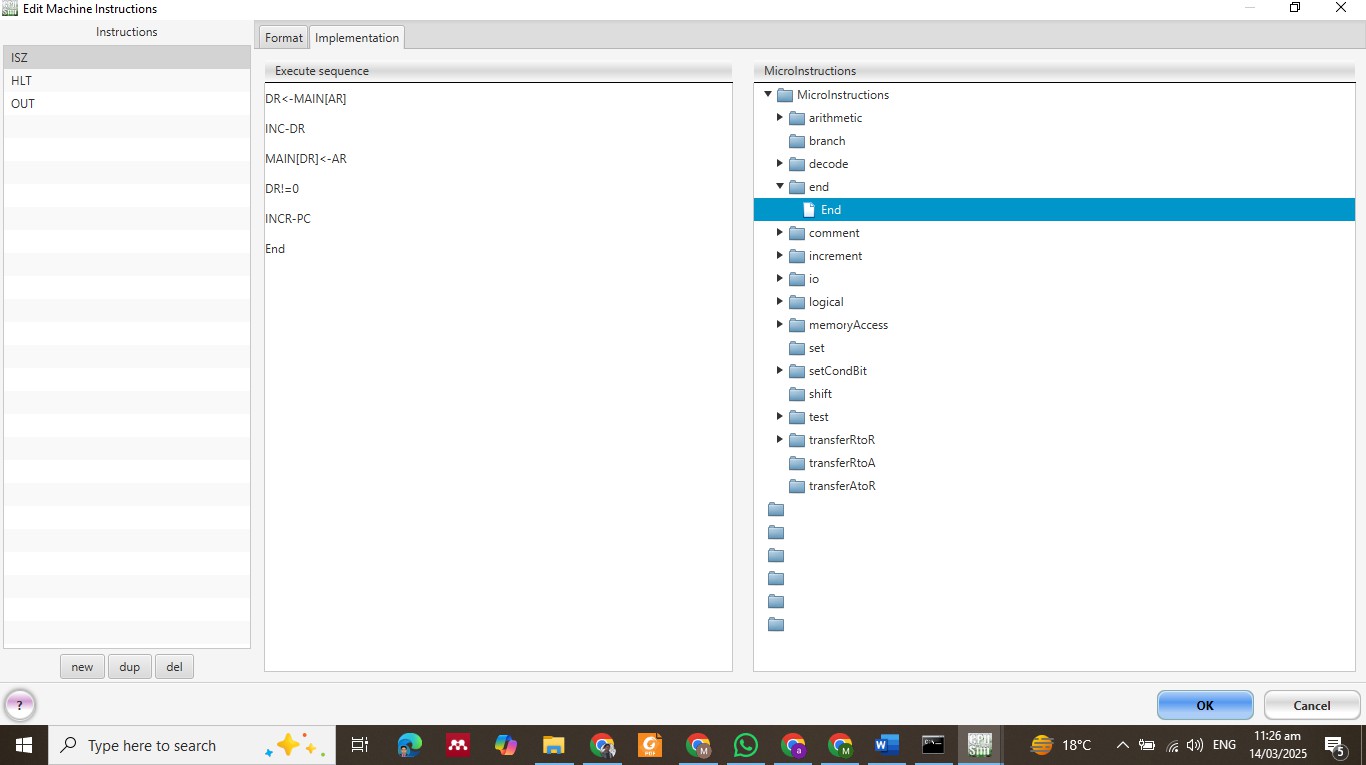
***DR != 0 ; If DR is not zero, proceed normally***

* **If the value is zero, increment the Program Counter (PC) to skip the next instruction:**

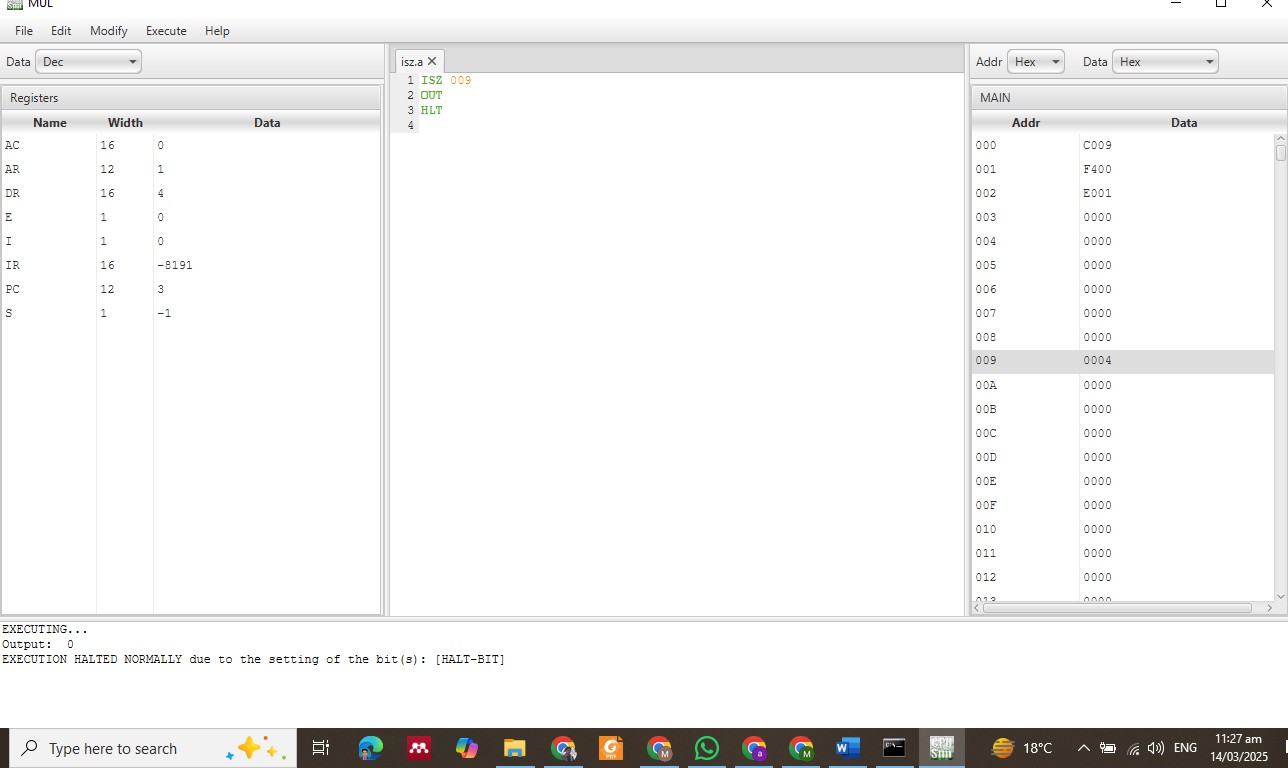
***INC-PC ; Skip next instruction if DR is zero***

**

* **End the microinstruction sequence:**

***END ; Finish execution***

Results :



Lab Tasks

Task Requirements:

1. **Set memory location 009 to -3. Increment by 3 and make it 0. Perform ISZ operation. Show the results.**
2. **Show that ISZ doesn’t skip when result ≠ 0.**