**Ayesha Zubair**

**52916**

**Lab 5**

**Tasks**

**Task 1**: Write a **detailed explanation** of how the Fetch-Decode-Execute cycle works.

The Fetch Decode Execute cycle is a process in which a computer extracts, interprets the instructions from memory, and then runs. This cycle is repeated continuously as the computer turns on. This is the basic mechanism that the processor can work.

**Step 1: Fetching the instruction**

The CPU fetches instruction from memory.

The program counter (PC) contains the address of the instruction that is to be executed next.

When it’s time to start processing the instruction, the CPU copies the instruction’s memory address and stores the copied data to another register on the CPU called the Instruction Register (IR).

 Then the PC increases and displays the following command in order.

**Step 2: Decoding**

The CPU decodes the instruction stored in the IR to determine what action needs to be performed.

Control unit interprets the instructions and prepares the required components of the CPU (for example, ALU or register) for execution.

**Step 3: Performance**

The CPU performs decoded instructions. If arithmetic or logical work is included, the arithmetic logic unit (ALU) performs calculations.

When data movement is included, registers such as batteries (ACs) and Data Register (DR) play a role in storing and transmitting data. The work results are stored in the register or recorded in memory.

**Task 2:** Use a simple instruction as an example and describe each step.

**Example Instruction: ADD 300**

**Fetch:** The instruction ADD 300 is fetched from memory, and the PC is incremented.

**Decode:** The Control Unit interprets the instruction and prepares to fetch data from address 300.

**Execute:** The value at memory location 300 is added to the Accumulator (AC), and the result is stored in AC.

**Task 3:** Explain the role of **PC, AR, IR, AC and DR in your own words.**

**PC (Program Counter): Holds the address of the next instruction that is to be executed from memory.**

**AR (Address Register): Holds the address of the instruction that is to be fetched from memory.**

**IR (Instruction Register): Stores the instruction that is currently being executed.**

**AC (Accumulator Register): Stores the results**

**DR (Data Register):** Temporarily holds data being transferred between memory and the CPU.

**Task 4:** What is the function of the Arithmetic Logic Unit (ALU**)** in CPU operations?

How does ALU interact with registers and memory?

The Arithmetic Logic Unit (ALU) is responsible for performing all arithmetic (Addition, subtraction, multiplication, and division) and logical operations (AND, OR, XOR, NOT, and bitwise shifts) within the CPU.

Interaction:

The ALU receives input from the AC, DR, or other general-purpose registers.

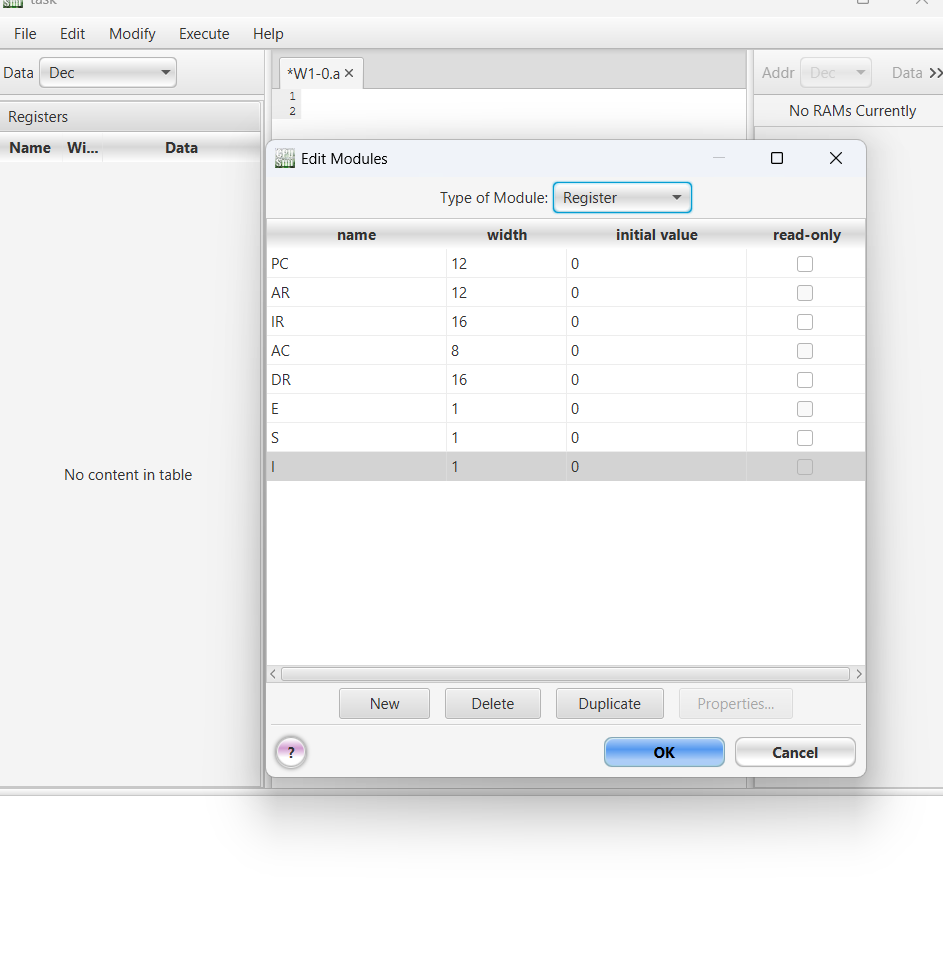
The Control Unit sends signals to the ALU to perform operations.

The ALU processes the data and sends the result back to the AC or another register.

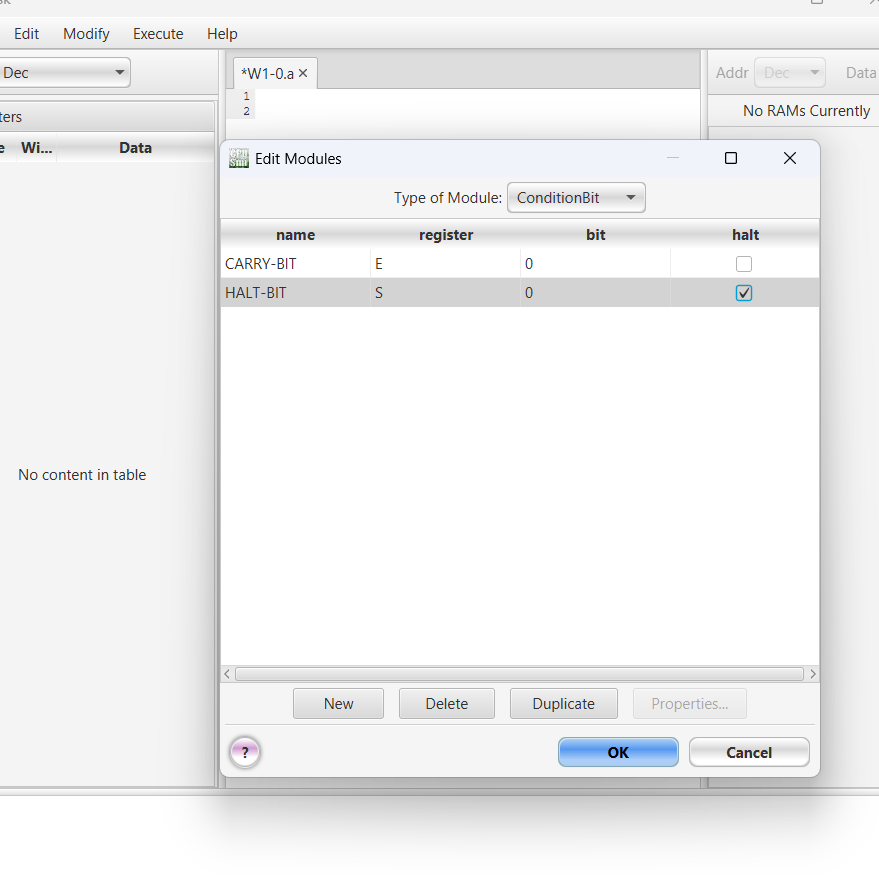
If necessary, the result is written to memory via the DR.

**Task 5:** Create a new base machine and set the width of AC to 8 bits.

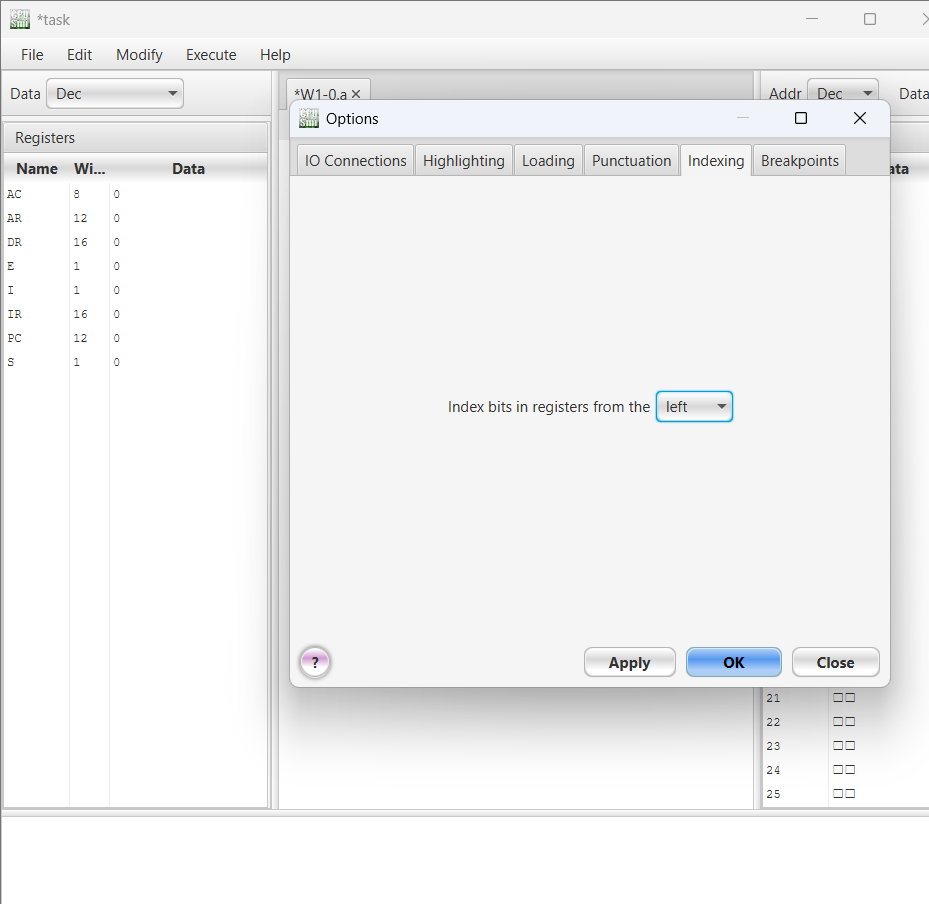
**Registers**

****

**Condition Registers**

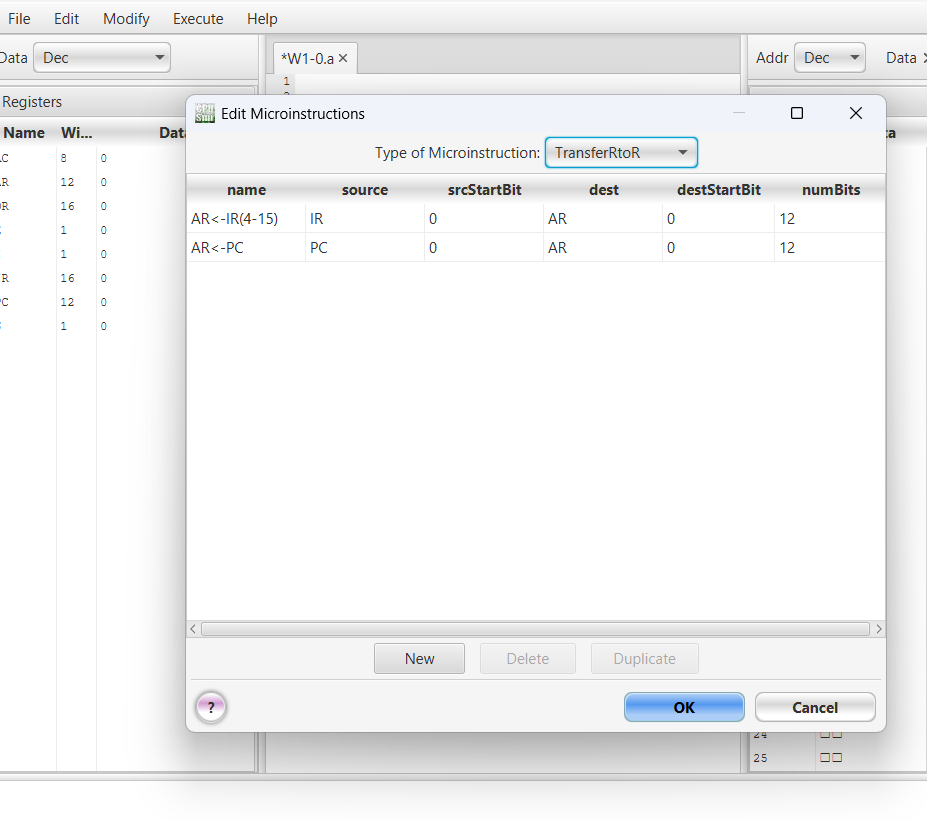
****

**Setting indexing to left**

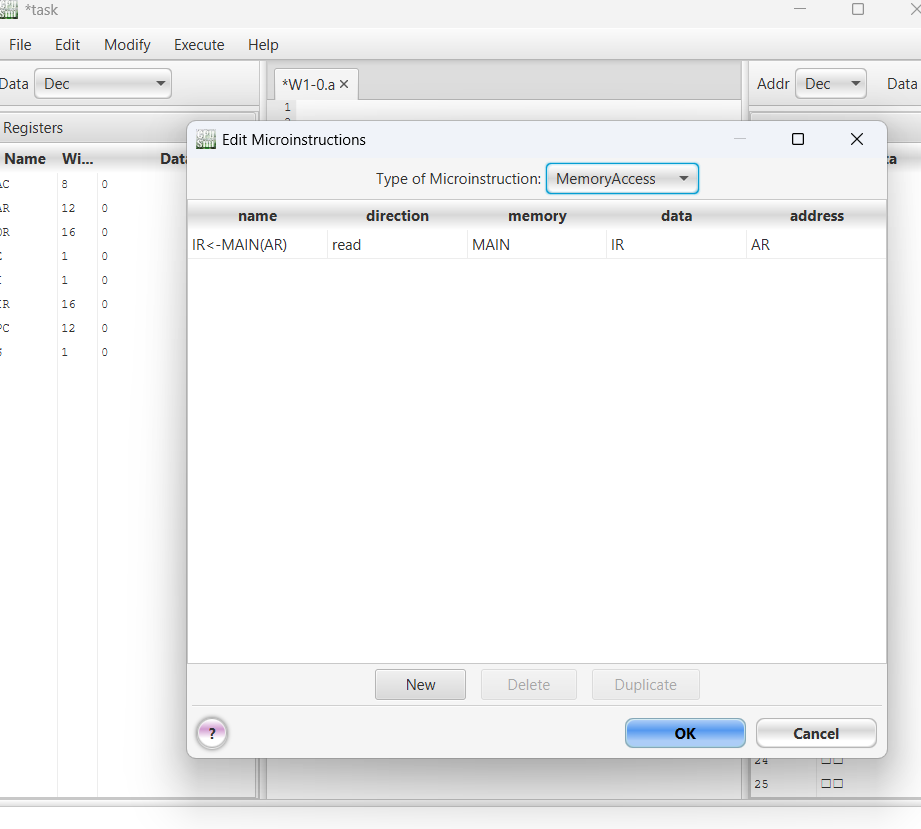
****

**Fetch Cycle**

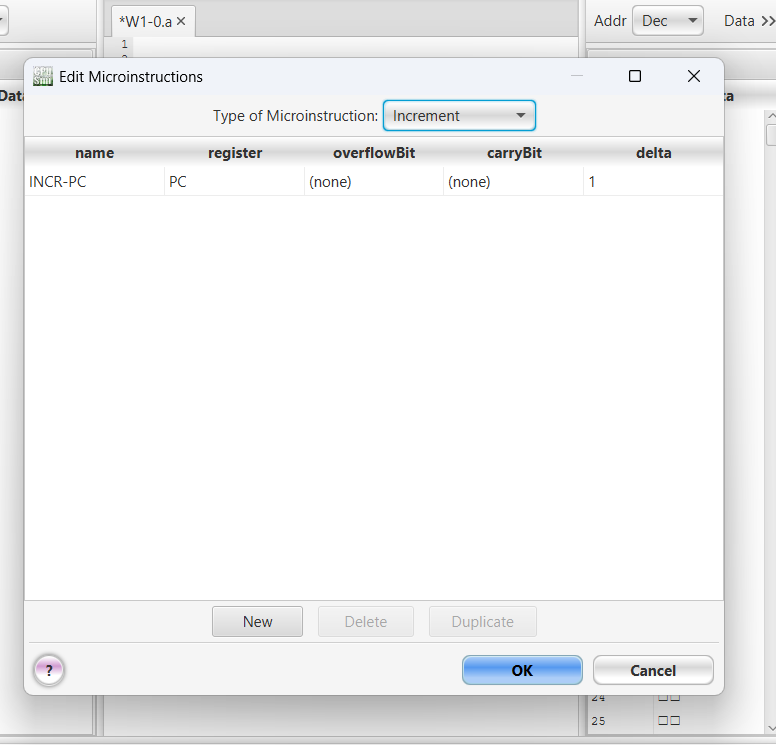
1. Transfer the address from **PC to AR**.

****

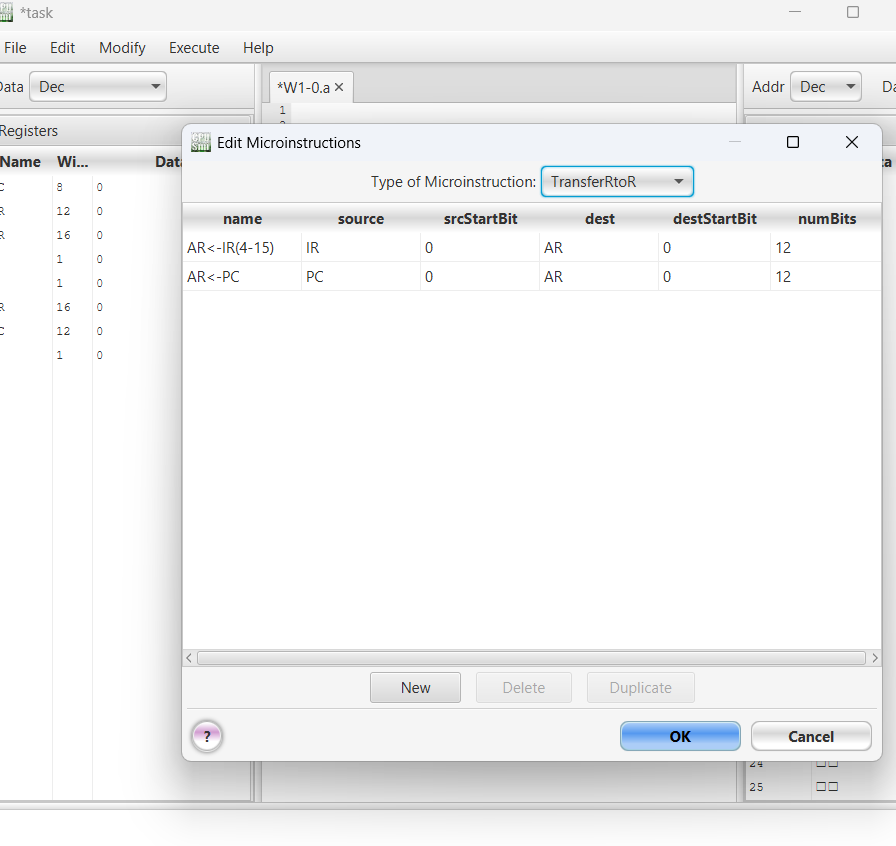
1. Read the instruction from memory into **IR**.

****

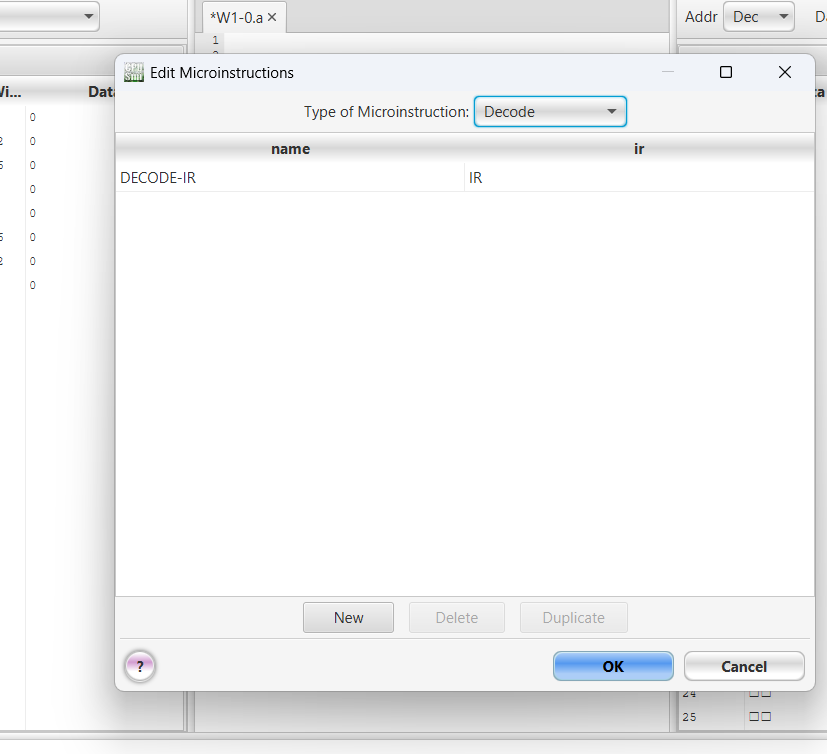
1. Increment **PC** to point to the next instruction.



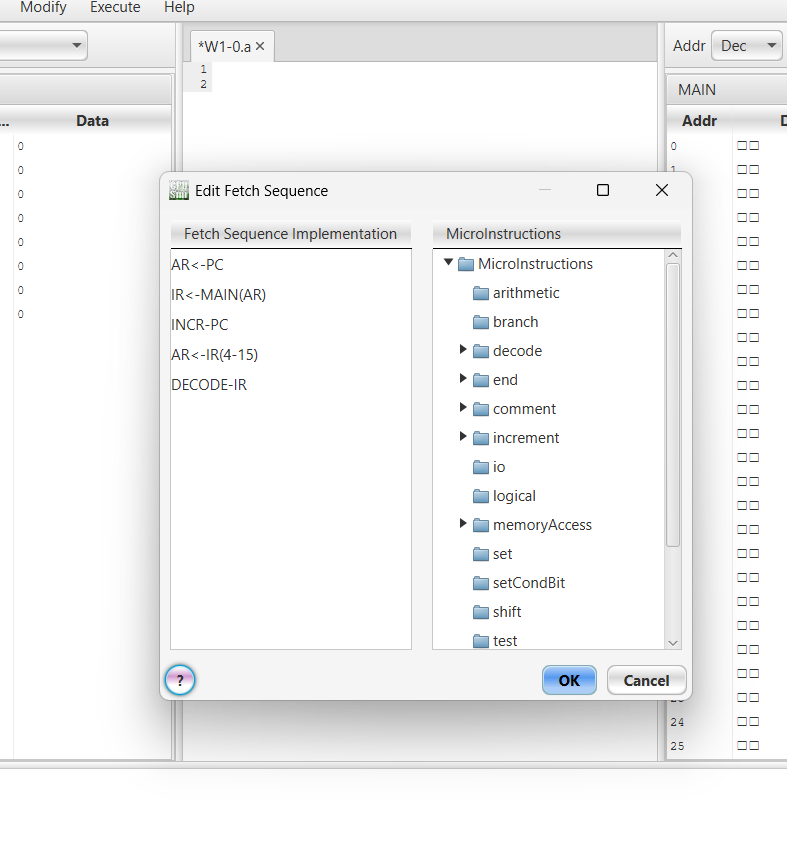
1. Extract the address part of the instruction and transfer it to **AR**.

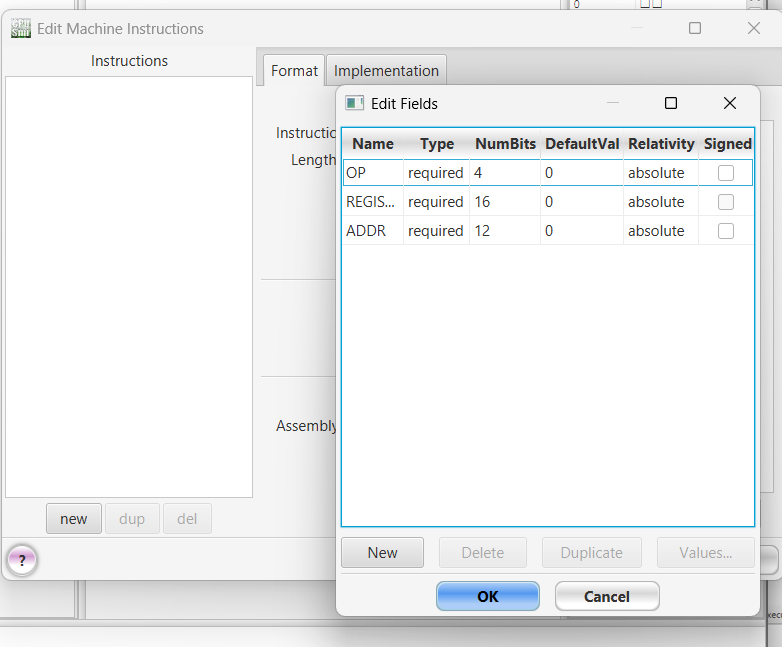


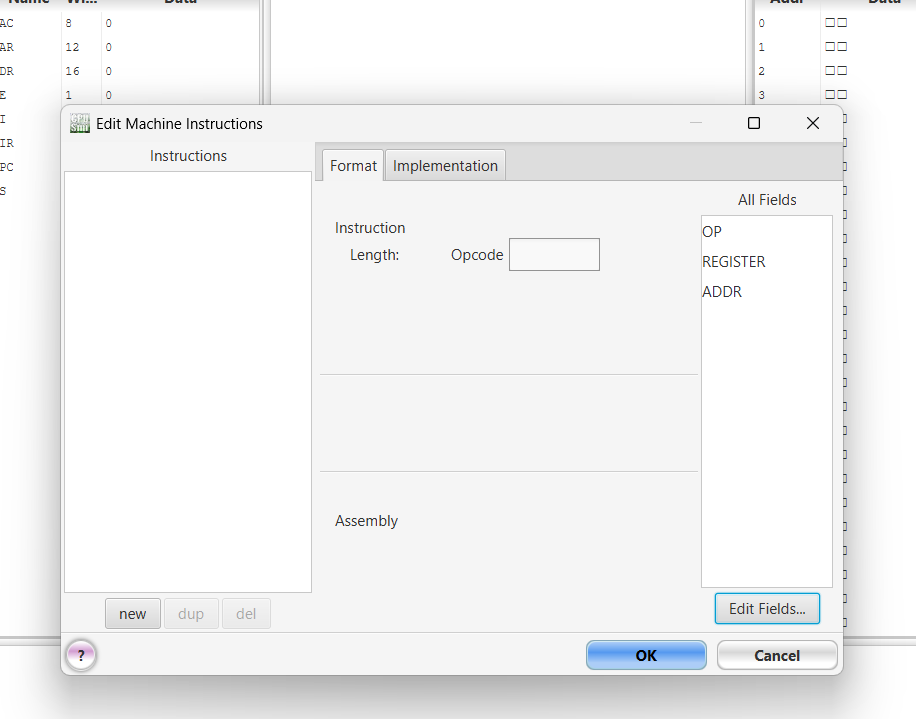
1. Decode the instruction for execution.



**Fetch Execution**

****

****

****