Project Name: Build a Virtual CPU Emulator

Week 03: Basic CPU Components

Objective: Architecting a complete CPU by implementing the major functional components of the design — ALU, CU, and registers; ALU will perform all the arithmetic and logical operations while CU will Sailor instruction execution process from feeding data to ALU to control signals. Registers are temporary data holder units that usually operate at very high speed. The following implementation will focus on how these interact to execute instructions, pass data around cleanly, and maintain effectively.

1. Build the ALU (Arithmetic Logic Unit)

The ALU performs arithmetic and logical operations. It would help if you implemented basic operations like addition, subtraction, AND, OR, etc.

Steps:

- Define a module/class for the ALU.
- Accept inputs (operands and operation code).
- Output the result based on the operation code.

2. Implement General-Purpose Registers

Registers temporarily store data for the CPU to process. General-purpose registers are used for intermediate operations.

Steps:

- Create a register file with a fixed number of registers.
- Provide methods to read from and write to specific registers.

3. Create the Program Counter and Instruction Register

- Program Counter (PC): Keeps track of the next instruction's address.
- Instruction Register (IR): Stores the current instruction being executed.

Steps for the Program Counter:

- Increment the address to point to the next instruction.
- Allow manual setting of the counter for jumps.

Steps for the Instruction Register:

- Load the instruction at the current PC address.