MIPS Instruction Set Simulator

Overview

Build a command-line simulator that mimics the behavior of a basic MIPS processor. It parses and executes a subset of MIPS assembly instructions and tracks changes to registers, memory, and the program counter.

Tools Required

Programming Language:

- Python or C++

Data Structures:

- Dictionary for 32 registers (`$t0`, `$s0`, etc.)
- List or array to simulate main memory

Development Steps

1. Define Register File and Memory

- Initialize 32 MIPS registers with zero values
- Set up a memory array (e.g., 1024 words) for data segment

2. Instruction Parser

- Read MIPS instructions from a text file or user input
- Support a basic instruction set:
    - `add`, `sub` (arithmetic)

    - `lw`, `sw` (load/store)
    - `beq` (branch if equal)
    - `j` (jump)

3. Execution Engine

- Parse each instruction line-by-line
- Simulate instruction behavior and update:
    - Register values
    - Memory contents
    - Program counter (PC)

4. Display Output

- Print trace of instructions as they are executed
- Show updates to register file and PC after each instruction

---

Expected Output

- CLI-based interactive or batch simulator
- Sample MIPS code input (in text form)
- Trace of instruction execution with updated registers and PC
- Educational tool to understand MIPS pipeline and ISA fundamentals