

VIRTUAL CPU EMULATOR

Week 1: Project Planning & Setup

- Defined core CPU functionalities.
- Chose Python for implementation.
- Set up GitHub for version control.

Week 2: Instruction Set Architecture (ISA)

- Designed basic instruction set: ADD, SUB, LOAD, STORE, JMP, and I/O.
- Documented instruction formats.
- Created a simple assembler-like instruction loader.

Week 3: Basic CPU Components

- Implemented ALU functions (addition, subtraction).
- Created general-purpose registers.
- Established program counter (PC) and instruction register (IR).

Week 4: Instruction Execution

- Developed fetch-decode-execute cycle.
- Integrated instruction execution within the loop.
- Successfully tested with simple programs.

Week 5: Memory Management

- Allocated simulated memory space.
- Implemented memory read/write operations.

Week 6: I/O Operations

- Designed an I/O instruction.
- Implemented output display functionality.

Week 7: Advanced Features

- Added jump instructions.
- Began structuring for branching and control flow.

Week 8: Performance Optimization

- Optimized fetch-execute loop.
- Enhanced memory efficiency.

This plan ensures a systematic approach to completing the virtual CPU project.