

Project Name: Build a Virtual CPU Emulator

Group Formation:

Group Size: Maximum 3 members per group.

Group Formation: Groups will be formed by 30/10/2024.

Group Submission: Each group must submit a list of group members on the provided Google [Sheet](#) by 30/10/2024.

The project will be divided into 10 weeks. Each week, specific tasks will be assigned. Groups will be evaluated based on the accuracy, completeness, and depth of the work submitted.

Weekly Tasks:

Week 1: Project Planning & Setup

Objective: Define project scope, gather resources, set up development environment. **Tasks:**

- Outline the features of the virtual CPU.
- Choose a programming language (Python/C++) and tools.
- Set up version control (e.g., GitHub).

Week 2: Instruction Set Architecture (ISA)

Objective: Design the ISA for the virtual CPU.

Tasks:

- Define basic instructions (ADD, SUB, LOAD, STORE, etc.).
- Document the instruction formats.

- Create a simple assembler to convert assembly code into machine code.

Week 3: Basic CPU Components

Objective: Implement core components of the CPU.

Tasks:

- Build the ALU (Arithmetic Logic Unit).
- Implement general-purpose registers.
- Create the program counter and instruction register.

Week 4: Instruction Execution

Objective: Develop the instruction fetch-decode-execute cycle. **Tasks:**

- Implement the instruction fetching mechanism.
- Decode instructions and execute them using the ALU and registers.
- Test with simple programs.

Week 5: Memory Management

Objective: Implement memory management for the virtual CPU. **Tasks:**

- Set up a simulated memory space.
- Implement memory read/write operations.
- Handle address mapping and memory segmentation.

Week 6: I/O Operations

Objective: Enable basic input/output operations.

Tasks:

- Implement simulated I/O devices (keyboard, display). ●

Create I/O instructions and integrate them with the CPU. ●

Test with I/O-intensive programs.

Week 7: Advanced Features

Objective: Add advanced CPU features.

Tasks:

- Implement branching and control flow instructions.
- Add support for subroutines and interrupts.
- Integrate a simple pipeline mechanism.

Week 8: Performance Optimization

Objective: Optimize the emulator for better performance. **Tasks:**

- Profile the emulator to identify bottlenecks.
- Optimize critical code paths.
- Enhance the assembler for better instruction encoding.

Week 9: Final Testing & Debugging

Objective: Thoroughly test and debug the emulator. **Tasks:**

- Test with a variety of assembly programs.
- Debug and fix any issues.
- Validate performance against benchmarks.

Week 10: Documentation & Presentation

Objective: Document the project and prepare for presentation. **Tasks:**

- Write comprehensive documentation.
- Prepare a project report and presentation slides.
- Conduct a demo session.