INDIAN SOCIETY FOR TECHNICAL EDUCATION, NITK CHAPTER

Implementation RISC-V pipelined processor

Project Mentors: Nikhitha Mathew
Project Mentees: Madoori Sowmith, Hafiz Raffi, Venkatesh S, Rishi M, Vishal K

Project Objectives

Design a 5-stage pipelined processor for RISC-V instructions.
Develop a Python-based assembler to convert assembly code into 32-bit hexadecimal machine code.

Introduction

The RISC-V architecture is an open standard instruction set architecture (ISA) that allows freedom in processor design. In this project, we implemented a five-stage pipelined RISC-V processor and built a custom Python-based assembler to translate assembly code into machine code.

Theory

The pipeline includes five stages: Instruction Fetch (IF), Instruction Decode (ID), Execute (EX), Memory Access (MEM), and Write Back (WB). Pipelining improves CPU throughput by overlapping execution stages. Hazard detection and forwarding mechanisms are implemented to handle data dependencies.

Methodology/Procedure

- Wrote Verilog modules for each pipeline stage.
- Implemented ALU, Control Unit, Register File, and Data Memory.
- Created a Python-based assembler that supports R, I, S, B, and J-type instructions.

Project Outcomes

- A functioning pipelined RISC-V processor design.
- Successfully developed assembler generating .hex files.
- Verified correctness with waveform outputs in GTKWave.

Results

Processor handled instructions with proper pipelining and forwarding.
Assembly code converted correctly to hex using assembler.
Output signals matched expected behavior in simulation.

Project Relevance

- Highlights the potential of open-source hardware in modern computing.
- Demonstrates practical knowledge of processor design and verification.
- Encourages deeper understanding of computer architecture for students.

Conclusion

The project successfully demonstrated the design and simulation of a RISC-V pipelined processor along with a working assembler, enhancing understanding of computer architecture and hardware-software integration.