CS2323: Extending the RISC-V Simulator to Support Pipelining and Hazard Handling

Devansh Tripathi

Contents

1 Objective

To enhance the existing single-cycle RISC-V simulator by implementing an instruction pipeline (IF, ID, EX, MEM, WB) adn supporting data, control, and structural hazard handling mechanisms.

The goal is to demonstrate the impact of pipelining on performance and compare various pipeline configurations.

2 Motivation

The current in-house simulator executes one instruction per cycle (single-cycle). However, this does not reflect the behaviour of real processors nowadays.

Pipelining is an architectural optimization that improves the cycle count by overlapping multiple instructions.

This project aims to modify the current simulator by introducing configurable pipelining functionality to it, which can demonstrate:

- Pipeline performance improvements.
- Differetn types of hazards and their effects.
- The impact of hazard mitigation techniques like forwarding and branch prediction.

3 Proposed Enhancements

The enhanced simulator will support the following configurable execution modes:

Mode	**Description**
0	Single-cycle
1	Pipelined with no hazard handling
2	Pipelined with hazard detection (stalling only)
3	Pipelined with hazard detection and data forwarding
4	Pipelined with hazard + static branch prediction
5	Pipelined with hazard + dynamic 1-bit branch prediction

These modes can be toggled with a configuration file or a command line flag.

4 Functional Scope

1. **Pipeline Stage Simulation**
   - Implement the five stages (IF, ID, EX, MEM, WB)
   - Introduce pipeline registers between stages (IF/ID, ID/EX, EX/MEM, MEM/WB)
2. **Hazard Handling**

5 Testing and Validation

6 Deliverables