CS2323: Initial Implementation Report

Devansh Tripathi - CS24BTECH11022

Contents

0.1 Introduction

This project adds 5-stage pipelining functionality to the in-house RISC-V processor simulator.

The simulator can be configured to run in different modes, such as enabling single-stage or multi-stage, with or without hazard detection, and so on.

This can be used to demonstate how pipelining works in real world simulators.

0.2 Implementation Status

As of the time of writing this report, the following has been implemented:

- The four pipeline registers - IF/ID, ID/EX, EX/MEM, MEM/WB have been implemented as `structs` in C++, separate from the register file.
- The five pipeline stages have been added, refactoring the code from the single cycle simulator so as to take inputs from the pipeline registers and write to the pipeline registers.
- The previously used intermediate variables have been retained for convenience, although they may be removed in the future to keep the code clean and maintainable.
- The current pipelining design does not support hazard detection.
- Support for floating point registers has been added.
- The benchmark testing scripts are currently being written, so as of now there are no performance comparions. However, the following can be said with certainty:
  - Number of cycles is more than the single cycle simulator.
  - Since it is a simulation and the stages are still executed one-after-another and not simultaneously, time taken for one cycle may be roughly equal to that of a single cycle simulator because the stages are executed sequentially in a similar fashion, the only difference being that here it is executed in backward order to prevent pipeline registers from being overwritten.
  - Due to these two factors, the time taken by this simulator would be more than that taken by a single cycle simulator.