Project Report of RISC-V CPU

Course Project of MS108 Computer System(1)

Bohan Hou (侯博涵)
ACM honor class @ Shanghai Jiao Tong University

1 Introduction

This project is a RISC-V CPU with Dynamic Sceduling and Superscalar implemented in Verilog HDL, which is runnable on FPGA(tested on xc7a35t).

2 Design

2.1 Features

Main features of this RISC-V CPU are briefly introduced in the table below.

Feature	RISC-V CPU
Clock Rate	100MHz
Pipelining	3 stages(Fetch, Decode, Execution)
Dynamic Sceduling	Tomasulo Algorithm
Superscalar	Multiple Issues(2 issues per clock at most) and FUs
Cache	512B 2-way associative I-cache
Branch Policy	stall

2.2 Summerize

- At first, I implemented a 4-stage pipelined CPU with Tomasulo Algorithm and reorder buffer but without speculation and passed all tests on FPGA. Then I intended to implement dynamic branch prediction as speculation, but I found it difficult to handle misprediction, which calls for complicate logic and design because of dynamic scheduling.

- Superscalar is another way to improve throughput, so I removed ROB in my design and made a 3-stage pipelined design with multiple issues. In testcase pi.c, it shortens running time from 2.44s to 2.15s. Finally I changed the directed indexed I-Cache to 2-way associative cache to improve hit rate, which shortens pi.c to 2.0s under 100MHz.
- In all, branch stall is still a main problem to handle, and such a relatively complicate design makes it more difficult to simply increase clock rate to decrease cpu time compared with standard 5-staged pipeline, and I don't think it meaningful to do so as a course project work.

2.3 Design Graph

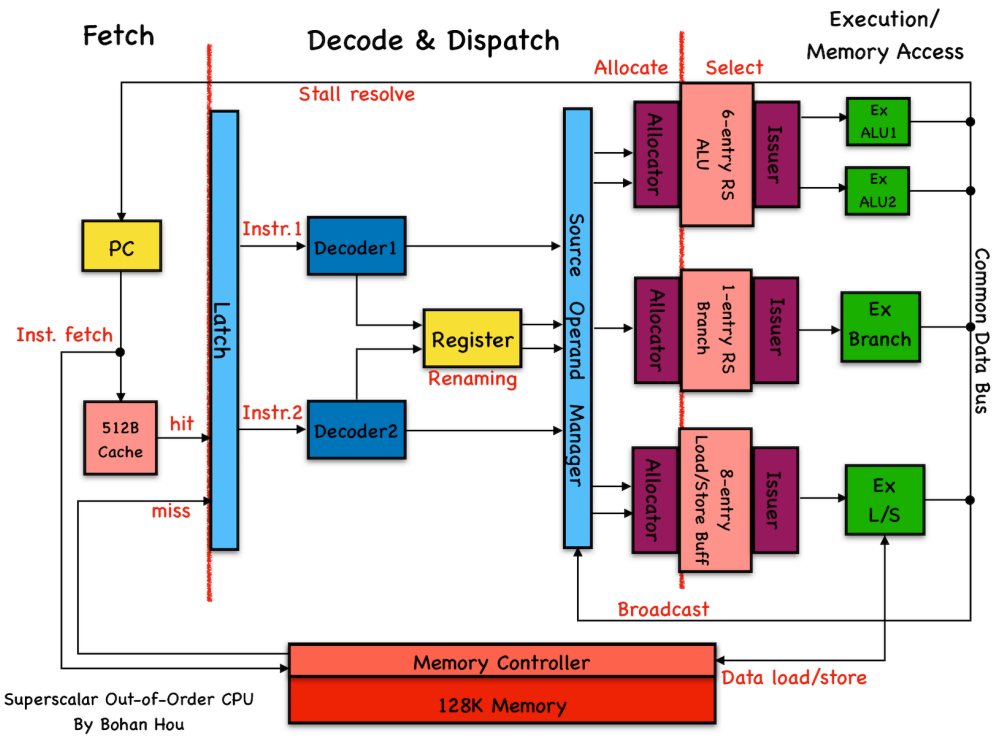

Fig. 1: Overview of CPU Design

References

[1] John L. Hennessy, David A. Patterson, et al. *Computer Architecture: A Quantitative Approach*, Sixth Edition, 2019.

[2] Arch Lab. in Tokyo Institute of Technology *RIDECORE (RIsc-v Dynamic Execution CORE)*