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

GitHub repository of my CPU project: https://github.com/spectrometerHBH/Chaos

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
ISA	RISC-V RV32I subset
Clock Rate	100MHz
UART	Baudrate 115200 bit/s
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
Memory	128K BRAM memory on chip
Branch Policy	stall

2.2 Design Graph

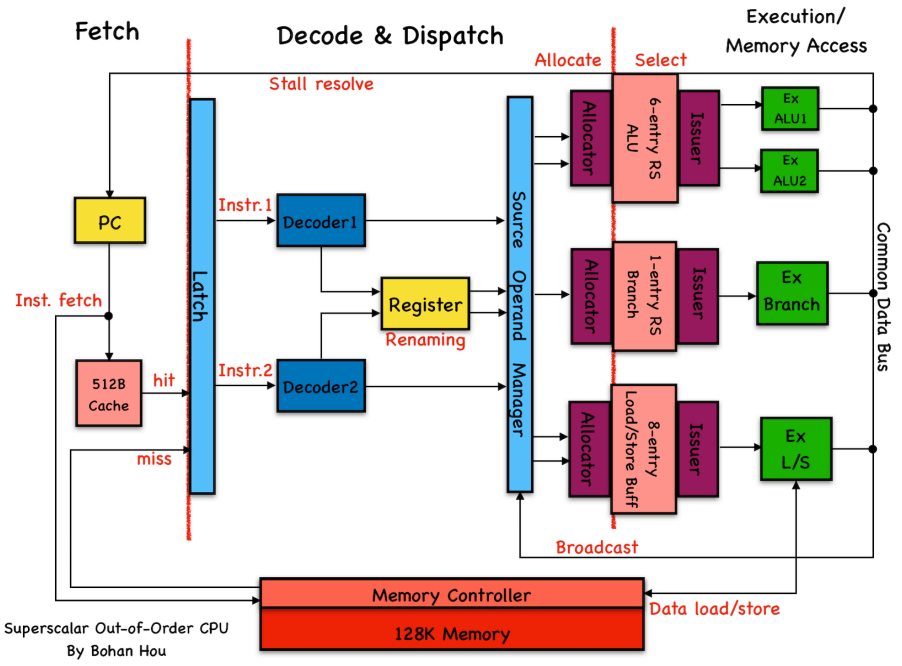

Fig. 1: Overview of CPU Design

2.3 Specification

- Fetch - Fetcher module sends signal to cache and memctrl at the same time. $T_{hit} = 1$, $T_{penalty} \leq 15$ If PC and PC+4 both hit in cache, Fetcher will send 2 instructions to Decode stage. Fetcher will stall PC when branching or jumping.
- Decode - Decoder and Dispatcher will decode the 2 instructions and allocate 2 locks to each of them. Dispatcher will address the dependence between the two instructions.
- Execution - ALU reservation station will send at most 2 instructions to Execution stage.

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)*