RISC-V Report

1 Abstract

This is a report of the project of MS108, computer architecture homework implementing a simplified CPU supporting some of the RISC-V A extension instruction sets, and my solution is based on Tomasulo algorithm.

2 Feature

2.1 Tomasulo architecture

The Tomasulo architecture aims to make the execution be Out-of-Order(abbreviated by OOO), and solves the problem triggered by OOO, such as WAR, WAW, and precise execution. The key is to use renaming, reservation station(abbreviated by RS) and ROB to deal with those problem.

In my design, each FU has its own RS, and the size of each RS is different, since some instructions take longer time to execute. Renaming is based on the line where the instruction is in ROB.

2.2 Cache

I implemented a 512B I-cache in my project, and later I tried to make it a 2-way associative one, but there is almost no improvement. I guess this is because the testbench only contains short codes, and a 512B I-cache can almost cover the total program.

My 2-way associative cache is stored in /backup/Units.

2.3 Branch Policy

I first design a one without branch prediction and its policy is stall. Since branch instructions holds about sixth of all instructions, it stalls frequently and is not so OOO. This one can run with respect to a frequency of 100MHz.

Then I finish a one with branch prediction. Like what BOOM(Berkeley Out-of-Order Machine) does, it uses shadow registers and branch bits. Each instruction is given a branch mask to evidence if it follows an in-flight branch instruction. If a branch instruction is executed, it boardcasts to update other instructions' branch mask. A Load/Store instruction can execute only when its branch mask is cleared.

This one can run with respect to a frequency of 70MHz.

This sounds cool, but when I really implemented it, I found that the FPGA is almost full(99% LUT used), since to deal with the boardcast of branch instruction, a large amount of logic units are in need. (let alone that the shadow registers are huge)

But it runs faster and is really cool.

3 Summary

The main problem I face is that I do not understand how to make improvement in organization level. Once I notice that I repeat calculating one same thing in each RS, so I pick it out: calculate it outside and send the result to each RS.

The result is crazy: the number of LUT increases considerably.

The Tomasulo is efficient especially there is a set of Load/Store instructions. In such tests, even the naive version (the one whose branch policy is stall) makes sense.

4 Reference

1. https://github.com/riscv-boom/riscv-boom, an out of order RISC-V CPU using chisel, with branch bits(named branch mask there)

2. http://www.kroening.com/diplom/diplom/main003.html, details about the hardware of Tomasulo Architecture.