

多发射乱序 CPU 的 RTL 级实现与验证

Yingkun Zhou, 2015K8009929023

key word: RISC-V, MIPS, CPU, out-of-order, multi-issue

乱序多发射的 CPU 设计并不容易, 而且比设计更加困难的是调试, 如何保证复杂执行的 CPU 能够有条不紊, 毫无错误的运行比如像 Linux 操作系统这样的大型软件更加难上加难。参考 RISC-V 开源部分的资料, 采用的方案如下:

1. 对于单一时钟, 同步复位的电路, 采用 chisel 来编写, 考虑到其对于逻辑的刻画效率比 verilog 更高, 而且对于可综合电路的描述能够做到和 verilog 同样的精确
2. 其次乱序的 CPU 如今也有开源的 BOOM 代码实现, 可以很好的参考借鉴
3. RISC-V 目前有一整套的调试手段可以学习借用
4. RISC-V 有整套的软件栈可以运行在自行设计的 CPU 之上

综上, 计划的第一步是实现一个基于 RISC-V ISA 的乱序双发射的 CPU, 调试完毕能够跑通简单的测试程序。本人也深知这绝非易事 (希望能够比较顺利完成)。接着在时间允许的情况下做操作系统的移植, 继而能够支持多核。注意到一开始的设计必须考虑到微结构和 ISA 之间的解耦合, 这样将 RISC-V 改成 MIPS 工作量会降低很多。得到 MIPS 版本的 CPU 目的在于采用相同的微结构可以客观的对比两个 ISA 的优劣。同时作为工作的一部分, 性能的优化以及设计空间的搜索必不可少, 期望能够在跑相同 benchmark 的情况下持平甚至超过 ARM 的 Cortex A53 的水平。