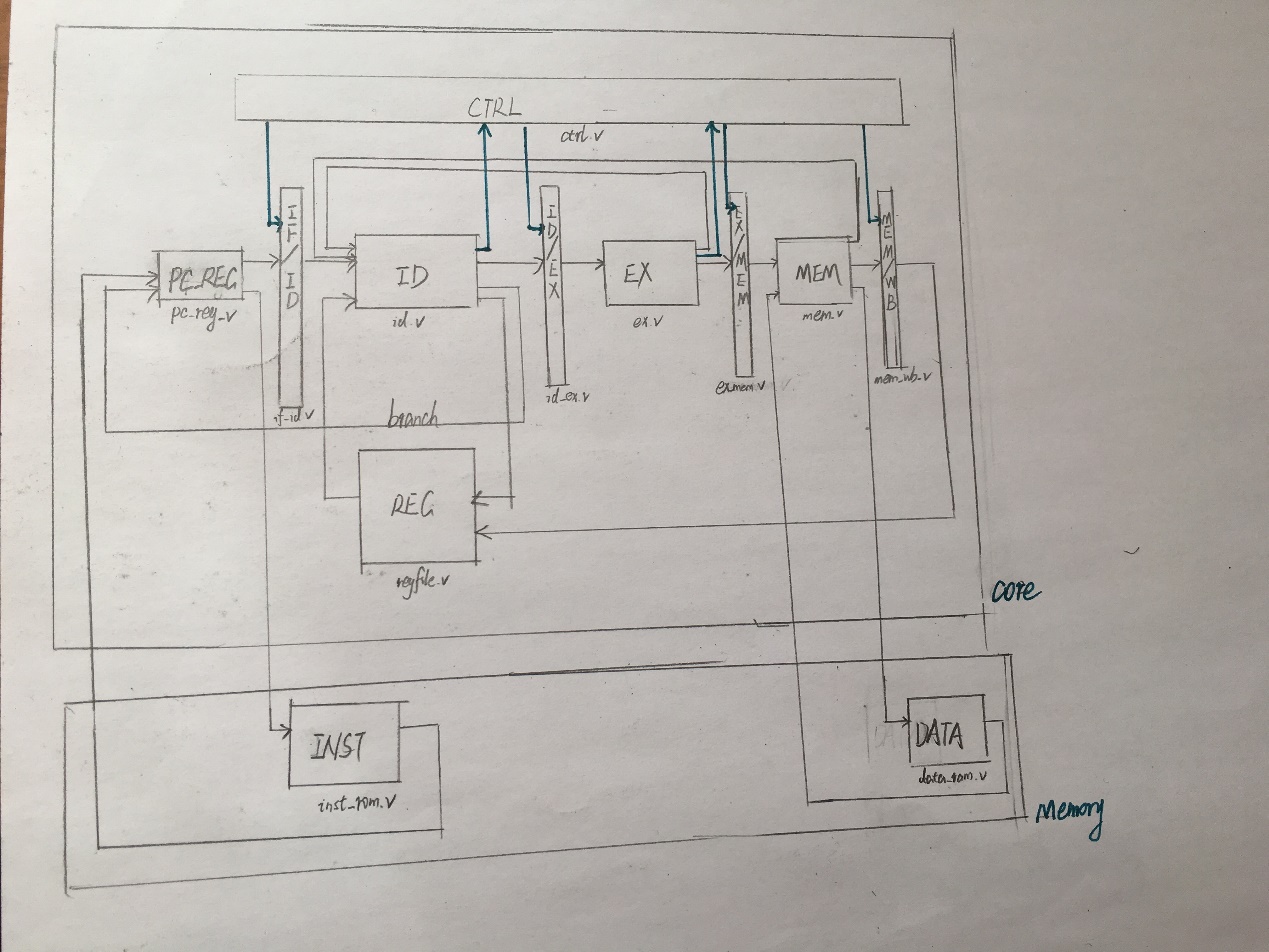
RISCV CPU 报告

ACM2016 朱立臻

1. 基本介绍

这个大作业就是简单地实现了一个RISCV指令集的CPU。整体架构如图



采用了标准的五级流水，并使用了forwarding。对于不能一拍完成的使用了stall。

1. 创新

并没有太多的创新，本来准备在cache上做一些优化，比如写多路组关联，victim cache, pseudo-associativity,并观察对于取数据的访问速度有何影响，但是由于cache还没有调出来，也没有写什么非常有价值的测试数据，只能作罢。

1. 遇到问题
2. 对于verilog和vivado不够熟悉，这是与c++完全不同的一种语言，理解它的运作就很重要。调代码的时候，首先经常因为一些语法错误查了很久的bug，其次，对于波形图，刚开始看的时候还是有一点吃力，不知道要先看那个模块的。
3. 阅读代码的能力不高。最开始是想把uart也写完的，但是助教的代码就看了很久，本身也有一点畏难心理吧，总觉得会出一些时序问题，还剩两天的时候就放弃把自己的代码和助教的板子连起来。最后选择写了cache还没有调出来，因为在体系结构课上重点讲了cache,有很多实现方法，也有很多优化，对于有些优化的性能，讲解的时候其实自己是有一点疑义的，很想自己实现一下，这次没能做出来还是很大的遗憾吧，还要继续加油。

最后特别感谢一些两位大佬：感谢冯思远大佬教我如何高效调代码，感谢孙雪晖大佬在我跑完模拟之后告诉我应该干什么以及怎么做（虽然这个之后就没真的实现什么过）。