

这是我的最终设计图, 比较简略, 没有画forwarding和SMC

就是经典的五级流水

Top

|

CPU

|

Controller---aluCtr

pcBuf---memory---ifidBuf--- | ---SE ---idexBuf---alu---exmmBuf---memory---mmwbBuf---

Registers

一些特点:

与大多数人的设计图略有不同, 很多改变都是为了减轻controller的复杂度, 并且避免出现冲突

比如 1.把译码工作在aluCtr里解决,

2.把forwarding的工作在controller之外判断解决

3.把reset, stall等工作拿在controller之外进行

因为这些事是和controller没有关系的, 主要是取决于流水线中数据的位置关系, 以我的理解, controller也只是一个发出指令的原件罢了

4.运用了很多 ? : 指令, 这样运算速度更快, 并且不用专门写一个选择模块, 这样稳定性也 更强, 只是代码的可读性会降低, 不过由于结构比较简单, 调试的时候也并没有遇到太大的困 难.

其余的并无太大差异

遇到的纠结的问题:

1. STALL指令无法工作导致无法bubble: 看了一下波形, STALL变量在上阶沿变成1的时候又瞬间变成0了, 看了一下波形, 发现STALL指令被改掉了, 于是我在他的赋值处加了一些判断就解决了问题.
2. Buuble之后beq无法工作. 同样看了一下波形, 发现branch被STALL指令影响, 从而beq失效了, 同样改了一下赋值语句, 就解决了问题.

虽然对硬件描述语言不熟悉, 也觉得很抽象, 但是熟悉了之后, 发现观察波形非常方便, 熟悉一下就驾轻就熟了, 非常爽感.

写完了之后再谈感受, 觉得最困难的阶段就是刚开始的时候, 因为并不熟悉硬件描述语言, 也没学过数电, 也没有对流水线的深刻感觉, 只知道数据是一拍一拍流过来的, 还要有控制信号, 但是都不知道怎么运作的.

刚写的时候很迷茫, 只能先写一些简单的部分, 全都写完了才觉得真正明白了五级流水是怎么运作的, 才找到了硬件描述的感觉,

最后总结一点心得, 就是一定要想清楚在写.

好像每次大作业得出的都是这个心得, 可是时间都不够想清楚的, 或者说只有写着写着才能想得更清楚, 尤其这次处理器基本上都要自己设计, 写的过程中还有许多设计与改动, 以后还是要早点动手或者动脑.

======================================================================================

Cache设计

其实会写了之后就没什么问题了, 想清楚之后开始写就行了. 第二阶段代码量也不大, 也就一两个模块.

L1Cache和L2Cache采用相同的结构, 直接映射, write through, Most Recently Used, inclusive

一些特点:

为了避免冗杂带来bug, 尽量使结构简单化…

两级Cache的结构相同, 这样更稳定, 也方便编程和调试

遇到的问题:

之前写的流水线中没有遇到什么时序的问题, 因为都是buffer一打开, 数据都一同流过去了, 这下加了两层cache, 就遇到了很多问题, 甚至要修改之前的结构

Debug的过程遇到了很多麻烦, 之前只要看看波形就可以了, 这回甚至还要动态调试, 看来模拟硬件还是不是那么简单的. 所幸由于模块结构比较简单, modelsim也比较好用, 没有遭遇太大麻烦. 大概三天之内就写好了.