

Lab2 实验报告

PB18051098 徐碧涵

实验目标

实现一个流水线 CPU，能够完成逻辑运算指令，条件分支指令，无条件分支指令，load 指令，store 指令以及 CSR 指令的执行并处理数据相关。

实验环境和工具

Vivado 2018.3

实验内容和过程

阶段一 + 阶段二

我跳过了阶段一的测试，在两个阶段合并完成之后一起进行测试。

该阶段的主要工作是对各类指令生成相应的控制信号以及处理数据相关。

控制信号的生成

需要考察各类指令在执行过程中所使用到的功能部件以及所需数据的流向，进而通过产生控制信号控制相关的功能部件以及控制数据的正确流向，从而正确执行相应的指令功能。

Hazard 模块

在 load-use 型数据相关的情况下需要 stall 流水线，分支指令需要 flush 流水线，jalr 以及 branch 指令（分支成功）需要 flush 掉两条指令，jal 指令需要 flush 掉一条指令。

其余数据相关通过将数据 forwarding 到 EX 段来实现。数据可以从 MEM 段以及 WB 段转发至 EX 段，在判断数据相关时，需要注意如果目标寄存器是 0 号寄存器，无需转发，以及 MEM 段和 WB 段都发生数据相关时，MEM 段转发的数据具有更高的优先级。

阶段三

CSR 指令功能的实现

为了实现 CSR 指令功能，在数据通路中新增了 CSR 寄存器文件，在 ID 段进行从 CSR 指令中获取立即数以及 CSR 地址，传送到 EX 段寄存器并在 EX 段进行 CSR 的读操作，在对读出的数据进行处理（rs/立即数中为 1 的位将 csr 值相应位置 0/1）或直接选择立即数/rs 的值（CSRRW，CSRRWI）得到的结果在 EX 段直接写回 CSR 寄存器，之后将之前读出的 csr 值写入 MEM 段寄存器最终在 WB 段写入 rd 寄存器。

实验总结

问题

在本次实验中遇到的问题主要是在对 **branch** 指令产生控制信号 **Alusrc2** 时由于疏忽错误的产生了选择 **imm** 的信号值，从而在测试时分支一直失败，**debug** 了很久才发现问题。

关于时间分配：

阶段一 + 阶段二：共 10h，2h 主要进行控制信号的分析，其余时间完成代码的填写与 **debug**

阶段三：共 4h，两小时用于构思需要新增的数据通路以及所需产生的控制信号，其余时间用于代码的编写以及测试样例的编写。

收获

通过 lab2 我对于流水线 CPU 的工作原理以及数据通路中每个部件的作用以及控制信号有了更加深入的认识。

改进意见

希望关于测试用例的编写方法能够提供更加细致的介绍，自己摸索需要花费很长的时间。