

实验 2 报告

学号 2016K8009907007

姓名 黄熠华

一、实验任务（10%）

设计五级流水线 CPU，每条指令经过取址、译码、执行、访存、写回五个阶段，添加乘除法器以及相应指令，考虑数据相关，最终仿真验证功能正确并上板通过。

二、实验设计（30%）

五级流水线是以 CPU 内部硬件分工来划分五级的，因此需要将 CPU 划分为五个模块。方便起见我在 my_cpu.v 文件里用注释的星号行划分出五个模块，而并不写成模块并实例化。五个模块的硬件分别以 PC, 译码器, ALU, 访存接口, RegFile 为主体，各自进行相关阶段的操作。

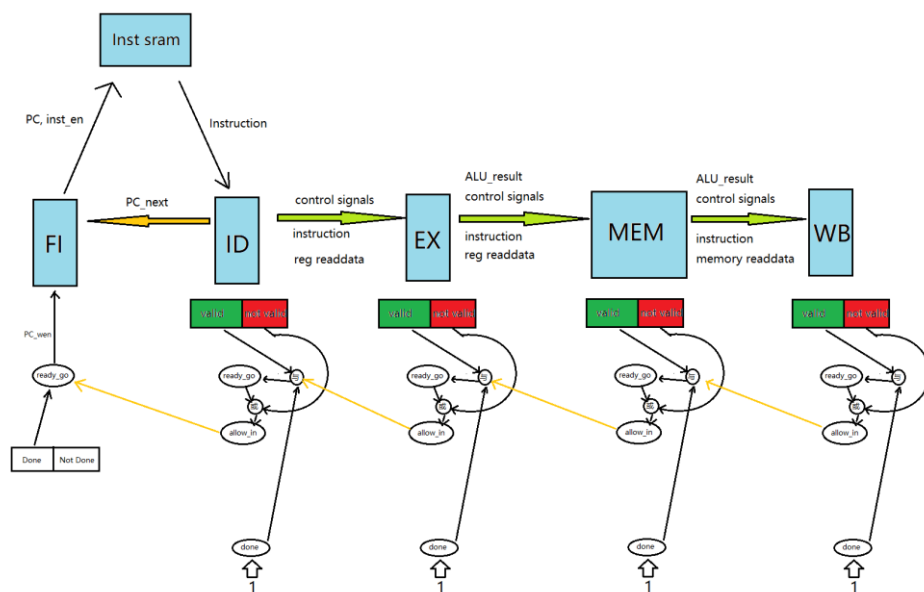
注意到流水线的五个阶段并不是相互独立的，而存在着许多数据依赖，因此有必要将一些数据在五级中某些级之间进行流动。译码阶段得到的控制信号，执行阶段得到的计算结果，访存阶段得到的数据等，都需要流到后面的阶段并加以使用。所以流水线 CPU 的“流水”体现在指令以及一些控制信号的流动。

代码中定义了五条 done 信号，用以表明当前阶段是否结束。不完全参考老师上课所讲授的，我给每个状态设定了 valid, ready_go 和 allow_in 信号，并以将它们链式关系连接。一个阶段的 valid 信号拉高表明其包含了有效内容，ready_go 信号拉高表明该阶段的指令在下一时钟上升沿会流动到下一个阶段，allow_in 信号的拉高意味着这个阶段在下一时钟上升沿可以接收来自上一级的指令。

在考虑了访存、乘法器未出结果给译码阶段需要读取的寄存器等阻塞条件后，将相关逻辑赋值到译码阶段的 ID_done 上，使得译码阶段可以在合适时刻阻塞。执行阶段若是除法，则阻塞等到除法器发出 complete 信号再拉高 EX_done。这样阻塞情况就都能够实现了。

数据前递技术有助于减少处理器的阻塞时间，提高处理器工作效率。代码中利用了寄存器将 EX 以及之后各个阶段的写入寄存器堆的数据保存起来，在判断需要前递的条件下，将相应的数据直接提供给译码阶段即可。需要注意的是，寄存器堆之外的 HI 和 LO 两个特殊寄存器同样需要考虑数据相关。另外在数据前递过程中，需要考虑前递的优先级，越晚产生的数据优先级越高。

流水信号设计图如图一所示



图一

三、实验过程（60%）

（一）实验流水账

10.6 构思 CPU 的设计。

10.7-10.8 通宵码代码，调试。

（二）错误记录

1、错误 1

（1）错误现象

条件跳转错误。

（2）分析定位过程

分析汇编代码后发现是测试关于 HI 数据相关的代码段，发现 HI 的前递出现了问题。

（3）错误原因

未考虑将 HI 前递出来。

（4）修正效果

给各阶段写寄存器的数据依照条件前递 HI 和 LO，并将这些数据依照条件前递给译码阶段。

（5）归纳总结（可选）

涉及到读取的数据一定会有数据相关，因为数据的读取和写入不在同一个时期。

2、错误 2

（1）错误现象

译码错误。

（2）分析定位过程

检查波形后发现，在译码阶段被堵塞的下一个时钟上升沿，输入的指令数据会不可避免的变成下一条指令，译码信号也变成了下一条的信号。因此便失去了在译码阶段停顿的效果。

(3) 错误原因

未在译码阶段阻塞时用寄存器存下仍处于译码阶段的指令。

(4) 修正效果

增加保存译码阶段的寄存器后，根据是否阻塞的条件决定译码阶段译输入的指令数据或指令寄存器，仿真通过，效果拔群。

(5) 归纳总结（可选）

这是个考虑不周的低级错误。

3、错误 3

(1) 错误现象

除法器结果不对。

(2) 分析定位过程

查看波形发现符号位溢出。

(3) 错误原因

我用的加减交替法，需要两位符号位，并且最终的余数和商的符号需要额外的调整，开销较大。

(4) 修正效果

改用恢复余数法，代码量减少，器件简单，效果拔群

(5) 归纳总结（可选）

四、实验总结（可选）

五级流水的数据相关的处理是流水线的灵魂，没有数据相关的处理，流水效率会极大降低。