

# 实验二报告（阶段一）

2016K8009929049

万之凡

## 一、实验任务

设计并运行静态五级流水简单 MIPS CPU，可以采用原有的 CPU 继续修改使用，也可以重新设计 CPU 以更好地适应流水结构；实现更多的指令；将延迟槽搭建完成以适应任意指令。

在设计阶段需要将实验一中转移到现有实验平台的 CPU 添加流水结构，并实现延迟槽技术，步骤分别为：

- （1）设计流水结构，保证结构相关和控制相关，暂不考虑数据相关；
- （2）实现 MIPS 架构的延迟槽技术，用于存放任意指令。

实现阶段需要实现指令的增加、流水和延迟槽的搭建，步骤分别为：

- （1）根据《MIPS 指令系统规范》对 CPU 进行新指令的添加；
- （2）添加 5 级流水结构；
- （3）给各信号添加延迟槽。

验证过程中根据 SoC\_Lite 实验环境的工作原理，分别检查 Vivado 实现的波形、inst\_ram 的仿真编译结果，在得到 pass 后上板验证。

- （1）更换 soft 文件夹内原有的仿真验证文件，Vivado 仿真运行 mycpu；
- （2）在波形正确后查看仿真结果，pass 后生成 bit 文件上板实验；
- （3）若上板显示结果正确，mycpu 验证成功。

## 二、实验设计

总体思路：五级流水线可以分为 IF,ID,EX,MEM,WB 五个阶段，前三个阶段只需要按部就班执行就可以，只有 WB 和 MEM 会产生冲突。因此前三个阶段还需要为后面的两个阶段解决冲突提供指导信号。

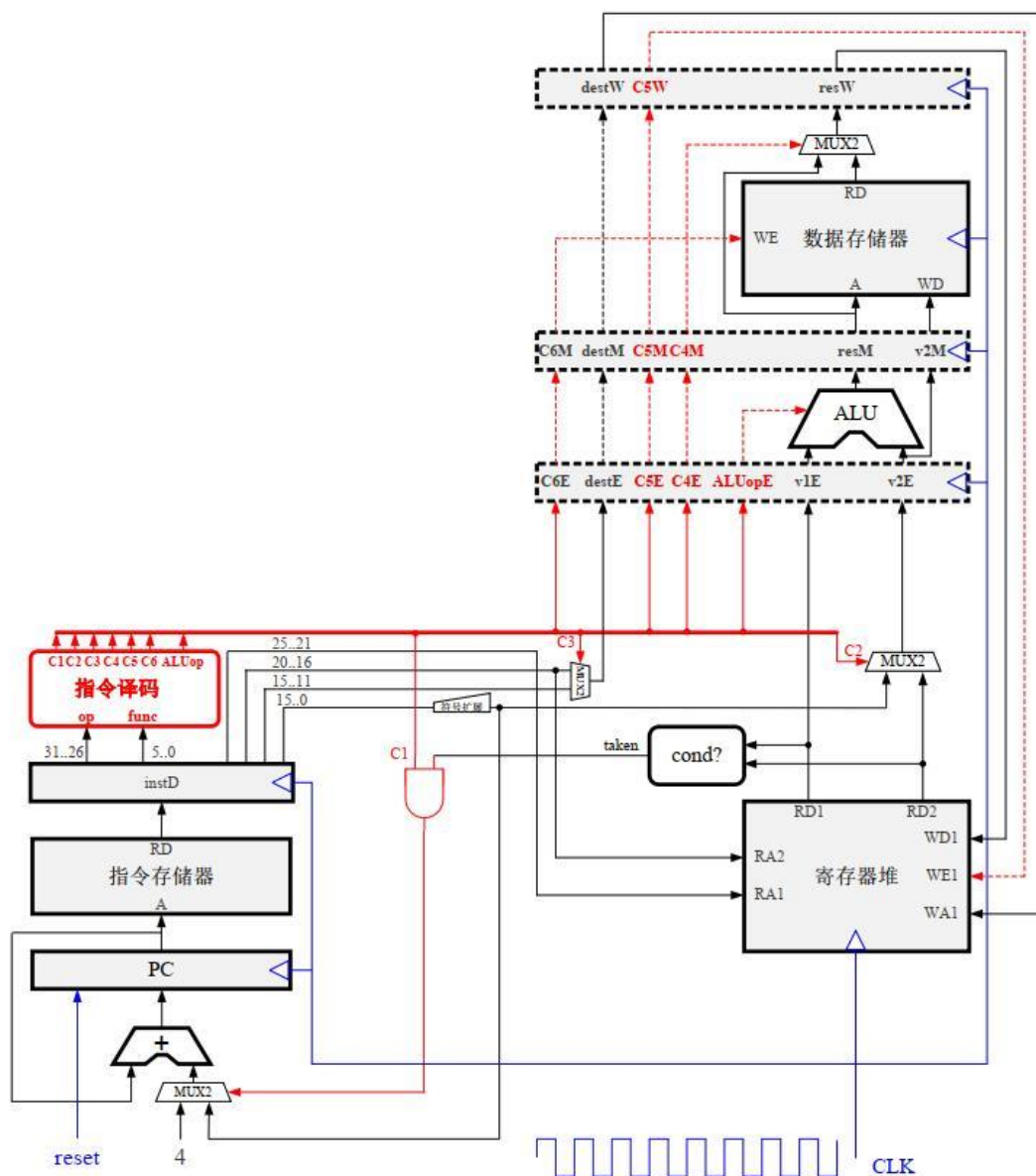
其中一个特殊的是具有跳转功能的信号，如 BEQ, JR 等。这里就需要使用延迟槽技术，但是在当前阶段的延迟槽中并不需要考虑很多，实际上只需要将传递到 EX 之后阶段的信号置零即可，就像是我们手动添加 NOP 进入这些 bubble 中。也可以根据额外提供的讲义中说的，在目前的简单静态流水线中，增加额外的计算开销，在译码阶段就能够获得是否为跳转指令从而节省流水线中的空间。

其余需要传入的信号，分别从他们的起源进行追溯即可。在现有的多周期状态机 CPU 的基础上，以 wdata 信号的来源为例：lw 这类信号需要 MEM 阶段指导；ADDU 之类的只需要从 EX 阶段读取信号即可；因此我们可以

使用 wdata\_en\_mem, wdata\_en\_ex 来分别判断。若不考虑其他因素（因为这些因素影响方式都很相似），则可以写出：

```
assign wdata = ({DATA_WIDTH{wdata_en_mem}}&&wdatamem)|({DATA_WIDTH{wdata_en_ex}}&&wdataex)
```

所有的信号都可以使用类似的方式进行分类分配，故在此基础上不断简洁代码风格、使用用 wire 信号连接上级信号即可实现 CPU 结构。



图一 标准 5 级静态流水线

### 三、实验过程

#### （一）实验流水账

2018/09/18, 根据研讨课上内容了解流水和延迟槽相关技术并开始编写试验报告；

---

2018/09/19, 根据 lab1 的 CPU 初步尝试设计流水结构;

2018/09/21, 对原有流水进行修改与检查;

2018/09/22, 编写 CPU 中需要实现的新指令;

2018/09/23, 发现自己 lab1 使用了 lsCPU 进行验收, 推翻之前的结构并开始修改 lab1;

2018/09/24, 继续 debug 并且尝试上板, 未能彻底完成。

2018/09/25, 编写实验报告。

## (二) 错误记录

### 1、使用状态机编写流水

#### (1) 错误现象

基本无法实现流水的多级功能, 性能没有显著提升。

#### (2) 分析定位过程

阅读各类资料, 进行分析修正。

#### (3) 错误原因

对流水本身的认识并不到位。

#### (4) 修正效果

将原有状态机拆散进行重组, 按照不同状态下分类部件, 修改成模块化的方式, 以此实现流水。

#### (5) 归纳总结

基础知识要到位。

### 2、五级流水内部逻辑不清晰

#### (1) 错误现象

仿真过程中在第一个检查点就会 ERROR, 但是生成 bitstream 文件上板通过。

#### (2) 分析定位过程

对比波形定位 debug\_PC 信号, 发现是 WB 写回导致错位。

#### (3) 错误原因

流水编写过程中内部逻辑出现矛盾, 应该有信号被重复调用出现结构相关。

#### (4) 修正效果

还在 debug 中, 未能修正。

#### (5) 归纳总结

先能够熟练画出五级流水结构图, 理清各个 reg 和 wire 的信号关联; 之后再进行相应的代码实现。

### 3、定制 inst\_ram 出现的问题

#### (1) 错误现象

OK 按钮无法点击、出现两个 inst\_ram.coe 且来自不同 func 等不能顺利进行仿真的问题。

#### (2) 分析定位过程

自己在网上查阅资料、从 piazza 上获得助教的指导。

#### (3) 错误原因

不熟悉 vivado 的使用、没有完全阅读研讨课讲义。

#### (4) 修正效果

---

根据助教和同学的答案进行正确操作，能够顺利进行仿真和测试。

#### (5) 归纳总结

在之前的强调之下已经对交叉编译有了比较深刻的认识，但是对于.xic 和.coe 的问题还是第一次遇到，之后也还是需要不断总结提高自己。

#### 4、发现某些变量忘记赋值、忘记连线等粗心问题不再赘述

## 四、实验总结

没能及时发现 lab1 的问题、没能及时更换思路、还有其他的外力因素（拉肚子直接虚脱了一整天）……直到截止之前都没能彻底做完很遗憾，只能说自己实力不够，之后逐渐补上吧。