

# 实验 5 报告

2016K80099  
黄熠华

## 一、实验任务（10%）

将 CPU 的对外接口修改为 AXI 接口，利用第一阶段写好的接口转换器，使得 CPU 能够通过 AXI 总线正常进行访存写存操作，并通过功能测试。

## 二、实验设计（30%）

本次实验通过状态机来记录访存取址的状态，并通过各状态机之间的相互联络来保证不产生冲突。

以取指和读数据为例，IF\_state 表示取址阶段的状态，其可能已经完成 3，也可能只实现了地址握手 2，也可能还未握手 1，甚至可能被访存级所阻塞 0。同时 MEM\_state 表示访存的状态，其可能被取址所阻塞或握手完毕 0，也可能还未握手成功 1，也可能地址握手成功 2。状态机间的通讯信号较为复杂，画图表示较为直观，请见图 1。

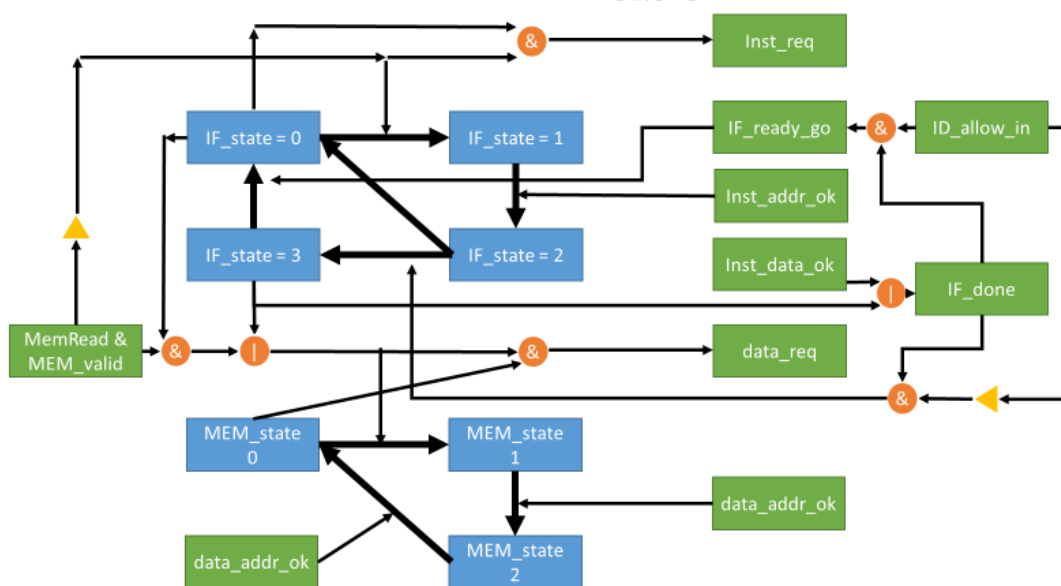


图 1

如图所示，一次只可能有 data\_req 拉高或是 inst\_req 拉高，且两状态机之间可以相互阻塞。经过调试之后，该系统运行非常稳定。

## 三、实验过程（60%）

### （一）实验流水账

12.10 构思同时读，同时读写的逻辑。

---

12.11 放弃同时读，下午构思访存与取址不同时的逻辑，并写代码，但是并未调通。

12.12 - 1.4 忙于体系结构 Lab6、操作系统实验、作业、期末考以及出国、报道等各种期末事情。

1.5 晚 - 1.6 中午(洛杉矶时间) 调试通过，发文件给同学请求帮忙验收。

1.7 晚，5-3 随机延迟后出现错误，改了一小处就过了。

## (二) 错误记录

### 1、错误 1

#### (1) 错误现象

中断发生后 PC\_next 变为 bfc00380 后迅速变会源代码段的下地址，最终并不发生 PC 的跳转。

#### (2) 分析定位过程

经过分析发现，在中断信号在一个时钟周期后丢失了。

#### (3) 错误原因

我的 CPU 的中断原理是通过 cancel 系列信号来取消某一级指令的合法性，拉低其 valid。中断发生后，中断触发指令很快被自己取消掉了合法性，但此时取址并未完成，PC 并未更新成新的值。因此中断信号丢失，PC\_next 变为原代码段的下地址。

#### (4) 修正效果

为 cancel 系列信号，以及中断信号增加寄存器来保存它们，避免它们因为触发中断的指令失效而失效。

#### (5) 归纳总结（可选）

这个错误是由取址的时序发生改变而造成的，增加了寄存器之后就可以避免此情况发生。

### 2、错误 2

#### (1) 错误现象

各种中断的 epc 发生错误。

#### (2) 分析定位过程

发现之前用于标记延迟槽指令的信号是组合逻辑，会因为时序的拉长而失效。

#### (3) 错误原因

标记延迟槽指令的信号很快就因时序拉长而失效。用个阶段

#### (4) 修正效果

为 cancel 系列信号，以及中断信号增加寄存器来保存它们，避免它们因为触发中断的指令失效而失效。

#### (5) 归纳总结（可选）

这个错误还是由取址的时序发生改变而造成的，增加了寄存器之后就可以避免此情况发生。

### 3、错误 3

#### (1) 错误现象

取址失败。

#### (2) 分析定位过程

发现取址和写存居然发生了冲突，肯定是我的状态机出现了问题。

#### (3) 错误原因

访存和写存共用同一个 data\_data\_ok，会使得取址会把写存误当成访存而被阻塞，接着因为写存的状态机并没

---

有与取址状态机的通讯机制导致取址级挂机。

#### (4) 修正效果

把取址状态机的状态转移逻辑的某处 `data_data_ok` 与上访存信号 `MemRead_mem`。然后就过了。

#### (5) 归纳总结（可选）

这个实验太简单了。

### 四、实验总结（可选）

本次实验的关键是设计出仲裁机制、访存的阻塞机制。我通过打草稿设计出了复杂的状态机以及其相互通讯的机制来保证运行正确。

除此之外基本就是为组合电路加上寄存器的大改，来保证信号不会因为时序的拉长而丢失。但是代码量不小，正好赶上一学期最忙的时间，让我很难受。忙碌了许久来到美国，第二天想歇一歇，便去了迪士尼玩了解风情，但是因为该实验还未完成而没有心思，一百多刀的门票，结果没等到晚上的烟火，在中午便急急忙忙赶回住处继续调试。

5-3 和 5-2 好像没啥区别，老师是来送分的吗哈哈。