实验 5-2 报告

学号：2016K8009929011
姓名：段江飞

一、实验任务（10%）

本实验要求将 CPU 顶层的对外接口信号修改为一个 AXI 接口，CPU 通过 AXI 接口取指令和数据。在实验 5-1 的基础上，利用完成的类 SRAM-AXI 转接桥，将 CPU_CORE 通过类 SRAM 接口连接到转换桥，然后转换桥仲裁指令和数据请求，从 AXI 接口的 RAM 中取指令和取数据。5-2 要求实现的 CPU 能从固定延迟的 RAM 中正常取指令和取数据，并通过之前的全部功能点测试。

二、实验设计（30%）

（一）CPU 结构的改变

之前写的流水线控制信号，一直都是当前级的信号由上一级的流水线控制信号控制，就是下面这种情况：

```
always@(posedge clk)
begin
    if(ID_to_EX_valid & EX_allowin)
        ResM <= alu_result;
end
```

这是由于之前设计的 CPU，这里的每一级的流水线信号(valid, readygo, allowin)表示的其实是当本级有新的有效信号进来时，才进行流水，这样每一级的有效与否并不是像讲义上标准写法那么显而易见，而且阻塞不好处理，当时这么写是因为对那个标准写法没完全理解，自以为得其意，就按自己理解写了代码，导致在一段时间内，对别人的流水线信号产生了迷惑，但终究还是一条路走到了黑，一直按这样写了下去。然而情况总是朝着最坏的方向发展，这样写一直没出问题，我觉得也是对的，但实际上只是问题还不够大，还不足以给我一记暴击，等到了总线，我想了半天感觉这个阻塞、等待实在是有些难以处理，再加上以前瞎写的阻塞，所有的问题爆发了，于是我想修改流水线的结构，任重而道远。

简单来讲也就是将之前的那样错位的控制信号修改为标准的当前级的流水线信号控制当前级的信号，也就是如下：

```
always@(posedge clk)
begin
    if(EX_to_MA_valid & MA_allowin)
        ResM <= alu_result;
end
```

这样每一级的 valid 就表示当前级是否有效，然后阻塞的时候就很好阻塞了。我最初准备先修改，然后通过

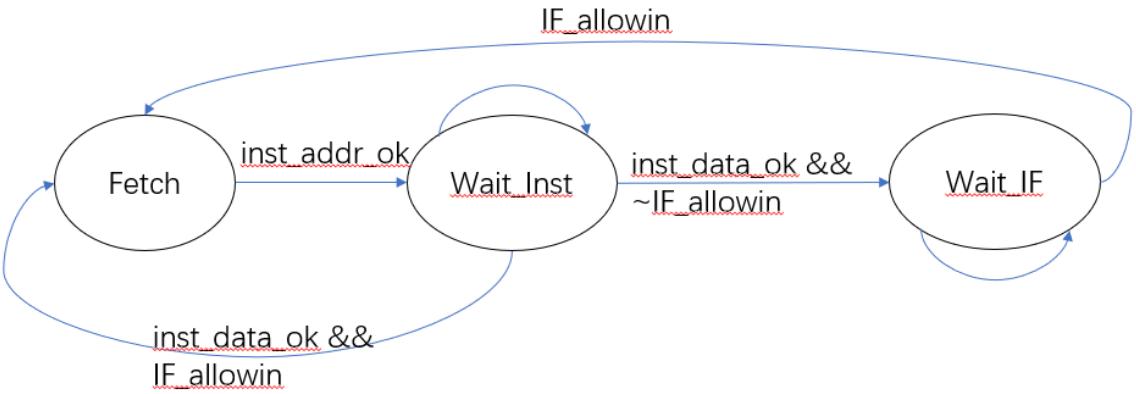
lab4 的测试，然后再去写总线，但是现实总是让人头疼，改起来是那么的复杂，中间出现的数据相关也不是很好处理（或许是太晚了，脑子不清楚了），于是决定放弃，不改了，就按原来的结构写总线，神挡杀神，佛挡杀佛。

开始写总线的时候，在开头写个状态机，这个状态机最初也没完全想清楚，然后又给访存加了一个状态机，。最初进行仿真的时候，刚开始还挺好，后来在阻塞的时候遇到了问题，我去检查我的逻辑，感觉很难处理，考虑到下周随机延迟，即便现在过了，下周可能也会爆炸，就又决定修改结构，不过这次是直接在总线实验里修改，遇到问题在直接 debug。流水线就是利用那个标准的流水线写法去写的了。

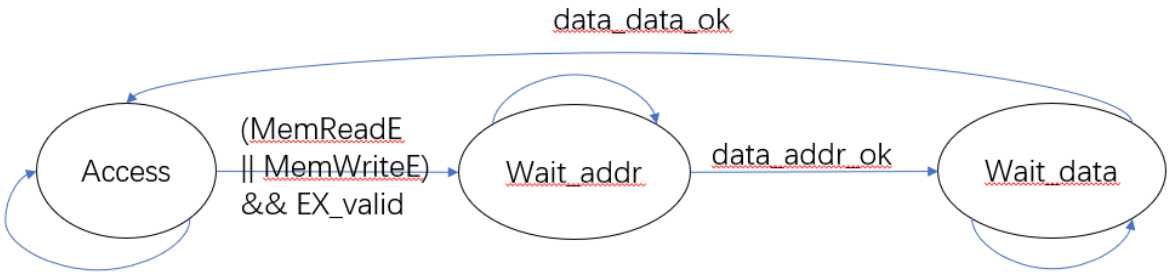
（二）总线

1、状态机实现

首先是取指和访存利用状态机控制发请求，这两个状态机主要的不同的在于起始状态不一样。指令的起始状态就是发取指请求，然后等待指令返回，如果指令返回了，这时候可以进行下一次取指令，就发取下一条指令的请求，否则，就等待可以发取下一条指令的请求。访存的状态机的其实状态是等待发访存请求，如果不需要，就直接流水，如果需要访存，就发访存请求，等待地址接受之后，进入等待数据返回的阶段，数据返回之后，就回到起始状态。



图一



图二

2、IR

由于利用了总线，取指和访存延迟都比较长，这样导致译码的时候，没有指令寄存器用起来不是很方便，而

且对译码级的阻塞也不是很好处理，于是添加了指令寄存器，译码利用指令寄存器中的指令进行译码。

3、例外处理

例外大概是加上总线之后，最难处理的一部分了。首先，考虑到例外提交时在访存级提交，例外发生时，一定不会进行访存，这时候可能正在进行取指，于是，例外的提交有两种处理方式：

一是例外直接提交，提交之后请流水线，这时候如果可以立即发取指请求，就将 PC 修改为例外处理入口，如果有一条指令正在取指，就再另外设置一个状态机，标记发生了例外提交了，等到下一次取指的时候，将 PC 修改为例外处理入口，同时这条取出来的指令也清除。

二是例外先不提交，而是阻塞在访存级，对访存的状态机进行修改，等待正在取指的指令取指完毕，将例外提交，PC 修改为例外处理入口地址。

eret 提交的处理和例外提交处理基本一致，而且更加简单。

对于时钟中断和软中断，我采取的是第一种方式，其他的例外，我所采用的是第二种方式，对访存级的状态机做了修改，添加了例外等待状态，因为例外发生时，到了访存级是不会访存的，这时候除了就比较简单了，直接进入一个等待状态，等待例外提交。

好吧，例外处理部分截止到此处，之前的是最初的设计，确实通过了仿真，但是…综合爆炸，出现了组合环，我又修改了设计，全部的例外处理都是利用第一种方式，直接提交，附加在访存级指令上，如果访存级指令无效，就等到访存级指令有效在提交。

（三）验证

1、仿真验证

在 vivado 上仿真，测试 94 个功能点，通过测试。

2、上板验证

数码管一侧从 1 加到 5E，另一侧中间卡了好多次，高低 8 位不同步累加，上板没过啊。

三、实验过程（60%）

（一）实验流水账

1、12 月 8 日下午 10 点到凌晨 3 点

理解实验，写代码，试图修改 CPU 结构。

2、12 月 9 日上午 11 点到凌晨 2 点

正是开始写总线，并找 bug，一直到通过了除了时钟中断和软件中断的例外测试。

3、12 月 10 日上午 10 点到凌晨 4 点

通过仿真测试，综合发现组合环，找出组合环（工作量翻倍哦），原想上板测试，并完成实验报告，结果上板过不了，死了。

4、12 月 11 日上午 11 点到下午 6 点

试图通过上板测试，但怎么修改都过不了，一直是 lw，sw 出错。

（二）错误记录

1、错误 1

（1）错误现象

流水线没法继续流动，PC 保持一个值不变。

（2）分析定位过程

根据 PC 变化的逻辑去查看 IF_allowin 的值，发现这个值为 0，此时不应该写入 PC，但是我继续发了请求。

（3）错误原因

根本原因是取指的状态机没有写对，没有考虑到在流水线阻塞时取下一条指令需要等待，仔细考虑了开头的状态机所需要的状态之后，将状态完善，完善的状态如图一。

2、错误 2

（1）错误现象

流水线阻塞出现问题。

（2）分析定位过程

根据比对错误找到相应的汇编，发现是 lw 数据相关。

（3）错误原因

在进行修改时，发现对流水线的控制信号有点问题，lw 的时候由于访存阻塞了，但是这时候，由于我的流水线控制信号是前一级的控制流入当前级的，这样在 MEM 阻塞的时候，EX 的有效信号还可以流水，这样阻塞就需要阻塞 EX 阶段，但是 EX 阶段本来就有对除法的阻塞什么的，这样的逻辑实在是难以处理。

而且之前对 ID 和 EX 的 lw 数据相关采用阻塞 IF 级来解决的，现在对加入总线之后，如果还继续阻塞 IF 级，处理起来很麻烦，应该是需要直接阻塞 ID 级，考虑到 ID 级都是组合逻辑，阻塞也不好阻塞，因为指令会随取指变化，于是加入了一个指令寄存器，将指令存一拍。

综合考虑，觉得重新写一下流水线的结构，改成当前级流水线控制信号控制当前级信号。

（4）修正效果

一下子通过了除了例外的全部测试。

3、错误 3

（1）错误现象

例外一直发生，一直进入 0xbfc000380 例外处理入口。

（2）分析定位过程

抓取例外提交信号，发现例外提交信号一直为高，然后抓取控制例外提交信号的信号，发现一直为由例外发生的状态。

（3）错误原因

例外提交之后，对例外信号没有清理，导致例外信号一直为 1，这样就一直在报例外。

（4）修正效果

例外不会重复报了。

4、错误 4

（1）错误现象

写入 EPC 错误。

（2）分析定位过程

根据错误指令发现是从 EPC 取值，很明显是写入 EPC 错误，找到上一个例外发生的地方，看 EPC 值的变化，发现 EPC 写入了正确的值，但是紧接着又改变了。抓取 EPC 的写使能信号，发现问题。

（3）错误原因

在例外发生的访存级，EPC 的写使能信号一直为高，导致 EPC 最后被改变为一个错误值，关键原因在于没有读写使能在例外提交时做清零。

（4）修正效果

通过了这个测试，到了时钟中断。

5、错误 5

（1）错误现象

仍然是写入 EPC 错误。

（2）分析定位过程

根据错误指令发现是从 EPC 取值，很明显是写入 EPC 错误，但是这时候，是比对和我的相差 4，我以为是分支延迟槽的问题，但是找到个例外发生的地方，发现是时钟中断。

（3）错误原因

我对时钟中断的处理也是在访存级阻塞，然后等下一次取指时提交，这时候的情况是，取指的那条指令是应该标记例外的指令，但是由于他还没取出来，之前的那一条写 COUNT 寄存器的指令已经执行完毕，等到他取指完毕，COUNT 和 COMPARE 相等了，进行了例外提交，然后标记到了 mtc0 上，这是不应该这样子的，应该标记的指令在访存级。于是，我将对时钟中断的处理采取了之前说的第一种方式，例外直接提交，清流水线，然后等待取指，取指的时候，将译码级的指令写入 EPC。

（4）修正效果

通过时钟中断的测试，但是发现软中断也会这个样子，也这样处理了，然后仿真通过了。

5、错误 5

（1）错误现象

综合的时候，出现了组合环。

（2）分析定位过程

对不起，不会分析，定位不了。

我分析逻辑分析了很久，但是找不到哪里出了问题。最后利用很蠢的方式，不断的讲可能出现问题的 assign 语句赋成常数，然后综合，看看 是否还有组合环，综合了好多次之后，重要找到那一句出了问题。

（3）错误原因

出问题的是例外提交的赋值，因为我最初采用了两种方法去进行处理例外，导致例外提交判断比较复杂，然后我也不知道哪里形成了环路，于是，我换了种方式，全部按照第一种方式处理。

（4）修正效果

仿真 PASS，且没有组合环！

6、错误 6

（1）错误现象

上板过不了….

（2）分析定位过程

通过上板现象，在测试 lw，sw 功能点的时候卡住，应该是 lw 和 sw 出现了问题，但是我没找到….

（3）错误原因

No idea.

（4）修正效果

没效果…

四、实验总结

又又写出了组合环，组合环找出来全靠运气。跟组合环过不去了，好吧，其实还是设计问题，本以为很快就做完了，组合环使得工作量翻倍…

天呐，竟然上板不过…死了