

实验 2-3 报告

学号：2016K8009929011

姓名：段江飞

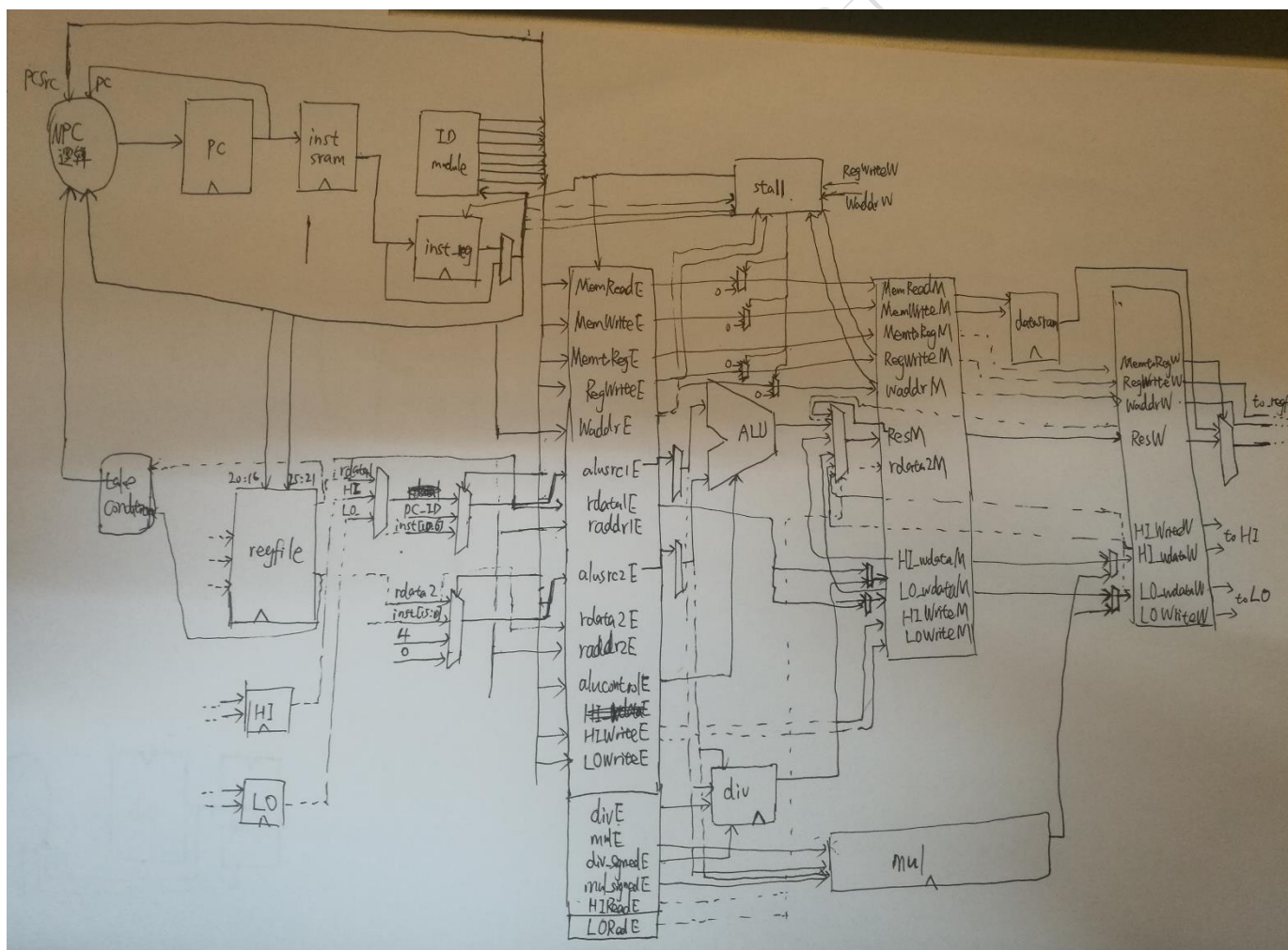
一、实验任务（10%）

本实验要求完善之前的五级流水 CPU，新添加 18 条机器指令，要实现非对齐指令，并进行性能测试。

设计的 CPU 要通过 68 个功能点的测试，仿真验证在控制台打印 68 个功能点 PASS 和最终的测试结束的 PASS，同时在上板验证时，数码管的值要从 0 累加到 68，高低 8 位要一直相等，LED 灯的变化要和讲义中一致。

二、实验设计 (30%)

如下静态五级流水 CPU 结构图



图一 CPU 结构图

（一）跳转指令

1、跳转部分

跳转部分的判断条件和之前的分支基本相同，复用之前的分支判断条件即可。J 和 Jalr 跳转就是直接跳转即可。

2、Link

新添加的跳转指令的关键在于 Link 操作，Link 操作要写入寄存器，而且 jalr 和 jr 会根据寄存器值跳转，会产生数据相关，这些数据相关和跳转有关，进行判断时单纯的按照非跳转指令的相关判断会产生一些问题，需要对跳转指令的相关额外进行判断，处理的时候同时进行处理即可。

（二）非对齐指令

Load 和 store 的非对齐指令根据 ALU 的计算结果(ResM)选择读数的地址，然后 store 指令根据 ResM 的后两位选择写入 data_sram 的数据位数并对写使能信号高低电平的赋值，对需要 store 进数据的位赋高电平，否则，赋低电平。Load 的非对齐指令在写入寄存器堆时判断的依据变为 ResW 的后两位，数据是由内存读的数和寄存器里本来的数拼接而成，因为寄存器堆的写使能是 1bit 的。

（三）实现功能

静态五级流水能正常运行，通过 68 个指令功能点测试。

（四）验证

1、仿真实验

在 vivado 上仿真，首先编译提供的测试代码，然后给龙芯 CPU 重新定制 instram，跑测试文件生成比对文件。之后，在自己的 CPU 上进行仿真实验，知道打印 68 个功能点 PASS，通过验证。

2、上板验证

将实验箱连接到 vivado，编程后看实验箱的数码管和 LED 灯的变化，数码管的高低 8 位一直同时累加到 0x44，LED 灯的颜色变化和讲义一致，则通过上板验证。

三、实验过程（60%）

（一）实验流水账

1、10 月 14 日晚上 7 点到凌晨 1 点 40

添加指令，并找出 bug。

2、10 月 15 日下午 3 点到晚上 12 点

解决 bug，并进行仿真实验、上板验证和性能测试。

（二）错误记录

1、错误 1

（1）错误现象

比对的 PC 发生错误。

```
[2040837 ns] Error!!!
```

```
reference: PC = 0xbfc54ef8, wb_xf_wnum = 0x15, wb_xf_wdata = 0x00000000  
mycpu    : PC = 0xbfc54edc, wb_xf_wnum = 0x1f, wb_xf_wdata = 0xbfc54ee4
```

图二

(2) 分析定位过程

找到错误的指令，发现是跳转指令，把相关信号拿出来，发现把 RegWrite 信号置高了，这时候我去看我实现的指令，发现忘记实现 BGEZ 这条指令，CPU 吧 BGEZ 按照 BGEZAL 解析了，导致错误。

(3) 错误原因

忘记实现一条指令。

(4) 修正效果

出现了另一个 bug。

2、错误 2

(1) 错误现象

进行比对的 PC 错误。

(2) 分析定位过程

找到对应的汇编指令，发现是跳转指令错误，看当时的跳转过程，应该不进行跳转，却发生了跳转，然后把跳转的比较大的信号拿出来看，发现错误。

(3) 错误原因

条件 BGEZ 跳转指令之后，又产生了数据相关没有解决。

(4) 修正效果

跳转指令

3、错误 3

(1) 错误现象

PC 是对的，但是写入的数据不对。

(2) 分析定位过程

先根据错误的 PC 找到汇编指令，发现是 BGEZAL 指令，然后把和这个写入数据相关的信号拿出来看。

(3) 错误原因

还是数据相关的问题，BGEZAL 是分支指令且写内存，这个时候的数据相关处理有一点问题，对代码修改之后，添加了可以判别分支跳转和同时写内存的数据相关。

(4) 修正效果

通过了 BGEZAL 的测试。

4、错误 4

(1) 错误现象

比对时写入数据错误。

(2) 分析定位过程

首先找到指令，发现是 `mfhi` 指令，我记得已经处理好了这条指令，而且这时候没有相关，但是波形信号显示的数据确实是错误的，虽然 `HI` 的几个值都是正确的，于是我看了一下代码，发现是下图这样的。

```
begin
  if(HIRead)
    alu_src1E <= HI_rdataE;
  else if(LORead)
    alu_src1E <= LO_rdataE;
  if(ID_WB_rlt1)
    alu_src1E <= wdata_mid;
  else
    alu_src1E <= rdata1;
end
```

图三

(3) 错误原因

`else if` 中少打了 `else`。

(4) 修正效果

通过了测试。

5、错误 5

(1) 错误现象

比时 `PC` 错位，当前的 `PC` 不对，但写数据和地址都对。

(2) 分析定位过程

看波形图的 `PC` 变化发现是阻塞问题，然后去看指令，看是那个地方出了问题。

(3) 错误原因

阻塞的时候，指令的流动不太对，正常的阻塞应该是在当前阻塞(`ID`)之后，后面的指令继续执行，还会在波形图上显示一个倾斜的锯齿状的 `PC` 变化，但是我的阻塞的时候，就是直接 `PC` 停滞，然后再继续执行。通过分析发现在阻塞的时候应该将后面的信号继续传递，我的好像中间的逻辑写的有点问题，能够进行阻塞，但是看起来感觉不是特别对，就对阻塞的逻辑进行了修改，去除了之前多余的判断，直接将 `Readygo` 置为 0，按照 `xx_to_xx_valid` 信号高低对控制信号等进行赋值，为高则正常执行，否则往下一拍送 0。

(4) 修正效果

通过了测试。

四、实验总结

实验中扭曲指令在组成原理的实验课上已经完成，就直接利用了之前的代码，进行修改。其他的就主要是没考虑全的数据相关。