

写在前面：

这篇文章作为写计算机系统实验时的每日代码心得，待到代码全部写完之后会以 pdf 形式上传到到 github 上，然后我们三个人用的是一个人的 cg 服务器，我发现这个效率是最高的也不用担心做重复工作（其实还有一个原因就是我的服务器不知道为什么被搞烂了，我也不知道为什么 sos 自从模 60 之后就再也用不了了）

qwj 11.2 把模 10 和模 6 都给调好了，感觉不是很难哈

wzx 11.3 这个模六十还是比较好弄的，就是不知道为什么这个的调用机制其实我没太明白，这俩放一起怎么就能直接用了，不过反正是成功实现了，具体问题到正经实验上再说吧

wzx 12.4 没想到下一次写出东西居然已经是一个月之后了，这个一号点太难了，完全没思路啊，我觉得我现在最主要的问题是会不会用这个波形图看问题，我问了同学他们说这恶鬼地方除了具体的数据通路，好像还有问题要解决，我们这连用 ori 的点都过不了，这可咋整。

wzx 12.6 好紧张好难受，这两天打游戏都玩不进去压力好大啊，这个代码还是不懂，加什么东西啊这破一号点是就过不去了吗，这可咋整，我看别人都过一号点了，我们还蒙圈状态呢

wzx 12.11 我觉得这么难受下去也不是个事，我去问了问过了一号点的同学，他们跟我说这个地方有数据相关，麻木了，我计算机体系结构的课应该认真听的，数据相关是啥来着，我去看一下吧

wzx 12.12 看完数据相关那章我明白了，合着就是有个读数据读早了，所以这个地方要用一个 forwarding 的前递技术，害我还以为是什么高端玩意，我试一下吧，而且我问了一个同学关于怎么看波形图，我觉得我现在波形图也是基本能看懂了，这下把这个数据相关的对应几个什么 ex 到这几段的东西都补上了，好像下面就是加 subu 指令了，这个应该不是很难吧

wzx 12.12 夜 把 subu 等几个简单的算数指令添加了一下，下面是跟跳转有关的指令了，这个应该也不难，感觉添加指令就是对着插口接，也没啥太大难度，感觉总算是有一点上道了

wzx 12.14 新冠，寄！实验课怎么都讲到乘除法了，这是啥玩意啊，算了保存了，病好再说

wzx 12.18 好一点了，把几个跳转指令给写了，其实差别不大，主要的区别就是在于有的代码不只是在 id 端译码那么简单，一号点 pass！

Qwj 12.18 和老魏一起添加的指令

Cyb 12.21 二号点有个 stall 的问题，加确实是加了，但是没有反应，不清楚为什么没有反应。

Wzx 12.21 刚刚考完数据库，继续搞这个 stall 问题，不是说搞好一号点就搞一半了吗，怎么刚过个二号点就又卡住了，麻木了，这个 stall 原来是在 control 文件里实现的，这里说主要有 load 相关还有一个 ex 相关，不过这个点应该是 load 相关跟 ex 没关系，那就先这样。

Cyb 12.21 晚 老魏发现 core 里有根线注释掉了，少连一根线，当然过不去。老魏把 stall 问题解决之后就该加指令了。

Cyb 12.22 添加了 subu,jal,jr,addu,sll,or,xor,bne,sltu,slt,slti 指令，过到 9 号点。

Cyb 12.23 添加 j,add,addi,sub,and,andi,nor,xori,sllv,sra,srav,srl,srlv 指令，过到 36 号点。

Cyb 12.24 早上起来发烧了，没心思写了。

Wzx 12.24 ok 经过这一系列代码的添加现在到了激动人心的时刻，我们来到了 hilo 双寄存器这个位置了，自从过了二号点之后过点速度飞快，确实是万事开头难

Cyb 12.26 烧退了，感觉好了一点，添加了 mflo 和 mfhi 指令，以及 hilo 指令相关的线

和寄存器，但是添加完成后没有反应，还是卡在 42 号，根据下面报的错寻找原因，未果。

**Wzx 12.26** 现在要处理一下 hilo 寄存器莫名其妙有 reg 没赋值的问题，找一下之后完全没发现为什么赋值的使能线没动过？？？

**Wzx 12.26 夜** 四个小时过去了，我终于找到了问题，mflo 指令按理说是取值，但是如果这个是第一条和 hilo 寄存器相关的指令的话，那肯定是取不到值得，因为没有赋值语句，于是我往前面找，找了好久才发现，蓦然回首那人却在灯火阑珊处，我一开始以为需要的指令肯定也是跟 hi/lo 相关的，没想到就是前面一句那个不起眼的 div，除法指令原来也是需要 hilo 寄存器的（都怪我当时讲乘法指令的时候生病了在摆烂睡大觉没注意），于是我开始复习之前的乘除法的视频，但是我在看助教的视频时候我发现我和助教的代码逻辑有一点不同，我对 hilo 是单独开线的，不过我觉得这个问题应该不大，而且助教的那个除法给的线和一开始文件里有的线好像不太一样，我就按照思路自己搞了一下，最后算是成功的把除法搞出来了，至于乘法就是填个 rdata，没啥技术含量。在这个点过后我们的代码迅速达到了 58 号点。

**Qwj 12.26** 今天在继续跟老魏一起找代码的 bug，不得不说这个 bug 找了这么久是真的很麻人

**Wzx 12.28** 今天是找 bug 的日子，因为之前我就已经把所有的指令都添加了，但是我们的代码仍然没法通过 58 号点，于是我开始找 bug，一开始是对着 id 端检查代码出了什么问题，因为代码出错虽然说取值的地方，但是经过上次的问题，我估计问题可能不是出自读指令，而是出自前面的 sw，在找了一段时间后，我改掉了一个 sw 写线的问题，自以为完成了（但是事实上是仍然没成），于是我去 ex 端找问题，最终在在波形图中发现一条线的位宽是有问题的，于是发现是万恶的 sram 与 ram，我们在前面定义 sram 时给 sram 设置为宽的时候，吧 sram 写成了 ram，于是在后面 assign 的时候位宽被默认掉了（应该是这样），于是我+上了 r 字母之后，太不容易了，我们的代码立刻通过了 64 号点，感恩。