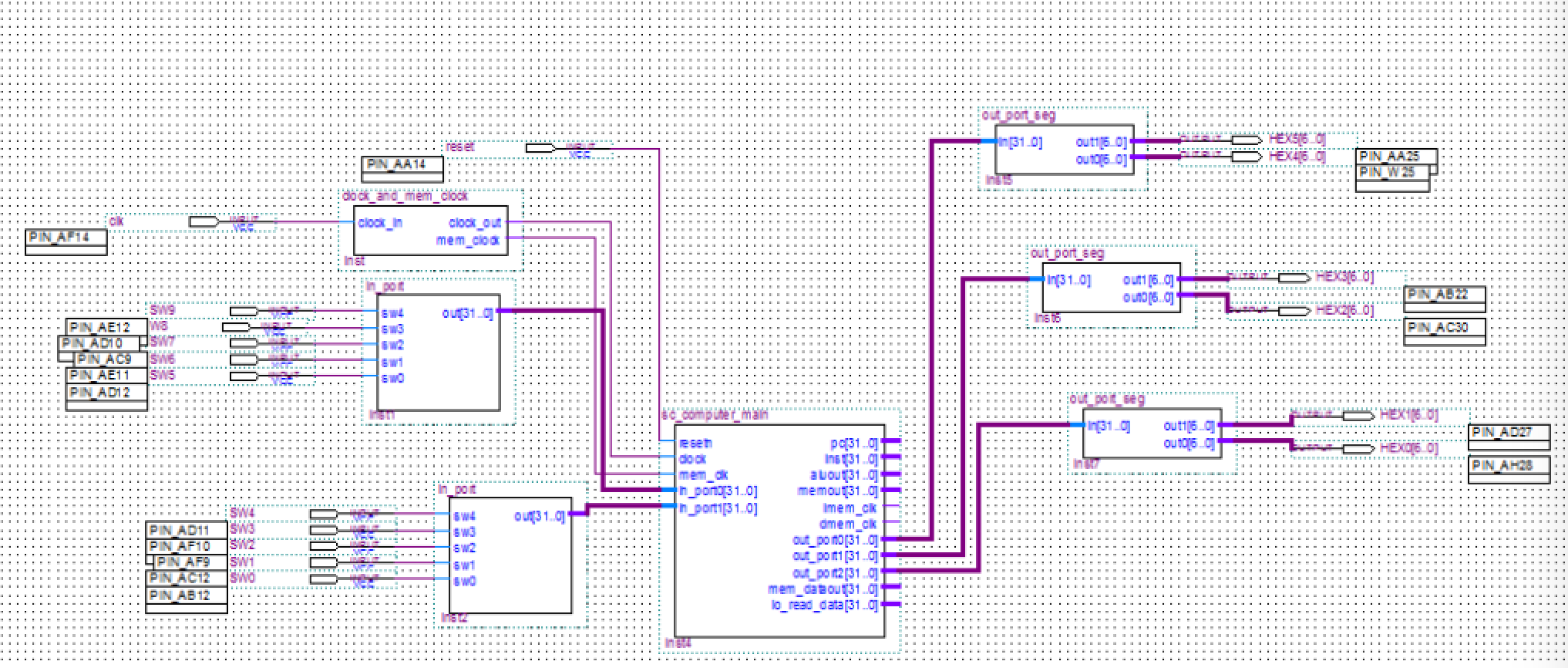
**流水线cpu实验报告**

515030910211 姜子悦

1. 实验目的
2. 理解流水线cpu
3. 掌握暂停、转发等流水线技术
4. 学会了模块化的电路编程
5. 实验电路



1. 实验步骤

第一步：对单周期cpu的主电路进行了修改，将sc\_cpu模块改为pipeline模块，代码来自实验书的设计。

第二步：五个流水线寄存器写好，该步骤比较简单，只需要reset置0，其余赋值即可。

第三步：将ir、mem、wb三个模块写完，只需要在单周期的基础上修改即可，inport、outport、mem等都直接使用单周期时的模块即可

第四步：写exe模块，计算alu本身与单周期时一样，无需修改，修改了数据来源

第五步：实现最难的部分——id。采用逐步处理的思想，先确定所有input，然后确定所有output，对于每个output，想明白它是如何计算出来的，然后针对output，实现id内部的模块和连线。

第六步：修改id中的cu模块。在单周期的基础上，加入转发（forward）逻辑需要的信息。

第七步：仿真调试，然后在板子上测试，使用了上次一样的加法汇编代码

1. 结果分析与改进

跑出来的结果没有问题，不过其实对cu中转发的信号处理还是不太明白，照着ppt上的代码输入了这些判断逻辑，其中的原理只明白了通过几个信号判断是否有目前指令需要读后面还未写回的寄存器，然后能转发的转发，load这种单靠转发不行的，再加入stall。具体转发的判断，不是特别明白。

1. 实验总结

这个流水线cpu应该是这学期目前为止写过最难的数字部件作业了，本身对着电路图弄清各个信号、设计相应模块就比较复杂，去回忆ics课的内容理解处理hazzard的电路实现更加困难。

在写流水线的过程中，觉得比较有用的一个方法是先确定一个模块的input、output然后根据每个output怎么生成的来进一步设计模块。还有调试过程中modelsim确实是个神器，上次单周期因为比较简单，没有怎么用到modelsim，这次模块较多、比较复杂的流水线，用modelsim调试清晰又能定位错误发生在哪里。

相对来说这个实验的电路还是对初学者很友好的，电路本身十分准确、完整，而且又不像以前ics学的y86等流水线流程中还要考虑jmp之后执行错误指令需要用bubble排空，这些hazzard的处理。

感觉收获十分大，从一片空白开始写一个可以跑汇编的流水线十分有成就感。