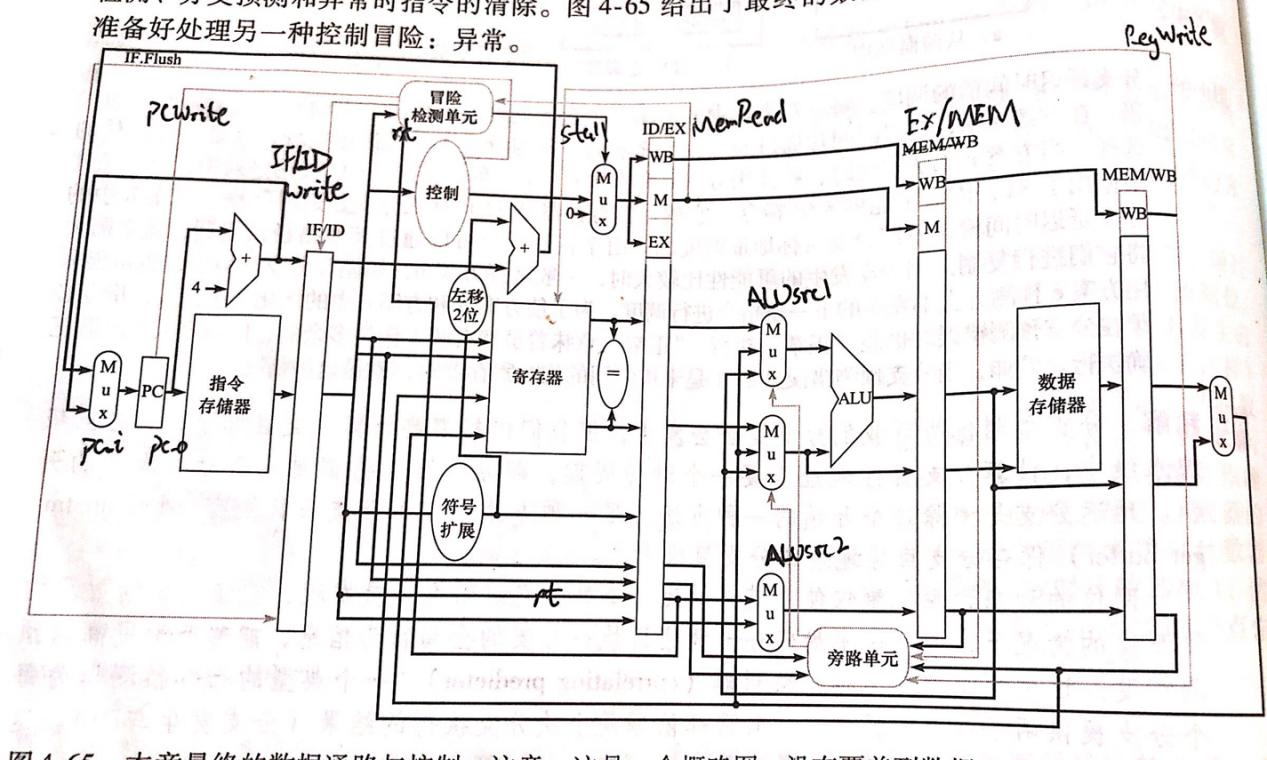
2018300003058辛嘉宇 计算机系统基础2实验5

--多周期cpu设计

数据通路近似如下：



流水线cpu主要设计思路：

将指令的执行分成5个阶段：IF、ID、EX、MEM、WB，分别执行读取指令、译码、执行、访存、写回的功能。

冒险处理思路：

1、对于EX-EX、MEM-EX类的数据冒险，使用书上给出的旁路结构以及判断条件即可。

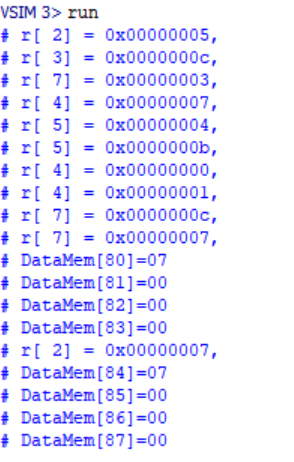
2、对于sw产生的对rs寄存器的数据冒险，无论上一条指令是R型或是装载指令，都使用一条从RegWriteData（WB级多选器后）连到MemWriteData（MEM级DM的写数据输入）的一条新旁路来解决冒险问题。R型指令和装载指令在WB级多选器后得到的结果一定是最新的，所以可以完美解决冒险问题。

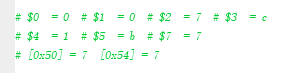
3、对于jr、jalr类寄存器跳转产生的寄存器数据冒险，选择在EX阶段进行处理，即进行阻塞直到得到跳转地址，直接使用EX阶段的第一个旁路单元多选器后的结果作为寄存器跳转地址，本质上该种数据冒险和普通的数据冒险原理相同，

4、对于分支类指令，设计一个Detect单元来对静态预测进行正确性检验，输入NPCType信号（PC+4/Branch/Jump），输出最终PC的选择信号NextType。我设计的为预测分支总是发生的情况。

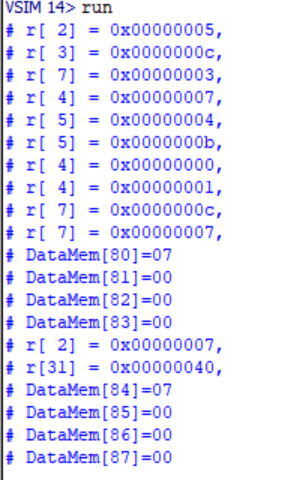
测试截图如下：

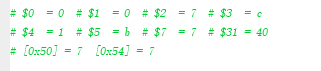
mipstestloop\_sim:



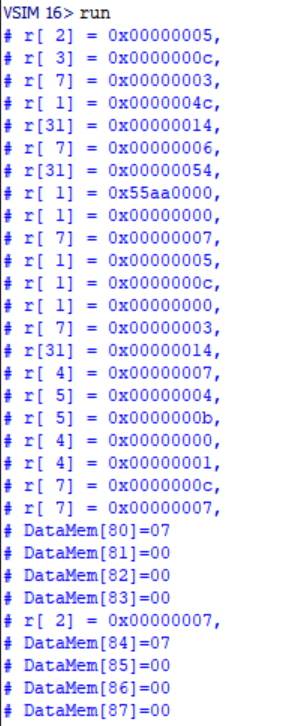


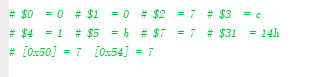
mipstestloopjal\_sim:



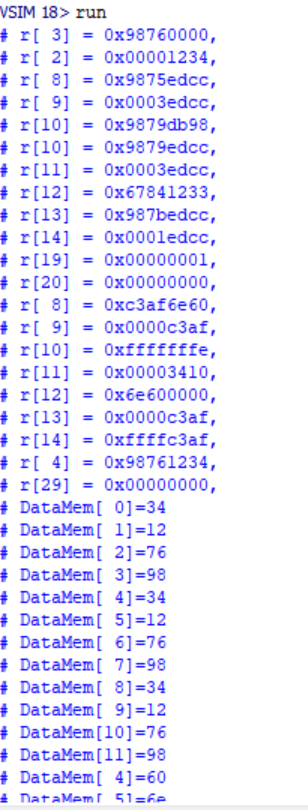


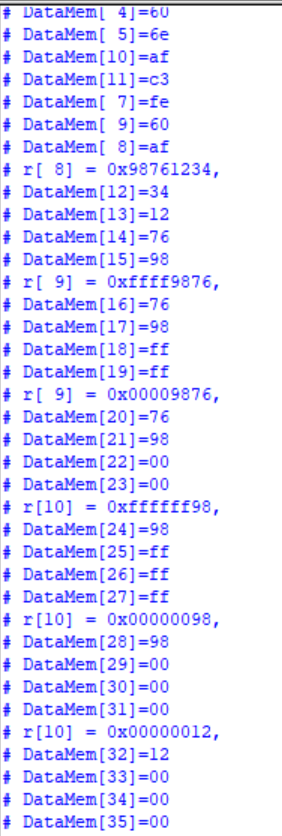
mipstest\_extloop:

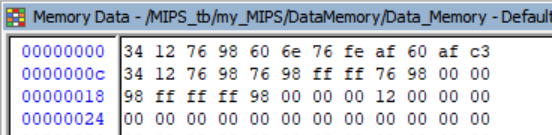


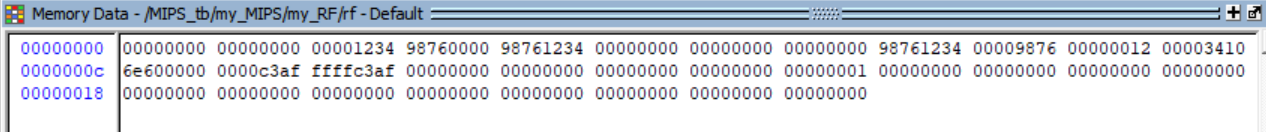


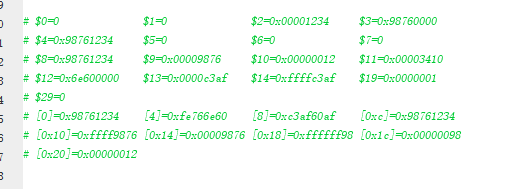
extended\_test:



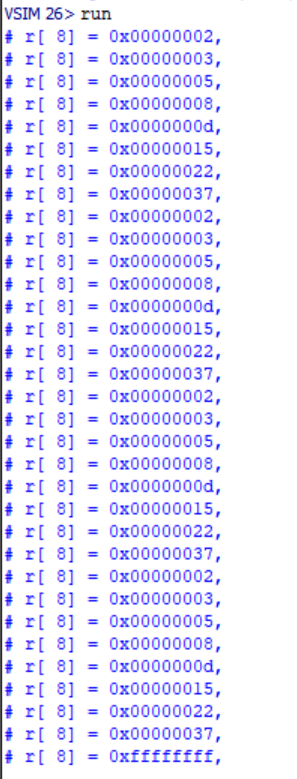




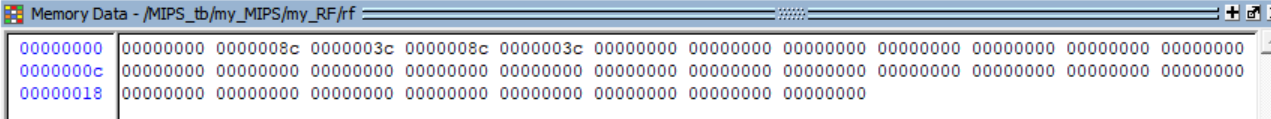


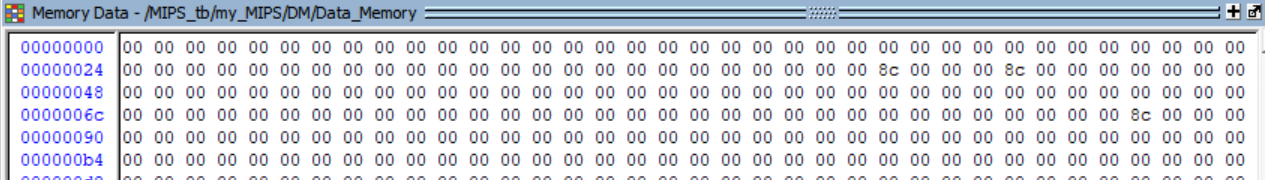


mipstest\_branch:

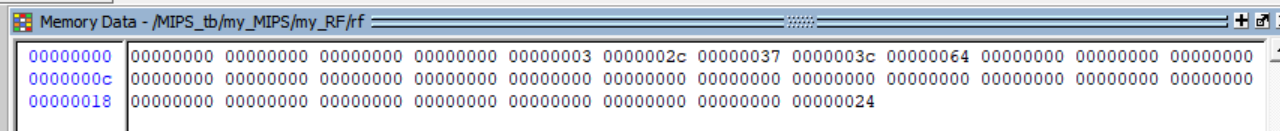


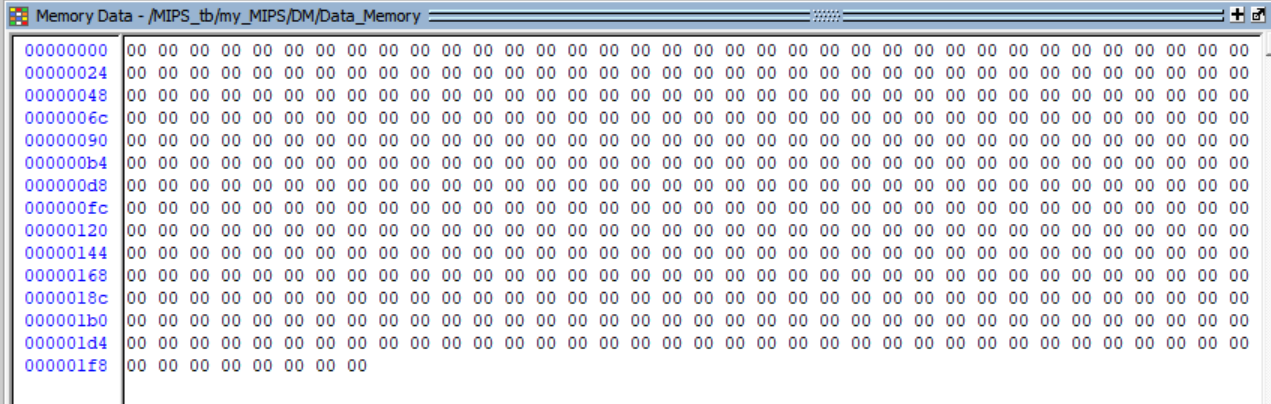
hazard\_test:





hazard\_test2:





所有测试结果均无误

遇到的问题：

在设计流水线cpu时，我感觉其思考量是多周期cpu的好几倍。有繁多的信号需要去考虑，用这个或用那个，每种信号代表的含义，以及如何正确地使用信号是顺利设计的关键。

我依然感到疑惑的一个地方是always@中敏感信号列表的内容，posedge或negedge我完全就是碰运气碰出来的，而理性的推理得出的结果写上去时发现结果又不对，所以整个人有点蒙。

另一个感到不确定的地方是阻塞信号以及PCsrc的产生源。一开始我尝试过在Control\_Unit中产生PCsrc，但后来发现存在总是无法得到正确的结果。后来发现需要结合预测的结果进行分类讨论，因此PCsrc应该放在Hazard\_Detect单元更为合适。到最后发现阻塞相关的信号可以全部由Hazard\_Detect单元负责，将相关的信息输入后统一由它输出相关的阻塞信号。

总结：

实验过程中参考了同学的设计思路。经过互相交流之后，对原有的结构进行了一部分的改进，精简、美化了代码格式，不过依然有很多考虑不够周到的地方，如果再要对其进行指令扩展，修改起来会很麻烦。待提升的地方还有很多。总的来说，是一次很不容易的实验经历，在debug中磨炼了心态。感觉自己动手、工程能力还是偏弱，以后希望自己加强这方面的锻炼。