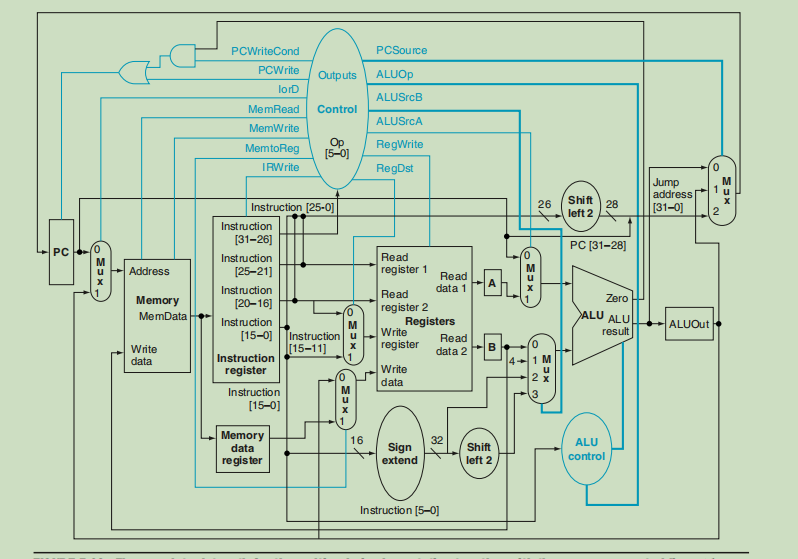
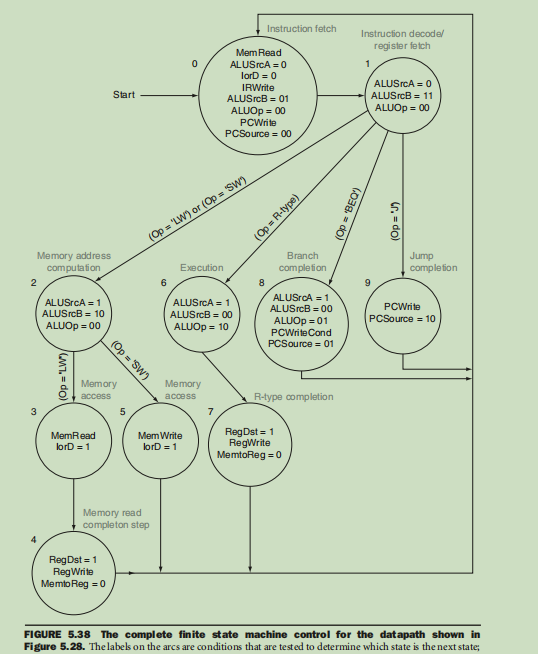
2018300003058辛嘉宇 计算机系统基础2实验5

--多周期cpu设计

数据通路近似如下：



下一状态分析：



控制信号分析：

PCWrite:0--PC不写

/1--PC写

PCWriteCond:0--当前指令不为跳转指令

/1--是跳转指令

IRWrite:0--IR不写

/1--IR写

RegDst: 00--写寄存器为rt

/01--写寄存器为rd

/10--写寄存器为$ra($31)

RegWrite :000--不执行写操作

/001--执行写32位操作(lw\R\_type)

/010--执行写16位操作(lh)

/011-执行写8位操作(lb)

/100-lhu

/101-lbu

ALUsrc1: 0--ALU第一个输入来自寄存器

/1--第一个输入来自shamt(Inst[10:6])

ALUsrc2: 0--ALU第二个输入来自寄存器

/1--第二个输入来自指令低16位(Inst[15:0])

PCsrc: 00--PC+4

/01--分支

/10--立即数跳转

/11--寄存器跳转

MemWrite: 00--不执行写

/01--执行写32位操作(sw)

/10--执行写16位操作(sh)

/11--执行写8位操作(sb)

MemWrBits:000--lw

/001--lhu

/010--lh

/011--lbu

/100-lb

MemRBits:00--sw

/01--sh

/10--sb

MemtoReg: 00--寄存器写数据来自ALU输出

/01--来自DM

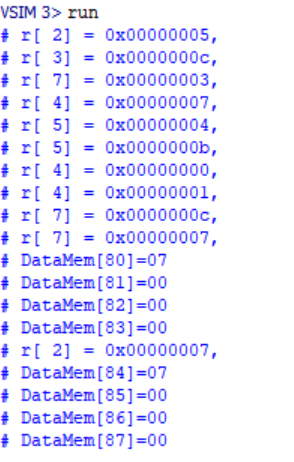
/10--来自PC+4

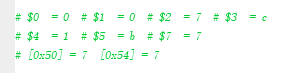
EXTOp: 0--0扩展（16->32位）

/1--符号扩展

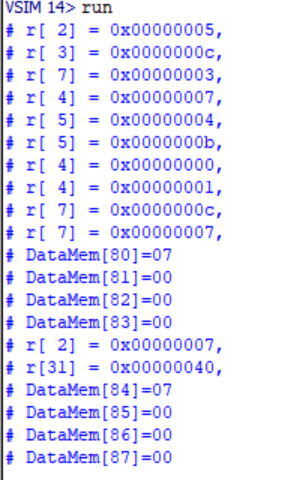
测试截图如下：

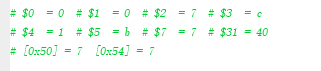
mipstestloop\_sim:



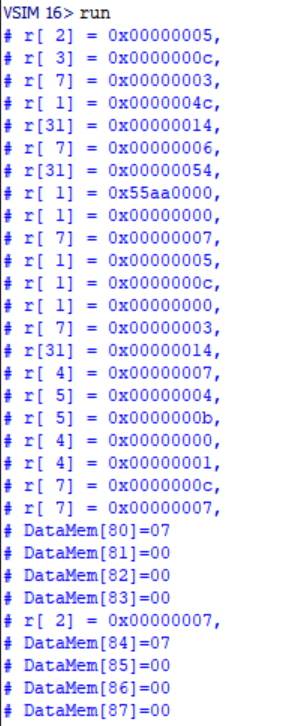


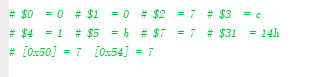
mipstestloopjal\_sim:



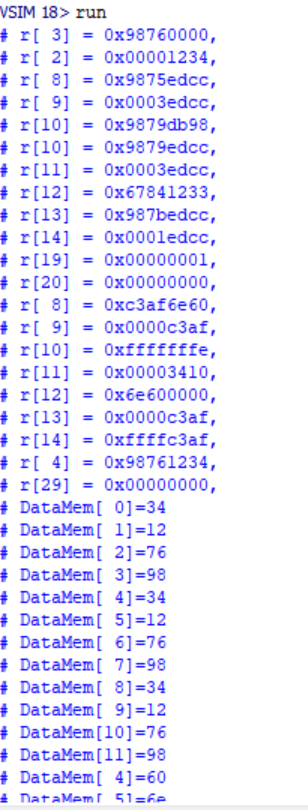


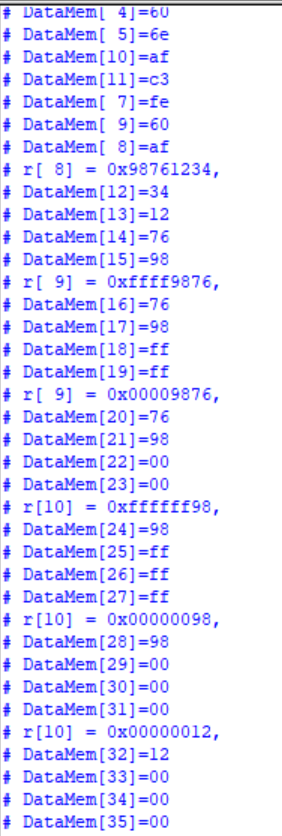
mipstest\_extloop:

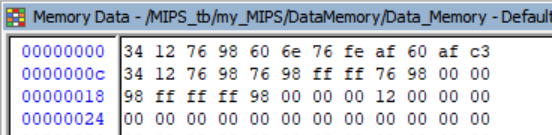


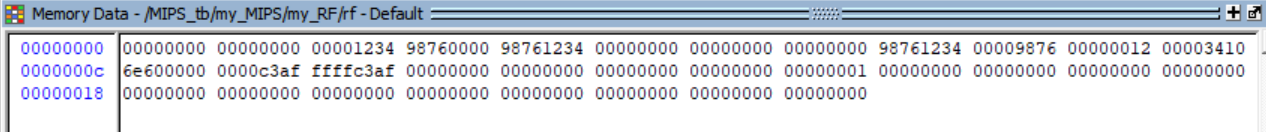


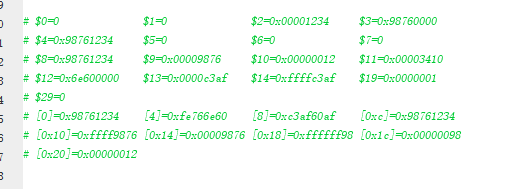
extended\_test:



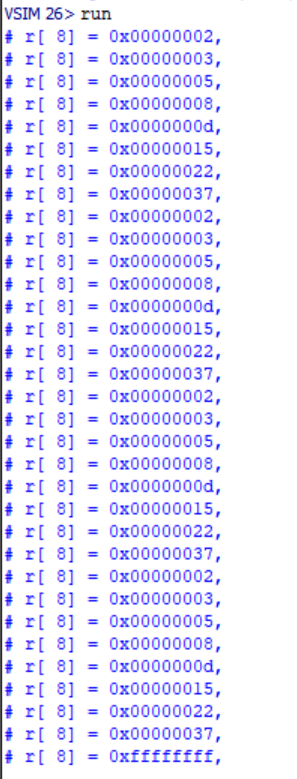








mipstest\_branch:



所有测试结果均无误

遇到的问题：

在设计多周期cpu时，我浏览了多个版本的开源教程，写法不一。自己也尝试了几种写法，比如控制单元分为5级像流水线一样，或者是像第三版书一样分为11种状态。

1、我遇到的最大的问题就是，自己在写单周期cpu时，并没有很规范地像书上一样写，变量的命名也很随意，不够规范，这带来了巨大的检查工作量，并且想要换一种思路构造的时候，需要改动的地方非常之多而且不易改动。

2、可能是受到网上其他代码的影响，对于每个模块中always中敏感变量的选择一直在困扰我，后来跟同学进行讨论之后明确了Control\_unit中上升沿发出信号，其他模块在下降沿进行响应，腾出半个周期的时间以便能正确地取到想要的数据。

3、一开始调试的时候，我是看着波形图来调的，结果发现state总是对不上号。后来跟同学交流之后才得知state是有一个延迟性的。

感想：

往后写代码时，命名最好有自己的一套规则以便于修改和debug。最好不要在某一个地方钻了牛角尖，及时和同学讨论有助于少走弯路。