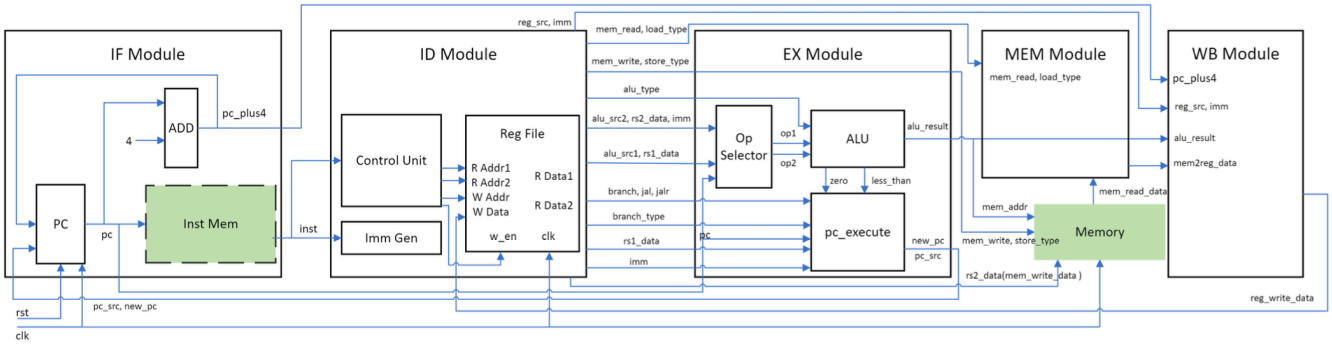
CAIC Lab3 Report

总体思路

与tutorial上的一致。注意区分写寄存器还是写。以及tutorial上画绿色的那几个部分是已经给出的部分，且在riscv_top.v中进行连接，而不是在5个stage的文件中进行连接。



以下是一些信号的选择：

- `pc_src`信号，取0标志选择`pc_plus4`，取1标志选择`new_pc`；
- `alu_src1/2`信号，取0标志用`rs_data1/2`，取1标志选`pc/imm`。

总体来说，就是默认情况下src=0，一些特殊情况下src=1进行选择。

Debug心路历程

道路是曲折的，前途是光明的，好心的助教给了我们tb和ref，现在对我debug的过程进行一个记录

Tb0: load&store

这个tb主要是找到了ld byte或者halfword时，到底load是哪一段。

```
case(load_type)
`LB:begin
    // temp[31:24] = mem_read_data[31:24];
    // mem2reg_data = $signed(temp)>>24;
    temp[31:24] = mem_read_data[7:0];//yes!
    mem2reg_data = $signed(temp)>>>24;
end
`LBU:begin
    //temp[31:24] = mem_read_data[31:24];
    temp[31:24] = mem_read_data[7:0];
    mem2reg_data = temp>>24;
end
`LH:begin
    //temp[31:16] = mem_read_data[31:16];
    temp[31:16] = mem_read_data[15:0];
    mem2reg_data = $signed(temp)>>>16;
end
`LHU:begin
```

此图中，注释掉的是错误写法，剩下的是正确写法。但是我还是不太清楚，这样ld可以保证符号位（应该在31位罢大概也许）不出错咩?

Tb1: rtype

Debug出来俩：首先是sra，记得写了signed，忘了将>>改成>>>；其次是上面的LB LH，也是这个问题。

Tb2: itype

srli和srai出了问题。发现是funct7忘截取了。一个debug经验是跟着指令一行一行来，看instr来匹配。而且大概行数*2ns就是时间。

但其实有个问题：imm和shamt的扩展。一开始我对于imm和shamt都是进行的符号位扩展，但是这次debug过程中我发现对于shamt不能如此，只能进行高位补0的扩展（因为有一个tb中的shamt是16，写成二进制是10000，进行符号位扩展后数字比较灾难）。

Tb3: btype

这个的debug尤其艰难：

- 那个less than在我的思路中其实不太好用，因为在我的思路中，ALU进行地址的加法（pc+跳转位移），判断是否跳转在pc_ex中进行，所以那个less than比较的是两个op而不是rs_data
- 一开始因为顶层模块接线问题，pc信号出了问题，再次警醒出问题的时候从顶层模块连线开始检查
- regfile的read用连续赋值(assign)比较好，且无论写不写都读，写用非阻塞，避免冲突

Tb4: utype

lui&auipc的操作，主要处理了一下符号问题

Tb5,6 : jal & jalr

与btype的debug类似，主要是分清楚op从哪里来以及write back的来源在哪里。值得注意的是lui的alu使用add的话一个操作数是imm一个操作数是0，这里我们相信regfile中取出的是0（x0）。

Tb7,8

综合测试，水到渠成。

总的来说，这痛苦的debug经历告诉了我一些经验：

- 当出现X信号的时候，基本是线没接好；
- 一些output如果要进行assign，那么左边（output）不能是reg型的
- 比较输出建议是用vscode的比较两个文件，而不是肉眼比较

提交结果

在反复提交后，收获了正确的结果。（交了这次之后，删掉了一些注释进行了重新提交）

答案正确

作者: zzq0219

Performance

score: 100

时间	ID	状态	问题	作者
2023-3-26 11:29:20	1566aa385e36	答案正确	50	zzq0219