

Lab2 Report

何跃强 PB22111649

2025/4/15

一、实验简介

实验目的：掌握五级流水线CPU的设计方法，熟悉RISC-V的指令集和数据通路，了解其设计背后的思想

实验要求：本次实验需要完成RISC-V的部分指令(算数，访存，跳转指令)，同时完成CSR指令的设计。并且通过测试

二、代码编写

阶段一：

需要完成如下指令：

- SLLI、SRLI、SRAI、ADD、SUB、SLL、SLT、SLTU、XOR、SRL、SRA、OR、AND、ADDI、SLTI、SLTIU、XORI、ORI、ANDI、LUI、AUIPC，且不需要完成数据冲突

代码思路：按照线路图完成decoder, ALU模块, ImmExtend 即可。

Decoder模块：算数指令分为R Type和I Type，只需要把根据指令类型选择OP2是读取寄存器还是立即数。ALU指令按照func3确定。如果是AUIPC，需要把OP1的输入变成PC。

ALU模块：ALU模块按照指令确定算数方法即可，需要注意的是例如SLT等需要用 `$signed()` 来确定是有符号比较。

编写测试代码：因为阶段一不需要完成Hazard模块，因此在每条算数指令中加入nop指令，其余每条指令编写一条测试指令即可。

阶段二：

需要完成如下的指令：

- JALR、LB、LH、LW、LBU、LHU、SB、SH、SW、BEQ、BNE、BLT、BLTU、BGE、BGEU、JAL
- 此时**需要处理数据相关**，实现 Hazard 模块内部逻辑

代码思路：按照线路完成需要完成的模块（Decoder, NPC_Generator, BranchDecision, DataExtend, ），同时考虑到数据相关实现Hazard模块。

Decoder模块：按照parameter.v实现编写Decoder，同时SType需要读MEM，其余按照线路图完成即可。

NPC_Generator模块：按照JALR=BR>JAL>PC的顺序进行选择，实现NPC的选择。

BranchDecision模块：按照Branch的类型确定Br是否为1。

DataExtend模块：根据访存类型（字节长度）来对取出的数进行处理，同时需要注意是否是Unsigned。

Hazard模块：需要判断几种冲突：

- 如果需要从MEM存储读出（LD类型）与后一条指令冲突，后面的指令需要停一个周期
- 如果需要从EXE计算的结果与下一条指令冲突，只需要前递即可

- 如果是JAL,JALR,BR类型，从该指令的执行一直回退到取指阶段的指令都需要变成NOP从而抑制错误。

这样，我们就完成了上面的所有指令的设计，同时也处理了冲突。

阶段三：

需要完成以下指令：

- 与CSR相关的指令：CSRW、CSRS、CSRC、CSRWI、CSRSI、CSRCI

代码思路：需要完成以下模块的设计：Decoder, CSR_Regfile, CSR_EX1

Decoder模块：按照CSR的逻辑选择是读取立即数还是读取CSR的专属寄存器，以及根据指令选择ALU的计算指令，CSR寄存器写允许信号需要为1，实现对CSR指令进行解码。

CSR_Regfile模块：按照Regfile模块实现即可，需要有0x1000个地址，同时零寄存器不需要一直保持为0。

CSR_EX1模块：根据Stall和flush指令判断是继续，暂停(stall)还是清零(flush)。

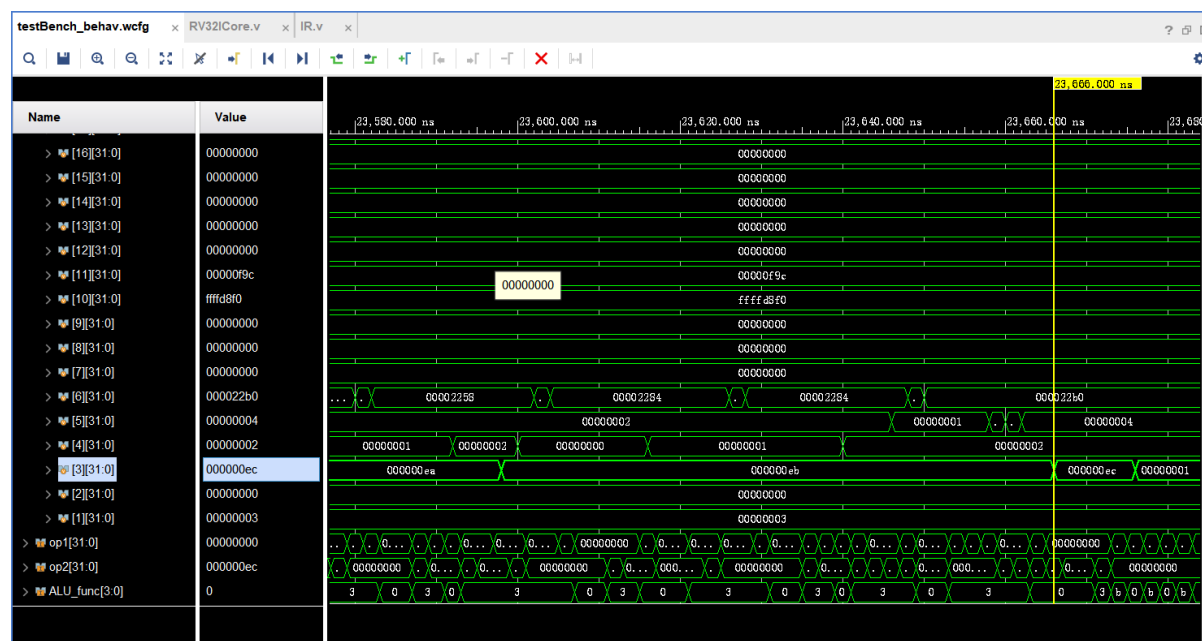
由于测试指令不要求完成CSR的数据冲突，因此不需要处理Hazard模块，至此，我们完成了CSR指令的设计要求。

具体的实验代码可以查看同样提交上去的代码的压缩文件!

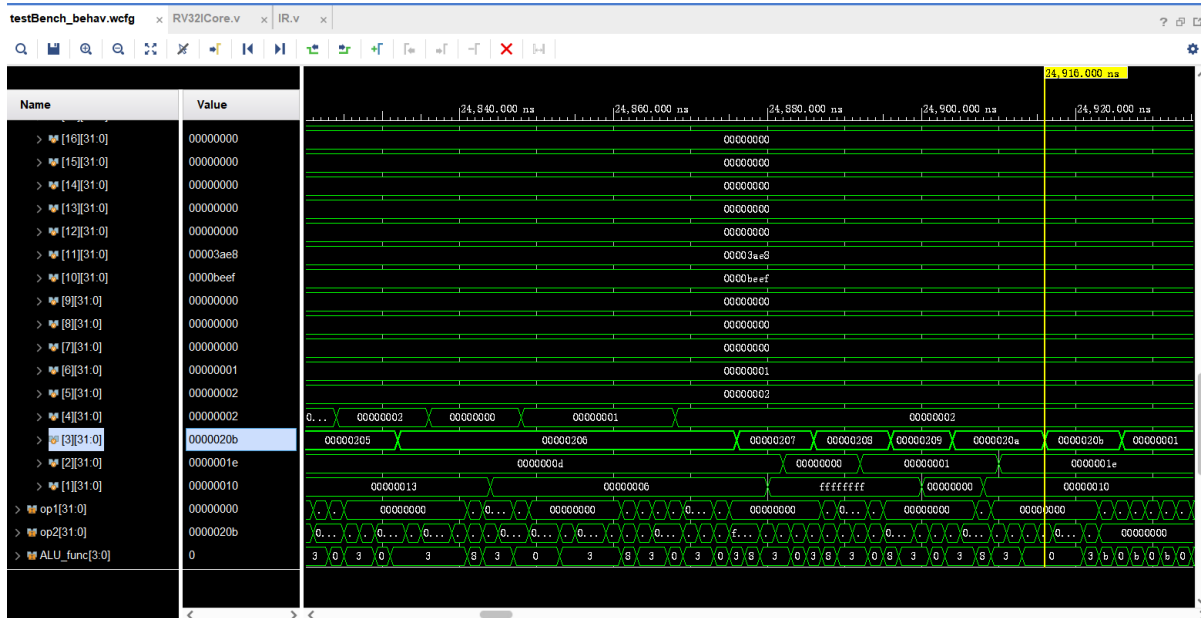
三、实验结果与仿真波形截图

通过修改仿真文件中的文件路径实现对四个测试集测试的仿真。同时可以查看三号寄存器 (gp) 内容检查是否执行成功, 如果一直递增最后保持为1即成功。

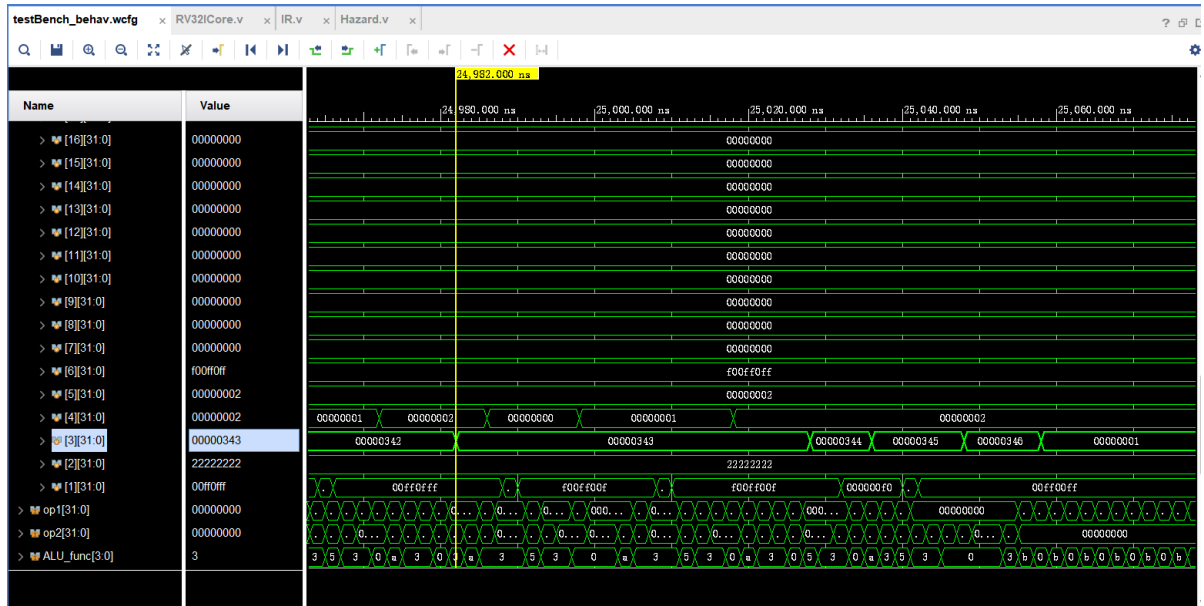
测试1：



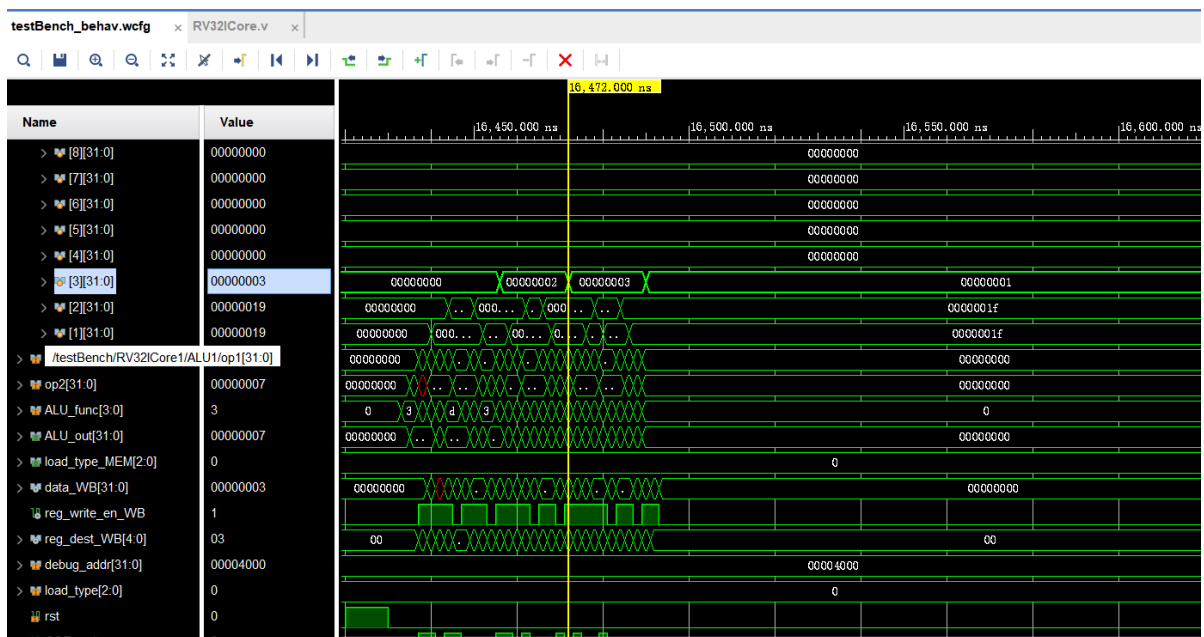
测试2:



测试3:



CSR指令测试:



我们可以关注三号寄存器，其一直递增最后保持为1，可以得知我们的测试符合预期并且代码编写通过了测试。

四、遇到的问题与解决方案

困难一与解决方案：在存储指令中无法确定SB,SD,SH等对应的Decoder该如何处理，通过查看RISCv指令手册解决此问题。

困难二与解决方案：在Hazard模块中遇到了选择与结果不匹配的问题，经过排查，发现是利用了两个if else模块，从而导致相同的条件可能会进入两个条件语句，因此经过改造，实现了合并为一个if else，解决了遇到的问题。

困难三与解决方案：CSR数据通路存在疑惑点，最终通过查阅RISCV手册解决了问题。

五、实验收获

- 1.更加了解了RISC V的设计思想与指令集，了解了如何编写测试代码对自己的verilog代码进行测试。
- 2.了解五级流水线CPU的设计思想，为未来的体系结构学习奠定了基础。

六、实验建议

希望未来能够完善parameters.v的有关store的参数。同时希望可以提醒具体有哪些模块需要完成（例如DataExtend等模块）。