

Computer Architecture Lab2

PB19071501 李平治

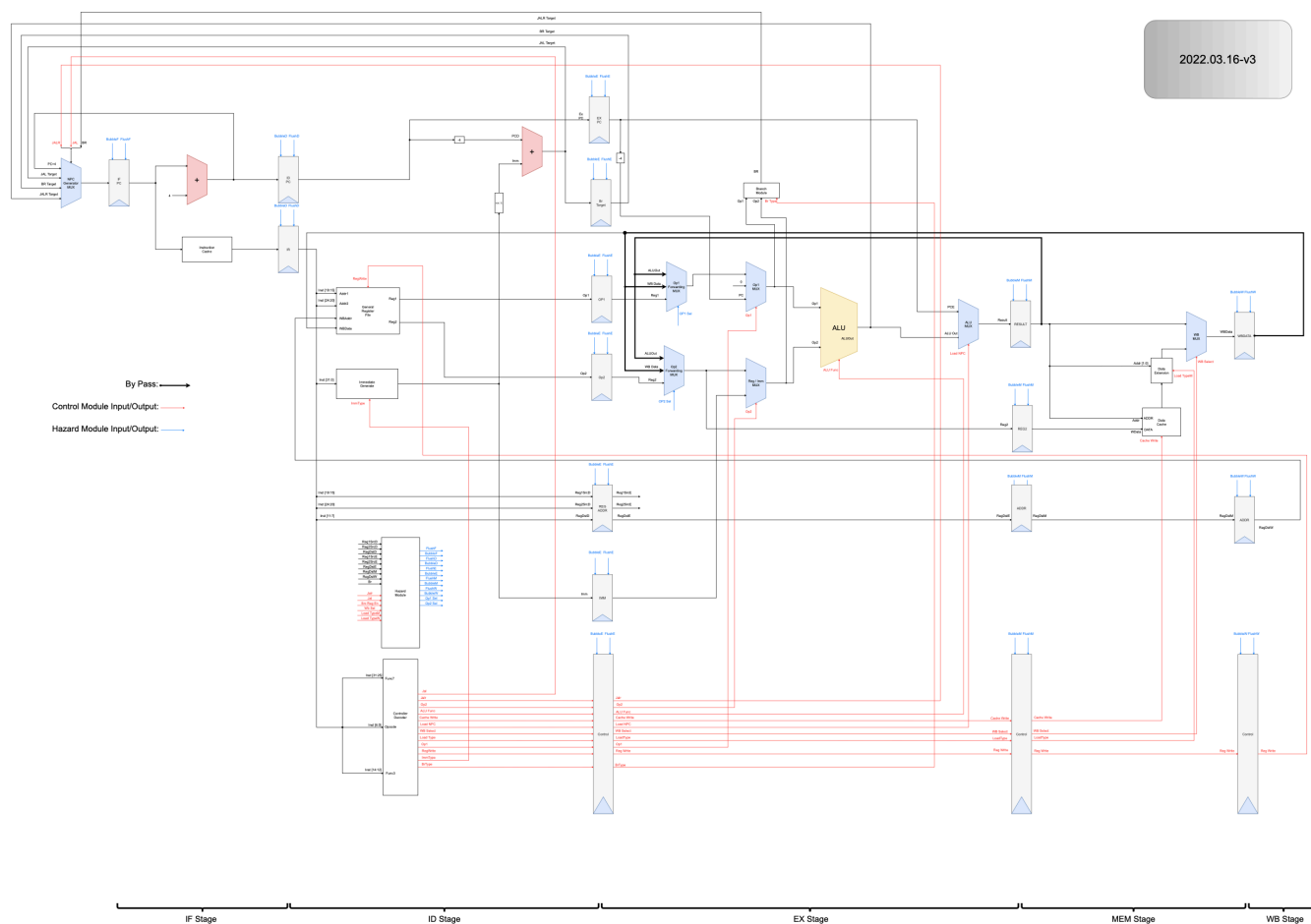
实验目的

- 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
- 进一步提高使用verilog编码和调试的能力，学会使用仿真（Simulation）调试
- 用VerilogHDL实现RV32I流水线CPU，包括 SLLI、SRLI、SRAI、ADD、SUB、SLL、SLT、SLTU、XOR、SRL、SRA、OR、AND、ADDI、SLTI、SLTIU、XORI、ORI、ANDI、LUI、AUIPC、JALR、LB、LH、LW、LBU、LHU、SB、SH、SW、BEQ、BNE、BLT、BLTU、BGE、BGEU、JAL等指令，以及CSR指令（CSRRW、CSRRS、CSRRC、CSRRWI、CSRRSI、CSRRCI），并用Hazard模块解决数据相关。

实验环境

- Vivado 2020.2
- VMWard Fusion12.1.2虚拟机中的Windows 10

实验设计



数据通路

测试：使用RISCV官方提供的测试指令

- 测试点形如：

```

1  000101e4 <test_13>:
2      101e4: 00000093          li ra,0
3      101e8: fff00113          li sp,-1
4      101ec: 00208f33          add t5,ra,sp
5      101f0: fff00e93          li t4,-1
6      101f4: 00d00193          li gp,13
7      101f8: 01df0463          beq t5,t4,10200 <test_14>
8      101fc: 14c0206f          j 12348 <fail>
9

```

- 若未通过某一个测试点，则跳转到fail代码段并陷入死循环

```

1  00012348 <fail>:
2      12348: 00000a6f          jal s4,12348 <fail>

```

- 若通过所有测试点，则进入最后的pass代码段

```

1  0001234c <pass>:
2      1234c: 00100193          li gp,1

```

并将3号寄存器赋值为1，因此最终只需要看3号寄存器，即可判断有没有通过测试

实验过程

阶段一与阶段二

1. 目标

- 完成 SLLI、SRLI、SRAI、ADD、SUB、SLL、SLT、SLTU、XOR、SRL、SRA、OR、AND、ADDI、SLTI、SLTIU、XORI、ORI、ANDI、LUI、AUIPC、JALR、LB、LH、LW、LBU、LHU、SB、SH、SW、BEQ、BNE、BLT、BLTU、BGE、BGEU、JAL等指令
- 通过测试样例1testAll, 2testAll, 3testAll

2. 过程

- 补全ALU.v, BranchDecision.v, DataExtend.v, Hazard.v, ImmExtend.v, NPCGenerator.v
- 将CSR/CSR_EX.v, CSR/CSR_Regfile.v输出端口全部赋值为0
- 运行测试样例1testAll, 2testAll, 3testAll

3. 结果

- 1testAll

[illegible]

- 2testAll

[illegible]

- 3testAll

[illegible]

三个测试均通过

阶段三

1. 目标

- 完成CSR_{RW}、CSR_{RS}、CSR_{RC}、CSR_{RWI}、CSR_{RSI}、CSR_{RCI}等指令
- 通过测试样例CSR_{test}

2. 过程

- 补全CSR/CSR_EX.v, CSR/CSR_Regfile.v
 - CSR/CSR_EX.v: 作为CSR指令在EX段寄存器上的补充, 将生成CSR指令的立即数和CSR寄存器地址放入
 - CSR/CSR_Regfile.v: CSR寄存器, 结构上和普通寄存器无差别
- 运行测试样例CSR_{test}

3. 结果

> [7][31:0]	00000000000000000000	00000000000000000000000000000000
> [6][31:0]	0000000000000000000001	00000000000000000100010110000
> [5][31:0]	0000000000000000000000	0000000000000000000000000000100
> [4][31:0]	0000000000000000000000	0000000000000000000000000000010
> [3][31:0]	0000000000000000000000	0000000000000000000000000000001
> [2][31:0]	0000000000000000000000	0000000000000000000000000000000
> [1][31:0]	0000000000000000000000	00000000000000000000000000000011
> SaveDataCacheFile[31:0]	11111111111111111101	1111111111111111011000111100000
> SaveInstCacheFile[31:0]	11111111111111111101	1111111111111111011000111100000
> [31:0]	00000000000000000000	00000000000000000000000000000000

表明测试通过

实验总结

本次试验框架设计文档友好, 代码读懂, 并有着VHDL中相对容易的debug方式。

本次试验完成了RV32I流水线CPU的实现, 加深了对RSIC-V架构的理解, 通过实践体会RSIC-V指令的巧妙之处。此外, 通过CSR的具体实现, 使得在实验一的基础上对CSR理解更近了一步。

本次试验共花费约9.5h:

- 1h阅读代码框架
- 3h首次完成代码
- 5h运行测试用例和Debug
- 0.5h总结并完成实验报告