



## 阶段一

### 阶段目标

1. 完成 SLLI、SRLI、SRAI、ADD、SUB、SLL、SLT、SLTU、XOR、SRL、SRA、OR、AND、ADDI、SLTI、SLTIU、XORI、ORI、ANDI、LUI、AUIPC 等指令。
2. 设计测试用例。
3. 在不考虑数据相关的情况下，CPU 正常执行符合测试用例预期。

## 实验过程

1. 完成 `ALU.v`, `ContorlUnit.v`, `DataExt.v`, `ImmOperandUnit.v`, `NPC_Generator.v`, `WBSegReg.v` 的修改和补全。
2. 将 `BranchDecisionMaking.v`, `Harzard.v` 等设为默认情况。
3. 编写测试用例。
4. 运行测试用例, 评估CPU设计是否符合阶段预期。

## 运行结果

因实验过程中阶段一&阶段二同步完成，故未设计阶段一专用的测试用例，略去该阶段运行结果。

## 阶段二

### 阶段目标

1. 完成 JALR、LB、LH、LW、LBU、LHU、SB、SH、SW、BEQ、BNE、BLT、BLTU、BGE、BGEU、JAL 等指令。
2. 运行助教提供的测试用例。
3. 在考虑数据相关的情况下，CPU 正常执行符合测试用例预期。

## 实验过程

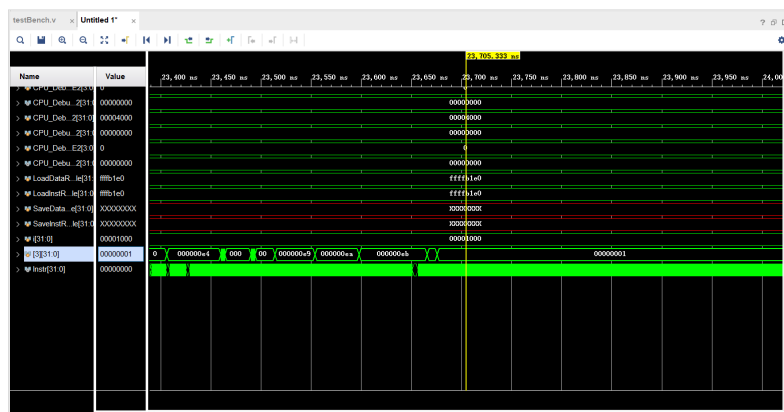
1. 继续完成 `BranchDecisionMaking.v`, `Harzard.v` 等的设计。对 `ContorlUnit.v` 进行补全, 增加对新的指令的控制。
2. 运行测试用例。
3. 根据测试用例运行结果评估CPU的功能实现是否符合预期, 并进行相应修改, 直至CPU正确运行。

## 运行结果

测试用例 1testAll, 2testAll, 3testAll 用于测试阶段一, 二设计的指令及数据相关的处理, 在通过测试点后, 3号寄存器递增, 在通过所有测试点后, 3号寄存器置为1。

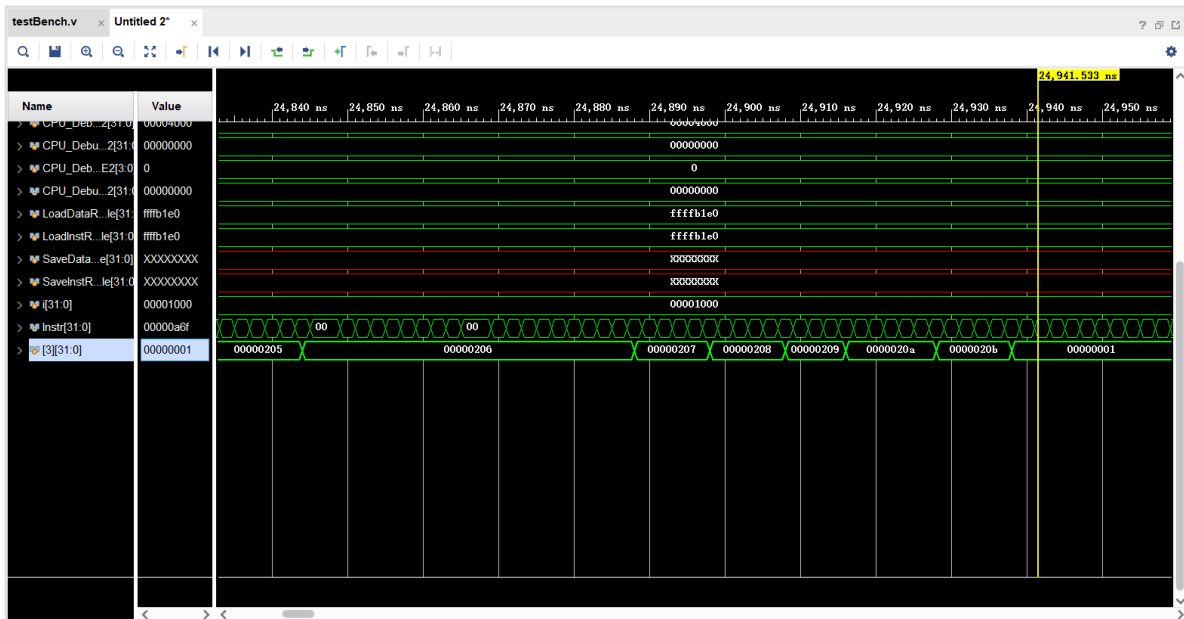
## 1testAll

运行结果如下，符合预期。



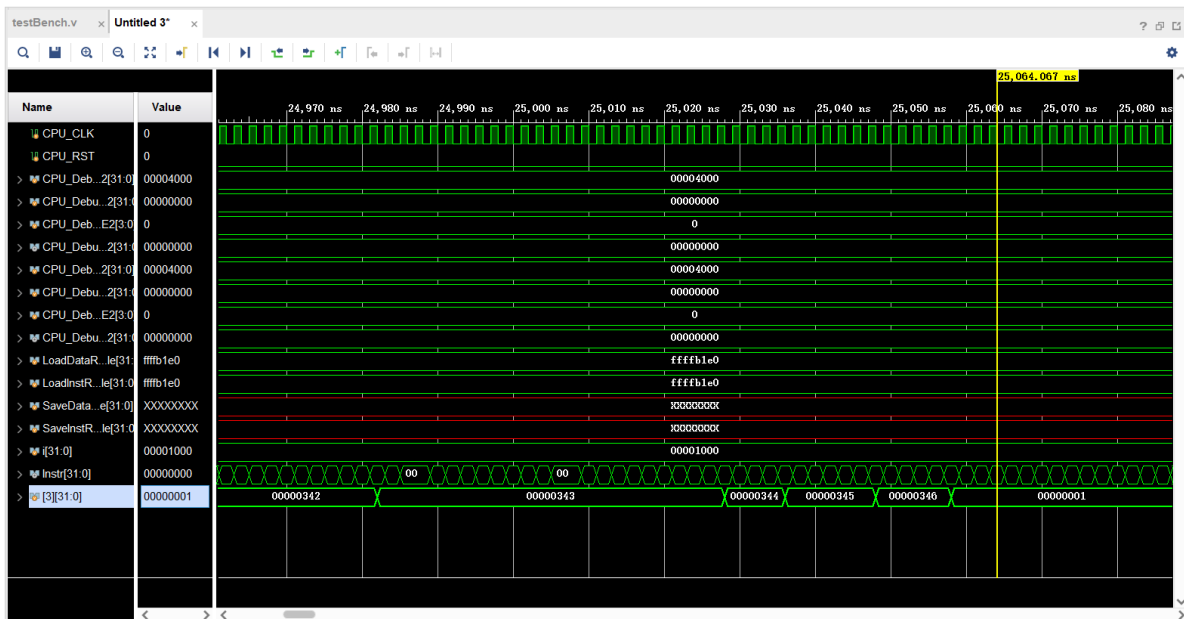
## 2testAll

运行结果如下，符合预期。



## 3testAll

运行结果如下，符合预期。



## 阶段二

### 阶段目标

1. 完成 CSR数据通路设计
2. 完成 CSRRW、CSRRS、CSRRC、CSRRWI、CSRRSI、CSRRCI 等CSR指令。
3. 运行设计的测试用例。
4. 在考虑数据相关的情况下，CPU正常执行符合测试用例预期。

### 实验过程

1. 增加CSR的寄存器堆，计算用的ALU，所需的控制信号 CSRreg\_write\_en，CSRALU\_func，CSRwb\_select，完成相关数据通路。
2. 扩展 Harzard.v，EXSegReg.v，MEMSegReg.v，WBSegReg.v 等的设计。对 ContorlUnit.v 进行补充，增加对新的指令的控制。

- 3. 设计并运行测试用例。
- 4. 根据测试用例运行结果评估CPU的功能实现是否符合预期，并进行相应修改，直至CPU正确运行。

运行结果

测试用例 CSRtestALL 用于测试阶段三设计的指令及数据相关的处理，在通过所有测试点后，指令进入 success的循环，3号寄存器置为1，否则进入failed循环。

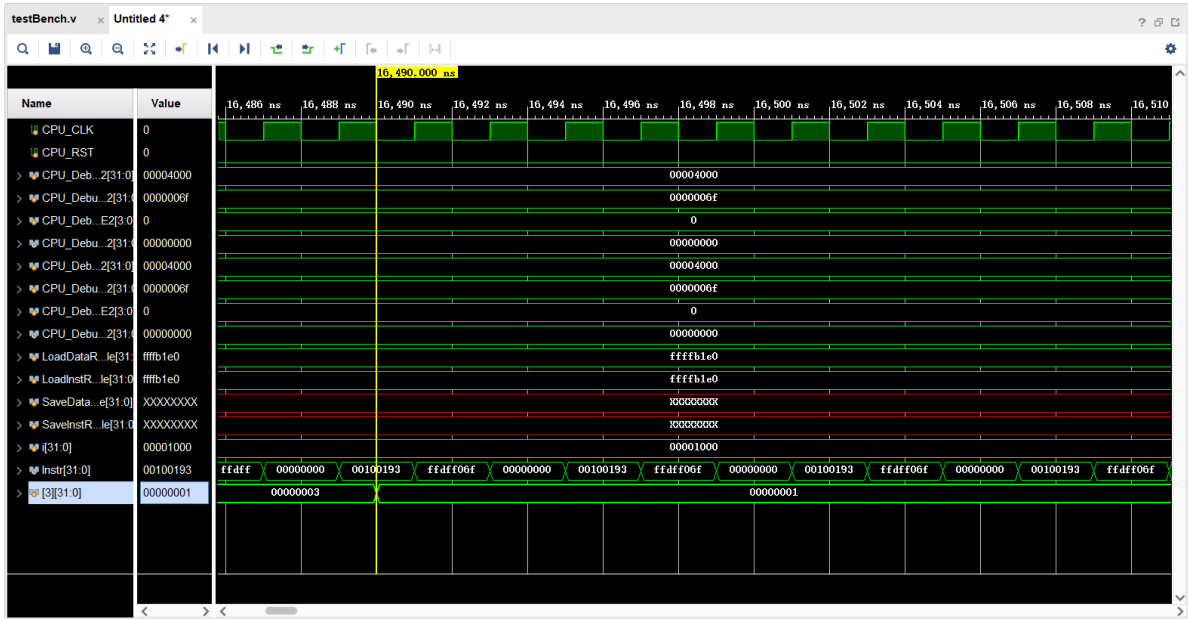
```
00010054 <test_0>:
    10054:  00000193      li  gp, 0
    10058:  00f00093      li  ra, 15
    1005c:  00009073      csrw  ustatus, ra
    10060:  00003173      csrrc  sp, ustatus, zero
    10064:  06111063      bne  sp, ra, 100c4 <failed>
    ...

000100bc <success>:
    100bc:  00100193      li  gp, 1
    100c0:  ffdff06f      j    100bc <success>

000100c4 <failed>:
    100c4:  0000006f      j    100c4 <failed>
```

CSRtestALL

运行结果如下，符合预期。



实验总结

实验过程中遇到的问题

- 1. 问题：ALU计算所用的操作数不正确。  
解决方案：修改AluSrc部分的设计。
- 2. 问题：分支指令所用的操作数不正确。  
解决方案：修改AluSrc及Forwarding部分的设计。
- 3. 问题：CSR设计工程中发现原本的设计不足以满足计算需求。  
解决方案：增加一个专用的ALU，以及相关的控制信号数据通路。

4. 问题：CSR的数据相关问题。

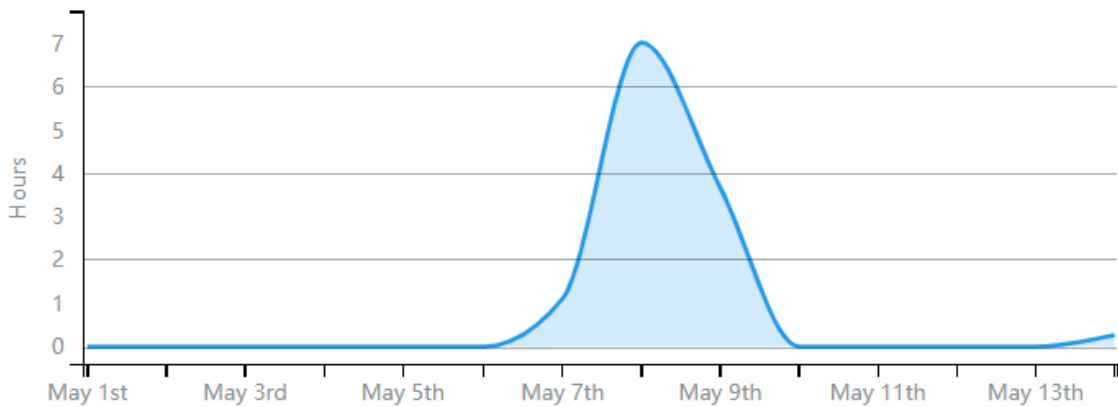
解决方案：增加CSR的Forwarding模块。

## 实验收获总结

在本次实验中，完成了RV32I流水线CPU的实现，加深了对RISC-V架构理解，通过实践体会RISC-V指令的巧妙之处。此外，在CSR相关的设计过程中，初次涉及CSR的具体实现，相较实验一的阅读文档的方式。这种方式对CSR的理解更加深刻具体。

## 实验用时安排

本次实验用时约12小时（数据来源：[CA2021 labs · WakaTime](#)），包括但不限于编写 `verilog` 代码，撰写文档，设计测试用例。



## 意见

希望阶段一也能提供测试用例。

## 附

本次实验所有代码保存于Github仓库[Alpha-Girl/CA2021 labs](#)。