Project Of SI100B EE Part

RISCV miniCPU Design in Logisim

@Chaofan Li, lichf2025@shanghaitech.edu.cn
@Yunxiang He, heyx2025@shanghaitech.edu.cn
@Qihan Ding, dingqh2025@shanghaitech.edu.cn
@Quanyu Chen, chenqy2022@shanghaitech.edu.cn
@Qihang Yan, yanqh2022@shanghaitech.edu.cn
@Kailun Jin jinkl2022@shanghaitech.edu.cn
@Jun Wang wangjun2023@shanghaitech.edu.cn

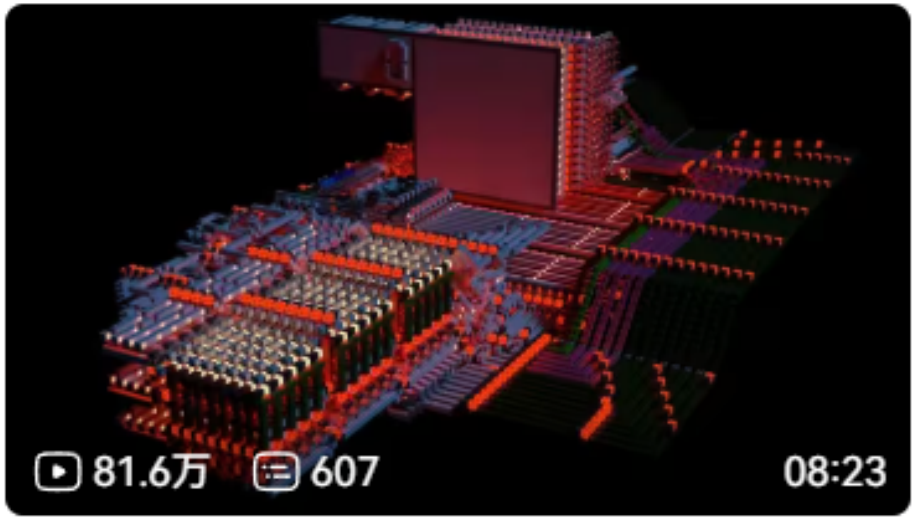
Contents

Contents

Background——What is CPU?

【Minecraft】这是你从未见过的1Hz超强红石电脑

UP 1055Lab · 2021-12-12



一台1Hz的红石计算机

UP 小涵Naiwenel · 2024-10-01

Background——Compile

C source #1

Save/Load Add new... Vim C

```c
/* Type your code here, or load an example. */
int main(){
    int a =3;
    int b =4;
    int c = a + b;
}
```

RISC-V (32-bits) gcc 15.2.0 (Editor #1)

RISC-V (32-bits) gcc 15. Compiler options...

```
main:
        addi    sp,sp,-32
        sw      ra,28(sp)
        sw      s0,24(sp)
        addi    s0,sp,32
        li      a5,3
        sw      a5,-20(s0)
        li      a5,4
        sw      a5,-24(s0)
        lw      a4,-20(s0)
        lw      a5,-24(s0)
        add     a5,a4,a5
        sw      a5,-28(s0)
        li      a5,0
        mv      a0,a5
        lw      ra,28(sp)
        lw      s0,24(sp)
        addi    sp,sp,32
        jr      ra
```

Background——Assembly

PC	Machine Code	Basic Code	Original Code
0x0	0xFE010113	addi x2 x2 -32	addi sp,sp,-32
0x4	0x00112E23	sw x1 28(x2)	sw ra,28(sp)
0x8	0x00812C23	sw x8 24(x2)	sw s0,24(sp)
0xc	0x02010413	addi x8 x2 32	addi s0,sp,32
0x10	0x00300793	addi x15 x0 3	li a5,3
0x14	0xFEF42623	sw x15 -20(x8)	sw a5,-20(s0)
0x18	0x00400793	addi x15 x0 4	li a5,4
0x1c	0xFEF42423	sw x15 -24(x8)	sw a5,-24(s0)
0x20	0xFEC42703	lw x14 -20(x8)	lw a4,-20(s0)
0x24	0xFE842783	lw x15 -24(x8)	lw a5,-24(s0)
0x28	0x00F707B3	add x15 x14 x15	add a5,a4,a5
0x2c	0xFEF42223	sw x15 -28(x8)	sw a5,-28(s0)

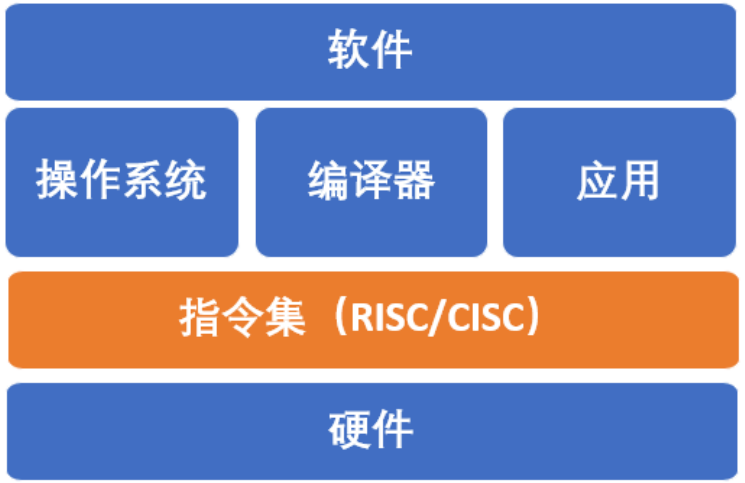
(x2)	
gp (x3)	0x10000000
tp (x4)	0x00000000
t0 (x5)	0x00000000
t1 (x6)	0x00000000
t2 (x7)	0x00000000
s0 (x8)	0x00000000
s1 (x9)	0x00000000
a0 (x10)	0x00000001

Background

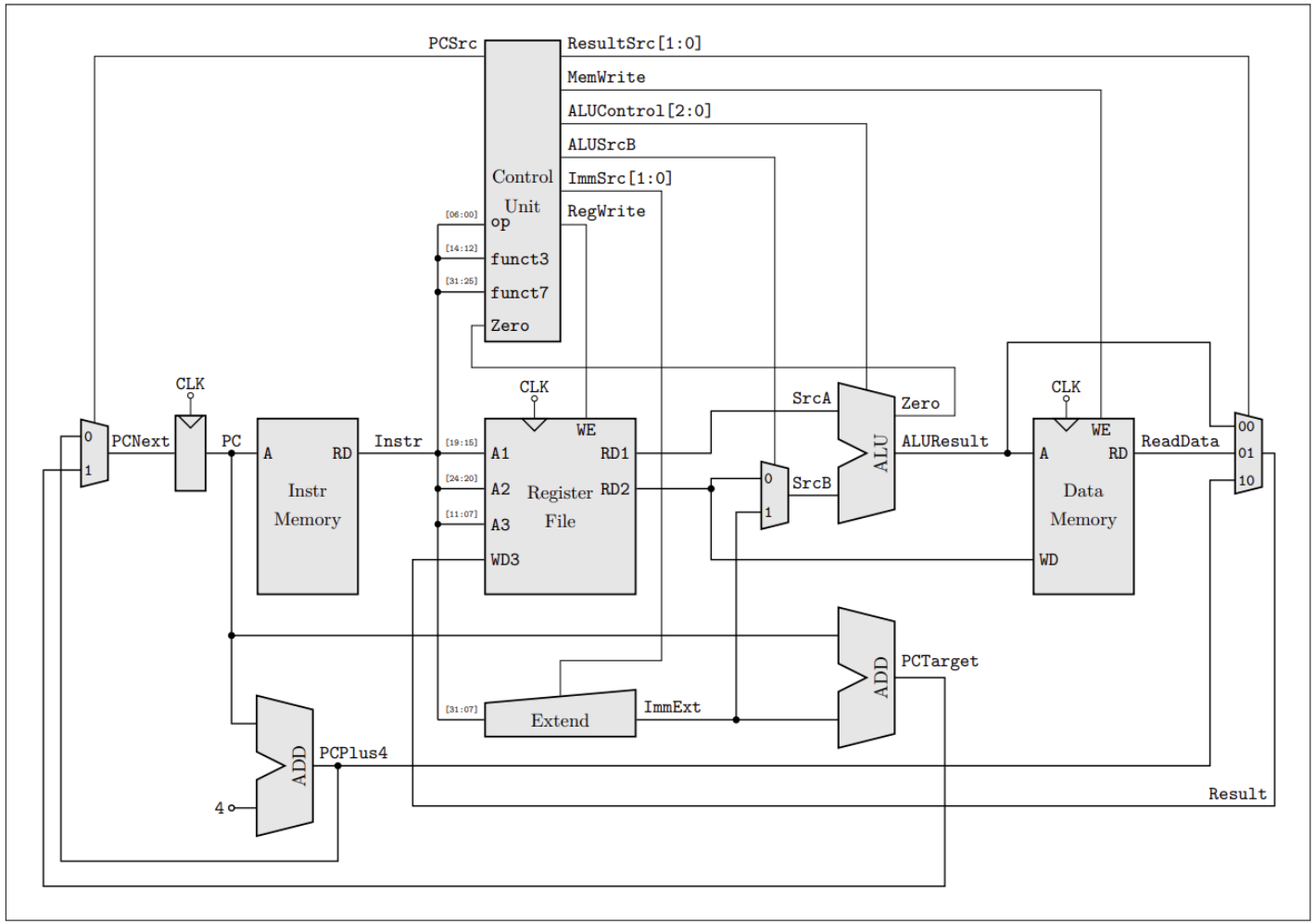
What is RISC-V?

在计算机体系结构中，**指令集架构**（ISA）定义了处理器的基本指令规范，而**微架构**则负责具体实现这些指令的硬件电路设计。

主流的指令集包括 x86、ARM、MIPS 等。与它们不同的是， RISC-V 作为一个全新的**开源指令集架构**正在迅速发展。与闭源的 x86 和 ARM 不同，RISC-V 采用开放标准，允许任何厂商自由使用和实现。



Overview of RISC-V Architecture



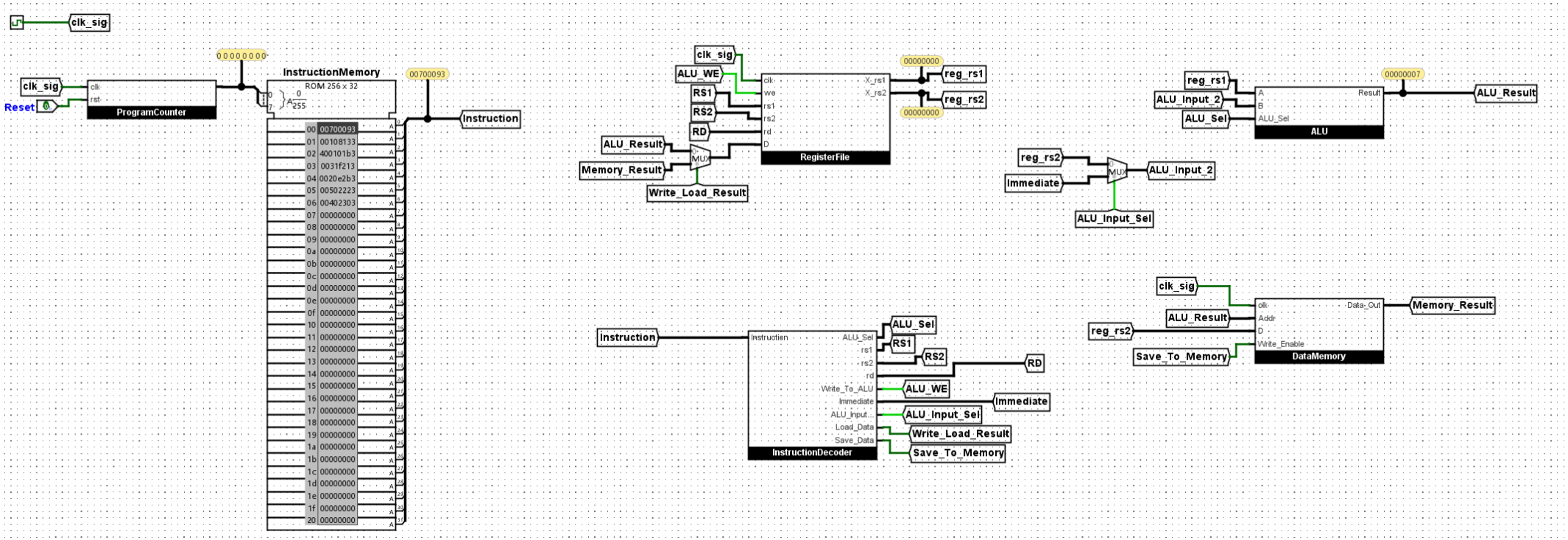
Tasks

Must Part

1. 实现 miniCPU 的 Datapath，包括 RegFile、ALU、PC、IMem、DMem 等。（20%）(在gradescope上有测试点)
2. 实现 miniCPU 的 Controller，包括指令译码、控制信号生成等，并与 Datapath 部分连接，实现指令执行。（20%）
3. 使用 Python 编写一个简单的汇编器，将 RISC-V 汇编代码转换为机器码。（20%）
4. 手动编写汇编程序，完成指定功能，加载到 miniCPU 的指令存储器中，观察 miniCPU 的运行情况。（15%）
5. 实验报告。（25%）

Tasks

Must Part (Result Example)



Bonus

1. 为单周期 miniCPU 添加流水线。

> ☞ 添加流水线之后，你可能会遇到被称之为“冒险”（Hazard）的问题，你需要设计合适的解决方案来处理这些冒险问题。

2. 为 miniCPU 增加更多指令的支持。

> ☞ 我觉得自己特别厉害！那么可以尝试以下 Bonus

3. 为 miniCPU 添加向量运算单元，并支持一些向量指令的支持。
4. 为 miniCPU 支持标量-向量双发射。
5. 使用 Spinal HDL 完成 miniCPU 的设计与实现！ （**Recommended**）
   - 如果你决定使用 Spinal HDL 来实现 miniCPU，你可以不实现 Logisim 版本。请务必提前联系助教！完成相同的任务，使用 Spinal HDL 实现的项目将获得额外的加分。

Grading Policy

About AI?

- 在我们的 Project 中，我们**允许任何 AI 工具的使用**。但是，请**务必确保你理解你的 AI 做了什么**。如果在最终的 Check 中你的回答是“AI 这么写的，我也不太清楚”，那么我们可能会认为你并没有真正理解你的实现，从而影响你的评分。
- 请同学们在合作中遵守基本的学术诚信原则。**组间的抄袭**是严格禁止的。
- 允许参考公开的资源实现，但请按照自己的理解进行调整和修改。如果你参考了公开的资源（例如 GitHub 上的开源项目），请务必在报告中注明引用来源。非复制粘贴的参考不会影响你的评分。

- 即使存在 Bug，如果你能准确描述 Bug 的原因，并展示你为解决 Bug 所做的努力，我们也会给予一定的分数。实现固然重要，但我们更看重你在实现过程中所展现的理解和思考！

Grading Policy for Bonus Tasks

Bonus

- Bonus 部分的分数将在基础分数的基础上进行加分。
- 这部分不会有 OJ 测试，但我们会在面对面 Check 的时候，检查你们的实现。
  - 你们需要自己设计合适的测试用例，以证明你们的实现是正确的。否则，我们可能会认为你们的实现是不完整的，从而你们的加分也会受到影响。
- 和 Must 一样，即使你们的实现不完整，如果在报告中展现了你们的尝试和思考，并做出有意义的尝试和分析，我们也会给予一定的加分。

Who should choose our project?

- EE方向对数字集成电路方向感兴趣，
  - 渴望学习CPU相关知识但是又不想学那门workload很大的CS课——计算机体系结构
  - 想加入我们课题组但是不了解我们具体在干什么
- CS方向但是对数集或者架构与系统方向感兴趣，想预习大二必修课——计算机体系结构
- 其他专业但是对架构与系统和对数字集成电路方向感兴趣或者考虑转专业