|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | 硬件课程设计报告 | | C++实现RISC-V模拟器 |  |  | | --- | | **组长：薛皓天 210340170**  **组员：薛傲翔 210340169**  **杨馥银 210340172** | |

目 录

[一、设计目的 3](#_Toc177911290)

# 一、设计目的

RISC-V是一种开源的指令集架构（ISA），具有高度的灵活性和可扩展性。RISC-V规范定义了基本整数指令集（RV32I/RV64I，分别对应32位和64位），并允许通过可选扩展来添加额外的功能，如浮点运算（RV32F/RV64F）、向量运算（RVV）、原子操作（RVA）、压缩指令集（RVC）等。这种模块化设计使得RISC-V能够根据不同的应用需求进行定制，从而优化性能、功耗和成本。

C++作为一种强大的编程语言，提供了丰富的编程特性和库支持，使得我们可以根据需要灵活地实现RISC-V模拟器的各种功能。此外，由于RISC-V架构的模块化设计，模拟器可以很容易地进行定制和扩展，以适应不同的应用场景和需求。

本次硬件课程设计，我们小组使用C++实现RISC-V模拟器目的在于：

1、通过亲手编写模拟器，可以更深入地理解RISC-V指令集架构的内部工作机制，包括指令的执行流程、寄存器的使用、内存访问等关键概念。这将有助于我们在未来的学习和工作中更好地应用RISC-V架构。

2、编写RISC-V模拟器是一个将计算机体系结构理论知识付诸实践的好机会。我们将能够应用所学到的关于处理器设计、指令集、流水线、缓存、存储层次结构等概念，通过编程实现这些概念在模拟器中的具体表现。

3、在编写模拟器的过程中，你们可以尝试实现不同的RISC-V扩展指令集，如浮点运算、向量运算等，以探索这些扩展如何影响处理器的性能和功能。这将有助于你们更深入地理解RISC-V架构的灵活性和可扩展性，并为未来的研究和开发提供经验。

# 二、设计过程

（设计过程说明，说明设计思路，模型机位数选定、模型机结构示意图等等……）

2.1指令集设计

（指令格式、指令集、修正后的指令集）

2.2 模块设计

2.2.1寄存器堆

（寄存器及寄存器堆的设计图、程序代码，仿真结果……）

2.2.2 运算器模块

（2个DR+ALU的设计图、程序代码，仿真结果……）

2.2.3 存储器模块

（AR+RAM的设计图、程序代码，仿真结果……）

2.2.4 其他模块

（PC、TIMER的设计图、程序代码，仿真结果……）

2.2.4 控制器模块

（微程序控制：rom相关部分：流程图、微指令格式表，.mif文件；跳转控制相关部分：程序/电路；整体：仿真结果）

（硬布线控制：流程图、微操作表、控制程序；整体仿真结果）

（状态机控制：流程图、状态转移图、状态控制信号表、控制程序；整体仿真结果）

2.3整机调试

2.3.1测试程序设计

（根据指令集设计程序，格式参考教材P90的表6-7，给出类汇编代码及内存中保存的16进制代码ram.MIF截图）

2.3.2 仿真结果

（可通过标注、示意等方法，显示各条指令及其结果）

2.3.3 下载后执行结果

（模式选择+管脚绑定截图+输出结果说明）

三、实验总结及心得体会

学生评价-问卷调查（学生填写）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | 完全同意 | 同意 | 基本同意 | 不同意 | 完全不同意 |
| 课程目标1 | 培养学生系统设计能力，使学生掌握基本计算机的设计能力，重点掌握CPU的设计。培养学生独立完成计算机各个模块的设计，掌握各模块间的关系及作用。培养学生对数据流、地址流、控制流的设计能力，总线形式的选择及设计能力。 |  |  |  |  |  |
| 课程目标2 | 培养学生各类工程设计能力，包括逻辑电路设计能力、硬件描述语言的程序设计能力，以及电路图与程序语言配合的设计方法等。在QuartusⅡ环境下，完成模型机的逻辑设计，该模型机需符合RISC机型的特点并能够正常工作。 |  |  |  |  |  |
| 课程目标3 | 培养学生掌握多种工程测试手段，包括仿真指令测试、仿真程序测试、以及实验仪上脱机程序测试的方法，进行整机调试，以验证模型机设计的完整性。培养学生独立思考、查找资料、设计、修改、调试的能力；不设上限的开放性设计目标，促使学生不断完善和提高。 |  |  |  |  |  |