

2019-2020 年秋计算机系统（1）大作业说明

李嘉森 李照宇

2019 年 9 月

1 介绍

- 1.1 本次体系结构作业为用 Verilog 硬件描述语言实现 RISC-V ISA。作业用到了数字电路与逻辑设计的部分知识,在具体实现上需要使用 Verilog 硬件描述语言编写程序并分别进行软件模拟（建议使用 Vivado）和在 FPGA 硬件上进行烧录及测试。建议在开始作业之前学习相关内容。
- 1.2 本次大作业有 bonus, 基础分数要求仅为通过所有测试即可, bonus 包含流水、上板、cache、分支预测、TLB、其他框架 (Tomasulo) 等等。
- 1.3 有可能有些 bonus 我们没有意识到, 评分的时候请提醒我们。

2 RISC-V ISA

2.1 简介

- 2.1.1 本次作业主要目标是使用 Verilog 完成一个能实现 32 位 RISC-V 指令集整数部分用户级别实模式的 CPU。RISC-V 是 UC Berkeley 设计的开源指令集架构, 在设计上属于精简指令集。

2.2 内容

2.2.1 架构

- 2.2.1.1 为了方便调试 CPU 及运行程序, 本次作业采用在板上实现内存, 利用 UART 接口进行数据传输及调试的方式来支持 CPU 的运行。整个顶层架构由 CPU, 通信接口模块, 输入输出接口, 板上内存, 以及内存总线等模块构成。

2.2.2 行为

- a. 由于没有特权指令和虚拟内存的实现, CPU 将不能支持操作系统的运行。因此, 一次运行只能执行内存中固定位置的单一程序。计算机主机通过 UART 连接 FPGA 并上传已编译的二进制程序。FPGA 的通信模块接收数据并存放到内存中。如程序需要读取输入, 则将输入文件上传到 FPGA 的输入缓冲区暂存。
- b. 主机端通过向 FPGA 的通信模块发送指令以控制及调试 CPU。(详见课程仓库 README 及源代码)
 1. 运行指令: 将 CPU 的 rdy_in 信号置为 1, 表示 CPU 应该开始运行。CPU 将从内存的 0x00000000 位置开始读取并执行内存中的指令。
 2. 暂停指令: 将 CPU 的 rdy_in 信号置为 0, 暂停 CPU 运行。此时可以读写内存, 以及查看 CPU 的状态等。
 3. 可以自行添加指令以便调试 CPU (读写寄存器, 设置断点等)。
- c. 内存的有效地址为 0x00000 至 0x20000 (17 位), 大小为 128Kb。0x30000 起 (第 18 位为 1) 的地址被映射为输入输出模块的端口。端口的行为定义如下。
 1. 读取 0x30000 (字节): 弹出并返回输入缓冲区的一个字节。
 2. 写入 0x30000 (字节): 通信模块向主机端发送该字节 (0x00 被忽略)。

3. 读取 0x30004 (32 位无符号整数): 返回从开始运行到当前为止 CPU 经过的周期数 (可能溢出)。
4. 写入 0x30004 (字节): 通信模块向主机发送 0x00 以标志程序结束。
5. 可自行添加端口以支持自定义特性。

2.3 任务

- 2.3.1 实现任意结构的能在模拟下正常运行给定测试程序的 CPU 模块。
- 2.3.2 在 FPGA 上实现 CPU 并正确运行测试程序 (选做)。
- 2.3.3 写一份报告, 简述一下自己的设计, 创新之处及遇到的问题等。(不超过两页)

2.4 评分

- 2.4.1 如提交能通过正确性测试的 bitstream 文件, 将进行测试并按照 FPGA 上的实际运行表现适当加分。模拟正确性演示给助教看, 有适当加分。
- 2.4.2 FPGA 测试将基于仓库给定的顶层架构 (移除 CPU 端所有调试端口) 以及仓库给定的主机端程序 (无调试操作) 进行。将测试 CPU 在 FPGA 上的实际表现 (运行时间及周期数为主要指标)。测试中运行的程序与仓库中的类似。
- 2.4.3 如不能提交 bitstream, 根据 Code Review 时演示模拟正确性适当加分。
- 2.4.4 无流水线给基础分, 实现多级流水线, Tomasulo, Scoreboarding 或自己的架构等视表现适当加分。
- 2.4.5 实现额外的特性视实用性和趣味性适当加分。如果实现了乱序执行并且能够运行操作系统, 可以直接满分, 并且可以作为特邀嘉宾参与 code review 影响给 bonus 的情况。
- 2.4.6 Code Review 中阐述自己的架构特点和创新点均适当加分。

2.5 中期检查

- 2.5.1 暂定你们期中考试后。