2018-2019 年秋计算机系统（1）作业说明

林耘丰

2018 年 9 月

1 介绍

本次体系结构作业一共有两个，均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 或 iverilog）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。

2 作业一：加法器

2.1 介绍

加法器是进行加法操作的电路,它常用于各类计算机处理器的算术逻辑单元(ALU),以及用于处理器的其他单元，进行地址计算、下标计算等操作。

1. 用 Verilog 硬件描述语言编写一个加法器，进行模拟测试，以及在 FPGA 硬件上测试（可选）。加法器模板及用于模拟测试的源文件均位于仓库的“adder”文件夹下。
2. 回答问题。

2.2 内容

1. 仅使用门电路，实现一个涟波进位 16 位无符号数加法器（Ripple-carry adder）并测试。
2. 仅使用门电路，实现一个超前进位 16 位无符号数加法器（Carry-lookahead adder）并测试。
3. 回答问题：为什么超前进位加法器变快了？它到底优化了什么？
4. * 在机器学习等高性能计算场景下，CPU 已经不够用了。目前常用的硬件中竞争最激烈的是 FPGA 与 ASIC。了解它们的概念并说说你认为这两者哪种更适合此类场景，或者用什么样的硬件能做得更好。

2.3 提交

1. 将小问题的答案保存为名为“answer”的 PDF 或纯文本文件
2. 同第一问的 adder.v, adder2.v 打包为压缩文件 [System2018] 学号_姓名_task1.zip
3. 邮件标题命名为 [System2018] 学号_姓名_task1，发送至 linyunfeng@sjtu.edu.cn
4. 提交截止日期：2018 年 11 月 1 日（暂定）

2.4 评分

1. 分数取决于两种加法器能否通过软件模拟测试。
2. 在 FPGA 上的测试不计入分数。
3. 回答问题不计入分数，但是不回答可能会扣分。
4. 带星号的问题不回答也不扣分。

3 作业二：RISC-V ISA

3.1 介绍

作业二是本学期的大作业，主要目标是使用 Verilog（或其它 HDL）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。

3.2 内容

3.2.1 架构

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。

3.2.2 行为

1. 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。

2. 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）

   (a) 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。

   (b) 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。

   (c) 可以自行添加指令以便调试 CPU（读写寄存器，设置断点等）。

3. 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。

   (a) 读取 0x30000（字节）：弹出并返回输入缓冲区的一个字节。

   (b) 写入 0x30000（字节）：通信模块向主机端发送该字节（0x00 被忽略）。

   (c) 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）。

   (d) 写入 0x30004（字节）：通信模块向主机发送 0x00 以标志程序结束。

   (e) 可自行添加端口以支持自定义特性。

3.3 任务

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. 在 FPGA 上实现 CPU 并正确运行测试程序。

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）

3.4 提交

（暂定）

1. 提交作业时，请在附件中包含 CPU 部分的全部代码，可用于烧写的 bitstream 文件以及报告（PDF 格式）。

2. 如不能提交 bitstream 文件，则提交代码并在 Code Review 上演示模拟正确性。

3. 如不能进行正确模拟，则提交代码并在 Code Review 上说明作业完成情况。

4. 将需要提交的材料打包为压缩文件 [System2018] 学号_姓名_task2.zip

5. 邮件标题命名为 [System2018] 学号_姓名_task2，发送至 linyunfeng@sjtu.edu.cn

6. 作业提交截止日期：19 年 1 月初

7. Code Review 日期: 19 年 1 月初

3.5 评分

（暂定）

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。

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。

3. 如不能提交 bitstream，根据 Code Review 时演示模拟正确性适当加分。

4. 无流水线给分，实现多级流水线，Tomasulo，Scoreboarding 或自己的架构等视表现适当加分。

5. 实现额外的特性视实用性和趣味性适当加分。

6. Code Review 中阐述自己的架构特点和创新点均适当加分。

3.6 补充

1. 需要实现的指令包括文档中 RV32I 2.1-2.7 提到的所有指令（部分指令如 FENCE 可能只有当实现了乱序执行时才会有实际作用）。

2. 内存的行为类似同步 SRAM。地址总线宽度为 32 位（仅低 17 位有效）。数据总线的宽度为 8 位。读取需要 2 周期（发出请求后下一周期取数据），写入需要 1 周期（发出请求后不等待）。可以在 CPU 模块内实现一个简单的内存控制器以访问内存。

3. 如需改变 CPU 运行频率，可以利用 Vivado 提供的锁相环 IP 核来输出自定频率的时钟信号。注意保证修改顶层文件里的 SYS_CLK_FREQ 参数等于时钟频率。

4. 0x30000 起的内存地址为 IO 端口，不应该被缓存。

5. CPU 以外的模块在调试阶段可以任意修改，在测试时以仓库最新版本为准（部分参数可自定）。

6. 模拟暂不支持带输入的测试程序，可使用无输入的程序或自己编写测试程序。

7. 根据班级最终完成的情况决定根据 Code Review 或由 FPGA 上的实际表现给分。