

模块化与层次化设计

概述

在本实验，同学们需要将 P3 中使用 Logisim 搭建的 MIPS 单周期 CPU 电路转用 Verilog HDL 进行描述，并在其基础上支持更多指令。同时，本实验也是首个跨文件、多模块的 Verilog 工程化项目，我们会给予一定的设计指导，并且设置一些思考题，来帮助大家更快地学会较为复杂工程的开发方法。

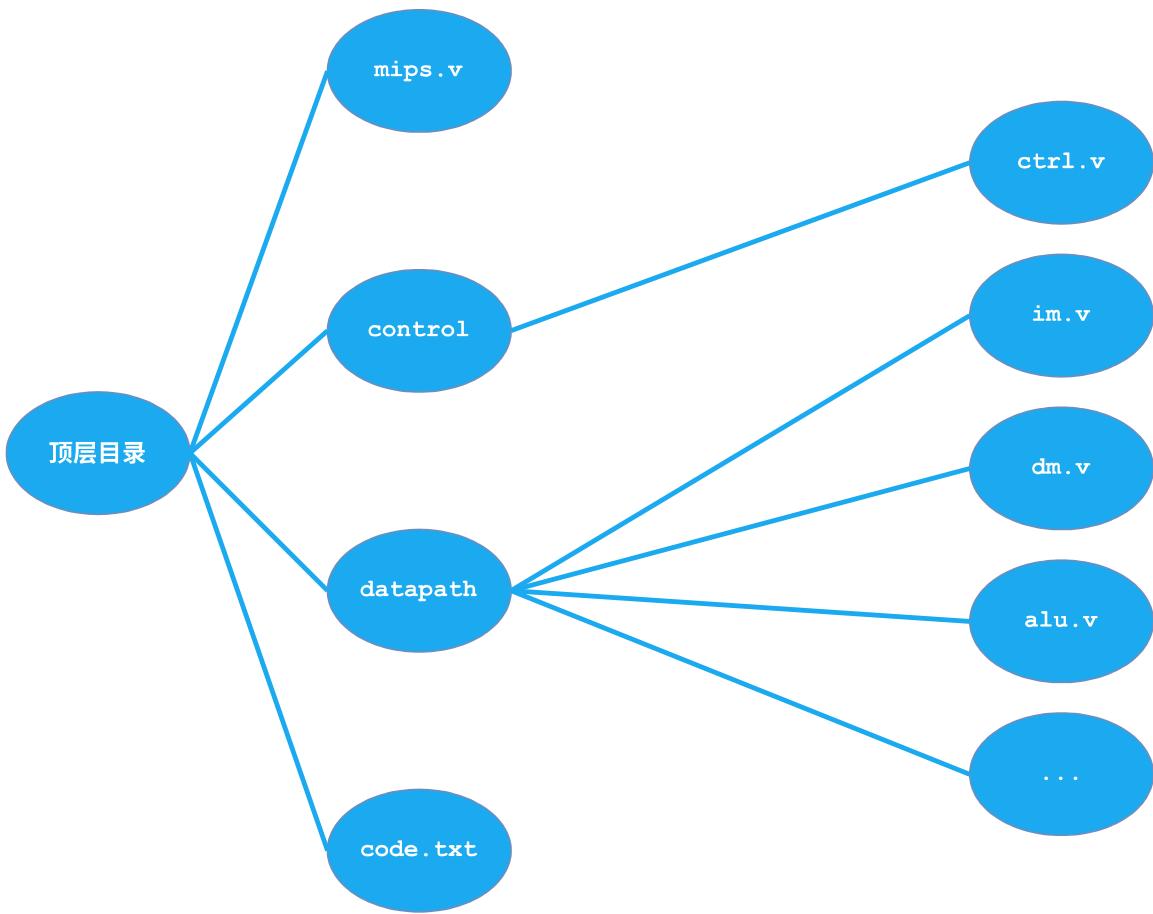
复杂数字逻辑电路和系统的**层次化、结构化**设计隐含着对系统硬件设计方案的逐次分解。虽然现在设计的电路并不复杂，但是我们希望大家能通过这种设计方法的实践学到更多东西，同时，这种思想方法对于整个实验的完成也是非常有帮助的。

在这个过程中，我们也希望同学们可以不断完善自己的 CPU 设计草稿，它是进行电路搭建的指南。**值得一提的是，我们将会在课下以及课上测试环节对文档中的内容（设计文档、思考题）加以检查，请做好准备！**

基本思路

在 P3，你已经使用 Logisim 工具设计出了 MIPS 单周期处理器的电路图，这对于 P4 的完成是非常有帮助的，你要做的只是完成从 Logisim 电路到 Verilog 代码的映射。

首先，参考在 P3 设计的 MIPS 单周期处理器的电路图，Verilog 设计的 MIPS 单周期处理器可基本归结为下图所示：(图中的目录结构和文件命名仅供参考)



整体设计已经完成，接下来就是具体的模块化与层次化设计：

- 单周期处理器包括顶层模块、控制器和数据通路，将其放入顶层目录下。code.txt 中存储相应的机器码。
- 控制器模块占一个独立的 Verilog HDL 文件，实现控制器这个单一的职责，降低模块间的耦合度。
- 顶层目录中的每个 module 都由一个独立的 Verilog HDL 文件组成。在增加相应功能或者修改代码时，减小了对其他模块的影响和修改，条例更加清晰，且对后续的设计也是非常有用和方便的。

这样设计的好处是整个工程具有极好的**局部性**（包括你所理解的**模块化**等等）。记住模块化的要领不是从模块的大小出发，而是从**模块功能的独立性**出发。

设计说明

- 处理器为 32 位单周期处理器，不考虑延迟槽，应支持的指令集为： `add, sub, ori, lw, sw, beq, lui, jal, jr, nop`，其中：
 - `nop` 为空指令，机器码 `0x00000000`，不进行任何有效行为（修改寄存器等）。
 - `add, sub` 按无符号加减法处理（不考虑溢出）。

- 需要采用**模块化**和**层次化**设计。顶层文件为 **mips.v**，有效的驱动信号要求包括且仅包括**同步复位信号 reset 和时钟信号 clk**，接口定义如下：

```
module mips(  
    input clk,  
    input reset  
);
```

- 推荐同学们在设计 CPU 前请**通读本章所有内容**，以避免无谓的修改；在设计 CPU 时参考之后各节给出的要求与建议，形成 CPU 设计草稿，并回答思考题。请注意：
 - 设计草稿与思考题为必做内容，请合并为 CPU 设计文档提交。
 - 设计草稿形式与篇幅不限，电子版可采用文本、截屏等形式，纸质版可直接拍照。
 - 最后需要提交的内容为：**CPU 设计文档、包含全部 Verilog 文件的 zip 格式压缩包**。
- 通过课下测试**并不意味着你的设计完全不存在问题**，建议自行构造测试用例以验证设计的正确性（如果进行了相关工作，请写入设计文档，课上问答时将酌情加分）。