

## 实验三：单周期 RISC-V 处理器实现

RISC-V 架构 CPU 的流程可分为取指、译码、执行、访存、回写 (Instruction Fetch, Instruction Decode, Execution, Memory Request, Write Back), 五阶段。

### 1 实验目的

- i. 进一步熟悉 RISC-V 指令集;
- ii. 掌握不同类型指令在数据通路中的执行路径;
- iii. 理解单周期 CPU 执行指令的过程;
- iv. 掌握单周期处理中执行阶段的划分;
- v. 理解模块化设计思想;

### 2 实验环境 (推荐)

与实验一相同:

- i. IDE: vscode
- ii. verilog compiler: iverilog
- iii. waveform viewer: GTKwave

### 3 实验任务

本实验要求编写一个单周期 RISC-V 处理器。

#### 3.1 实验描述

图1给出了单周期 RISC-V 处理器设计图。图中按照 CPU 的五阶段执行模式, 将 RISC-V 处理器分为 5 个 Module, 分别完成取指、译码、执行、访存、回写工作, 其中比较复杂的模块是译码 (lab2 已实现) 和执行模块。

#### 3.2 实验要求

图1中的一部分模块已经在前面的 Lab 中实现完毕, 但在组建 CPU 时需要做一定的修改, 需要修改与额外实现的模块描述如下:

1. PC: 位于取指阶段, 为 D 触发器结构的 PC 寄存器。需要在 Lab2 实现的基础上增加两个输入, 分别为 pc\_execute 模块计算并传递的"pc\_src", "new\_pc", PC 寄存器根据"pc\_src" 在"new\_pc" 和"pc\_plus4" 中间选择数据更新。

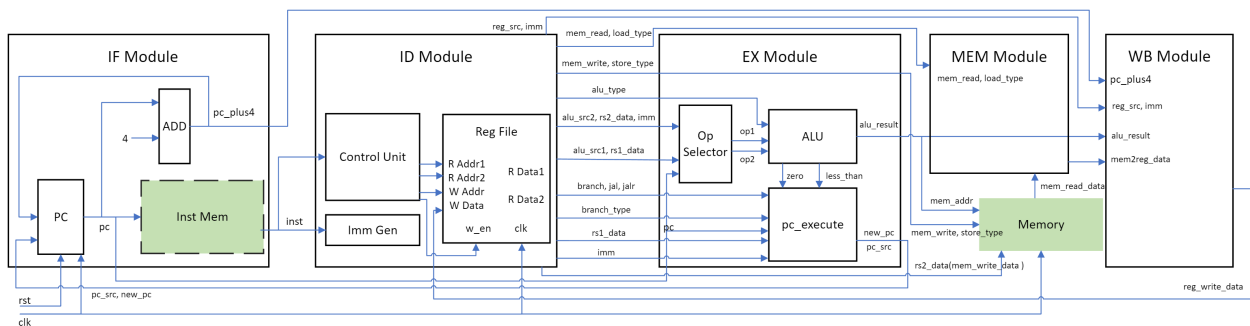


图 1: 5-stages RISC-V CPU

- Op Selector: 位于执行阶段, 需要根据 ID 模块的控制信号选择数据, 生成用于 ALU 计算的两个操作数。需要实现六个输入, 两个输出, 输入均来自 ID 模块, 包括"alu\_src1"、"pc"、"rs1\_data"、"alu\_src2"、"imm"、"rs2\_data"。根据"alu\_src1/2", 生成输出"op1/2" 传递给 ALU 模块。
  - ALU: 需要实现三个输入, 三个输出。根据信号"alu\_type", 对 Op Selector 生成的两个操作数进行相应运算或比较并生成结果信号、"alu\_result" (传递给 WB 模块) "zero" 和 "less\_than" (传递给 pc\_execute 模块)。
- Tips: 传递给 pc\_execute 模块的信号为用于判断跳转的控制信号。*
- pc\_execute: 需要实现如图1所示的九个输入, 两个输出, 根据各种控制信号和 ALU 的计算结果生成用于选择新 pc 的"pc\_src" 和 "new\_pc"。
  - MEM Module: 需要实现三个输入和一个输出。接收 id 模块的"mem\_read" 信号、"load\_type" 信号, 根据前者决定是否产生内存读操作, 根据后者决定对从内存读入数据"mem\_read\_data" 字段选取, 并将需要写入寄存器的结果"mem2reg\_data" 传递给 WB 模块。
  - WB Module: 需要实现五个输入和一个输出。根据输入的"reg\_src" 信号, 从"pc\_plus4"、"imm"、"alu\_result"、"mem2reg\_data" 中选择一个生成写回 Reg File 的数据"reg\_write\_data"。

### 3.3 模块组织

各个模块的引用关系如表1所示。为了工程的美观性和可读性, 我们建议大家不要把所有 Verilog 文件都放到项目文件夹下, 最好按模块的包含层次把文件组织到子文件夹中。表1提供了一种文件夹组织方案, 供大家参考。(我们不强制大家这样组织, 大家可以按照自己的习惯组织自己的工程)

### 3.4 信号设计

注: 我们只会测试 riscv 核的整体行为, 不会测试具体某个阶段输出的信号。因此可以自行设计所需信号, 表2仅供参考。

## 4 测试与评分

请在本地调试完毕后, 再将代码提交到测试平台, 并查看测试平台中显示的分数。

riscv_top.v	顶层模块, 连接 riscv 核和存储器 (已经给出, 请勿更改)
—inst_mem.v	指令内存 (已经给出, 请勿更改)
—mem.v	内存 (存储数据) (已经给出, 请勿更改)
—riscv.v	riscv 核
—five_stages/if.v	取指模块
—dp_components/pc_reg.v	pc 模块
—dp_components/add.v	加法器模块
—five_stages/id.v	译码模块
—cp_components/control_unit.v	控制逻辑生成
—cp_components/alu_control.v	alu 运算符选择信号生成
—dp_components/imm_gen.v	立即数生成
—dp_components/reg_file.v	register file
—five_stages/ex.v	执行模块
—dp_components/op_selector.v	根据控制信号生成用于 ALU 计算的两个操作数
—dp_components/alu.v	运算单元, 根据信号选择运算类别
—dp_components/pc_execute.v	根据控制信号生成跳转使能信号和跳转的目标地址
—five_stages/mem.v	访存模块
—five_stages/wb.v	写回模块

表 1: module organization

some signal	meaning	some signal	meaning
write_data	32bit, 寄存器写入数据	inst	32bit, 指令内容
rs1/2_data	32bit, rs1/2 数据	imm	32bit, 立即数
branch	是否为 branch 指令	alu_src1/2	1bit, alu 操作数来源
jal	是否为 jal 指令	alu_type	4bit, 运算符选择
jalr	是否为 jalr 指令	reg_src	2bit, 写寄存器的数据来源
mem_read	是否进行内存读	branch_type	3bit, branch 判断方式
store_type	3bit, store 的方式	load_type	3bit, load 的方式
mem_write	是否进行内存写		
<i>some Lab3 new signals</i>			
op1/2	32bit, ALU 操作数	mem_write_data	32bit, 写内存数据
zero	用于跳转的零标志	mem_addr	32bit, 读写内存地址
less_than	用于跳转的小于标志	mem_read_data	32bit, 读内存数据
new_pc	跳转的目标 pc	mem2reg_data	32bit, 要写回寄存器的读内存数据
pc_src	pc 的来源	alu_result	32bit, alu 运算结果
pc_plus4	32bit, pc+4 的值	reg_write_data	32bit, 写回寄存器的数据

表 2: signal meanings

## 4.1 tb 验证思路

由于本次 lab 以取指译码为目标，并不涉及完整的指令执行，因此 tb 文件主要考察以下两方面（详见 tb 文件）。

1. 能分别正确完成各类型指令功能（test\_code0-6）
2. 各类型指令配合能正确运行一个程序（test\_code7-8）

## 4.2 平台使用

1. 课程平台地址：[微处理器设计与智能芯片课程实践](#)
2. 请按规定字段进行注册，并牢记自己的密码。
3. 从测试平台上下载的代码包包括模板文件和简化的测试文件，请先仔细阅读 README，其中提到了你至少应该上传哪些文件，可以自行定义其他辅助模块一并压缩上传。
4. 测试平台会用自己的测试用例覆盖提供给同学们的用例，可以不再上传与测试相关的文件。
5. [请珍惜实验室服务器资源，切勿恶意提交代码。](#)
6. 与平台、实验有关的任何问题可以与助教联系。
7. 欢迎大家提出有关平台、实验的各种建议，让本课程和配套实习越来越好。

## 4.3 评分规则

本次测试共给出了 9 类 test\_code，每一类对应 11 分，通过任意一项 test 可以获得一个绿色的 PASS，9 类 test 都通过即可获得 100 分。

[注：平台评分主要作为对你完成度的提示，并不等于你最终课程实习部分的分数。](#)

## 4.4 实验报告

在实验的同时和实验之后撰写实验报告是非常良好的习惯。这不仅有助于提高实验效率，理清实验思路，帮助 debug，也有助于记录实验过程和结果，还可以和其他同学交流和分享。因此，我们需要大家在 lab 完成过后为我们提交一个实验报告，实验报告有以下几点注意事项：

1. 包含你的设计思路，可以包括相应的模块组织层次、一些不同于助教提示的地方等。
2. 你 debug 的过程，或崩溃或大起大落或柳暗花明的心路历程等。
3. [最终用于 Lab 评分的那次提交：包括代码包、分数截图、提交时平台为本次提交生成的 ID。](#)
4. 实验报告不做字数要求，体现思考和实验过程即可（反卷第一名）。即使最后有 bug 没解决，没有以满分通过平台测试，讲一讲你的 debug 过程与思考也会获得一些分数的补偿：)

## 5 提示与帮助

会在课程网站上动态更新，大家遇到问题先去网站的 Q&A 板块看一下。