实验四：五周期流水线 RISC-V 处理器实现

Lab3 已介绍过 RISC-V 架构 CPU 五级流水线划分，分别是取指、译码、执行、访存、回写 (Instruction Fetch, Decode, Execution, Memory Request, Write Back)，五阶段。

单周期 CPU 虽然 CPI 为 1，但由于时钟周期受限于执行延迟最长的指令 (如 lw、sw)，从而难以达到较好的性能。因此，流水线技术被引入。它通过划分阶段的方式，不仅可以实现指令的并行执行，还可以提升硬件的利用率。本次实验中，我们将 Lab3 中实现的的单周期处理器拓展为五阶段流水线处理器，并解决课本提及的冒险 (hazard) 问题。

1 实验目的

i. 掌握流水线 (Pipelined) 处理器的思想；

ii. 了解流水线处理器中的数据通路；

iii. 了解流水线处理器遇到的冒险；

iv. 掌握数据前递 (Data Forwarding)、流水线暂停等冒险解决方式。

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

本实验要求将 Lab3 实现的单周期 RISC-V 处理器通过增加寄存器、增加数据通路等方式，拓展为一个五周期流水线处理器。

3.1 实验描述

本次 lab 对单周期 RISC-V 处理器的改动大致分为三个步骤。

(1) 在五个阶段之间插入寄存器：改为五级流水线后，每一个阶段所需要的控制信号仅为 ID 阶段产生的信号中的一部分。因此，我们需要通过触发器将控制信号逐级传递到对应的阶段。此外，在各阶段产生的数据信号也可能被后续阶段使用，这些数据信号也需要通过流水线寄存器逐级传递。

(2) 增加一条必须的数据通路：对于需要写寄存器的指令，写入寄存器的地址在 ID 阶段生成，写入寄存器的内容在 WB 阶段确定。因此需要在寄存器中存入写入的目标寄存器地址，直到 WB 阶段才将该写回地址与写回数据一起送给 register file。

(3) 冒险的解决：冒险大致分为三类，结构冒险、数据冒险、控制冒险。其中，由于我们单独实现了指令内存，因此与内存访问相关的结构冒险被天然的避免了. 但大家仍需要考虑如何解决同一个周期内**同时读写寄存器的结构冒险**。此外，大家还需要通过实现课件上提到的**数据前递机制**来解决各类数据冒险，并且通过实现课件上提到的**分支前递机制**（将分支预测提前到译码阶段）来解决各类控制冒险。对各种冒险情况的讨论与解决的细节请参考课件。

注：本次实验不要求大家实现课件上提到的分支预测机制。感兴趣的同学可以自行实现。

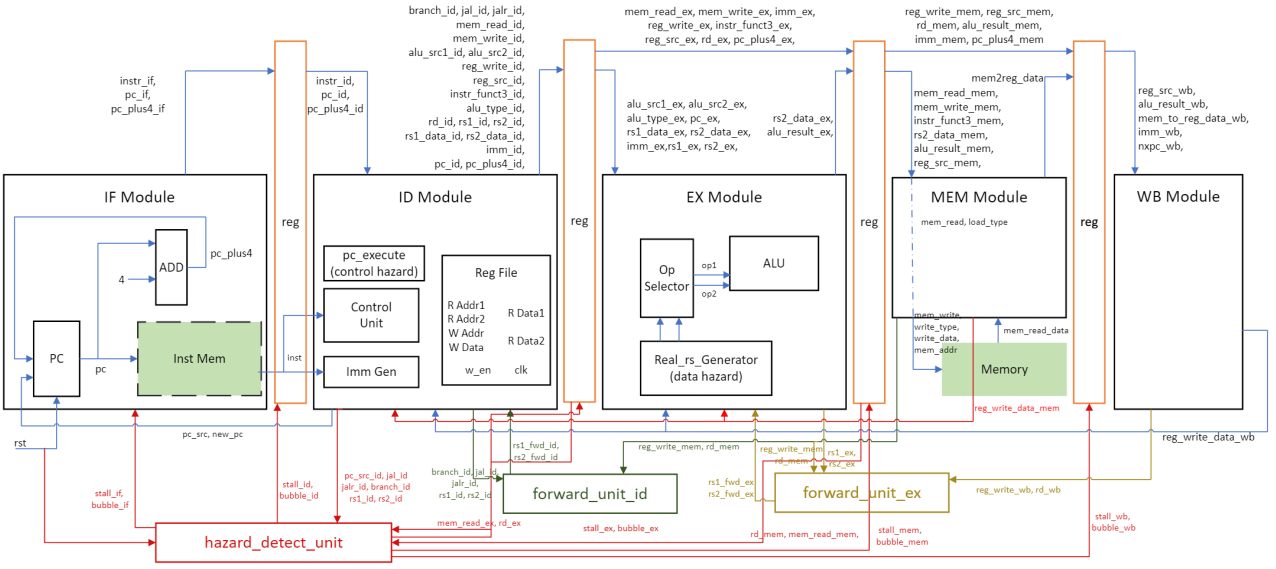

图 1: pipelined RISC-V CPU

3.2 实验要求

为了实现上述功能，本次实验需要在 Lab3 的基础上实现如下模块/功能（模块组织仅供参考，大家可以根据自己的思路设计模块）：

1. reg：流水线寄存器，大家需要在五阶段之间插入四个流水线寄存器，用于传递各种信号。我们在图1中标注出了各阶段可能需要传递的信号，供大家参考。

2. pc_execute（control hazard）：该模块用于提前计算”new_pc”, ”pc_src”，以减少跳转时流水线冲刷带来的损失。注意此处在判断是否跳转时，操作数要考虑到从 MEM 模块前递来的数据。且 pc_execute 模块的前移会给 alu_control 模块带来一些功能上的剪枝。 此模块可以直接实现在 *ID* 内，不强制单独设计模块。

3. Real_rs_Generator（data hazard）：根据选择控制信号”rs1/2_fwd_ex” 在寄存器中读出的数据”rs1/2_data”、MEM 模块前递的数据”reg_write_data_mem” 和 WB 模块前递的数据 ”reg_write_data_wb” 中选择出真正应该被运算的 rs1/2。此模块可以直接实现在 *EX* 内，不强制单独设计模块。

4. forward_unit_id：根据课上所授相关规则，判断是否有必要数据前递。生成用于指导 ID 模块选择正确数据的控制信号”rs1_fwd_id” 和 rs2_fwd_id”。

5. forward_unit_ex：根据课上所授相关规则，判断是否有必要数据前递。生成用于指导 EX 模块选择正确数据的控制信号”rs1_fwd_ex” 和 rs2_fwd_ex”。

6. hazard_detect_unit：检测冒险情况的发生，发出”stall”、”bubble” 的控制信号，控制流水线的暂停与冲刷。注意零寄存器不可被改写的特质导致涉及它的一些冒险不是真正的冒险。

3.3 模块组织

表1给出了各个模块的功能、引用关系和建议的文件夹归类方式。

|riscv_top.v	顶层模块，连接 riscv 核和存储（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）
|—templates/pipe_dffs.v	寄存器模板文件 （已经给出，请勿更改）
|–riscv.v	riscv 核
|—hazard_unit/forward_unit_id.v	id 数据前递控制信号生成
|—hazard_unit/forward_unit_ex.v	ex 数据前递控制信号生成
|—hazard_unit/hazard_detect_unit.v	流水线暂停/冲刷控制信号生成
|—pipe_regs/if_id.v	if/id 寄存器
|—pipe_regs/id_ex.v	id/ex 寄存器
|—pipe_regs/ex_mem.v	ex/mem
|—pipe_regs/mem_wb.v	mem/wb 寄存器
|—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
|—five_stages/mem.v	访存模块
|—five_stages/wb.v	写回模块

表 1: module organization

3.4 信号设计

由于改成流水线，总的信号数目变得非常多，但绝大多数功能和之前一致，只是区分了各自阶段，本节只提一些助教认为需要注意的或和以前不完全相同的信号。

some signal	meaning	function
rs2_data_ex	EX 内数据前递选择后生成的全新 rs2 值	用于后续涉及 rs2 的操作
reg_write_data_mem	在 MEM 阶段能获取的写回寄存器的值	用于数据前递
reg_write_data_wb	在 WB 阶段能获得的写回寄存器的值	用于真正写回与数据前递
rs1/2_**	rs1/2 对应的寄存器标号	用于判断冒险的发生
rs1/2_fwd_**	rs1/2 是否应该选择数据前递来的哪个值	用于数据冒险
stall/bubble	寄存器控制信号	用于暂停/冲刷流水线

表 2: signal meanings

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。

4.1 tb 验证思路

由于流水线处理器与单周期处理器功能上并不会产生区别，只会在效率和控制上有所变化，我们给出的 tb 文件相比 lab3 并没有发生变化。

1. 能分别正确完成各类型指令功能（test_code 0-6）
2. 各类型指令配合能正确运行一个程序（test_code 7-8）

4.2 重要说明

如课件所述，解决流水线冒险可以简单的暂停、冲刷流水线，也可以采用数据前递与分支前递等方式来提升 CPU 的执行效率。平台只做功能性测试，因此请在实验报告里清晰地注明你实现的优化思路，完成如3.1所述的优化的同学获得实验意义上的满分。

4.3 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。

4.4 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 都通过平台则会显示 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。

4.5 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。