实验报告成绩:	成绩评定日期:

2022 ~ 2023 学年秋季学期

《计算机系统》必修课

课程实验报告

班级：人智 2202

组长：邓欣

组员：贾竣棋

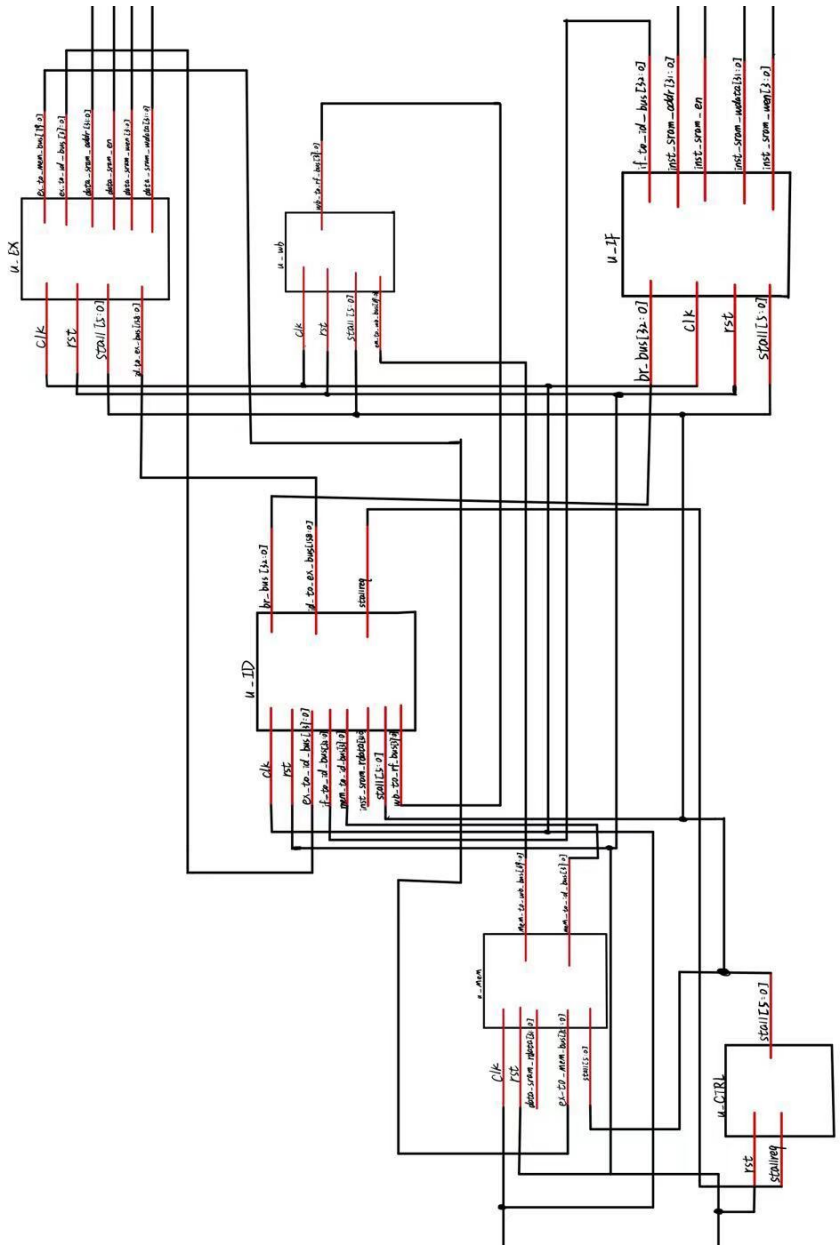
报告日期：2025.1.5

一．工作量划分

姓名	完成任务	工作量
邓欣	完成了第一个 passpoint 的实现，实现了接口和连线的构建。搭建了 github 仓库，完成实验报告的编写。尝试了下面的检测点但是没有成功。	60%
贾竣棋	完成了模块设计图的绘制。参与了代码的编写，完成了部分指令的添加，	40%

二．总体设计

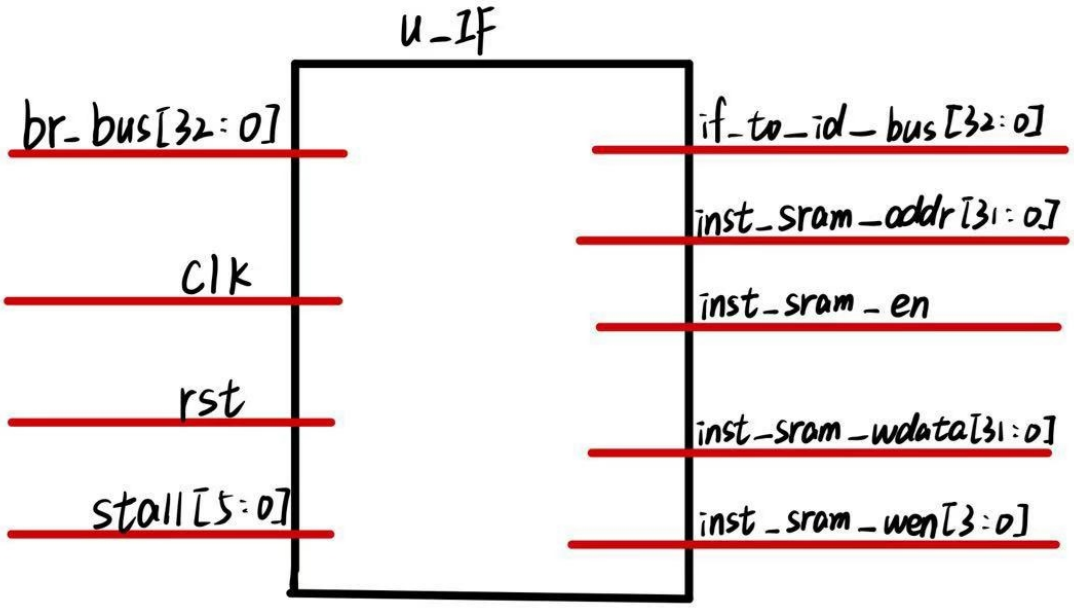
我们的 cpu 包含 IF 段、ID 段、EX 段、MEM 段和 WB 段。具有 32 位寄存器。我们的流水线只能完成实验课要求的第一个测试点，。



三．IF 模块

IF 模块主要用于指令的获取。该模块有四个输入接口 clk、rst、stall 和 br_bus,五个输出接口 if_to_id_bus，inst_sram_en，insr_sram_wen，inst_sram_addr，insr_sram_wdata。首先定义 32 位寄存器 pc_re 和一位寄存器 ce_reg，用于保存当前的 pc 值和控制指令是否获取。接着对二者进行更新，计算下一条指令地址，对指令存储器信号赋值，最后对 ID 阶段传递的信号赋值。

名称	尺寸	方向	功能
Clk	1	输入	时钟信号
rst	1	输入	复位信号
stall	6	输入	暂停信号，是否允许更新 pc
br_bus	33	输入	分支信息总线，是否跳转
If_to_id_bus	33	输出	向 ID 阶段传递的信号，使能信号和 pc 值
Inst_sram_en	1	输出	存储器使能信号，是否进行读取操作
Insr_sram_wen	4	输出	存储器写使能信号，指令获取阶段填充 0
Inst_sram_addr	32	输出	指令存储器的访问地址，用来获取下一条指令
Insr_sram_wdata	32	输出	写入指令存储器的数据，同样填充 0

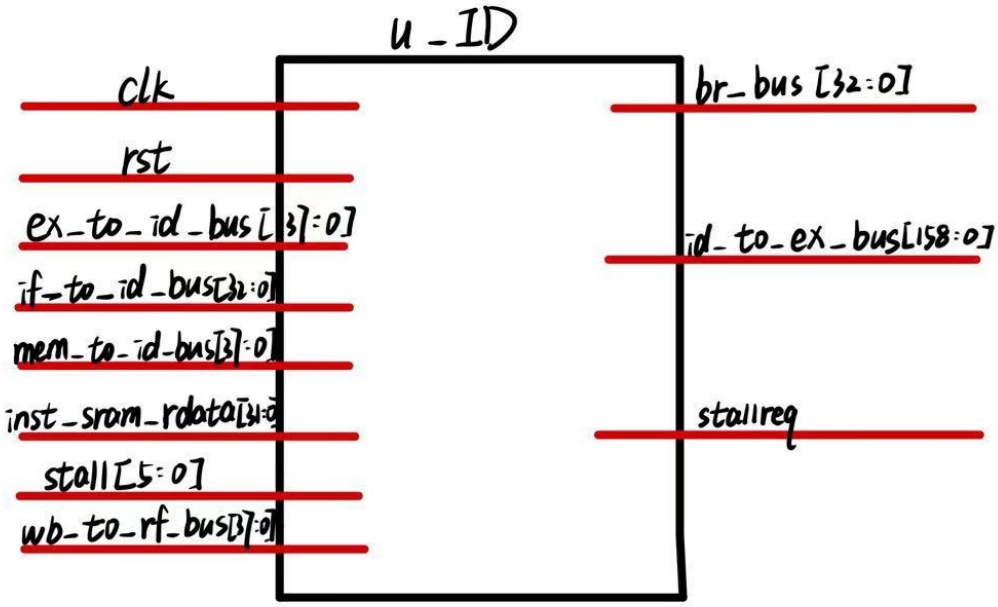


四．ID 模块

ID 阶段的主要功能是对从指令获取阶段传来的指令进行解码，并为后续 EX 段准备所需的操作数和控制信息。在 ID 阶段发生了与 Ex 和 MEM 的数据相关，因此应该添加从 EX 和 MEM 到 ID 段的连线。首先声明各个接口，包括 clk, rst, ex_to_id_bus, if_to_id_bus, mem_to_id_bus, inst_sram_rdata, stall, wb_to_rf_bus, br_bus, id_to_ex_bus, stallreq.对于数据传输，我们仿照 wb 阶段的写回控制，也声明了从 EX 和 MEM 段的写回控制信号。声明两个临时数据 tdata1 和 tdata2，作为最终传递给下游的数据，而不是直接传递 rdata1 和 rdata2.接着通过 assign 语句将从下游阶段传来的数据赋值给前面声明的各个阶段。后面增加了一个数据转发的逻辑判断，如果 rs 寄存器的值已经在 EX、MEM 或 WB 阶段被更新，则从对应阶段的寄存器值中选取最新的值，反之则使用 rdata。这里越靠

前的阶段的寄存器中的值才是最终的值。接着进行寄存器的实例化。最后定义运算指令和 ALU 的对应的操作信号以及控制分支的相关信号。

名称	尺寸	方向	功能
Clk	1	输入	时钟信号
rst	1	输入	复位信号
Ex_to_id_bus	38	输入	Ex 段发给 id 段的数据，用来处理数据相关
If_to_id_bus	33	输入	If 段发给 id 段的数据
Mem_to_id_bus	3	输入	Mem 发给 id 段的数据，用来处理数据相关
Inst_stam_rdata	32	输入	指令地址中存储的值
stall	6	输入	控制暂停信号
Wb_to_rf_bus	38	输入	Wb 段传给 id 的寄存器的内容
Br_bus	33	输出	Id 给 if 的数据，判断下一条指令的地址
Id_to_ex_bus	159	输出	Id 传给 ex 的总线
stallreq	1	输出	从 id 段发出的暂停信号

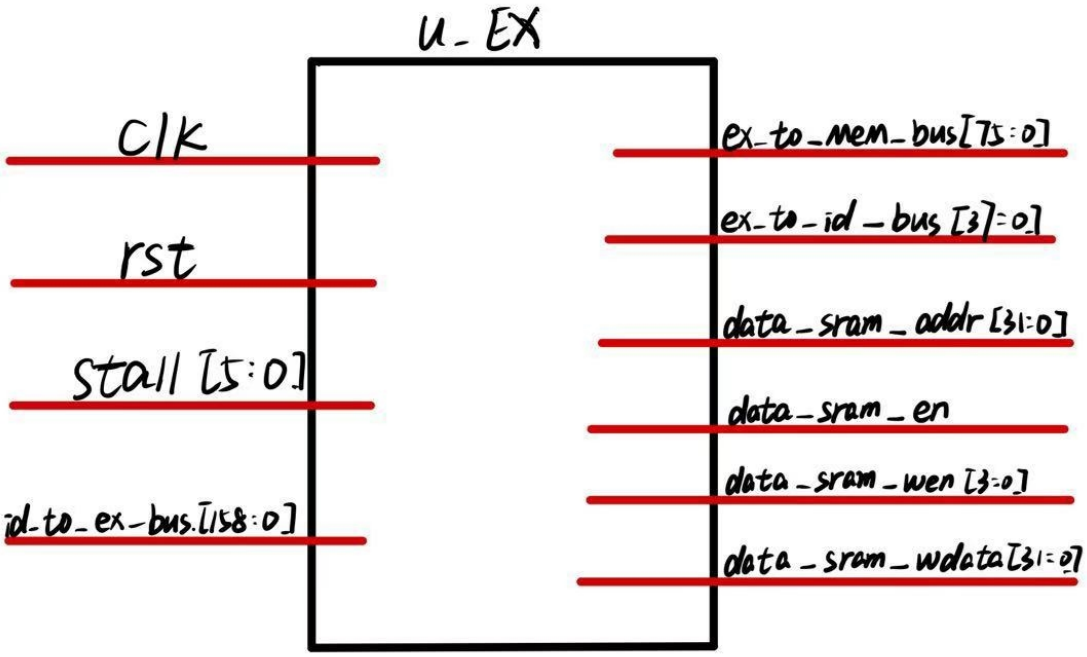


五．EX 模块

EX 模块使用了寄存器 id_to_ex_bus_r 来同步 ID 阶段传来的数据，这样以确保数据在时钟的上升沿被稳定接收和处理，避免组合逻辑中的毛刺。接着将从 id 段传入寄存器总的信号分配给不同的信号，后续用来进行 alu 运算。EX 阶段还会将计算结果、寄存器写使能、寄存器写地址等信息传递回 ID 阶段，用于后续的处理。

名称	尺寸	方向	功能
Clk	1	输入	时钟信号
Rst	1	输入	复位信号
stall	6	输入	暂停信号
Id_to_ex_bus	159	输入	从 id 传到 ex 的数据总线

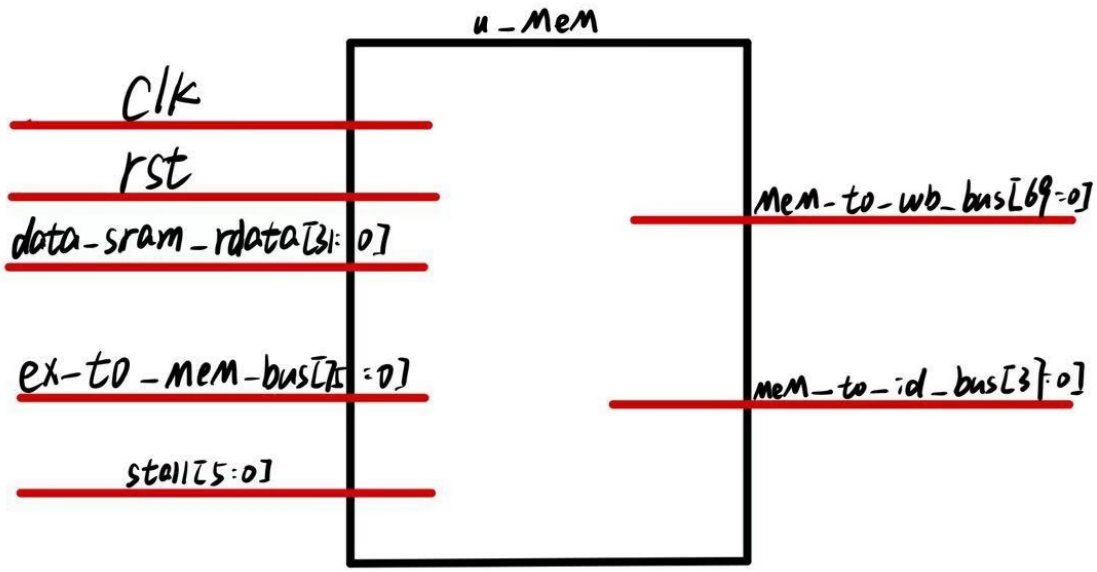
Ex_to_mem_bus	76	输出	从 ex 传到 mem 的数据
Ex_to_id_bus	38	输出	Ex 传到 id 的数据，用于处理数据相关
Data_sram_addr	32	输出	指定访问 SRAM 的地址
Data_sram_en	1	输出	启用或禁用对 SRAM 的访问
Data_sram_wen	4	输出	控制是否允许对 SRAM 进行写操作
Data_sram_wdata	32	输出	传输写入 SRAM 的数据



六．MEM 模块

Mem 模块主要用于内存访问，读取内存中相应地址对应的值。首先声明输入和输出的信号以及一个临时寄存器 ex_to_mem_bus_r.通过一个时序模块根据流水线的流动和暂停改变该寄存器的值，声明 mem_pc 等信号并将该临时寄存器分解给这些信号。最后根据不同的指令类型，选择从内存读取或写入的数据，并进行相应的位扩展，根据控制信号选择内存阶段的结果或执行阶段的结果，并写入寄存器文件

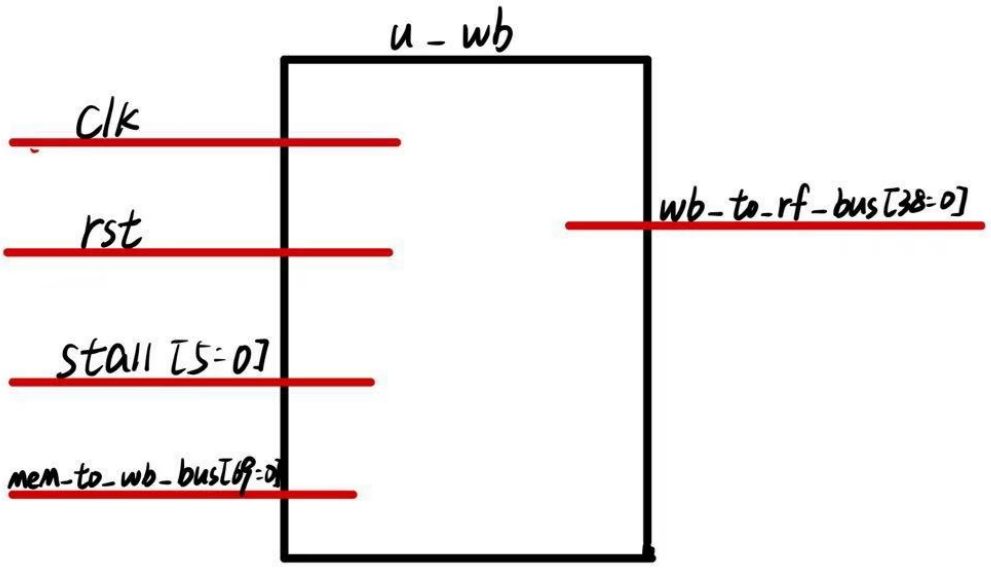
名称	宽度	方向	功能
Clk	1	输入	时钟信号
rst	1	输入	复位信号
stall	6	输入	暂停信号
Ex_to_mem_bus	76	输入	Ex 段传给 MEM 段的信号
Data_sram_rdata	32	输入	内存读取结果，将要写入寄存器的值
Mem_to_wb_bus	70	输出	Mem 传给 wb 的数据总线
Mem_to_id_bus	38	输出	Mem 与 ID 的连线，处理数据相关



七．WB 模块

WB 阶段负责将之前计算或从内存中读取的数据存储到寄存器，确保处理器的寄存器堆中保存了正确的数据供后续指令使用。

名称	宽度	方向	功能
Clk	1	输入	时钟信号
rst	1	输入	复位信号
stall	6	输入	暂停信号
Mem_to_wb_bus	70	输入	Mem 段传给 wb 段的数据
Wb_to_rf_bus	38	输出	Wb 传给寄存器文件的数据



八．实验心得

总体而言，我们组在在这次实验中的表现不佳，仅完成了一个检查点的实现，远远没有达到实验的要求。一方面是因为一开始没有理解实验的目的和要求，以及对 cpu 的结构以及仿真软件的不熟悉，另一方面由于时间上的安排不均匀，导致我们没有按照预定的时间开始着手进行实验。这是一个值得吸取的教训。尽管完成率不高，但是经过一段时间的摸索，我也对 cpu 流水线的结构以及数据流通有了一定的理解，弥补了理论课上的缺失，也让我学到了很多理论课上学不到的东西。在此还要

感谢助教老师在实验期间为我提供的各种帮助。总体而言，这次实验还是让我受益匪浅。