

《计算机组成原理》实验报告

年级、专业、班级	2022级计算机科学与技术03班/01班/06班	姓名	叶旭航,解吴雪,李佳玲
实验题目	实验三简单单周期CPU实验		
实验时间	2024 年 4 月 16 日	实验地点	DS1410
实验成绩	优秀/良好/中等	实验性质	<input type="checkbox"/> 验证性 <input checked="" type="checkbox"/> 设计性 <input type="checkbox"/> 综合性
教师评价: <input type="checkbox"/> 算法/实验过程正确; <input type="checkbox"/> 源程序/实验内容提交; <input type="checkbox"/> 程序结构/实验步骤合理; <input type="checkbox"/> 实验结果正确; <input type="checkbox"/> 语法、语义正确; <input type="checkbox"/> 报告规范; 其他: <div>评价教师: 冯永</div>			
实验目的 (1)掌握不同类型指令在数据通路中的执行路径。 (2)掌握Vivado仿真方式。			

报告完成时间: 2024年 5月 18日

1 实验内容

阅读实验原理实现以下模块：

- (1) Datapath, 其中主要包含alu(实验一已完成), PC(实验二已完成), adder、mux2、signext、sl2(其中adder、mux2数字逻辑课程已实现, signext、sl2参见实验原理),
- (2) Controller(实验二已完成), 其中包含两部分, 分别为main_decoder, alu_decoder。
- (3) 指令存储器inst_mem(Single Port Ram), 数据存储器data_mem(Single Port Ram); 使用Block Memory Generator IP构造指令, 注意考虑PC地址位数统一。(参考实验二)
- (4) 参照实验原理, 将上述模块依指令执行顺序连接。实验给出top文件, 需兼容top文件端口设定。
- (5) 实验给出仿真程序, 最终以仿真输出结果判断是否成功实现要求指令。

2 实验设计

这一节, 主要描述各个模块的功能、接口、逻辑控制方法(状态机控制方法)等。(红字为内容说明, 请删除)

2.1 数据通路

2.1.1 功能描述

简单描述实现的功能即可, 一句话亦可(红字为内容说明, 请删除)

2.1.2 接口定义

接口定义请使用表格, 需要包括接口信号名、方向、宽度、含义(红字为内容说明, 请删除)

表 1: 接口定义模版

信号名	方向	位宽	功能描述
valid	Output	1-bit	If CPU stopped or any exception happens, valid signal is set to 0.

3 实验过程记录

3.1 问题1: 设计 Datapath 模块连接

问题描述: 由给出的参考书得知 Datapath 模块的各个组成部分及其连接关系, 因为 datapath 内的模块大多已给出或前述实验已完成, 所以重点在需要详细分析各部分的功能和数据流向, 确保模块之间的正确连接。

解决方案:

1. PC 模块连接:

- `adderPC_inst` 实例化了一个加法器模块, 用于将当前 PC 地址加上一个偏移量, 得到下一个 PC 地址。
- `mux2 PC_select` 实例化了一个多路选择器模块, 根据 `pcsrc` 信号选择下一个 PC 地址。
- `PCnext_inst` 实例化了一个多路选择器模块, 根据 `jump` 信号选择是否跳转到指定地址。
- `PC_inst` 实例化了 PC 模块, 用于存储并更新 PC 地址。

2. Branch 计算和存储:

- `sign_extend_inst` 实例化了一个符号扩展模块, 将指令中的 16 位符号扩展成 32 位。
- `adderBranch_inst` 实例化了一个加法器模块, 用于计算分支目标地址。

3. 寄存器文件和数据路径选择:

- `regfile_inst` 实例化了寄存器文件模块, 用于读取和写入寄存器数据。
- `mux2 select_dst` 实例化了一个多路选择器模块, 根据 `regdst` 信号选择写入的寄存器地址。

4. ALU 计算和数据选择:

- `select_srcB` 实例化了一个多路选择器模块, 根据 `alusrc` 信号选择 ALU 的第二个操作数。
- `ALU_inst` 实例化了一个 ALU 模块, 用于执行算术逻辑运算。
- `mux2 select_res` 实例化了一个多路选择器模块, 根据 `memtoreg` 信号选择 ALU 的输出或数据存储器的读取数据作为最终结果。

5. 其他连接: 连接各个模块之间的数据和控制信号, 确保正常的的数据流和操作顺序。

3.2 问题2: 两个存储器实现

问题描述: 使用 Block Memory Generator IP 构造指令存储器。

解决方案: 使用 IP 实例化数据存储器和指令存储器, 配置不同的宽度和深度参数。指令存储器需要导入指定的 `coe` 文件以支持所需操作。

3.3 问题3: 端口适配

问题描述: 使其余模块端口适配顶层文件。

解决方案：由于 MIPS 软核顶层文件和设计顶层文件 `top` 均已给出，故修改了底层的各模块的输入输出端口，以适配顶层文件的调用需求。

3.4 问题4: 仿真调试

问题描述：调用 `top` 模块进行仿真。

解决方案：根据已给出的仿真文件对实现的单周期 CPU 进行仿真，通过仿真图和控制台打印的输出判断仿真是否成功。

4 实验结果及分析

需要仿真图一张，控制台打印输出图一张，要求仿真图中包含 `pc`、`instr`、`rs`、`rt`、`rd`、`result` 信号，仿真图应在控制台打印输出 **Simulation succeeded** 时截图。控制台打印输出图为此时截图。

A Datapath代码

其他模块不需要填写(红字为内容说明，请删除)