中期答辩自述文档

1. 项目分工

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 学号 | 实验课时间 | 负责的工作（不是扣分项，但建议写，方便问答） |
| 沈泓立 | 12311016 | 周一56王晴 | CPU子模块、IO子模块 |
| 郑袭明 | 12311011 | 周一56王晴 | CPU顶层模块、仿真测试 |
| 刘安钊 | 12311020 | 周一56王晴 | 汇编代码、CPU内存子模块 |

1. 代码规范：结构化设计（是）

命名规范：下划线命名

注释要求：是，有必要的注释

符号化常量的定义及使用：是

1. CPU特性 (单周期), CPU时钟( 23.8 MHz) , **ISA**( RISC-V)

**存储方案**(哈佛)（指令和数据分开存储，可同时访问）

**IO方案**(MMIO)

本次大作业中要实现的指令集(请罗列)：

add, sub, xor, or, and, sll, srl, sra, addi, xori, ori, andi, slli, srli, srai, lw, sw, beq, bne, blt, bge, bltu, bgeu, jal, jalr, lui, auipc, ecall

1. CPU架构设计（内部模块及连线关系）/接口设计说明

IFetch:该模块实现了一个支持跳转和分支的指令获取单元，根据当前PC和跳转控制信号计算下一条指令地址，并从程序存储器中读取对应指令

instruction\_control:该模块根据输入的 RISC-V 指令译码生成控制信号，以驱动处理器的数据通路执行算术、跳转、存储、IO 等操作。

MemOrIO:该模块根据控制信号在内存和 I/O 之间进行数据读写选择，生成对应地址、写出数据及片选信号，并将读取数据送回寄存器文件。

reg\_and\_imm:该模块实现通用寄存器读写功能并根据指令类型生成相应的立即数，用于后续指令译码与执行。

Data\_mem:该模块作为数据存储单元，利用 RAM 根据读写控制信号对指定地址进行数据的读写操作

cpuclk:该模块通过行为级建模方式将输入时钟信号分频，生成近似于23MHz频率的输出时钟信号。

ALU:该模块根据控制信号对寄存器或立即数执行算术、逻辑与移位等运算，并输出结果、零标志和分支判断信号。

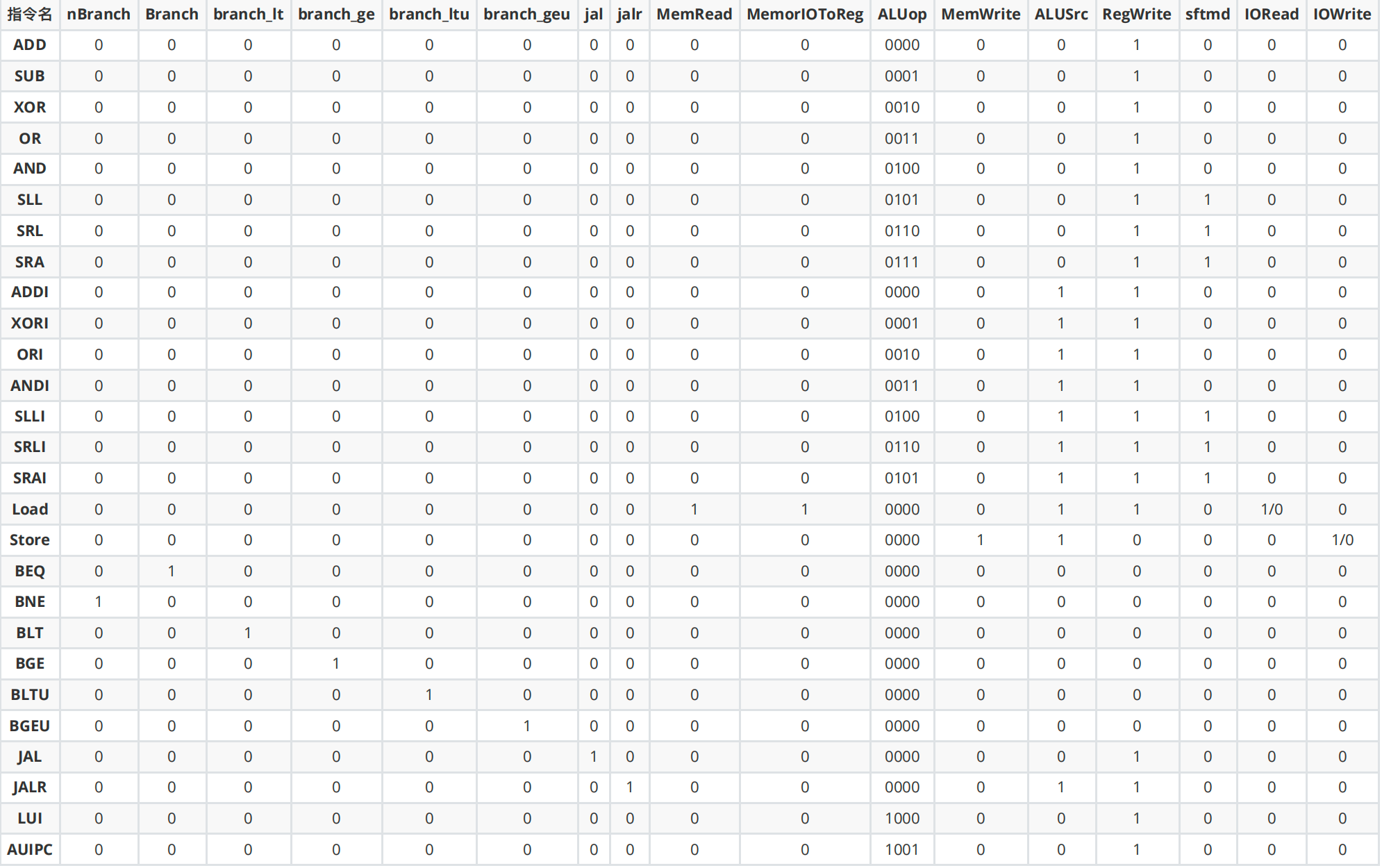
CPU:该模块集成各子模块实现一个简化的五级流水 RISC-V 处理器数据通路，支持指令获取、译码、执行、内存与 I/O 访问及结果写回。

switch:该模块根据地址和控制信号从输入开关中提取不同格式的数据输出，供 CPU 读取用于测试编号、符号扩展等场景。

tub\_control:该模块根据输入的 4 位十六进制数据控制 7 段数码管的显示，用于显示数字或字母 A–F。

tub:该模块采用时分复用方式轮流激活八位数码管，通过计数器依次显示输入的八个段码，实现动态刷新显示效果。

1. CPU指令于控制信号的关系



1. 项目实施的情况，已完成的项目代码

|  |  |  |
| --- | --- | --- |
| 单项 | 完成百分比 | 备注（待完成/待解决） |
| 文档 | 0% | 待完成 |
| CPU模块设计 | 80% | 更多指令待完成 |
| CPU顶层设计 | 70% | IO指令未放入顶层模块，待完成 |
| CPU 单元测试 | 80% | 已完成的子模块均完成仿真测试 |
| CPU 集成测试 | 50% | 尚未将IO模块放入顶层中测试，还没上板测试 |
| IO 设计 | 50% | 完成switch，led子模块设计，尚未完成整合 |
| IO 测试 | 0% | 待完成 |
| 程序下发方式 | 50% | ip核直接加载coe文件 |
| 工具链 | 100% | 采用rars，vivado |
| 基本测试场景1 | 80% | 完成汇编代码，上板测试待完成 |
| 基本测试场景2 | 0% | 待完成 |
| 附加测试场景 | 0% | 待完成 |

1. 计划使用/开发的工具链

rars vivado

1. 项目整体进度（\_60\_%）当前困难或问题、预计最终答辩时间、后续计划

当前困难：上板测试尚未完成，不清楚IO模块的效果

预计最终答辩时间：15周

后续计划：完成IO设计和基本测试场景的汇编代码，并上板测试