# RISC-V指令集CPU设计报告 F1603024 516030910582刘啸远

## 任务描述

实现一个支持RV32I Base Integer Instruction Set V2.0中2.1-2.7指令的CPU，并烧在FPGA(Basys3)上，使用UART协议与电脑通信，使用电脑内存作为其内存。

## 模块划分

该CPU结构类似TOMASULO，支持多个ALU超标量，但与TOMASULO在预约栈、ROB的实现上有所不同，以下为其模块简图。

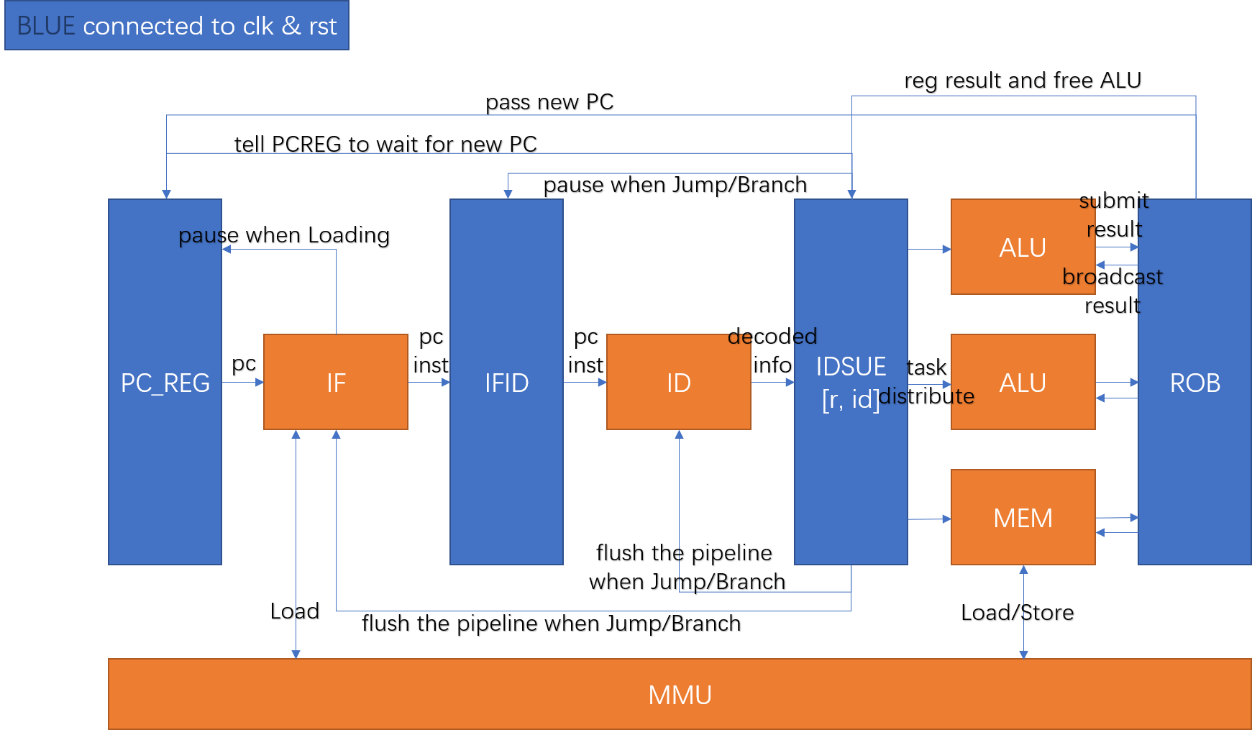


图1：CPU模块简图

|  |  |  |
| --- | --- | --- |
| 模块名 | clk? | 功能 |
| PC\_REG | √ | 生成新指令地址，跳转时接受新地址 |
| IF | × | 接收pc值并访存取指令，在读取失败时暂停PC\_REG |
| IFID | √ | 传递pc、指令内容 |
| ID | × | 将指令分解为具体运算类型、寄存器、立即数等具体部分 |
| IDSUE | √ | 接收、解析指令（寄存器改名），并选择空闲的处理单元发射出去 |
| ALU | × | 接收运算任务，运算，提交至ROB |
| MEM | × | 接收存储/读取任务，结束后提交至ROB |
| ROB | √ | 依次向IDSUE提交运算结果修改寄存器，并对结果广播 |
| MMU | × | 作为CPUCORE和外层的中继，修改访问格式 |

## 设计特点

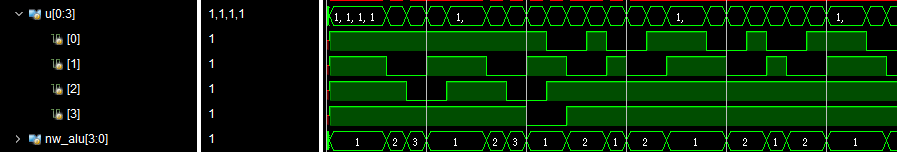
1. 类似Tomasulo，允许超标量

图2：连接3个ALU（图中[1][2][3]）和1个MEM（图中[0]）时工作情况（低电平为工作）

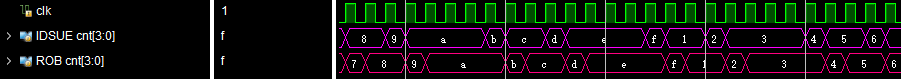
1. 预约栈合并进入ALU，ALU监听ROB广播来获取未知操作数
2. “ROB”不再是所谓Buffer，而是通过循环编号（1-f）来与IDSUE同步

图3：两阶段内置计数器进行同步，舍弃Buffer表项

1. 跳转通过IDSUE暂停流水线和向ROB报Exception完成，由ROB向PC\_REG提交新pc

## 心得体会

1. **技能生疏**：verilog语言不熟，不熟悉硬件设计思路，烧板子之前没做过
2. **环境配置**：linux下板子不识别，VIVADO装了好几遍
3. **结构不够简化**：控制线思路上会乱，总会遗漏某些情况
4. **没有自己实现UART**：由于通信不是很了解，所以直接套了助教代码的CPU\_core
5. **缺少设计经验**：经历了板子LUT不够（25000+>20800）以至于砍设计；经历了不满足时间要求（WNS -3ns左右）以至于降频到50MHz。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Name | Slice LUTs | | cpu | 25498 | | UART (uart\_comm) | 205 | | MEM\_CTRL (memory\_controller) | 91 | | d7s (display\_7seg) | 73 | | CORE (CPUCORE) | 24606 | | \_rob (ROB) | 924 | | \_pcreg (PCREG) | 66 | | \_mmu (MMU) | 104 | | \_mem (MEM) | 216 | | \_ifid (IFID) | 1544 | | \_if (IF) | 11 | | \_idsue (IDSUE) | 18971 | | \_id (ID) | 693 | | \_alu\_2 (ALU\_2) | 659 | | \_alu\_1 (ALU\_1) | 781 | | \_alu\_0 (ALU) | 677 | | COMM (multchan\_comm) | 521 | | clk (clk\_wiz\_0) | 0 | |

图4：初稿使用LUT超出板子20800的限制

1. **时间不是很够：**基本上把整个结构思路理清的时候已经考试周了，没有足够的时间进行仔细的调试和出数据，也没有时间完整的学习助教的通信方式导致踩了不少坑。板子上最后也没有调对，十分遗憾。

## 参考资料以及感谢

感谢范舟同学和吴章昊同学提供的测试数据；感谢张凯羿同学在二进制文件方面提供的帮助；感谢孙雪晖同学在烧板子方面提供的协助；感谢其他同学在我完成该项作业时的支持

**参考资料**如下：

RISC-V官方文档 <https://riscv.org/specifications/>

烧板子相关<https://wenku.baidu.com/view/c9486ce6a45177232f60a2f6.html>

Zzk助教的mipsCPU <https://github.com/sxtyzhangzk/mips-cpu>

《自己动手写CPU》 雷思磊

《自己设计制作CPU与单片机》 姜咏江

《深入理解OpenRISC体系结构》 甄建勇

《圈圈教你玩USB》 刘荣

|  |  |
| --- | --- |
|  |  |