YZU-王桑平 NSCSCC2023 设计报告

扬州大学
王桑平

一、设计简介

提交内容：用 verilog 设计的主频为 40MHZ 顺序单发射的五级流水线的 CPU，其实现个人赛的 mips 所需的全部指令，同时设计了串口控制器与 SRAM 控制器。

二、设计方案

（一）总体设计思路

本作品大体可以分为 CPU 核与串口控制器，SRAM 控制器组成的外围电路，该 CPU 核为顺序单发射五级流水线，将指令的执行分为:取指，译码，执行，访存，回写五个部分，而其中对于数据冒险与控制冒险的解决尤其重要，对于数据冒险我使用数据前推方案，利用四选一多路选择器将执行，访存，回写的数据前推到译码阶段，而对于 LW 和 LB 指令，在译码阶段如果发现执行阶段的指令是 LW 或 LB 指令并且译码阶段需要 LW 或 LB 指令取出的值时，译码将会阻塞直到 LW 或 LB 取到数据，并通过数据前推通路前推到译码。对于控制冒险，由于我的跳转指令的执行在五级流水线的执行阶段，故当跳转指令需跳转时将形成两个“气泡”,但由于延迟槽的存在，我需处理一个气泡所带来的影响，故当跳转指令需跳转时将清空译码寄存器的值。

（二）CtrlAssemblyLine 模块设计

该模块是对流水线进行控制，主要是对于是否需要数据前推，流水线的阻塞和当执行跳转指令时是否要清除译码寄存器的值的控制。

CtrlAssemblyLine 模块设计（主要信号）	
rs_addr	rs 地址
Rt_addr	Rt 地址
Rs_en	Rs 使能
Rt_en	Rt 使能
EX_rd_en	执行阶段数据是否需要回写到寄存器堆

EX_vaild	执行阶段数据是否有效
EX_rd_addr	执行阶段数据回写到寄存器堆的地址
MEM_rd_en	访存阶段数据是否需要回写到寄存器堆
MEM_vaild	访存阶段数据是否有效
MEM_rd_addr	访存阶段数据回写到寄存器堆的地址
WB_rd_en	回写阶段数据是否需要回写到寄存器堆
WB_vaild	回写阶段数据是否有效
WB_rd_addr	回写阶段数据回写到寄存器堆的地址
Pc_stop	译码阶段是否阻塞
IF_ID_clear	译码寄存器是否清空
alu_in1_sel	rs 是否需要数据前推
alu_in2_sel	rt 是否需要数据前推

（三）PC 模块设计

PC 模块是取指的核心，其在主要用于更新 PC 的值。

PC 模块设计（主要信号）	
IF_ID_allowin	译码阶段是否允许进入
PC_to_IF_ID_vaild	当前 PC 值是否有效
Im_addr	当前 PC 值
Nextpc	下一 PC 值

三、设计结果

（一）设计交付物说明

| design.pdf :本文档

| README.md

| thinpad_top.xpr :vivado 启动文件

+ .ci-scripts :ci 文件

+ .git

+ asm

\---thinpad_top.xpr

+ constrs_1 :约束文件

+ sim_1 :仿真文件

/---sources_1

+ ip

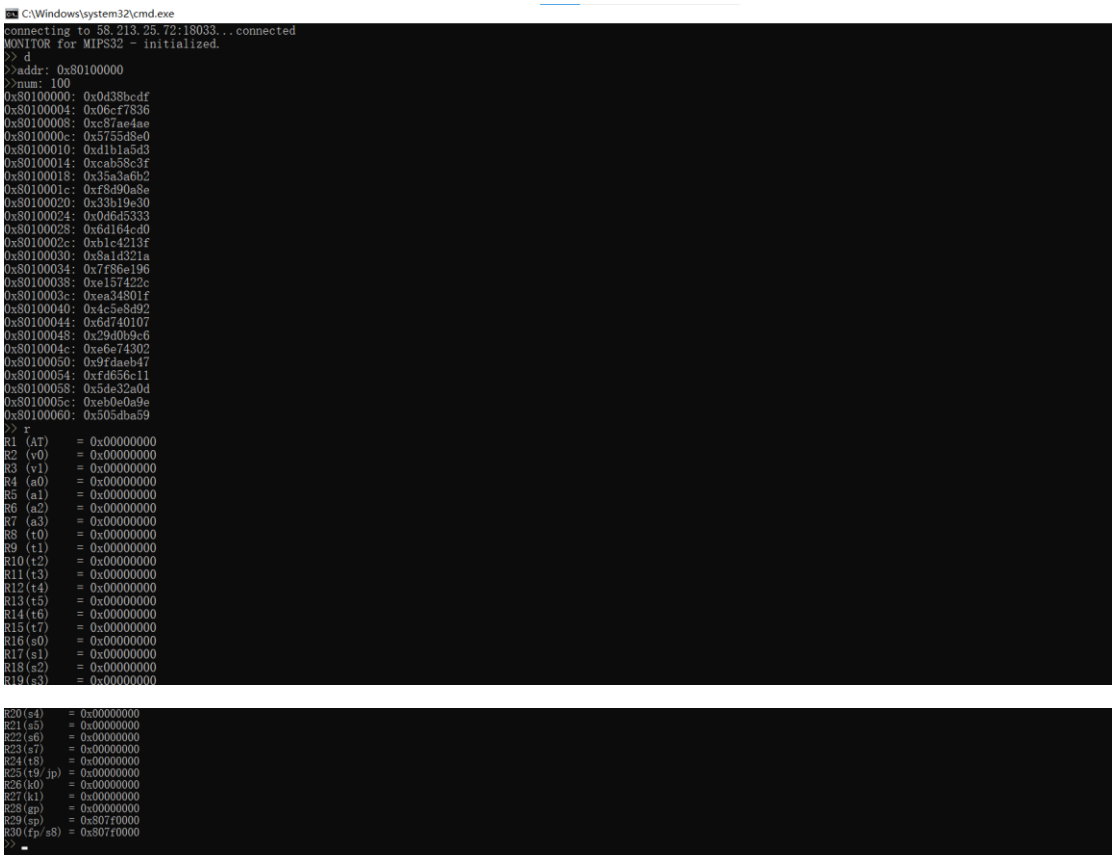
| \---pll_example

+ myCPU :SRAM,串口控制器和 CPU 的 verilog 源码

+ new :顶层文件

（二）设计演示结果

展示通过比赛提供的 win10 环境连接线上实验平台，运行监控程序 kernel.bin 的结果。



四、参考设计说明

借鉴了《CPU 设计实战》中对于流水线电路的 ready_go,allowin 和 valid 信号的定义与使用。

五、参考文献

【1】姜文祥，刑金璋　《CPU 设计实战》
【2】雷思磊　《自己动手写 CPU》