龙芯杯个人赛 CPU 设计报告

北京科技大学
郭睿

一、设计简介

本次设计是一个支持 MIPS 指令集子集的简单流水线 CPU，实现了 25 条指令，其中覆盖了 MIPS-C3 指令集，并额外添加了 ADDI,SRLV,BLEZ 三条指令，可与远程服务器上的 sram 和串口进行协调工作，并可正确运行大赛指定程序通过所有自动测试。

初期使用 visio 设计 CPU 的总体结构图，后期根据结构图使用 verilog 实现，并使用 vivado 仿真，综和，实现并生成比特流。

此流水线 CPU 设计支持 50MHz 时钟频率，每条指令从取指到写回寄存器堆需要至少 4 个时钟周期，也就是相当于四级流水，流水级间支持数据前递，并且该设计可在译码段得出分支指令的分支结果，所以分支时不需要插入气泡，由于大赛程序的特殊性，指令与数据可共用 ram，所以在取指与访存指令访问同一块 ram 时需要暂停取指并插入气泡。

此 CPU 在大赛指定的所有程序上都能正常工作。

二、设计方案

（一）总体设计思路

遵循 MIPS 的规定，CPU 中需要 32 个通用寄存器，可组成一个寄存器堆。存储器使用大赛平台上的 base_ram 和 ext_ram，cpu 需要根据指令的执行情况综合判断生成他们的控制信号。Mips 中的指令大致需要经过 5 个阶段，取指令，指令译码，执行运算，访存，写回寄存器。考虑到只有 lw，lb，sw，sb 四条指令在执行时需要访存，所以在我的设计中将访存工作与 ALU 运算工作放到一个时钟周期内完成。在本设计中，一条指令执行过程如下，在内存访问控制器的控制下从 ram 中取指令，然后到译码段进行指令译码，生成各部件的控制信号与 ALU 的操作符与操作数，然后进行 ALU 的运算或对内存的访问，最后将结果写回寄存器堆。总体设计图如下(其中简写或省略了部分接口与信号)：

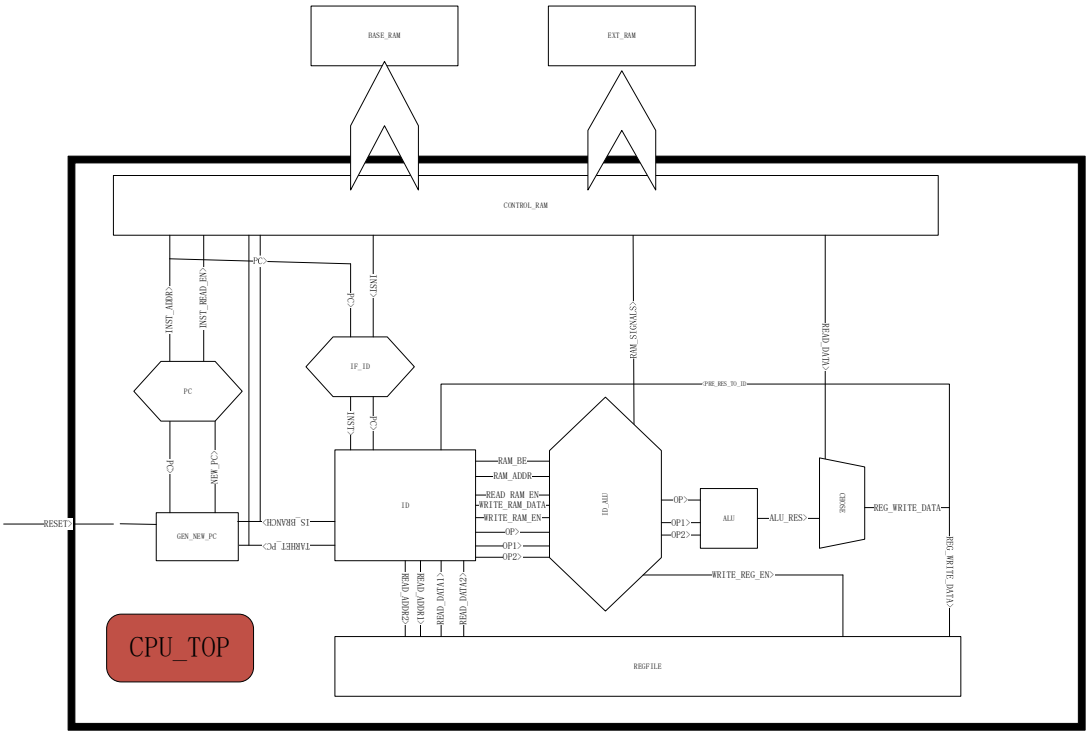

Figure 1

串口与 CPU 交互的简单设计图如下：

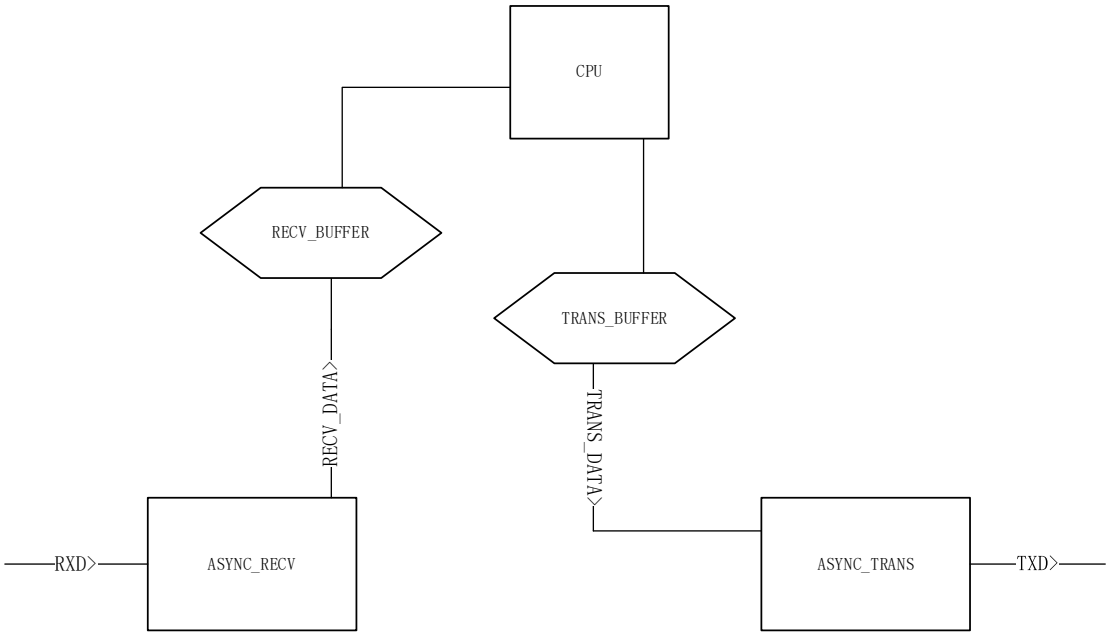

Figure 2

译码时会检测是否有寄存器写后读冲突并通过将写回值前递到译码的方式解决冲突。

在设计时需要实现以下模块功能：

Table 1

模块名	模块功能与接口的粗略描述

Control_ram	根据即将取指的指令地址和访存指令的访存请求以及地址判断生成 base_ram,ext_ram 的控制信号和暂停取值信号。
Inst_fetch,	根据当前指令地址与 Control_ram 模块交互从 ram 中取指令并综和跳转信号生成下一条指令的地址。
Inst_decoder	根据 Inst_fetch 模块取得的指令对指令进行译码，生成访存控制信号送给 Inst_excute 模块，寄存器读写信号，以及跳转信息和执行操作与操作数，送给 Inst_excute 模块。
Inst_excute	根据指令的运算操作和操作数进行运算或者访存，最后二选一作为最后写回寄存器堆的值。
Reg_file	32 个 32 位的通用寄存器的集合，提供两个异步读口和一个同步写口。

CPU 顶层为 GR_core，然后在 thinpad_top 中实例化并与串口和 base_ram,ext_ram 控制信号相连。

三、设计结果

RTL 级视图：

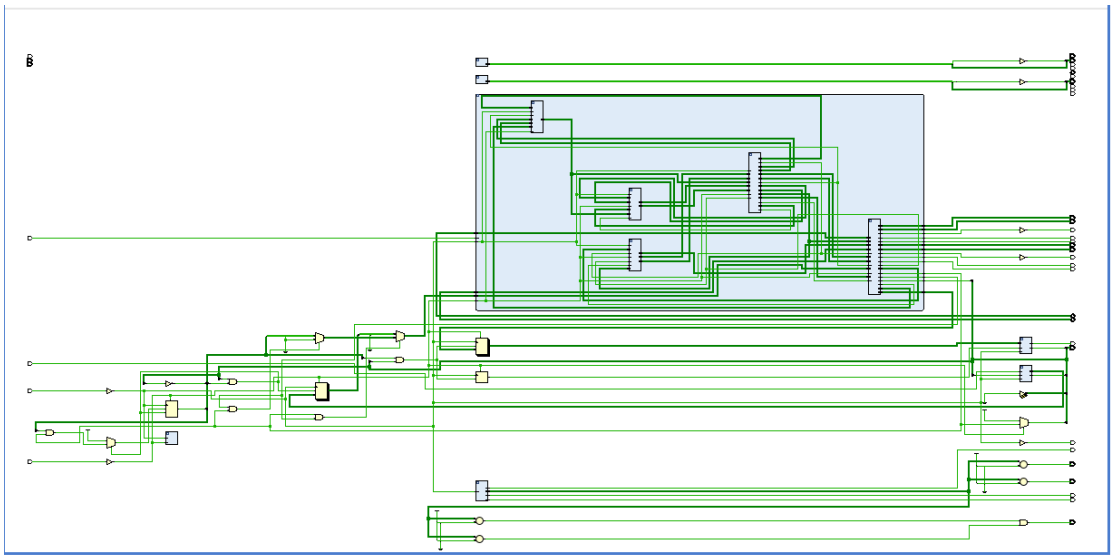

Figure 3

（一）设计交付物说明

由于我的主要工作代码编写都在/thin_pad.srcs/下完成的，且其他部分的向.gitignore 文件和 ci 文件基本都为赛方提供的，所以这里就不再赘述。

-thin_pad.srcs/

-constrs_1/ 赛方提供的，不再展开

-sim_1/ 包含行为级仿真时的 testbench 文件，不再展开

-sources_1/

-ip/ ip 核目录

-new/ 设计源文件目录

-gr_cpu/ 包含 CPU 设计文件

-inst_decode/ 译码段文件

-Inst_decoder.v Inst_decoder 模块

-inst_exc/ 执行访存段文件

-ALU.v ALU 模块

-Inst_ex.v 执行与访存模块

-inst_fetch/

-Gen_new_PC.v 生成下一个 PC 的模块

-PC.v PC 寄存器

-Inst_fetch.v 取指令模块

-Control_ram.v 内存访问控制模块

-GR_core.v CPU 核心部分的顶层模块

-inst_define.v 指令操作码以及 funct 段，还有 ALU 运算符的宏定义

-Reg_file.v 寄存器堆模块

-async.v 赛方提供的串口接收以及发送模块

-thinpad_top.v 赛方提供的顶层模块

。。。。其余不再赘述

综和实现全部由 ci 完成。

（二）设计演示结果

设计并实现的 50MHz 四级流水线 支持数据前递的 CPU 在性能测试上的结果：

Table 2

程序	STREAM	MATRIX	CRYPTONIGHT
时间(s)	0.110	0.179	0.461

性能测试截图：

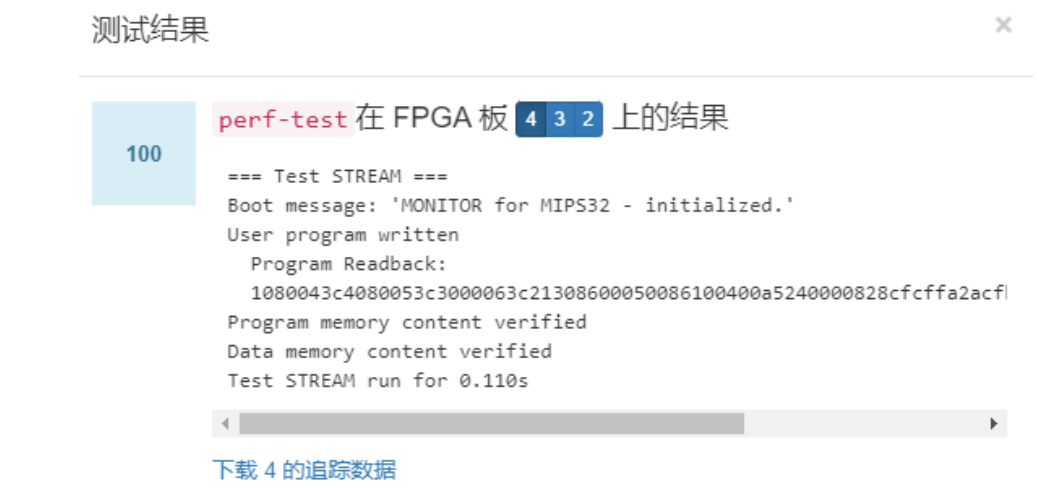

Figure 4

Figure 5

Figure 6

四、参考设计说明

赛方提供的顶层模块使用了 MMCM 可生成不同时钟频率的 IP 核，但是此设计中 CPU 使用了外部时钟，并没有用到该 IP 核。

在流水线以及 MIPS 指令集，CPU 等理论上参考了汪文祥老师主编的《CPU 设计实战》这本书，在 ALU 以及 Reg_file 模块实现上参考了这本书中的实现。

五、参考文献

[1] 汪文祥. 《CPU 设计实战》[M]. 出版地:北京 机械工业出版社，出版年:2021.