龙芯杯个人赛设计报告

学校 清华大学
姓名 尤忆晨

一、设计简介

单发射五级流水线 CPU，采用两周期访存，添加了 icache 模块，主频达到 **117.86MHz**，通过了所有测试，性能测试成绩为 0.067s + 0.099s + 0.236s = **0.402**s

二、设计方案

（一）总体设计思路

本设计采用经典的五级流水线 CPU 架构，参考了《自主设计 CPU》中的架构，流水线部分为 IF、ID、EX、MEM、WB 五个部分，在此基础上添加了串口控制模块以及 sram 控制模块，同时将访存控制模块单独作为一个模块，以方便调整访存周期数和后续添加 dcache。流水线的步骤如下：

- 取指阶段：取指阶段的主要职责是从指令存储器中提取下一条待执行的指令。在这个阶段，处理器会使用程序计数器（PC）中的值作为地址，访问指令存储器并读取相应的指令。读取到的指令随后被放置在指令寄存器中。与此同时，PC 的值会被更新，通常是增加一个固定值，以指向下一条指令的地址，为下一个周期的取指做好准备。
- 译码阶段：译码阶段的核心任务是对取回的指令进行解析和准备。在这个阶段，处理器会分析指令的结构，提取出操作码并确定指令的具体类型。根据指令类型，处理器会执行不同的操作，如确定需要使用的寄存器、计算可能的跳转地址等。此外，译码阶段还负责从寄存器文件中获取指令所需的源操作数，并为执行阶段做好准备。
- 执行阶段：执行阶段是整个流水线中最为关键和复杂的部分，它负责实际执行指令所指定的操作。在这个阶段，处理器会根据指令的类型和操作码，利用算术逻辑单元（ALU）执行相应的运算。这可能包括各种算术操作（如加法、减法、乘法）、逻辑操作（如与、或、非）等。对于访存指令，执行阶段还负责计算有效的内存地址。此外，执行阶段还可能涉及条件判断、数据移动等多种操作。
- 访存阶段：访存阶段主要处理与数据内存之间的交互。在这个阶段，处理器会利用在执行阶段计算得到的内存地址，进行数据的读取或写入操作。对于加载（Load）指令，访存阶段会从指定的内存地址读取数据；对于存储（Store）指令，则会将数据写入指定的内存地址。对于不需要访问内存的指令，这个阶段可能不会执行任何实质性的操作。
- 写回阶段：写回阶段是指令执行过程的最后一个环节，主要负责将指令执行的结果写回到寄存器文件中。在这个阶段，处理器会将执行阶段或访存阶段产生的结果写入到指令指定的目标寄存器中。这个过程确保了计算结果能够被后续的指令使用，从而维持了程序执行的连续性。写回阶段的完成标志着一条指令执行周期的结束，为下一条指令的执行做好了准备。

（二）icache 模块设计

在基础的**直接映像**指令 cache 基础上，我相继设计了**最近最少替换策略的 2-路组相联** cache 以及**先入先出替换策略的 4-路组相联** cache，经过测试发现，由于性能测试的指令数

量较少，直接映像就可以存储大部分指令，2 路和 4 路由于增加了额外的逻辑，效果不如直接映像好，但文件中保留了设计文件，以方便决赛时直接替换。

（三）mem_controller 模块设计

为了方便后续添加 dcache 和调整访存周期数，我将访存控制部分单独作为一个模块。现有的设计采用状态机控制的**两周期访存**，主频可以达到 117.86MHz，设计中曾为了提高主频采用三周期访存，但是主频仅提高了不到 10MHz，性能测试结果反而不如两周期访存，后续考虑继续优化关键路径，采用三周期继续提高频率。

（四）乘法器模块设计

开始时采用**六层华莱士数**设计，但是时序分析时发现华莱士数占用 lut 资源过大且布线延时过大，因此采用 **0 周期的 multplier ip 核**代替，使用 dsp 资源来降低 lut 资源使用；后续计划使用流水线乘法器来优化关键路径。

三、设计结果

（一）设计交付物说明

表一：文件交付说明

文件夹		
cache	icache_2way.v	两路组相联 cache
	icache_4way.v	四路组相联 cache
	icache_direct.v	直接映像 cache
	icache_direct_3.v	三周期访存 cache
controller	mem_controller.v	访存控制
	mem_controller_3.v	三周期访存控制
	sram_controller.v	sram 控制
	stall_controller.v	流水线暂停控制
	uart_controller.v	串口控制
mul	adder.v	3-2 全加器
	wallace.v	六层华莱士树
	defines.v	宏定义
	if_state.v	取值阶段
	if_id_reg.v	if/id 寄存器

	id_state.v	译码阶段
	id_ex_reg.v	id/ex 寄存器
	ex_state.v	执行阶段
	ex_mem_reg.v	ex/mem 寄存器
	mem_state.v	访存阶段
	mem_wb_reg.v	mem/wb 寄存器
	regfile.v	寄存器
	yycpu.v	主程序

（二）设计演示结果

表二：功能测试成绩

一级评测	100
二级评测	100
三级评测	100
性能测试	100

表三：性能测试成绩

STREAM	0.067s
MATRIX	0.099s
CRYPTONIGHT	0.236s

图一：STREAM

图二：MATRIX

图三：CRYPTONIGHT

四、参考设计说明

1. 串口控制和 cache 参考了 2023 年龙芯杯个人赛开源项目 mycpu 的设计
2. 五级流水线框架和接口参考了雷思磊的《自己动手写 CPU》中的设计
3. 华莱士数乘法器部分参考了博客中的设计（最终提交版本未使用）

五、参考文献

[1] 雷思磊. 自己动手写 CPU. 北京: 电子工业出版社,2014.
[2] David A. Pattreson, John L. Hennessy. 计算机组成与设计. 北京: 机械工业出版社,2015