重庆大学课程设计报告

课程设计题目:	MIPS SOC设计与性能优化
学　　院:	计算机学院
专 业 班 级:	计算机科学与技术01班
年　　级:	2019
学　　生:	姓名1　　姓名2
学　　号:	学号1　　学号2
完 成 时 间:	2019年　12月　30日
成　　绩:	90
指 导 教 师:	钟将

重庆大学教务处制

3*项目	3*分值	优秀 $100 > x \geq 90$	良好 $90 > x \geq 70$	中等 $80 > x \geq 70$	及格 $70 > x \geq 60$	不及格 $x < 60$	3*评分
		参考标准					
学习态度	15	学习态度认真，科学作风严谨，严格保证设计时间并按任务书中规定的进度开展各项工作	学习态度比较认真，科学作风良好，能按期圆满完成任务书规定的任务	学习态度尚好，遵守组织纪律，基本保证设计时间，按期完成各项工作	学习态度尚可，能遵守组织纪律，能按期完成任务	学习马虎，纪律涣散，工作作风不严谨,不能保证设计时间和进度	
技术水平与实际能力	25	设计合理、理论分析与计算正确，实验数据准确，有很强的实际动手能力、经济分析能力和计算机应用能力，文献查阅能力强、引用合理、调查调研非常合理、可信	设计合理、理论分析与计算正确，实验数据比较准确，有较强的实际动手能力、经济分析能力和计算机应用能力，文献引用、调查调研比较合理、可信	设计合理，理论分析与计算基本正确，实验数据比较准确，有一定的实际动手能力，主要文献引用、调查调研比较可信	设计基本合理，理论分析与计算无大错，实验数据无大错	设计不合理，理论分析与计算有原则错误，实验数据不可靠，实际动手能力差，文献引用、调查调研有较大的问题	
创新	10	有重大改进或独特见解，有一定实用价值	有较大改进或新颖的见解，实用性尚可	有一定改进或新的见解	有一定见解	观念陈旧	
论文(计算书、图纸)撰写质量	50	结构严谨，逻辑性强，层次清晰，语言准确，文字流畅，完全符合规范化要求，书写工整或用计算机打印成文；图纸非常工整、清晰	结构合理，符合逻辑，文章层次分明，语言准确，文字流畅，符合规范化要求，书写工整或用计算机打印成文；图纸工整、清晰	结构合理，层次较为分明，文理通顺，基本达到规范化要求，书写比较工整；图纸比较工整、清晰	结构基本合理，逻辑基本清楚，文字尚通顺，勉强达到规范化要求；图纸比较工整	内容空泛，结构混乱，文字表达不清，错别字较多，达不到规范化要求；图纸不工整或不清晰	

指导教师评定成绩：

指导教师签名：

MIPS SOC设计报告

姓名1、姓名2

1 设计简介

本次硬件综合实训设计了一个基于经典五级流水线可以上板运行 57 条 MIPS 指令并通过 axi 总线连接了 cache 的 CPU。实验的最初代码来自于小组成员在计算机组成原理课程实验 4 中设计的可以运行 10 条 MIPS 指令的 CPU 和实验五实现的写回 cache。在此基础上经过以下 5 个阶段的修改，我们逐步完成了最终的 CPU 设计。

- 第一阶段：在原有代码的基础上我们按照逻辑运算指令-> 移位运算指令-> 数据移动指令-> 算术运算指令-> 转移指令的顺序，逐步扩展到 52 条指令。这一部分主要修改了 controller 、alu 和 datapath ，除此之外还添加了 hilo 寄存器，乘除法器等功能模块。
- 第二阶段：我们在扩展到 52 条指令过后就通过 sram 接口连接 soc。然后调试了 hazard 模块。
- 第三阶段：这个阶段添加的 5 条指令全部和异常处理有关，包括特权指令和自陷指令，为此我们添加了 cp0 协处理器。
- 第四阶段：在这个阶段，我们将 CPU 通过类 sram 接口连接到 axi 总线，这里参考了计算机组成原理实验 5 的代码，复用了部分 axi 接口相关的文件和完成的写回 cache，经过我们的调试后先后通过了 axi 的功能测试和性能测试并上板。
- 第五阶段：尝试优化 CPU 性能，如 cache ，除法器。

1.1 小组分工说明

- 邹正强：负责逻辑运算、数据移动指令、连接axi总线。
- 黄优文：算术运算指令、转移指令、异常处理指令。
- 罗梓元：移位运算指令、Cache设计、harzard 模块调试与修改。

2 设计方案

2.1 总体设计思路

本次硬件设计主要设计和完善了一个可以基于 AXI 总线协议运行的 SOC 系统。其中,系统内存 (Ram)、外设 (Confreg), 以及负责管理和仲裁上述从设备与处理器间的数据交互请

图 1: SOC顶层结构图

求, 并进行数据传输的 AXI 1X2 Bridge 模块均已在课程资料包中提供。因此, 本次设计的主要目标是构建一个支持 AXI 协议的 MIPS32 架构的 CPU, 即下图中的mycpu_top模块。本项目设计的mycpu_top主要包含以下模块组件：

中央处理器MIPS：在计算机组成原理实验四所实现的五级流水线CPU基础上进行拓展，实现了MIPS32指令系统中所有非浮点MIPSI指令以及MIPS32中的ERET指令，还实现了少量CP00寄存器来支持中断和系统调用功能，但未实现TLB、MMU和特权等级。

内存管理单元MMU：采用直接映射方式，通过读取地址高位将CPU需访问的虚拟地址转换成物理地址，并判断数据访问请求是属于内存访问还是外设访问。

桥Bridge 1x2：依据MMU给出的数据访问请求类型，对内存访问请求和外设访问请求进行分离或合并操作。需要注意的是，仅内存访问请求会经过数据Cache。

高速缓存Cache：旨在弥补CPU与内存之间的速度差异，由指令缓存InstCache和数据缓存DataCache两部分构成。

桥Bridge 2x1：是Bridge 1x2的逆向工程。与Bridge 1x2不同，Bridge 2x1在某一时刻可能会同时接收到数据Cache发起的内存访问请求和CPU发出的外设访问请求，其内部仲裁策略为优先处理内存访问请求。

AXI接口AXInterface：mycpu_top内部采用类SRAM方式实现数据交互，所有对外的类SRAM请求均由AXI接口统一转换为AXI请求，同时AXI接口会将外层返回的数据重新转换为类SRAM格式供内层模块使用，其内部仲裁策略为优先指令访问

2.2 XX模块设计（可选）

对模块内部设计方案进行更进一步描述。可以包含：模块的功能意图，模块的输入输出，模块内部的数据通路和控制逻辑，以及可能的软硬件交互机制。

3 设计过程

3.1 设计流水账

2024 年 12 月 25 日 12:30 ~ 15 : 30
到教室上课，听助教讲解硬件综合设计课程的大致规划和任务目标。

2024 年 12 月 25 日 12:30 ~ 15 : 30
配置了 vivado 2020.1 的环境，并将 vivado 的编辑器连接到了 vscode，这样可以获得更清晰的代码高亮和补正。并选择 lab4 代码作为我们小组的初版代码，将其代码连接到了资料包中的 lab4 工程并成功运行。剩下的时间开始各自学习重庆大学硬件综合设计实验文

档的内容，修改代码。

2024 年 12 月 26 日 12:30 ~ 15 : 30
邹正强与罗梓元添加了逻辑运算指令和移位指令。

2024 年 12 月 28 日 12:30 ~ 15 : 30
邹正强添加了数据移动指令，小组一起讨论了 hilo 寄存器的放置位置，最终决定将其放置在 MEM流水线级。

2024 年 12 月 30 日 12:30 ~ 15 : 30
黄优文添加了算术运算指令。注意到拓展任务中的除法器的优化，小组开始查阅资料，寻找优化除法器的方法。

2024 年 12 月 31 日 12:30 ~ 15 : 30
黄优文添加了分支跳转指令。

2025 年 1 月 1 日 12:30 ~ 15 : 30
罗梓元添加了访存指令。

2025 年 1 月 2 日 12:30 ~ 15 : 30
邹正强连接了 sram-soc 运行功能测试，发现参考代码中的除法器不能通过功能测试中的除法，经过调试后通过，同时也解决了一些其他冒险模块的 bug 后通过功能测试的大部分测试点。

2025 年 1 月 3 日 12:30 ~ 15 : 30
小组观看了前几届学长录制在 B 站的视频，学习了异常处理有关的知识和注意事项，讨论了异常处理的相关设计思路，小组决定直接调用参考代码中的 cp0。

2025 年 1 月 4 日 12:30 ~ 15 : 30
黄优文完成了异常处理相关的数据通路的添加，开始调试。罗梓元完成了 cp0 相关数据通路的添加，开始调试。

2025 年 1 月 5 日 12:30 ~ 15 : 30
小组在与其他小组互相交流调试心得后，完成了代码的进一步调试。

2025 年 1 月 6 日 12:30 ~ 15 : 30
小组观看了 B 站学长的视频，准备开始连接 axi 总线，但是发现本次实验的资料包中没有提供相关的转接口代码，为了加快进度避免重复的工作，决定复用计算机组成与结构实验 5 的接口代码。

2025 年 1 月 7 日 12:30 ~ 15 : 30

邹正强连接好 axi 总线后开始测试 axi 项目的功能测试。发现优化后的除法器不能通过部分测试点，黄优文开始修改除法器，最后通过了功能测试。

2025 年 1 月 8 日 12:30 ~ 15 : 30
决定采用罗梓元的 cache 代码来进行后续的开发。连接了计组实验 5 中编写的写回 cache，成功通过性能测试并上板得到了性能得分，并继续优化 cache。

2025 年 1 月 9 日 12:30 ~ 15 : 30
在确定代码的大致通路后，邹正强开始用 Viso 绘制数据通路图。罗梓元与黄优文开始编写部分报告内容。罗梓元替换二路组相联写回cache 后发现上不了板，最终决定采用原来的写回cache。

2025 年 1 月 10 日 12:30 ~ 15 : 30
小组现场添加指令和答辩。

2025 年 1 月 11 日 12:30 ~ 15 : 30
小组撰写剩余的报告内容。

3.2 错误记录

3.2.1 错误1

(1) 错误现象:描述这个错误产生时的现象。
(2) 分析定位过程:说清楚你碰到这个问题是如何分析定位出错原因的。可能你分析定位过程中经历了多轮尝试，把它们都记录下来。
(3) 错误原因:给出一个出错原因的正式说明。
(4) 修正效果:说明你修正这个错误的方法，并说明它是否有效。
(5) 归纳总结（可选）:说说你觉得这个错误是哪种类型的，今后如何提前规避。

3.2.2 错误2

4 设计结果

请不要大篇幅地直接粘贴代码。

4.1 设计交付物说明

说明所提交设计的目录层次，各目录下对应的内容是什么。提供所提交设计进行仿真、综合、上板演示的必要操作提示步骤。

4.2 设计演示结果

以文字、图、表等形式展示设计的演示结果。

5 参考设计说明

如果在提交设计中使用了第三方IP或者借鉴了他人的部分源代码，请在此处逐一列举，并说明出处。所谓“借鉴”是指从模块划分、接口定义、数据通路结构、状态机、关键信号含义这些方面均与原设计存在较高的相似度。

6 总结

7 供同学们吐槽之用。有什么问题都可以直接写在这。

7.1 姓名1

.....

7.2 姓名2

.....

8 参考文献

由于Latex有自带的参考文献引用方式，可以直接采用谷歌学术、百度学术搜索论文、课本的方式，选择bib格式的内容合并至一个文件（reference.bib），添加到该目录内，使用下方命令即可自动生成。，提交时请删除本段section