

哈爾濱工業大學

本科毕业论文（设计）开题报告

题 目：基于 LA32R 的处理器微结构设计及性能分析

专 业 计算机科学与技术

学 生 杨舒翔

学 号 2200400728

班 号 2004104

指导教师 华栋

日 期 2023 年 10 月 31 日

哈尔滨工业大学教务处制

1 课题来源及研究的目的和意义

1.1 课题背景

CPU 被称为计算机的大脑和心脏，是国家的重要战略产品，同时也是一个十分复杂的系统。我国处理器研发起步相对较早，但发展历程却曲折不平^[1]。上世纪 60 年代，我国自主设计出与国际水平相近的计算机系统。然后到 70 年代，由于经济条件和国际技术封锁等原因，我国丧失了对第四代计算机系统的研究能力。从“七五”到“九五”，国家对国产 CPU 的支持力度明显下降，直接导致我国的 CPU 设计能力基本丧失。

然而，随着国内信息化的快速发展以及电子信息制造业的迅速崛起，“芯片荒”的问题再次受到国家的高度重视。自“十二五”以来，国家采取了一系列措施扶持国产 CPU 产业，培育出了一大批国产 CPU 设计单位和研究机构，百花齐放。通过校企合作和共同努力，我国的 CPU 设计领域终于走向正轨。

本课题正是源自我国完全自主知识产权的 CPU 供应商——龙芯中科。龙芯中科一直秉持“自主研发计算机处理器和芯片，降低对国际技术的依赖”的理念。从最初基于 MIPS 指令集的我国首枚具有自主知识产权的通用高性能微处理器芯片龙芯 1 号到后来基于我国首个 100% 自研的指令集^[14] LoongArch（简称“LA”）的龙芯 3A5000 处理器，但在性能方面龙芯 3A5000 与市面上主流的酷睿/锐龙处理器相比还存在一定差距。为了弥补这一差距，龙芯中科的研发团队不懈努力，于今年 8 月发布了龙芯 3A6000 处理器，其性能已经媲美 Intel 的第十代酷睿处理器。这标志着国产 CPU 的性能达到了一个新的高度。为了更广泛普及我国完全自主研发的 LoongArch 指令集，龙芯中科推出了 LA32R 这种 32 位的精简版指令集，旨在在全国各大高校推广教学使用。本课题正是基于 LA32R 指令集进行 CPU 的设计。

随着对处理器性能需求的增长，处理器设计师开始探索更多的设计技术，以满足不断增长的计算需求和处理复杂工作负载的能力。这包括使用多核处理器、采用超标量技术以提高指令并行执行的能力；采用 Cache、预取等技术改进内存层次结构；集成硬件加速器搭建异构系统，提高计算机的图形计算能力等方法。然而，如何评估某种设计技术是否确实能够提供更高的性能，需要对处理器进行性能分析。

经过数十年性能分析技术的发展，目前主流的性能分析方法是：选择用于性能评估的指标，运行诸如 SPEC 等基准测试程序评估处理器在通用工作负载下的性能；然后编写实际应用程序以了解处理器在实际工作负载下的表现。

1.2 研究的目的和意义

本课题源于龙芯中科与我国高等院校（包括北京航空航天大学、哈尔滨工业大学等）的 MIPS 教学合作。旨在通过这一合作更好地推广国内高等院校的处理器设计，以解决芯片领域的“卡脖子”问题。龙芯中科的首席 IP 设计师汪文祥提出了将 MIPS 迁移到 LA 的计划以更好地普及 Loong Arch 架构，更好地适应国情需要，培养更适合中国本土处理器的人才。本课题正是这一计划的一部分。

在先前的 MIPS 合作的基础上，我对课题中所迁移实现的 LA32R 指令集的五级流水线处理器进行了优化，引入了多核、Cache 以及静态双发射、预取等微结构技术。随后，我将在该 CPU 上移植 Linux 操作系统，以便进行实际应用的运行和更好地性能分析。

LA32R 是龙芯中科为了方便教学和科研而推出的 32 位精简指令集，相较 LA64 来说它实现简单、开发时间短且可以运行操作系统，门槛较低。但是目前开源的用于教学使用的 LA32R CPU 仅仅局限于五级流水，性能不高，无法满足现在对高性能处理器研究的教学和科研需求，本课题正是弥补这一空白，采用双发射、Cache、预取等微结构技术提高 LA32R CPU 的性能。

本课题的实践过程将大大提升我的通用处理器微结构设计能力。采用 LA32R 指令集将为我未来参与龙芯的课题研究奠定坚实的基础。此外，通过从零开始移植 Linux 系统，我将深入了解操作系统的底层机制，提高对系统调用效率的认识，这将在今后的编程和软件开发工作中产生深远的影响。我还将对新增的微结构改进的 CPU 进行性能分析，这亲身经历处理器性能分析过程，为未来的研究学习提供便利。

2 国内外在该方向的研究现状及分析

2.1 国外研究现状及分析

由于上世纪 70 年代以来的国际形势以及国内政策等多种因素，直到本世纪初处理器领域的话语权一直在国外^[1]。早在上世纪 80 年代末和 90 年代初，David W. Wall 在其论文《Superscalar Microprocessor Design》^{[7][11]}中详细阐述了超标量处理器设计的原理和技术，对本课题采用的多指令静态双发射技术进行了深入的讨论。而在 2000 年代初国外开始进行了对多核处理器的研究，Luiz André Barroso 和 Kourosh Gharachorloo 等人在 2007 年发表的论文《Piranha: A Scalable Architecture Based on Single-Chip Multiprocessing》中详细介绍了多核处理器的设计理念。在此期间，国外研究者在性能模拟和分析工具方面也做出了很多成果^[9]，例如 SPEC Benchmark 以及性能模拟器 Gem5 和内存分析工具 Valgrind 等。

但是由于 Loong Arch 是龙芯中科于 2021 年推出的由我国完全自主知识产权研发的指令集架构。国外在该 ISA 上所做的实践少之又少。

2.2 国内研究现状及分析

虽然在 20 世纪 70 年代到本世纪初，我国的处理器设计能力相对滞后，与国际水平存在较大的差距，但自“十二五”计划以来，我国积极跟随国际发展潮流，学习国际上先进的微结构设计方法^{[2][3]}和性能分析技术^[4]，并在此基础上进行创新。特别值得一提的是龙芯中科，经过 20 年的不懈努力，已经由最初的追随者发展成为能够完全独立自主设计处理器的实力派，最终在 2021 年正式发布了完全自主设计的 CPU 指令集——LoongArch。本课题正是基于 LA 架构 32 位精简版的指令集^[5]LA32R 展开研究。

在 LA32R 的研究领域，国内多家高校与龙芯中科展开了教学合作项目，例如，哈尔滨工业大学（深圳）的计算机设计与实验项目采用了 LA32R 的子集 miniLA，然而，该项目目前仅限于设计和仿真单周期 CPU，对 LA32R 的高级微结构设计尚未进行深入探讨。

而本课题即是为了弥补这一空白，补全 LA32R 实践的版图。

3 主要研究内容

3.1 基于教学合作完成从 MIPS 到 LA 的迁移

中科龙芯与北航合作的基于 MIPS 的实验平台包括五个实验，涉及到 Logisim 的使用、Verilog 部件的开发、MIPS 汇编语言的特定功能函数的实现、32 位单周期处理器和五级流水处理器的实现。本课题的首要任务是将基于 MIPS 的实现转变为基于 LA32R 的实现，具体包括将实验三中的基于 MIPS 的汇编语言转化为基于 LA32R 的汇编语言，并将处理器的指令集架构迁移为 LA32R 指令集架构。

3.2 对五级流水 CPU 微结构进行设计优化

完成 3.1 之后，本课题将展开对 LA32R 的高级微结构设计的探索。参考李亚民教授所著的《计算机原理与设计——Verilog 版》^[10]中关于基于 MIPS 的 CPU 的中断、Cache、TLB 以及多核实现的描述，我将这些设计的实现迁移到在 3.1 中实现的 LA32R 五级流水 CPU 上。此外，我还会参考国内外关于多发射和预取技术的设计实现对本课题的 CPU 进行优化。最终将设计好的 CPU 搭建在中科龙芯所提供的开发板上。

3.3 移植 Linux 操作系统以及编译工具

采用 LA32R 指令集架构所设计的 CPU 兼容 Linux。为了移植操作系统，首先需要创建一个适合于该 3.2 中完成的 CPU 的交叉工具链，以用于编译 Linux 内核和用户应用程序。然后，将 Linux 内核移植到 CPU 上，选择合适的 Bootloader 启动 Linux，并创建一个文件系统以支持操作系统和应用程序的运行。

3.4 运行实际应用以及基准程序进行性能分析

在这一部分，我将会使用宿主机 Linux 系统，并借助 Qt 应用程序开发一款扫雷游戏。为了确保系统调用的一致，我将会使宿主机 Linux 的版本与开发板所移植的 Linux 版本一致。开发完成后，我将打包游戏的 release 版本以便在目标机上进行部署和运行。随后，我将利用 SPEC 基准测试程序以及实际的扫雷游戏，进行深入的性能分析，以评估开发板上的 CPU 性能表现。

4 研究方案

4.1 基于教学合作完成从 MIPS 到 LA 的迁移

1. 按照实验教程学习 Logisim

这一部分研究的主要目标是按照实验要求学习 Logisim 的门电路、组合电路、时序电路，并进行相应的仿真调试。此外还需要掌握 Logisim 中的 RAM、ROM 的使用方法。最后的实验考核标准包括搭建 swap 电路、进行排序电路的组合逻辑设计、斐波那契数列的时序逻辑设计。

2. 使用 Verilog 语言进行部件设计

这一部分需要了解 Verilog 代码规范，然后使用 Verilog 语言完成实验要求的一

系列任务，包括算术逻辑单元 ALU、多路选择器、计数器、以及表达式状态机的设计。

3. 使用基于 LA32R 的汇编语言实现特定功能的函数

在这一部分，首先需要深入学习 LA32R 指令集手册^[5]，熟悉其中的汇编指令助记符、指令格式规则、寄存器命名规则、以及汇编代码规范等内容。随后，将进行一系列的实验，包括内存操作、回文串判断、卷积运算、全排列、高精度阶乘等功能的汇编实现。

4. 实现基于 LA32R 的单周期 CPU

在这一部分的移植工作中，需要参考基于 MIPS 的 22 条指令的单周期 CPU 的数据通路（如下图 4-1 所示），并将其扩展以支持 LA32R 的 54 条指令。具体来说，底层模块，如多路选择器、符号扩展、寄存器等的实现保持不变。所需要改变的只是集成的数据通路和控制器。

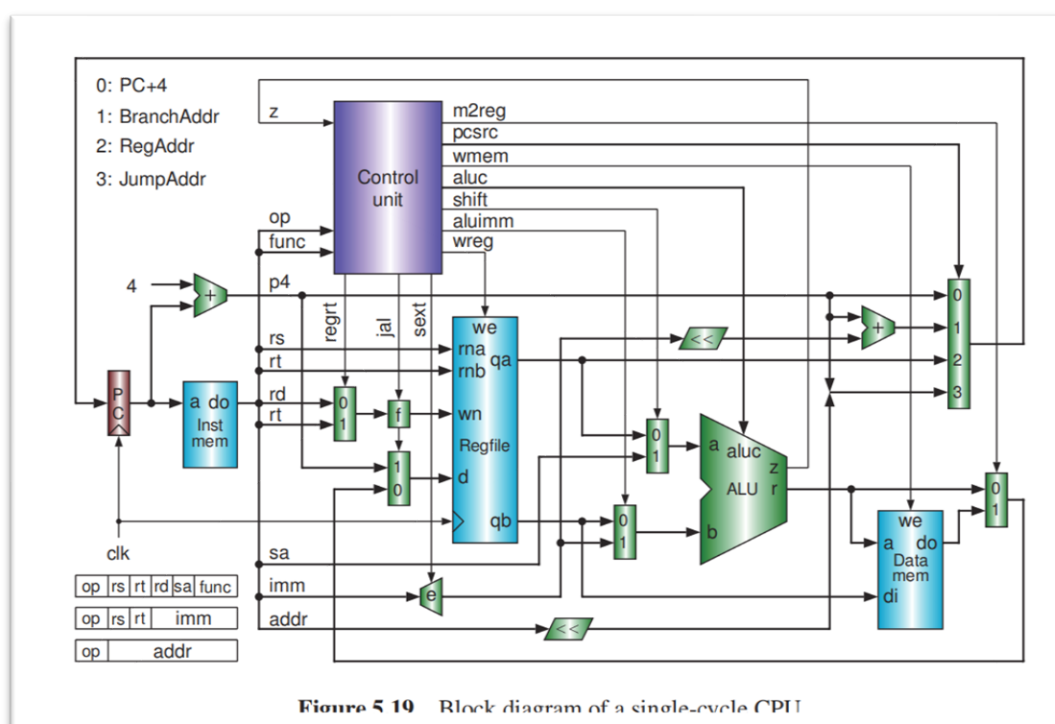


图 4-1 单周期 MIPS 数据通路

5. 实现基于 LA32R 的五级流水线 CPU

在这一部分的工作中，我将根据基于 MIPS 的五级流水 CPU 的设计实现，对之前实现的单周期 LA32R 的流水段进行划分和承接冒险的解决方法。需要注意的是，LA32R 引入了更多的指令，这可能导致更多的冒险情况需要考虑。此外，LA32R 去掉了 MIPS 的延迟槽机制，这也需要在流水线设计和冒险解决中进行相应的调整^[12]。

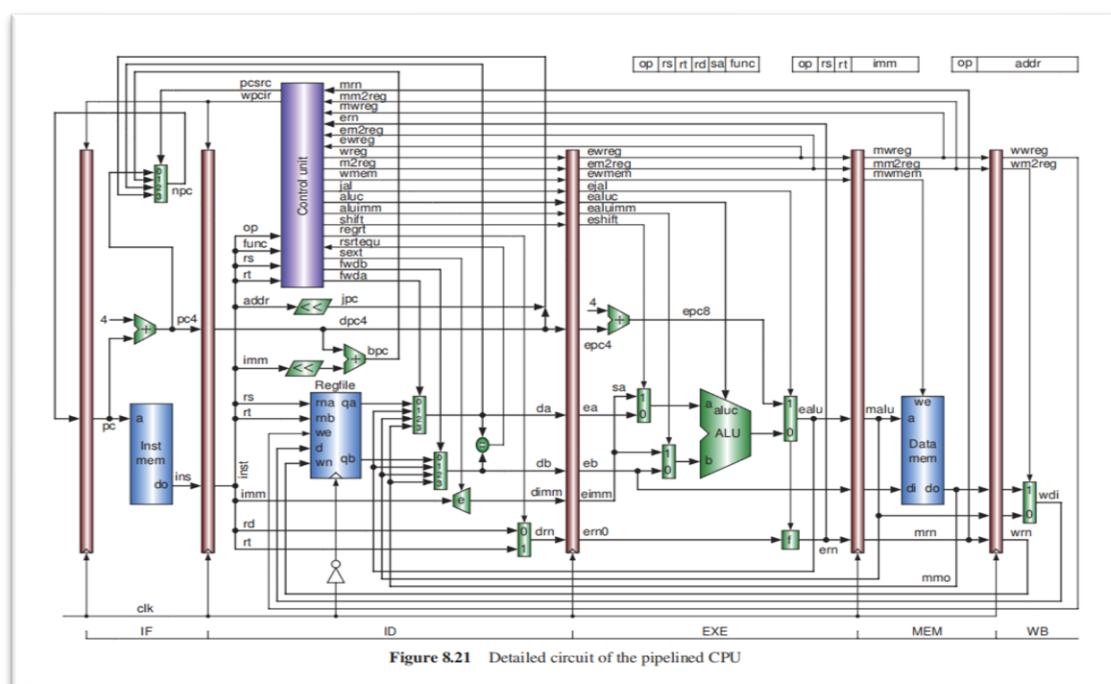


图 4-2 多周期五级流水线 MIPS CPU 数据通路

4.2 对五级流水 CPU 微结构进行设计优化

1. 根据李亚民教授的《计算机原理与设计——Verilog 版》设计 CPU 的中断机制、Cache、TLB 以及多核机制

本课题所实现的中断机制是精确中断，包括外部中断、以及算术溢出、系统调用、未实现的指令这三种内部异常。MIPS 是利用协处理器 CP0 的 Status、EPC、Cause 寄存器来实现中断。其数据通路如下所示：

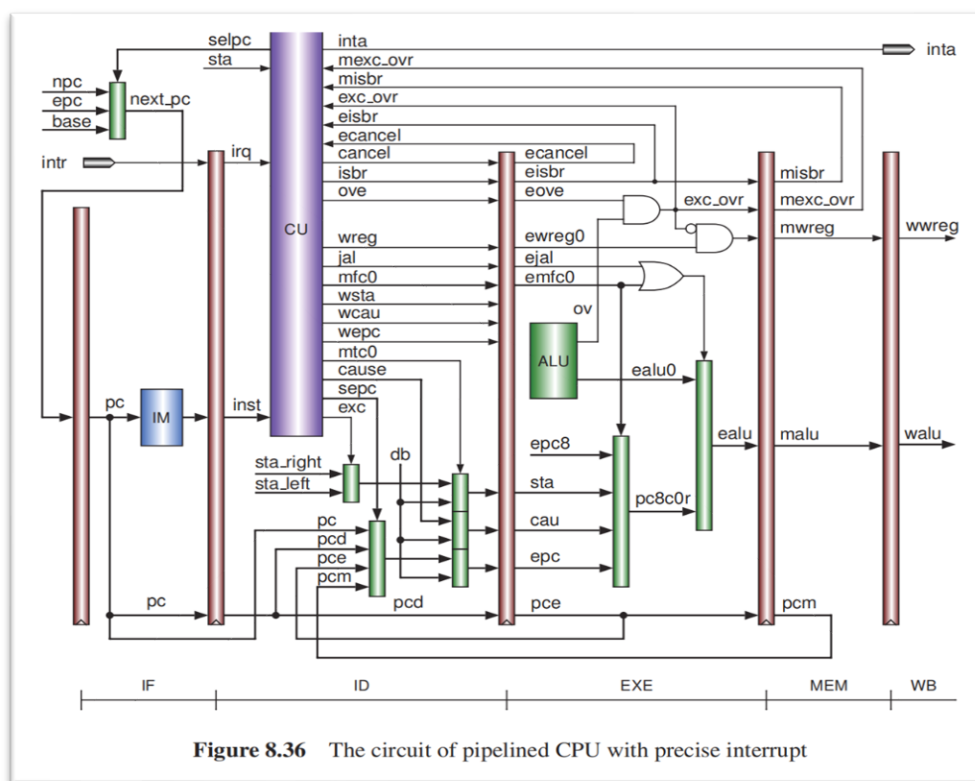


图 4-3 带有中断的流水线 MIPS CPU 数据通路

此外关于 Cache、TLB 和多核等方面的思想以及 MIPS 实现，在该著作中也有详细描述。通过深入学习和理解这些概念和技术，我将能够将它们成功应用到 LA32R 架构上的 CPU 设计中，为其实现相应的功能和性能提升提供支持。

2. 参考国内外以及龙芯杯中对静态双发射、预取的实现，实现 LA32R 的静态多发射、预取机制

静态发射是一种编译阶段技术，它可以将那些不存在冲突的指令组合在一起，合并成一个更长的指令字，通常称为“超长指令字 VILW”。若不存在可以合并的指令，则用空指令 NOP 填充。这种技术的应用在第三届龙芯杯中可以看到，清华大学计算机系团队设计了一个“十级流水线双发射 MIPS 处理器”，并荣获特等奖。该项目采用了静态双发射技术，基于 MIPS 指令集实现，并已在 Github 上进行了开源。本课题将借鉴该项目的经验，以完成静态双发射的实现。此外，为了实现指令预取机制，我们将参考论文《Data prefetch mechanisms》^{[8][13]}中的设计方法和实现技巧。

4.3 移植 Linux 操作系统以及编译工具

1. 了解目标硬件平台的体系结构、处理器类型、内存布局、外设等硬件特性。在本课题中所采用的是 LA32R 指令集架构所实现的支持静态双发射、预取以及 Cache 且带有中断的多核五级流水线 CPU，内存布局和外设具体根据开发板而定。
2. 下载 Linux 内核源码并进行配置。在 Linux 内核官网下载对应稳定版的内核源码，

- 根据目标硬件平台编写.config 配置文件，运行 make config 命令配置内核。
3. 配置交叉编译工具链。交叉编译工具链包括交叉编译器、链接器和其他编译工具，以便将内核源码编译为适用于目标硬件平台的可执行文件。
 4. 添加设备驱动程序。根据本课题所需要使用的硬件外设，添加和配置对应的设备驱动程序，负责与这些外设进行交互
 5. 编译内核源码生成内核映像文件。在配置好内核源码以及设备驱动程序后，使用交叉编译工具链编译内核源码生成可以在目标硬件平台上直接执行的二进制文件，称为内存映射文件
 6. 配置 Bootloader。配置启动加载程序 Bootloader 以便能够正确加载和启动内核。Bootloader 的配置需要指定内核映像文件的位置和名称，并设置启动参数，如命令行参数、设备树文件的位置等。
 7. 烧录内核映像文件。将生成的内核映像文件烧录到目标硬件平台的启动设备上，例如 SD 卡、NAND Flash 等。确保内核映像可以被启动加载程序正确识别和加载。

4.4 运行实际应用以及基准程序进行性能分析

在这个部分，我将利用宿主机上的 Linux 操作系统，使用 Qt 框架编写一个扫雷游戏。编写完成，我们将会打包程序的 Release 版本，以便能够在目标硬件平台的 Linux 文件系统中加载和运行。在运行这个应用程序的过程中，我将会选择一个或多个性能指标，以便衡量目标硬件平台的性能表现。此外，我还计划在目标硬件平台上运行 SPEC 基准程序，这些基准程序可以用来评估处理器性能。通过运行这些基准程序，我将能够更深入地了解目标硬件平台的性能表现，以便进行性能分析。

5 进度安排以及预期达到的目标

5.1 进度安排

表 5-1 进度安排时间表

时间	需要完成的任务
2023.10.25-2023.11.01	搜集和查阅资料与导师交流，确定选题，完成开题报告的撰写
2023.11.01-2023.11.03	将开题报告提交给导师做开题之前的审核，并根据导师意见进行修改
2023.11.04	开题答辩
2023.11.05-2023.11.12	根据答辩组老师意见对毕业设计开题报告进行修改
2023.11.13-2023.11.31	学习 LA32R 指令集手册，并完成研究内容 3.1，实现到 LA 的迁移

2023.12.01- 2024.01.10	学习静态双发射以及预取的设计实现，并在已实现的 CPU 上完成设计
2024.01.12- 2024.02.10	学习怎样在自己搭建的 CPU 上完成操作系统和编译工具的移植并加以实现
2024.02.12- 2024.03.01	使用 C++编写“扫雷”实际应用，并部署在所设计的开发板上
2024.03.05- 2024.03.25	完成性能分析工作
2024.04.01- 2024.05	完善所有功能，并撰写部分毕业论文，进行终期答辩

5.2 预期达到的目标

- (1) 可以在开发板上运行所设计的 CPU
- (2) 根据所移植的系统，运行所设计的“扫雷”程序
- (3) 产出性能分析报告

6 课题已具备和所需的条件、经费

课题已具备的条件：操作系统为 Win11 专业版 22H2、Ubuntu22.04 的个人电脑 LEGION Y700，使用 Vivado、Visual Studio Code、Qt 以及较为丰富的 MIPS CPU 设计经验。

所需的开发板、以及显示屏等硬件设备以及 LA 的指令集手册由龙芯中科提供

课题预计所需经费在 1000~1500RMB 之间

7 研究过程中可能遇到的困难和问题，解决的措施

7.1 可能遇到的困难和问题

1. 在实现从 MIPS 到 LA 的教学合作迁移过程中，可能会涉及到不兼容的指令和体系结构的差异
2. 在进行微结构设计优化过程中，会有一些优化方法，比如静态双发射、预取等技术并未接触过
3. 将 Linux 操作系统移植到 LA 架构可能需要进行大量的系统级编程工作，包括设备驱动、系统调用等。编译工具的支持也是一个问题
4. 使用 Qt 编写实际应用在物联网等课程设计中均已有所接触，难点主要在于性能测试和分析这方面并没有接触过

7.2 解决措施

1. 针对问题 1，可以详细研究 MIPS 和 LA 之间的差异，并且与龙芯中科的老师、学长保持密切联系，获取一定的技术支持。
2. 针对问题 2，通过阅读国内外所发表的一些相关设计论著以及在 RISC-V、MIPS

上所做出的实践经验，启发在 LA 架构上的实现。

3. 针对问题 3，需要借助 Linux 社区和开源工具，参考现有的移植经验和文档。开发和调试必要的设备驱动程序，确保操作系统适应新的架构。如果可能，与 LA 架构相关的编译工具也需要进行开发或适应。
4. 先查阅教程，对性能测试、分析的步骤有大致地了解。然后选择某个关键性能指标使用 SPEC Benchmark 进行性能测试分析

8 主要参考文献

- [1] 未来智库. 计算机处理器专题报告：国产 CPU 发展现状、机遇和前景展望[EB/OL]. 2019/11/07[2023/10/30]
- [2] 张登杰. 计算机多核处理器技术探究[J]. 中国科技纵横,2018,000(21):9-10
- [3] 赵斐, 王志刚, 张佳豪. 静态双发射架构的微处理器设计与性能分析[J]. 计算机工程与应用, 2017, 53(19): 51-54.
- [4] 徐洁,王华,吴晓华,等.浅析 SPEC 基准测试程序集及评价指标[J].实验科学与技术, 2010, 8(6):4.DOI:10.3969/j.issn.1672-4550.2010.06.009.
- [5] 龙芯中科研发部.32 位精简指令集手册 [EB/OL]. 2020/09/17[2023/10/30].
- [6] Patterson, D. A., & Hennessy, J. L. (2017). "Computer Organization and Design: The Hardware/Software Interface." Morgan Kaufmann.
- [7] Wall, D. W. (1989). "Superscalar Microprocessor Design." McGraw-Hill.
- [8] Vanderwiel S P , Lilja D J . Data prefetch mechanisms[J]. Acm Computing Surveys, 2000, 32(2):174-199.
- [9] J. L. Henning, "Spec CPU2000: Measuring CPU Performance in the New Millennium," ACM SIGARCH Computer Architecture News, vol. 28, no. 1, pp. 3-17, 2000.
- [10] Yamin Li,(2011) "Computer Principles and Design in Verilog HDL".Beijing:Tsinghua University Press
- [11] Smith, M. D., & Johnson, M. D. (1997). Out-of-order execution. Computer, 30(2), 54-62.
- [12] Chen Lin, Ma Xiao, Ji Xiang, "FPGA-based LoongArch Five-stage Pipeline CPU".Journal | [J] Journal of Physics: Conference Series. Volume 2450 , Issue 1 . 2023
- [13] Qadri Y M , Qadri N N , Fleury M, et al. Software-Controlled Instruction Prefetch Buffering for Low-End Processors[J]. Journal of Circuits, Systems and Computers, 2015, 24(10).
- [14] 徐志伟.龙架构：一种开放自主指令集架构的实践[J].计算机研究与发展, 2023, 60(01):1.