

# 基于 RISC-V 的服务器管理控制器 FPGA 原型设计<sup>①</sup>



李拓<sup>1,2</sup>, 邹晓峰<sup>1,2</sup>, 林宁亚<sup>1,2</sup>, 张璐<sup>1,2</sup>, 刘同强<sup>1,2</sup>, 周玉龙<sup>1,2</sup>, 李仁刚<sup>1,2</sup>

<sup>1</sup>(高效能服务器和存储技术国家重点实验室, 济南 250013)

<sup>2</sup>(浪潮电子信息产业股份有限公司, 济南 250013)

通讯作者: 邹晓峰, E-mail: zouxif@inspur.com

**摘要:** 服务器管理控制器是云计算装备关键部件之一, 目前主要基于 ARM 架构开发, ARM 较高的授权费推高了控制器设计成本, 不利于 SoC 相关产品的迭代和升级. RISC-V 是近年提出的一种开源的处理器架构, 与 ARM 同属精简指令集, 具有模块化、可扩展等诸多特点. 本文采用 RISC-V 开源处理器 BOOM 核心, 设计实现了一种基于 RISC-V 处理器的服务器管理控制器 FPGA 原型系统. 该系统基于 Xilinx 的 Virtex Ultra Scale 440 FPGA 进行了原型构建, 完成了实际应用场景下的功能测试和 CoreMark 测试, 结果显示处理器性能提升了 26%, 优于同级别的 ARM 核心, 系统功能符合设计预期. 此外, 该原型系统基于 OpenBMC 实现了 IPMI 等专用管理控制协议, 基本功能验证通过, 证明了通过 RISC-V 替换 ARM 优化 SoC 架构的可行性.

**关键词:** RISC-V; 服务器; 管理控制器

引用格式: 李拓, 邹晓峰, 林宁亚, 张璐, 刘同强, 周玉龙, 李仁刚. 基于 RISC-V 的服务器管理控制器 FPGA 原型设计. 计算机系统应用, 2021, 30(7): 136-141. <http://www.c-s-a.org.cn/1003-3254/8009.html>

## Design of FPGA Prototype of Server Management Controller Based on RISC-V

LI Tuo<sup>1,2</sup>, ZOU Xiao-Feng<sup>1,2</sup>, LIN Ning-Ya<sup>1,2</sup>, ZHANG Lu<sup>1,2</sup>, LIU Tong-Qiang<sup>1,2</sup>, ZHOU Yu-Long<sup>1,2</sup>, LI Ren-Gang<sup>1,2</sup>

<sup>1</sup>(State Key Laboratory of High-End Server & Storage Technology, Jinan 250013, China)

<sup>2</sup>(Inspur Electronic Information Industry Co. Ltd., Jinan 250013, China)

**Abstract:** The server management controller, as a key component of cloud computing equipment, is mainly based on ARM architecture at present. However, ARM's high licensing fees push up its design cost, which is not conducive to the iteration and upgrade of SoC-related products. RISC-V, which is an open-source processor architecture proposed recently, belongs to the reduced instruction set as ARM. It has many features such as modularity and scalability. In this study, we combine the BOOM core of a RISC-V open-source processor to design and implement an FPGA prototype system of a server management controller based on a RISC-V processor. In the system, we build a prototype based on Xilinx's Virtex Ultra Scale 440 FPGA and complete the functional and CoreMark tests in the actual application scenarios. The results show that the processor performance is improved by 26%, which is superior to that of products of the same level that use ARM as the core, and the system's functions are in line with design expectations. In addition, based on OpenBMC, this system implements special management control protocols such as IPMI, and its basic functions are verified, which proves the feasibility of optimizing SoC architecture by replacing ARM with RISC-V.

**Key words:** RISC-V; server; management controller

① 基金项目: 山东省重大创新工程项目 (2019TSLH0125)

Foundation item: Major Innovative Project of Shandong Province (2019TSLH0125)

收稿时间: 2020-10-27; 修改时间: 2020-11-23; 采用时间: 2020-12-18; csa 在线出版时间: 2021-06-30

指令集,是在CPU设计初期规定,用于控制处理器内部运行的一系列指令的集合.指令集设计的优劣,对CPU的性能有重要影响.RISC-V指令集<sup>[1]</sup>是2014年发布的一套完全开源的精简指令集(RISC),始于加州大学伯克利分校的相关项目.凭借前瞻性的设计与后发优势,RISC-V成为了开放指令集架构领域的佼佼者.

图灵奖得主 John Hennessy 曾经就未来的芯片设计提出了一种 DSA (Domain Specific Architecture) 的概念,即针对应用领域做优化的处理器架构.ARM 作为一种通用且固化的指令集架构,设计灵活性有限,而 RISC-V 架构不仅仅开源,同时还具有模块化、可扩展的特点,支持用户自定义指令.

服务器管理控制器是应用于服务器等数据中心基础设施的一种专用 SoC,主要功能包括对系统设备进行信息监测、记录以及恢复等,它以独立系统运行,通过网络还可以实现对服务器的远程访问、远程监控和管理<sup>[2]</sup>.目前,服务器管理控制器主要基于 ARM 架构处理器进行设计,尚未有公开的基于 RISC-V 设计方案.本文从 SoC 设计底层入手,基于 RISC-V 架构处理器进行服务器管理控制器的设计和测试,探究方案可行性并与 ARM 架构进行对比.

本文首先介绍了 RISC-V 及开源乱序执行处理器 BOOM<sup>[3,4]</sup>,然后详细描述了基于该处理器实现服务器管理控制器 FPGA 原型系统的过程.该原型系统移植了 OpenBMC 开源组件,实现了智能平台管理接口规范 (Intelligent Platform Management Interface, IPMI) 协议的基本功能,同时还介绍了 LXDE 等定制化组件.本文主要工作是验证了 RISC-V 替代 ARM 设计服务器管理控制器的可行性和性能水平.

## 1 RISC-V 与 BOOM 介绍

自 RISC-V 发布以来,RISC-V 的工具链与产品越发多样,生态逐步完善.2015年,非盈利组织 RISC-V 基金会 (RISC-V Foundation) 成立,着手构建开放、合作的软硬件交流平台,推动 RISC-V 生态的快速完善与发展.很快就有几百家加入了基金会,其中包括谷歌、IBM、英伟达、浪潮等国际企业以及加州大学伯克利分校、麻省理工学院、普林斯顿大学、中科院计算所等学术机构.

在国内,芯来科技于2017年10月开源了其“蜂

鸟”低功耗处理器,这是国内由企业开源的第一款基于 RISC-V 架构的处理器.目前在中国已经有多家成熟的 RISC-V CPU 企业,如睿思芯科、平头哥等.在物联网、智能家电等领域,RISC-V 相关产品已经有大量应用案例.目前,基于 RISC-V 架构的开源处理器已经涵盖嵌入式、AI、5G 射频、硬盘控制器等多种应用场景.处理器核心也由最初的低功耗标量处理器逐渐向乱序超标量的复杂架构演进<sup>[5]</sup>.产业环节中的各种相关工具如 GCC/LLVM 编译器、Linux 内核支持、高层次设计语言、总线通讯协议等也在不断完善.

在2015年,伯克利分校的研究团队推出了基于 RISC-V 架构的乱序执行核心 BOOM (Berkeley Out-of-Order Machine),并将其完全开源.这款 CPU 使用 Chisel 语言编写,指令集为 RISC-V 中的 RV64G 指令和特权指令,该核心是一款 64 位 RISC-V 超标量 Berkeley 乱序机的源存储库,计算性能与 ARM 中高端处理器相当,经过多次流片,实际性能表现优异.

BOOM 的流水线结构主要分为 6 个阶段:取指、译码/重命名/指令调度、发射/读寄存器、执行、访存以及写回.由于 Chisel 是一种面向硬件的高层次设计语言,BOOM 处理器可以实现参数的灵活配置,包括:取指宽度、译码宽度、指令发射宽度、物理寄存器堆的大小、有序发射或无序发射;L1 Cache 及 L2 Cache 的路数等参数. BOOM 流水线结构<sup>[3]</sup>如图 1 所示.

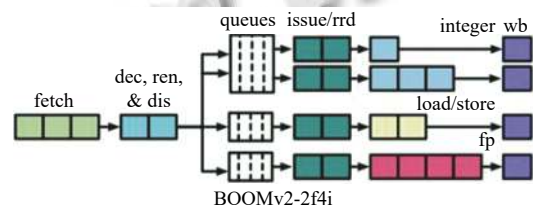


图 1 BOOM 处理器流水线结构

## 2 控制器原型设计

常见的服务器管理控制器的基本架构如图 2 所示.本文依托 RISC-V 开源生态环境,设计实现了管理控制器基本的外设和接口模块,包括:

- 1) I2C/SMBus: 用于实现与 CPLD 的通信和获取温度、电压等传感器回传的数据;
- 2) LPC: BIOS 与监控管理系统之间通信, BIOS 向监控管理系统上报启动信息;
- 3) KVM: 视频、鼠标和键盘的控制信息重定向到

监控管理系统;

- 4) SPI: BIOS 固件的升级更新接口;
- 5) JTAG: CPLD 的固件更新接口;
- 6) IPMB: ME (Management Engine) 与监控管理系统之间的通信;
- 7) Ethernet: 监控管理系统接入网络接口;
- 8) VGA: 实现视频流的显示;
- 9) PWM: 实现风扇转速的控制和监控;
- 10) USB: 鼠标、键盘等 USB 设备接口;
- 11) Timer、RTC、WDT 等基础模块: 实现复位、实时时钟等。

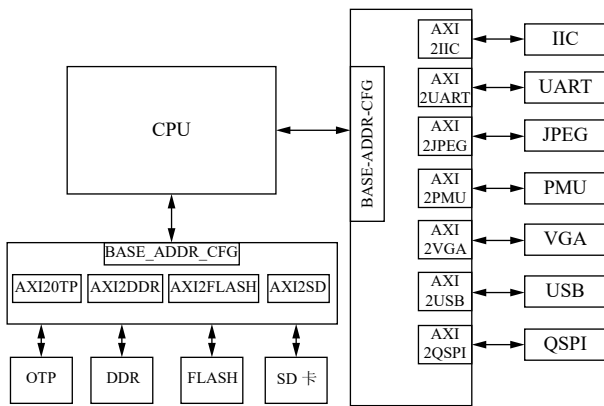


图2 通用 RISC-V 原型系统架构

RISC-V 控制器 FPGA 原型平台处理器选型工作中, 我们对比了开源的 Rocket<sup>[6]</sup>、BOOM、PULPino<sup>[7]</sup> 等, 通过公开的性能数据分析, 选用性能最优, 并经流片验证的 BOOM V2 核心, 并基于 BOOM V2 SoC 开发环境进行开发。将上述外设、接口模块与处理器核心进行集成后, 对系统地址空间进行了划分, 同时, 利用 RISC-V 的工具链对 OpenBMC 系统进行重新编译和文件系统移植。

本文主要从 4 个方面对 FPGA 原型平台进行了功能扩展和优化:

- 1) 原型系统主处理器实现了 BOOM 双核, 嵌入 L2 Cache 和一致性协议总线, 实现核间 Cache 一致性。实际应用可以对核数配置, 满足不同场景性能需求。
- 2) 针对 BOOM 总线进行了扩展优化, 增加对 AXI 总线的扩展支持, 并基于该总线扩展了 DDR 等外设接口。
- 3) 鉴于 RISC-V 指令集的灵活性, 对指令集进行了向量指令扩展<sup>[8]</sup> 的探索。RVGC64 指令集目前已经满足 SoC 设计需求, 扩展的向量指令主要目的是探究

RISC-V 扩展指令集开发流程, 为其他高性能应用场景提供支持。

4) DRAM 容量达到 8 GB, 系统启动时, 通过 DDR 将固件直接加载到 DRAM 中, 加快固件加载和系统启动速度。

### 2.1 基于 RISC-V 的控制器系统结构

服务器管理控制器主要利用传感器监控计算机或服务器的运行状态<sup>[9]</sup>, 具有本地和远程诊断、控制台支持, 配置管理, 硬件管理和故障排除等功能。

我们结合服务器管理控制器实际应用场景, 进行了 RISC-V SoC 的定向设计与优化, 控制器结构如图 3 所示。SoC 采用双核 BOOM 作为主处理器, 集成服务器管理控制器所需 IIC、LPC、QSPI 等外设接口, 并支持服务器监控管理协议 (IPMI), 实现与控制器上层接口的完全兼容。

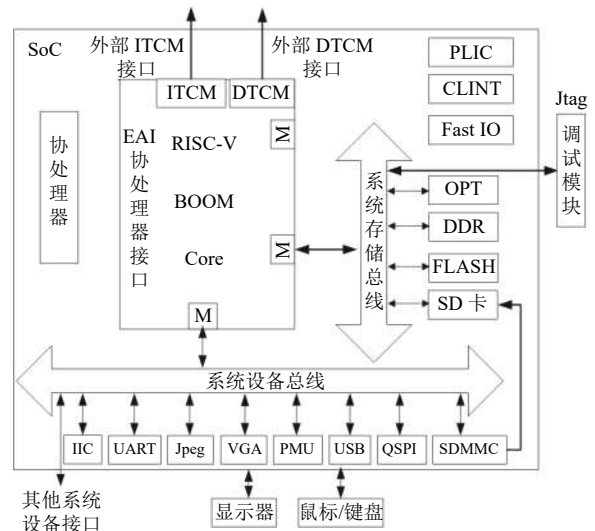


图3 特定场景优化后架构

### 2.2 系统总线

BOOM SoC 使用的是 TileLink 总线协议, 该总线是加州大学伯克利分校设计的一种高速片上总线协议<sup>[10]</sup>, 我们根据需求配置设计了 TileLink 转 AXI4 协议转换模块, 其总线互连和转换功能集中在 SystemBus 模块中, 详细功能模块划分如图 4 所示。

TileLink 设计之初, 考虑了满足支持缓存一致性 (cache coherence) 协议, 并针对现有的多种一致性协议, 其将总线设计与协议设计相分离, 提高了 TileLink 总线的适应性。

TileLink 总线的有向无环图结构主要是作为处理



器的 BoomTile, 可外接测控模块的 FrontBus 接口与 MMIO\_AXI4 的外设接口、PeripheryBus 外设接口、TLError 和 TLBroadcast 的数据和控制交互. 其中主要包含了两部分功能:

1) TileLink 网络拓扑实现, 包含了多主多从接口的 Crossbar Agent、一从多主的 Crossbar Agent 和主从转换接口的 Cache Agent 实现.

2) TileLink 协议转 AXI4 协议功能实现.

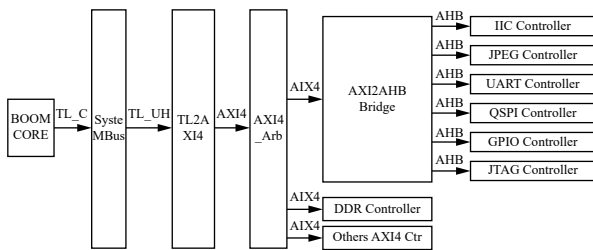


图4 总线设计架构框图

### 2.3 地址空间与外设

根据服务器管理控制器外设需求, 我们对内部地址空间进行了划分, 如表 1 所示.

表 1 地址空间划分

地址空间	地址划分
0x8000_0000	SPI2 flash (256 MB)
0x5000_0000	SPI1 flash (256 MB)
0x3000_0000~0x3000_7FFF	RTC、Watchdog 等
0x3000_8000~0x3000_8FFF	IIC 控制器模块
0x3000_9000~0x3000_9FFF	JPEG 控制器模块
0x3001_0000~0x3001_0FFF	UART 控制器模块
0x3002_0000~0x3000_2FFF	QSPI 控制器模块
0x3003_0000~0x3000_3FFF	GPIO 控制器模块
0x3004_0000~0x3000_4FFF	DDR 控制器模块

### 2.4 固件与 Linux 内核

RISC-V 内核启动过程中分为 4 部分: ZSBL、FSBL、BBL 和 Linux Core. 在启动过程中, 我们利用了 Register 存储空间的划分, 通过配置参数指定了各自的起始存储位置和大小, 启动程序可以读取并分析 Register 存储空间的配置参数, 获得 FSBL、BBL 和 Linux Core 在 SD 卡的存储位置和大小, 然后将其搬运到指定的 DDR 中运行.

基于改进的 RISC-V 内核启动流程, 我们实现了对 SD 卡的坏块进行管理的设计. 通过 Register 配置空间可以任意指定 ZSBL、FSBL、BBL 和 Linux Core 的存储起始位置和大小, 如果 SD 卡某一块出现坏块,

可以通过配置 Register 配置空间的参数修改存储位置, 跳过坏块部分, 后期可以将 Register 配置空间的参数通过寄存器的方式存储在 RISC-V 处理器中, 可以在 RISC-V 处理器启动前通过外部进行配置启动参数. 上述坏块管理操作可以大大提高 SD 卡的可用性.

针对服务器管理控制器在人机交互上的便捷性需求, 我们移植了 LXDE 作为简洁易用的交互界面. LXDE 是一款面向低功耗的轻量级桌面环境, 支持 GTK 特性, 用户界面更贴近 Microsoft Windows 的菜单. 其在设计之初就考虑到了用户体验性与较低的资源占用, 注重模块化设计, 组件可以灵活增删, 满足了定制化需求, 也实现了以极低的内存与 CPU 负载承担功能丰富的桌面环境. 以上优点同时使其更容易移植在服务器管理控制器的系统上, 而且在低运行频率的 FPGA 原型平台上就可以进行安装调试.

在应用层, 我们集成了 OpenBMC. OpenBMC 由微软, 英特尔, IBM, 谷歌和 Facebook 发起, 是一个开源的软件架构, 其功能是用来打造一个完整可用的负责基板管理控制的 Linux 代码库. OpenBMC 使用 Yocto Project 作为发布框架, 为移植提供了便利.

移植后使用浪潮自有的 UI 界面管理, 如图 5 所示.



图5 服务器管理界面图

随着对 OpenBMC 的持续优化, 其已经能够实现带外的在线升级, 在不占用服务器网络带宽的情况下, 通过 Web UI 界面和软件包实现固件的刷新和维护工作. 通过提升 OpenBMC 的模块化、规范化程度, 将来可以为各类数据中心, 尤其是大规模、超大规模数据中心, 提供下一代开放技术管理方案.

### 3 控制器功能验证方法

Boom-template 工程集成了 Verilator 仿真环境. 在 versim 目录下执行 make 命令, 可以生成有关 BOOM 的 simulator, 该 simulator 为一可执行文件, 名称为

simulator-boom-system-BoomConfig, 可以执行任何 RV64G 的 ISA 测试, 包括单一测试用例以及整个测试集.

仿真环境的组成如图 6, 其中 MMIO AXI4 和 MEM AXI4 接口分别接两块 mem, DMI\_debug port 接 SimDTM 模块, 该模块负责产生仿真时的激励.

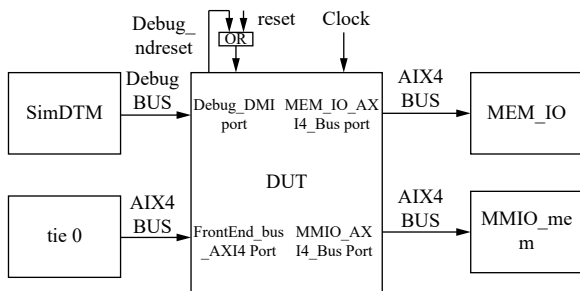


图 6 逻辑仿真示意图

SimDTM 模块提供了 C++ 与 Verilog 的接口, 由 C++ 产生激励, 并通过驱动 DMI\_debug bus 上的信号来驱动 DUT, 从而完成仿真.

## 4 原型实现与验证

### 4.1 工程实现

该系统基于 Xilinx 的 Virtex Ultra Scale 440 FPGA 进行了原型构建, 然后将整套系统从 VU440 移植到使用了 Kintex-7 系列的 FPGA 定制板卡上. Kintex-7 系列 FPGA 实现了最佳成本/性能/功耗平衡, 芯片功能强大, 资源丰富. 其 480T 型号 FPGA 支持 8 通道 PCIE 接口, 提供了 478K 逻辑单元, 高达 400 个 I/O 口, 完全满足原型验证需求. 测试使用 SD 卡启动的方式实现逻辑代码的加载.

我们为 FPGA 原型系统开发了服务器管理控制器相关的外设模块, 如 UART、Timer、SPI、外部存储、调试逻辑等, 结构如图 7 所示.

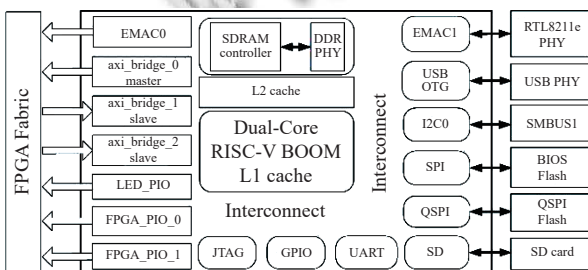


图 7 外围模块示意图

服务器管理控制器场景下 RISC-V SoC 的实现分两步: (1) 基于 BOOM 多核的原型平台, 针对服务器管

理控制器的各项需求进行扩展和优化, 集成外设接口, 进行稳定性和功能测试, 作为硬件研究和软件开发的基础平台. (2) 将整套系统从 VU440 移植到 FPGA 定制板卡上, 如图 8 所示. 该定制板卡使用了 Kintex-7 系列 XC7K480T 芯片, 集成网口、串口等用于通信、调试的接口, 可通过 PCIE 接口连接服务器主板, 借此可以顺利进行上层移植以及功能性能测试.



(a) XCVU440 原型板卡



(b) Kintex-7 定制板卡

图 8 测试板卡

### 4.2 FPGA 原型调试

当设计从模拟实现到硬件实现时, 用户对系统当前状态的控制和理解会急剧下降. 为了帮助启动和调试低级软件和硬件, 在硬件中内置良好的调试支持至关重要. 当 core 运行操作系统时, 软件可以处理许多调试任务. 但是, 在许多情况下, 硬件支持至关重要. 我们重新制作了根文件系统以及相关组件, 增加了更多便于调试的命令, 启动界面如图 9 所示.

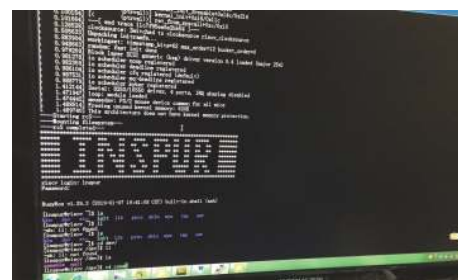


图 9 系统启动图

Debug 系统 debug host 和 RISC-V 平台组成, 其中 debug host 包括运行调试器的调试主机 (如笔记本电脑), 调试器经常使用的如 GDB 工具, 调试器通过调试转换器与调试传输硬件进行通信, 调试转换器模块可以使用 OpenOCD, 其可以包括硬件驱动器, DTM 为调试传输硬件, 即 USB-JTAG 适配器调试传输硬件将调试主机连接到平台的调试传输模块 (DTM). DTM 使用调试模块接口 (DMI) 提供对一个或多个调试模块 (DM) 的访问。

平台中的每个硬件线程都由一个 DM 控制. 硬件线程可能是异质的. 对硬件线程与 DM 之间的映射没有限制, 但通常单个 core 中的所有硬件线程都由相同的 DM 控制. 在大多数平台上, 只有一个 DM 可以控制平台中的所有硬件线程。

Program Buffer 允许调试器在硬件线程上执行任意指令. 这种机制也可用于访问内存. 可选的系统总线访问块允许在不使用 RISC-V Hart 来执行访问的情况下进行存储器访问。

每个 RISC-V 可以实现触发模块 (Trigger Module, TM). 当满足触发条件时, 硬件线程将停止并通知调试模块它们已停止。

DMI 连接调试模块 DM 和 DTM, 其中 DTM 是 master, DM 作为 slave. 调试模块接口可以是具有一个主设备和一个从设备的普通总线, 或者是更全功能的总线, 如 TileLink 或 AMBA 高级外设总线. DMI 使用 7 到 32 个地址位, 支持读写操作. 地址空间的开始段用于第一个 DM, 额外空间可用于自定义调试设备、其他内核、附加 DM 等. 如果此 DMI 上有其他 DM, 则 DMI 地址空间中下一个 DM 的基址将在 nextdm 中给出. 调试模块通过对其 DMI 地址空间的寄存器访问来进行操作。

### 4.3 性能测试

在使用 RISC-V BOOM 搭建的服务器管理控制器 FPGA 原型系统上, 应用层的程序均可正常运行, 与上一代 ARM 架构原型系统相比, 该系统启动时间缩短 15%。

我们对服务器管理控制器 FPGA 原型系统进行了软硬件测试和上层应用适配, 包括软件兼容性、系统稳定性、IO 时延等参数, 优化后的系统可满足产品化的各项指标. 在进行各方面的适配后, 经 Benchmarks 基准测试, 系统性能测试分数超过 ARM Cortex-A9, 具体性能如表 2 所示。

表 2 RISC-V Core 与 ARM Cortex-A9 跑分对照

地址空间	Dual-BOOM	Arm Cortex-A9 <sup>[9]</sup>
Core-mark	4.53	3.59
Ratio	1.26	1

## 5 结语

本文首先对 RISC-V 的 BOOM 处理器进行了相关的扩展和优化, 并集成了服务器管理控制器所需外设, 移植了开源的上层应用框架, 构建了一套基于 RISC-V 处理器核的服务器管理控制器原型系统, 并进行了 Benchmarks 测试. 测试结果显示, 该系统处理性能优于 ARM Cortex-A9 处理器 26%. 结果证明了 RISC-V 在面向具体应用场景上对 ARM 替代可行性。

### 参考文献

- Waterman A, Lee Y, Patterson D, *et al.* The RISC-V instruction set manual, Volume I: Base user-level ISA. Berkeley: EECS Department, University of California, 2011.
- 乔英良. 刀片服务器监控管理系统设计与实现 [硕士学位论文]. 济南: 山东大学, 2012.
- Celio C, Chiu PF, Nikolic B, *et al.* BOOM v2: An open-source out-of-order RISC-V core. Berkeley: EECS Department, University of California, 2017.
- Celio C, Chiu PF, Asanović K, *et al.* BROOM: An open-source out-of-order processor with resilient low-voltage operation in 28-nm CMOS. IEEE Micro, 2019, 39(2): 52–60. [doi: 10.1109/MM.2019.2897782]
- 雷思磊. RISC-V 架构的开源处理器及 SoC 研究综述. 单片机与嵌入式系统应用, 2017, 17(2): 56–60, 76.
- 杨维科, 贺光辉, 景乃锋. 基于 Rocket-Chip 开源处理器的 CNN 加速模块的设计及实现. 微电子学与计算机, 2018, 35(4): 17–21.
- Rossi D, Conti F, Marongiu A, *et al.* PULP: A parallel ultra low power platform for next generation IoT applications. 2015 IEEE Hot Chips 27 Symposium (HCS). Cupertino, CA, USA. 2015. 1–39.
- Collange C. Simty: Generalized SIMT execution on RISC-V. Proceeding of First Workshop on Computer Architecture Research with RISC-V (CARRV 2017). Boston, MA, USA. 2017.
- 苏振宇. 基于国产 BMC 的服务器安全启动技术研究与实现. 信息安全研究, 2017, 3(9): 823–831. [doi: 10.3969/j.issn.2096-1057.2017.09.008]
- Celio C, Patterson D, Asanović K. The Berkeley Out-Of-Order Machine (BOOM): An industry-competitive, synthesizable, parameterized RISC-V processor. Berkeley: University of California, 2015.