

引用格式:李介民,张善从. 五级流水线 RISC-V 处理器的研究与性能优化[J]. 微电子学与计算机, 2022, 39(3): 78-85. [LI J M, ZHANG S C. Research and performance optimization of five stage pipelined RISC-V processor [J]. Microelectronics & Computer, 2022, 39(3): 78-85.] DOI:10.19304/J. ISSN1000-7180. 2021. 0856

五级流水线 RISC-V 处理器的研究与性能优化

李介民^{1,2,3}, 张善从^{2,3}

(1 中国科学院大学 北京 100049;

2 中国科学院 空间应用工程与技术中心 北京 100094;

3 中国科学院 太空应用重点实验室 北京 100094)

摘 要: 国内基于 RISC-V 指令集的嵌入式处理器的研究在近几年内得到了快速发展. 在性能评估研究上多集中于 2-3 级流水的小规模、低功耗处理器, 针对 5 级流水架构处理器的性能量化研究较少. 针对该问题, 在传统 5 级顺序流水架构的基础上, 分别从 RISC-V 指令预测、流水线机制、乘除法算法、存储架构等方面分析处理器优化的策略. 针对不同策略优化的处理器以 AHB 片上互联、APB 桥接的方案实现外部模块的搭载. 在 FPGA 上完成软硬件协同仿真验证, 并在 Xilinx 的 XC7K325T 开发板上运行性能评估测试程序 CoreMark, 依据运行结果着重分析不同静态预测算法、不同周期乘除法运算、外挂存储的容量与设计等因素对处理器的影响. 最终实现的处理器基于 RV32IMZicsr 架构, 采用了半静态预测(资源优化)、流水线检测转发机制(处理优化)、短周期乘除法(计算优化)、最优存储架构(取指与访存优化)等性能优化策略. CoreMark 跑分达到 3.06 CoreMark/MHz.

关键词: RISC-V; 5 级流水线; 性能; CoreMark; 存储架构; AHB

中图分类号: TP332

文献标识码: A

文章编号: 1000-7180(2022)03-0078-08

Research and performance optimization of five stage pipelined RISC-V processor

LI Jiemin^{1,2,3}, ZHANG Shancong^{2,3}

(1 University of Chinese Academy of Sciences, Beijing 100049, China;

2 Technology and Engineering Centre for Space Utilization, Chinese Academy of Sciences, Beijing 100094, China;

3 Key Laboratory of Space Utilization, Chinese Academy of Sciences, Beijing 100094, China)

Abstract: In recent years, the research of embedded processor based on RISC-V instruction set has developed rapidly in China. The research on performance evaluation is mostly focused on the small scale and low power processors with 2-3 level pipelined architectures, and there are few studies on the performance quantification of the processors with 5-level pipelined architectures. To solve this problem, based on the traditional five-level sequential pipelining architecture, the strategy of processor optimization is analyzed from the aspects of RISC-V instruction prediction, pipelining mechanism, multiplication and division algorithm, storage architecture, etc. For the processors optimized for different strategies, AHB on-chip interconnection and APB-bridge are used to carry external modules. The software and hardware co-simulation verification is completed on FPGA, and the performance evaluation test program Coremark is run on Xilinx's XC7K325T development board. According to the running results, the influences of different static prediction algorithms, different cycle multiplication and division operations, capacity and design of plug-in storage on the processor are analyzed emphatically. The final processor is based on

收稿日期: 2021-06-09; 修回日期: 2021-07-21

基金项目: 中国科学院空间应用工程与技术中心所长基金(Y8031931QY)

RV32IMZicsr architecture, and adopts performance optimization strategies such as semi-static prediction (resource optimization), pipeline detection and forwarding mechanism (processing optimization), short-period multiplication-division method (computing optimization), and optimal storage architecture (pointing and memory access optimization). Coremark scored 3.06 Coremark /MHz.

Key words: RISC-V; 5-stage pipelined; performance; CoreMark; Storage architecture; AHB

1 引言

嵌入式微处理器以其性能高、功耗低的特点在物联网行业得到了广泛应用。其中以 STM32 为代表的 ARM 系列产品就占领了嵌入式产品的大部分市场。ARM 公司推出的 AMBA 片上总线协议^[1], 推动了基于 IP 核的 SoC 设计方法, 极大程度实现 IP 共享, 提高总线传输效率^[2]。

RISC-V 指令集作为新兴架构的出现, 以其模块化、低功耗、高性能、易扩展的特点表现出极强的竞争优势^[3]。国外开源项目有 Rocket^[4]、BOOM^[5]、SweRV EH1^[6] 为代表, 国内 RISC-V 代表的有无剑^[7]、蜂鸟 E203^[8]。纵观国内的研究, 针对 RISC-V 处理器的研究多集中于 2-3 级流水线的设计^[9-11], 5 级流水线的案例^[12-13] 较少, 大多数的设计仅考虑实现, 对于性能的评估略有不足。

本文以 AHB 总线为片上互联方案, 以 RISC-V 指令集(RV32IMZicsr)为基础, 采用经典 5 级流水线架构实现 RISC-V 处理器。以 Coremark 测试集为性能评估依据, 针对不同的优化方法量化分析性能的提

升。最后, 本处理器核在 FPGA 上完成了板级验证, CoreMark 测试性能为 3.06 CoreMark/MHz。

2 RISC-V 指令集

RISC-V 是典型的三操作数、加载-存储型架构^[14], 其指令集包括基本指令集(RV32I、RV32E、RV64I、RV128I, 其中 RV32E 是 RV32I 的一个子集)和多个扩展指令集(M-乘除法、A-原子操作、F-单精度浮点运算、D-双精度浮点运算、C-压缩等扩展), 提供了内部寄存器 CSR(也称之为 Zicsr 扩展)^[15]用以实现多级别(用户级 U、机器级 M、监管级 S)的处理器控制功能。本设计选择 RV32I 指令集为基本指令集, 并在此基础上实现 M、机器级 CSR 的 Zicsr 扩展(用于配置或记录一些运行的状态^[16].)。

3 流水线设计

本设计采用经典 5 级流水线架构, 包括取指、译码、执行、访存、写回, 具有顺序取指, 顺序写回的特点, 图 1 给出本设计的 5 级流水线架构。

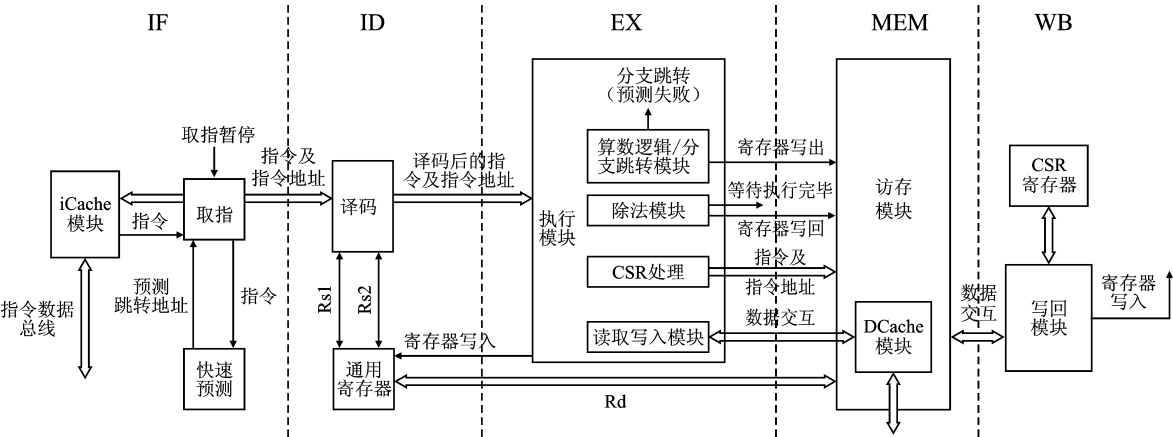


图 1 5 级顺序流水线设计图

Fig. 1 5-stage sequential pipelining architecture design

取指模块确定下一条需要进入流水的指令。如果流水出现暂停(由于数据相关性、内部错误引起的暂停), 则停止取指; 如果流水出现跳转(中断或者跳转指令), 则取指地址为跳转地址; 否则, 取指模块以

每周期 PC 指针加 4 的频率进行取指。
译码模块主要将取指模块的指令与地址按照 RISC-V 的编码格式进行译码。本设计中一共实现了 58 条 RISC-V 指令, 译码阶段对各类指令进行分

赖于 0000 执行完成,0008 的执行结果依赖于 0004 和 0000 的完成. 如果不等待上一条指令执行完成,下一条指令的处理就会出现错误. 但是如果每次数据相关都采取等待操作,随着指令流的增加,会大大影响处理器的性能. 因此,结合 5 级流水线的特点,设计了检测转发机制.

在每条指令的执行阶段进行一个检测,查看本条指令的源寄存器与上两条指令的目标寄存器是否相同,如果相同,则把处于 MEM 或者 WB 阶段的上一条或者上两条的指令的数据转发过来,在执行阶段进行优先级判断(MEN 优于 WB 优于 EX),决定最终进入处理的是原始寄存器内的数据还是转发过来的新数据.

对于目标寄存器 rs1 的执行代码如下所示:

```
if((mem_rd_reg_write_i==1'b1)&&(mem_rd_reg_addr_i==rs1_addr_i))
rs1_forwarded_r = mem_rd_reg_data_i;
else if((wb_rd_reg_write_i==1'b1)&&(wb_rd_reg_addr_i==rs1_addr_i))
rs1_forwarded_r = wb_rd_reg_data_i;
else rs1_forwarded_r = rs1_data_i;
```

同时,如果检测到本阶段处理指令为 LOAD 指令时,将 MEM 阶段的源寄存器地址与 WB 阶段的目标寄存器地址送入执行阶段,如果发现为同一个寄存器时,则暂停一个周期. 等到该条指令从存储器中取到数据后,再进行下一步操作.

4.3 乘除法模块

本设计支持 RISC-V 的 M 扩展,因此实现了 RV32M 指令,包括有无符号数乘法、除法、求余. 乘法支持 32 周期的移位法和单周期的硬件乘法两种模式. 除法支持 32 周期的辗转相除法和基于 Radix2 算法的 4 周期除法两种模式. 32 周期的辗转相除法的实现是在译码阶段得到计算类型是否为有符号数,先按照有符号的除法计算,通过 32 周期的试商得到运算结果,最后根据符号位得到最终计算结果. 基于 Radix2 算法的 4 周期除法采用 Xilinx 的除法器 IP 核实现,在 IP Catalog 中选择 AXI-stream 接口的除法器,选择 Radix2 算法并将 Latency Option 设置为 4 周期.

4.4 存储架构

图 3 给出了本设计的存储架构,采用 AHB 总线互联的哈佛架构,Cache 和 TCM(Tightly Coupled Memory)单独享有一个 AHB 总线主端,通过 AHB 仲裁器接入同一个总线. 同时通过 AHB-APB 桥接外部低速模块,本设计仅挂载 UART 模块用以

打印程序运行结果及 Debug 模块用以支持调试.

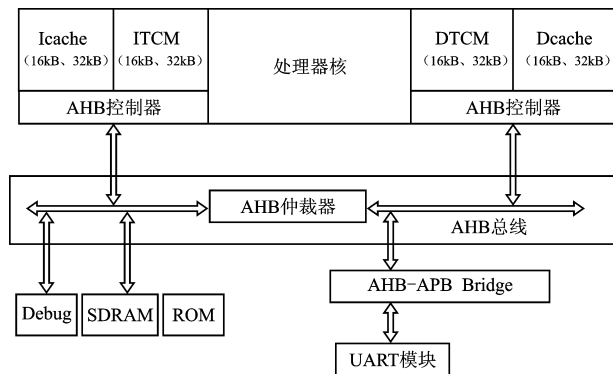


图 3 处理器存储架构设计图

Fig. 3 Processor storage architecture design

TCM 与 Cache 的区别在于,TCM 仅依据空间容量大小识别是否数据命中,当运行的程序范围超过了 TCM 的区域,则从外部存储器访问数据. 本研究中将 Cache 与 TCM 的大小分别设置为 16 kB、32 kB 两种. 同时,针对 Cache 的设计思路,考虑了直接映射法与两路组相连的设计方法进行对比研究. 本设计均支持 TCM 或 Cache 的使能. 在不使能 TCM 和 Cache 的情况下,处理器核通过 AHB 总线访问片上 ROM. 使能 ITCM 或 ICache 的同时,处理器在开始运行前预先加载片上 ROM 的数据到 ITCM 或 ICache 中,并从指定的地址开始运行程序. 在程序命中的前提下,处理器核与 ITCM 或 ICache 的访问周期为一个时钟周期,节省访问时间. DTCM 与 DCache 的访问方式同 ICTM 与 ICache.

5 验证测试

5.1 功能测试

功能验证使用 RISC-V 官方推出的 RISC-V compliance test 测试集,该测试集能够针对 RISC-V 核实现的所有指令和寄存器进行完整性测试.

本文使用 Modelsim 软件对处理器核进行测试,主要分为几个步骤:(1)从根目录文件下加载测试文件;(2)模拟串口发送将文件发送至 Debug 模块,写入片内 ROM 中;(3)模拟串口给 Debug 模块发送处理器启动指令,处理器完成程序文件的搬移;(4)开始运行测试程序至 pass 语句;(5)遍历所有的测试文件.

5.2 性能测试

性能选择 CoreMark 作为测试集. CoreMark 是一个综合基准,用于测量嵌入式系统中 CPU 的性能,旨在成为一个行业标准,取代过时的 Dhrystone

基准。

将 Coremark 程序移植到处理器核中运行，并重新编写了时钟函数与串口打印函数以支持 CoreMark 代码。板级测试在在 Xilinx 的 XC7K325T 开发板上进行，主频为 50 MHz，串口一端通过 USB 线转接至 PC 机端，最终测试的结果显示在串口工具中，测试板卡如图 4 所示。

6 测试分析

6.1 功能测试分析

针对 riscv-tests 测试集，在 modelsim 上仿真测试，结果表明指令测试通过。图 5 给出了算数逻辑指令 ADD 的仿真结果。图 6 给出了 CSR 指令寄存器的仿真测试结果。仿真结果表明指令执行至 pass 语句，表明结果通过。

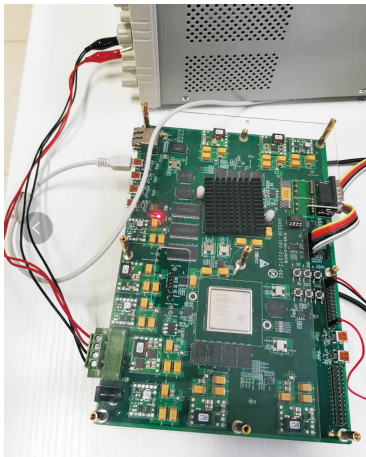


图 4 处理器板级验证环境

Fig. 4 Processor board level verification environment

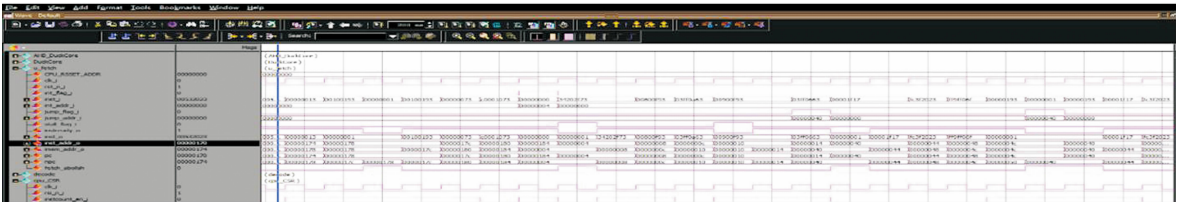


图 5 算术逻辑指令 riscv-tests 运行结果

Fig. 5 Arithmetic logic instruction riscv-tests running results

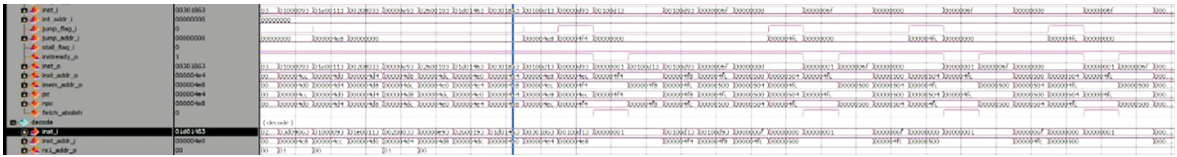


图 6 CSR 指令 riscv-tests 运行结果

Fig. 6 CSR instruction riscv-tests running results

6.2 测试与分析

图 7 给出了板级运行 CoreMark 程序的串口打印结果，在全部性能提升措施使能的条件下，本处理器的 CoreMark 最终得分为 3.06 CoreMark/MHz。

表 1 给出了本处理器设计的各项 CoreMark 跑分结果，其中 Cache 大小均设置为 32 kB。本设计在不使能所有优化模块的情况下，跑分仅有 1.72，增加 DCache 使能后跑分结果上升至 2.04，这是因为每次数据访存的效率提升，导致性能提升，但是由于 DCache 初始化未命中情况较多，所以性能提升有限。增加单周期乘法后，本处理器的跑分结果可达

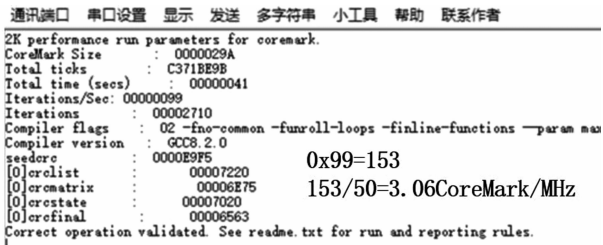


图 7 CoreMark 运行结果

Fig. 7 Coremark running results

2.08，这是由于乘法指令在整个 CoreMark 测试集中所占比例不高，提升有限。增加 ICache 使能后，跑分结果达到 2.7，性能提升 29.8%，这是因为当程序

运行在 AHB-ROM 中时,每次取指所需要的时间依赖于总线 AHB 的速率,由于 AHB 总线的特点,通过总线访问程序的方式需要至少 3 个时钟周期,导致取指延时,而使能 ICache 后仅需要一个周期. 增加了预测算法后,跑分结果达到 3.06,说明指令预测减少了由于条件跳转导致流水线冲刷浪费的时间,性能提高 11.33%. 增加快速除法后,跑分结果没有提升,因为除法指令在整个 CoreMark 指令集中比例最小,仅占不到 50 条指令,因此虽然运算速度提升,但跑分结果不变.

表 1 本处理器 CoreMark 跑分详细结果

Tab.1 Detailed results of CoreMark running of the processor

ICache (是否 使能)	DCache (是否 使能)	乘法(是否使 用单周期 乘法)	除法(是否 使用快速 除法)	预测(是否 使用静态 预测)	Core Mark/ MHz
×	×	×	×	×	1.72
×	√	×	×	×	2.04
×	√	√	×	×	2.08
√	√	√	×	×	2.7
√	√	√	×	√	3.06
√	√	√	√	√	3.06

表 2 给出了在使能快读乘法、不使能快速除法、不使能预测情况下,不同容量的存储空间以及设计方法对处理器性能的影响. CoreMark 的测试集编译后的程序大小为 20 kB,在该情况下分析 ITCM 及 ICache 的使用对处理器性能的影响. 由 3、8 组可知不使能 TCM 或者 Cache 的情况下,性能均未得到提升. 由 2、6、7 组可知,在 32 kB 使能 ICTM 或 ICache 的情况下,两者的跑分一致,且 Cache 的设计不影响性能. 由 1、4、5 组对比可知,当存储空间小于为 16 kB 时(小于 32 kB),使用 ITCM 或者 ICache 的性能都有所下降,原因在于都出现了无法命中的情况. 4、5 组对比说明两路组相连的设计方法优于直接映射法. 1、4 组对比可知,直接映射法与 TCM 的性能差距较小,原因在于直接映射法的实现也是基于固定地址映射,但是映射的区域由设计者自己选择,映射的区域可控. 实际上,由于本处理器在设计 ICache 的过程中,增加了 ICache 的初始化过程,导致 ICache 内部已经存储了 ROM 空间的数据(该实现方法与 TCM 类似),因此,使能 ICache 的性能得以与使能 ITCM 一样. 如果不实现该操作,则性能上会略低于使能 ITCM 的情况.

表 2 各开源处理器跑分结果对比

Tab.2 Comparison of the results of this processor storage architecture

组号	分类情况(使能 DCache,32kB)	CoreMark/MHz
1	ITCM(使能,16 kB)	2.4
2	ITCM(使能,32 kB)	2.7
3	ITCM(不使能)	2.04
4	ICache(使能,16 kB,直接映射法)	2.43
5	ICache(使能,16 kB,两路组相联)	2.52
6	ICache(使能,32 kB,直接映射法)	2.7
7	ICache(使能,32 kB,两路组相联)	2.7
8	ICache(不使能)	2.04

本处理器引用了蜂鸟 E203 的静态预测方法与本处理器预测方法进行对比,结果跑分结果一致. 原因是由于本测试程序使用的 CoreMark 跑分 bin 文件中,并未出现 JALR 指令. 笔者对测试程序进行代码冗余,将测试 bin 文件增加至 50 kB 以上,JALR 指令的使用频率依旧不大. 因此,本静态预测方法,在程序体量小的情况下能够切实节省一定的资源.

表 3 给出了本设计的 FPGA 资源情况与其他设计的对比. 综合分析,本设计的处理器性能优于大部分的开源处理器^[8,17-19],跑分结果优于 STM32L152^[20] 处理器(Cortex-M3). 并且对比之下,本设计的资源占用量不大.

表 3 各开源处理器跑分结果对比

Tab.3 Comparison of running results of each open-source processor

	位宽	架构	Slice Registers	Slice LUTs	CoreMark MHz (运行环境)
蜂鸟 E203	32	RISC-V	1935	4456	2.14(ITCM)
V-scale	32	RISC-V	1003	2700	2.1(片上 RAM)
Tinyriscv	32	RISC-V	1984	5754	2.4(片上 RAM)
Ultrascale-riscv	32	RISC-V	2706	3598	2.94(ICache)
STM32L152	32	ARM Cortex-M3	—	—	2.89(internal Flash)
本处理器	32	RISC-V	2124	2988	3.06(ICache)

6.3 技术对比

表 4 给出了本处理器一些关键技术指标同部分开源处理器之间的对比情况. 同大多数 5 级流水开源处理器相比,本处理器性能较优,但低于规模较大的多级流水的处理器.

表 4 本处理器的关键技术对比情况

Tab. 4 Comparison of key technologies of this processor

处理器核	Rocket	SweRV EH1	VexRiscv	Ultrascale- riscv	本处 理器
指令集架构	RV64 MAFD	RV32 IMC	RV32 IM	RV32 IM	RV32 IMZicsr
位宽	64	32	32	32	32
流水线深度	5	9	5	5	5
总线接口	TileLink	AHB/AXI	AXI	AXI	AHB
Cache	Y	Y	Y	Y	Y
顺序/乱序	顺序	顺序	顺序	顺序	顺序
静态预测/动态预测	动态	动态	动态	无预测	静态
CoreMark	2.32	4.9	2.35	2.94	3.06

7 结束语

本文基于 RISC-V 指令集设计一款 5 级顺序流水线处理器,并在指令预测、流水线、乘除法、存储架构做性能的优化.实现的处理器在 FPGA 上完成了板级验证,并通过 CoreMark 测试,性能上优于大部分同规模的开源项目.该处理器在性能优化上还有一定空间,可以结合 RISC-V 指令集的特点考虑在动态预测、多线程等方式做进一步的研究.

参考文献:

[1] OLUGBON A, KHAWAM S, ARSLAN T, et al. An AMBA AHB-based reconfigurable SoC architecture using multiplicity of dedicated flyby DMA blocks[C]//Proceedings of the ASP-DAC 2005. Asia and South Pacific Design Automation Conference, 2005. Shanghai, China; IEEE, 2005. DOI: 10.1109/ASPDAC.2005.1466571.

[2] PANG X M, YU D J, LI J B, et al. Design and application of IP core in SoC technology[C]//Proceedings of 2010 Third International Symposium on Information Science and Engineering. Shanghai, China; IEEE, 2011; 71-74. DOI: 10.1109/ISISE.2010.94.

[3] 雷思磊. RISC-V 架构的开源处理器及 SoC 研究综述[J]. 单片机与嵌入式系统应用, 2017, 17(2): 56-60. DOI: CNKI:SUN:DPJY 0.2017-02-019.

LEI S L. Research on open source processor and SoC based on RISC-V[J]. Microcontrollers & Embedded Systems, 2017, 17(2): 56-60. DOI: CNKI: SUN: DPJY 0.2017-02-019.

[4] CHIPS Alliance. Rocket-chip[EB/OL]. (2021-09-23) [2021-06-09]. <https://github.com/chipsalliance/rocket-chip>.

[5] BOOM: the berkeley out-of-order RISC-V Processor. Riscv-boom[EB/OL]. (2021-05-01)[2021-06-09]. <https://github.com/riscv-boom/riscv-boom>.

[6] CHIPS Alliance. Cores-SweRV[EB/OL]. (2021-10-09) [2021-06-09]. <https://github.com/riscv-boom/riscv-boom>.

[7] T-Head Semiconductor Co., Ltd. Wujian100_open[EB/OL]. (2021-07-30) [2021-06-09]. https://github.com/T-head-Semi/wujian100_open.

[8] 胡振波. 手把手教你设计 CPU—RISC-V 处理器[M]. 北京: 人民邮电出版社, 2018: 6.

HU Z B. Teach you how to design CPU: RISC-V processor[M]. Beijing: People's Posts and Telecommunications Publishing House, 2018: 6.

[9] 张迅珍, 梁青, 李涛. RV32I 控制单元设计与实现[J]. 微电子学与计算机, 2018, 35(3): 74-78. DOI: CNKI:SUN:WXYJ 0.2018-03-019.

ZHANG X Z, LIANG Q, LI T. Design and implementation of control element of RV32I[J]. Microelectronics & Computer, 2018, 35(3): 74-78. DOI: CNKI:SUN:WXYJ 0.2018-03-019.

[10] 李雨倩, 焦继业, 刘有耀, 等. 嵌入式 RISC-V 乱序执行处理器的研究与设计[J]. 计算机工程, 2021, 47(2): 261-267. DOI: 10.19678/j.issn.1000-3428.0057141.

LI Y Q, JIAO J Y, LIU Y Y, et al. Research and design of embedded RISC-V out-of-order execution processor[J]. Computer Engineering, 2021, 47(2): 261-267. DOI: 10.19678/j.issn.1000-3428.0057141.

[11] 郝振和, 焦继业, 李雨倩. 基于 AHB 总线的 RISC-V 微处理器设计与实现[J]. 计算机工程与应用, 2020, 56(20): 52-58. DOI: 10.3778/j.issn.1002-8331.1911-0331.

HAO Z H, JIAO J Y, LI Y Q. Design and implementation of RISC-V microprocessor based on AHB bus[J]. Computer Engineering and Applications, 2020, 56(20): 52-58. DOI: 10.3778/j.issn.1002-8331.1911-0331.

[12] 李东泽, 曹凯宁, 曲明, 等. 五级流水线 RISC-V 处理器软硬件协同仿真验证[J]. 吉林大学学报(信息科学版), 2017, 35(6): 612-616. DOI: 10.3969/j.issn.1671-5896.2017.06.006.

LI D Z, CAO K N, QU M, et al. Hardware and software Co-simulation of RISC-V processor with five-stage pipeline[J]. Journal of Jilin University (Information Science Edition), 2017, 35(6): 612-616. DOI: 10.19292/j.cnki.jdxxp.2017.06.006.

[13] 李其高. 面向 IoT 终端设备的 RISC-V 微控制器设计

- 与分析[J]. 单片机与嵌入式系统应用, 2018, 18(3): 64-66.
- LI Q G. Design and analysis of RISC-V microcontroller for IoT endpoint devices[J]. Microcontrollers & Embedded Systems, 2018, 18(3): 64-66.
- [14] WATERMAN A, LEE Y, PATTERSON D A, et al. The RISC-V instruction set manual, Volume I: user-level ISA, version 2.1[R]. Berkeley: University of California, 2016.
- [15] WATERMAN A, LEE Y, PATTERSON D A, et al. The RISC-V instruction set manual, Volume I: user-level ISA version 2.0[R]. Berkeley: University of California, 2016.
- [16] PATTERSON D, WATERMAN A. The RISC-V reader: an open architecture atlas[M]. Berkeley: University of California, 2017.
- [17] ASANOVIĆK, AVIZIENIS R, WATERMAN A, et al. The rocket chip generator[R]. EECS Department, Berkeley: University of California, 2016.
- [18] Liangkangnan. Tinyriscv [EB/OL]. (2021-10-15) [2021-06-09]. <https://gitee.com/liangkangnan/tinyriscv>.
- [19] Ultraembedded. Riscv[EB/OL]. (2021-09-19)[2021-06-09]. <https://github.com/ultraembedded/riscv>.
- [20] Ebedded Microprocessor Benchmark Consortium. EEMBC benchmark score viewer [EB/OL]. (2015-01-12). https://www.eembc.org/viewer/?benchmark_seq=1686.
- [21] Austin. VexRiscv[EB/OL]. (2021-08-01)[2021-06-09]. <https://gitee.com/peasent/VexRiscv#cpu-generation>.

作者简介:

李介民 男,(1992-),博士. 研究方向为计算机应用技术、计算机体系结构、高可靠计算机应用.

E-mail:1137377326@qq.com.

张善从 男,(1977-),博士,二级研究员. 研究方向为高可靠计算机应用、航天电子学工程.