

《计算机系统结构》

调研报告

开源指令集架构 RISC-V



学院： 计算机学院（国家示范性软件学院）

班级： 2018211314

姓名： 李志毅

学号： 2018211582

摘要

RISC-V 是基于精简指令集计算(RISC)原理建立的开放指令集架构(ISA)，V 表示为第五代 RISC(精简指令集计算机)，表示此前已经四代 RISC 处理器原型芯片。RISC-V 是在指令集不断发展和成熟的基础上建立的全新指令。

RISC-V 指令集完全开源，设计简单，易于移植 Unix 系统，采用模块化设计，拥有完整的工具链，同时有大量的开源实现和流片案例。

本文基于文献阅读，具体而细致的调研了 RISC-V 指令集的特点、应用以及发展前景。

关键词：RISC、架构、指令调度、指令集

1 调研目的

1.1 引言

RISC-V 是一个基于精简指令集（RISC）原则的开源指令集架构（ISA），与大多数指令集相比，RISC-V 指令集可以自由地用于任何目的，允许任何人设计、制造和销售 RISC-V 芯片和软件。虽然这不是第一个开源指令集，但它具有重要意义，因为其设计使其适用于现代计算设备（如仓库规模云计算、高端移动电话和微小嵌入式系统）。设计者考虑到了这些用途中的性能与功率效率。该指令集还具有众多支持的软件，这解决了新指令集通常的弱点。在此之前，已经历经了四代 RISC 处理器原型芯片，由加州大学伯克利分校的 David A. Patterson 教授带领研发，其设计充分考虑用途中的性能与功率效率，具有众多支持的软件，适用于现代计算设备。^[1]

1.2 调研方式

本次调研的主要方式是基于文献调研，分析 RISC-V 指令集架构的体系结构特点以及 RISC-V 体系结构指令调度的方法，并以文字和图片的形式进行总结梳理。主要基于的文献数据库平台有：知网、万方、超星等。

2 调研成果

2.1 RISV-V 的基本设计

RISC-V 是一个典型三操作数，加载存储形式的 RISC 架构，包含 3 个基本指令集和 6 个拓展指令集，如下图所示为 RISC-V 的指令集组成

表 1 RISC-V 的指令集组成			
指令集类型	名 称	指令数	说 明
基本指令集	RV32I	47	整数指令,包含:算术、分支、访存。 32 位寻址空间,32 个 32 位寄存器
	RV32E	47	指令与 RV32I 一样,只是寄存器数量变为 16 个,用于嵌入式环境
	RV64I	59	整数指令,64 位寻址空间,32 个 64 位寄存器
	RV128I	71	整数指令,128 位寻址空间,32 个 128 位寄存器
扩展指令集	M	8	包含 4 条乘法、2 条除法、2 条余数操作指令
	A	11	包含原子操作指令,比如:读-修改-写,比较-交换等
	F	26	包含单精度浮点指令
	D	26	包含双精度浮点指令
	Q	26	包含四倍精度浮点指令
	C	46	压缩指令集,其中的指令长度是 16 位,主要目的是减少代码大小

基本指令集的名称后缀都是 I，表示 Integer，任何一款采用 RISC-V 架构的处理器都要实现一个基本指令集，根据需要，可以实现多种扩展指令集，例如：如果实现了 RV32IM，表示实现了 32 位基本指令集和乘法除法扩展指令集。如果实现了 RV32IMAFD，那么可以使用 RV32G 来表示，表示实现了通用标量处理器指

令集。RV32I 指令集有 47 条指令，能够满足现代操作系统运行的基本要求，47 条指令按照功能可以分为如下几类：

- ① 整数运算指令，实现算术、逻辑、比较等运算。
- ② 分支转移指令，实现条件转移、无条件转移等运算，并且没有延迟槽。
- ③ 加载存储指令，实现字节、半字、字的加载和存储操作，采用的都是寄存器相对寻址方式。
- ④ 控制与状态寄存器访问指令，实现对系统控制与状态寄存器的原子读一写、原子读一修改、原子读一清零等操作。
- ⑤ 系统调用指令，实现系统调用、调试等功能。

2.2 RISV-V 指令集架构的体系结构特点

2.2.1 架构篇幅短小精悍

处理器领域当今最主流的架构有 x86 和 ARM 架构，在历经几十年的发展和技术的不断沉淀，为了保证架构的向后兼容性，其保留了很多过时的定义，导致如今 x86 指令数目多，指令冗余严重，而随着如今商用产品的发展，过于臃肿的指令无法适用于希望得到快速而精确相应的设备，其商业价值展现出局限性。

RISC-V 指令集架构基于已成熟的技术进行构建，抛弃向后兼容的历史包袱，真正意义上实现了短小精悍，RISC-V 的规范文档仅有 145 页，而特权架构文档的篇幅仅为 91 页，基础指令集只有 40 多条。^[2]

2.2.2 支持可扩展定制指令

x86 和 ARM 不具备模块化的架构，其对应支持各领域之间互不兼容，而 RISC-V 架构相对于其他成熟的商业架构的最大一个不同还是在于它是一个模块化的架构，因此 RISC-V 架构不仅短小精悍，而且其不同的部分还能以模块化的方式组织在一起，从而试图通过一套统一的架构满足各种不同的应用。RISC-V 最基本也是唯一强制要求实现的指令集部分是由 1 字母表示的基本整数指令子集，使用该整数指令子集，便能够实现完整的软件编译器。其他的指令子集部分均为可选的模块，具有代表性的模块包括：

基本指令集	指令数	描述
RV32I	47	32 位地址空间与整数指令，支持 32 个通用整数寄存器
RV32E	47	RV32I 的子集，仅支持 16 个通用整数寄存器
RV64I	59	64 位地址空间与整数指令，及一部分 32 位的整数指令
RV128I	71	128 位地址空间与整数指令，及一部分 64 位和 32 位的指令
扩展指令集	指令数	描述
M	8	整数乘法与除法指令
A	11	存储器原子操作指令和 Load-Reserved/Store-Conditional 指令
F	26	单精度（32 比特）浮点指令
D	26	双精度（64 比特）浮点指令，必须支持 F 扩展指令
C	46	压缩指令，指令长度为 16 位

2.2.3 可配置的通用寄存器组

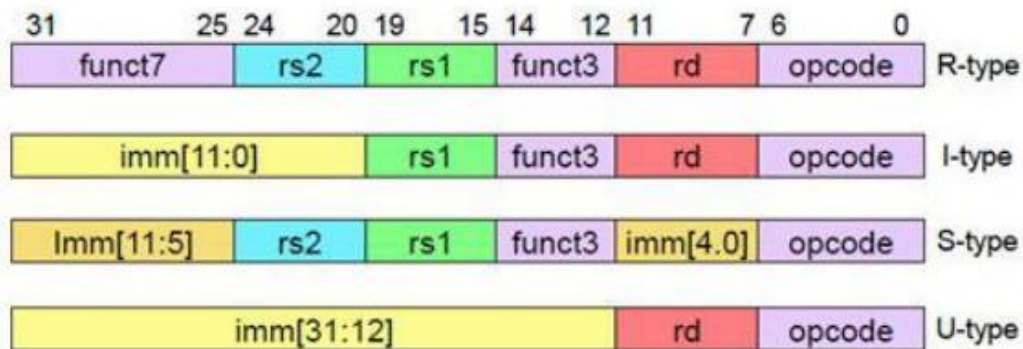
RISC-V 架构支持 32 位或者 64 位的架构，32 位架构由 RV32 表示，其每个通用寄存器的宽度为 32 比特；64 位架构由 RV64 表示，其每个通用寄存器的宽度

为 64 比特。RISC-V 架构的整数通用寄存器组，包含 32 个（I 架构）或者 16 个（E 架构）通用整数寄存器，其中整数寄存器 0 被预留为常数 0，其他的 31 个（I 架构）或者 15 个（E 架构）为普通的通用整数寄存器。

如果使用了浮点模块（F 或者 D），则需要另外一个独立的浮点寄存器组，包含 32 个通用浮点寄存器。如果仅使用 F 模块的浮点指令子集，则每个通用浮点寄存器的宽度为 32 比特；如果使用了 D 模块的浮点指令子集，则每个通用浮点寄存器的宽度为 64 比特。

2.2.4 规范的指令编码格式

随着 RISC 架构的不断更新，指令的添加和修改，现在大部分 RISC 架构的指令呈现凌乱状态，译码工作操作较为复杂，对译码器的要求也较高，现在，得益于后发优势和总结经验，RISC-V 架构实现规整指令集编码，指令所需的通用寄存器的索引都被放在固定的位置，指令译码器可以非常便捷的译码出寄存器索引然后读取通用寄存器组。



2.2.5 简化的分支跳转指令和静态预测机制

RISC-V 架构有两条无条件跳转指令，jal 与 jalr 指令。跳转链接指令 jal 可用于进行子程序调用，同时将子程序返回地址存在链接寄存器中。跳转链接寄存器指令 jalr 指令能够用于子程序返回指令，通过将 jal 指令（跳转进入子程序）保存的链接寄存器用于 jalr 指令的基地址寄存器，则可以从子程序返回。

RISC-V 架构有 6 条带条件跳转指令，这种带条件的跳转指令跟普通的运算指令一样直接使用 2 个整数操作数，然后对其进行比较，如果比较的条件满足时，则进行跳转。因此，此类指令将比较与跳转两个操作放到了一条指令里完成。

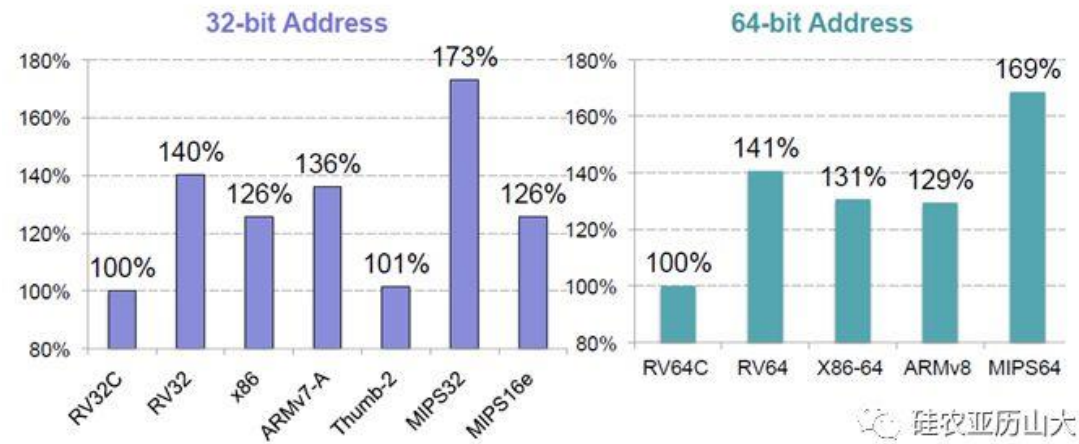
除此之外，RISC-V 对于静态预测机制的设计也是重要的一点，对于没有配备硬件分支预测器的低端 CPU，为了保证其性能，RISC-V 的架构明确要求其采用默认的静态分支预测机制，即：如果是向后跳转的条件跳转指令，则预测为“跳”；如果是向前跳转的条件跳转指令，则预测为“不跳”，并且 RISC-V 架构要求编译器也按照这种默认的静态分支预测机制来编译生成汇编代码，从而让低端的 CPU 也能得到不错的性能。

2.2.6 压缩指令子集

基本的 RISC-V 基本整数指令子集（字母 I 表示）规定的指令长度均为等长的 32 位，这种等长指令定义使得仅支持整数指令子集的基本 RISC-V CPU 非常容易设计。但是等长的 32 位编码指令也会造成代码体积（Code Size）相对较大的问题。

为了满足某些对于代码体积要求较高的场景（譬如嵌入式领域），RISC-V 定义了一种可选的压缩（Compressed）指令子集，由字母 C 表示，也可以由 RVC 表示。RISC-V 具有后发优势，从一开始便规划了压缩指令，预留了足够的编码空间，16 位长指令与普通的 32 位长指令可以无缝自由地交织在一起，处理器也没有定义额外的状态。

RISC-V 架构的研究者进行了详细的代码体积分析，如图 3 所示，通过分析结果可以看出，RV32C 的代码体积相比 RV32 的代码体积减少了百分之四十，并且与 ARM，MIPS 和 x86 等架构相比都有不错的表现。



2.3 RISV-V 指令集架构的优势

2.3.1 与开源指令集架构比较

RISC-V 发布之前，实际上已经有几种开源指令级架构，包括 SPARC V8、OpenRISC，其中 SUN 发布的开源多核多线程处理器 OpenSparcT1、OpenSparcT2，以及欧空局的 LEON3 采用的就是 SPARC V8，OpenRISC 也有同名的开源处理器，RISC-V 与前两者的比较如下表所列。^[3]

项 目	SPARC V8	OpenRISC	RISC-V
分为基本指令集+扩展指令集		是	是
支持压缩指令			是
支持四倍精度浮点运算	是		是
寻址空间			
32bit	是	是	是
64bit		是	是
128bit			是
软件支持			
GCC	是	是	是
LLVM	是	是	是
Linux	是	是	是
QEMU	是	是	是

2.3.2 与商业指令集架构比较

UBC 的研究人员设计了一款采用 RISC-V 指令集架构的开源处理器 Rocket，

并且成功流片了 11 次，其中采用台积电 40nm 工艺时的性能，与采用同样工艺、都是标量处理器的 ARM Cortex-A5 的性能对比如下表：

项 目	ARM Cortex-A5	RISC-V Rocket	Ratio
寄存器宽度	32	64	2
主频	>1GHz	>1GHz	1
Dhrystone	1.57DMIPS/MHz	1.72DMIPS/Hz	1.1
面积(不包含 Cache)	0.27mm ²	0.14mm ²	0.5
面积(包含 16KB Cache)	0.53mm ²	0.39mm ²	0.7
动态功耗	<0.08mW/MHz	0.034mW/MHz	>0.4

由表中可以看出，Rocket 占用面积更小，且功耗更低，性能更优。

3 调研结论

RISC-V 的发展十分迅速，除了前文已述的机遇 RISC-V 的开源处理器、开源 SoC 大量涌现，还有很多的商用处理器也计划采用 RISC-V 指令集架构。RISC-V 的发展一方面得益于自身设计吸取了 RISC 接近 40 年的经验教训，使得架构设计更加合理，另一方面得益于日渐成熟的软件生态。RISC-V 后发优势，成为了全新的架构，一个基于精简指令集（RISC）原则的开源指令集架构（ISA）。

可以预见，RISC-V 即将进入一个快速发展的阶段，应该会在以下几个方面有突破进展：有若干成熟的、可商业化的、采用 RISC-V 架构的芯片问世，并得到大规模应用；性能逼近主流桌面处理器；主流处理器与采用 RISC-V 架构的开源处理器组成的异构系统；移植到 RISC-V 架构的操作系统更加稳定可靠；采用上百个简单 RISC-V 核的多核并行计算；计算机教学中采用 RISC-V 作为范例教学；调试功能得到进一步加强。对于国内而言，RISC-V 提供了一个很好的参考，可以用来实现自主可控的处理器

4 参考文献

[1] 芯来科技市场部. RISC-V 发展研究报告(一) [N]. 电子报, 2021-04-04(002).
[2] 俞灵琦. RISC-V 的“芯”技术[J]. 华东科技, 2021(02):18-21.
[3] Mark Patrick, 禾沐. RISC-V 为什么会成为热点?[J]. 单片机与嵌入式系统应用, 2020, 20(11):15-16+20.