



# RISC-V简单介绍

魏堃

# 目录

## CONTENTS

背景

发展

意义

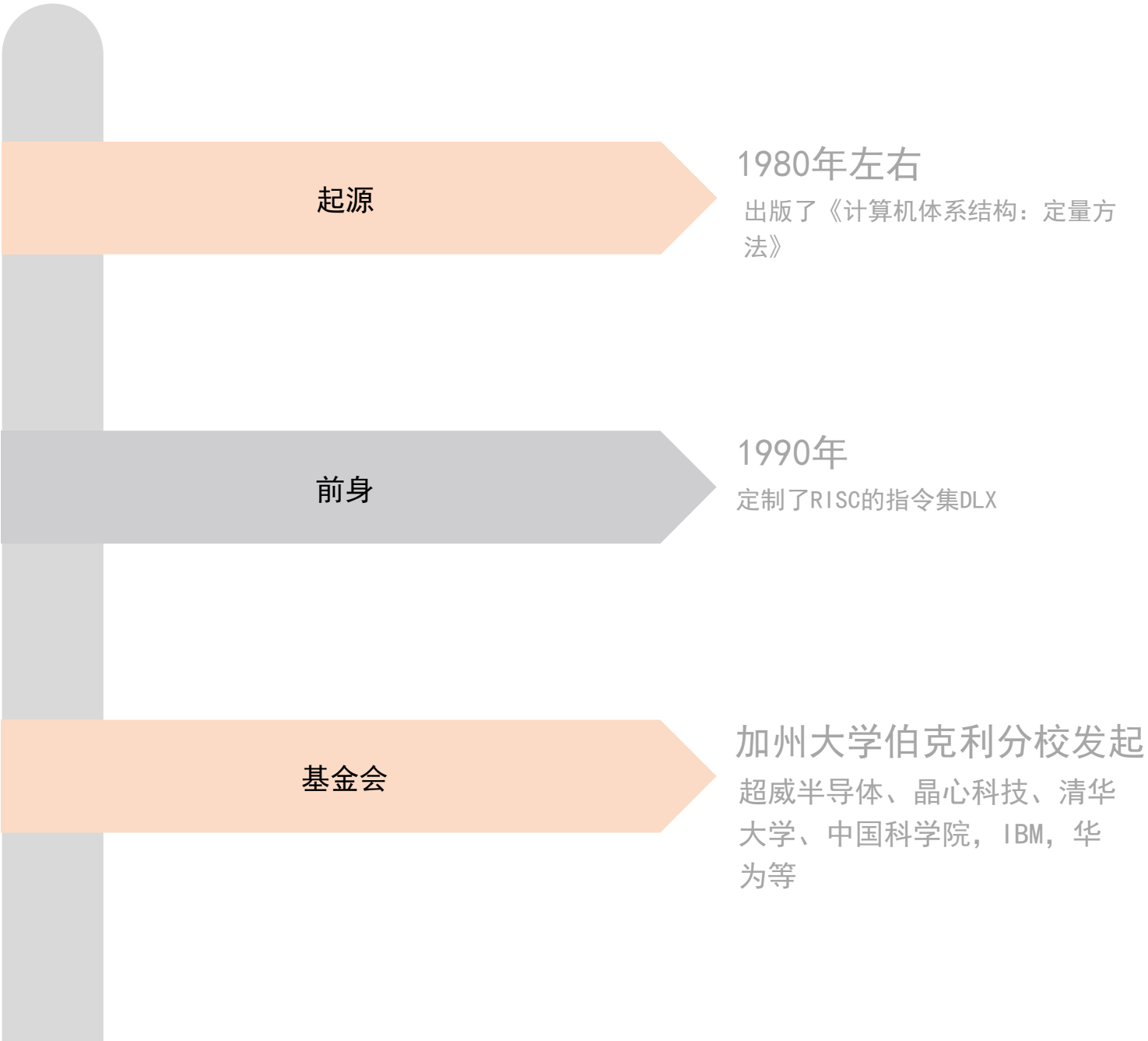
软件



01

背景





起源

1980年左右

出版了《计算机体系结构：定量方法》

前身

1990年

定制了RISC的指令集DLX

基金会

加州大学伯克利分校发起

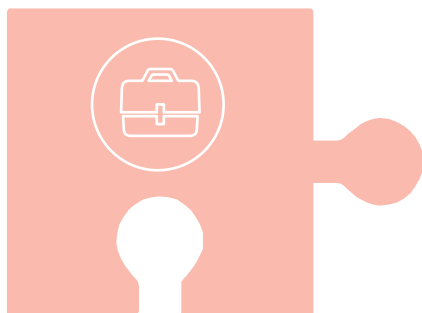
超威半导体、晶心科技、清华大学、中国科学院，IBM，华为等

An orange semi-circle graphic with a white number '02' inside it.

02

发展

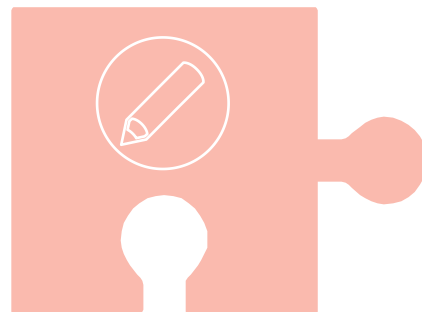
A thin horizontal line with small dots at each end, positioned below the Chinese characters.



RISC设计者们认为指令集因为位于硬件和软件之间，所以是电脑主要的沟通桥梁，因此如果有一个设计良好的指令集是开源而且可以被任何人使用的，就可以让更多的资源能够重复利用，而大大的减少软件的成本。而这样的指令集也会增加硬件供应商市场的竞争力，因为硬件供应商们可以挪用更多资源来进行设计，减少处理软件支持的事务。



设计者们表示，RISC-V指令集是给实际上的电脑使用的，它不像其他学术上的指令集设计，只有为了比较好阐述理念而做优化。而RISC-V指令集有一些功能是可以增加电脑速度又可以减少成本和电源使用。这些特色包含，[Load/store架构](#)，在CPU里面的比特表示方法来简化[MUX（数据多任务器）](#)，以标准为基础来简化的浮点数，架构中立的设计和把[MSB \(Most significant bit\)](#)放到固定位置来加速[Sign extension](#)。而sign-extension常常就是[静态时序分析](#)里面的关键路径（Critical timing path）



RISC-V也可以拿来做学术上的使用。它拥有简化的整数指令子集允许学生拿来作基本的练习，而整数指令子集就是一个简单的指令集架构（ISA）让软件可以控制研究上的机器。而不定长度的指令集架构也允许扩展来满足研究或是学生练习上的需求。[\[10\]](#)分割出来的特权指令集可以支持在不重新设计编译器的情况下，进行操作系统方面的研究。RISC-V的开放的知识产权允许相关的设计被发布、使用和修改。



03

意义

---

RISC-V的作者们旨在提供数种可以在[BSD许可证](#)之下自由使用的CPU设计。该许可证允许像是RISC-V芯片设计等派生作品可以像RISC-V本身一样是公开且自由发行，也可以是闭源或者是专有财产。

相比而言，[ARM控股](#)和[MIPS科技](#)等商业芯片供应商会对使用其[专利](#)者收取高额的许可费用。<sup>[5]</sup>他们也要求在接收其描述设计优点的文件和指令集前，还需要签署保密协议。许多设计优点为完全专有，从来不会披露给客户。这种保密制度阻碍了公共教育用途和安全审核，以及开发公共、低成本的[自由及开放源代码软件编译器](#)和[操作系统](#)。

开发一个CPU需要多种专业的设计知识，包括电子逻辑、编译器和操作系统。这种资源在专业工程团队之外很难见到。所以现代且高质量的通用计算机指令集近年来除了学术环境以外并没有在任何地方被广泛使用，甚至没有被阐述。正因如此，许多RISC-V贡献者将此视为整个社群付出的成果，而这也是RISC-V有很多任务程上的应用的一项原因。

RISC-V的作者们还有大量研究和用户体验来验证他们在硅片和仿真中的设计。RISC-V指令集是从一系列的学术计算机设计项目直接发展而来的。它一开始的目的有一部分是为了帮助这些项目。



An orange semi-circle graphic with a white number '04' centered inside it.

04

软件

A thin horizontal line with small dots at both ends, positioned below the text '软件'.



一个新的CPU指令集通常的问题是缺少CPU的硬件设计跟软件的支持。在RISC-V的网站有一个user mode指令集的规格，还有1个用来支持操作系统模式的优先指令集的初步规格。

设计软件包括了1个设计编译器Chisel，它可把设计编译成Verilog代码。网站上还有测试用的参考资料可以用来验证设计的正确性。

RISC-V目前提供的软件有GNU Compiler Collection (GCC) toolchain (具有调试器GDB)、一套LLVM toolchain、OVPsim模拟器（以及RISC-V快速处理器模式的软件参考库）、Spike 模拟器，以及一套在QEMU上运行的模拟器。

操作系统的支持包括Linux 核心、FreeBSD, 以及NetBSD，但是监督模式的指令在2016年11月10日还没有标准化，所以这方面的支持还不是正式的。有一个早期的RISC-V的FreeBSD操作系统已经在2016年2月上传到开放源代码社群，而且包含在FreeBSD 11.0。Debian跟Fedora的版本也有人在移植，并且在逐渐稳定中。已经有人做了1个U-Boot的移植版本。<sup>[19]</sup>UEFI Spec v2.7 定义了RISC-V UEFI的规格，而且惠普公司的工程师已经做好一个TianoCore的移植版本，并且将会上传到开放源代码社群。已经有人做好了一个L4 microkernel family的移植。还有一个在网页上用JavaScript写的RISC-V Linux系统模拟器<sup>[21]</sup>。



谢谢大家