

Risc-v 开发

Risc-v 基础知识

北京飞利信科技股份有限公司

2018 年 5 月

Risc-v 应用方向

Risc-v ISA 指令集架构，完成了多款 RISC-V 处理器实现，包括几款硅片制造。如下图所示。

名字	流片日期	工艺	ISA
Raven-1	2011-05-29	ST 28nm FDSOI	RV64G1 Xhwacha1
EOS14	2012-04-01	IBM 45nm SOI	RV64G1p1 Xhwacha2
EOS16	2012-08-17	IBM 45nm SOI	RV64G1p1 Xhwacha2
Raven-2	2012-08-22	ST 28nm FDSOI	RV64G1p1 Xhwacha2
EOS18	2013-01-06	IBM 45nm SOI	RV64G1p1 Xhwacha2
EOS20	2013-07-03	IBM 45nm SOI	RV64G1p99 Xhwacha2
Raven-3	2013-09-26	ST 28nm SOI	RV64G1p99 Xhwacha2
EOS22	2014-03-07	IBM 45nm SOI	RV64G1p9999 Xhwacha3

图 1.2 已经制造的 RISC-V 测试芯片

首款被制造的 RISC-V 处理器使用 Verilog 书写的，并作为 Raven-1 测试芯片，于 2011 年在 ST 的预先量产 28nm FDSOI 工艺上制造。在 Krste Asanović 的指导下，Yunsup Lee 和 Andrew Waterman 研发了两个内核，并一同被制造：1) 一个具有错误检测触发器的 RV64 标量内核，2) 一个具有 64 位浮点向量单元的 RV64 内核。最开始的微体系结构被称为“火车失事 (TrainWreck)”，主要是因为使用不成熟的设计库在短时间内完成了设计。

接着，一个全新的、按序、去耦合 RV64 内核微体系结构，在 Krste Asanović 指导下，由 Andrew Waterman、Rimas Avižienis 和 Yunsup Lee 研发出来，继续铁轨模式，代号“火箭 (Rocket)”，按照 George Stephenson 成功的蒸汽机车设计而命名。Rocket 是使用 Chisel 语言书写的，Chisel 是由 UC Berkeley 开发的一种新的硬件设计语言。Rocket 中的 IEEE 浮点单元是由 John Hauser、Andrew Waterman 和 Brian Richards 开发的。由此，Rocket 被进一步修正和改进，并 2 次在 28nm FDSOI 工艺上制造 (Raven-2、Raven-3)，为一个光子学项目而 5 次在 IBM 45nm SOI 工艺上制造 (EOS14、EOS16、EOS18、EOS20、EOS22)。现在正在做的工作就是将 Rocket 设计做成一个可参数化的 RISC-V 处理器生成器。

EOS14-EOS22 芯片包含了一个早期版本的 Hwacha，它是一个 64 位 IEEE 浮点向量单元，由 Yunsup Lee、Andrew Waterman、Huy Vo、Albert Ou、Quan Nguyen、Stephen Twigg 在 Krste Asanović 的指导下研发的。EOS16-EOS22 包含了具有 cache

一致性协议的双核，在 Krste Asanović 的指导下由 Henry Cook、Andrew Waterman 研发。EOS14 芯片成功地运行在 1.25GHz，EOS16 芯片遭受 IBM pad 库的一个 bug 影响，EOS18 和 EOS20 成功地运行在 1.35GHz。

Raven 测试芯片的贡献者包括 Yunsup Lee、Andrew Waterman、Rimas Avizienis、Brian Zimmer、Jaehwa Kwak、Ruzica Jevtić、Milovan Blagojević、Alberto Puggelli、Steven Bailey、Ben Keller、Pi-Feng Chiu、Brian Richards、Borivoje Nikolić 和 Krste Asanović。

EOS 测试芯片的贡献者包括 Yunsup Lee、Rimas Avizienis、Andrew Waterman、Henry Cook、Huy Vo、Daiwei Li、Chen Sun、Albert Ou、Quan Nguyen、Stephen Twigg、Vladimir Stojanović 和 Krste Asanović。

Andrew Waterman 和 Yunsup Lee 开发了 C++ ISA 仿真器“Spike”，作为研发过程中的黄金模型，其名字来源于在 US 横贯大陆的铁路竣工庆典上所使用的黄金铁道钉。Spike 已经成为一个 BSD 开源项目。

Andrew Waterman 完成了一个具有初步 RISC-V 压缩指令集设计的硕士论文。

完成了各种各样的 RISC-V FPGA 实现，主要是 Par Lab 项目研究的集成演示的一部分。最新的 FPGA 设计具有 3 个 cache 一致性的 RV64IMA 处理器，运行了一个研究用的操作系统。FPGA 实现的贡献者包括 Andrew Waterman、Yunsup Lee、Rimas Avizienis 和 Krste Asanović。

RISC-V 处理器被用于 UC Berkeley 的数门课程。Rocket 被用于 2011 年秋的 CS250，课程项目的基础，Brian Zimmer 是课程 TA。在 2012 年春的本科生 CS152 课程上，Christopher Celio 使用 ChiSel 语言书写了一系列适合教学用途的 RV32 处理器，命名为“Sodor”，这是“托马斯小火车”和他的小伙伴们生活的小岛。这个套件包括一个微编码内核 (microcoded core)、一个非流水内核、2 级流水内核、3 级流水内核、5 级流水内核，并在 BSD 授权下公开共享。这个套件于 2013 年春在 CS152 课程中得到更新和再度使用，Yunsup Lee 是课程 TA，2014 年春 Eric Love 是课程 TA。Christopher Celio 也开发了一个乱序执行的 RV64 设计，称为 BOOM (Berkeley Out-of-Order Machine)，并具有流水线可视化功能，被用于 CS152 课程。CS152 课程还使用了由 Andrew Waterman 和 Henry Cook 开发的具有 cache 一致性版本的 Rocket 内核。

2013 年夏,定义了 RoCC(Rocket Custom Coprocessor)接口,以简化向 Rocket 内核添加定制加速器的工作。在 2013 年秋, Rocket 和 RoCC 接口被大量应用于 CS250 VLSI 课程。Hwacha 向量单元作为 RoCC 协处理器被重写。

Quan Nguyen 和 Albert Ou, 在 2013 年春,成功的将 Linux 移植到 RISC-V 上。

在 2014 年 1 月, Colin Schmidt 成功的完成了 RISC-V 2.0 的 LLVM 后端。

在 2014 年 3 月, Bluespec 的 Darius Rad 贡献了软浮点 ABI 支持的 GCC 移植。

几个其他的 RISC-V 内核实现, 包括一个由 Tommy 用 Verilog 书写, 一个由 Rishiyur Nikhil 用 Bluespec 书写。

参考文献: riscv-spec-v2.2