

RISC-V开源软件生态现状

以及为什么你现在就应该加入

报告人：吴伟（[@lazyparser](#) on Github/Bilibili）

中国科学院 软件研究所 智能软件研究中心 PLCT实验室

欢迎参与我们的开源项目 <https://github.com/isrc-cas/>



如果希望参与RISC-V开源软件项目开发，欢迎加我微信询问 ☺

关于我

- 软件出身（没做过硬件），从软件的层面（偏见和盲视）去看待和剖析

关于我

- 软件出身（没做过硬件），从软件的层面（偏见和盲视）去看待和剖析
- 主要从业时间在编译器、虚拟机、模拟器领域；2016年开始关注 RISC-V 的发展

关于我

- 软件出身（没做过硬件），从软件的层面（偏见和盲视）去看待和剖析
- 主要从业时间在编译器、虚拟机、模拟器领域；2016年开始关注 RISC-V 的发展
- 2019年成立PLCT实验室，作为ISRC内的一个课题组，邢明杰高级工程师为组长

关于我

- 软件出身（没做过硬件），从软件的层面（偏见和盲视）去看待和剖析
- 主要从业时间在编译器、虚拟机、模拟器领域；2016年开始关注 RISC-V 的发展
- 2019年成立PLCT实验室，作为ISRC内的一个课题组，邢明杰高级工程师为组长
- 2009年加入HelloGCC社区，后成为OSDT/HelloGCC/HelloLLVM 社区负责人

关于我

- 软件出身（没做过硬件），从软件的层面（偏见和盲视）去看待和剖析
- 主要从业时间在编译器、虚拟机、模拟器领域；2016年开始关注 RISC-V 的发展
- 2019年成立PLCT实验室，作为ISRC内的一个课题组，邢明杰高级工程师为组长
- 2009年加入HelloGCC社区，后成为OSDT/HelloGCC/HelloLLVM 社区负责人

本次报告观点仅代表我个人及 PLCT Lab，不代表ISRC/ISCAS

今天演讲的主题内容 (Takeaway)

- RISC-V 已经成为未来的主流架构
 - 任何芯片公司都应投入；任何软件公司都需要适配；从业者应该立刻开始学习

今天演讲的主题内容 (Takeaway)

- RISC-V 已经成为未来的主流架构
 - 任何芯片公司都应投入；任何软件公司都需要适配；从业者应该立刻开始学习
- 拥抱开源软件并积极成为贡献者是企业生存、胜出的必要条件
 - 要么积极融入开源社区，要么被开源软件吞噬；不被upstream支持等于活在ICU

今天演讲的主题内容 (Takeaway)

- RISC-V 已经成为未来的主流架构
 - 任何芯片公司都应投入；任何软件公司都需要适配；从业者应该立刻开始学习
- 拥抱开源软件并积极成为贡献者是企业生存、胜出的必要条件
 - 要么积极融入开源社区，要么被开源软件吞噬；不被upstream支持等于活在ICU
- RISC-V 软件生态领域依然有大量机会可以把握
 - 以中国科学院软件研究所PLCT实验室在RISC-V领域的贡献和成果为例

今天演讲的主题内容 (Takeaway)

- RISC-V 已经成为未来的主流架构
 - 任何芯片公司都应投入；任何软件公司都需要适配；从业者应该立刻开始学习
- 拥抱开源软件并积极成为贡献者是企业生存、胜出的必要条件
 - 要么积极融入开源社区，要么被开源软件吞噬；不被upstream支持等于活在ICU
- RISC-V 软件生态领域依然有大量机会可以把握
 - 以中国科学院软件研究所PLCT实验室在RISC-V领域的贡献和成果为例

为什么突然冒出来一个 RISC-V ?

为什么突然冒出来一个 RISC-V？

- 其实收钱的、开源自由不要钱的ISA，过去现在未来都有

为什么突然冒出来一个 RISC-V ?

- 其实收钱的、开源自由不要钱的ISA，过去现在未来都有
 - OpenRISC、SPARC V32、以及 x86 的开源core都是有的，也默默流片不少

为什么突然冒出来一个 RISC-V？

- 其实收钱的、开源自由不要钱的ISA，过去现在未来都有
 - OpenRISC、SPARC V32、以及x86的开源core都是有的，也默默流片不少
- 国内也有不少指令集，例如阿里巴巴收购的中天微，就维护着C-SKY指令集

为什么突然冒出来一个 RISC-V？

- 其实收钱的、开源自由不要钱的ISA，过去现在未来都有
 - OpenRISC、SPARC V32、以及x86的开源core都是有的，也默默流片不少
- 国内也有不少指令集，例如阿里巴巴收购的中天微，就维护着C-SKY指令集
 - 仅仅国内就有好几家（曾今和现在）有自己维护的指令集，同时所有ISA国内都有备胎

为什么突然冒出来一个 RISC-V？

- 其实收钱的、开源自由不要钱的ISA，过去现在未来都有
 - OpenRISC、SPARC V32、以及x86的开源core都是有的，也默默流片不少
- 国内也有不少指令集，例如阿里巴巴收购的中天微，就维护着C-SKY指令集
 - 仅仅国内就有好几家（曾今和现在）有自己维护的指令集，同时所有ISA国内都有备胎
- RISC-V是众多新ISA脱颖而出的一个，「大家」都买账

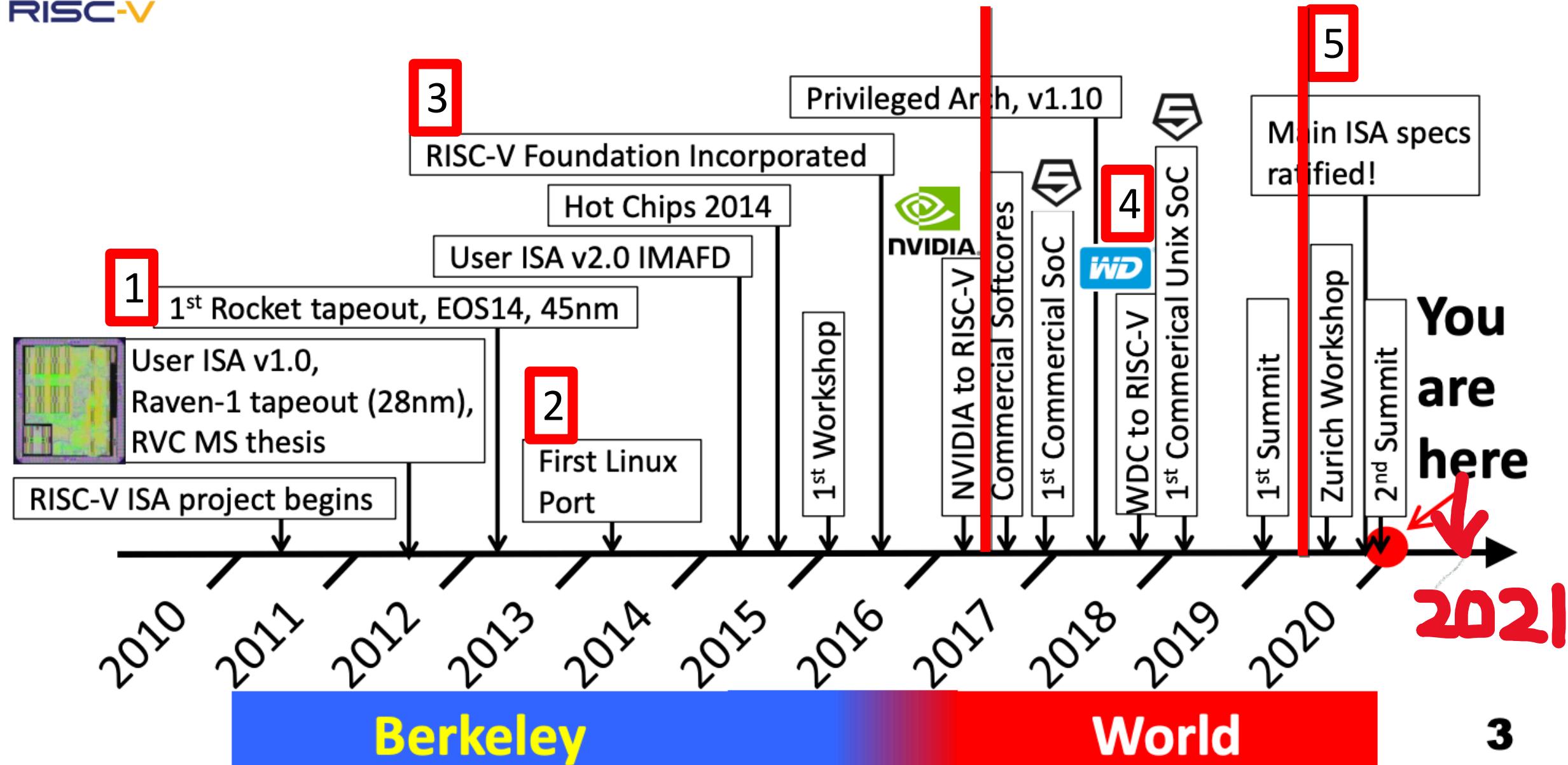
为什么突然冒出来一个 RISC-V？

- 其实收钱的、开源自由不要钱的ISA，过去现在未来都有
 - OpenRISC、SPARC V32、以及x86的开源core都是有的，也默默流片不少
- 国内也有不少指令集，例如阿里巴巴收购的中天微，就维护着C-SKY指令集
 - 仅仅国内就有好几家（曾今和现在）有自己维护的指令集，同时所有ISA国内都有备胎
- RISC-V是众多新ISA脱颖而出的一个，「**大家**」都买账

↑
思考：“大家”是谁？



RISC-V Timeline



RISC-V 已经成为未来的主流架构

是因为 RISC-V 在合适的时间、合适的地点、提出了足够好的设计

- 20年前，摩尔定律开始失效，Domain-Specific Architecture 时代到来
- DSA 需要添加定制的指令、显著缩短的研发周期、大量的设计选择权衡
 - ❖ 上手容易、基础指令简单、有编译器和操作系统支持、预留了大量编码空间

RISC-V 已经成为未来的主流架构

是因为 **RISC-V** 在**合适的时间**、**合适的地点**、提出了足够好的设计

- 20年前，摩尔定律开始失效，**Domain-Specific Architecture** 时代到来
- **DSA** 需要添加定制的指令、显著缩短的研发周期、大量的设计选择权衡
 - ❖ 上手容易、基础指令简单、有编译器和操作系统支持、预留了大量编码空间

问：为什么在**RISC-V**之前那么多个**ISA**都没有成为主流？

RISC-V 已经成为未来的主流架构

是因为 RISC-V 在合适的时间、**合适的地点**、提出了足够好的设计

- 2010年在伯克利，Krstic 教授的研究组
 - ❖ 需要个简单、免费、自由的ISA进行科研，作为更宏大的芯片设计创新项目的一部分
 - ❖ 支持全新的ISA需要借助众多基础软件的开发者的力量，并进行有效地协作

RISC-V 已经成为未来的主流架构

是因为 RISC-V 在合适的时间、**合适的地点**、提出了足够好的设计

- 2010年在伯克利，Krsti 教授的研究组
 - ❖ 需要个简单、免费、自由的ISA进行科研，作为更宏大的芯片设计创新项目的一部分
 - ❖ 支持全新的ISA需要借助众多基础软件的开发者的力量，并进行有效地协作

如果 RISC-V 诞生在中国，我们还需要补上哪些技术人才？

RISC-V 已经成为未来的主流架构

是因为 RISC-V 在合适的时间、合适的地点、提出了足够好的设计

RISC-V 已经成为未来的主流架构

是因为 RISC-V 在合适的时间、合适的地点、提出了足够好的设计

- 提出了模块化设计的概念
 - ❖ 最基础的RV32I仅使用了40+条指令编码：一名大三学生一周就能完成一个能跑的设计
- 提供了高度灵活的配置空间

RISC-V 已经成为未来的主流架构

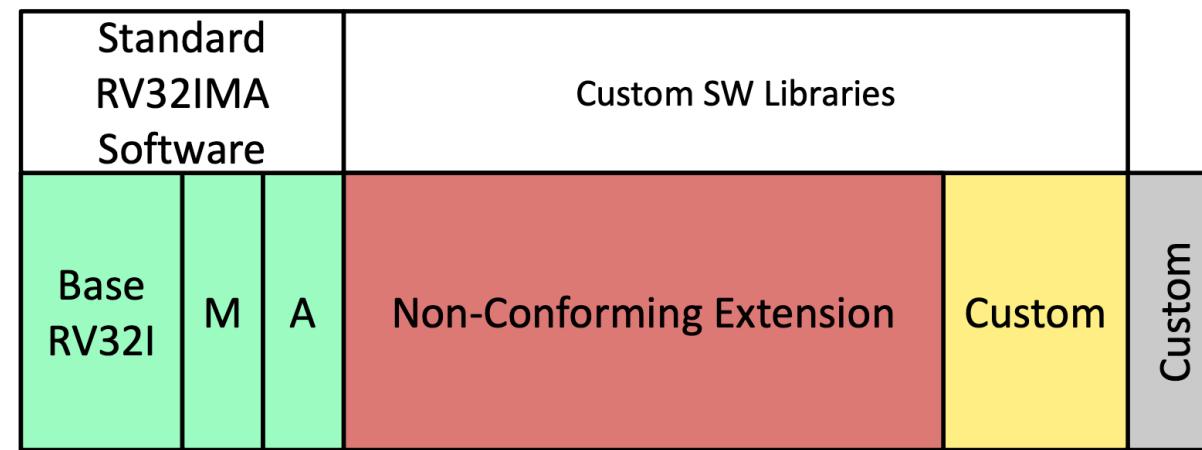
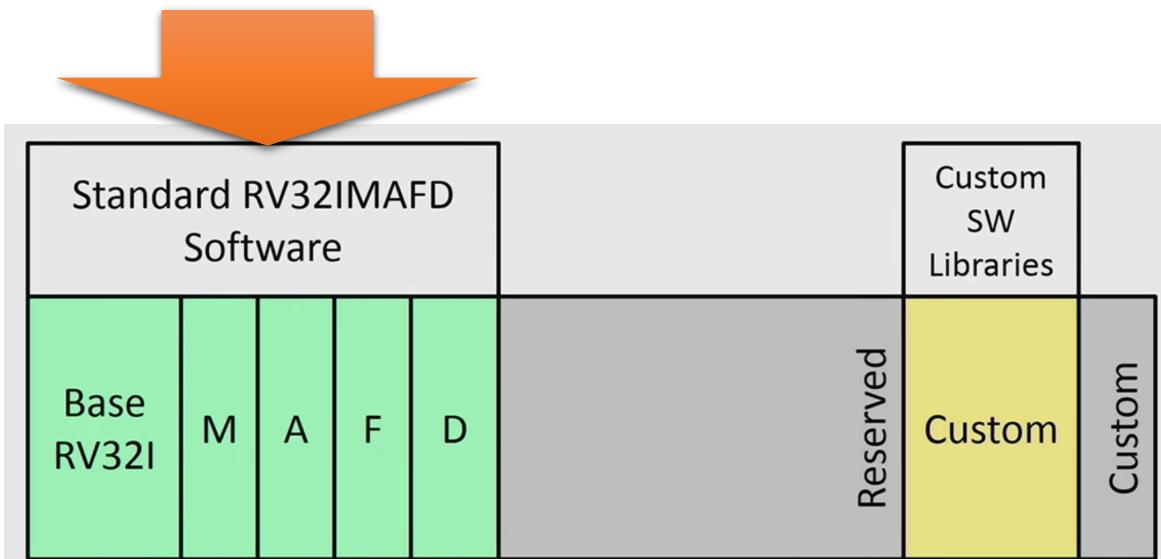
是因为 RISC-V 在合适的时间、合适的地点、提出了足够好的设计

- 提出了模块化设计的概念
 - ❖ 最基础的RV32I仅使用了40+条指令编码：一名大三学生一周就能完成一个能跑的设计
- 提供了高度灵活的配置空间
- 同时平台标准提供了足够的软件支持

RISC-V 已经成为未来的主流架构

是因为 RISC-V 在合适的时间、合适的地点、提出了足够好的设计

- 提出了模块化设计的概念
 - ❖ 最基础的RV32I仅使用了40+条指令编码：一名大三学生一周就能完成一个能跑的设计
- 提供了高度灵活的配置空间
- 同时平台标准提供了足够多的软件支持



RISC-V 自身做对了什么

RISC-V 自身做对了什么

- 清晰的定位：指令集自由开放，作为硬件和软件之间“交谈”的接口

RISC-V 自身做对了什么

- 清晰的定位：指令集自由开放，作为硬件和软件之间“交谈”的接口
 - ❖ 在 RISC-V 之前，主流ISA对硬件自由创新禁止，尤其是64位
 - ❖ 相比于 OpenRISC、SPARC 等架构，RISC-V 提供了更好的可扩展的设计

RISC-V 自身做对了什么

- 清晰的定位：指令集自由开放，作为硬件和软件之间“交谈”的接口
 - ❖ 在 RISC-V 之前，主流ISA对硬件自由创新禁止，尤其是64位
 - ❖ 相比于 OpenRISC、SPARC 等架构，RISC-V 提供了更好的可扩展的设计





Software



Hardware

Open-source cores:
Rocket, BOOM, RI5CY,
Ariane, PicoRV32, Piccolo,
SCR1, Shakti, Serv, Swerv,
Hummingbird, ...

RISC-V Ecosystem

Open-source software:
GCC, binutils, glibc, Linux, BSD,
LLVM, QEMU, FreeRTOS,
ZephyrOS, LiteOS, SylixOS, ...

Commercial software:
Lauterbach, Segger, IAR,
Micrium, ExpressLogic, Ashling,
AntMicro, Imperas, UltraSoC ...

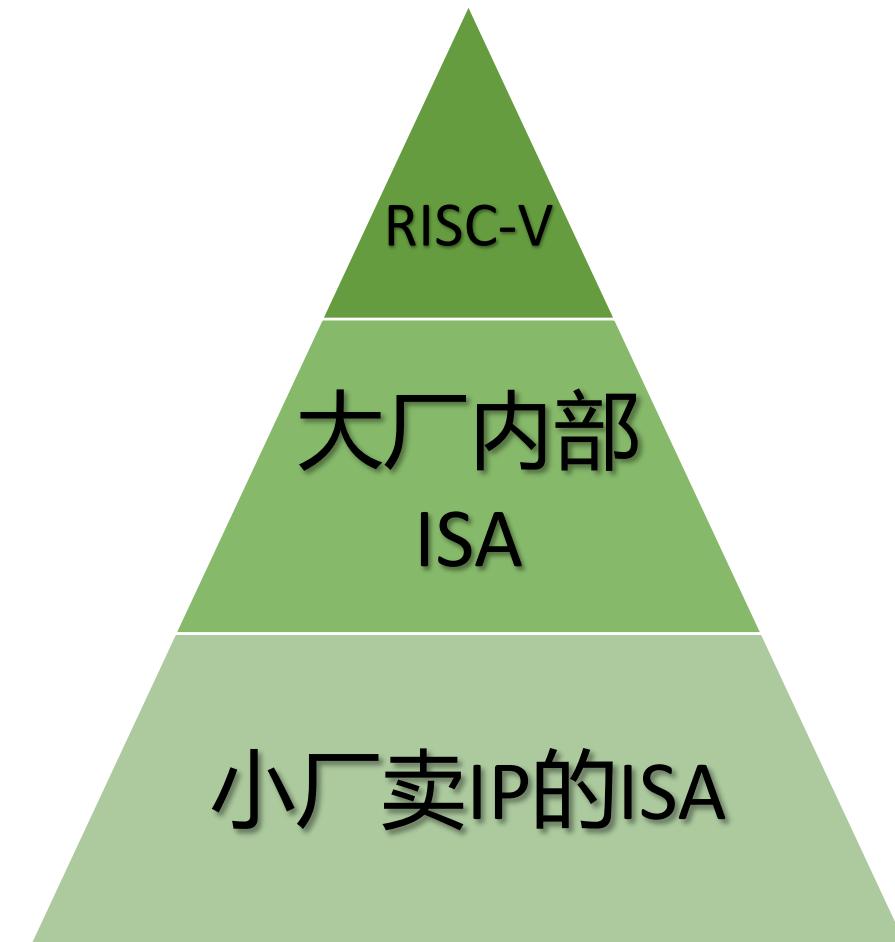
ISA specification

Golden Model

Compliance

Commercial core providers:
Alibaba, Andes, Bluespec,
Cloudbear, Codasip, Cortus,
InCore, Nuclei, SiFive,
Syntacore, ...

Inhouse cores:
Nvidia, WD, +others



Software



Hardware

RISC-V Ecosystem

Open-source software:
GCC, binutils, glibc, Linux, BSD,
LLVM, QEMU, FreeRTOS,
ZephyrOS, LiteOS, SylixOS, ...

Commercial software:
Lauterbach, Segger, IAR,
Micrium, ExpressLogic, Ashling,
AntMicro, Imperas, UltraSoC ...

ISA specification

Golden Model

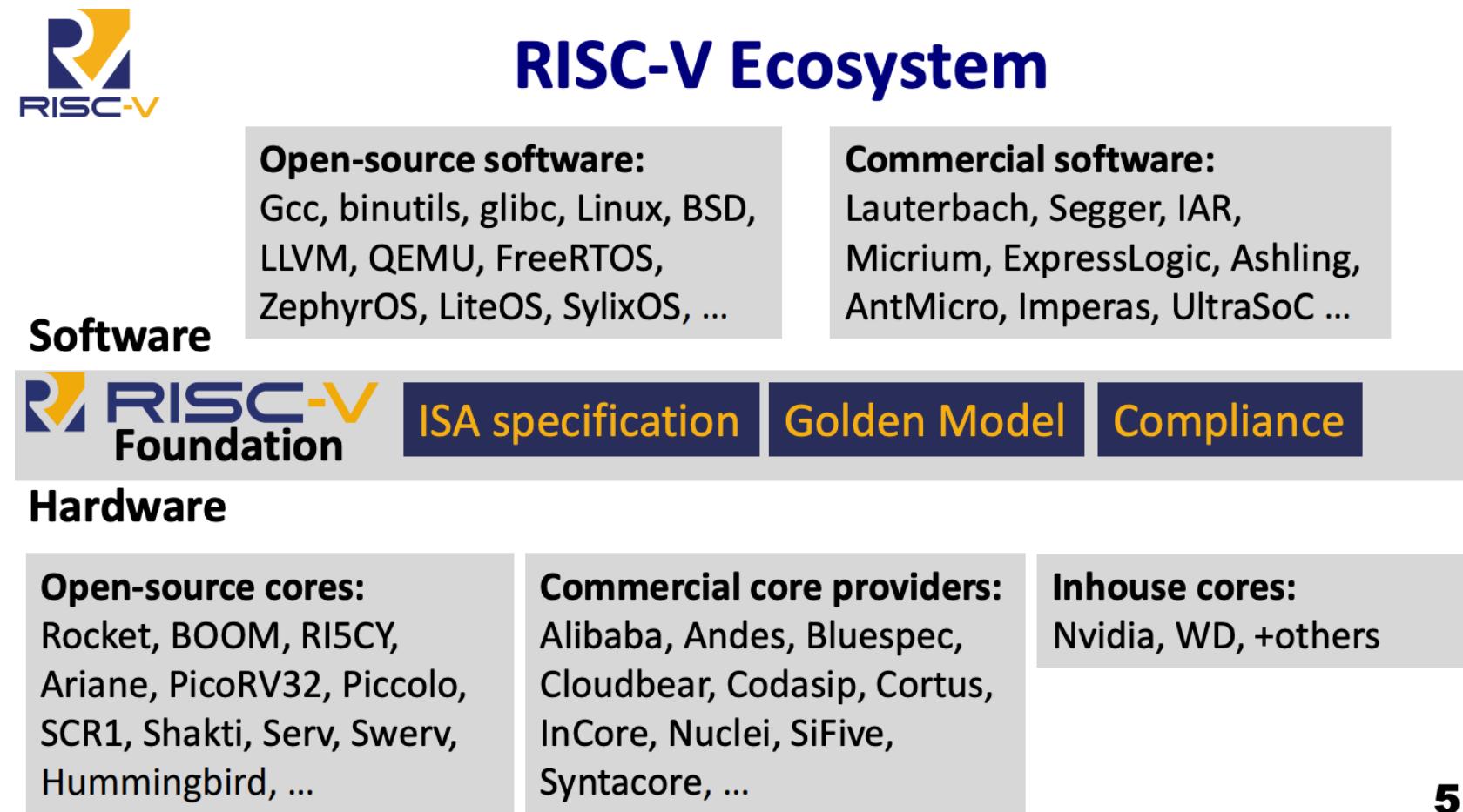
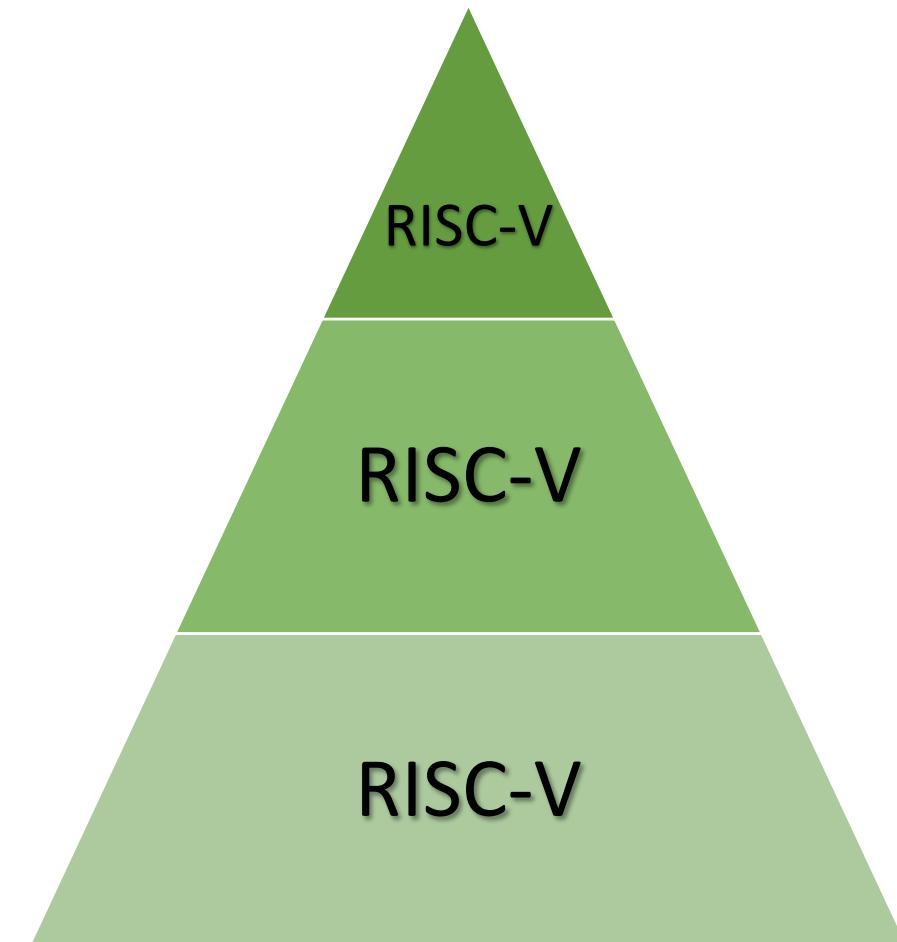
Compliance

Open-source cores:
Rocket, BOOM, RI5CY,
Ariane, PicoRV32, Piccolo,
SCR1, Shakti, Serv, Swerv,
Hummingbird, ...

Commercial core providers:
Alibaba, Andes, Bluespec,
Cloudbear, Codasip, Cortus,
InCore, Nuclei, SiFive,
Syntacore, ...

Inhouse cores:
Nvidia, WD, +others

内卷警告：所有还在做自研的指令集的朋友，该思考下出路了



RISC-V 自身做对了什么

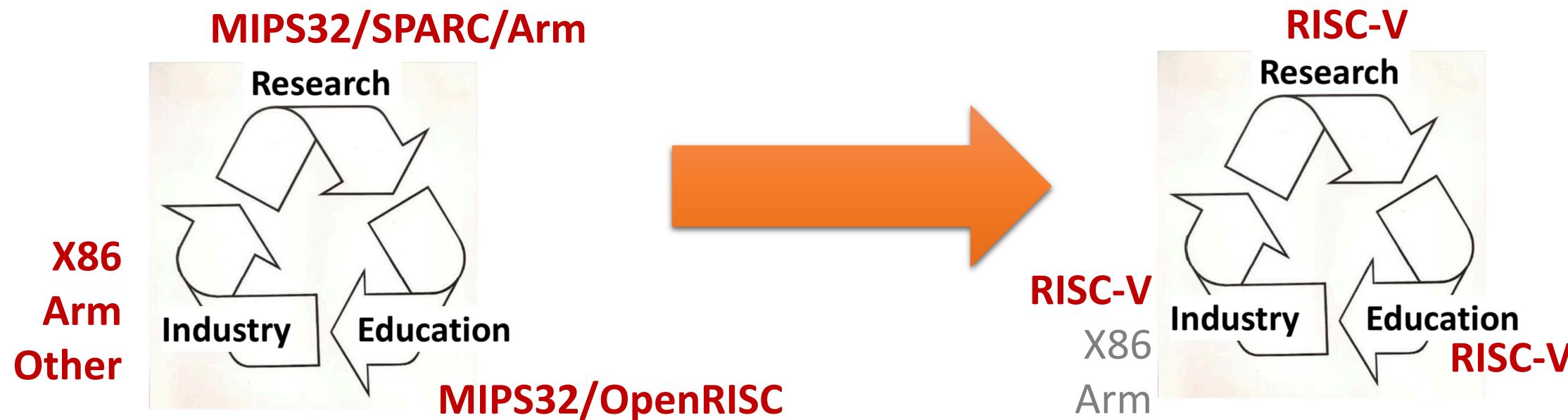
- 清晰的定位：指令集自由开放，作为硬件和软件之间“交谈”的接口
- **设计优美简洁，支持了从教学、研究到工业界的无缝切换**

RISC-V 自身做对了什么

- 清晰的定位：指令集自由开放，作为硬件和软件之间“交谈”的接口
- **设计优美简洁，支持了从教学、研究到工业界的无缝切换**
 - ❖ 前人成功的案例，有Linux和LLVM
 - ❖ 在这之前，教育用MIPS/SPARC，进入产业之后变成X86/Arm，研究成果无法直接转化

RISC-V 自身做对了什么

- 清晰的定位：指令集自由开放，作为硬件和软件之间“交谈”的接口
- 设计优美简洁，支持了从教学、研究到工业界的无缝切换
 - ❖ 前人成功的案例，有Linux和LLVM
 - ❖ 在这之前，教育用MIPS/SPARC，进入产业之后变成X86/Arm，研究成果无法直接转化



图源：Krstic 演讲：RISC-V State of Union

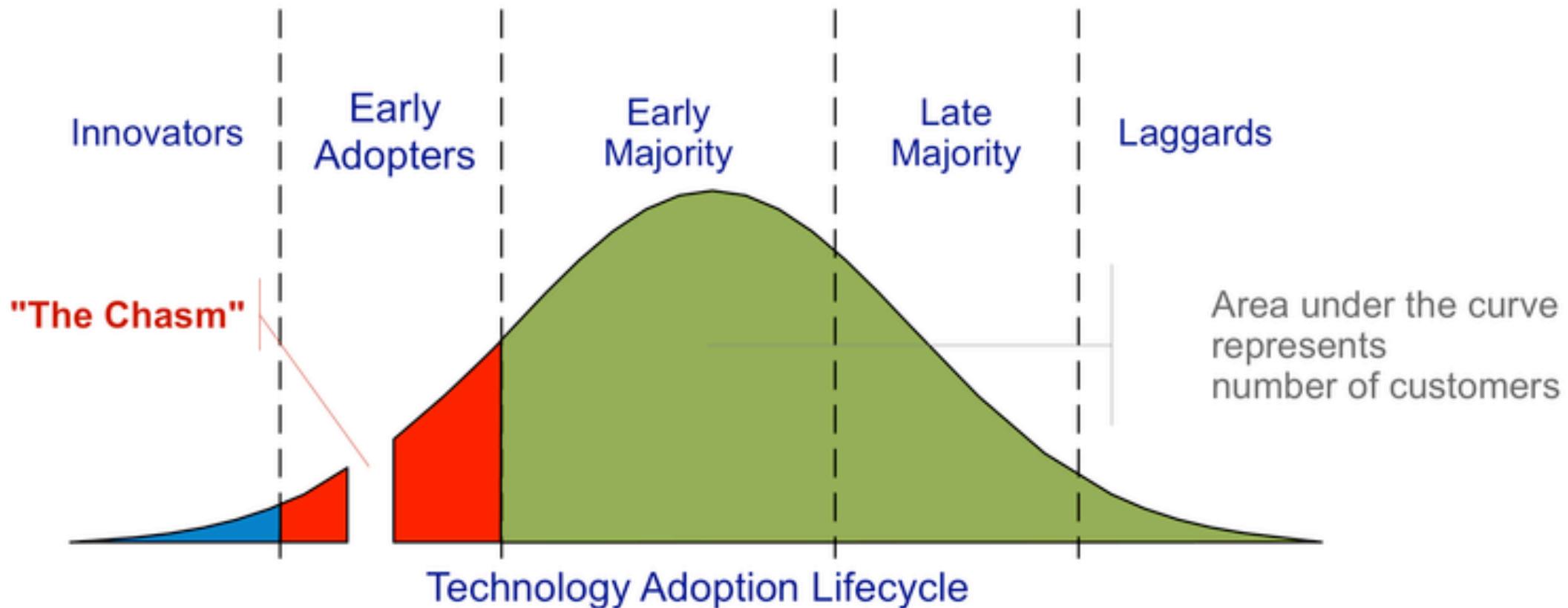
RISC-V 自身做对了什么

- 清晰的定位：指令集自由开放，作为硬件和软件之间“交谈”的接口
- 设计优美简洁，支持了从教学、研究到工业界的无缝切换
- 左手成立RISC-V基金会，右手创办SiFive

RISC-V 自身做对了什么

- 左手成立RISC-V基金会，右手创办SiFive

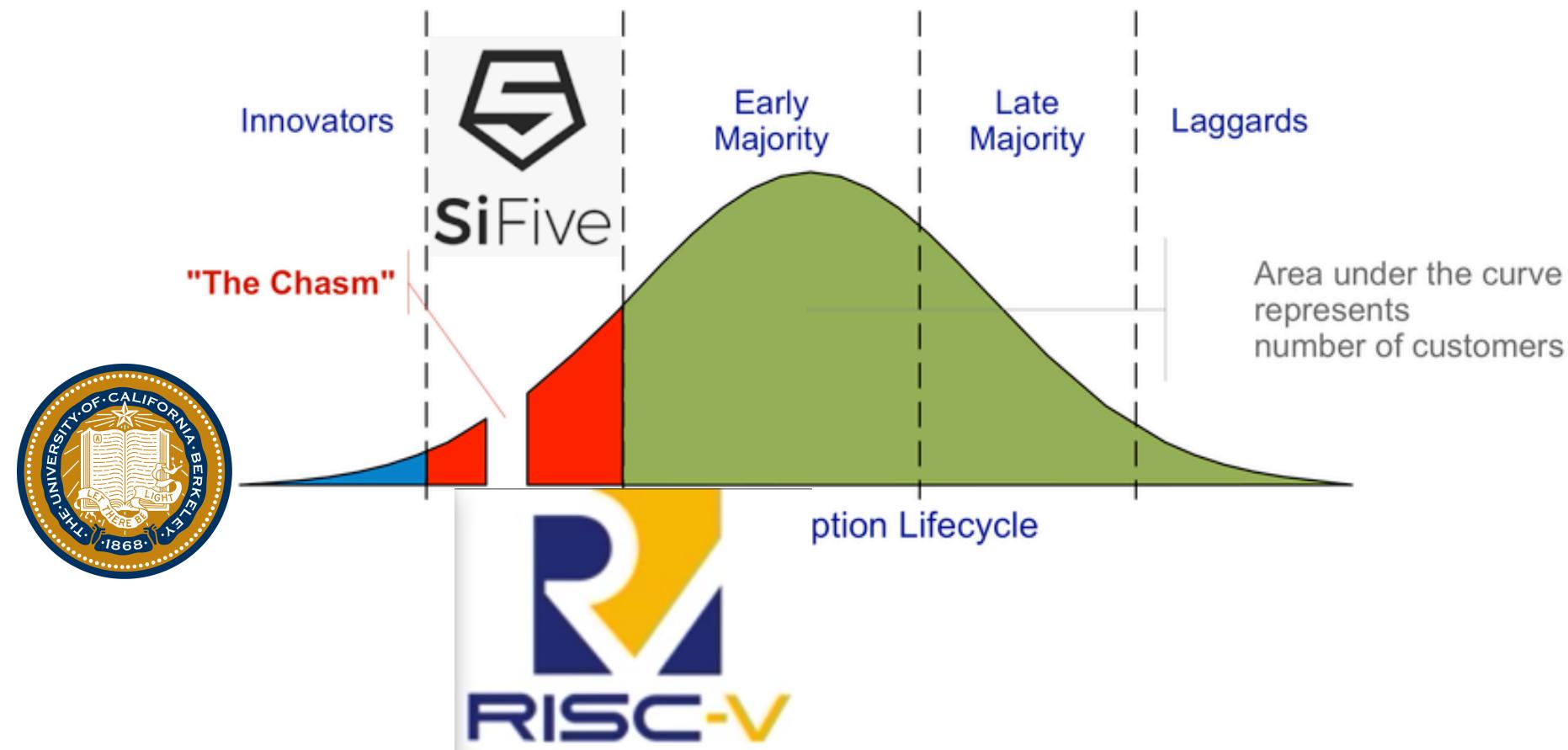
- ❖ 兼顾ISA中立和推动商业化



RISC-V 自身做对了什么

- 左手成立RISC-V基金会，右手创办SiFive

- ❖ 兼顾ISA中立和推动商业化



RISC-V 自身做对了什么

- 清晰的定位：指令集自由开放，作为硬件和软件之间“交谈”的接口
- 设计优美简洁，支持了从教学、研究到工业界的无缝切换
- 左手成立RISC-V基金会，右手创办SiFive
- 顺应了时代的潮流：DSA导致的设计方式的改变，引发对ISA的新需求
 - ❖ “RISC-V 的成功是因为（出现了）新的商业模式” —— Krste 教授

那么，为何是伯克利搞成了RISC-V，而不是中国的高校或企业？

那么，为何是伯克利搞成了RISC-V，而不是中国的高校或企业？

- 一呼天下应的出身（代表着技术领域的信用货币）
- 有钱、正确的烧钱（进行了十几次流片*）
- 符合时宜的设计理念（模块化、定制化）
- 简洁（连一个本科生都可以接近独立的设计出短流水线MCU）
- 非常注重软件栈、借力
- 天下桃李，悉在公门矣

<https://blog.csdn.net/moxibingdao/article/details/106667615>

那么，为何是伯克利搞成了RISC-V，而不是中国的高校或企业？
我们缺少什么？如何补缺？如何追赶？

- 一呼天下应的出身（代表着技术领域的信用货币）
- 有钱、正确的烧钱（进行了十几次流片*）
- 符合时宜的设计理念（模块化、定制化）
- 简洁（连一个本科生都可以接近独立的设计出短流水线MCU）
- 非常注重软件栈、借力
- 天下桃李，悉在公门矣

<https://blog.csdn.net/moxibingdao/article/details/106667615>

RISC-V 已经成为未来的主流架构：要么在车里，要么在车底



今天演讲的主题内容 (Takeaway)

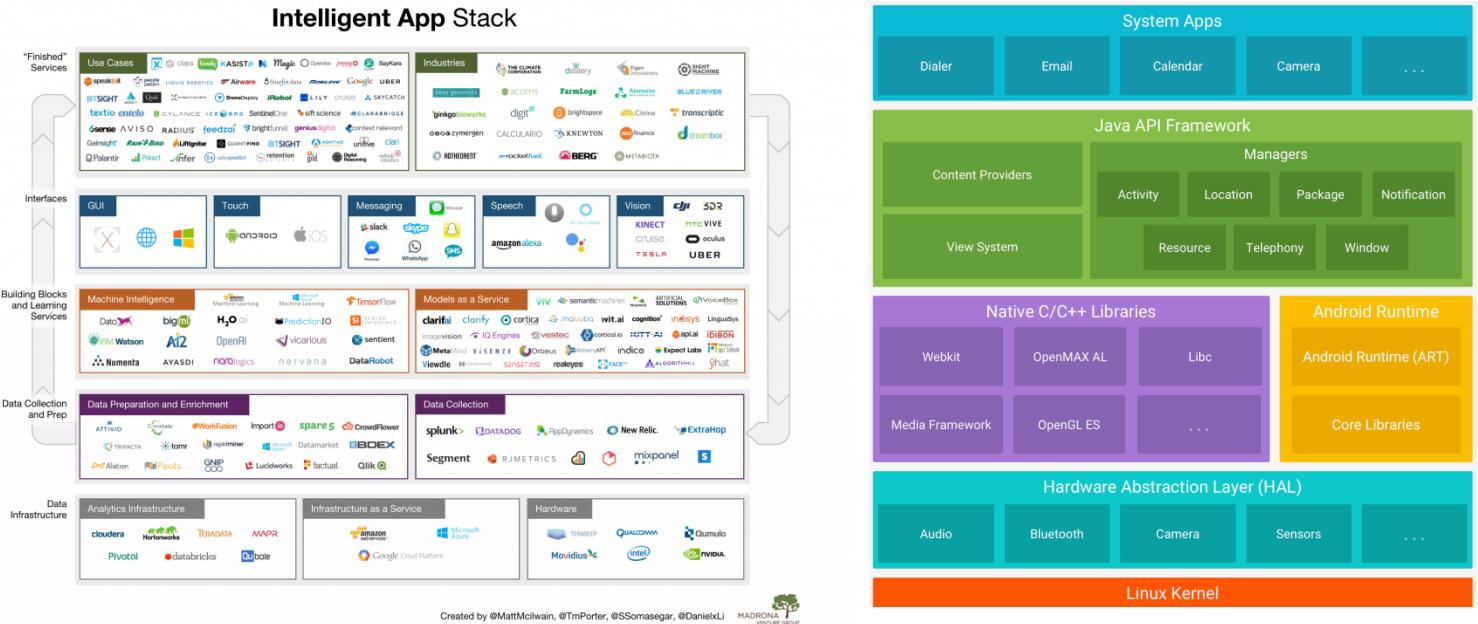
- RISC-V 已经成为未来的主流架构
 - 任何芯片公司都应投入；任何软件公司都需要适配；从业者应该立刻开始学习
- 拥抱开源软件并积极成为贡献者是企业生存、胜出的必要条件
 - 要么积极融入开源社区，要么被开源软件吞噬；不被upstream支持等于活在ICU
- RISC-V 软件生态领域依然有大量机会可以把握
 - 以中国科学院软件研究所PLCT实验室在RISC-V领域的贡献和成果为例

拥抱开源软件：软件吞噬世界，开源软件吞噬软件

- 1 <https://developer.ibm.com/blogs/how-open-source-software-is-eating-the-world/>
- 2 <https://algorithmia.com/blog/wp-content/uploads/2016/06/Screen-Shot-2016-06-08-at-3.35.53-PM-1024x730.png>
- 3 <https://developer.android.com/guide/platform>

拥抱开源软件：软件吞噬世界，开源软件吞噬软件

是什么让开源不可避免？

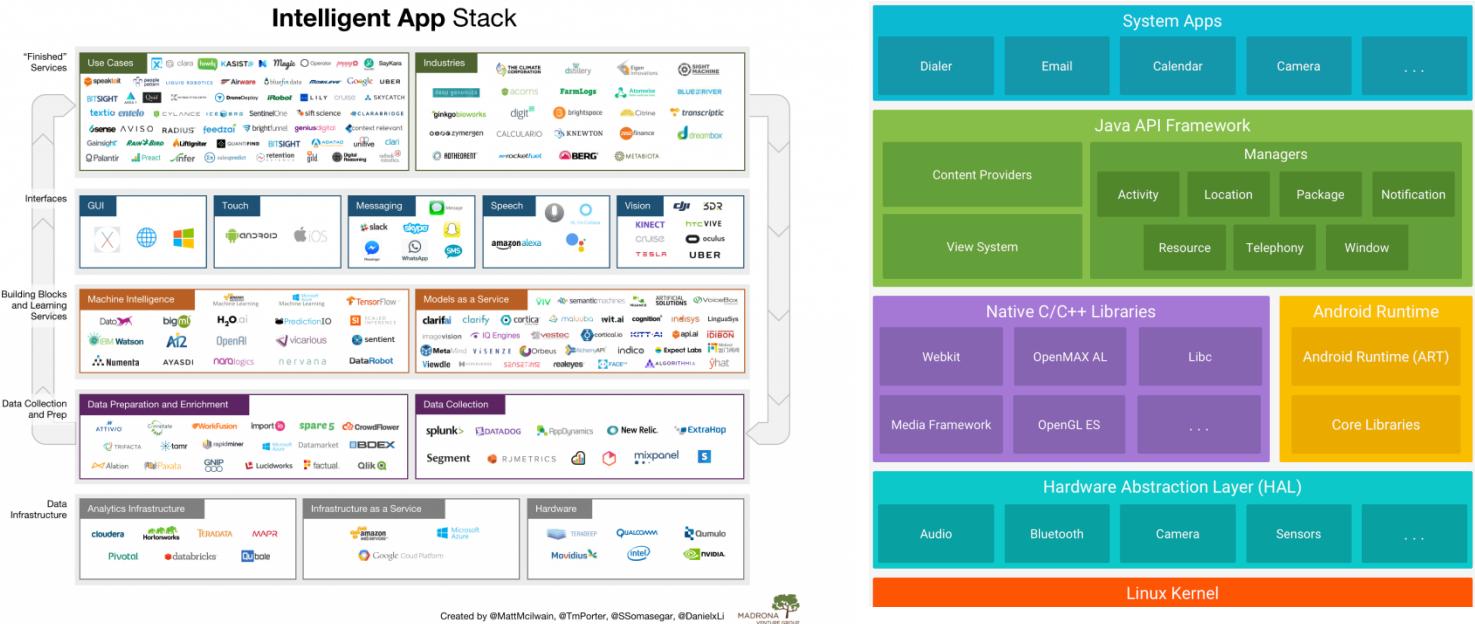


- 1 <https://developer.ibm.com/blogs/how-open-source-software-is-eating-the-world/>
- 2 <https://algorithmia.com/blog/wp-content/uploads/2016/06/Screen-Shot-2016-06-08-at-3.35.53-PM-1024x730.png>
- 3 <https://developer.android.com/guide/platform>

拥抱开源软件：软件吞噬世界，开源软件吞噬软件

是什么让开源不可避免？

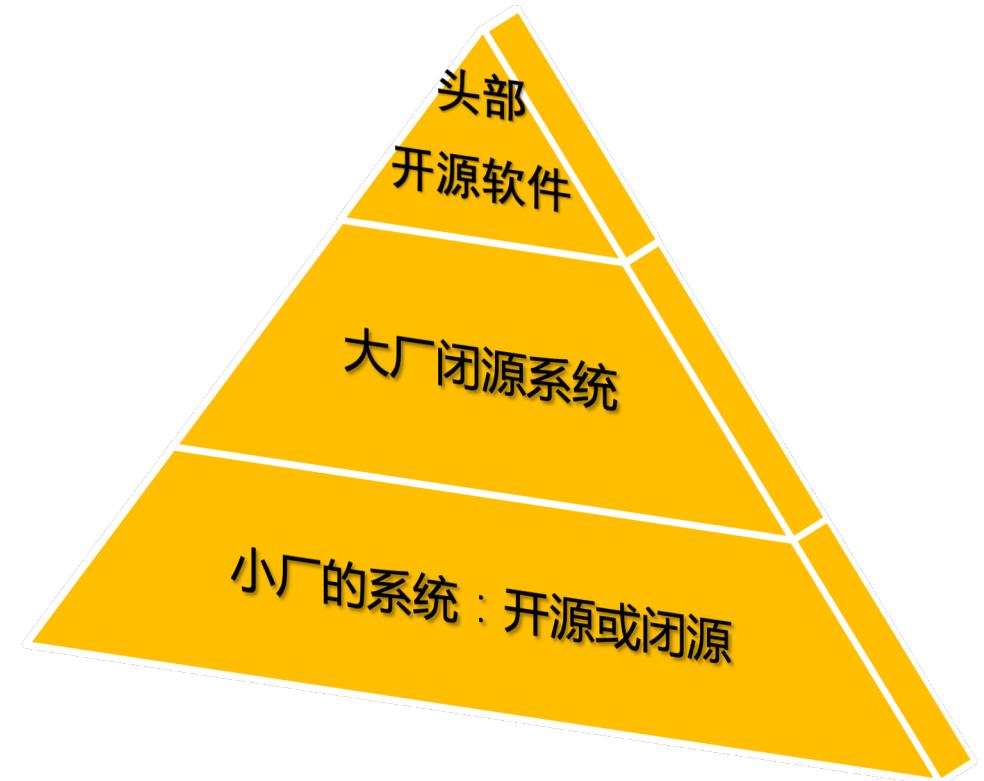
软件系统的规模



- 1 <https://developer.ibm.com/blogs/how-open-source-software-is-eating-the-world/>
- 2 <https://algorithmia.com/blog/wp-content/uploads/2016/06/Screen-Shot-2016-06-08-at-3.35.53-PM-1024x730.png>
- 3 <https://developer.android.com/guide/platform>

拥抱开源软件，或者被吞噬

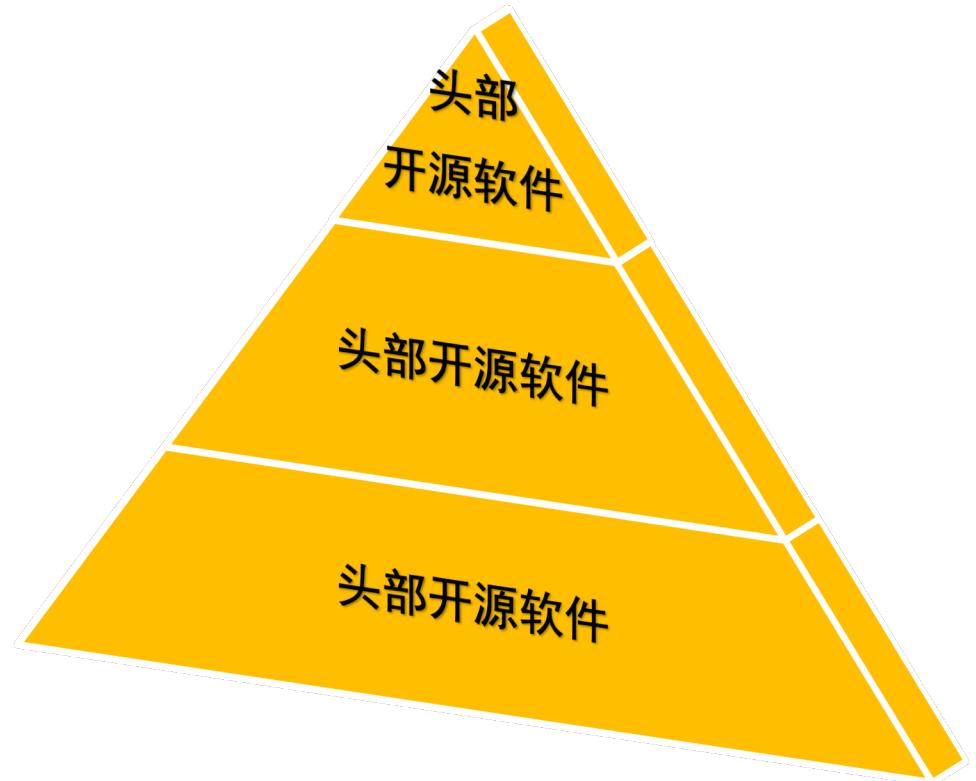
- 开源软件正在吞噬世界
 - ❖ **GNU/Linux、FreeRTOS、*BSD**
 - ❖ **Hadoop/Spark、Docker/k8s.....**
- 不被开源软件支持将付出极大的维护成本
 - ❖ **进而失去竞争优势被淘汰**



拥抱开源软件，或者被吞噬

开源软件的世界很残酷的，只有头部的社区容易吸引用户和贡献者

- 开源软件正在吞噬世界
 - ❖ **GNU/Linux、FreeRTOS、*BSD**
 - ❖ **Hadoop/Spark、Docker/k8s.....**
- 不被开源软件支持将付出极大的维护成本
 - ❖ **进而失去竞争优势被淘汰**



拥抱开源软件

- 成功的ISA架构在开源软件领域都有着巨大的投入
 - ❖ Intel、Arm 有着长期巨大的投入，投入不够一定会被打败
 - ❖ 就像是武林高手对掌拼内力，谁先撤力谁出大问题

拥抱开源软件

- 成功的ISA架构在开源软件领域都有着巨大的投入
 - ❖ Intel、Arm 有着长期巨大的投入，投入不够一定会被打败
- RISC-V 做对了两件事让其成为未来的主流
 - ❖ 早期的团队研发实力和伯克利的领袖光环，实现了基本工具链支持
 - ❖ 在2017年之后顺应产业界的需求从芯片创新转向以软件为中心

RISC-V 正确的转向



Changing Priorities



<u>2010</u>	<u>2020</u>
<ol style="list-style-type: none">1. Be simple, efficient, extensible2. Revisit legacy design decisions3. Have basic software <p><i>(computer-architecture-driven project)</i></p>	<ol style="list-style-type: none">1. Run all software2. Be feature complete<ul style="list-style-type: none">➤ see #13. Be stable<ul style="list-style-type: none">➤ see #14. Support innovation<ul style="list-style-type: none">➤ conflicts with #1,#2,#3? <p><i>(software-driven project)</i></p>

RISC-V 正确的转向

或许这就是
国内的技术领袖们
稀缺的远见



Changing Priorities



2010

1. Be simple, efficient, extensible
2. Revisit legacy design decisions
3. Have basic software

(computer-architecture-driven project)

2020

1. Run all software
2. Be feature complete
 - see #1
3. Be stable
 - see #1
4. Support innovation
 - conflicts with #1,#2,#3?

(software-driven project)

RISC-V 正确的转向：或许这是我们（国内）稀缺的远见

- 做硬件（芯片）出身的专家往往看不见软件生态才是决定生态生死的国王

RISC-V 正确的转向：或许这是我们（国内）稀缺的远见

- 做硬件（芯片）出身的专家往往看不见软件生态才是决定生态生死的国王
- 而 RISC-V 初创团队是硬件出身，却能够
在**2010**年就看到软件占据了系统成本的大头
在**2020**年明确提出将从体系架构主导变更为软件主导

RISC-V 正确的转向：或许这是我们（国内）稀缺的远见

- 做硬件（芯片）出身的专家往往看不见软件生态才是决定生态生死的国王
- 而 RISC-V 初创团队是硬件出身，却能够
在**2010**年就看到软件占据了系统成本的大头
在**2020**年明确提出将从体系架构主导变更为软件主导

国内的许多企业，提前跑多年，为什么没有成为RISC-V？

拥抱开源软件：怎么拥抱？

- 开源软件：会跑起来和会自己修改、演化是两个不同的境界
- 充分的消化、吸收，变成自己的知识

拥抱开源软件：怎么拥抱？

- 开源软件：会跑起来和会自己修改、演化是两个不同的境界
 - ❖ 「国外一开源，国内就自主可控」是劣币驱逐良币
 - ❖ 国内想要超越，就不能一直搭顺风车
- 充分的消化、吸收，变成自己的知识
 - ❖ 尊重知识消化吸收的规律，避免拔苗助长
 - ❖ 技术结果导向，在早期识别浑水摸鱼的南郭先生

拥抱开源软件：怎么拥抱？

- 开源软件：会跑起来和会自己修改、演化是两个不同的境界
 - ❖ 「国外一开源，国内就自主可控」是劣币驱逐良币
 - ❖ 国内想要超越，就不能一直搭顺风车：请不要裁掉基础软件工程师！
- 充分的消化、吸收，变成自己的知识
 - ❖ 尊重知识消化吸收的规律，避免拔苗助长
 - ❖ 技术结果导向，在早期识别浑水摸鱼的南郭先生
 - ❖ 能者多得，而不仅仅是能者多劳

面向RISC-V的开源软件生态：国内的贡献度起步迟了

- 重要的开源软件的 RISC-V Ports 都已经有志愿者维护
 - ❖ **GNU Toolchain、glibc/newlib、Linux Kernel、Clang/LLVM**
- 谁在上游 (Upstream) 贡献了代码，谁就掌控了话语权
 - ❖ **欧洲的EPI和ZTH、北美的Google和WDC、印度的IIT、中国台湾的Andes**

面向RISC-V的开源软件生态：国内的贡献度起步迟了

- 重要的开源软件的 RISC-V Ports 都已经有志愿者维护
 - ❖ **GNU Toolchain、glibc/newlib、Linux Kernel、Clang/LLVM**
- 谁在上游 (Upstream) 贡献了代码，谁就掌控了话语权
 - ❖ 欧洲的EPI和ZTH、北美的Google和WDC、印度的IIT、中国台湾的Andes
- 不被上游维护的代码就像是活在ICU里：费用昂贵、死亡率高
 - ❖ 举例：几乎所有国内的芯片厂商和系统厂商

面向RISC-V的开源软件生态：仍有大量机会

面向RISC-V的开源软件生态：仍有大量机会

- 基础软件领域的狂欢：已经很久没有这么多轮子可以发明了
 - ❖ 各种编程语言的编译器 $\times N$

面向RISC-V的开源软件生态：仍有大量机会

- **基础软件领域的狂欢：已经很久没有这么多轮子可以发明了**
 - ❖ 各种编程语言的编译器 × N
 - ❖ 语言虚拟机和运行时 × N

面向RISC-V的开源软件生态：仍有大量机会

■ 基础软件领域的狂欢：已经很久没有这么多轮子可以发明了

- ❖ 各种编程语言的编译器 × N
- ❖ 语言虚拟机和运行时 × N
- ❖ 系统虚拟机、指令集模拟器 × N

面向RISC-V的开源软件生态：仍有大量机会

■ 基础软件领域的狂欢：已经很久没有这么多轮子可以发明了

- ❖ 各种编程语言的编译器 $\times N$
- ❖ 语言虚拟机和运行时 $\times N$
- ❖ 系统虚拟机、指令集模拟器 $\times N$
- ❖ 全系统性能分析和优化、软硬件协同演进的定制和优化
- ❖ 二进制工具 $\times N$

面向RISC-V的开源软件生态：仍有大量机会

■ 基础软件领域的狂欢：已经很久没有这么多轮子可以发明了

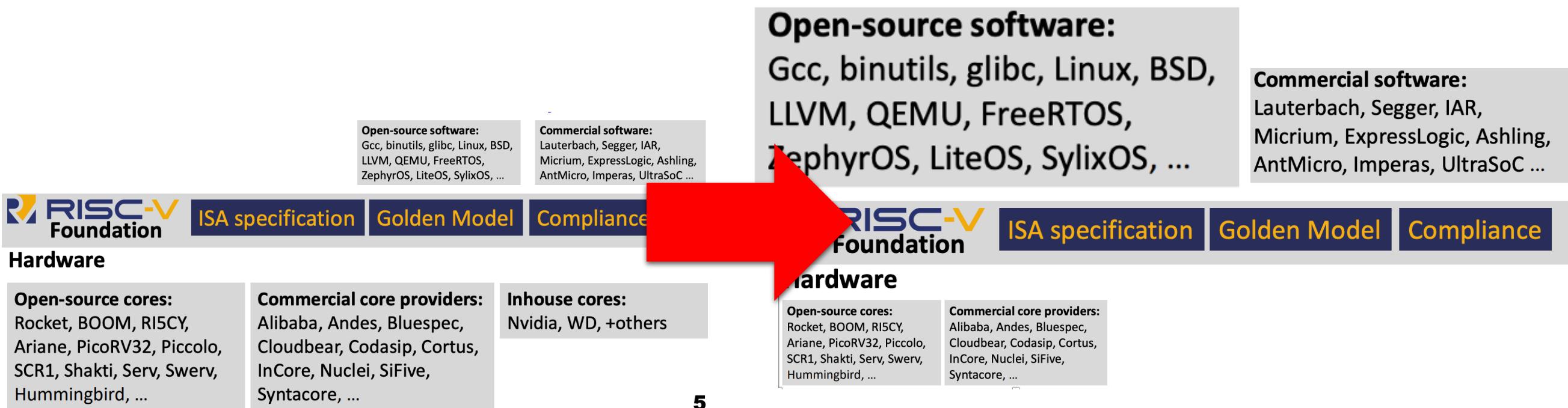
- ❖ 各种编程语言的编译器 $\times N$
- ❖ 语言虚拟机和运行时 $\times N$
- ❖ 系统虚拟机、指令集模拟器 $\times N$
- ❖ 全系统性能分析和优化、软硬件协同演进的定制和优化
- ❖ 二进制工具 $\times N$
- ❖ 全新的系统安全视角和可能性

请务必看看 bunnie 的 keynote https://www.youtube.com/watch?v=zXwy65d_tu8

国内的企业有哪些机会

- MCU领域已经形成了成功的模式
- 通用计算（PC/Server）和AIoT领域正在发力
- 纯CPU创业会很辛苦，因为有大财团互联网公司在旁边准备投你的友商
- 各类AI加速器初创公司还有机会，但是如何面对NVDLA的坚壁清野
- 目前国内的差距主要体现在软件能力，而不是硬件研发能力上(我不清楚)
- 但是纯做开源软件并不能养活自己，这是个问题

RISC-V生态的边界正在经历一个商业场景的变迁：卖硬件到卖系统



今天演讲的主题内容 (Takeaway)

- RISC-V 已经成为未来的主流架构
 - 任何芯片公司都应投入；任何软件公司都需要适配；从业者应该立刻开始学习
- 拥抱开源软件并积极成为贡献者是企业生存、胜出的必要条件
 - 要么积极融入开源社区，要么被开源软件吞噬；不被upstream支持等于活在ICU
- RISC-V 软件生态领域依然有大量机会可以把握
 - 以中国科学院软件研究所PLCT实验室在RISC-V领域的贡献和成果为例

面向RISC-V的开源软件生态：以软件所的贡献为例

■ 基础软件领域的国家队，有义务为国内企业提供「开源软件公共品」

- ❖ 编译器领域：RISC-V 向量扩展的LLVM参考实现
- ❖ 虚拟机领域：V8 for RISC-V （V8 是 Chrome 使用的JS引擎）
- ❖ 模拟器领域：面向国内厂商的QEMU、Spike支持
- ❖ 应用领域：OpenCV for RISC-V
- ❖ 操作系统领域：开源供应链安全计划、点亮计划、暑期2020计划
(软件所操作系统团队成果，非PLCT工作)

PLCT Lab 的使命

程序语言与编译技术实验室（PLCT）致力于成为编译技术领域的开源领导者，推进开源工具链及运行时系统等软件基础设施的技术革新，具备主导开发和维护重要基础设施的技术及管理能力。与此同时，努力成为编译领域培养尖端人才的黄埔军校，推动先进编译技术在国内的普及和发展。

PLCT 重点项目：RVV-LLVM 已开源、已跟上游合作

<https://github.com/isrc-cas/rvv-llvm>

- 实现RISC-V 向量扩展指令集支持，维护适用于国内RISC-V芯片厂商的定制版本
- 以RVV作为切入点，逐步参与RISC-V基金会技术决策
- 为本土小芯片厂商提供工具链及开发工具的支持，填补短板
- 作为实战培训项目培养编译器开发新员工并壮大团队



PLCT 重点项目：V8 for RISC-V已开源、已跟上游合作

<https://github.com/v8-riscv/v8>

- Google V8 是目前全球占有率第一大 JavaScript 引擎，是 Node.js 生态领域的基础
 - ❖ V8 和 Node.js 的 RISC-V 支持被列在了 RISC-V 基金会的官网许愿清单中
- 建立一支掌握虚拟机关键技术的小队，能够为国内企业提供支持
- 尽可能多的培养掌握虚拟机开发能力的新人



PLCT 重点项目：面向国内RISC-V厂商的QEMU支持计划

<https://github.com/isrc-cas/plct-qemu>

- 为目前国内RISC-V初创企业提供QEMU模拟器的技术支持
 - ❖ 为芯来科技开发的设备提供QEMU模拟实现
 - ❖ 已开源、准备推送上游
- 建立一支有能力快速掌握QEMU核心技术的小队，并形成培养梯队
- 为国内芯片厂商输送模拟器、仿真器方面的技术骨干



PLCT 培育项目：OpenCV for RISC-V

<https://github.com/opencv/opencv>

- 融入OpenCV社区，成为OpenCV社区在RV方面的贡献者
- 与OpenCV的国内研发团队形成战略合作，相互扶持
- 初步完成 RISC-V 向量扩展的支持，随 v4.5 之后的版本发布（大概）
 - ❖ 正在进行进一步的bugfix和性能的优化
 - ❖ 由于 RISC-V 向量扩展仍在演化中，OpenCV项目也会持续更新和维护



敬请关注：开源软件供应链点亮计划（软件所操作系统团队成果，非PLCT工作）

对标谷歌暑期编程计划 GSoC

- **活动规模**：34 所高校、42 个开源社区、92 家组织机构、
246 位社区核心开发人员（社区导师）、388 个开源项目
- **技术范围**：8 大技术方向，操作系统、内核与编译器、人工智能、分布式系统、
大数据与数据管理、容器与虚拟化、Web 技术、编程开发
- **华为参与**：7个开源社区（占比16.7%），126个开源项目（占比32.4%），
5大技术方向（操作系统、内核与编译器、容器与虚拟化、分布
式系统、人工智能）
- **已有成果**：146 位同学提交了结项报告，已有 3 位同学入选 Apache APISIX 及
社区 Committer

The screenshot shows the homepage of the 'Open Source Promotion Plan - Summer 2020' website. At the top, there is a navigation bar with links for 首页 (Home), 开源社区 (Open Source Community), 活动规划 (Activity Planning), 活动进程 (Activity Progress), 大咖说开源 (Open Source Experts), and 帮助 (Help). The main title 'ISCAS & openEuler Community Open Source Promotion Plan - Summer 2020' is displayed prominently. Below the title, a sub-section titled '开源软件供应链点亮计划鼓励大家关注开源软件和开源社区，发掘和培养更多优秀的开发者。10月底，组委会将公告结项评审结果，敬请关注网站更新' provides information about the program's goals and schedule. On the left side, there is an illustration of a person working on a laptop. On the right side, another illustration shows a person working on a laptop. Below the illustrations, there are three blue rectangular boxes listing key dates: '6月3日-6月5日 暑期2020-活动宣讲直播' (Viewing back), '9月30日 学生提交最终项目报告截止日期' (Activity Progress), and '11月14日 开源软件供应链2020峰会 盛大开幕' (Invited to attend).

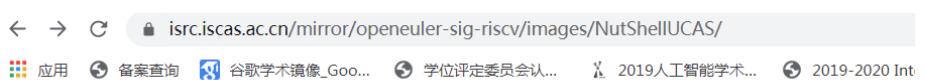
暑期2020活动已经圆满收官，明年将重点支持RISC-V生态内的开源软件

操作系统发行版：移植openEuler支持国科大“一生一芯”果壳处理器

(软件所操作系统团队成果，非PLCT工作)

■ 社区发布

- ❖ openEuler社区: <https://gitee.com/openeuler/community/>
- ❖ SIG-RISC-V工作组 : <https://gitee.com/openeuler/RISC-V/>
- ❖ 中科院软件所托管镜像：
<https://isrc.icsas.ac.cn/mirror/openeuler-sig-riscv/images/NutShellUCAS>



Index of /images/NutShellUCAS/

...		
Oboot2bb1.JPG	25-Sep-2020 02:48	43465
1welcome2openEuler20_03LTS.JPG	25-Sep-2020 02:48	176044
2loginintyPS0.png	25-Sep-2020 02:48	158086
3PYNQ-Z2.jpeg	25-Sep-2020 02:56	124603
README.txt	25-Sep-2020 11:49	2371
RV_BOOT_UCAS_COOSCA1.0_V1.BIN	25-Sep-2020 02:16	8969492
oe-UCAS_COOSCA1.0-rootfs.v1.tar.gz	25-Sep-2020 03:31	380382177

1. Descriptions: materials preparation

1.1 The processor (PL-Programmable Logic bitstream)
Please refer to the NutShell site <https://github.com/OSCPU/NutShell>.
You can use the packaged hardware-related binary file <https://github.com/OSCPU/NutShell/blob/master/fpga/boot/pynq/0>
You can also try to make from NutShell source to get the PL and FSEL.

1.2 RV_BOOT_UCAS_COOSCA1.0_V1.BIN
The oe OS bootloader(bb1) and oe OS kernel ported for NutShell UCAS COOSCA1.0.

1.3 oe-UCAS_COOSCA1.0-rootfs.v1.tar.gz
The roots of the operating system ported for NutShell UCAS COOSCA1.0.

1.4 FPGA
Xilinx PYNQ-Z2, as shown by picture 3PYNQ-Z2.jpeg.

2. How to run the ported openEuler OS on the NutShell UCAS COOSCA1.0

2.1 Format microSD card
We suggest the capacity of SD card is at least 6GB. Format the SD card into two partitions.
If SD card is accessed by SD ADAPTER, the partitions are probably named as /dev/mmcblk0p1 and /dev/mmcblk0p2, but if by USB reader, they are probably named as /dev/sd1 e.g. /dev/sd1, and /dev/sd2. Format the first partition by FAT32 with suggested size 100MB or somewhat larger, and format the second partition by ext4.

2.2 The hardware related (Including FSEL and UCAS COOSCA1.0 PL bitstream)
Mount the /dev/mmcblk0p1 on some dir, then copy the packaged hardware related binary file into the partition mmcblk0p1.

2.3 The bootloader and kernel
Copy RV_BOOT_UCAS_COOSCA1.0_V1.BIN into the /dev/mmcblk0p1 partition.

2.4 The rootfs
Extract oe-UCAS_COOSCA1.0-rootfs.v1.tar.gz into the /dev/mmcblk0p2 partition.

2.5 Boot the PYNQ-Z2
Provide power by Micro-USB, then start the board by put the power switch (near the Micro-USB) on.
For more information about the board, please refer to the PYNQ-Z2 manual.

2.6 Get a console from ttyPS0
Execute command below to get interaction with the OS / Hardware by serial port:
sudo picocom -b 115200 /dev/ttyUSB1

If everything goes successfully, you will get a console provided by ttyPS0 in the systemd and systemd services booting period. The console will prompt to login the OS by input username and password.

3. A quick glance in advance
Boot to bb1. As shown by the picture Oboot2bb1.JPG

开源项目 > 其他开源 > 操作系统

openEuler / community Go MulanPSL-2.0

代码 Issues 85 Pull Requests 18 Wiki 3 DevOps 服务

指数 86 Watch 66 Star 89 Fork 266

开启的 I1132 sig-RISC-V: NutShell(果壳, UCAS) CPU处理器

zhoupeng01:master → openEuler:master

zhoupeng01 创建于：5小时前 [openeuler-claes/stat/need-squash](#)

sig-RISC-V: NutShell(果壳UCAS) CPU: Support openEuler to run on NutShell UCAS COOSCA CPU.
Srcs, images and some descriptions.

NutShell UCAS COOSCA1.0 CPU is RISC-V architecture. The ISA is rv64imafdc, and mmu is sv39.

共8条评论, 3人参与

zhoupeng01 推送了代码 3小时前

展开详细操作日志

zhoupeng01/autobuild-openeuler4riscv
forked from openEuler/RISC-V
Tools scripts for auto-building openEuler SRPMs for RISC-V
最近更新: 1小时前

zhoupeng01/community Go
forked from openEuler/community
Community governance is listed in the repository.
最近更新: 3小时前

zhoupeng01/openEuler-riscv-glibc-NutShell
riscv-glibc, for building openEuler OS running on NutShell UCAS COOSCA CPU. The initial version is import from <https://github.com/isrc/riscv-glibc>.
最近更新: 7小时前

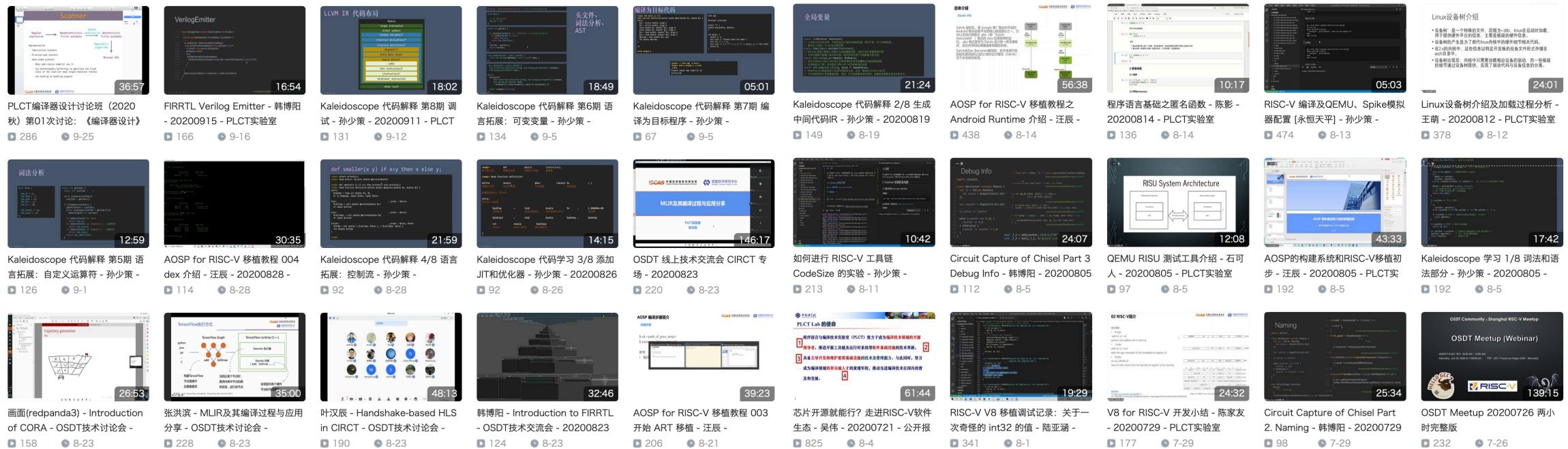
zhoupeng01/openEuler-systemd-NutShell
Support openEuler OS run on NutShell UCAS COOSCA CPU in Xilinx FPGA PYNQ-Z2 emulation environment. Based on openEuler systemd-243-18.oe1src.rpm (rpm2cpio systemd-243-18.oe1src.rpm|cpio -iv)
最近更新: 8小时前

zhoupeng01/openEuler-riscv-pk-NutShell
bb1, dtb, build system ant etc. to build kernel and bootloader to support openEuler OS running on NutShell UCAS COOSCA CPU. The initial version is import from <https://github.com/OSCPU/riscv-pk>.
最近更新: 1天前

zhoupeng01/openEuler-Kernel-NutShell C
Porting openEuler kernel to support NutShell UCAS COOSCA CPU. Initial version is from <https://gitee.com/openeuler/kernel/tree/kernel-4.19>.
最近更新: 1天前

观看PLCT/OSDT的技术分享、一起学习和进步

- <https://space.bilibili.com/296494084/video>
- 已更新技术分享140多个，每周三技术分享后上传
- 每周五、周六下午3点有编译技术入门与实战讨论班（线上），欢迎加入



每个月1号和16号，「PLCT开源进展」半月刊更新最新工作进展

- <https://www.zhihu.com/column/plct-lab>
- <https://github.com/isrc-cas/PLCT-Weekly>

欢迎关注专栏、点赞和评论

知乎 | 专栏
软件所PLCT实验室

软件所PLCT实验室

小乖他爹 · 34 篇内容 · ...

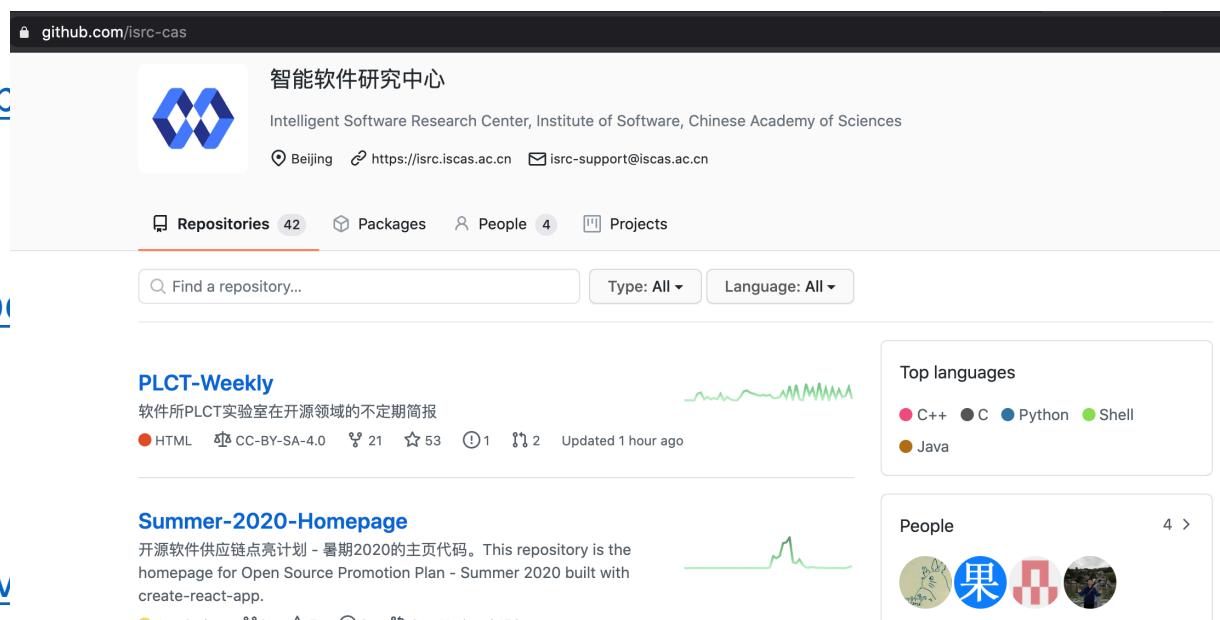
PLCT开源进展·第10期·2020年10月01日

卷首语 感谢最近刚刚进入RISC-V国际基金会董事会的包云岗老师，将我们引荐给了基金会的CTO并进行了几次技术交流。现在，RISC-V基金会终于知道软件所PLCT实验室的存在了。通过这几次会议，我才注意到所有的RISC-V基金会的技术会议实际上都是公开的。不仅有列表列出了所有的时间，甚至提供了一个 ICS 日历文件，让我... 阅读全文 ▾

▲ 赞同 26 ▾ 1条评论 分享 收藏

欢迎使用、参与开发、提问、报告bugs、提 Pull Requests

- PLCT Weekly <https://github.com/isrc-cas/PLCT-Weekly>
- C910 LLVM 支持 <https://github.com/isrc-cas/c910-llvm>
- RISC-V Vector Extension Support (rvv-llvm) <https://github.com/isrc-cas/rvv-llvm>
- V8 for RISC-V <https://github.com/v8-riscv/v8>
- 永恒天平 <https://github.com/isrc-cas/eternal-balance/>
- PLCT 公开报告 <https://github.com/isrc-cas/PLCT-Open-Report>
- QuickJS for RISC-V <https://github.com/isrc-cas/quickjs-riscv>
- PLCT-QEMU <https://github.com/isrc-cas/plct-qemu>
- Flounder <https://github.com/isrc-cas/flounder>
- OpenCV for RISC-V <https://github.com/isrc-cas/opencv-riscv>
- pacific (方舟编译器玩具运行时) <https://github.com/isrc-cas/pacitic>
- rvv-benchmark <https://github.com/isrc-cas/rvv-benchmark>
- 工具箱 <https://github.com/isrc-cas/PLCT-toolbox>



感谢 & 提问

- 如何加入RISC-V国际基金会？**线上免费申请成为个人会员**
- 如何参与国内RISC-V开源技术社区的工作？**联系PLCT！**
- 如果参与PLCT实验室的工作？**来实习吧！**
- 有问题但是不好意思现场提问？**可以加微信问**😊

