



芯片开源就能行？

走进RISC-V开源指令集及其软件生态

报告人：吴伟 (PLCT Lab)

中国科学院 软件研究所 智能软件研究中心

wuwei2016@iscas.ac.cn

[Https://github.com/isrc-cas/](https://github.com/isrc-cas/)



自我介绍

- 作为 PLCT Lab 的成员

1

➤ 在 ISRC 主导建立了 PLCT Lab , 目前 $15 + 3 + 12 = 30$ 人规模

- 作为 OSDT 社区的负责人

➤ 2007 年 HelloGCC 成立

2

- 作为开发者

➤ 编译器、虚拟机技术、开源软件、网络课老师

本次报告仅代表个人观点，带有软件出身技术人员的 bias

3



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**

1

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，

2

3

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。

4



PLCT Lab 的战略及行动原则

在编译工具等基础核心软件领域，

- (全球) 从零到壹
- (大陆) 从无到有
- (领域) top1

PLCT所有自研项目需满足以上至少一个标准。

1

定位类似特种突击小队，应对难点问题；处理不了规模化工程。

2



今天讨论的内容

- RISC-V 的背景、现状、未来规划：RISC-V基金会视角
- 为什么是 RISC-V、为什么是现在：报告人的个人视角
- 如何在战略上促进RISC-V发展并在关键技术点立足：PLCT视角
- PLCT Lab RISC-V开放项目介绍



RISC-V 的背景、现状、未来规划：RISC-V基金会视角

- 本节内容节选自 Krste Asanovic 教授在 RISC-V Summit 2019 上题为《State of the Union》的主题报告、以及 RISC-V 基金会CEO Calista Redmond 的开场报告。1
- 根据本次讨论的需要，进行了部分删减。图源地址：
 - <https://riscv.org/2019/12/risc-v-summit-2019-proceedings/>
 - <https://content.riscv.org/wp-content/uploads/2019/12/12.10-9.20-StateOfUnion.pdf>
 - <https://content.riscv.org/wp-content/uploads/2020/01/Calista-Summit-2019-RISC-V-Revolution-12-09-2019.pptx>2



State of the Union

Krste Asanovic

UC Berkeley, RISC-V Foundation, & SiFive Inc.

1

krste@berkeley.edu

RISC-V Summit
San Jose Convention Center, CA, USA
December 10, 2019

2



RISC-V Origin Story

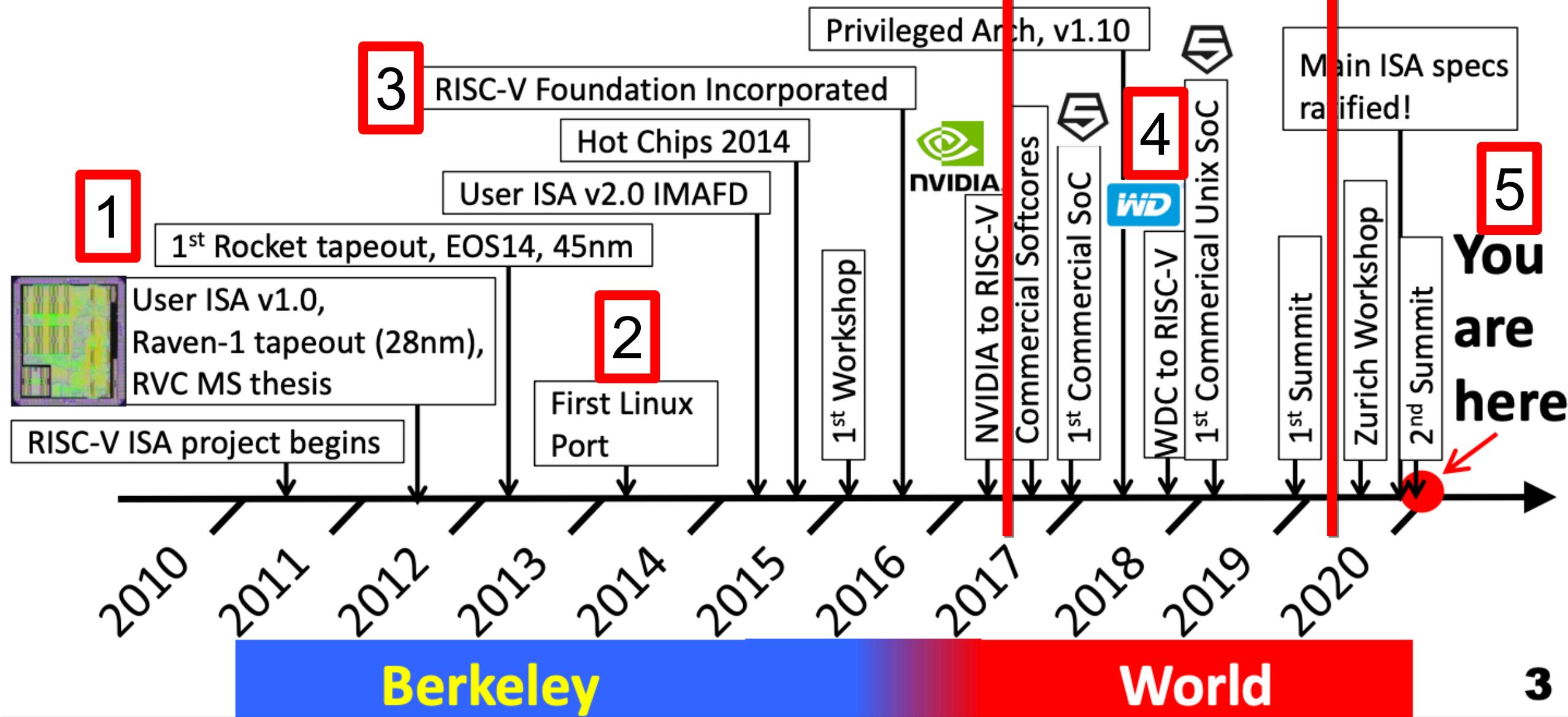
For Berkeley Par Lab project 2010, we needed an ISA:

- 1■ simple to implement
- efficient
- extensible
- 2■ no constraints on sharing with our work with others

No existing ISA met needs, so created RISC-V (*Andrew Waterman, Yunsup Lee, David Patterson, Krste Asanovic*)

- 3■ Turns out, many others wanted same things
- From initial roll out in 2014, rapid uptake everywhere 4

RISC-V Timeline



What's Different about RISC-V?

- **Simple**

1

- Far smaller than other commercial ISAs

- **Clean-slate design**

- Clear separation between user and privileged ISA
 - Avoids μarchitecture or technology-dependent features

- A **modular** ISA designed for **extensibility/specialization**

2

- Small standard base ISA, with multiple standard extensions
 - Sparse and variable-length instruction encoding for vast opcode space

- **Stable**

3

- Base and standard extensions are frozen
 - Additions via optional extensions, not new versions

- **Community designed**

4

- With leading industry/academic experts and software developers



RISC-V Ecosystem

Software

2

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

1

Hardware

Open-source cores:

Rocket, BOOM, RI5CY,

3

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

5



Why is RISC-V so popular?

- Engineers sometimes “*don’t see forest for the trees*”
- The movement is **not** happening because some benchmark ran 10% faster, or some implementation was 30% lower power
- The movement **is** happening because ***new business model*** 1
changes everything
 - Pick ISA first, then pick vendor or build own core
 - Add your own extension without getting permission 2
- Implementation features/PPA will follow
 - Whatever is broken/missing in RISC-V will get fixed

70,000

60,000

50,000

40,000

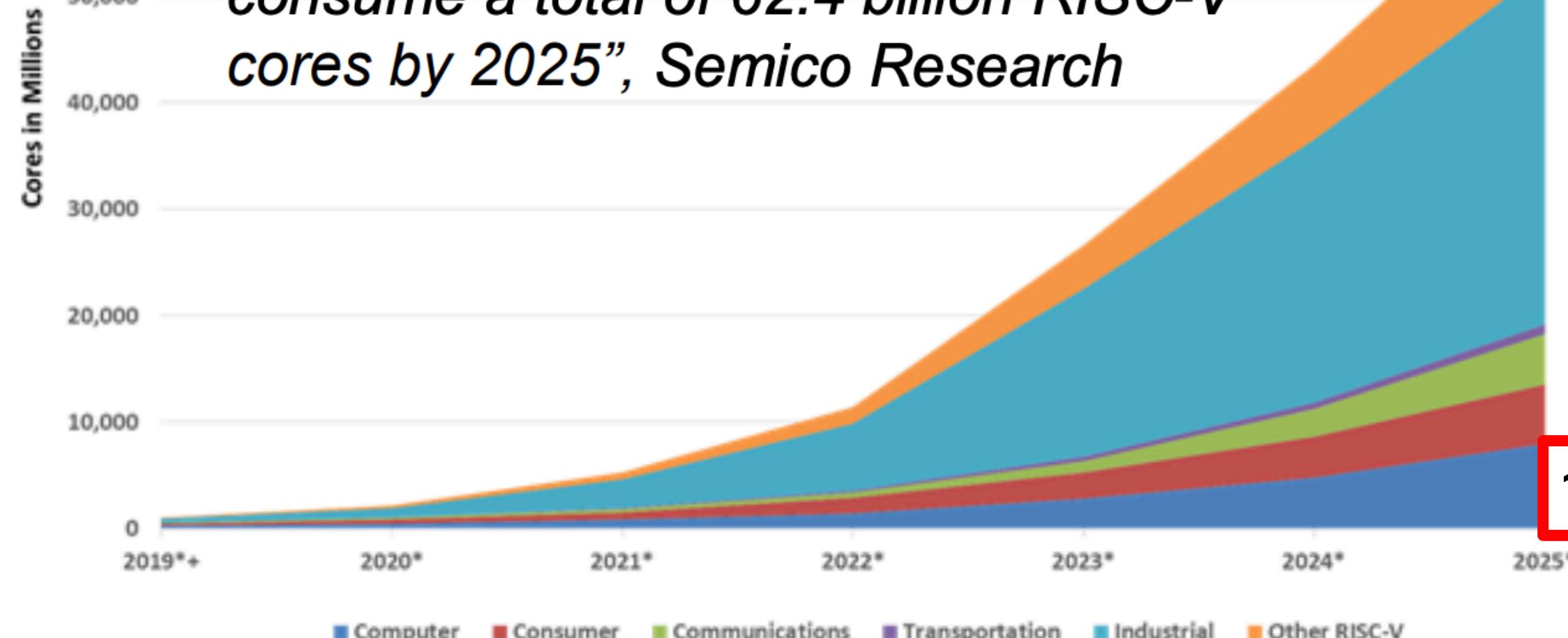
30,000

20,000

10,000

0

“Based on the already sizeable adoption of RISC-V, we forecast that the market will consume a total of 62.4 billion RISC-V cores by 2025”, Semico Research



1



Our Modest RISC-V Project Goal

1 *Become the industry-standard ISA for all computing devices*

2 *This is happening!*

*Far faster, more domains, than anyone predicted
Demand at every performance level (low to ludicrous)*

Demand for all features of all other ISAs

3 *Plenty of work to do, hardware and software!*



2019: The ISA Foundation is in Place!

- RV32IMAFDQC , RV64IMAFDQC ratified 1
 - +Zifencei, +Zicsr
- Privileged Arch 1.11 ratified
- Base memory model RVWMO ratified
- Debug spec 0.13 ratified
- SAIL formal spec adopted as golden model
- Compliance Suite v0.1 imminent

- Substantial upstream software support 2



How we structure RISC-V standards

1 ▪ ISA specs

- Modular ISA, each module eventually ratified and frozen
- Each module can contain options
- Modules designed to combine in system without conflicts

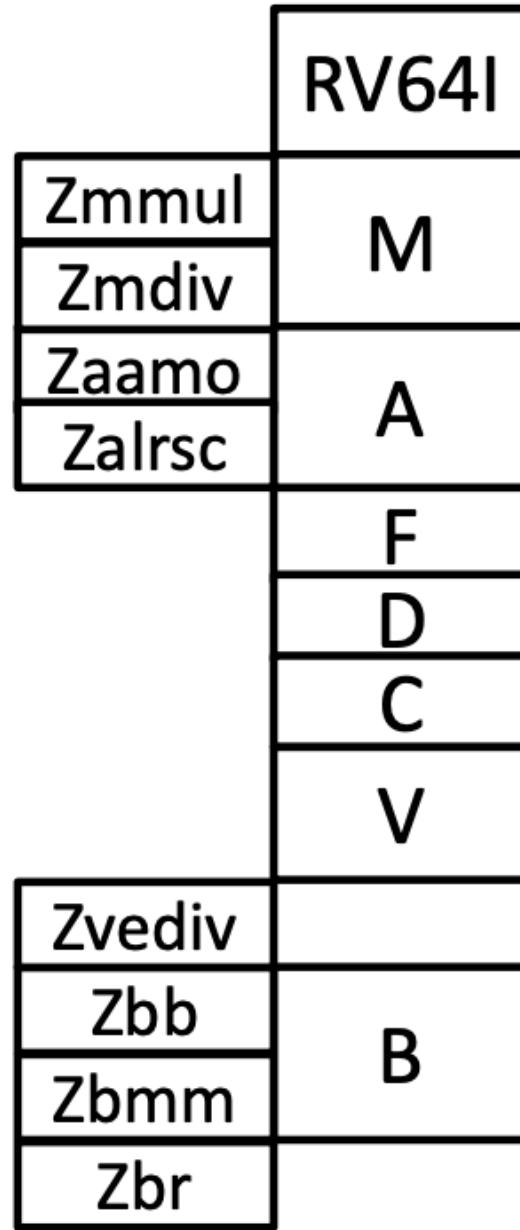
2 ▪ Platform specs

- Custom ISA space separated from standard ISA space
- *ISA specs do not constrain how the modules are used*

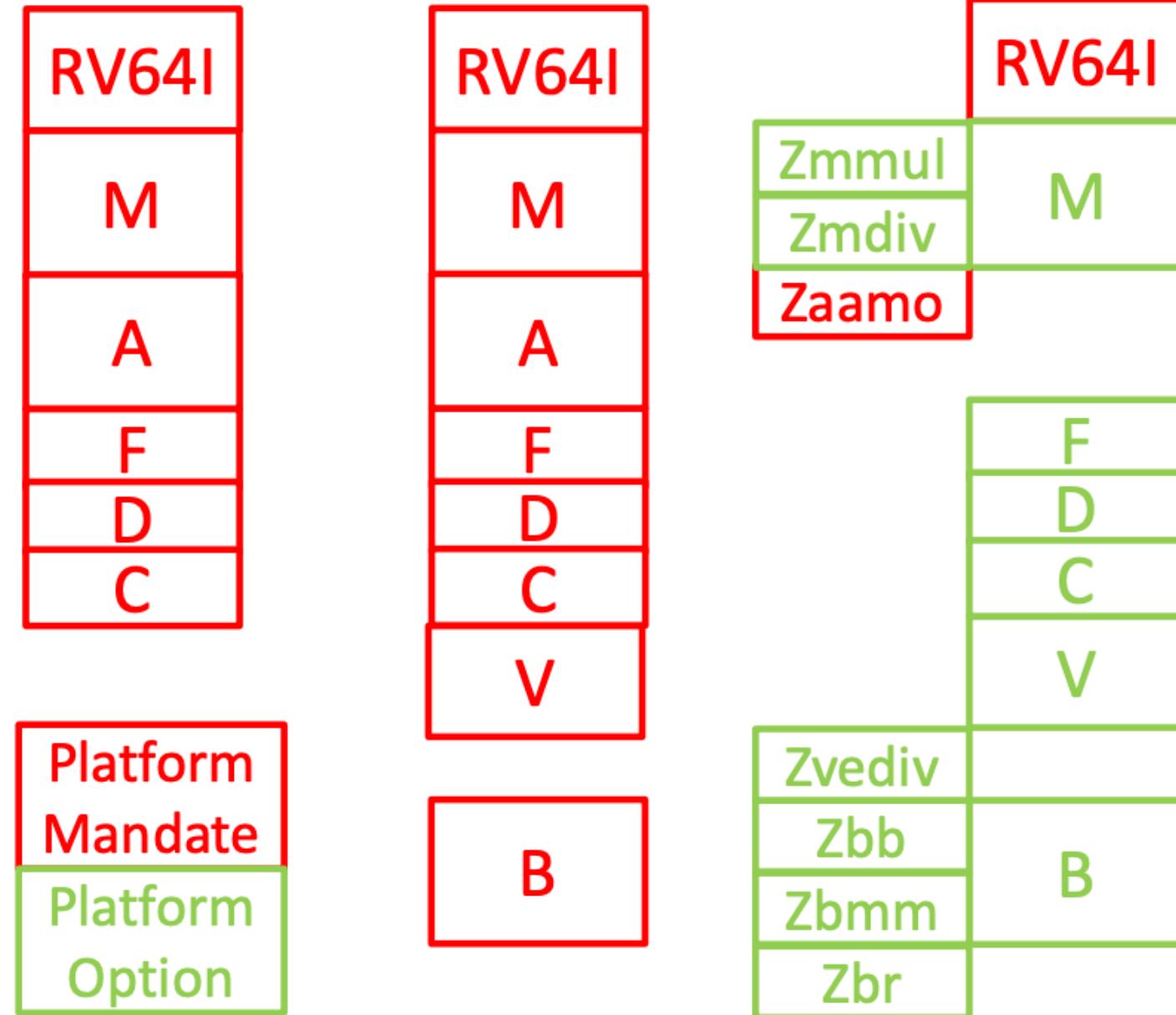
3 ▪ Platform specs

- Driven by software ecosystem (e.g., FreeRTOS, server)
- Provides constraints on choice of modules and options so software is straightforward to port

ISA modules



Linux 2017 Linux 2020



FreeRTOS



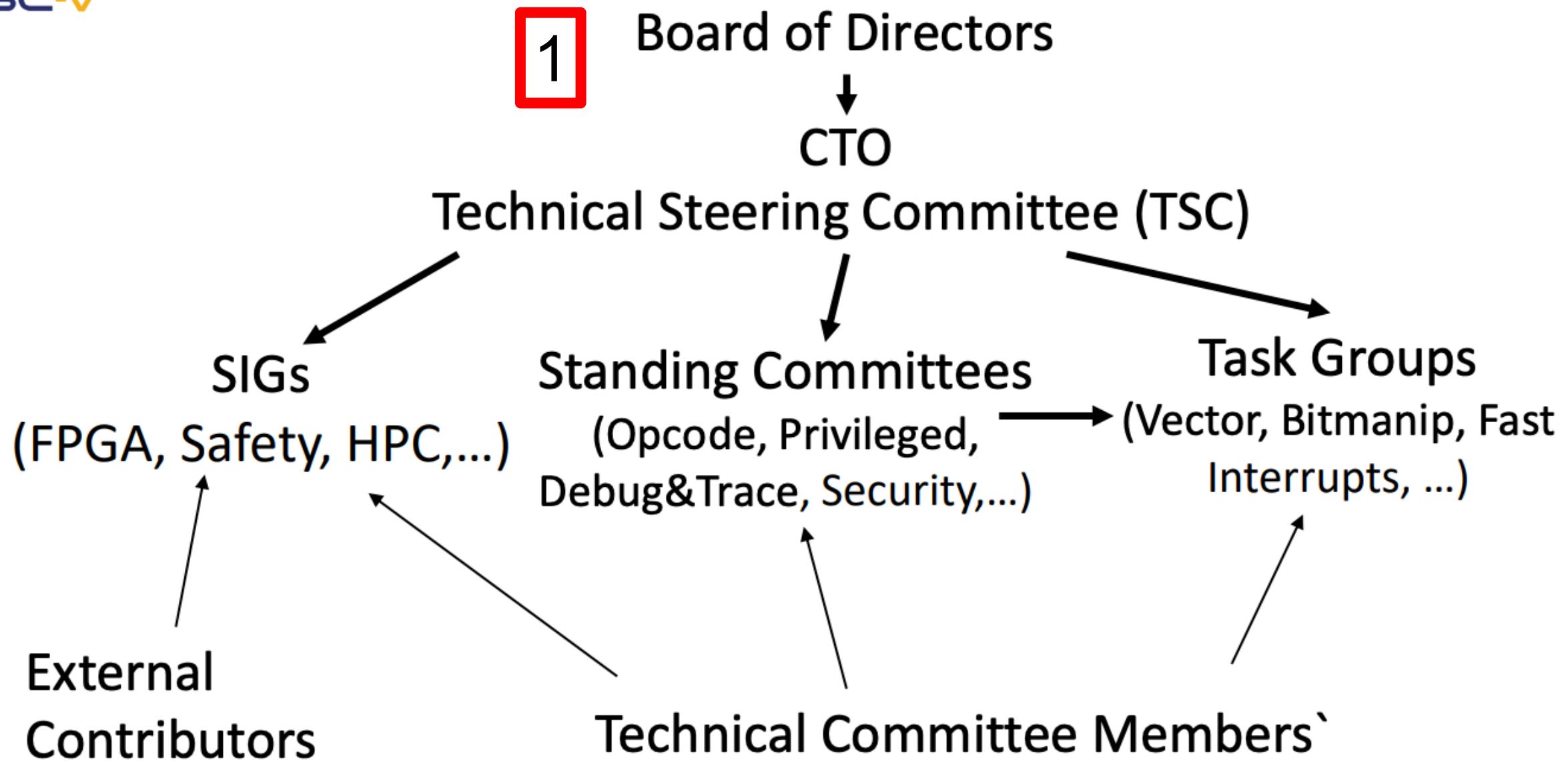
How is RISC-V Avoiding Fragmentation?

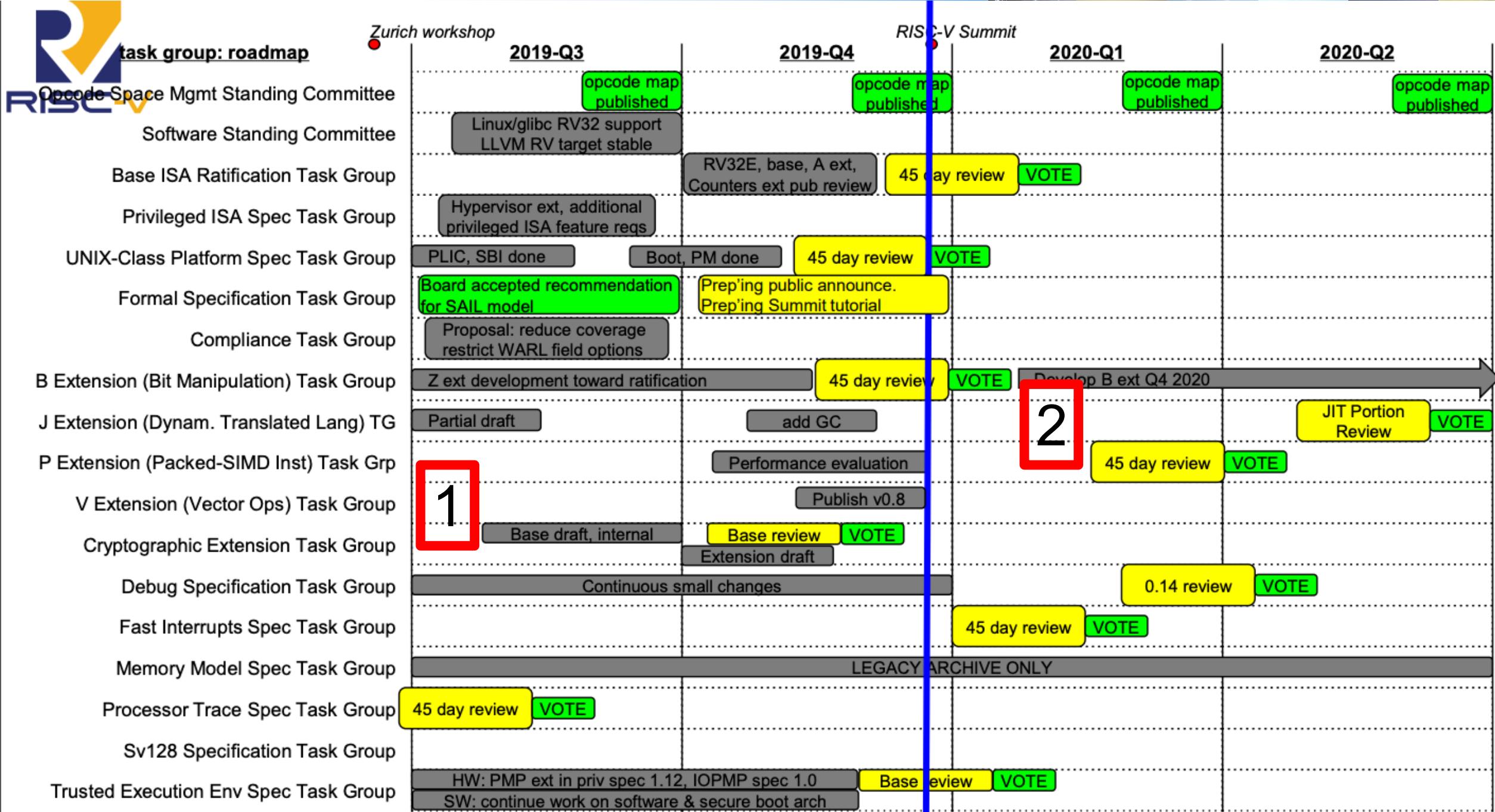
Two powerful forces keep fragmentation at bay: 1

- **Users:** No one wants a repeat of vendor lock-in.
- **Software:** No one, not even nation state, can afford 2 their own software stack. Upstream open-source projects only accept frozen/ratified Foundation standards.



Scaling the RISC-V Standards Org





1 “V” Vector Extension

- Provides scalable vector instructions
 - From 32x32b vector registers, to thousands bits per vector
 - From embedded fixed-point to HPC floating-point
- Release of version 0.8 of spec at 2nd Summit
- Binutils, spike, Imperas model and compliance tests available
- Multiple implementations in flight
- Anticipate freeze in mid-2020, ratify by end of 2020 2

“J” Extension

- J TG focusing on support for dynamically translated languages 1
- Main topic is new instruction coherence scheme
 - Important for JIT compilers
 - FENCE.I was too heavyweight for some systems



Fast Interrupts "CLIC" Task Group

- Provides low-latency, preemptive nested interrupts local to each core
- Most functionality defined in software, supports multiple ABIs
- Spec at version 0.9, refining details of what is supported in base profile
- To be ratified in early 2020 1 2

RISC-V Predictions for 2020

- In 2020, see significant investment in high-end RISC-V 1
implementations (server, mobile, auto)
- We're laying the ISA and software foundations to enable these
 - Hypervisor
 - Vectors
 - 2 - Security
 - Power-management
 - Server platform profile
 - 3 - Functional Safety

Accelerating the software ecosystem

- We have an aggressive roadmap for 2020
- Current RISC-V software/ecosystem gaps:
 - RV32 support for Linux
 - Upstream EDK2
 - Upstream Linux distribution support
 - Rust Linux/RV64 support
 - Support for new RISC-V extensions (Hypervisors, BitManip, Vector)
 - Toolchain code size optimizations
 - Javascript JITs
 - Browsers
 - OpenJDK
 - Low cost Linux based development boards

Accelerating the software ecosystem

- We have an aggressive roadmap for 2020
- Current RISC-V software/ecosystem gaps:
 - RV32 support for Linux
 - Upstream EDK2
 - Upstream Linux distribution support
 - Rust Linux/RV64 support
 - Support for new RISC-V extensions (Hypervisors, BitManip, Vector)
 - ◦ Toolchain code size optimizations
 - ◦ Javascript JITs
 - ◦ Browsers
 - OpenJDK
 - ◦ Low cost Linux based development boards

We need your help!

- RISC-V BoD wants to invest more in software enablement as well as prevent potential fragmentation in the community
- 1
- We will reach out to you shortly
 - What do you need from the RISC-V software ecosystem?
 - What RISC-V software are you working on internally?
 - Are there ways where you can contribute to the RISC-V ecosystem?
- Together, we can help build out the missing pieces in the software ecosystem and make RISC-V the default ISA for computing.
- 2



欢迎就第一小节的内容提问

- ~~RISC-V的背景、现状、未来规划：RISC-V基金会视角~~
- 为什么是 RISC-V、为什么是现在：报告人的个人视角
- 如何在战略上促进RISC-V发展并在关键技术点立足：PLCT视角
- PLCT Lab RISC-V开放项目介绍



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，以及 x86 的开源core都是有的，也默默流片不少
- 例如阿里巴巴收购的中天微，就维护着自己的指令集
 - 仅仅国内就有好几家（曾今和现在）有自己维护的指令集
- RISC-V 是众多新ISA脱颖而出的一个，**大家都买账**



为什么是现在？

- 并不是现在，RISC-V的起步是2010年
 - 比2010更早的2003年，摩尔定律就被宣布死亡
 - 要加速和加功能？只能自己上了
 - 但是即使不要钱，Arm和x86的decoder已经不是一个人能搞定的了
 - Andrew “bunnie” Huang代表的硬件黑客的促进
- (然而，摩尔定律不死，变成了每隔18个月被重新定义一次 😊)



为什么是 RISC-V（而不是大陆的企业）？

- 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——包云岗老师说的）
 - <https://blog.csdn.net/moxibingdao/article/details/106667615>
- 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



为什么左手成立基金会，右手拉起来 Sifive？

- 没有任何一个公司甚至国家可以完整建立软件栈
- 如果没有一家公司带头烧钱，进展会很慢
- ~~多少赚点儿~~



RISC-V 相比国内做的差距最大的地方在哪里

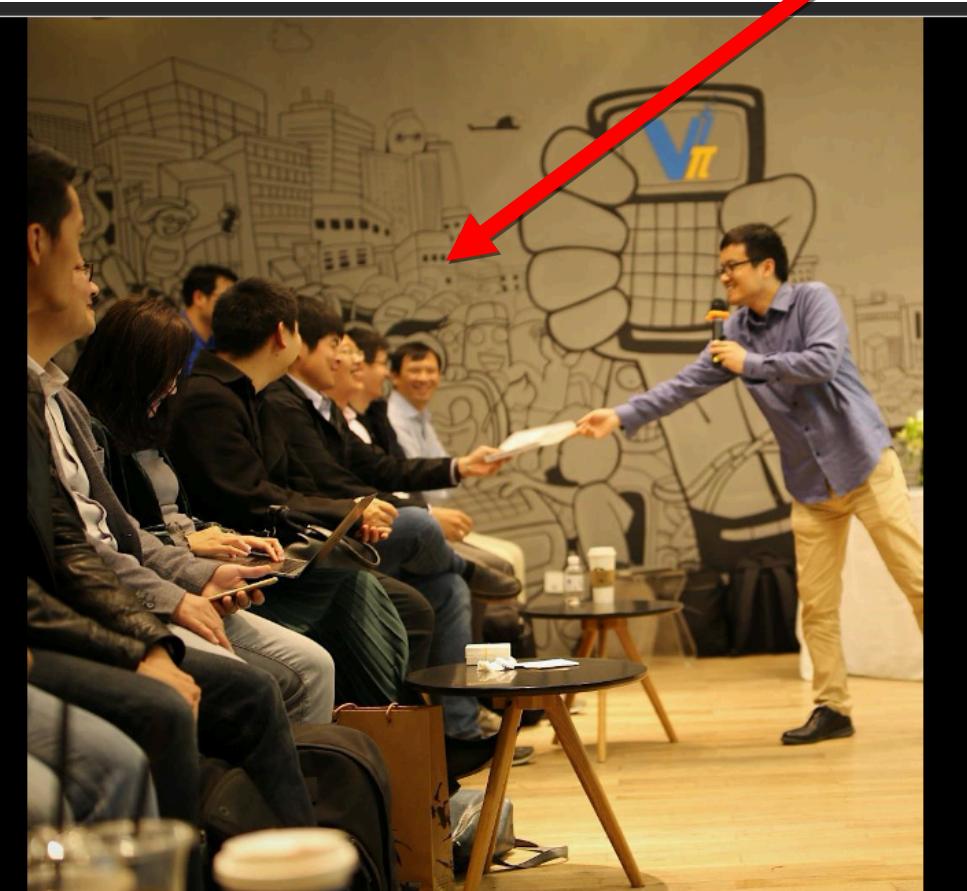
- 软件上游！Upstream！
- 在基础软件领域，PLCT Lab 可能已经超过了大多数国内团队 1 2



国内现在 RISC-V 火么？

- 非常火、但是.....

1



2

CHINA

MORE THAN 200 MEMBERS HAVE JOINED THE CRVA AND CRVIC ASSOCIATIONS AS WELL AS 33 MEMBERS IN THE GLOBAL RISC-V FOUNDATION. ACROSS 2019 WE'VE LED DISCUSSIONS IN 6 CITIES AS WELL AS HOSTED A DAY OF TALKS AT THE WORLD INTERNET CONFERENCE. IN NOVEMBER, WE HOSTED 500 ATTENDEES AT THE CHINA RISC-V FORUM.





有关于 RISC-V 的八卦，可以在本节提问 ;-)

- ~~RISC-V 的背景、现状、未来规划：RISC-V基金会视角~~
- ~~为什么是 RISC-V、为什么是现在：报告人的个人视角~~
- 如何在战略上促进RISC-V发展并在关键技术点立足：PLCT视角
- PLCT Lab RISC-V开放项目介绍



为什么 PLCT Lab 会 All in RISC-V

- RISC-V 战略方向是软件所关注的方向之一
- 基础软件领域已经很久没有新的轮子需要重新发明了
 - 编译器 x N
 - 语言虚拟机 x N
 - 系统虚拟机 x N
 - 全系统性能分析
 - 二进制工具
- PLCT Lab 相信 RISC-V 代表了未来的IT的发展方向
 - 在正确的时间、正确的方式解决了重要的问题



如何促进和立足：PLCT Lab 视角

- 为国内芯片初创企业提供婴儿阶段的看护
 - 国内有能力和经验的基础软件类技术人员非常少
 - 初创芯片公司往往硬件团队创业，没有预估到适配软件的巨大开支
 - ~~(基本)不赚这些企业的钱~~
- All Upstream (接下来会介绍)
 - RISC-V V Ext LLVM Impl
 - V8 for RISC-V
 - QEMU
 - Spike
 - OpenCV



国内的企业有哪些机会

- MCU领域已经形成了成功的模式
- 通用计算和IoT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
- 各类AI加速器初创公司还有机会
- 目前的差距主要体现在软件能力，而不是硬件研发能力上



结合 PLCT 使命的 RISC-V 战略布局

- PLCT Lab 负责看护编译器、模拟器、虚拟机三大领域
- 操作系统和软件供应链由ISRC操作系统团队保证
- 通过代码分析员计划（Code Analyst Project）填补知识黑洞



软件所ISRC其他组的 RISC-V 战略布局

● 操作系统组

- **开源软件供应链安全的评估、度量、保障**
- **面向 RISC-V 的操作系统关键技术研究**



所有有关 PLCT Lab 的疑问，可以在本节提问 ;-)

- ~~RISC-V的背景、现状、未来规划：RISC-V基金会视角~~
- ~~为什么是RISC-V、为什么是现在：报告人的个人视角~~
- ~~如何在战略上促进RISC-V发展并在关键技术点立足：PLCT视角~~
- PLCT Lab RISC-V开放项目介绍



PLCT 重点项目：RVV-LLVM 已开源、已跟上游合作

短期目标：实现 LLVM 版本的 RISC-V 向量扩展指令集支持，并长期维护适用于国内 RISC-V 芯片厂商的编译器定制版本。

长期目标：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。

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

共有资深专家1名、研究生1名、实习生及预备员工4名。



PLCT 重点项目：RVV-LLVM

已经达到的成果：

- 完成 v0.7.1、v0.8 草案的参考实现，正在实现 v0.9 草案增补功能
- PLCT版本的实现已经成为至少一家国内芯片初创公司的基础版本
- 与平头哥、芯来科技等国内RISC-V厂商在开源领域建立合作关系
- 已经进入RISC-V基金会相关讨论组
- 进行了一系列技术报告分享及开放日推广活动、吸引更多学生参与

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

共有资深专家1名、研究生1名、实习生及预备员工4名。



PLCT 重点项目：V8 for RISC-V已开源、已跟上游合作

短期意义：Google V8 是目前全球占有率第一大 JavaScript 引擎，同时是 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。

长期意义：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。



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

目前小队共有7人，3名资深员工，1名新人员工，3名实习生



PLCT 重点项目：V8 for RISC-V

已经达到的成果：

- 在RISC-V平台上正确编译和启动。能够正确运行 helloworld.js
- 已经成为第一个开源的 V8 for RISC-V 并与 RIOS Lab x FutureWei 合作

- 持续分享PLCT内部的V8技术报告PDF及视频（是目前除V8官方提供的少量技术分享之外唯一可以公开获得的V8核心代码分析资料）





PLCT 重点项目：面向国内RISC-V厂商的QEMU支持计划

短期目标：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、准备跟上游合作

长期目标：

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。

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



PLCT 重点项目：面向国内RISC-V厂商的QEMU支持计划

已经达到的成果：

- 具备QEMU扩展开发能力并已经完成某商业交付项目（横向）
- 完成对芯来科技RISC-V开发板的QEMU虚拟化支持，已经开源
- 与 RT-Thread 建立了长期紧密的社区合作，即将开源
- 借助OSDT社区进行了多次QEMU主题的技术分享



PLCT 重点项目：方舟编译器支持计划

短期目标：

促进方舟开源编译器社区的发展。

为方舟开源编译器添加 RISC-V64 后端及运行时环境

长期目标：

作为编译器开发的培训项目进行新人培养和人才输送（不仅输送给方舟）。



PLCT 重点项目：方舟编译器支持计划

已经达到的成果：

- 完成60+方舟源代码分析专栏连载（唯一技术连载，填补官方空缺）
- 完成目前唯一的方舟源代码分析的技术书籍（已交由出版社准备印刷）
- 方舟开源社区（除华为员工外）最活跃的第三方贡献者



PLCT 培育项目：永恒天平

短期目标：Eternal Balance（永恒天平）项目是PLCT内部为了跟踪和对标RISC-V工具链而建立的测评体系，偏重编译系统的性能（速度）、安全性、代码体积三个方面。

长期目标：建立统一标准，成为Mozilla在JS引擎的AWFY那样的中立「调停者」，实时跟踪所有开源项目的执行速度、代码体积、安全性评估。

部分结果开源：<https://github.com/isrc-cas/eternal-balance> （计划2020Q4完成开源）



PLCT 培育项目：OpenROAD

短期目标：跟进掌握 OpenROAD 核心技术、在国内推广普及。

战略层面：以开源 EDA 作为切入点，在未来5-7年内扶持国内EDA企业完成编译技术等方面的关键技术攻坚。

部分结果开源： TBD (计划2020Q4完成开源)



PLCT 培育项目：MLIR

短期目标：跟进MLIR社区进展，成为MLIR社区贡献者。

战略层面：以MLIR作为切入点，保持对AI技术栈的跟进，保持快速进场能力。
。

开源：合并进入MLIR官方仓库（计划2020年9月份完成合并）



PLCT 培育项目：OpenCV for RISC-V

短期目标：融入OpenCV社区，成为OpenCV社区在RV方面的贡献者。

长期目标：与OpenCV的国内研发团队形成战略合作，相互扶持。

开源：合并进入OpenCV官方仓库（计划2020年9月份完成合并）



非常感谢大家参与交流讨论！可能的合作点？

■ 学术研究领域

- ❖ 编译技术、编译器测试技术
- ❖ 虚拟机技术
- ❖ 软硬件融合的优化或性能防护技术

■ 工程领域

- ❖ RISC-V 软件生态领域涉及编译器、虚拟机、模拟器的均在PLCT看护范围

欢迎关注PLCT实验室的GitHub <https://github.com/isrc-cas/PLCT-Weekly>