

循序渐进，学习开发一个 RISC-V 上的操作系统



第 2 章 RISC-V ISA 介绍

汪辰

➤ ISA 的基本介绍

➤ RISC-V ISA 基本介绍

- 【参考 1】 : The RISC-V Instruction Set Manual, Volume I: Unprivileged ISA, Document Version 20191213
- 【参考 2】 : The RISC-V Instruction Set Manual, Volume II: Privileged Architecture, Document Version 20190608-Priv-MSU-Ratified
- 【参考 3】 : RISC-V手册 (中文版): <http://riscvbook.com/chinese/>

➤ ISA 的基本介绍

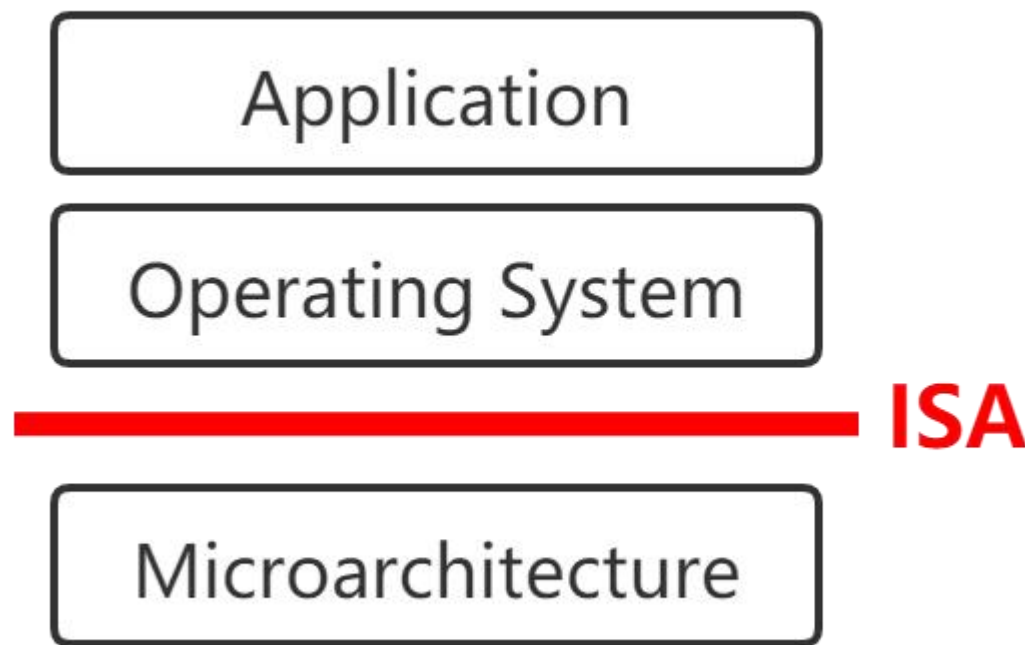
- ISA 是什么
- 为什么要 ISA
- CISC vs RISC
- ISA 的宽度
- 知名 ISA 介绍

➤ RISC-V ISA 基本介绍

➤ **ISA (Instruction Set Architecture) 指令集架构**：是底层硬件电路面向上层软件程序提供的一层**接口规范**。

➤ **ISA 定义了：**

- 基本数据类型：
BYTE/HALFWORD/WORD/.....
- 寄存器 (Register)
- 指令
- 寻址模式
- 异常或者中断的处理方式
- 等等



- 为上层软件提供一层抽象，制定规则和约束，让编程者不用操心具体的电路结构。
- IBM 360 是第一个将 ISA 与其实实现分离的计算机。



图片来自网络

➤ CISC 复杂指令集

(Complex Instruction Set Computing)

- 针对特定的功能实现特定的指令，导致指令数目比较多，但生成的程序长度相对较短。

➤ RISC 精简指令集

(Reduced Instruction Set Computing)

- 只定义常用指令，对复杂的功能采用常用指令组合实现，这导致指令数目比较精简，但生成的程序长度相对较长。

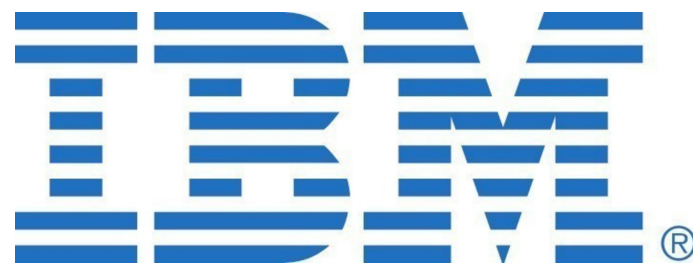
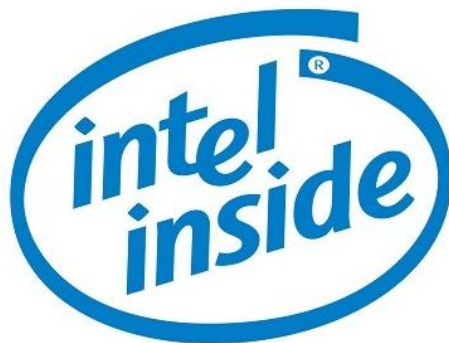
➤ 现如今，RISC 和 CISC 也逐渐有相互融合的趋势。

- **ISA（处理器）的宽度指的是 CPU 中 *通用寄存器* 的宽度（二进制的位数），这决定了寻址范围的大小、以及数据运算的能力。**

通用寄存器的宽度	寻址范围	应用场景
8 位	$2^8 = 256$	早期的单片机 8051
16 位	$2^{16} = 65536$	X86系列的鼻祖 8086，MSP430系列单片机
32 位	$2^{32} = 4294967296$	早期的终端，个人计算机和服务端
64 位	$2^{64} = 18446744073709551616$	目前主流的移动智能终端，个人计算机和服务端

- **注意一个问题：ISA 的宽度和指令编码长度无关。**

- X86
- SPARC
- Power
- ARM
- MIPS
- RISC-V
-



➤ ISA 的基本介绍

➤ RISC-V ISA 基本介绍

- RISC-V 的历史简介
- RISC-V 究竟是什么
- RISC-V 发展现状
- RISC-V 的特点
- RISC-V ISA 规范一览

➤ 对比软件业界的自由与开放

领域	开放的标准	开源的实现	闭源的实现
操作系统	POSIX	Linux, FreeBSD,	Windows,
编译器	C	Gcc, LLVM,	Intel icc, ARMcc,
数据库	SQL	MySQL,	Oracle, DB2,
ISA	???	X86, ARM,

➤ ISA 的未来在哪里?

- RISC-V 念作 “risk-five”，代表着 Berkeley 所研发的第五代精简指令集。
- 该项目 2010 年始于加州大学伯克利（Berkeley）分校，希望选择一款 ISA 用于科研和教学。经过前期多年的研究和选型，最终决定放弃使用现成的 X86 和 ARM 等 ISA，而是自己从头研发一款：
 - X86：太复杂，IP 问题
 - ARM：一样的复杂，而且在 2010 年之前还不支持 64 位，以及同样的 IP 问题。
- 主要研发人员：
 - Andrew Waterman, Yunsup Lee, David Patterson, Krste Asanovic



RISC-V 究竟是什么

- 一款高质量，免许可证，开放的 RISC ISA
- 一套由非营利的 RISC-V 基金会维护的标准: <https://riscv.org/>
- 适用于所有类型的计算系统：从微控制器到超级计算机
- RISC-V 不是一家公司，也不是一款 CPU 实现。

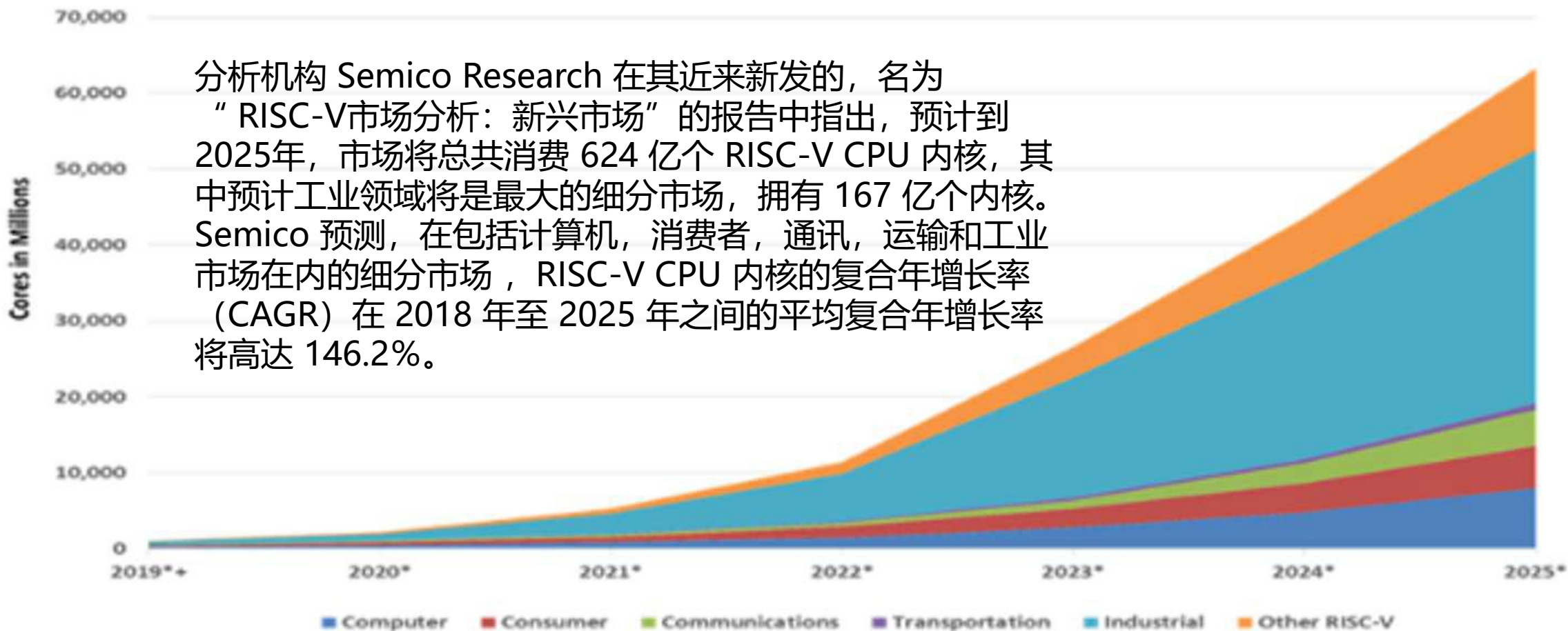


About RISC-V

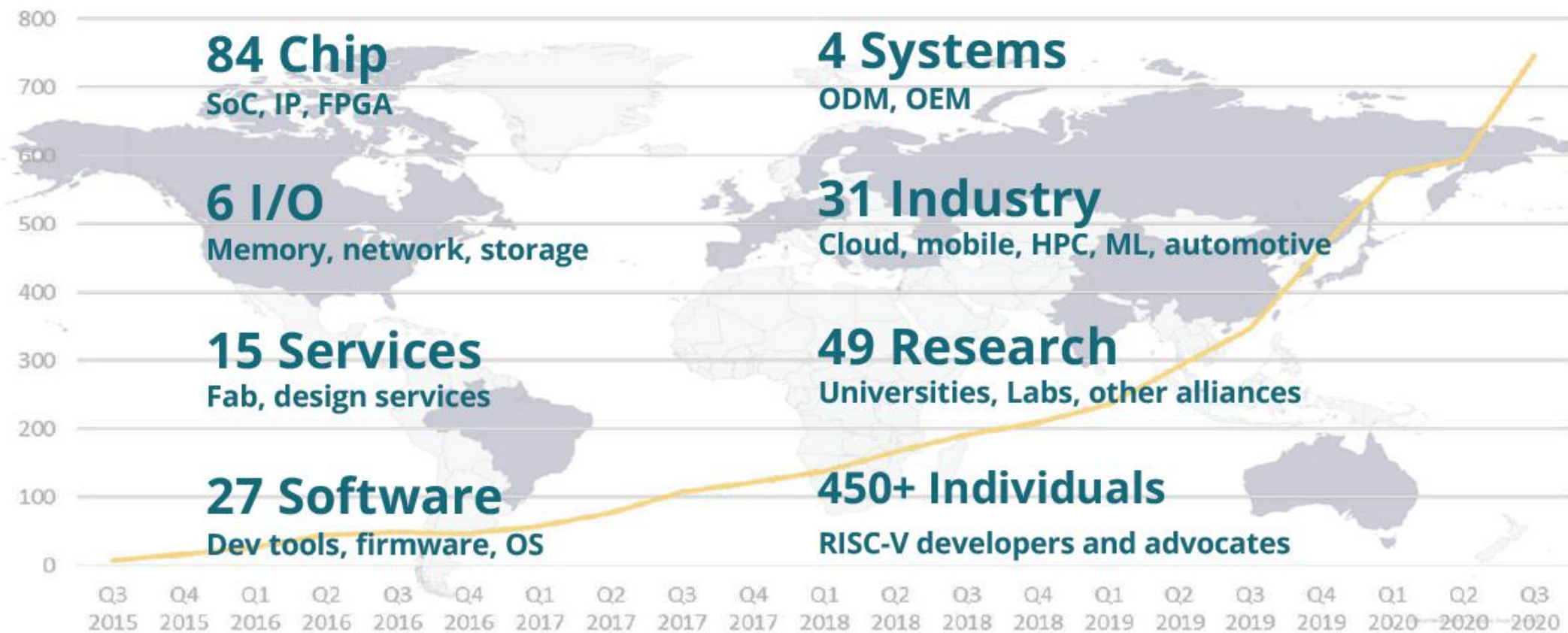
RISC-V is a free and open ISA enabling a new era of processor innovation through open standard collaboration.

The RISC-V ISA delivers a new level of free, extensible software and hardware freedom on architecture, paving the way for the next 50 years of computing design and innovation.

RISC-V 发展现状 (1)



RISC-V 成员已超过 700 个，分布在全球 50 个国家
<https://riscv.org/members/>



- 简单
- 清晰的分层设计
- 模块化
- 稳定
- 社区化

➤ 官方 ISA 标准下载地址(截至2021/3): <https://riscv.org/technical/specifications/>

ISA Specification

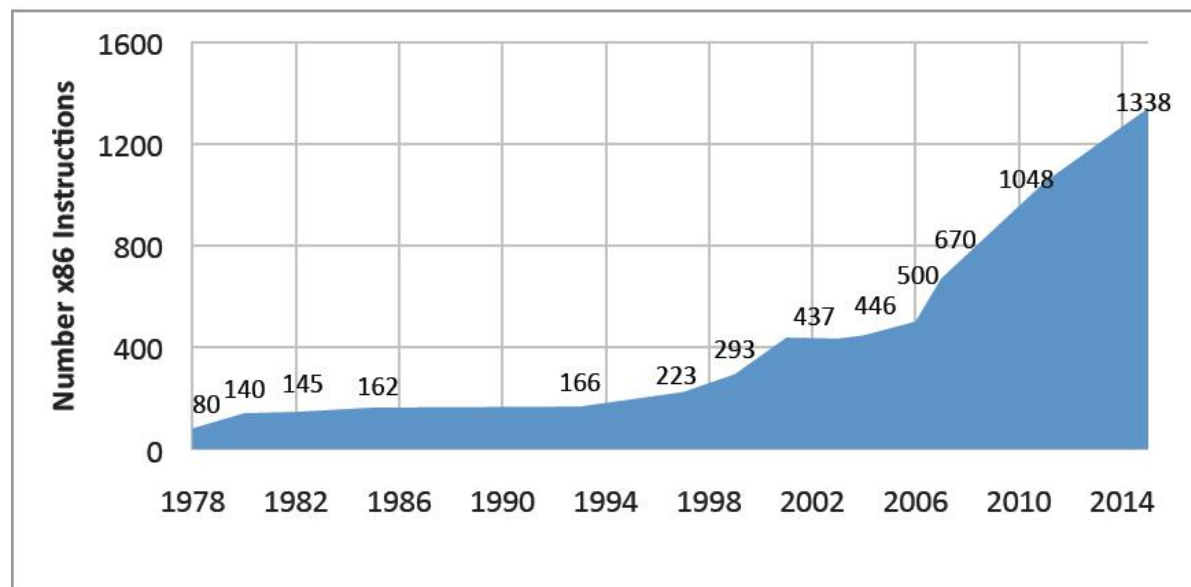
The specifications shown below represent the current, ratified releases. Work is being done on [GitHub](#).

- Volume 1, Unprivileged Spec v. 20191213 [[PDF](#)]
- Volume 2, Privileged Spec v. 20190608 [[PDF](#)]

Past ratified releases include the term “ratified” in the [release tag](#).

- **ISA 命名格式：RV[###][abc....xyz]**
 - RV：用于标识 RISC-V 体系架构的前缀，即 RISC-V 的缩写。
 - [###]：{32, 64, 128} 用于标识处理器的字宽，也就是处理器的寄存器的宽度（单位为 bit）。
 - [abc...xyz]：标识该处理器支持的指令集模块集合。
- **例子：RV32IMA, RV64GC**

- **增量 ISA:** 计算机体系结构的传统方法，同一个体系架构下的新一代处理器不仅实现了新的 ISA 扩展，还必须实现过去的所有扩展，目的是为了保持向后的二进制兼容性。典型的，以 80x86 为代表



【参考 3】图 1.2

- **模块化 ISA:** 由 1 个基本整数指令集 + 多个可选的扩展指令集组成。基础指令集是固定的，永远不会改变。

RISC ISA = 1 个基本整数指令集 + 多个可选的扩展指令集

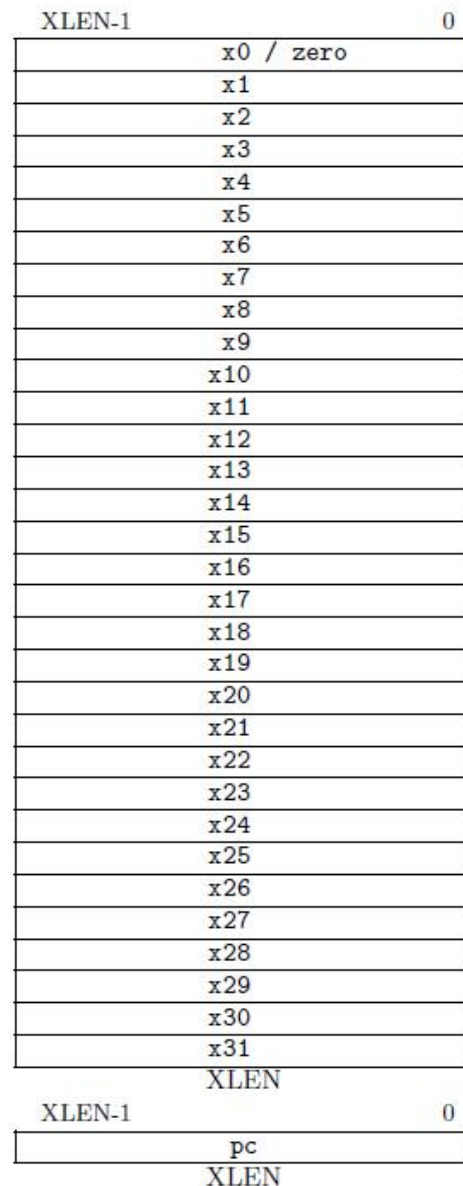
- **基本整数 (Integer) 指令集**
唯一强制要求实现的基础指令集，其他指令集都是可选的扩展模块。
- **扩展模块指令集：**
 - RISC-V 允许在实现中以可选的形式实现其他标准化和非标准化的指令集扩展。
 - 特定组合“IMAFD”被称为“通用 (General) ”组合，用英文字母 G 表示。
- **例子：**
 - RV32I: 最基本的 RISC-V 实现
 - RV32IMAC: 32 位实现，支持 Integer + Multiply + Atomic + Compressed
 - RV64GC: 64 位实现，支持 IMAFDC

基本指令集	描述
RV32I	32 位整数指令集
RV32E	RV32I 的子集，用于小型的嵌入式场景
RV64I	64 位整数指令集，兼容 RV32I
RV128I	128 位整数指令集，兼容 RV64I 和 RV32I

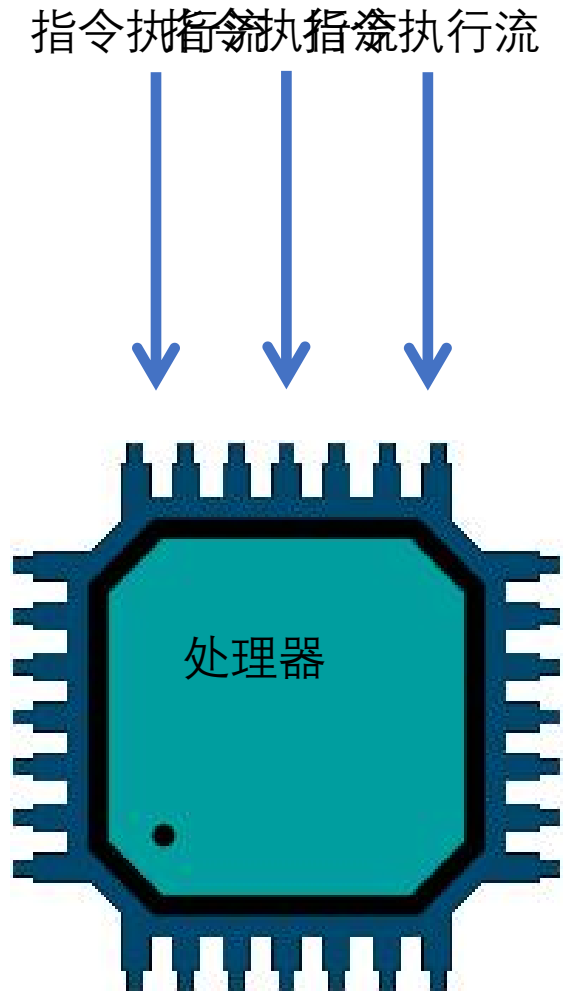
扩展指令集	描述
M	整数乘法 (Multiplication) 与除法指令集
A	存储器原子 (Atomic) 指令集
F	单精度 (32bit) 浮点 (Float) 指令集
D	双精度 (64bit) 浮点 (Double) 指令，兼容F
C	压缩 (Compressed) 指令集
.....	其他标准化和为标准化的指令集

通用寄存器 (General Purpose Registers)

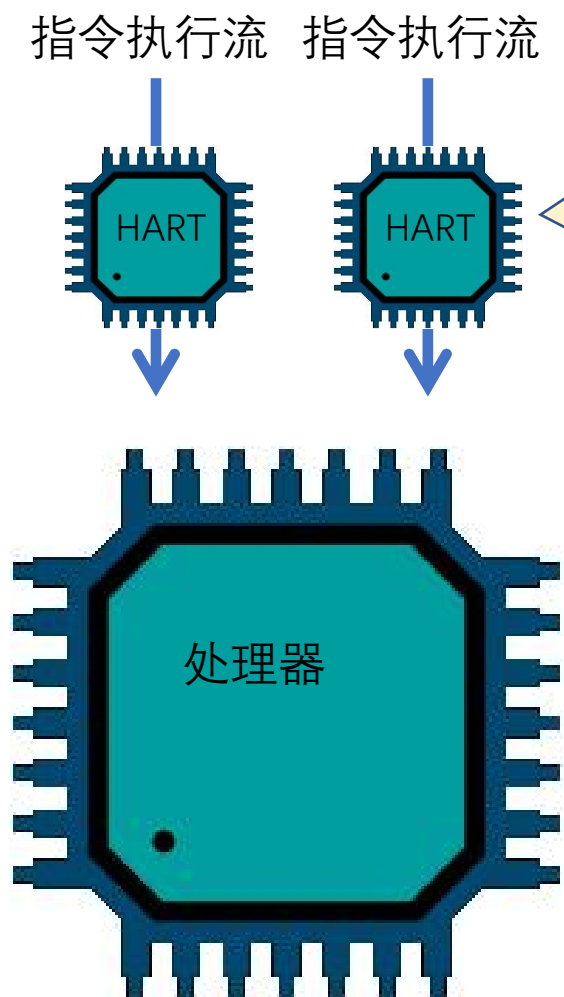
- RISC-V 的 Unprivileged Specification 定义了 32 个通用寄存器以及一个 PC
 - 对 RV32I/RV64I/RV128I 都一样
 - 如果实现支持 F/D 扩展则需要额外支持 32 个浮点 (Float Point) 寄存器。
 - RV32E 将 32 个通用寄存器缩减为 16 个。
- 寄存器的宽度由 ISA 指定
 - RV32 的寄存器宽度为 32 位, RV64 的寄存器宽度为 64 位, 依次类推。
- 每个寄存器具体编程时有特定的用途以及各自的别名。由 RISC-V Application Binary Interface (ABI) 定义。



➤ HART = HARdware Thread



➤ HART = HARdware Thread



From the perspective of software running in a given execution environment, a hart is a resource that autonomously fetches and executes RISC-V instructions within that execution environment.

-- 【参考1】 1.2 章节

- RISC-V 的 Privileged Specification 定义了三个特权级别 (privilege level)
- Machine 级别是最高的级别，所有的实现都需要支持。
- 可选的 Debug 级别

Level	Encoding	Name	Abbreviation
0	00	User/Application	U
1	01	Supervisor	S
2	10	<i>Reserved</i>	
3	11	Machine	M

【参考 2】Table 1.1: RISC-V privilege levels.

Number of levels	Supported Modes	Intended Usage
1	M	Simple embedded systems
2	M, U	Secure embedded systems
3	M, S, U	Systems running Unix-like operating systems

【参考 2】Table 1.2: Supported combinations of privilege modes.

Control and Status Registers (CSR)

- 不同的特权级别下时分别对应各自的一套 Registers (CSR)，用于控制（Control）控制和获取相应 Level 下的处理器工作状态。
- 高级别的特权级别下可以访问低级别的 CSR，譬如 Machine Level 下可以访问 Supervisor/User Level 的 CSR，以此类推；但反之不可以。
- RISC-V 定义了专门用于操作 CSR 的指令（【参考1】中定义的“Zicsr”扩展）。
- RISC-V 定义了特定的指令可以用于在不同特权级别之间进行切换（【参考1】中定义的 ECALL/EBREAK）。

Name	Description
Supervisor Trap Setup	
sstatus	Supervisor status register.
sedeleg	Supervisor exception delegation register.
sideleg	Supervisor interrupt delegation register.
sie	Supervisor interrupt-enable register.
stvec	Supervisor trap handler base address.
scounteren	Supervisor counter enable.
Supervisor Trap Handling	
sscratch	Scratch register for supervisor trap handlers.
sepc	Supervisor exception program counter.
scause	Supervisor trap cause.
stval	Supervisor bad address or instruction.
sip	Supervisor interrupt pending.
Supervisor Protection and Translation	
satp	Supervisor address translation and protection.

Table 2.3:
Currently allocated RISC-V supervisor-level CSR addresses.
【参考 2】

➤ 物理内存保护 (Physical Memory Protection, PMP)

- 允许 M 模式指定 U 模式可以访问的内存地址。
- 支持 R/W/X，以及 Lock

0x0800-0000



0x0000-0000

Physical Address Map

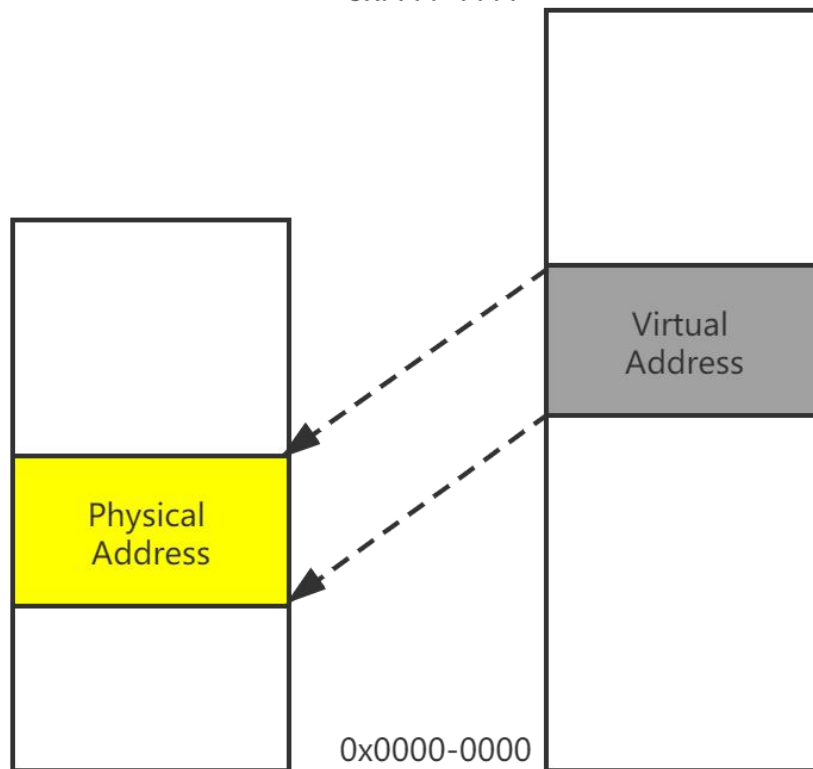
➤ 虚拟内存 (Virtual Memory)

- 需要支持 Supervisor Level
- 用于实现高级的操作系统特性 (Unix/Linux)
- 多种映射方式 Sv32/Sv39/Sv48

0xFFFF-FFFF

0x0800-0000

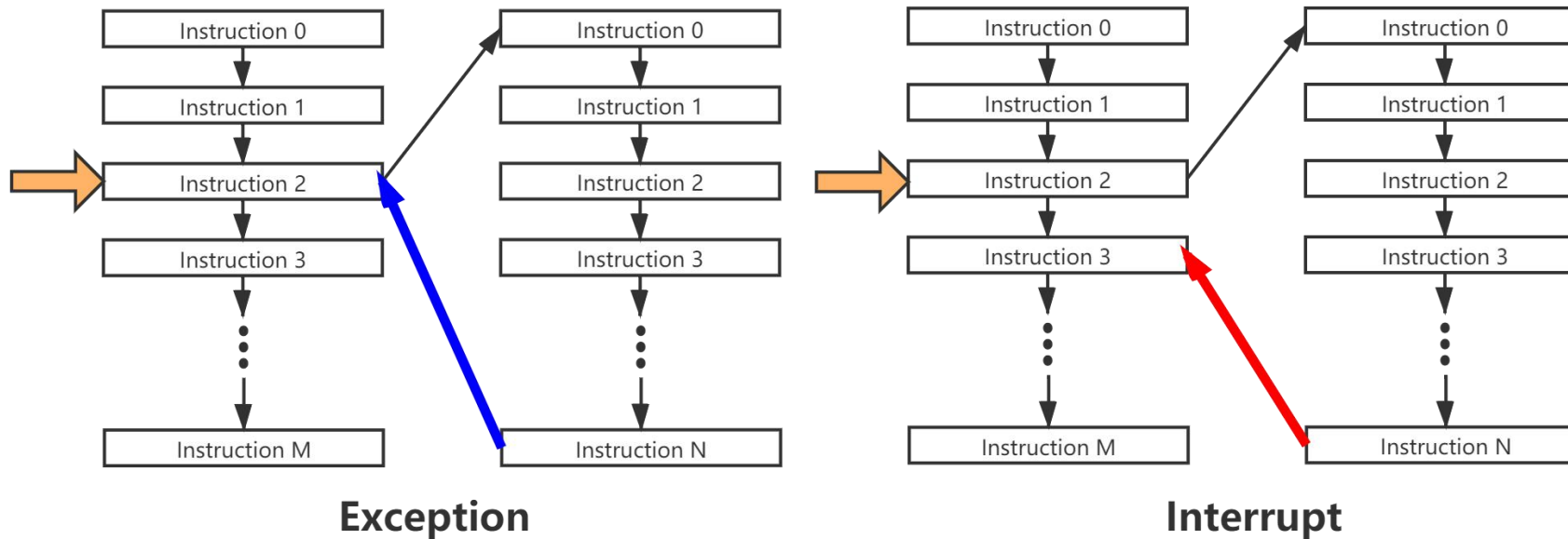
0x0000-0000



Physical Address Map

Virtual Address Map

- **异常 (Exception)** : “an unusual condition occurring at run time associated with an instruction in the current RISC-V hart”
- **中断 (Interrupt)** : “an external asynchronous event that may cause a RISC-V hart to experience an unexpected transfer of control”



谢谢

欢迎交流合作