



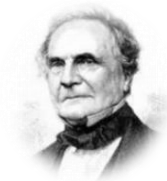
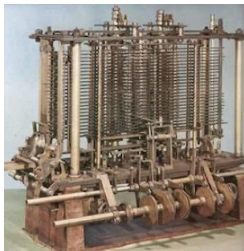
PART 01

CPU概述

- CPU是什么
- 发展历程
- 指令集架构
- 指令集授权规则

计算机的诞生 冯-诺依曼架构

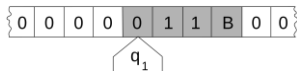
分析机
1834年



通用计算机之父

查尔斯·巴贝奇，英国数学家、发明家兼机械工程师。由于提出了差分机与分析机的设计概念

图灵机
1936年



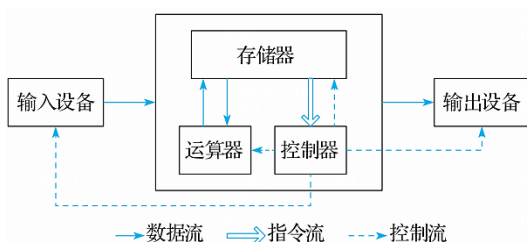
将人的计算行为抽象化为一种计算模型，能模拟人类所能进行的任何计算过程



计算机科学之父

阿兰·图灵，英国电脑科学家、数学家、逻辑学家、密码分析学家和理论生物学家

冯·诺依曼架构
1945年



现代计算机之父

冯·诺伊曼，出生于匈牙利的美国籍犹太人数学家，理论计算机科学与博弈论的奠基者

- ✓ 第一个完整的计算机体系结构
- ✓ 是实现图灵机的一种简单方法
- ✓ 业界近**70年**的使用，依然是主流架构

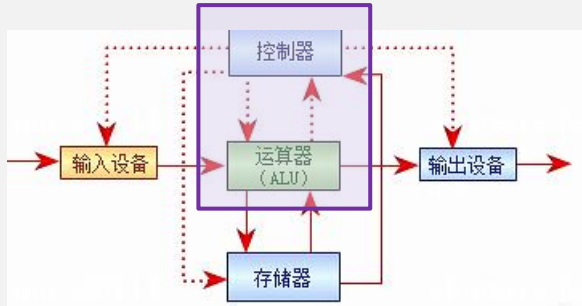
冯·诺依曼架构特点：

- 计算机由存储器、运算器、控制器、输入设备和输出设备五部分组成
- 存储器是按地址访问的线性编址的一维结构，每个单元的位数固定
- 采用**存储程序方式**，即指令和数据不加区别混合存储在同一个存储器中
- 控制器通过执行指令发出控制信号控制计算机的操作。指令在存储器中按其执行顺序存放，由指令计数器指明要执行的指令所在的单元地址
- 以运算器为中心，输入输出设备与存储器之间的数据传送都经过运算器

CPU是计算机的核心，信息时代的基石

■ **CPU** (Central Processing Unit) 中文称为“**中央处理器**”，是计算机的大脑。CPU通过内部的电路（晶体管）进行数学和逻辑运算，执行一系列指令来完成任务，这些一系列指令被称为程序。

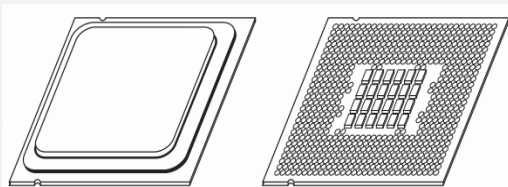
CPU = 控制器 + 运算器



计算机模型（冯诺依曼架构）

CPU主要任务

- 取指令
- 解码指令
- 执行指令
- 写回结果



称谓：

- CPU
- 处理器

CPU主要指标： 指令集 频率 核心数 缓存大小

应用场景



个人电脑



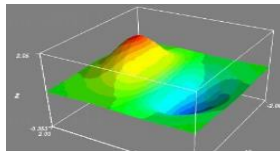
服务器



移动设备



嵌入式系统

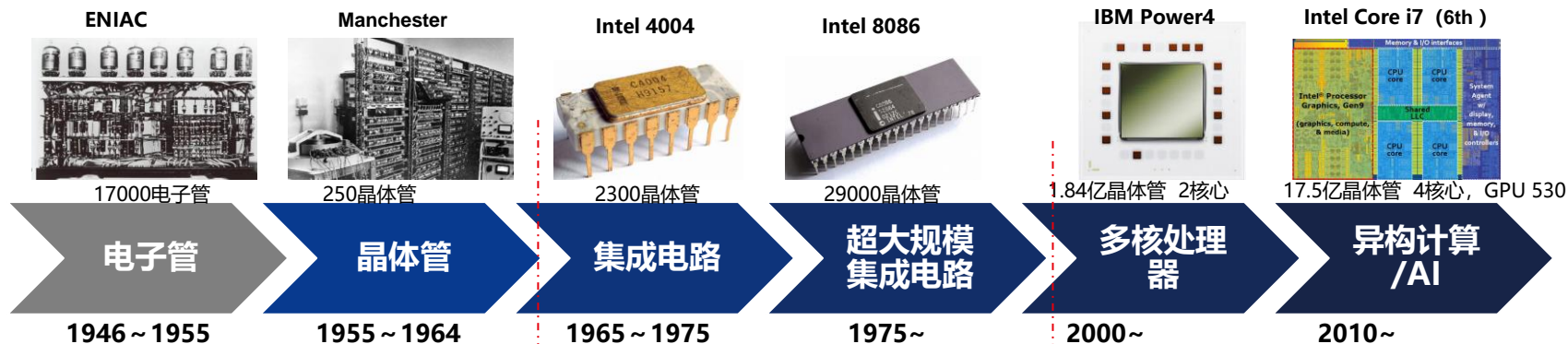


科学计算



人工智能

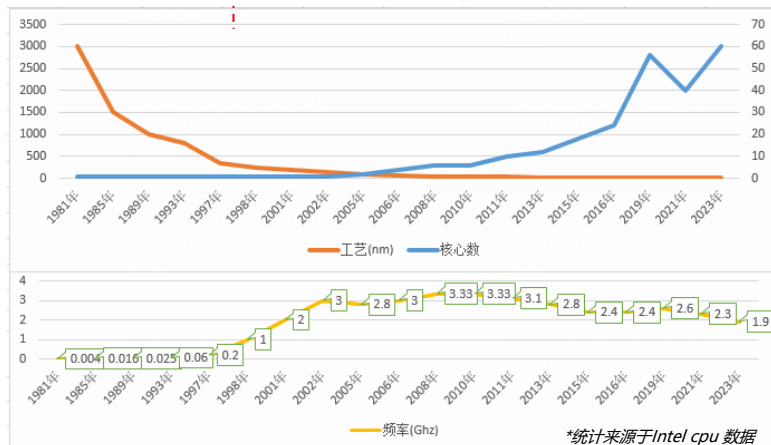
CPU的发展历程：分立器件-集成电路-多核-异构



- **摩尔定律**推动着CPU集成电路的快速发展 (1965~2000)
- 从2000年以后摩尔定律开始变缓, 采用增加核心数提升CPU性能、异构计算方案提升系统性能
- CPU核心数增多, 存在散热限制, 导致频率持续提高受限

摩尔定律 (Moore's law)

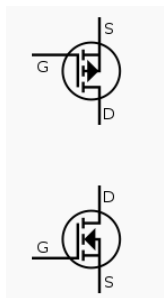
- 由英特尔 (Intel) 创始人之一戈登·摩尔提出, 其内容为: 集成电路上可容纳的晶体管数目, 约每隔两年便会增加一倍



CPU的底层组成：晶体管-门电路-加法器-ALU

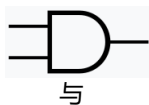
晶体管

晶体管是CPU的基本元件，用于控制电流的流动。晶体管通过控制开关状态来实现逻辑电路的功能



门电路

门电路是由多个晶体管组成的电路，用于执行逻辑运算、数据存储和控制操作。它们是CPU的构造单位，通过组合不同的门电路可以实现更复杂的功能



与



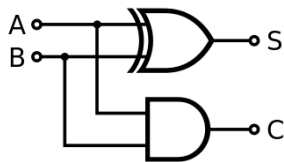
或



非

加法器

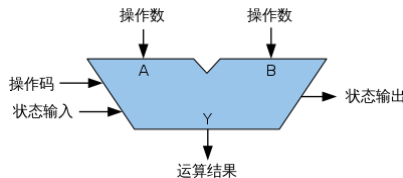
加法器是一种特殊的电路，用于对数字数据进行加法运算。它是CPU中的一个重要组成部分



输入		输出	
A	B	C	S
0	0	0	0
1	0	0	1
0	1	0	1
1	1	1	0

算术逻辑单元 (ALU)

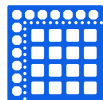
ALU是CPU的核心组件，负责执行各种算术和逻辑运算。它包含多个加法器以及其他逻辑电路，能够执行加法、减法、乘法、逻辑与、逻辑或等操作



CPU的分类

- CPU分类可以按照不同角度进行分类，常见的有按照工艺、位宽、应用场景、指令集维度进行划分

按照制造工艺



- 电子管CPU
- 晶体管CPU
- 集成电路CPU
- 超大规模集成电路CPU

按处理器位宽



- 8位CPU：处理器位宽为8位
- 16位CPU：处理器位宽为16位
- 32位CPU：处理器位宽为32位
- **64位**CPU：处理器位宽为64位

* 8位 (bit) = 1字节 (Byte)

按应用场景

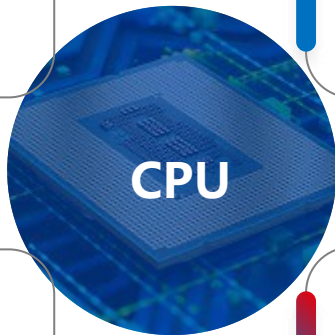


- 普通PC和服务器处理器
- 嵌入式处理器
- 移动设备处理器
- 高性能计算处理器

按照指令集



- CISC 复杂指令集：**指令数目多，单条指令功能强**，如x86处理器
- RISC 精简指令集：**指令数目少，单指令功能简单**，多指令组合完成复杂功能，如ARM、RISC-V处理器



指令集架构 和 微架构

■ 指令集架构 (ISA, Instruction Set Architecture) 是CPU硬件和软件之间的抽象接口

■ 微架构是遵循指令集架构的具体硬件实现方案

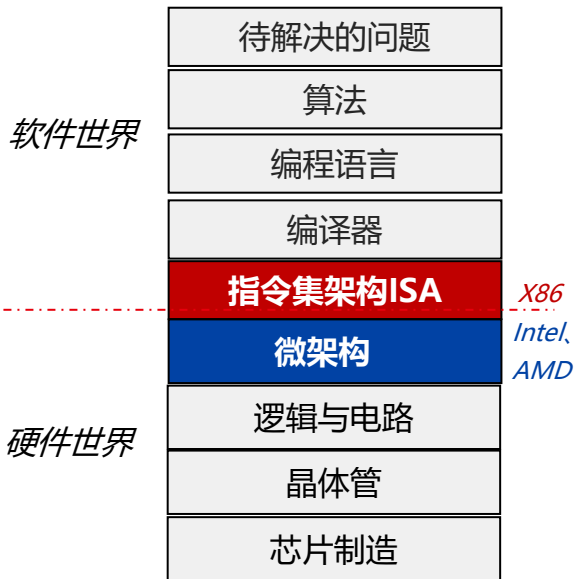
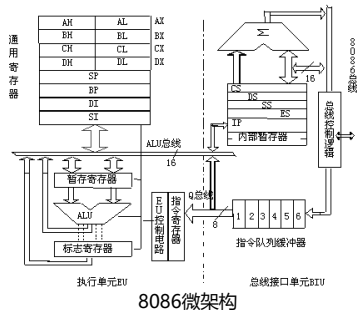
指令集架构

- 指令集：数据处理和存储、控制流、系统控制等
- 寄存器集
- 存储模型
- 数据类型
- 中断和异常处理
- 外部I/O

		指令示例	
Instruction	Format	Meaning	
add rd, rs1, rs2	R	Add registers	
sub rd, rs1, rs2	R	Subtract registers	
sll rd, rs1, rs2	R	Shift left logical by register	
srl rd, rs1, rs2	R	Shift right logical by register	
sra rd, rs1, rs2	R	Shift right arithmetic by register	
and rd, rs1, rs2	R	Bitwise AND with register	
or rd, rs1, rs2	R	Bitwise OR with register	
xor rd, rs1, rs2	R	Bitwise XOR with register	

微架构

- 微架构是处理器的具体硬件实现方案
- 遵循指令集架构，可以使用不同的硬件实现，
比如X86指令集的AMD和Intel的处理器



指令集分类

指令集架构主要分为：复杂指令集CISC，精简指令集RISC

- **CISC** (Complex Instruction Set Computing) ,一条指令完成一个**复杂的基本功能**。单条指令功能强，指令类型丰富完善，通用场景下性能具有优势
- **RISC** (Reduced Instruction Set Computing) ,一条指令完成一个**基本动作**，**多条指令组合**完成一个复杂的基本功能。指令集架构在不断完善，译码效率高，低功耗

复杂指令集

VS

精简指令集

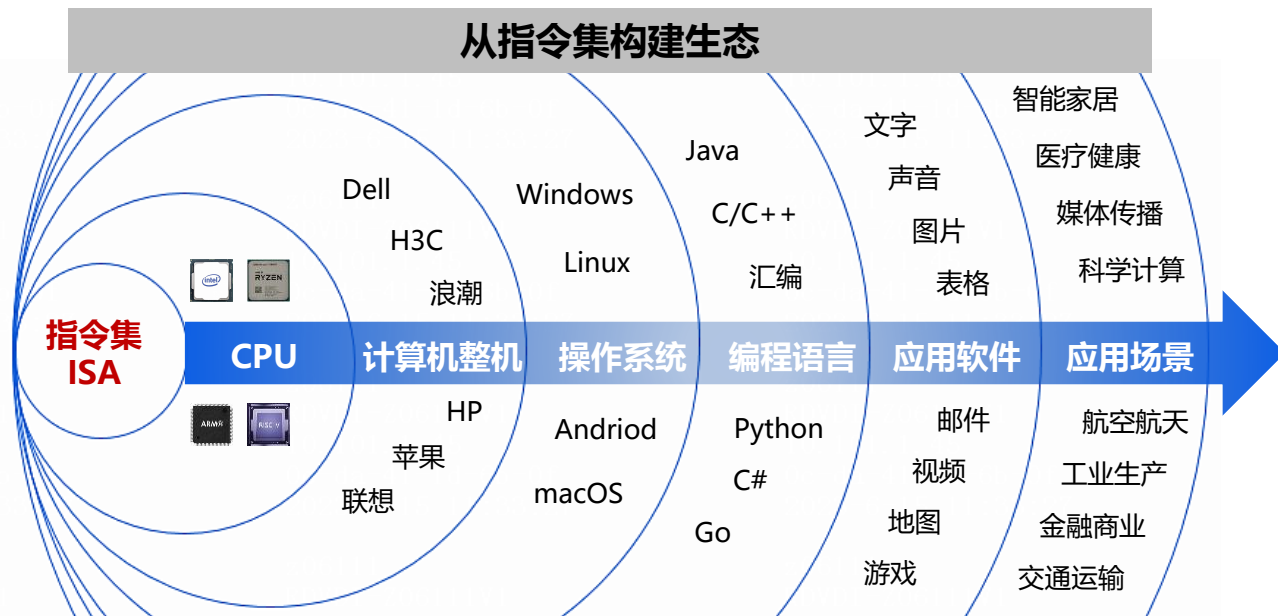
给我接杯水来

- 右转
- 向前一步
- 左转
- 向前一步
- 重复，直到饮水机旁
- 伸出左手
- 拿起水杯
- 抬起右手
- 打开出水开关
- 等待水满了关闭开关
- 左转
- 向前一步
- 右转
- 向前一步
- 重复上述步骤直到回来

架构	复杂指令集 (CISC)	精简指令集 (RISC)		
	X86	ARM	RISC-V	MIPS
主要特点	指令数繁多，复杂度高 单核高性能 功耗较高	指令集较少 单核性能较高 功耗适中	基本指令集40余条，极简 单核性能偏弱 功耗较低 扩展性好	指令集较少 单核性能较高 功耗适中
典型厂家	国外：Intel，AMD 国内：海光、兆芯...	国外：Marvell，Nvidia，Ampere... 国内：华为海思，飞腾、平头哥...	国外：英伟达、高通、Intel 国内：平头哥，华米科技，西部数据	龙芯
自主性	自主受限 兆芯/VIA：可使用2018.4前的Intel指令集 海光：授权使用AMD Zen1架构	自主受限 飞腾：v8指令集永久授权 华为：v9指令集99yr授权 平头哥：v9 N2 IP授权	开放指令集 完全开源、开放	自主指令集
指令集文档	数千页	数千页	不足300页	/
目前应用场景	服务器 、工作站、个人计算机、智算中心	服务器 、分布式存储，移动终端、消费类电子	物联网设备\嵌入式。 未来向数据中心等领域逐步渗透	桌面终端、低端服务器、消费类电子、无线通信、工业
生态	数据中心级应用生态完善，巨头垄断，部分开源	数据中心级应用逐步完善，部分开源	诞生时间短，生态发展迅速	生态不足是商业化的主要瓶颈

* RISC-V是美国伯克利大学开发一种精简指令集的第五代

指令集承载着整个生态的构建



- 指令集承载了一个生态，是生态的源头。承载着计算机整机、操作系统、编译软件（工具链）、应用软件、应用场景等
- 多因素（性能、功耗、成本、特定应用需求、商业许可等）导致多个指令集的诞生

- **X86**架构凭借其完善的生态、高性能优势，仍然是服务器产品的主流架构
- **ARM**架构随技术不断迭代，性能不断提升，功耗和成本具备优势，有望抢占X86市场更多份额
- **RISC-V**目前主要提供端测能力，挑战ARM地位，未来将会扩展至高性能计算领域，生态迅速完善中
- MIPS处理器体系结构的开发现已停止，MIPS 架构即将落幕

X86指令集：强大的狮子

X86指令集

是一种被广泛使用的计算机指令集架构，最初由Intel公司在1970年代末开发。名称“x86”来源于Intel的一系列早期微处理器（例如8086、80186、80286、80386、80486），这些处理器的名字都以“86”结尾

主要厂家

- Intel
- AMD --> 海光
- VIA (Cyrix) --> 兆芯

主要特点

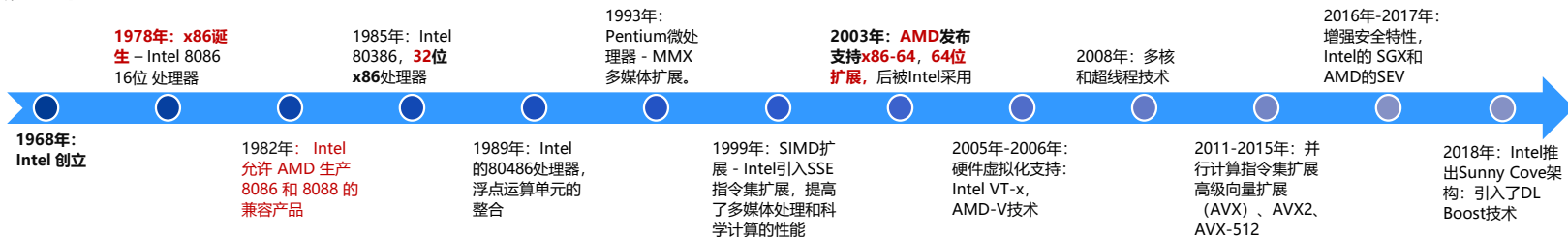
- 复杂指令集
- 性能高
- 兼容性好

统治领域 (~90%份额)

- 个人计算机
- 服务器领域



发展历程



X86成功主因: PC时代 + 商业模式 + 技术

Intel Tick-Tock战略：分而治之 持续领先

- Tick-Tock模型（通常译作“嘀嗒模式”或“钟摆模式”）是Intel公司提出的CPU发展路线，其含义是采用“两步走”的交替策略，应用**先进制造工艺**和**改进微架构**设计来提升CPU性能



Tick：制程升级

- 使用新的制程技术（如，从45纳米制程转到32纳米制程）来制造其处理器。这些新处理器的微架构和前一代产品基本相同，但制程的改进带来了功耗减少、体积缩小和性能提升

Tock：架构升级

- 在这个阶段，Intel在前一阶段的制程技术基础上，推出新的微架构。新的微架构通常会引入新的指令集、优化的性能、更高效的电力管理等特性

Process-Architecture-Optimization (PAO) 2017~



PAO “三步走” 战略

- “制程—架构—优化”（Process-architecture-Optimization）的，每次迭代周期拉升到3年。增加的“优化”步骤是指在制程及架构不变的情况下，进行小幅度的修复和优化，以及修正bug、提升主频等

*2021.3月 intel重启tick-tock战略

这种“分而治之”策略使得Intel能够以一种**可预见的方式持续地提升其处理器的性能**，**同时降低推出新技术时的风险**。由于技术的复杂性不断提高，后来Intel在维持这种“Tick-Tock”节奏上遇到了挑战，目前已经改变了这种战略，转向了更灵活的产品更新周期

MIPS指令集：优雅的孔雀



MIPS指令集架构（Microprocessor without Interlocked Pipelined Stages）是一种基于RISC（精简指令集）原则的处理器指令集架构。MIPS的设计目标是实现高性能、低成本以及高效能源的处理器解决方案，特别是对于嵌入式系统和低功耗应用

主要应用场景

- 工作站、服务器
- 嵌入式系统
- 科研教学



主要厂家

- MIPS → 龙芯

主要特点

- 精简指令集
- Load/store架构
- 大量通用寄存器

发展历程

1981年：斯坦福大学 John Hennessy教授领导开始研究MIPS

1985年：MIPS发布R2000，全世界第一款商用的RISC处理器

1989年：MIPS推出R4000处理器，世界第一款商用的64位RISC处理器。

1998年：MIPS被SGI剥离出来，继续开发和销售MIPS处理器。

2013年：推出了第五代MIPS架构，也就是MIPS64和MIPS32 Release 5

2020年：Wave Computing申请破产

1984年：John Hennessy教授团队创建MIPS Technologies

1986年：MIPS推出R3000，在许多领域取得了巨大成功，包括工作站、服务器和嵌入式系统

1992年：MIPS被SGI收购

2012年：MIPS被Imagination Technologies收购 Arm收购了其接近500项专利

2017年：MIPS被Wave Computing收购，其指令集架构被开源，为了与ARM和RISC-V的竞争

2021年3月，Wave Computing宣布停止MIPS架构的开发。该公司已加入RISC-V基金会，未来的处理器设计将基于RISC-V架构

MIPS因商业运作原因，不敌X86和ARM，已逐步走向没落

ARM指令集：稳扎稳打的蚂蚁

ARM

(Advanced RISC Machines) 指令集是一种使用广泛的处理器架构，其以低功耗和高性能而著名，尤其在移动和嵌入式系统中广泛应用

主要应用场景

- 移动终端
- 嵌入式系统
- -> 服务器



主要厂家：

包括Apple、Intel、IBM、LG半导体、NEC、SONY、飞利浦、Atmel、Broadcom、Cirrus Logic、NXP、Actions、三星、华为、高通等

主要特点：

- 精简指令集
- 低功耗
- 高性能

发展历程

1983年：ARM公司成立，Acorn公司开发一种新的处理器架构，以满足其个人电脑的需求

1986年：ARM2处理器问世应用

1994年：ARM7处理器，在嵌入式设备和移动电话中取得了巨大的成功

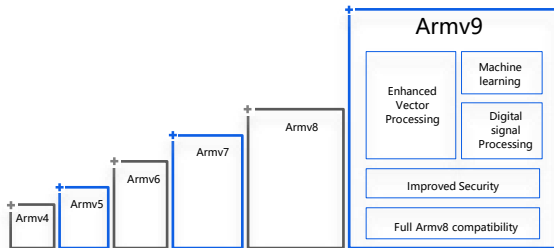
2004年：ARMv7架构和Cortex系列的问世 - ARMv7架构引入了Thumb-2指令集，NEON SIMD扩展，以及硬件浮点运算的支持

2013年：Marvell发布了基于ARMv7架构的Armada XP处理器，专门针对微型服务器

2016年：SoftBank收购ARM - 日本的SoftBank集团以约320亿美元的价格收购ARM Holdings

2018年：Amazon推出了Graviton处理器

2020年：Apple发布M1芯片的Mac，ARM架构进入了主流的个人电脑市场。



1985年：ARM1- 首款ARM处理器，主要用于内部的开发和测试

1990年：独立的ARM创建—Advanced RISC Machines Ltd (即ARM Holdings)

2001年：ARM进入3G时代 - 推出Cortex-A系列，为高性能处理器

2011年：ARMv8架构，64位

2014年：Cavium发布了基于ARMv8架构的ThunderX系列处理器，高性能计算、大数据、存储、安全以及网络等领域

2016年：Qualcomm发布了Centriq 2400系列处理器，这是基于ARMv8架构，特别是针对数据中心市场

2019年：华为发布了鲲鹏920服务器处理器

2021年：ARMv9架构推出，AI、矢量和DSP性能改进、安全性

ARM成功主因：商业模式+ 时代

RISC-V指令集：自由的麻雀

RISC-V

RISC-V (读作 “risk-five”) 是一种开源的指令集架构 (ISA) 基于精简指令集计算 (RISC) 原则。2010年由加州大学伯克利分校的研究人员提出，其目标是创建一个现代的、通用的、开源的指令集架构，能够满足当前和未来的计算需求

主要应用场景

- AIoT
- 嵌入式系统
- -> 服务器

主要厂家：

RISC-V已经在全球范围内得到了广泛的接受，有许多公司和组织正在研究和开发基于RISC-V的产品，包括当今主流芯片厂家

主要特点：

- 精简指令集、开放
- 模块化设计
- 适用于各种设备



<https://riscv.org/>
RISC-V

发展历程

2010年： RISC-V 项目在加利福尼亚大学伯克利分校启动

2014年： RISC-V 基金会成立

2017年： SiFive推出了世界上首款商用的RISC-V 微处理器，Freedom E310。

2019年： RISC-V 基金会发布了 RISC-V 的向量扩展 (RVV)。平头哥高性能的RISC-V 处理器**玄铁910**

2021年： IBM在其新的Power10 中使用 RISC-V

2011年： RISC-V 的第一版规格发布。这个版本定义了基本的整数指令集，包括32 位 (RV32I) 和 64 位 (RV64I) 的版本。

2015年： RISC-V 的浮点指令集 (RV32F 和 RV64F) 发布。Google, HPE, Microsoft, IBM, Oracle, NVIDIA, Qualcomm 等加入了 RISC-V 基金会

2018年： RISC-V 基金会发布了 RISC-V 的嵌入式指令集 (RV32E)，Western Digital 宣布将在其所有的存储产品中使用 RISC-V 处理器

2020年： 总部迁移到瑞士
NVIDIA 计划在其 GPU 和 AI 产品中使用 RISC-V

2022年： Intel加入基金会为高级会员
目前基金会成员超**3,100**

RISC-V开放、自由，正在发展中。。。

指令集授权规则

- **架构授权**：允许公司遵循指令集架构规范的基础上，设计和开发自己的处理器
- **核心授权**：授权已经设计好的处理器核心。可以根据需要选择不同的处理器核心集成到自己的芯片中。核心分为软IP核、硬IP核

指令集	(指令集) 架构授权	(IP) 核心授权
RISC-V (开放)	任何组织和个人可 免费 使用 (根据指令集规范自行设计芯片)	<ul style="list-style-type: none">• 有开源免费的IP，也有商业公司提供收费的IP• 美国公司IP可能会有出口管制，或者不符合国产化要求
X86 (非开放)	主要存在于英特尔和AMD之间， 没有公开的授权规则 <ul style="list-style-type: none">• “交叉许可协议”的契约（未公开），这种协议允许两家公司互相使用对方的X86专利技术。• 许可内容：协议允许AMD使用英特尔的x86指令集，同时也允许英特尔使用AMD开发的x86指令集的扩展（例如，AMD最早将x86指令集扩展到64位，称为AMD64或x86-64，英特尔后来也采用了这个扩展） 其他：海光 来源于AMD，兆芯来源于VIA	
ARM (非开放)	<ul style="list-style-type: none">• 架构和核心，两种授权模式都有• 费用：授权费+版税。具体根据需要商务谈判，确认授权价格 和商务模式• ARMV9新架构不再提供永久授权（本质上是增加收入）• 高性能IP Neoverse V系列受美国出口管制	

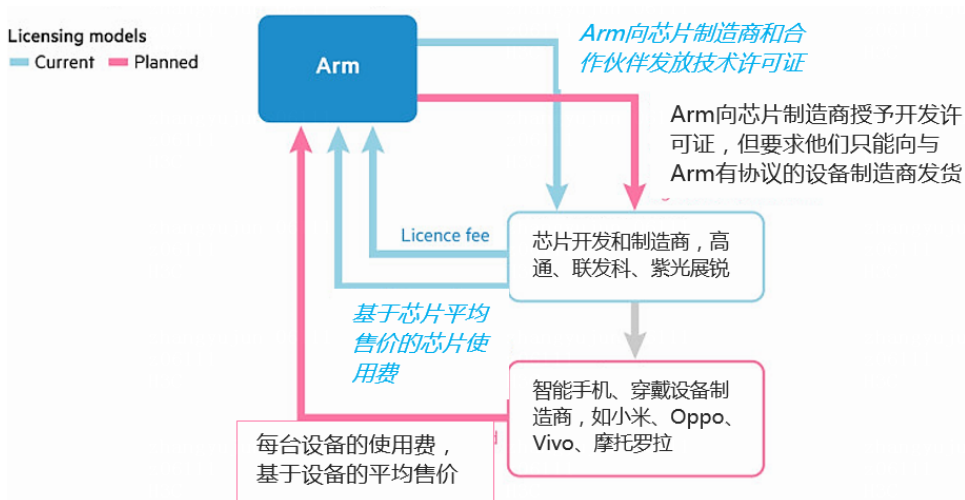
ARM授权模式可能改变

版税 = 芯片价格 * 费率 * 出货量



版税 = **整机价格** * 费率 * 出货量

ARM的授权模式可能的改变方式



对中国公司的影响和警示

- ARM公司对许可的授予和终止有非常大的**自由裁量权**
- ARMv9指令集架构**不再新提供永久授权**, 指令集升级后授权, **可持续性存在风险**
- 受美出口管制, ARMv9的先进性能IP不对中国企业授权。如Neoverse V系列
- ARM能对整机厂商进行“**长臂管辖**”
- ARM的新规则可能剥夺芯片企业自主发展权 (2024年后基于ARM公版CPU的SoC, 不可以使用外部GPU、NPU或ISP)