

# 芯动力——硬件加速设计方法

## 第七章 基于平头哥E902处理器的SoC设计

### ——(1)无剑100 SoC体系架构概述

邸志雄@西南交通大学

zxdi@home.swjtu.edu.cn

slides与源代码网址 <http://www.dizhixiong.cn/class5/>

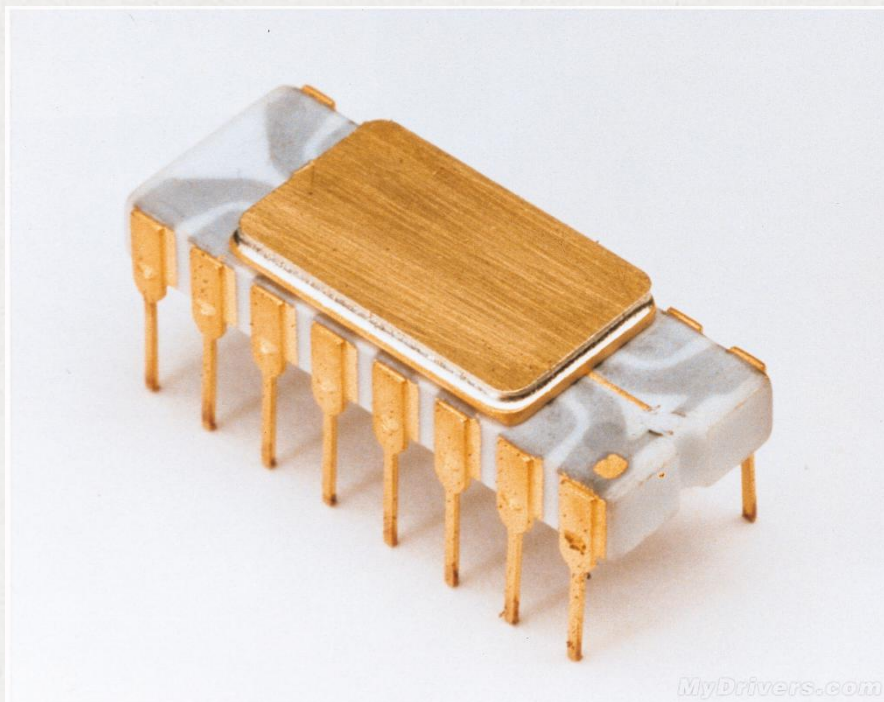
# 集成电路发展趋势



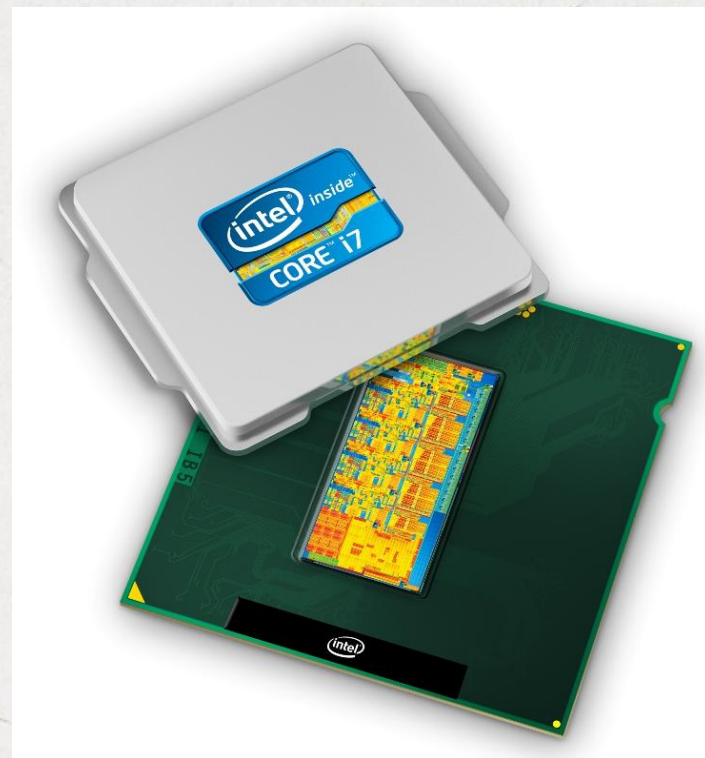
十年间，除了半导体工艺的进步，还有哪些技术推进了集成电路技术的发展？



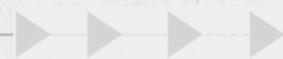
# 集成电路发展趋势



Intel 4004

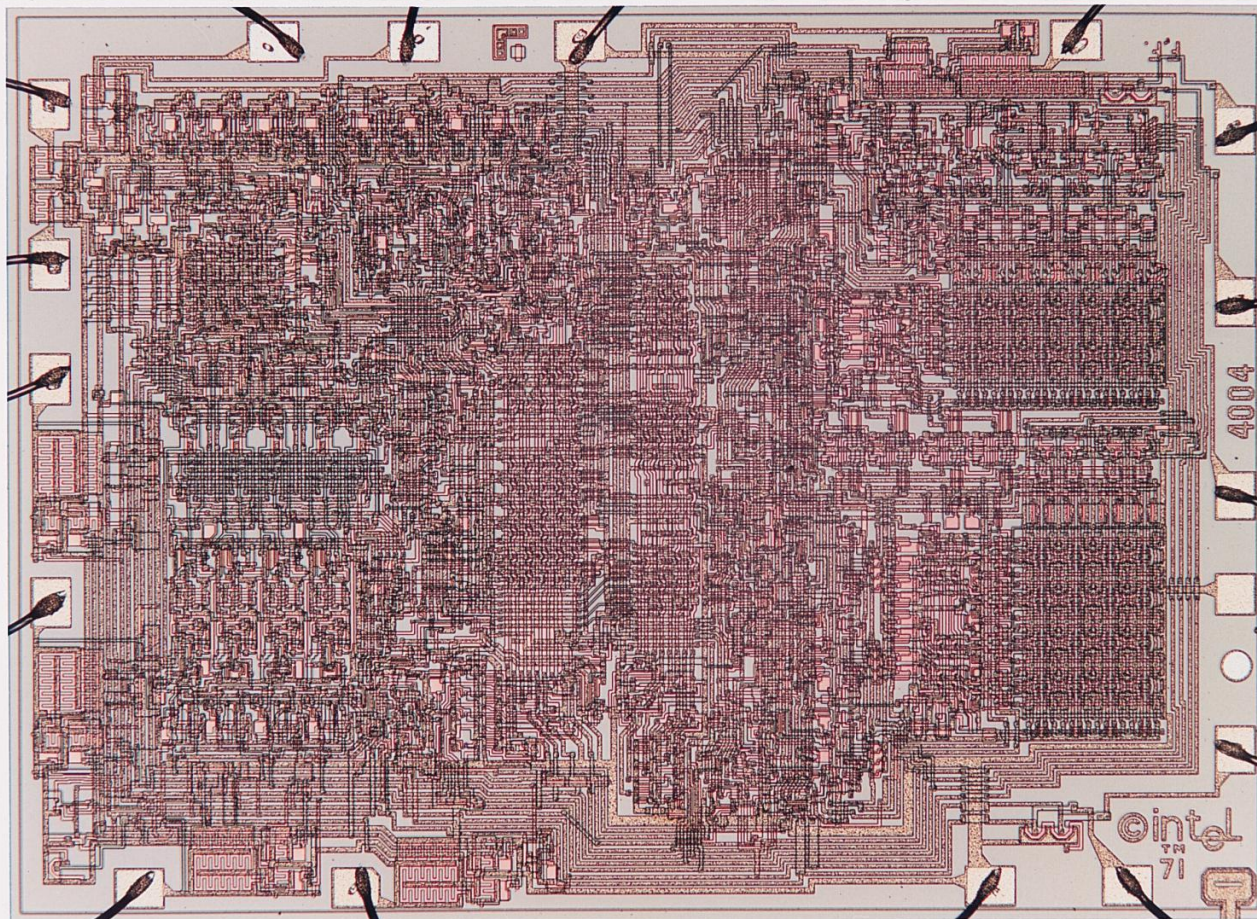


Sandy Bridge Core i7





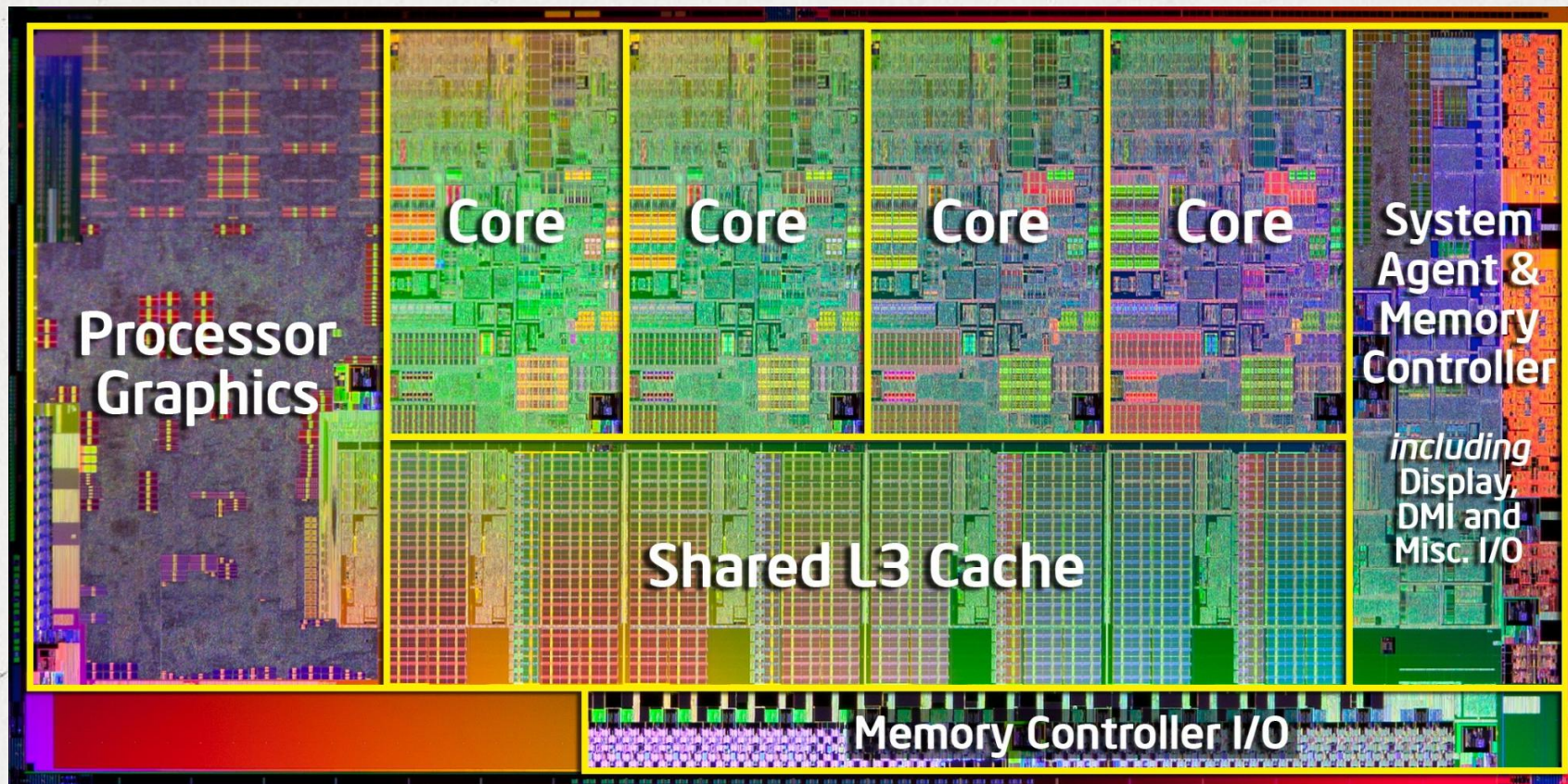
# 集成电路发展趋势



Intel 4004: 740KHz(0.74MHz), 10微米



# 集成电路发展趋势



Sandy Bridge Core i7处理器：3.4GHz，32纳米



4004



2300个晶体管

Core i7

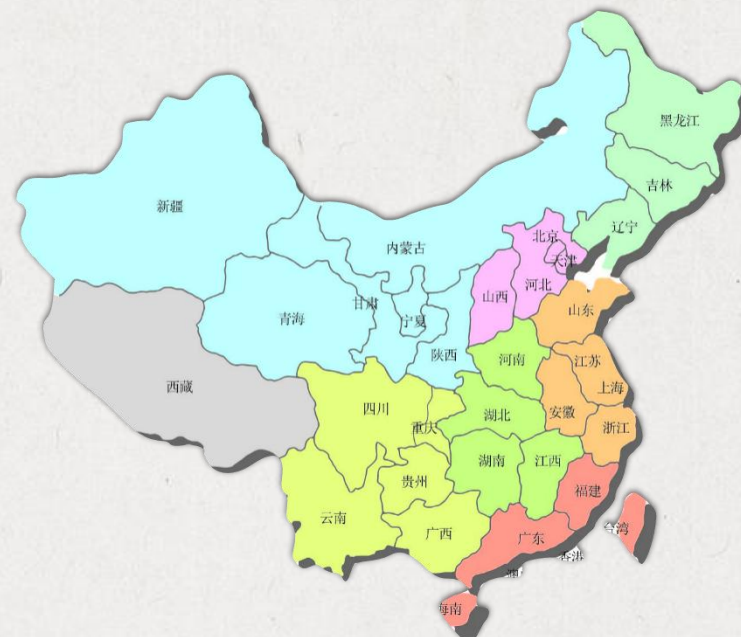


9.95亿个

# 集成电路发展趋势



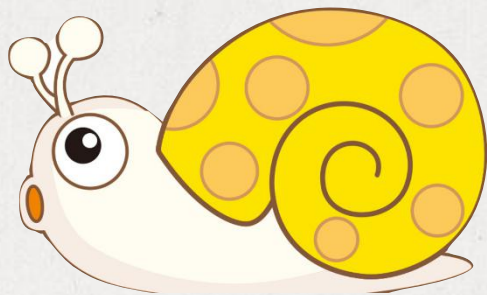
4004: 2300个



Core i7: 9.95亿个



# 集成电路发展趋势

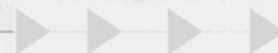


4004: 740KHz(0.74MHz)



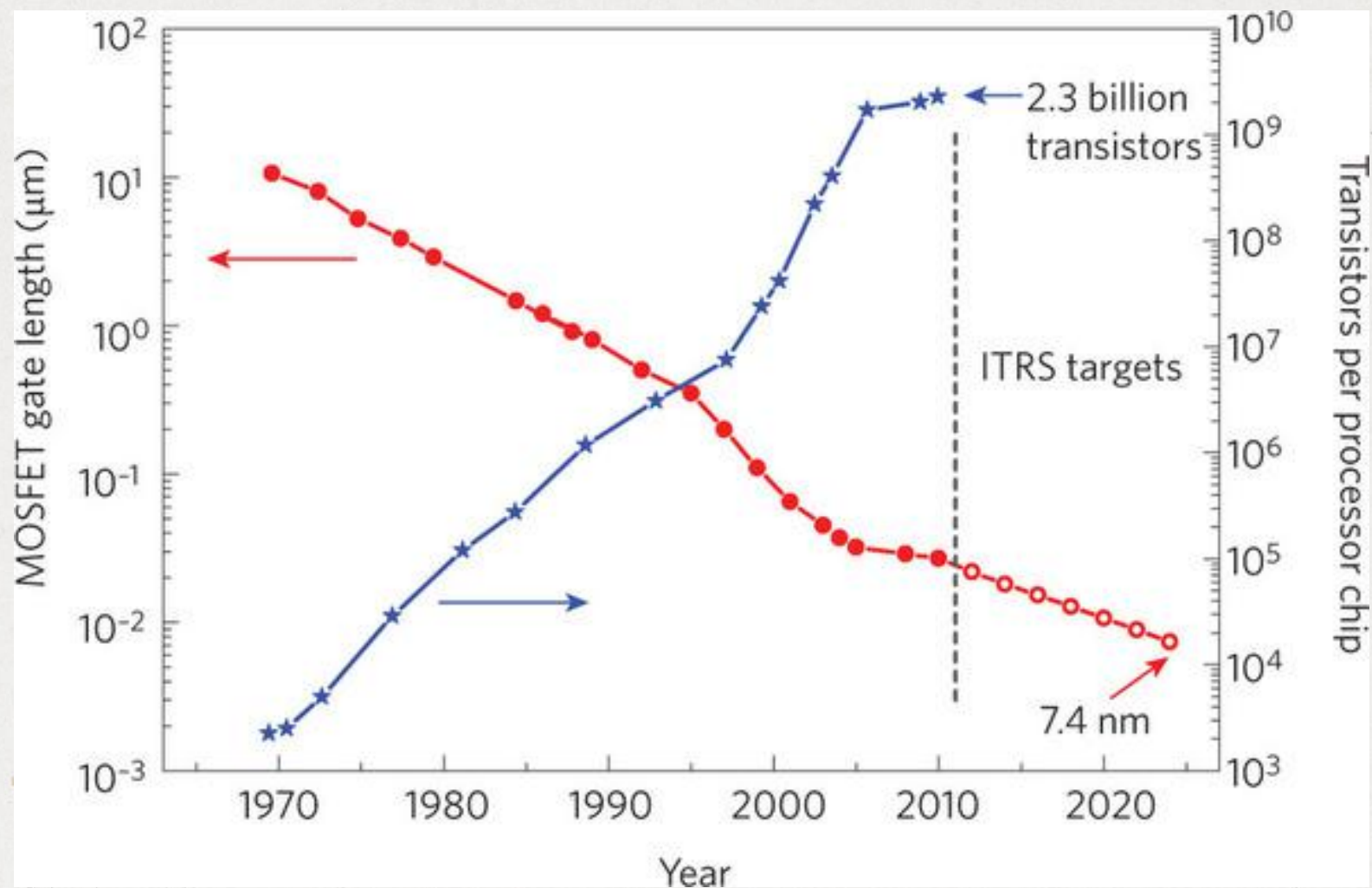
Core i7: 3.4GHz

从4004到Sandy Bridge, 晶体管的速度提升了5000倍, 功耗只有当初的5000分之一, 价格则降低到了50000分之一。





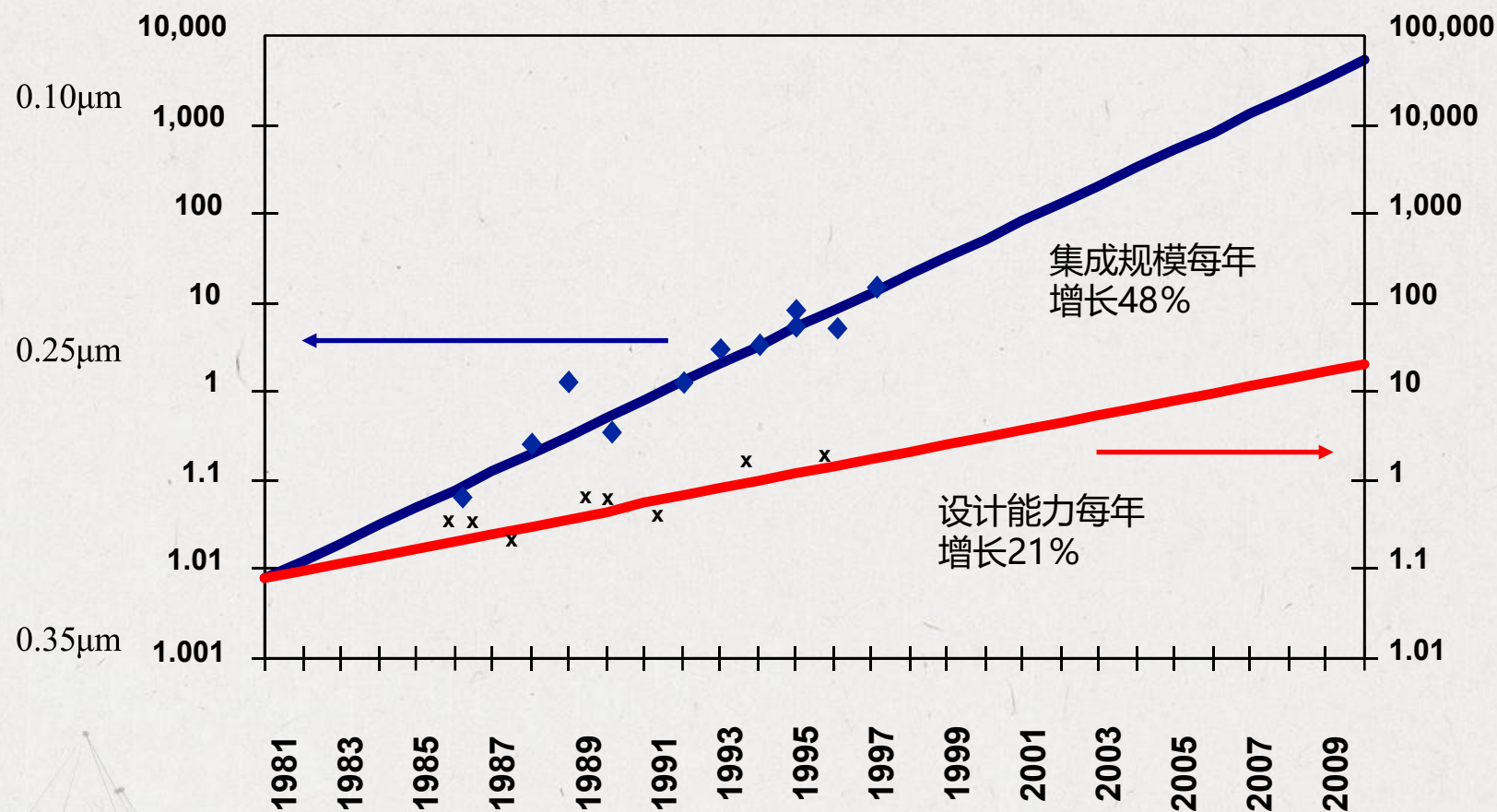
## 晶体管数量集成趋势





# 设计能力的增加跟不上芯片规模的提升

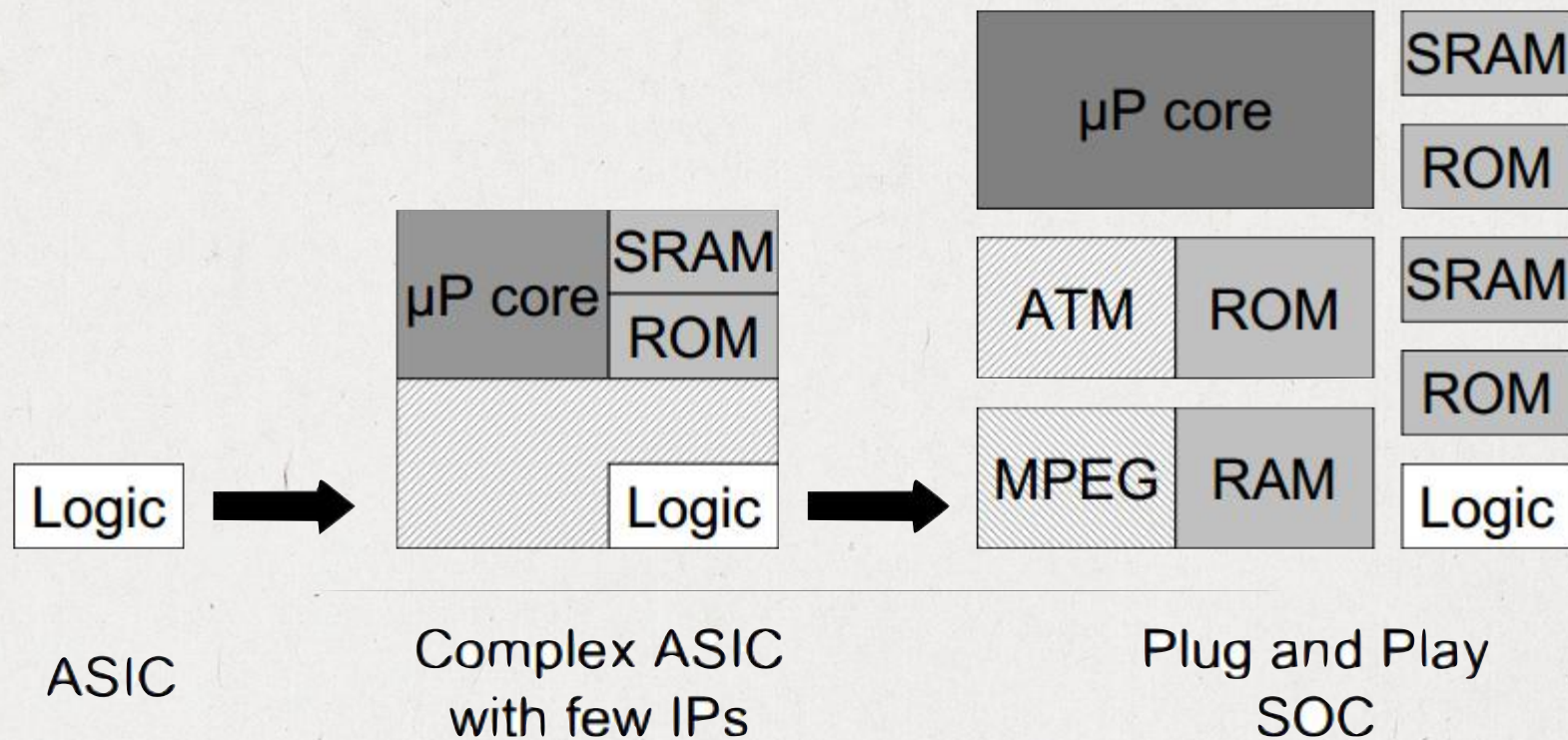
集成规模（每个数字IC芯片上的晶体管数）（M）



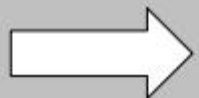
设计能力（每个设计人员每月设计的晶体管数）（M）



# 设计集成化



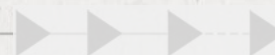
Timing-Driven  
Design



Block-Based  
Design



Platform-Based  
Design







## 推动芯片技术发展的技术

半导体工艺

SoC技术



## 设计方法学的发展

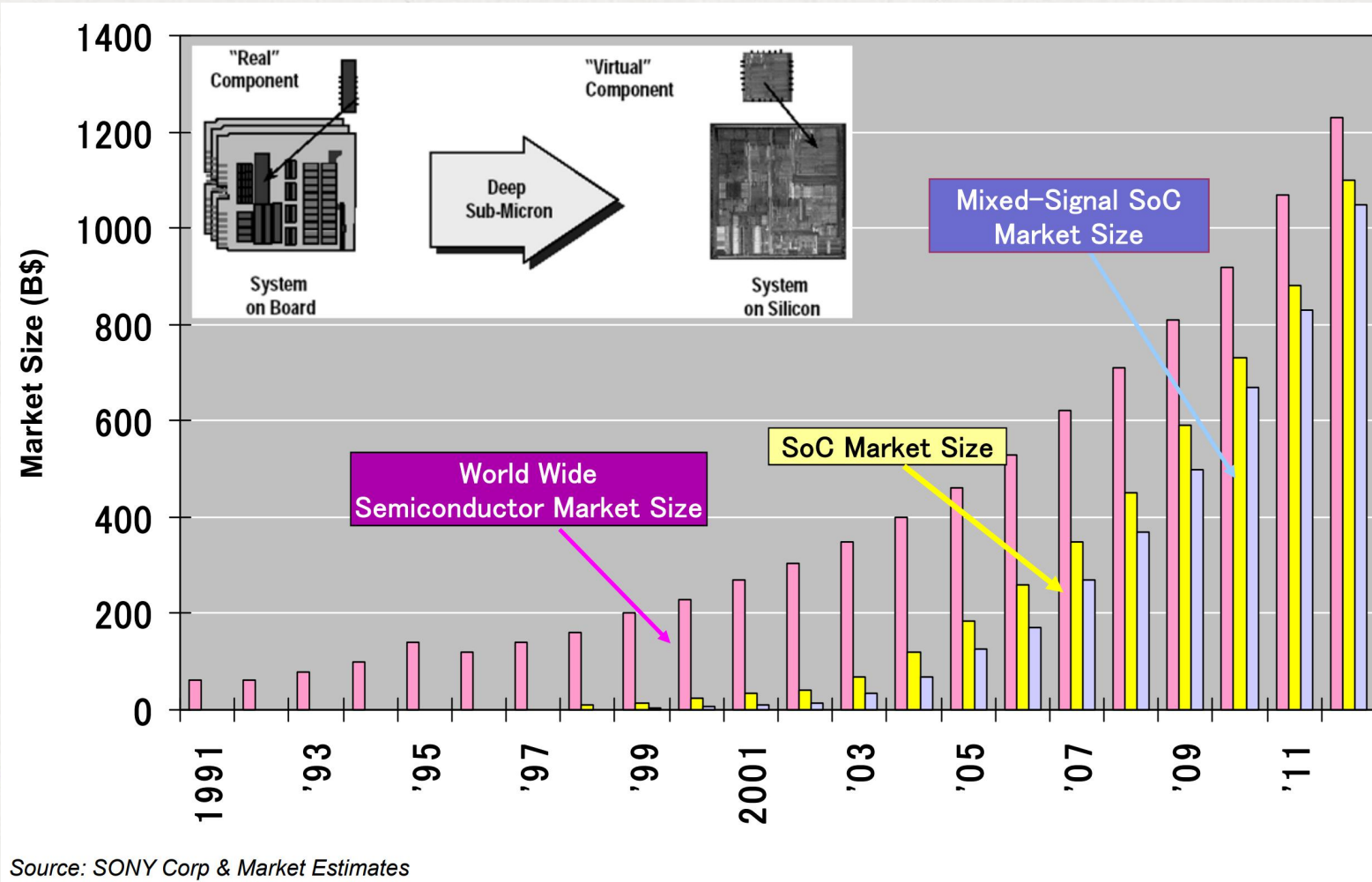


```
graph TD; A[设计方法学的发展] --> B[提高设计抽象层次]; A --> C[兼容并包];
```

提高设计抽象层次

兼容并包

# SOC设计的时代





# SoC定义及特征

SoC (System on Chip) 将构成一个系统的软/硬件集成在单颗的IC芯片里。

一般认为，如果一个集成电路芯片具有如下特征的话，即可称其为SoC，这些特征是：

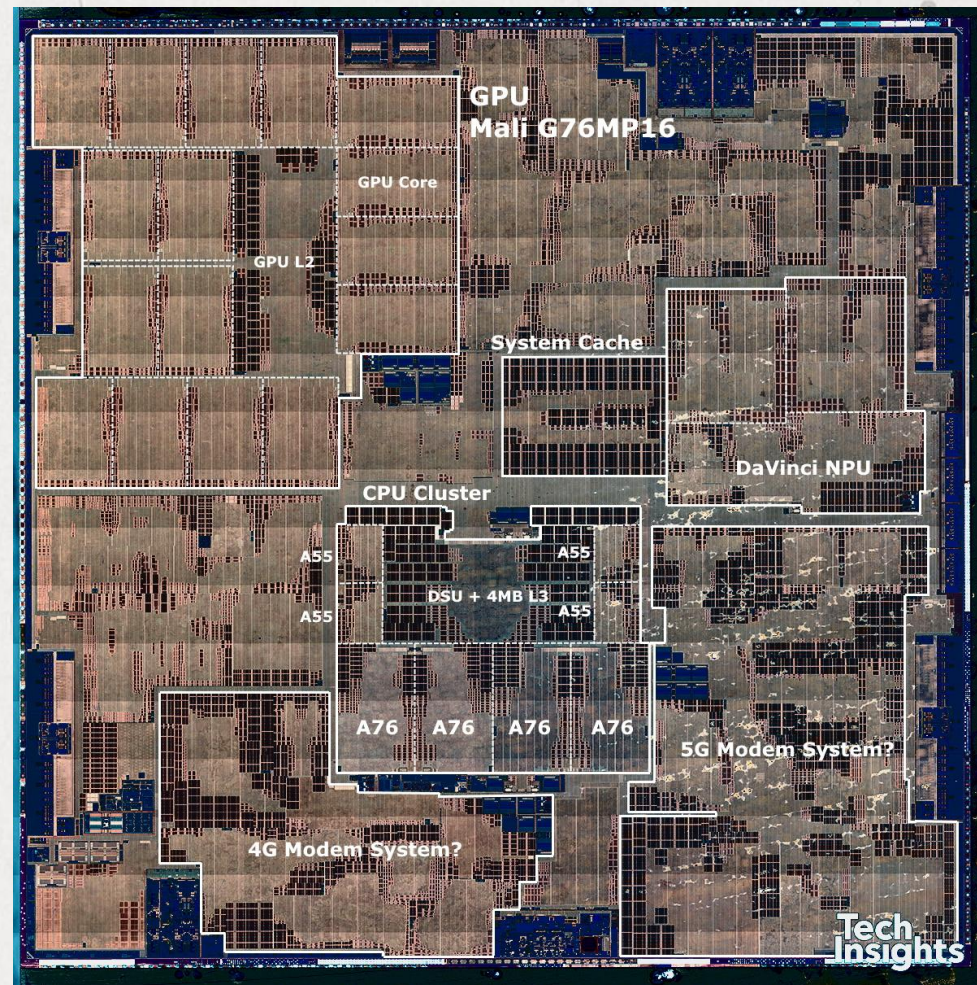
- 采用超深亚微米工艺技术实现复杂系统功能的VLSI
- 使用一个或多个嵌入式处理器或数字信号处理器 (DSP)
- 具备外部对芯片进行编程的功能
- 主要采用第三方的IP核进行设计

为了满足上市时间 (Time to Market) 的要求，目前大部分数字芯片都采用了SoC的设计方法。



# 麒麟990 SoC-2019年

- 7nm+制程工艺
- 103亿个晶体管，是世界上第一个晶体管数量过百亿的移动SoC
- 2个超大核+2个大核+4个小核三档CPU能效架构
- 16核Mali-G76 GPU
- 采用达芬奇架构NPU，创新设计NPU大核+NPU微核架构







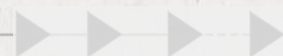
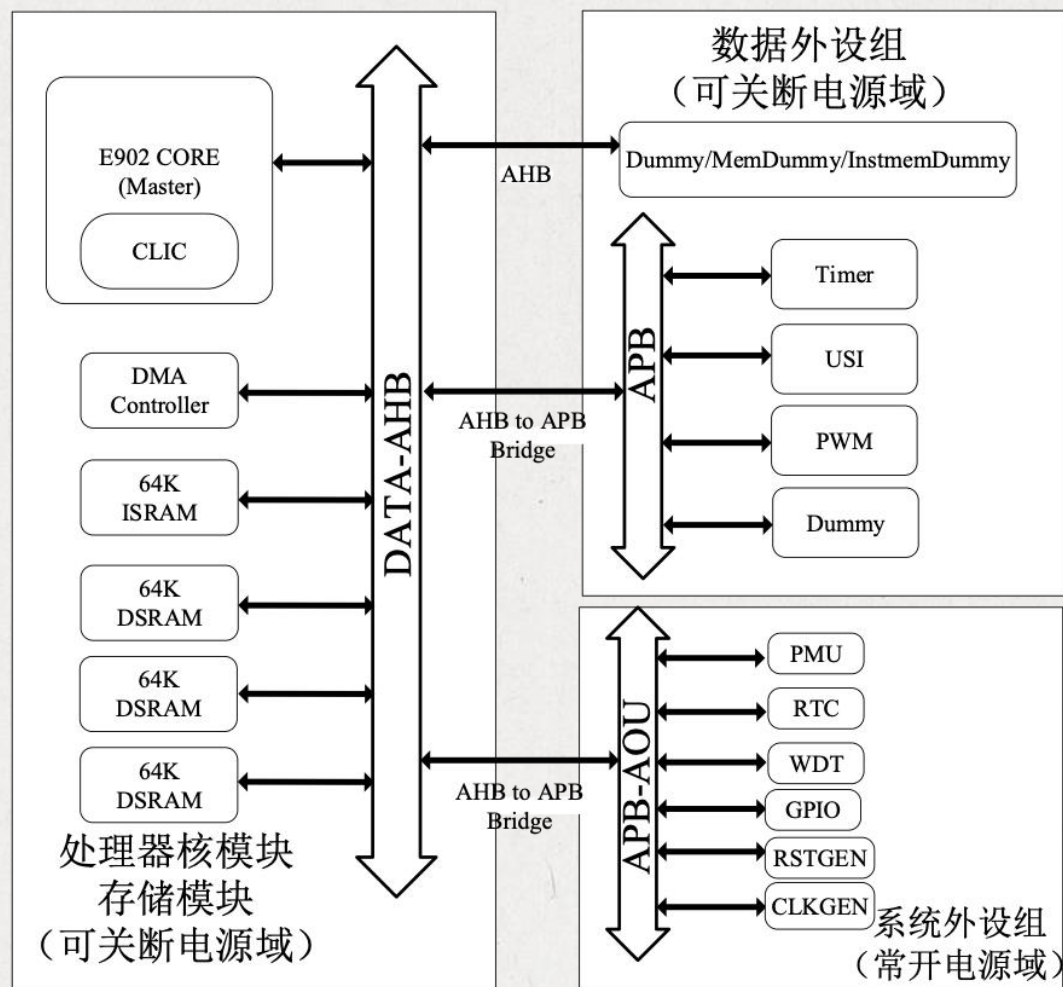
# SoC核心构成

处理器IP

硬件专用IP

互联总线

- SoC一般移植了操作系统，具备软硬件协同计算能力。
- 软硬件协同设计与协同验证在SoC设计中得到广泛应用。



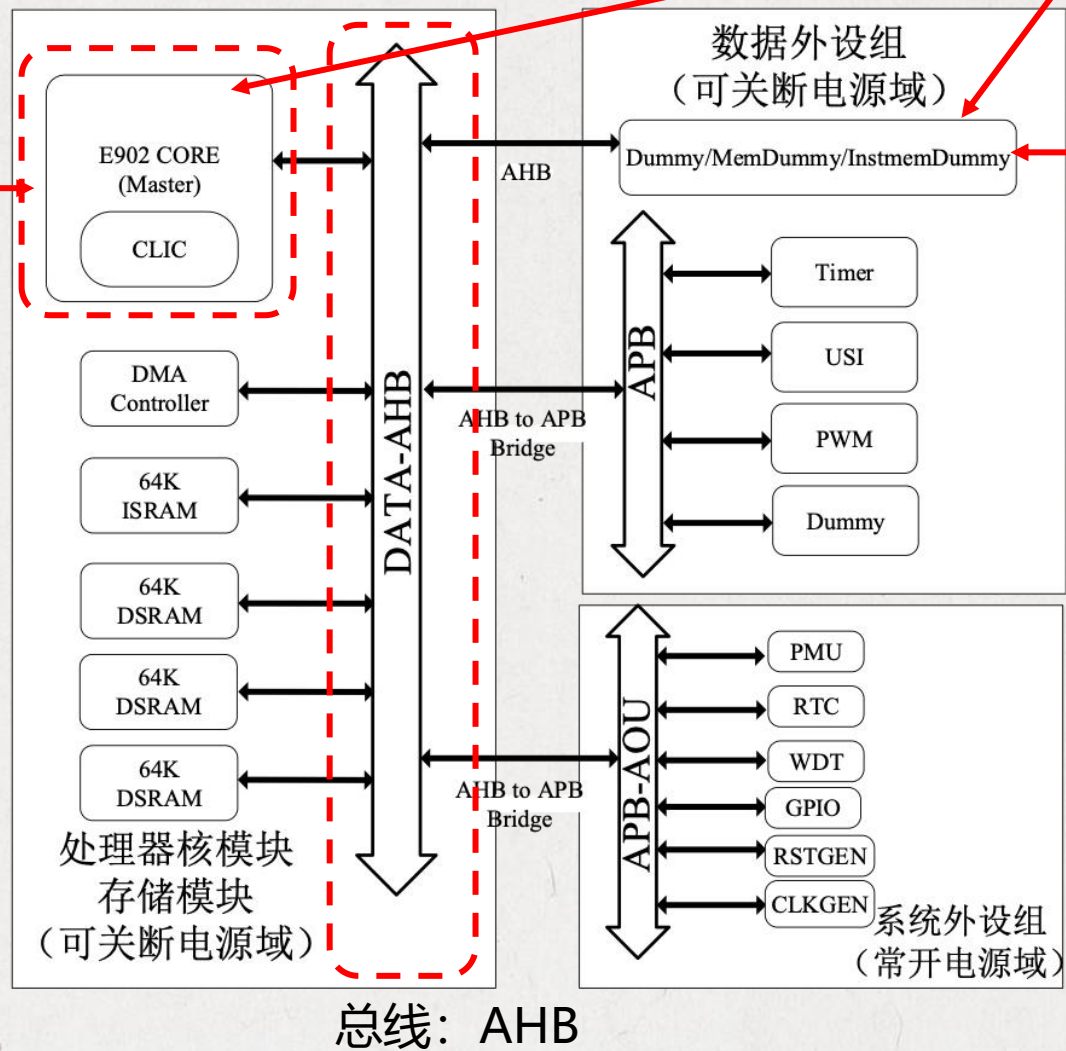
# SoC核心构成

处理器：  
RSIC-V玄铁E902

操作系统RT-thread

软硬件协同计算

硬件专用IP





# RSIC-V处理器概述

## RISC-V 起源

2010 年，加州大学伯克利分校的 David Patterson 教授与 Krste Asanovic 教授研究团队正在准备启动一个新项目，需要选择一种处理器指令集。他们分析了 ARM、MIPS、SPARC、X86 等多个指令集，发现它们不仅设计越来越复杂，而且还存在知识产权问题。

于是伯克利的研究团队临时组建一个四人小组，开展一个为期3个月的暑期小项目——从零开始设计一套全新的指令集！这个小项目的目标是新指令集能满足从微控制器到超级计算机等各种尺寸的处理器，能支持从 FPGA 到 ASIC 到未来器件等各种实现，能高效地实现各种微结构，能支持大量的定制与加速功能，能和现有软件栈与编程语言很好的适配。还有最重要的一点就是要稳定——不会改变，不会消失。



[引用自：开放指令集与开源芯片发展报告, 中国开放指令生态联盟, 2019年2月]

## 2011 年 5 月，第一版指令集正式发布

### 基础指令集

- 只包含了不到 50 条指令
- 可以用于实现一个具备定点运算和特权模式等基本功能的处理器。

### 扩展指令集

- 常用的原子操作指令、浮点运算指令等
- 用户也可以进行自定义

这套指令集既保留了“简单”这个大优点，又赋予了用户足够的灵活性。



RISC-V: The Free and Open RISC  
Instruction Set Architecture



伯克利研究团队在发布时做了两个决定：

- 将新的指令集命名为 RISC-V（读作 RISC-Five），表示为第五代 RISC（精简指令集计算机）。

每一代 RISC 处理器都由 David Patterson 教授领导与参与。也正是他与学生 David Ditzel 在 1980 年发表的那篇经典论文 “The case for the reduced instruction set computer” 中创造了 RISC 一词。

- 将 RISC-V 指令集彻底开放，使用 BSD License 开源协议设计了开源处理器核 Rocket Core。

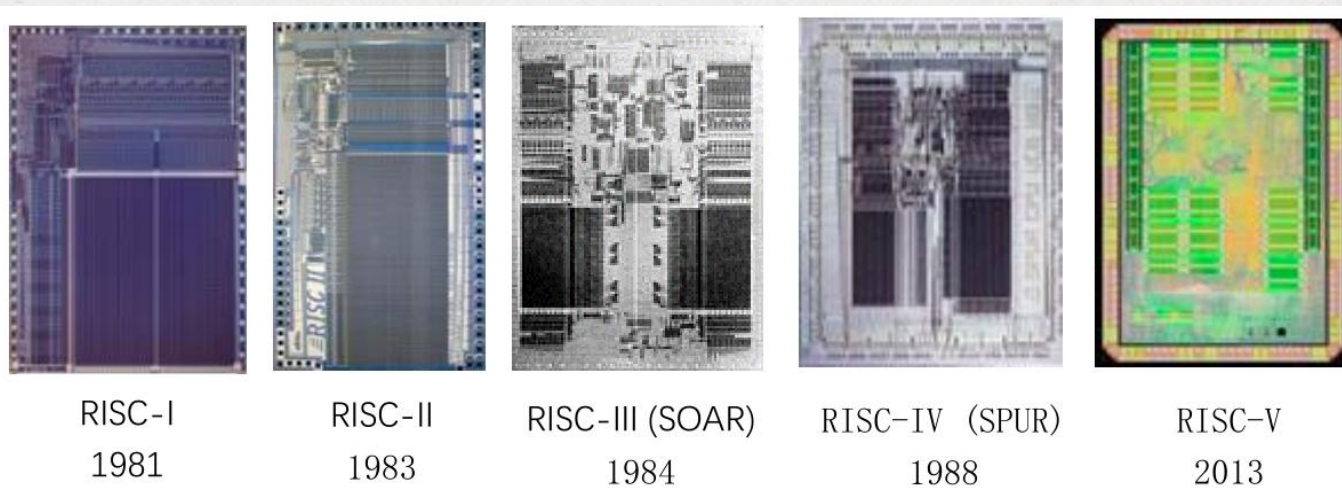


图 2. 五代 RISC 处理器

[引用自：开放指令集与开源芯片发展报告, 中国开放指令生态联盟, 2019年2月]



伯克利研究团队认为，指令集 ISA 作为软硬件接口的一种说明和描述规范，不应该像 ARM、PowerPC、X86 等指令集那样需要付费授权才能使用，而应该开放（Open）和免费（Free）。

他们选择的 BSD 开源协议给予使用者很大自由，允许使用者修改和重新发布开源代码，也允许基于开源代码开发商业软件发布和销售。因此 BSD 开源协议对商业集成很友好，很多的企业在选用开源产品时都会首选 BSD 开源协议。

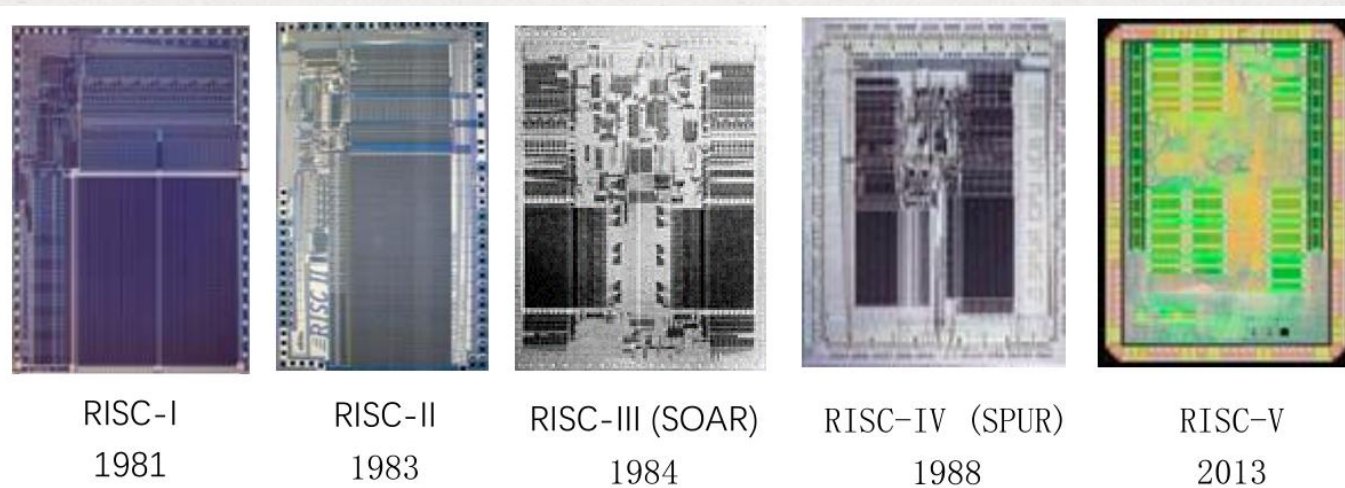



图 2. 五代 RISC 处理器

[引用自：开放指令集与开源芯片发展报告, 中国开放指令生态联盟, 2019年2月]

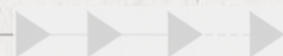






于是，一套全新的开放指令集 RISC-V 诞生了——全世界任何公司、大学、研究机构与个人都可以开发兼容 RISC-V 指令集的处理器，都可以融入到基于 RISC-V 构建的软硬件生态系统，而不需要为指令集付费。

伯克利研究团队对 RISC-V 寄予厚望，希望它能被应用到各种场合，从微控制器到超级计算机；也希望它能像 Linux 通过开源成为全世界操作系统的事实标准之一，最终成为全世界处理器指令集的事实标准，为下一个 50 年计算机系统设计与创新做出奠基性贡献。

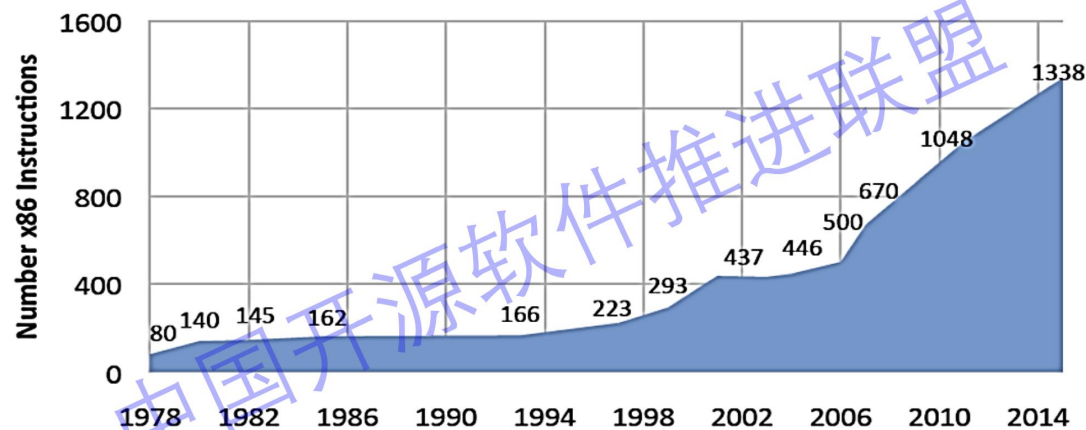
[引用自：开放指令集与开源芯片发展报告, 中国开放指令生态联盟, 2019年2月]



RISC-V 开源的特点保证了它的稳定性，因为它只属于一个开放的、非盈利性质的基金会。而商用指令集的盛衰往往与该商业公司的发展息息相关，历史上由于商业公司倒闭或者被收购而导致其商用指令集消失的例子比比皆是。

指令集设计者对以往指令集取其精华、去其糟粕后才得到了 RISC-V。

### RISC-V技术优势：丢弃历史包袱



x86指令集自诞生以来指令数量的增长  
自1978年诞生时的80条指令，到2015年增长了16倍，  
达1338条指令，且仍在增长，历史包袱沉重

[图片引用自：迎接开源芯片新潮流. 倪光南, 包云岗. 2020.07.15]



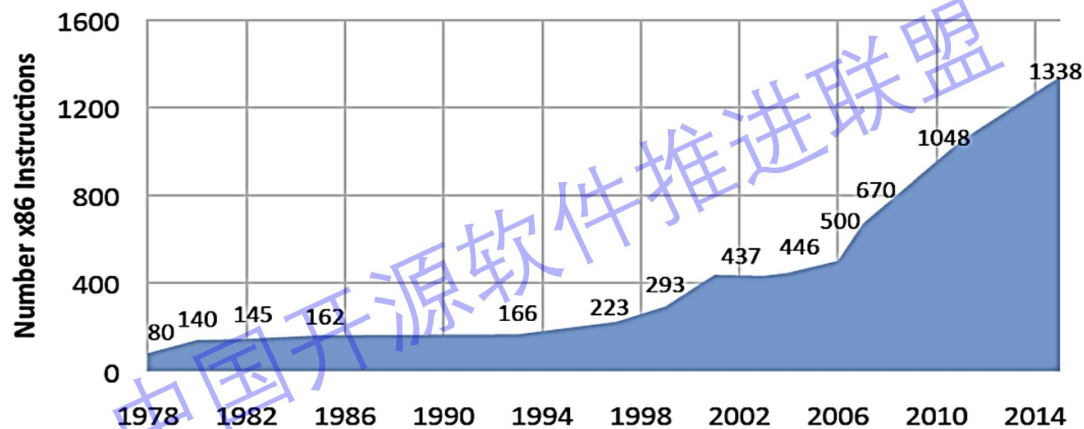
## RISC-V 最为重要的一个特点

## 模块化

传统的增量指令集架构设计要求保持向后的二进制兼容，因此新处理器必须负重前行，在实现所有历史设计的基础上实现新的设计，这使得指令集架构的复杂度随时间持续增长。

RISC-V的做法是将指令集划分为几个标准的子集，称为扩展，并保持一些基础的扩展永远不变。这一约定给编译器和操作系统相关的开发人员提供了稳定的目标。并且这些扩展是可选的，处理器的设计者可以根据需求选择实现不同的扩展，这对于嵌入式应用至关重要。

### RISC-V技术优势：丢弃历史包袱



x86指令集自诞生以来指令数量的增长  
自1978年诞生时的80条指令，到2015年增长了16倍，  
达1338条指令，且仍在增长，历史包袱沉重

[图片引用自：迎接开源芯片新潮流. 倪光南, 包云岗. 2020.07.15]

## RISC-V与x86及ARM指令集架构CPU的比较

技术与市场	x86或ARM架构	RISC-V
架构篇幅	数千页	少于三百页
指令数	指令数繁多，且不同分支不兼容	基本指令集40余条
模块化	不支持	支持
可扩展性	不支持	支持
硬件实现	复杂性高	硬件设计与编译实现非常简单
商业运作	X86封闭，ARM架构授权昂贵	开源、免费
生态环境	成熟	快速起步中
应用市场	服务器与桌面PC(x86)，移动和嵌入式(ARM)	从物联网切入，可覆盖从微控制器到超级计算机的全计算领域
应用特点	服务器、PC、移动和嵌入式应用领域居垄断地位	高性能（面积、功耗和性能）、普适（可自由扩展和裁剪）、可控（满足差异化和定制化）
应用风险	不可控、缺乏应用弹性、较高成本	生态不足、碎片化、专利问题

13

[图片引用自：迎接开源芯片新潮流. 倪光南, 包云岗. 2020.07.15]



## RISC-V与x86及ARM指令集架构CPU的比较

技术与市场	x86或ARM架构	RISC-V
架构篇幅	数千页	少于三百页
指令数	指令繁多，且不同分支不兼容	基本指令集 <b>40</b> 余条
模块化	不支持	支持
可拓展性	不支持	支持
硬件实现	复杂性高	硬件设计与编译实现非常简单
商业运作	<b>X86</b> 封闭， <b>ARM</b> 架构授权昂贵	开源，免费。
生态环境	成熟	快速起步中
应用市场	服务器与桌面 <b>PC（86）</b> ，移动和嵌入式（ <b>ARM</b> ）	从物联网切入，可覆盖从微控制器到超级计算机全计算领域。
应用特点	服务器， <b>PC</b> ，移动和嵌入式应用领域居垄断地位	高性能（面积，功耗和性能），普通（可自由扩展和裁剪），可控（满足差异化和定制化）
应用风险	不可控，缺乏应用弹性，较高成本	生态不足，碎片化，专利问题。

[图片引用自：迎接开源芯片新潮流. 倪光南, 包云岗. 2020.07.15]