

概述

教学信息

- ❖ 教师团队：白刚，李雨森，董前琨
- ❖ 教学辅助平台：云班课
 - ✧ 考勤
 - ✧ 课堂测验
 - ✧ 课后作业
 - ✧ 阅读文献
- ❖ 互动平台：飞书
- ❖ 答疑：课前和课后
- ❖ **特别提示**：不提供授课讲稿拷贝和下载，请同学们做好笔记。

课程成绩

- ❖ 课程成绩：平时成绩(70%) + 期末考试(30%)
- ❖ 平时成绩(70%)
 - ✧ 考勤(5%)
 - ✧ 课堂测验(5%)
 - ✧ 课后作业(30%)
 - ✧ 实验课程(30%)
- ❖ 期末考试(30%)
 - ✧ 基本概念
 - ✧ 基本原理
 - ✧ 单项应用能力
 - ✧ 综合能力

平时作业要求

- ❖ 独立完成，鼓励讨论，严禁抄袭。
 - ✧ 抄袭：把别人的“作品”做为自己的“成果”。
 - ✧ “抄袭者”与“被抄袭者”一律按零分计，且**不分原因**。
- ❖ 按时完成且提交，未经同意的迟交作业按零分计。
- ❖ 内容：简明扼要，逻辑清楚，重点突出，避免空洞和套话。
- ❖ 格式：DOCX 或 PDF
- ❖ 文件名称：学号(作业序号)，如：1911000(1)

❖ 自学：预习与复习

- ✧ 按部就班，循序渐进，持之以恒，不言放弃。

❖ 交流与讨论是成功的关键

- ✧ 相互交流成功与失败的经验，共同成长。

❖ 诚信 + 创新思路 + 实践

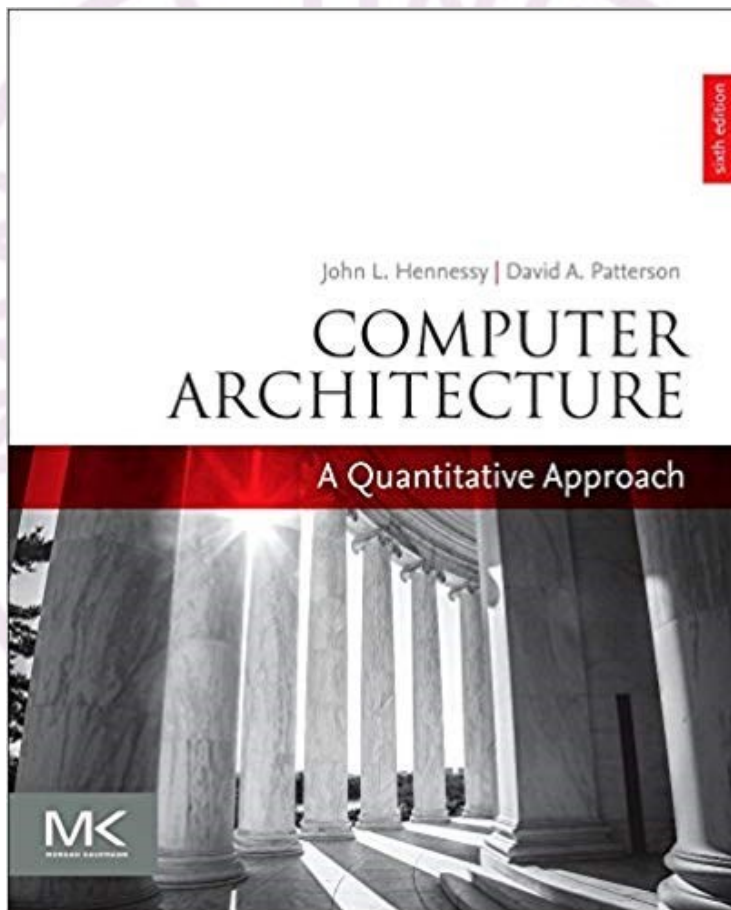
❖ 课堂纪律：

- ✧ 不迟到、早退和进食

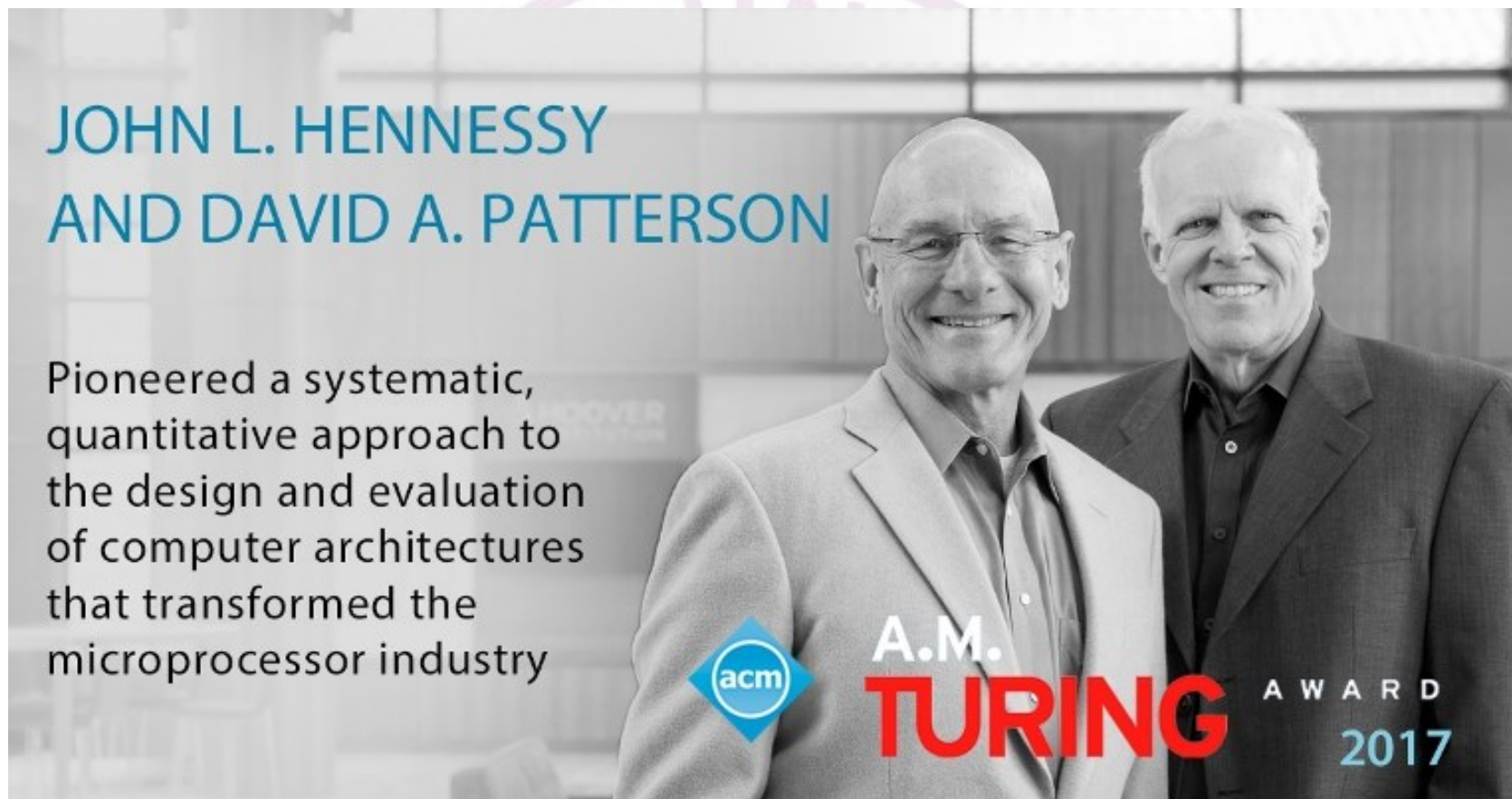
- ✧ 手机关闭或静音，不接打电话

- ✧ 不影响他人听课

- ❖ John L. Hennessy, David A. Patterson, *Computer Architecture: A Quantitative Approach (Sixth Edition)*, Elsevier Inc., 2019.



- ❖ John L. Hennessy 和 David A. Patterson 获得2017年度计算机科学最高奖项——图灵奖



设计与评估

计算机系统

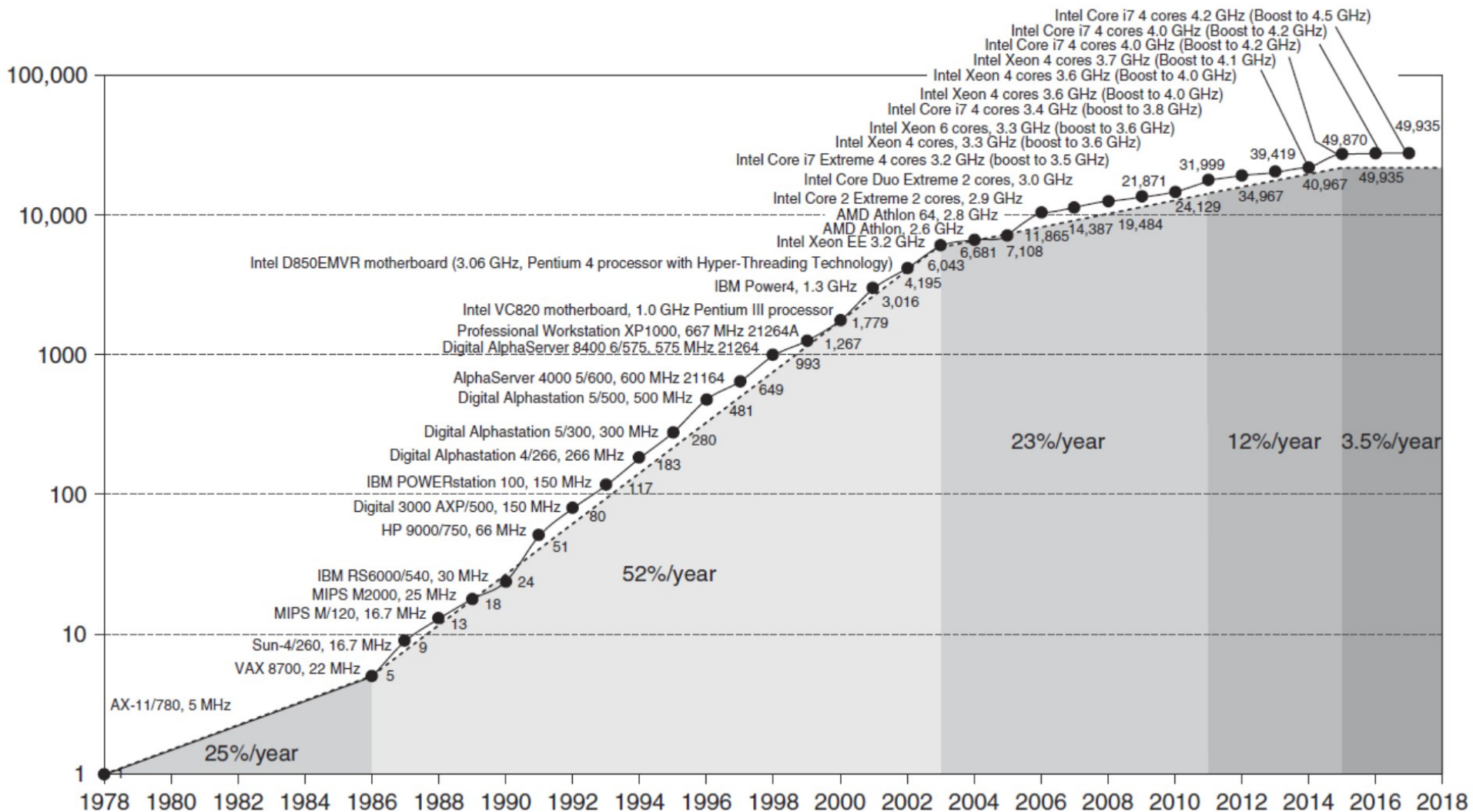
课程内容

- ❖ 量化设计和分析
- ❖ 存储层次
- ❖ 四个层次的并行性
 - ✧ 指令级并行性
 - ✧ 数据级并行性
 - ✧ 线程级并行性
 - ✧ 程序级并行性
- ❖ 特定领域（专用）计算机体系结构

The background of the slide features a large, faint, light purple watermark of the Tsinghua University seal. The seal is circular, with the words "TSINGHUA UNIVERSITY" around the top and "1911" at the bottom. In the center is a shield-like emblem with Chinese characters.

Any question?

处理器性能发展曲线



发展动力

❖ 半导体技术的进步

- ❑ 特性尺寸缩小
- ❑ 时钟速度提高

	1970-1980	1980-1990	1990-2000	2000-2010
晶体管数量	2K~100K	100K~1M	1M~100M	100M~2B
时钟频率	0.1~3MHz	3~30MHz	30MHz~1GHz	1GHz~15GHz
每周期指令数	0.1 IPC	0.1~0.9 IPC	0.9~1.9 IPC	1.9~2.9 IPC

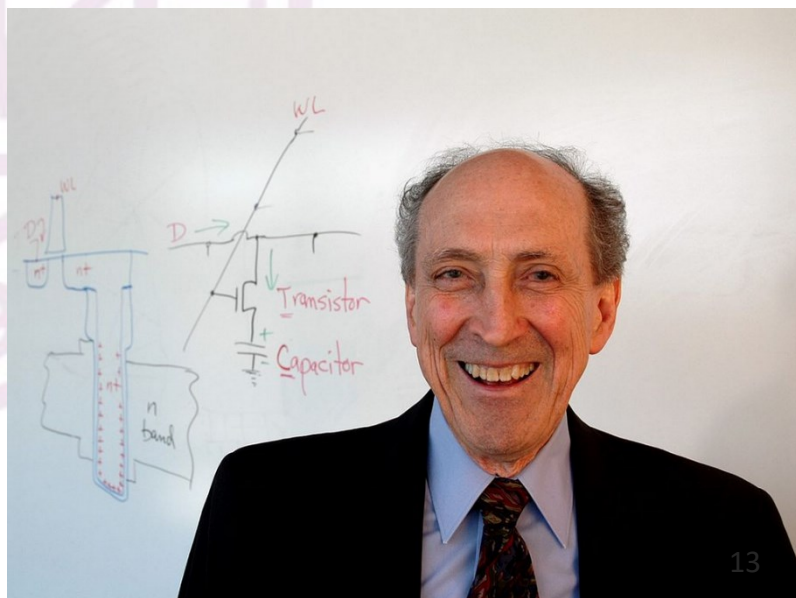
❖ 体系结构的创新

- ❑ 放弃汇编语言编程降低了对目标代码兼容性的需求
- ❑ 开发标准化且独立于系统供应商的操作系统，如UNIX和Linux，降低了推出新型体系结构的成本和风险

半导体技术的进步

❖ 登纳德缩放比例定律(Dennard Scaling):

- ❑ Robert H. Dennard 于1974年提出, 2009年以“发明和发展动态随机存储器”获得 Charles Stark Draper Prize (查尔斯 斯塔克 德拉普尔奖)
- ❑ **Dennard定律**: 随着晶体管尺度变得越来越小, 晶体管密度逐渐增加, 但它们的功率密度保持不变。
- ❑ 表明: 功率的使用与面积成比例, 电压和电流的规模与长度成比例。
- ❑ 2004年终结: 芯片尺度不变晶体管数量增加, 电流泄漏导致芯片升温。
- ❑ **Hennessy**: 谁能想到, 微处理器将不得不降低时钟速度或关闭部分内核以免被烧坏呢?
- ❑ 可能的答案: **专业化(Specialization)**



半导体技术的进步

❖ 摩尔定律(Moore's Law):

- ❑ Gordon Moore 于1965年提出,
- ❑ **Moore定律**: 芯片中晶体管数目大约每经过18个月翻一番。
- ❑ 在近两年终结
 - ✧ 2010年: Intel微处理器 1,170,000,000个晶体管
 - ✧ 2016年: Intel微处理器 1,750,000,000个晶体管 (实际)
18,720,000,000个晶体管 (Moore定律)



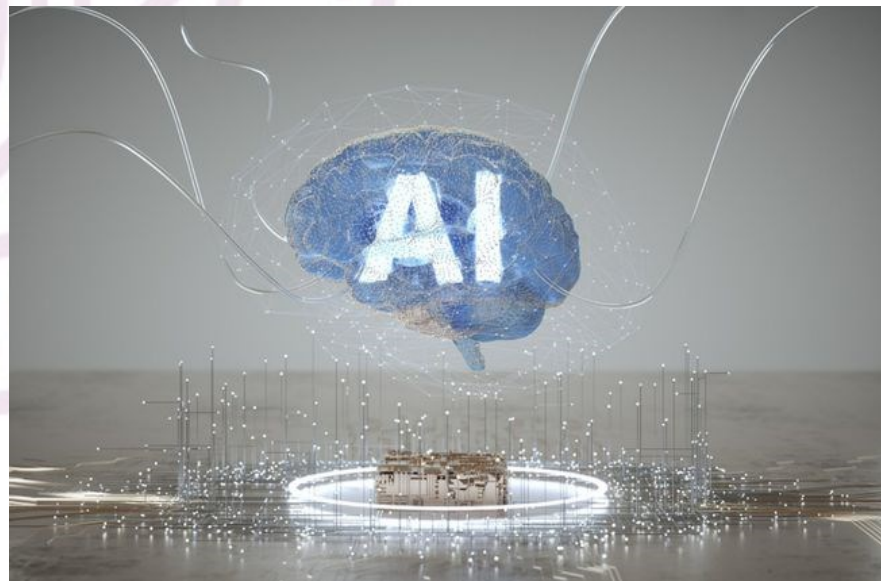
体系结构的创新

❖ 提高**功率-性能-成本**的唯一途径——**特定领域体系结构**

- ❑ 微处理器包含若干个**专用核(domain-specific core)**
- ❑ 每个核只需要很好地完成某类计算
- ❑ 整体性能明显优于**通用核(general-purpose core)**

❖ **关键因素**：足够规模的需求

- ❑ 抽取共性问题



简约指令集计算机

- ❖ 简约指令集计算机：Reduced Instruction Set Computer
- ❖ 关注点：两个关键技术
 - ✧ 开发指令级并行性
 - ✧ 早期：流水操作
 - ✧ 后期：多指令发出
 - ✧ 利用 Cache
 - ✧ 早期：简单形式
 - ✧ 后期：更加复杂的组成和优化
- ❖ RISC：具有简单指令集的计算机体系结构
 - ✧ 应对策略：跟上 或 消亡
 - ✧ Digital Equipment VAX：消亡
 - ✧ Intel x86：内部将x86指令转换为类RISC指令
 - ✧ 低端应用（移动电话）：以 ARM 架构为主流
 - ✧ ARM 芯片 2015年销售 148 亿片，大约是 x86 处理器芯片销量的 50 倍

RISC 带来的影响

- ❖ 显著地提高了计算机系统的能力
 - ✧ 高端微处理器超过20年前超级计算机的计算能力
- ❖ 性价比的显著提升带来了新型计算机系统
 - ✧ 20世纪80年代个人计算机和工作站的出现
 - ✧ 近十年智能手机和平板电脑代替个人电脑成为众人的首选计算平台
- ❖ 半导体制造业的进步使得微处理器计算机占据主导地位
 - ✧ 小型计算机被服务器取代
 - ✧ 大型机和高性能超级计算机均是微处理器的集合
- ❖ 硬件复兴直接对软件开发造成冲击
 - ✧ 到2003年，高性能微处理器的性能是单纯依靠技术进步所获得性能的7.5倍。
 - ✧ 高性能微处理器的性能：技术进步和体系结构创新的合力
 - ✧ 技术进步：半导体技术进步，线路设计改进

RISC 带来的影响

- ✧ 自1978年以来，5万倍的性能提升让程序员可以以性能换取生产率
 - ✧ 面向性能的编程语言（如 C 和C++）被托管编程语言（如 Java和 Scale）和脚本语言（如 JavaScript和Python）所取代。
 - ✧ 利用实时编译器和基于跟踪编译的解释器(interpreter)正在取代过去传统编译器和链接器。
 - ✧ 在软件部署方面，通过Internet的 SaaS (Software as a Service)取代了必须在本地计算机上安装和运行的客户端商业软件。

计算机系统的分类

❖ 分类标准

- ❑ 应用领域
- ❑ 性能需求
- ❑ 计算技术

❖ 计算机系统的五种类型

- ❑ 物联网 / 嵌入式计算机(Internet of Thing / Embedded Computers)
- ❑ 个人移动设备(Personal Mobile Device)
- ❑ 桌面计算系统(Desktop Computing)
- ❑ 服务器(Servers)
- ❑ 集群 / 仓储计算机(Clusters / Warehouse-Scale Computers)

物联网/嵌入式计算机

❖ 价格

- ❑ 系统价格: \$10~\$100,000
- ❑ 微处理器价格: \$0.01~\$100

❖ 关键问题

- ❑ 价格: 以最低价格满足对系统的性能需求
- ❑ 能耗
- ❑ 特定应用性能

❖ 物联网(IoT)

- ❑ 传感器: 收集有用数据
- ❑ 执行器: 与物理世界交互
- ❑ 产生各种“智能”应用, 如智能家居、智能城市 and 智能汽车等

❖ 没有量化设计和评估方法

个人移动设备

❖ 价格

- ❑ 系统价格: \$100~\$1000
- ❑ 微处理器价格: \$10~\$100

❖ 关键问题

- ❑ 成本
- ❑ 能耗 (电池能量, 散热, 存储访问)
- ❑ 媒体性能
- ❑ 响应能力 (实时性能)

❖ 带有多媒体用户接口的无线设备

- ❑ 移动电话
- ❑ 平板电脑

❖ 可以运行外部软件, 具有桌面计算机的许多特性

- ❑ 是否能够运行第三方软件是鉴别嵌入式计算机的标准

桌面计算系统

❖ 价格:

- ❑ 系统价格: \$300~\$2500
- ❑ 微处理器价格: \$50~\$500

❖ 关键问题

- ❑ 性能价格比: 最新、最高性能微处理器和低成本微处理器均往往首先出现在桌面计算系统中
- ❑ 能耗
- ❑ 图形能力

❖ 最大市场

- ❑ 低端网络本~高端工作站
- ❑ 每年超过一半产量为笔记本电脑 (自2008年)

服务器

❖ 价格:

- ❑ 系统价格: \$5,000~\$10,000,000
- ❑ 微处理器价格: \$200~\$2000

❖ 关键问题

- ❑ 吞吐量: 每分钟处理的交易数, 每秒处理的网页数
- ❑ 可用性: 7×24 服务, 宕机损失至少每分钟 \$100,000 (根据行业不同)
- ❑ 可扩展性: 计算能力, 内存, 存储系统, I/O带宽
- ❑ 能耗

❖ 提供更大规模、更可靠的文件和计算服务

- ❑ 替代传统大型机, 成为大规模企业计算的主干设备

集群/仓储计算机

❖ 价格:

- ❑ 系统价格: \$100,000~\$200,000,000
- ❑ 微处理器价格: \$50~\$250

❖ 关键问题

- ❑ 性能价格比
- ❑ 吞吐量
- ❑ 能耗
- ❑ 均衡性

❖ 集群计算机

- ❑ 使用局域网连接一组个人计算机或服务器来做为一台大型计算机
- ❑ 由 SaaS 应用的增长所带来的一种新型计算机类型

❖ 仓储计算机

- ❑ 最大的集群计算机, 将成千上万台服务器做为一个整体。

集群/仓储计算机

❖ WSC 与服务器

- ❑ 关键是**可用性**，系统失效的代价巨大。
- ❑ WSC 普遍使用冗余配置来提高可用性

❖ WSC 与超级计算机

- ❑ 超级计算机：主要用于浮点数计算，批处理程序每次可能运行一周
- ❑ WSC：交互式应用，大规模存储，可靠性，高Internet带宽

计算并行性：应用并行性

- ❖ 数据级并行性(Data-Level Parallelism, DLP)
 - ✧ 同时操作许多数据项
- ❖ 任务级并行性(Task-Level Parallelism, TLP)
 - ✧ 独立创建任务，基本上并行运行。

计算并行性：体系结构并行性

- ❖ 指令级并行性(Instruction-Level Parallelism, ILP)
 - ✧ 借助编译器在适当级别上使用流水思想利用数据级并行性
 - ✧ 在中等级别上使用推测执行思想
- ❖ 矢量体系结构 / 图形处理单元(GPUs) / 多媒体指令集
 - ✧ 利用数据级并行性，使用单条指令并行处理一组数据
- ❖ 线程级并行性(Thread-Level Parallelism, TLP)
 - ✧ 在紧耦合硬件模型（允许并行线程之间的交互）中利用数据级并行性和任务级并行性
- ❖ 请求级并行性(Request-Level Parallelism, RLP)
 - ✧ 利用程序员或操作系统指定的解耦任务之间的并行性

计算机系统的Flynn分类

- ❖ SISD: 单指令流 – 单数据流
- ❖ SIMD: 单指令流 – 多数据流
- ❖ MISD: 多指令流 – 单数据流
- ❖ MIMD: 多指令流 – 多数据流

单指令流 – 单数据流

- ❖ 单处理器
- ❖ 标准顺序执行计算机（程序员视角）
- ❖ 指令级并行性
 - ✧ 超标量处理器
 - ✧ 推测执行



单指令流 – 多数据流

❖ 多处理器

- ❑ 每个处理器具有独立的数据存储器
- ❑ 单个指令存储器和控制处理器

❖ 数据级并行性

- ❑ 矢量体系结构
- ❑ 多媒体扩展指令集
- ❑ 图形处理单元



多指令流 – 单数据流

- ❖ 无商业化实现的多处理器



多指令流 – 多数据流

❖ 任务级并行性

- ✧ 每个处理器执行自己的指令、处理自己的数据

❖ 线程级并行性

- ✧ 紧耦合 MIMD 体系结构

❖ 请求级并行性

- ✧ 松耦合 MIMD 体系结构
- ✧ 集群和仓储计算机

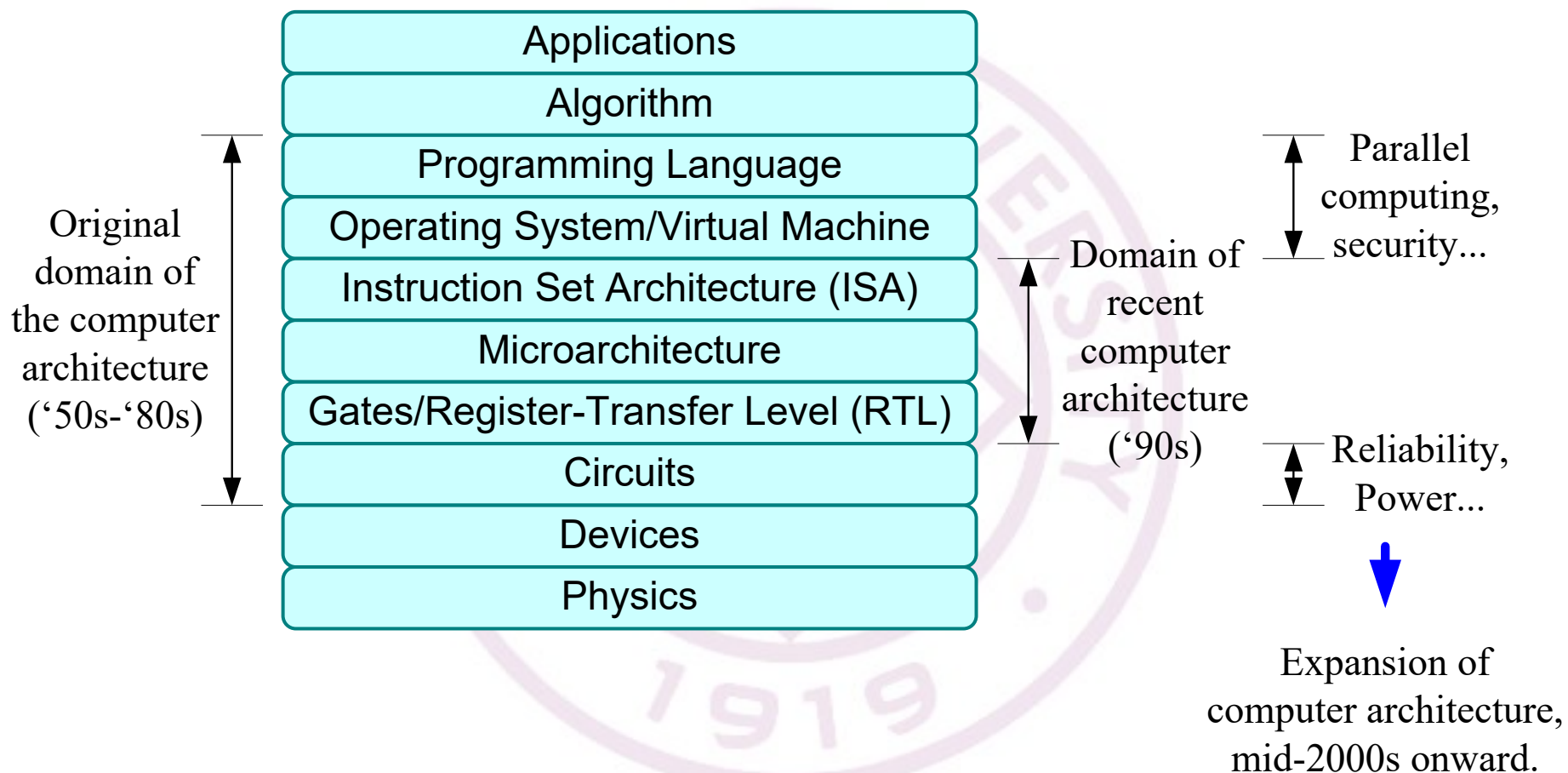
计算机体系结构的早期定义

- ❖ Amdahl (1964): 系统程序员所看到的系统属性。
 - ❑ 概念结构
 - ❑ 功能行为
 - ❑ 不同于数据流组织、控制逻辑实现和物理实现。
- ❖ 计算机体系结构
 - ❑ 指令集体系结构 +
 - ❑ 计算机组成（流水，存储层次，存储系统等）
- ❖ 计算机体系结构是计算机硬件和软件之间的“粘合剂”
- ❖ 计算机体系结构将确定计算机系统中软硬件之间的功能分配

计算机体系结构定义的演化

- ❖ 20 世纪 50 ~ 60 年代：计算机算术
- ❖ 20 世纪 70 年代：指令集设计
- ❖ 20 世纪 90 年代：CPU设计，存储器系统，I/O 系统，多处理器，指令集并行性
- ❖ 21 世纪：多核体系结构

计算机体系结构定义的演化



计算机体系结构定义的动态和静态界面

- ❖ **定义**：由指令集体系结构(ISA)决定
- ❖ **作用**：将编译时静态执行的操作与运行时动态执行的操作分开



计算机体系结构的现代定义

- ❖ 新型计算机系统的指定需求
- ❖ 在满足成本、功耗和可用性约束条件下，使性能和能效最大化的设计
 - ✧ 指令集设计
 - ✧ 功能组织
 - ✧ 逻辑设计
 - ✧ 实施（集成电路设计，包装，电源，冷却等）

再谈计算机体系结构定义

- ❖ 定义：设计符合目标和功能需求的**组成**和**硬件**
- ❖ **组成(organization)**：计算机系统高层设计，也称为**微结构**
 - ✧ 内容：存储系统，存储器内联，内部处理器设计等
 - ✧ 两个处理器可以有相同的指令集体系结构但不同的组成
 - ✧ 多核(multicore)：内置多个处理器的微处理器
- ❖ **硬件(hardware)**：计算机系统的细节
 - ✧ 内容：详细逻辑设计，计算机包装技术
 - ✧ 计算机系统可以有相同的指令集体系结构和非常相似的组成，但在硬件实现细节上是不同的。

本书对计算机体系结构的定义

- ❖ 指令集体系结构(Instruction Set Architecture)
- ❖ 组成(Organization) 或 微结构(Microarchitecture)
- ❖ 硬件(Hardware)



计算机体系结构的发展趋势

- ❖ 指令级并行性(Instruction-Level Parallelism, ILP)
 - ✧ 单处理器性能提升于 2003 年终结
- ❖ 性能提升的新途径
 - ✧ 数据级并行性(Data-Level Parallelism, DLP)
 - ✧ 线程级并行性(Thread-Level Parallelism, TLP)
 - ✧ 仓储计算机(Warehouse-Scale Computer, WSC)
 - ✧ 请求级并行性(Request-Level Parallelism, RLP)
- ❖ 显式并行性需要重新编写应用程序
 - ✧ 隐式并行性（透明的）：ILP
 - ✧ 由编译器和硬件完成，不需要程序员关注；
 - ✧ 显式并行性（不透明的）：DLP, TLP 和 RLP

计算机架构师(Computer Architect)

❖ 设计一台计算机必须满足

- ❑ 功能性需求
- ❑ 价格, 功耗, 性能和可行性目标

❖ 了解最重要的功能需求

- ❑ 应用领域: 五种计算机系统类型
- ❑ 软件兼容性: 确定现成软件的数量
 - ✧ 编程语言兼容, 目标代码或二进制兼容
- ❑ 操作系统需求: 支持所选操作系统的必要功能
 - ✧ 地址空间, 存储器管理, 存储保护
- ❑ 标准: 市场需要一定的标准
 - ✧ 浮点数, I/O接口, 操作系统, 网络, 编程语言

计算机架构师(Computer Architect)

- ❖ 关注相关技术和应用两个方面的发展趋势
 - ✧ 它们将影响未来的成本
 - ✧ 它们将影响架构的寿命



五种关键实现技术

❖ 集成电路逻辑技术（摩尔定律）

- ❑ 晶体管密度：每年大约增加 35%
- ❑ 晶片尺度：每年增加 10% ~ 20%
- ❑ 片上晶体管数目：每年增加 40% ~ 55%，或每 18 个月翻一番
- ❑ 目前片上器件数目仍在增长，但加速度为负值。

❖ 半导体 DRAM（动态随机访问存储器）

- ❑ 容量从每三年翻两番降速到每年增加 25% ~ 40%
- ❑ 销售产品
 - ✧ 2014 年开始销售 8Gb 产品
 - ✧ 2019 年开始销售 16Gb 产品
 - ✧ ? 年销售 32Gb 产品（可能永远不会发生）

五种关键实现技术

❖ 半导体闪存（电可擦写可编程只读存储器）

- ❑ PMD 上的标准存储设备
- ❑ 片容量每年大约增加 50% ~ 60%
- ❑ 每比特售价比 DRAM 便宜 8 ~ 10 倍

❖ 磁盘技术

- ❑ 密度增加明显
 - ✧ 1990年之前，每年大约增加 30%
 - ✧ 1990年之后，每年大约增加 60%，到 1996 年达到 100%
 - ✧ 2004年 ~ 2011年之间，下降到每年大约增加 40%
 - ✧ 当前，每年大约增加 < 5%
- ❑ 密度增加几率下降，盘片数目可能从 7 片增加到 9 片（空间上限）
- ❑ 最有希望的改进是读写头工作的 **HAMR（热辅助磁性写入）技术**

五种关键实现技术

- ✧ 相比闪存：每比特便宜 8 ~ 10倍；相比 DRAM：每比特便宜 200 ~ 300倍
- ✧ 未来发展趋势：机械硬盘主攻高容量，SSD主攻高性能。

❖ 网络技术

- ✧ 网络性能依赖于交换机和传输系统的性能

附录



性能趋势：带宽与延时

❖ 术语

- ❑ 带宽(bandwidth)或吞吐量(throughput): 给定时间内完成工作的总量
- ❑ 延时(latency)或响应时间(response time): 某事件从开始到完成之间的时间

❖ 处理器

- ❑ 带宽提高32,000~40,000倍
- ❑ 延时缩短50~90倍

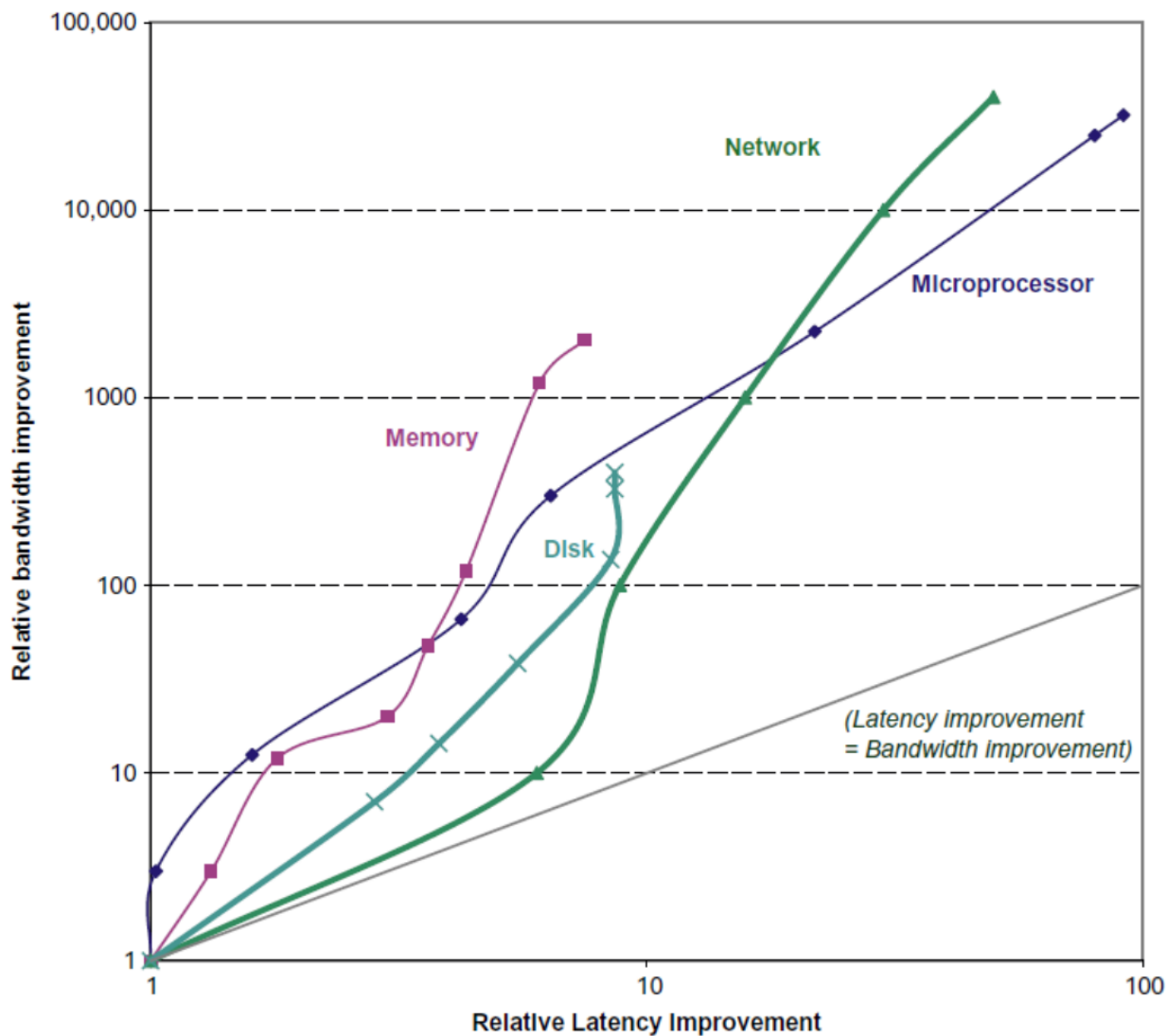
❖ 存储器和磁盘

- ❑ 带宽提高300~1200倍
- ❑ 延时缩短6~8倍

❖ 经验法则：带宽增长至少是延时增长的平方

性能趋势：带宽与延迟

❖ 25~40年间微处理器、存储器、网络 and 磁盘的性能关键点



晶体管性能与连线

❖ 术语

- ❑ 特性尺寸(feature size): 晶体管或连线在 x 或 y 方向上的最小尺寸

❖ 特性尺寸

- ❑ 描述集成电路的特性
- ❑ 1971年10微米, 2017年0.016微米 (16纳米), 7纳米正在路上
- ❑ 晶体管密度随着特性尺寸线性减少而平方增加
 - ✧ 早期: 4位, 8位, 16位, 32位, 直到64位微处理器
 - ✧ 近几年: 片上多处理器, 更宽的SIMD单元, 推测执行和Cache
- ❑ 晶体管性能改进与特性尺寸减少呈线性关系
 - ✧ 与集成电路中的连线无关

集成电路中功率和能量的趋势

❖ 能量(energy)是计算机设计者面临的最大挑战

❖ 功率

✧ 最大功率：确保正确操作的功率

✧ 持续能量：热设计功率(thermal design power, TDP)

✧ 峰值功率是热设计功率的1.5倍，高于平均功率

✧ 是冷却系统设计的匹配目标

❖ 晶体管

✧ 动态能量($0 \rightarrow 1$ 或 $1 \rightarrow 0$):

$$Energy_{dynamic} \propto \frac{1}{2} \times Capacitive\ load \times Voltage^2$$

✧ 动态功率

$$P_{dynamic} \propto \frac{1}{2} \times Capacitive\ load \times Voltage^2 \times Frequency\ switched$$

集成电路中功率和能量的趋势

❖ 途径

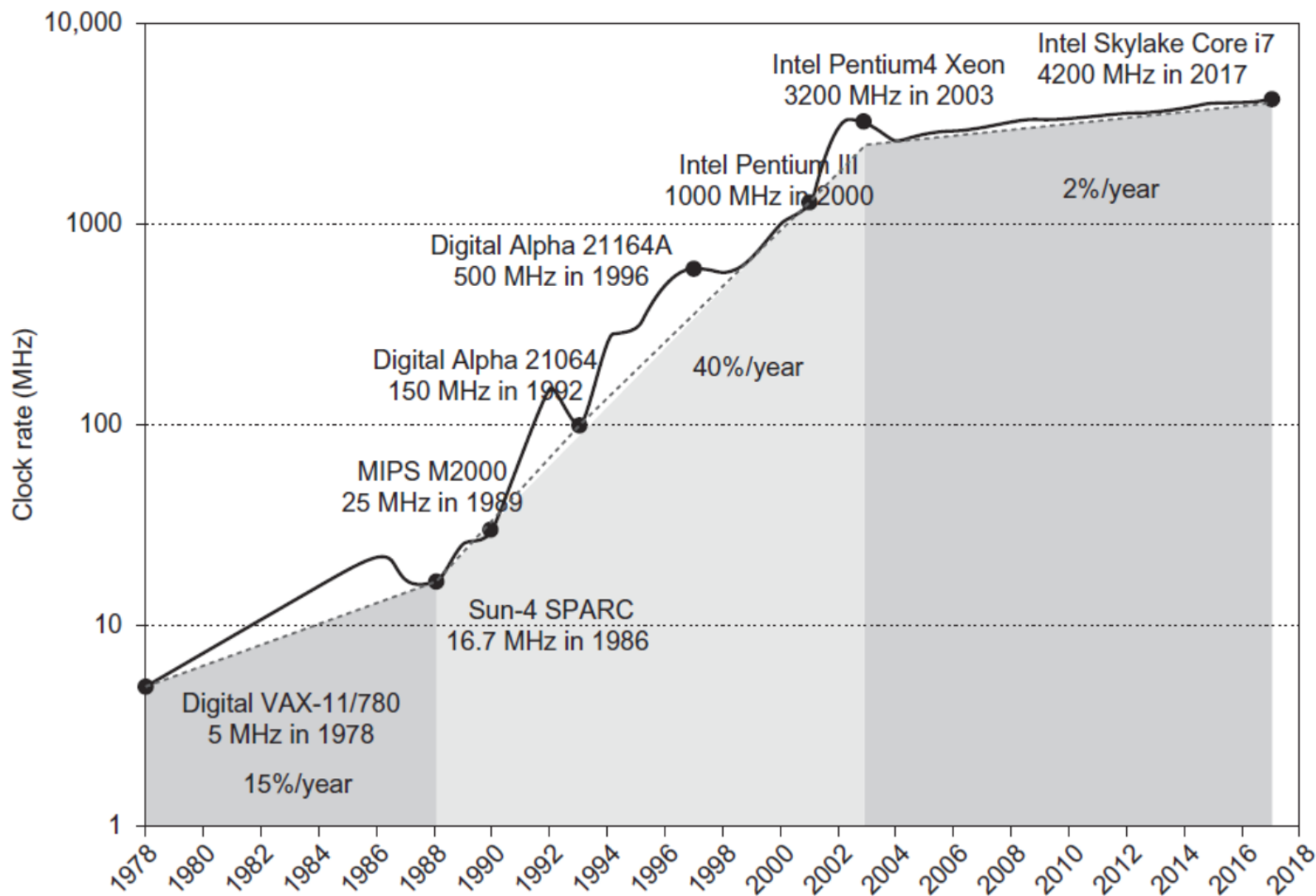
- ❑ 降低时钟频率只能减少**功率**，而不能降低**能量**
- ❑ 降低电压可以大幅度减少动态功率和动态能量，故 20 年来电压从 5V 降低到 1V。

❖ 芯片功率

- ❑ Intel 80386功率是2W
- ❑ 4.0GHz Intel Core i7-6700K功率是95W
- ❑ 1.5 平方厘米是空气冷却的极限

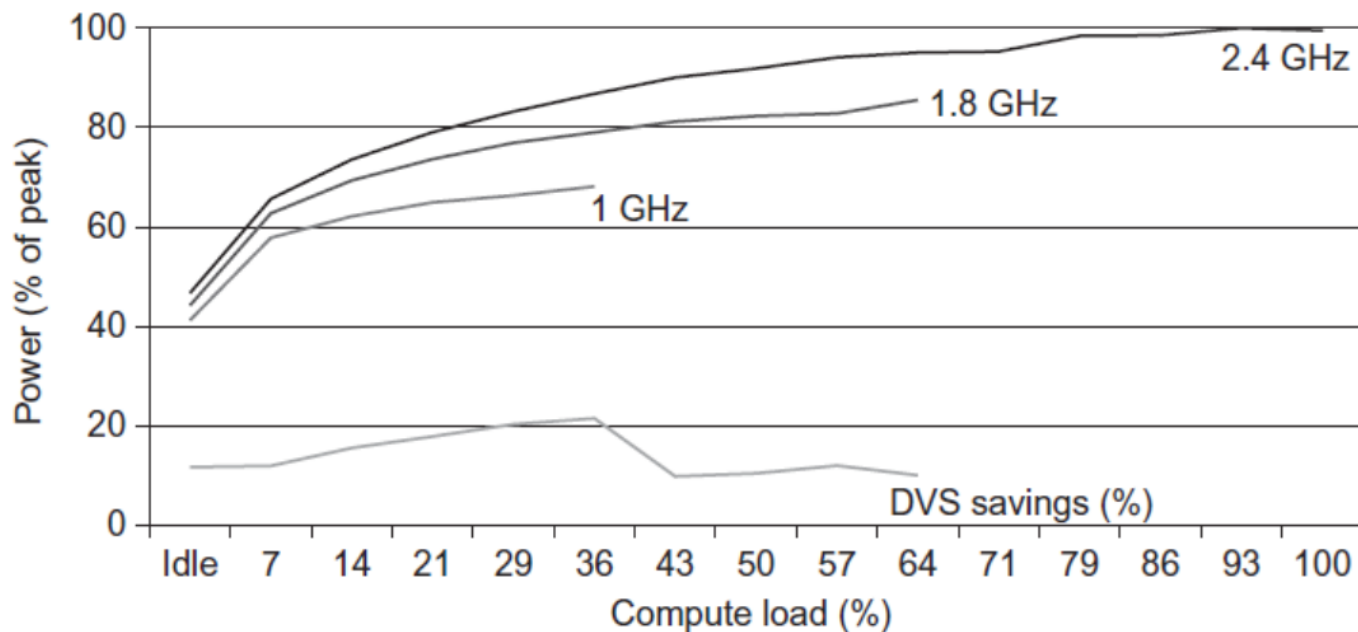
集成电路中功率和能量的趋势

❖ 如果不能降低电压或增加功率，那么时钟频率增长就会减慢。



提高能源效率的技术

- ❖ 关闭非活动模块的时钟
- ❖ 动态电压-频率比例(Dynamic Voltage-Frequency Scaling)



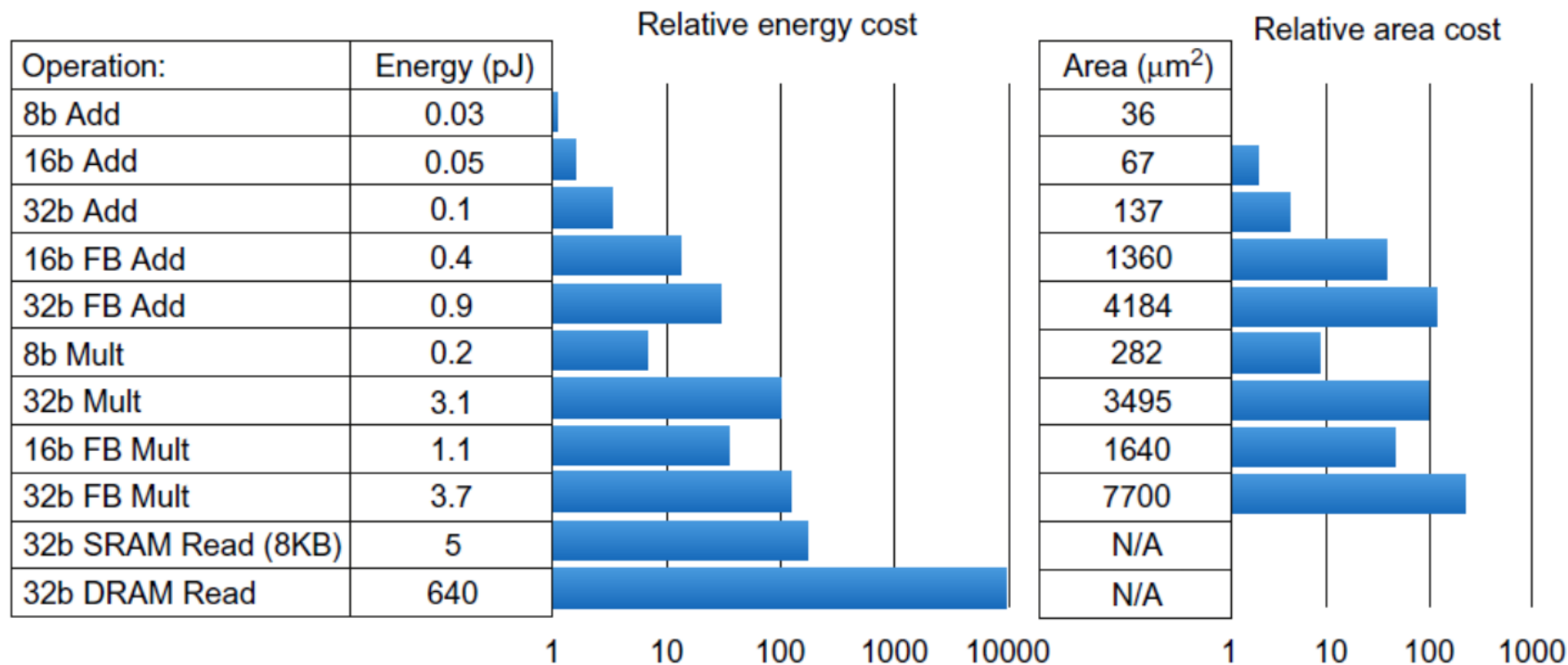
- ❖ DRAM和磁盘的低功率模式
- ❖ 超频; 对单线程代码关闭其它处理器核

静态功率

❖ 静态功率正比于晶体管数目，占总功率的 25% ~ 50%

✧ 计算公式: $Power_{static} \propto Current_{static} \times Voltage$

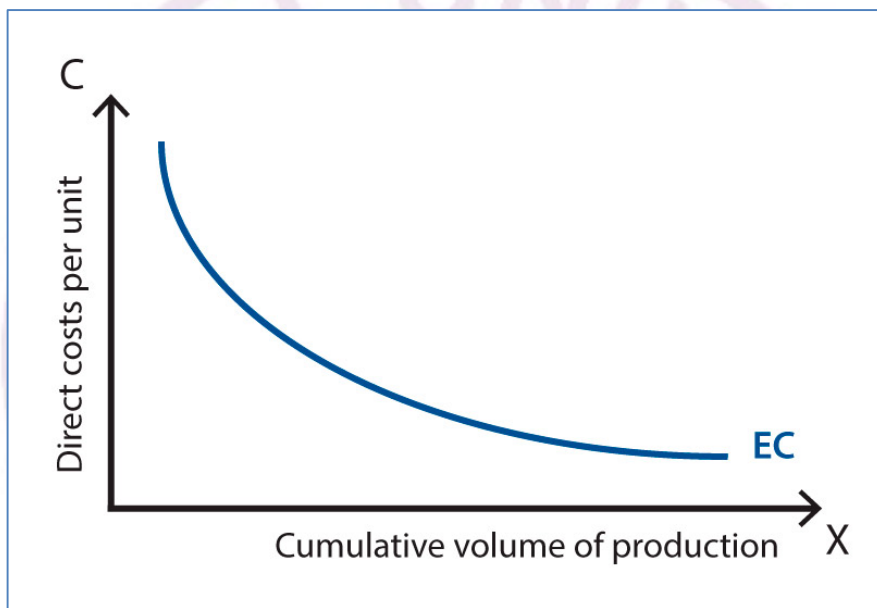
✧ 关闭非活动模块电源（随着晶体管尺度减少，漏电流增加）



成本趋势

❖ 成本降低遵循学习曲线

- ❑ 学习曲线：在一定时间内获得技能或知识的速率
- ❑ 生产计算机部件成本随着时间（产量）降低



❖ DRAM：价格紧跟成本

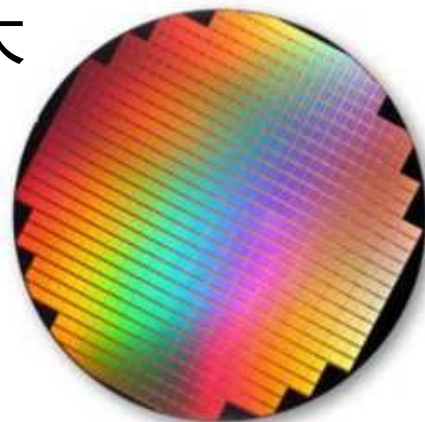
❖ 微处理器：价格取决于产量

- ❑ 经验法则：产量每增加一倍价格减少10%

集成电路成本

❖ 集成电路成本占计算机系统成本中的比例很大

- ❑ 测试晶圆(wafer)
- ❑ 切割成晶片(die)
- ❑ 封装晶片



❖ 集成电路成本

- ❑ 集成电路成本: $\frac{cost_{die} + cost_{testing\ die} + cost_{packaging\ and\ final\ test}}{yield_{final\ test}}$

- ❑ 晶片成本: $cost_{die} = \frac{cost_{wafer}}{number_{die\ per\ wafer} \times yield_{die}}$

$$number_{die\ per\ wafer} = \frac{\pi \times (diameter_{wafer}/2)^2}{area_{die}} - \frac{\pi \times diameter_{wafer}}{\sqrt{2 \times area_{die}}}$$

集成电路成本

❖ 玻色-爱因斯坦(Bose-Einstein)公式

$$yield_{die} = \frac{yield_{wafer}}{(1+defect \times area_{die})^N}$$

- ✧ 单位面积缺陷(*defect*): 0.016 ~ 0.057 (2010年)
- ✧ 过程复杂性因子(*N*): 11.5 ~ 15.5 (2010年, 40nm)

❖ 计算机设计者

- ✧ 改变晶片尺寸, 进而影响成本
- ✧ 二者既取决于晶片上包含或排除的功能, 也取决于I/O管脚的数量

❖ 模块可靠性

✧ 平均无故障时间(mean time to failure, MTTF)

✧ 设备预期能够持续正常工作的平均时间

✧ 失效时间(failures in time, FIT)是MTTF的倒数, 使用每10亿(10^9)小时的失效次数来表示。

✧ 平均修复时间(mean time to repair, MTTR)

✧ 修复设备使其能够正常工作所需要的平均时间

✧ 平均故障间隔时间(mean time between failures, MTBF)

✧ 设备在两次相邻故障期间正常工作的平均时间

✧ $MTBF = MTTF + MTTR$

❖ 模块可用性

$$Module\ availability = \frac{MTTF}{MTTF + MTTR}$$

❖ 可靠性(reliability)和可用性(availability)是量化指标。

诚信 创新 实践

