计算机体系结构

周学海
xhzhou@ustc.edu.cn
0551-63606864
中国科学技术大学

Review: Computer Architecture

- **计算机体系结构：**
  - 设计、选择和连接硬件组件，设计硬件/软件接口，以创建一个满足功能、性能、能耗、成本和其他具体目标的计算系统的科学和艺术。
- **目的：实现一组设计目标**
  - E.g., 在工作负载X, Y, Z上的最高性能
  - E.g., 最长的电池寿命，可装进口袋，成本< X
  - E.g., 在所有已知工作负载中以最佳性能/成本比获得最佳平均性能…
  - 设计一台超级计算机与设计一部智能手机设计目标是不同的，但是，许多基本原则是相似的

Review: 计算机体系结构研究范畴

早期的定义：Instruction-Set Architecture

程序员可见的计算系统的属性。包括：概念性的结构和功能行为。不包括：数据流和控制流的组织、逻辑设计以及物理实现。

– Amdahl, Blaauw, and Brooks, 1964

狭隘的观点（narrow view）：

包括：软件设计者与硬件设备设计者（VLSI）之间的中间层（ISA），以及微体系结构

扩展的观点（expanded view）：

包括：算法层、程序/语言层、系统软件层、ISA层、微体系结构、逻辑电路层、以及器件层

投票 最多可选1项

设置

- 将工程师或计算机科学家与门外汉区分开来的特征之一是管理复杂性的系统方法。
- 现代数字系统是由数百万甚至数十亿个晶体管组成的。没有人能够通过写出描述每个晶体管中电子运动的方程并同时解出所有方程来理解这些系统。
- 管理复杂性的方法：

  抽象 (Abstraction)+ 原则 (Discipline)+ 3'Y

  你知道解决复杂性问题的3'Y原则吗?

  A 知道

  B 不知道

提交

Chapter1 量化设计与分析基础

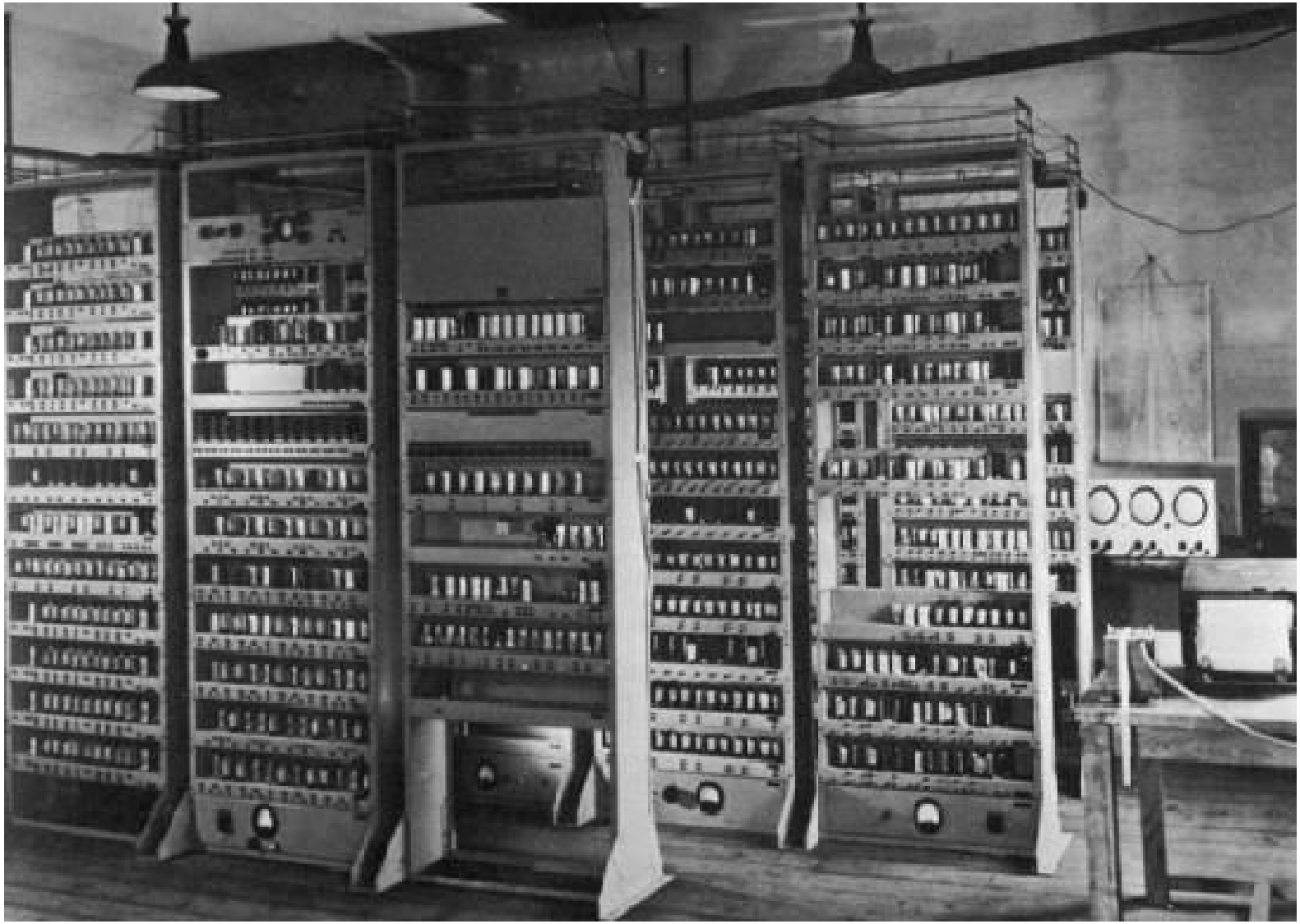
- **1.1 引言**
  - 计算机体系结构的定义
- **1.2 体系结构发展历史、现状及趋势**
  - 现代计算机系统发展趋势
- **1.3 定量分析基础**

1.2 体系结构的发展历史、现状及趋势



历史 现状 趋势

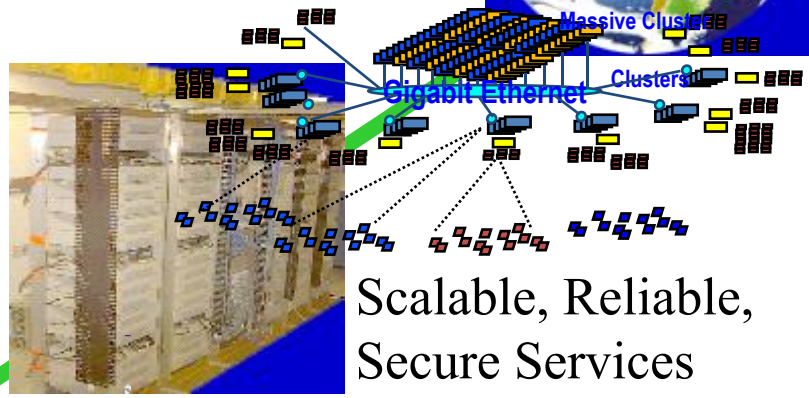
Computing Devices Then…



EDSAC, University of Cambridge, UK, 1949

Computing Systems Today

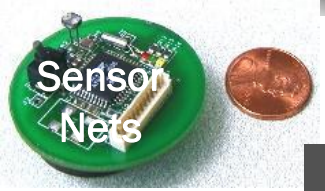
- The world is a large parallel system
  - Microprocessors in everything
  - Vast infrastructure behind them

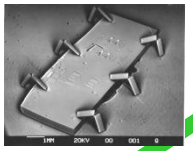


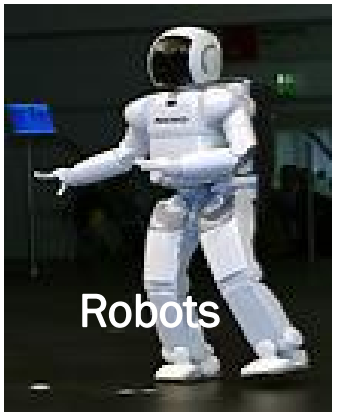
Internet Connectivity

Scalable, Reliable, Secure Services

Databases
Information Collection
Remote Storage
Online Games
Commerce



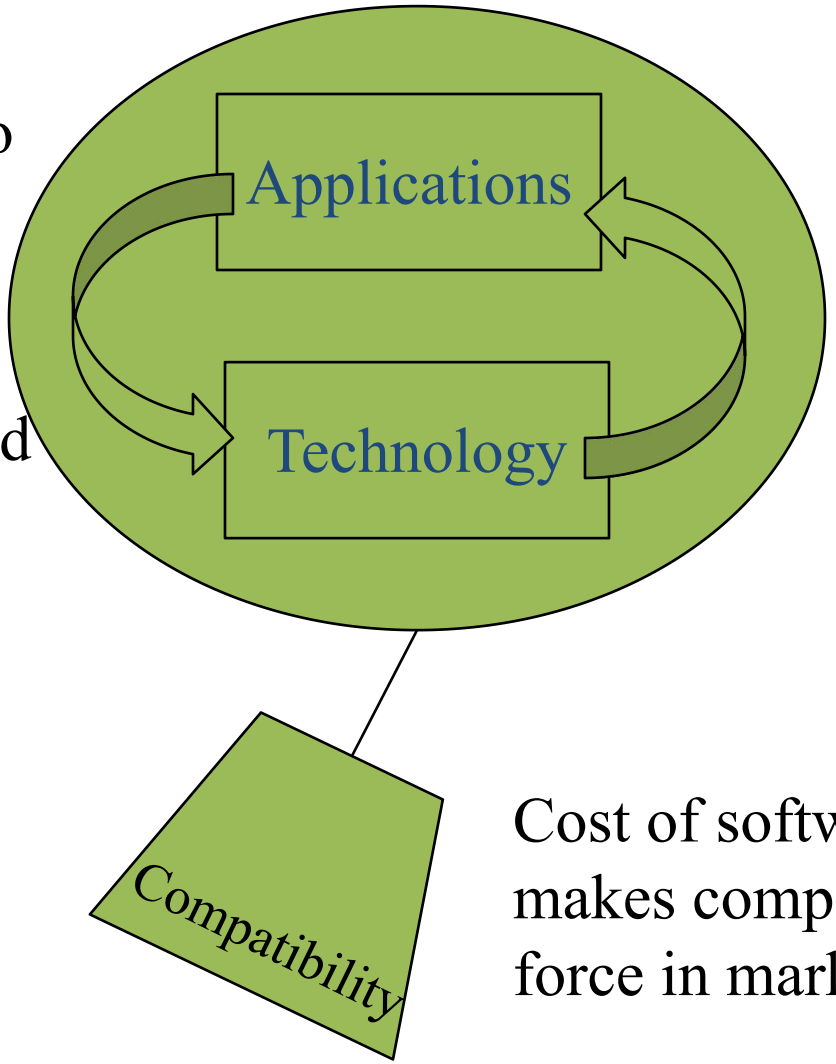




MEMS for Sensor Nets

体系结构发展的驱动力

Applications suggest how to improve technology, provide revenue to fund development



Improved technologies make new applications possible

Cost of software development makes compatibility a major force in market

体系结构的发展历史、现状及趋势*

- **体系结构发展历史**
  - Mainframes,
  - Minicomputers,
  - Microprocessors
  - RISC vs CISC，VLIW
- **体系结构现状及挑战**
  - Denard Scaling 及 Moore’s Law 的终结
  - Security
- **体系结构发展机遇**
  - Open Architectures
  - Domain Specific Languages and Architecture
  - Agile Hardware Development

*A New Golden Age for Computer Architecture: Domain-Specific Hardware/Software Co-Design, Enhanced Security, Open Instruction Sets, and Agile Chip Development

John Hennessy and David Patterson
June 4, 2018
https://www.youtube.com/watch?v=3LVeEjsn8Ts

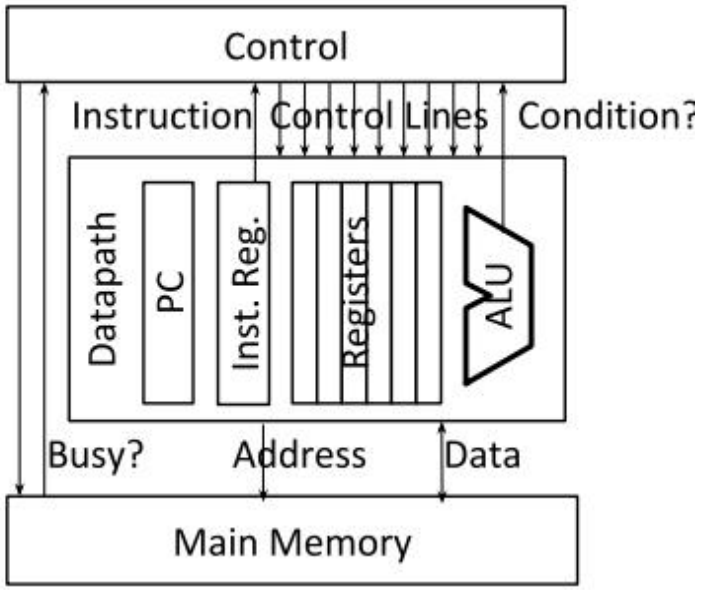
20世纪60年代初IBM机器兼容性问题

- **20世纪60年代初，IBM有4条产品线**
  - 701 ->7094
  - 650 ->7074
  - 702 -> 7080
  - 1401->7010
- **每个系统包含各自不同的**
  - ISA
  - I/O 系统及二级存储：磁带、磁鼓和磁盘
  - 汇编器、编译器、库等
  - 市场定位：商业应用、科学计算等

系列机：IBM System/360 – one ISA to rule them all

数据通路vs控制

- 处理器设计可划分为两部分
  - Datapath： 存储和操作数据
  - Control：产生控制信号作用于数据通路
- 过去对处理器设计师的最大的挑战是产生正确的控制序列



- **Maurice Wilkes 发明了微程序设计方法来设计控制部件* (1953)**
  - Logic expensive vs. ROM or RAM
  - ROM cheaper than RAM
  - ROM much faster than RAM

* "Micro-programming and the design of the control circuits in an electronic digital computer," M. Wilkes, and J. Stringer. *Mathematical Proc. of the Cambridge Philosophical Society,* Vol. 49, 1953.

IBM 360系列机的控制部件

Model	M30	M40	M50	M65
Datapath width	8 bits	16 bits	32 bits	64 bits
Microcode size	4k x 50	4k x 52	2.75k x 85	2.75k x 87
Clock cycle time (ROM)	750 ns	625 ns	500 ns	200 ns
Main memory cycle time	1500 ns	2500 ns	2000 ns	750 ns
Price (1964 $)	$192,000	$216,000	$460,000	$1,080,000
Price (2018 $)	$1,560,000	$1,760,000	$3,720,000	$8,720,000

Fred Brooks, Jr.

Writable Control Store

- **如果控制存储是RAM，就可以修改“固件”**
  “Writable Control Store”
  - 修复微程序中的错误
- **微程序研究在学术界很流行**
  - Patterson Phd Thesis*
  - 有专门的国际会议SIGMICRO**
- **Xerox Alto （Bit Slice TTL） （1973）**
  - 第1台具有GUI和网络的个人计算机
  - BitBlt和网络控制器用微码实现

Chuck Thacker

* *Verification of microprograms*, David Patterson, UCLA, 1976
** “The design of a system for the synthesis of correct microprograms,” David Patterson, *Proc. 8th Annual Workshop of Microprogramming*, 1975

IC技术、微程序技术催生CISC

- **逻辑电路、RAM和ROM使用同样的晶体管实现**
- **半导体RAM的速度与ROM的速度基本相同**
- **控制存储密度会持续增长（Moore's Law)**
- **由于采用RAM存储控制信号，容易修复微代码bug**

- **综上，允许更加复杂的ISAs**
- **例如：小型机（TTL server）**
  - Digital Equipment Corp. (DEC)
  - VAX ISA in 1977
  - 5K × 96b 微码

微处理器演进

- **上世纪70年代，在MOS技术进步的推动下，小型计算机和主机ISAs得到了迅速发展**
- **Intel 4004（1971，740kHz，2250 transistors)**
- **“Microprocessor Wars”：通过添加指令，调整汇编器**
- **Intel iAPX 432：最具雄心的微型计算机始于1975年**
  - 基于32位能力的面向对象体系结构，使用Ada编写的定制操作系统
  - 严重的性能、复杂性(多芯片)和可用性问题导致1981年发布
- **Intel 8086 (1978, 8MHz, 29,000 transistors)**
  - 要求：“Stopgap” 16-bit processor, 52 周开发新的芯片
  - 结果：10人3周开发了汇编级兼容8080的ISA
- **1981年IBM 采用Intel8088 （8位数据总线）研制了IBM PC机**
  - 希望1986年 销售25万台，实际销售1+亿台
  - PC软件的二进制兼容=> 8086前途光明

多选题 4分

设置

此题未设置答案，请点击右侧设置按钮

比较微程序控制器和组合逻辑控制器。选择正确的答案。

A 组合逻辑控制器性能高于微程序控制器

B 微程序控制器性能高于组合逻辑控制器

C 采用组合逻辑控制器设计有利于ISA的扩展

D 采用微程序控制器设计有利于ISA的扩展

提交

80年初：微程序控制机器分析

- **用高级语言编程成为主流**
  - 关键问题：编译器生成指令（ISA vs. Compiler)
- **IBM的John Cocke团队**
  - 为小型计算机801（ECL Server）开发了更简单的 ISA 和编译器
  - 移植到IBM370，仅使用IBM 370的简单的寄存器-寄存器及load/store指令
  - 发现：与原IBM 370相比，性能提高3X
- **80年代初，Emer和Clark （DEC）发现**
  - VAX 11/180 CPI = 10 !
  - VAX ISA 的 20%指令 （占用了60%的微码）仅占用了 0.2%的执行时间
- **Patterson：如何修复微处理器中的微程序bug，投稿'79DEC 后，引发对ISA合理性的研究**

* "A Characterization of Processor Performance in the VAX-11/780," J. Emer and D.Clark, *ISCA*, 1984.
** "RISCy History," David Patterson, May 30, 2018, Computer Architecture Today Blog

From CISC to RISC

- **CISC指令系统主要存在以下几方面的问题：**
  - 指令使用频度；CISC指令系统复杂-〉增加研制时间和成本，容易出错；
  - VLSI设计困难，不利于单片集成；
  - 许多复杂指令操作复杂，运行速度慢；
  - 各条指令不规整，不利于运用先进计算机体系结构技术来提高系统性能。
- **使用简单ISA**
  - 指令像微指令那样简单
  - 编译生成的指令仅是部分CISC指令
  - 能够采用流水线方式实现
- **技术基础**
  - 使用指令Cache
  - 芯片集成度有很大提高：80年代初，单芯片已经可以集成32位数据通路+小规模cache
  - Chaitin*的寄存器分配方法有利于Load-store型ISA

*Chaitin, Gregory J., et al. "Register allocation via coloring." *Computer languages* 6.1 (1981), 47-57

CISC vs. RISC Today

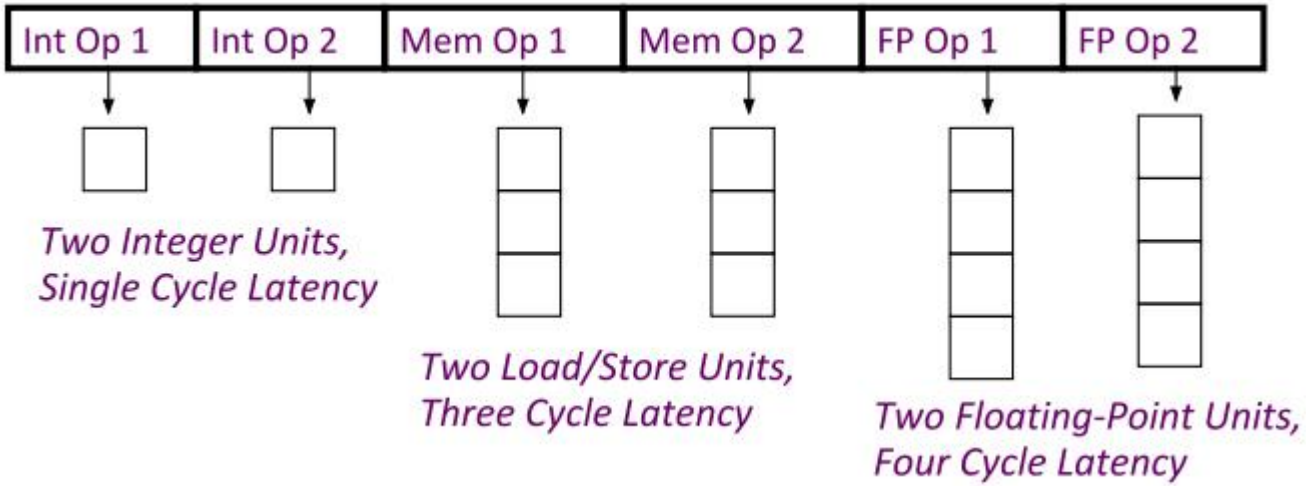
PC 时代

- 硬件将x86指令翻译为内部的RISC 指令
- 在MPU内部使用RISC技术
- x86 ISA在桌面和服务器市场占主导地位
- >350M / year

后PC时代: Client/Cloud

- SoC中的处理器核 vs. MPU
- 芯片面积、能耗和性能同样重要
- 99%的处理器是RISC
- >20B total/year in 2017
  - x86在2011年达到顶峰，现在每年下降~8%，
  - x86服务器=>Cloud ~10M servers
  - Total (0.05% of 20B)

VLIW: Very Long Instruction Word



- 一条指令中包含多个操作
- 每个指令槽表示固定的功能
- 指定了恒定的操作延时
- 体系结构必须保证
  - 指令中的操作是并行的 => no cross-operation RAW check
  - 数据未准备好不使用数据 => no data interlocks

From RISC to Intel/HP Itanium，EPIC IA-64

- **EPIC 是Intel为他们的VLIW结构的命名**
  - Explicitly Parallel Instruction Computing
  - 二进制 目标码兼容的 VLIW
  - 从**1994**年与HP合作开发
- **EPIC IA-64 是 Intel 32位x86的后继（64位ISA）**
  - IA-64= Intel Architecture 64-bit
  - AMD 有自己的AMD64 技术,2003年推出业界首款64位处理器
  - 第一款Itanium 2002年推出，不兼容IA-32
  - 很多公司放弃RISC转而选择Itanium，因为他们普遍认为这是必然的（Microsoft, SGI, Hitachi,…

VLIW的问题及EPIC的失败

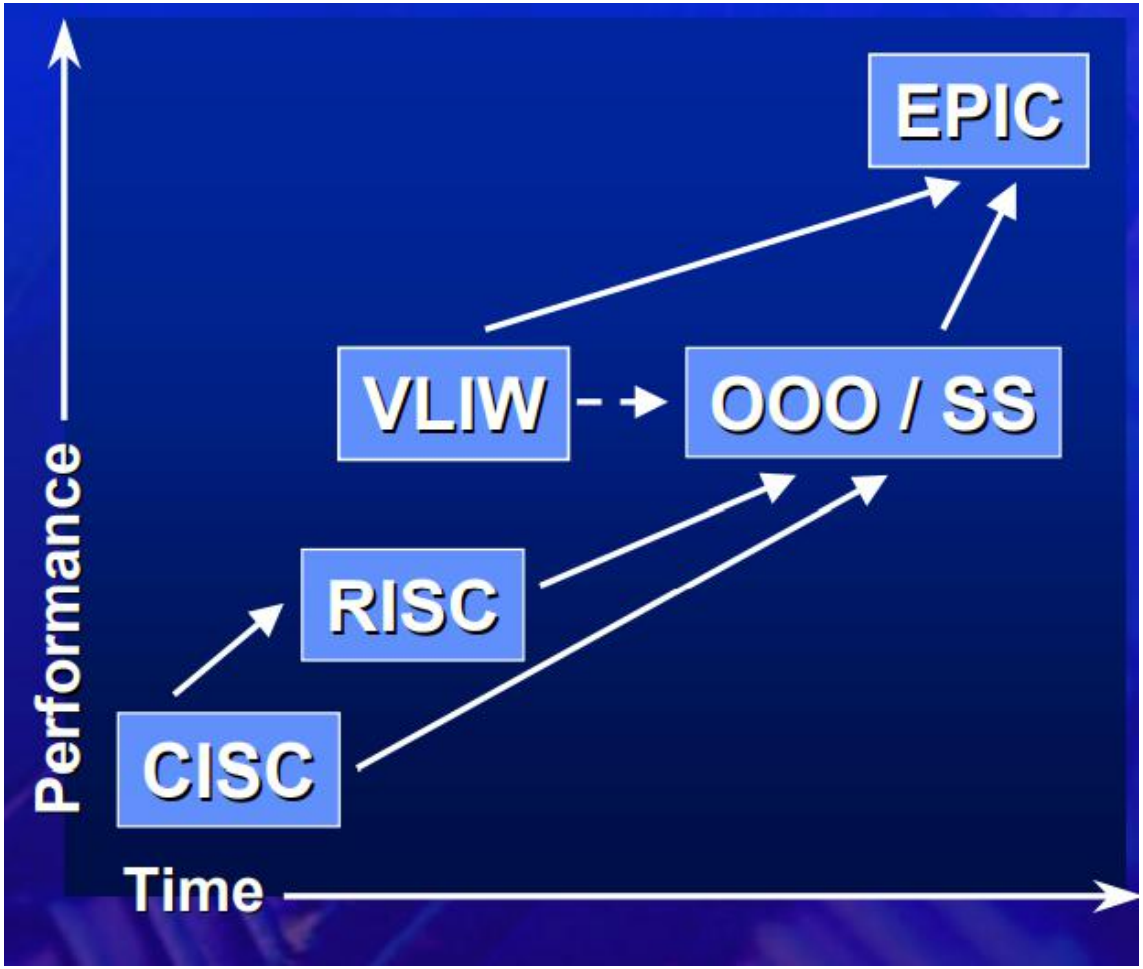
- 编译器无法处理整型类代码(指针)中的复杂依赖项
- 代码量膨胀
- **不可预知的分支**
- **可变的存储访问延迟（不可预知的cache失效）**
  - 乱序执行技术可处理Cache延迟
- **乱序执行覆盖了VLIW的优势**

"The Itanium approach...was supposed to be so terrific – until it turned out that the wished-for compilers were basically impossible to write."
- Donald Knuth, Stanford

小结：体系结构发展历史

- **重要事件**
  - IBM 系列机、微程序设计、RISC、VLIW、EPIC
- **30年来没有出现新的通用CISC ISA**
- **15年来没有出现新的通用VLIW**
  - 在通用计算领域VLIW是失败的
  - 复杂的VLIW结构接近顺序超标量结构，但在大型复杂应用中不存在优势
  - VLIW在嵌入式DSP市场比较成功
    - 简单VLIW，分支简单、没有Cache，小程序
- **RISC！普遍认为RISC原则是通用ISA的最好原则**
- **Great ideas in Computer Architecture**

体系结构设计理念(philosophy)



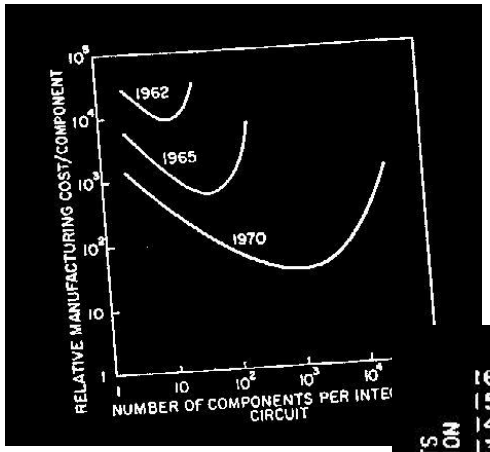
关注点：
SW/HW Interface
性能提升（ILP）

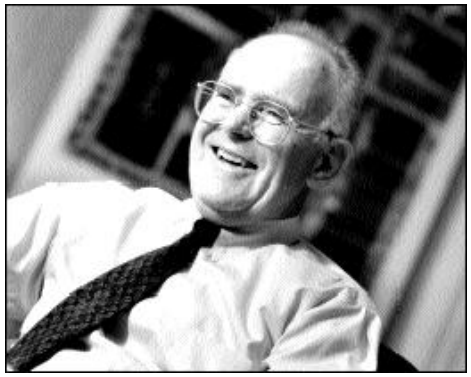
https://www.cs.helsinki.fi/u/kerola/tikra/IA64-Architecture.pdf

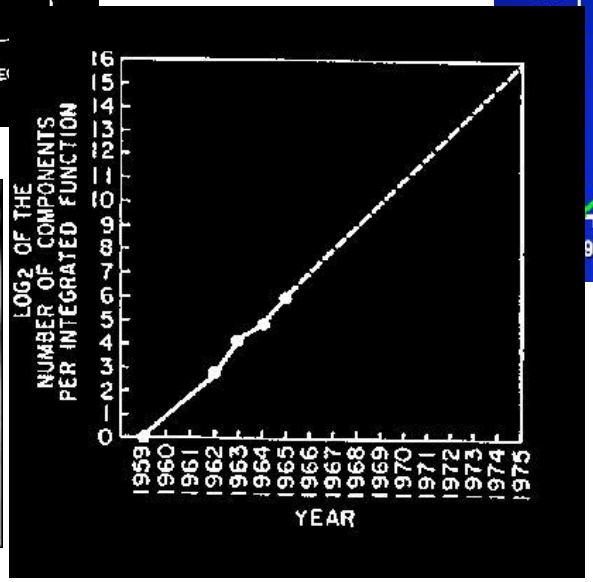
Great ideas in Computer Architecture

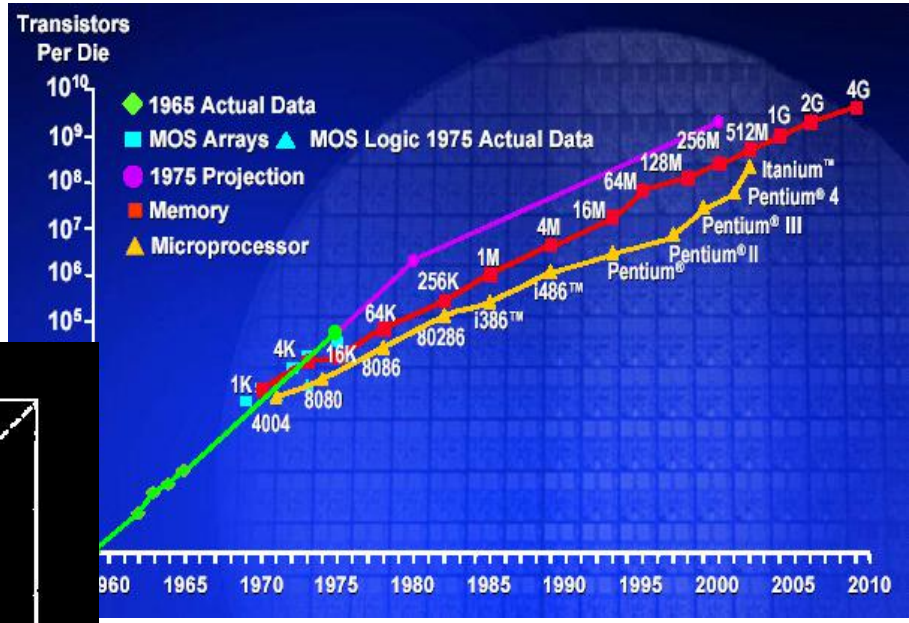
1. **Design for Moore's Law**
2. **Abstraction to Simplify Design**
3. **Make the Common Case Fast**
4. **Dependability via Redundancy**
5. **Memory Hierarchy**
6. **Performance via Parallelism/Pipelining/Prediction**

Moore's Law



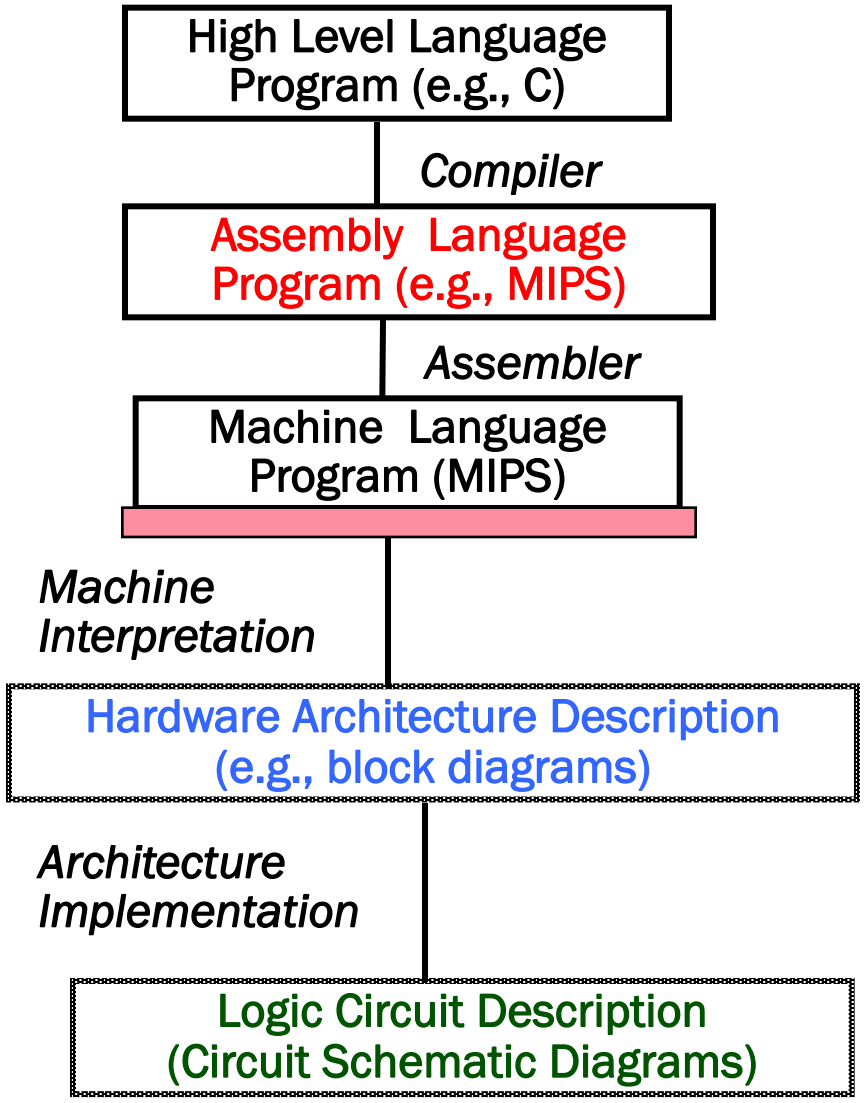






- "Cramming More Components onto Integrated Circuits"
  - Gordon Moore, Electronics, 1965
- # on transistors on cost-effective integrated circuit double every 18 months

Abstraction via Layers of Representation

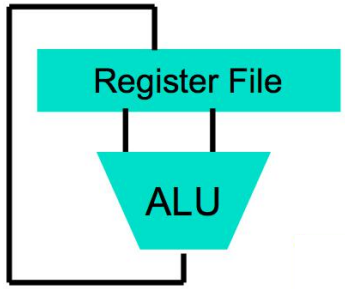


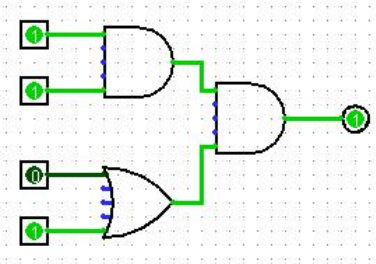

```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

```
lw    $t0, 0($2)
lw    $t1, 4($2)
sw    $t1, 0($2)
sw    $t0, 4($2)
```

Anything can be represented as a *number*, i.e., data or instructions

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```



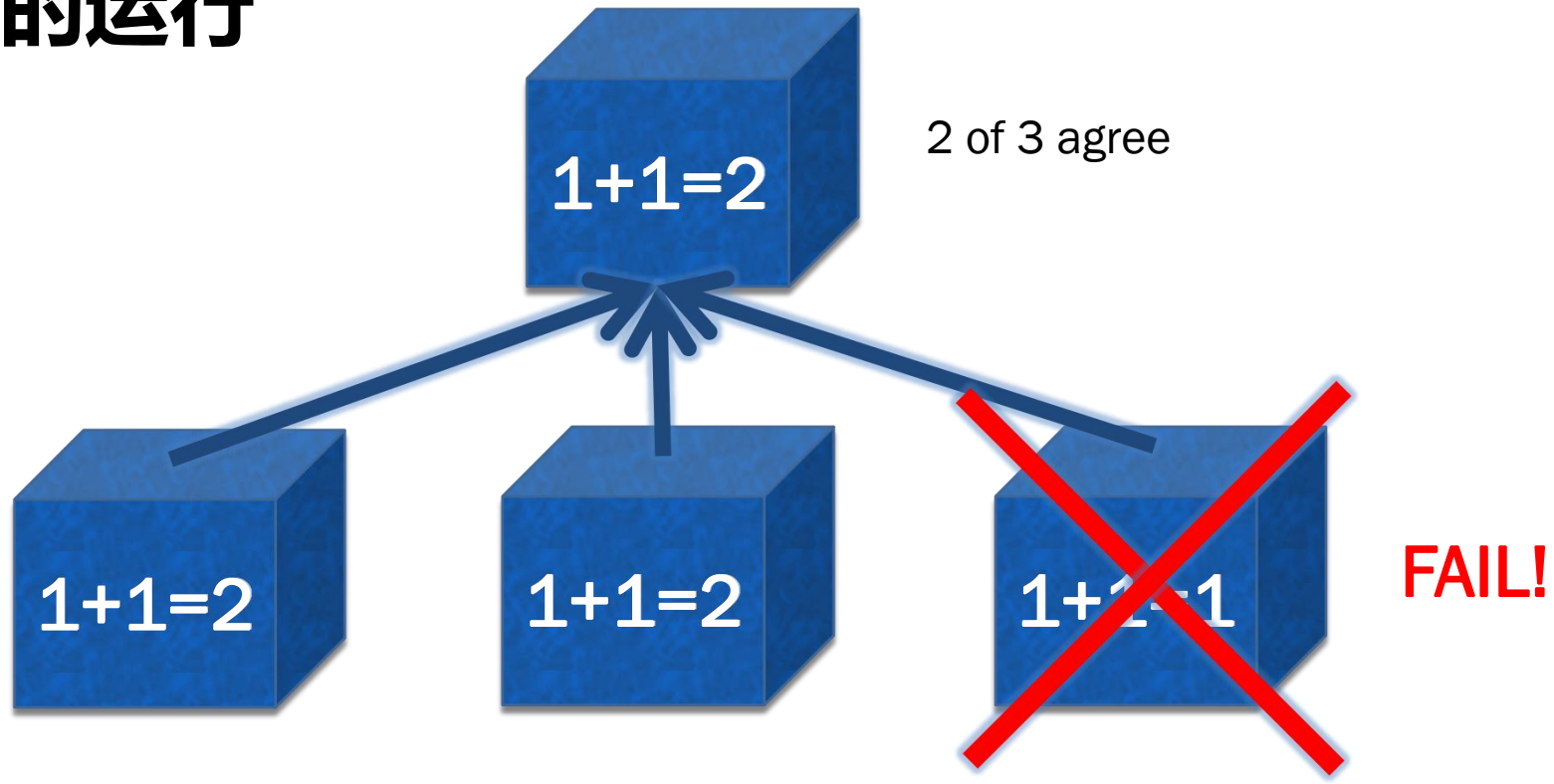


Make the Common Case Fast

- **在进行设计选择时，注重优化经常发生的事件**
- **优化不经常执行的代码意义不大**
- **选择一种（性能）度量方式来确定经常性事件（Common case）**

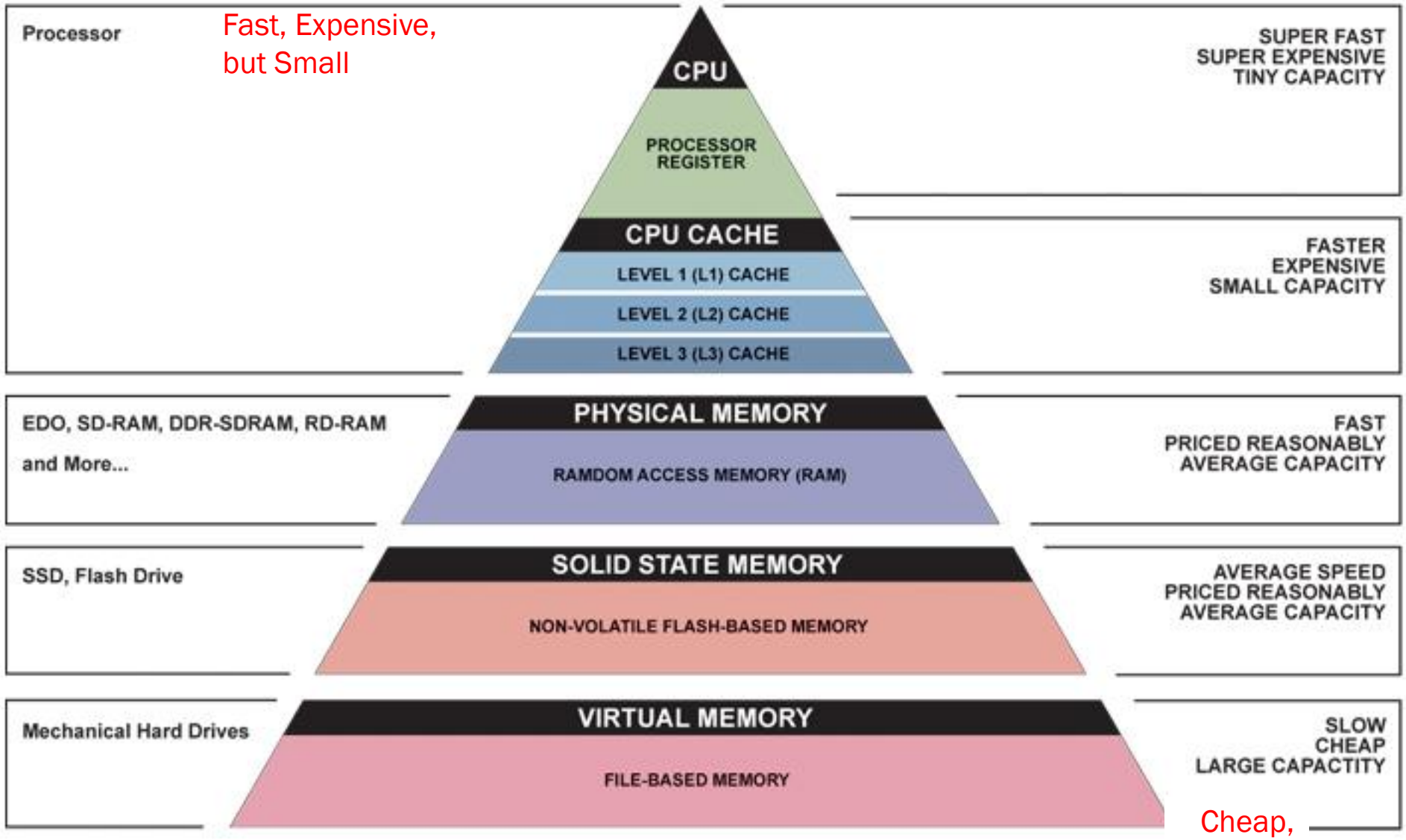
Dependability via Redundancy

- **通过冗余使得部分部件失效不影响整个系统的运行**

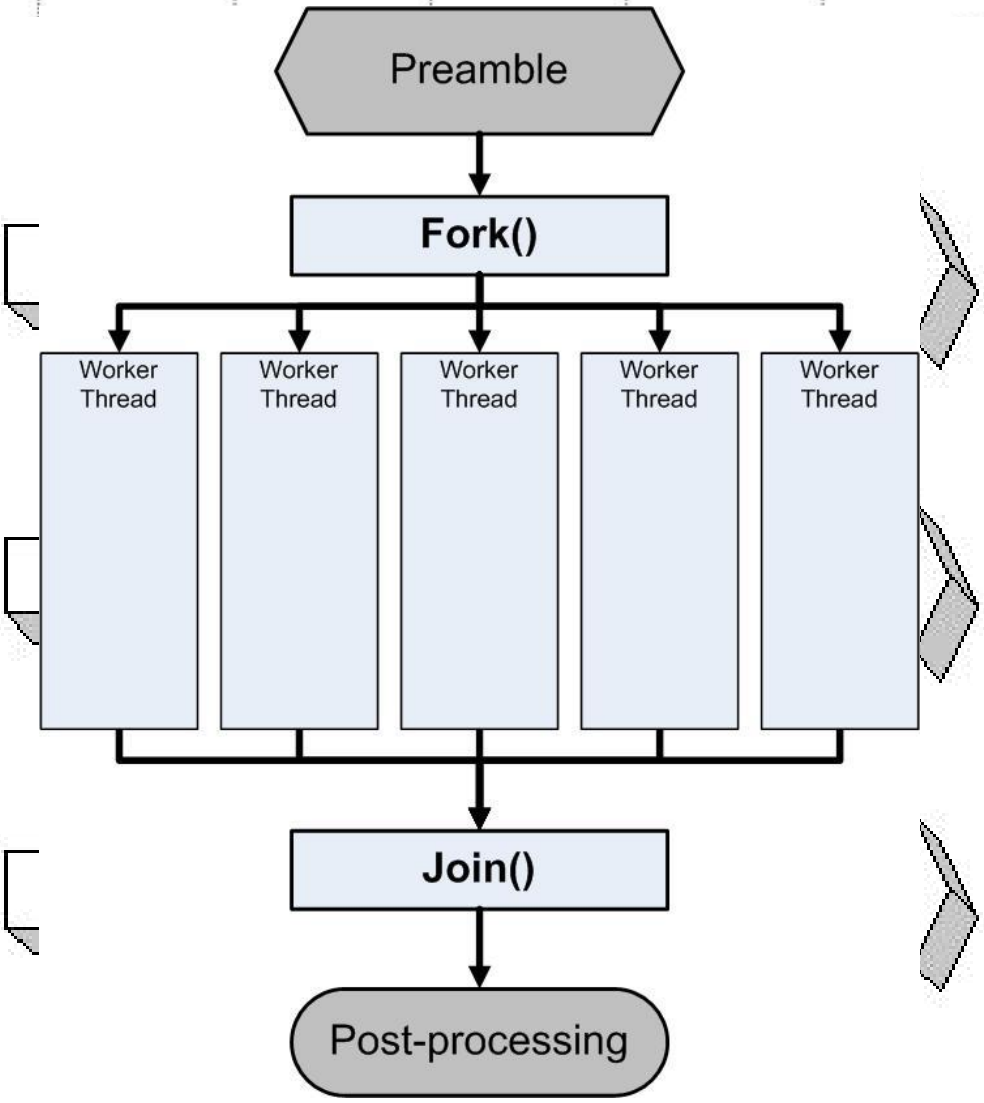


Increasing transistor density reduces the cost of redundancy

Memory Hierarchy



Parallelism/Pipelining/Prediction



1.2 体系结构的发展历史、现状及趋势



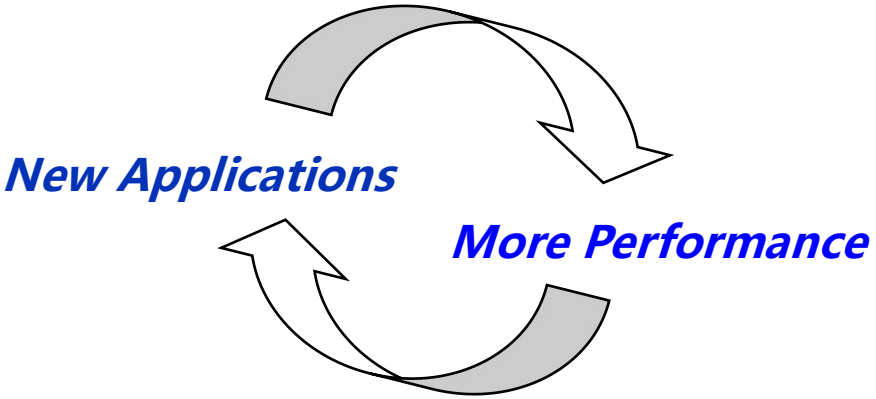
历史 现状 趋势

影响体系结构技术发展的主要因素

- **产品应用方面**
  - PC/Server => IoT, Mobile/Cloud
- **集成电路技术方面**
  - 登纳德缩放比例定律的终结：功耗成为关键制约条件
  - Moore定律的终结：晶体管的集成度提高延缓
- **体系结构方面**
  - 挖掘指令级并行性方面的限制和低效，04年结束了单处理器时代
  - Amdahl定律暗示了简单多核时代的结束

Application Trends

- **应用需求驱动硬件技术的发展**
  - 科学计算→ 网络服务、移动计算.....
- **硬件技术的发展催生新的应用**
  - 微处理器的性能随着主频的提升大幅提高
  - 对性能的需求推动着并行体系结构的发展
    - For most demanding applications



New "Great Ideas"

Personal Mobile Devices

Warehouse Scale Computer

New "Great Ideas"

Software

- Parallel Requests
  Assigned to computer
  e.g., Search "Katz"
- Parallel Threads
  Assigned to core
  e.g., Lookup, Ads
- Parallel Instructions
  >1 instruction @ one time
  e.g., 5 pipelined instructions
- Parallel Data
  >1 data item @ one time
  e.g., Add of 4 pairs of words
- Hardware Descriptions
  All gates functioning in parallel at same time
- Programming Languages

Leverage Parallelism & Achieve High Performance

Hardware

Warehouse Scale Computer

Smart Phone

计算机的分类

- **个人移动设备 (PMD)**
  - 智能手机、平板电脑
  - ARM-ISA兼容的通用处理器芯片（SoC）在市场上处于统治地位
  - SoC中包含大量的专用加速器 (radio, image, video, graphics, audio, motion, location, security, etc.)
  - 强调能效和实时性(energy efficiency and real-time)
- **桌面计算（Desktop Computing）**
  - 强调性价比(price-performance)
- **服务器（Servers）**
  - 强调可用性、可缩放性、吞吐率(availability, scalability, throughput)

Cost of downtime

Application	Cost of downtime per hour	Annual losses with downtime of		
		1% (87.6 h/year)	0.5% (43.8 h/year)	0.1% (8.8 h/year)
Brokerage service	$4,000,000	$350,400,000	$175,200,000	$35,000,000
Energy	$1,750,000	$153,300,000	$76,700,000	$15,300,000
Telecom	$1,250,000	$109,500,000	$54,800,000	$11,000,000
Manufacturing	$1,000,000	$87,600,000	$43,800,000	$8,800,000
Retail	$650,000	$56,900,000	$28,500,000	$5,700,000
Health care	$400,000	$35,000,000	$17,500,000	$3,500,000
Media	$50,000	$4,400,000	$2,200,000	$400,000

Figure 1.3 Costs rounded to nearest $100,000 of an unavailable system are shown by analyzing the cost of downtime (in terms of immediately lost revenue), assuming three different levels of availability, and that downtime is distributed uniformly. These data are from Landstrom (2014) and were collected and analyzed by Contingency Planning Research.

计算机的分类（续）

- **集群/仓储级计算机（Clusters / Warehouse Scale Computers）**
  - 每个数据中心包含10+万个处理器核
  - X86-ISA兼容的服务器芯片在市场上占统治地位
  - 专门的应用程序，加上虚拟机的云托管
  - 现在越来越多地使用gpu、fpga和定制硬件来加速工作负载
  - 其分支：Supercomputers：强调浮点运算性能和高速内部互联
  - 强调可用性和性价比(availability、price-performance、energy)
- **嵌入式计算机/物联网（Embedded Computers / Internet of Things）**
  - 有线/无线网络基础设施，打印机
  - 消费类产品(TV/Music/Games/Automotive/Camera/MP3)
  - 物联网（Internet of Things!)
  - 强调价格、能耗及面向特定应用的性能(Emphasis：price、energy、application-specific performance)

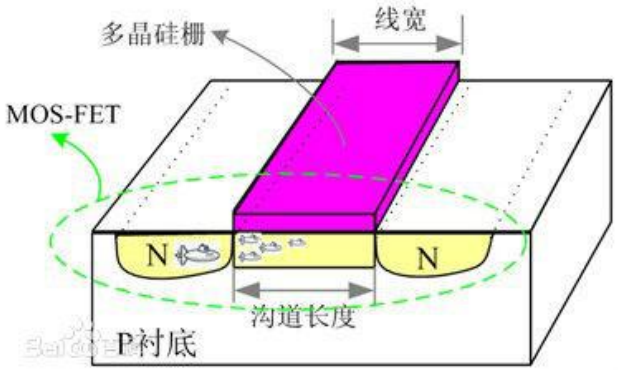
五类主流计算系统特点

Feature	Personal mobile device (PMD)	Desktop	Server	Clusters/warehouse-scale computer	Internet of things/ embedded
Price of system	$100–$1000	$300–$2500	$5000–$10,000,000	$100,000–$200,000,000	$10–$100,000
Price of microprocessor	$10–$100	$50–$500	$200–$2000	$50–$250	$0.01–$100
Critical system design issues	Cost, energy, media performance, responsiveness	Price-performance, energy, graphics performance	Throughput, availability, scalability, energy	Price-performance, throughput, energy proportionality	Price, energy, application-specific performance

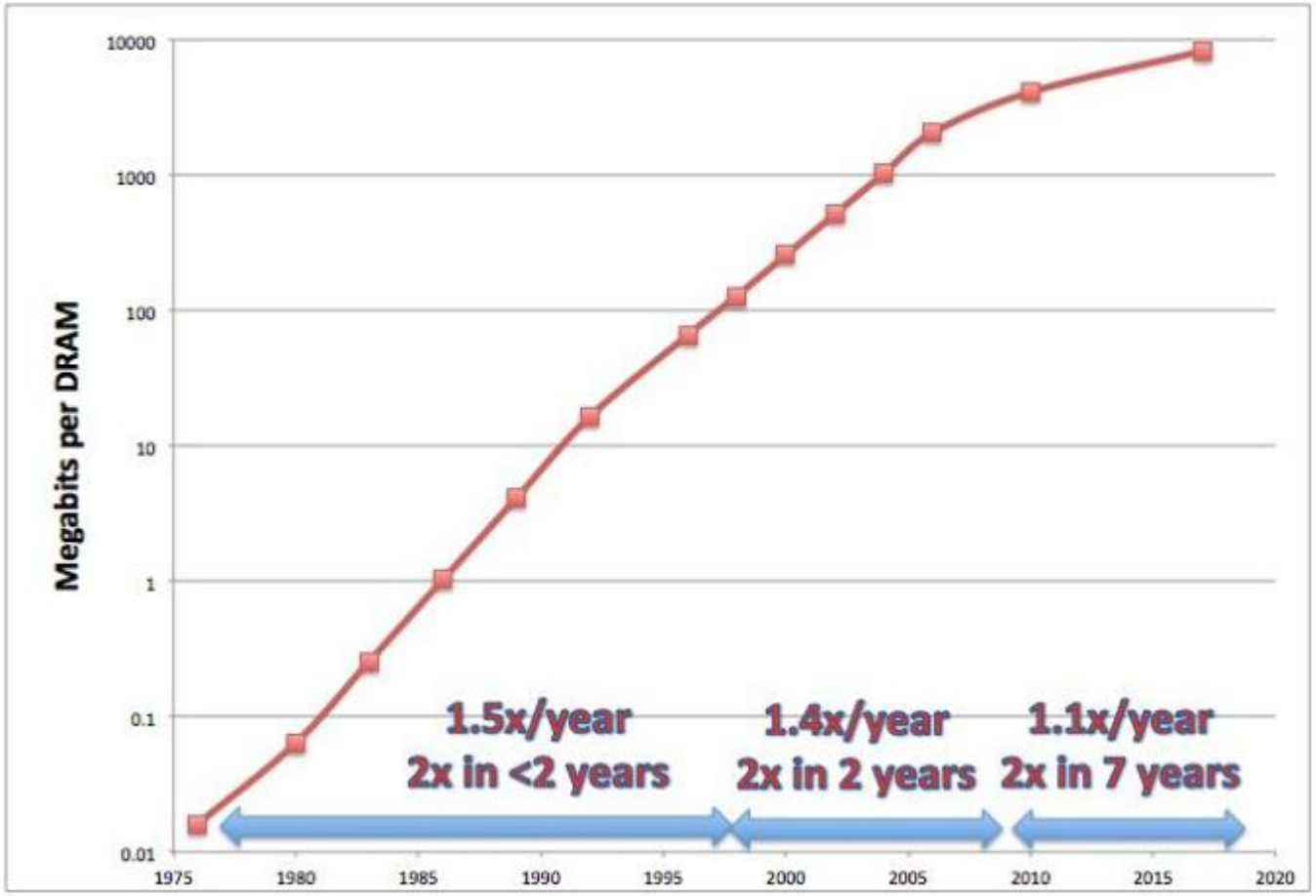
Figure 1.2 A summary of the five mainstream computing classes and their system characteristics. Sales in 2015 included about 1.6 billion PMDs (90% cell phones), 275 million desktop PCs, and 15 million servers. The total number of embedded processors sold was nearly 19 billion. In total, 14.8 billion ARM-technology-based chips were shipped in 2015. Note the wide range in system price for servers and embedded systems, which go from USB keys to network routers. For servers, this range arises from the need for very large-scale multiprocessor systems for high-end transaction processing.

Closer Look at Processor Technology

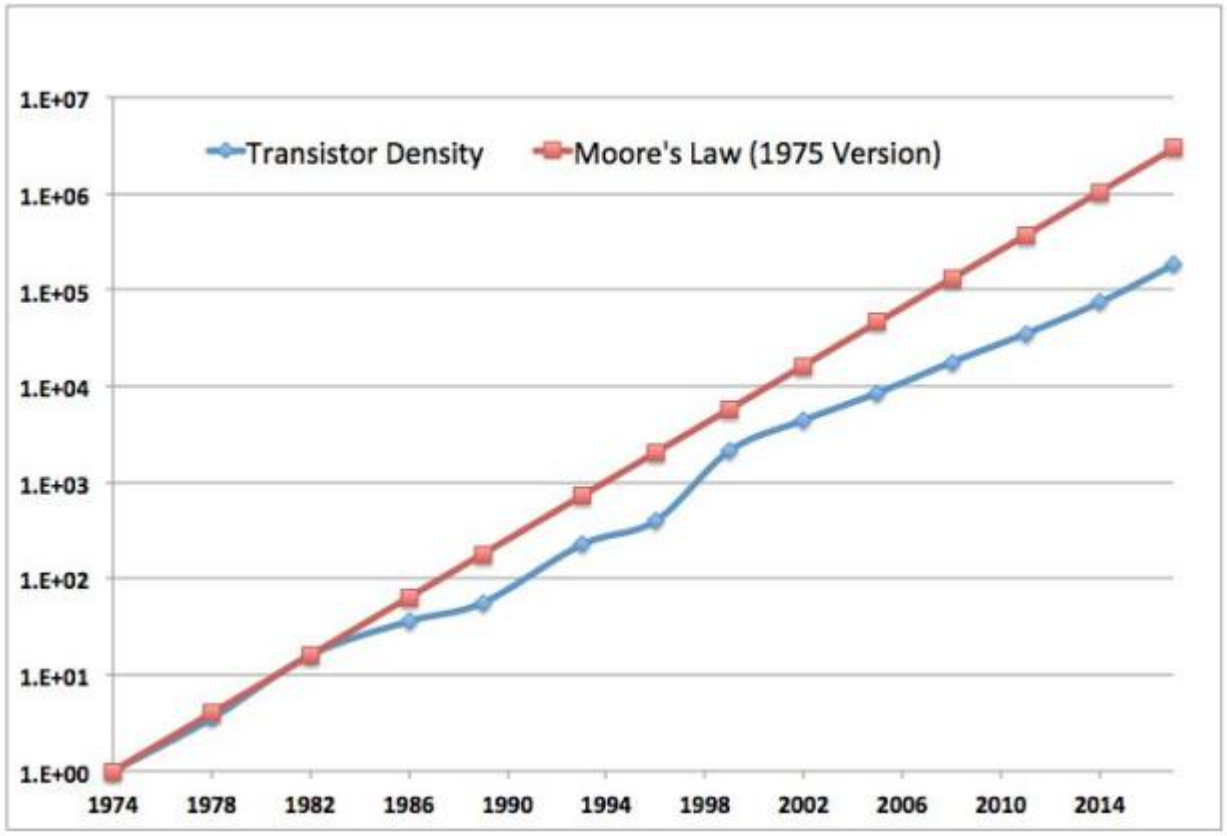
- **特征尺寸不断减小（feature size），芯片集成度不断提高**
  - 电路的速度在不断提高
  - 芯片的面积在不断增加
  - 主频在提高（功耗成为问题)
  - 单位面积的晶体管数量在增加
- **性能每10年提高100倍**
  - 时钟频率提高了10倍
  - DRAM 的密度每3年提高4倍
- **如何使用这些晶体管?**
  - Parallelism in processing: 有更多的功能部件
    - 每个时钟周期并行多个操作降低了CPI
  - Locality in data access: 更大的Cache
    - 降低了访存的延时从而降低了CPI，提高了处理器的利用率



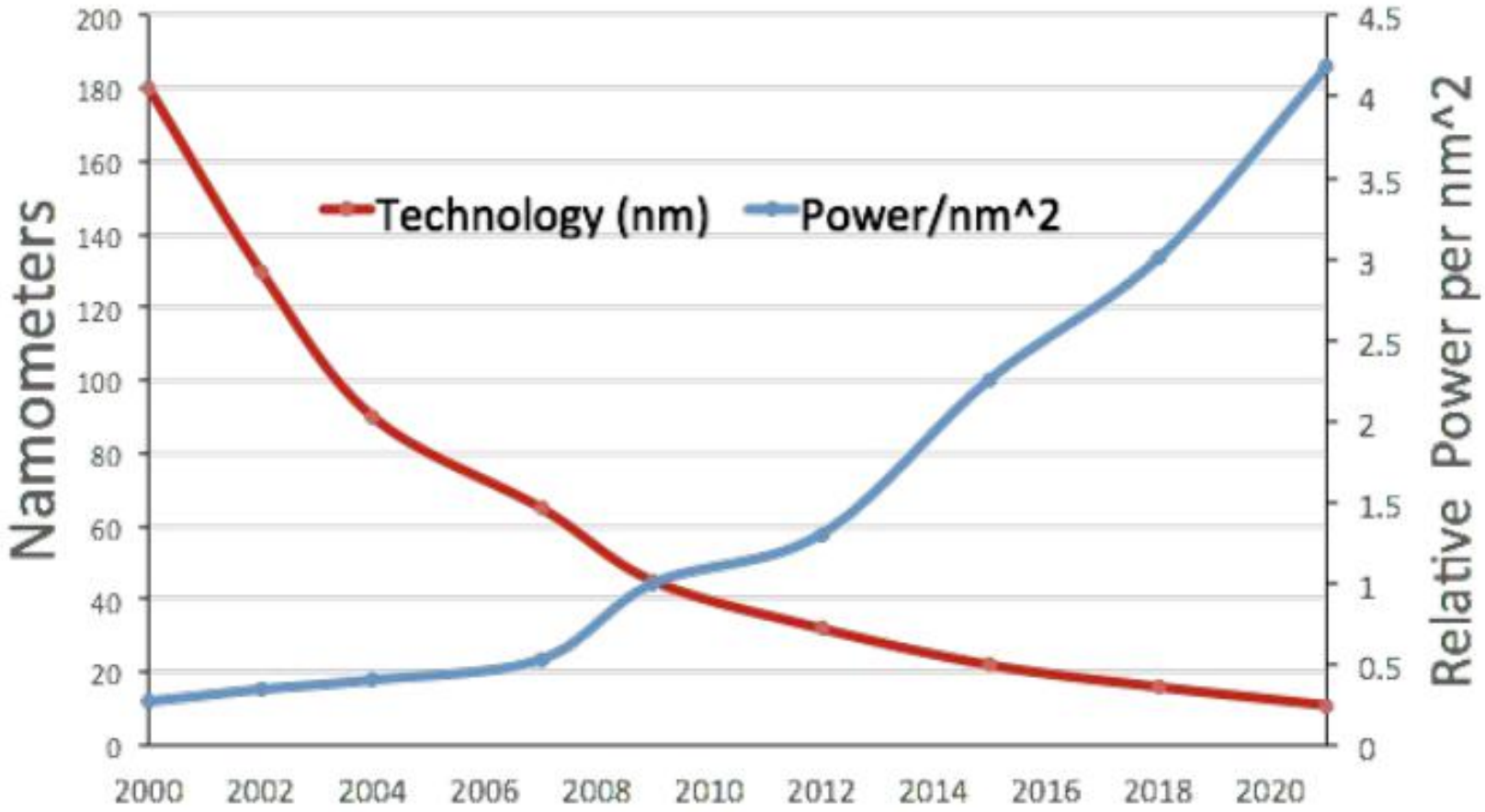
Moore's Law in DRAMs



Moore's Law Slowdown in Intel Processors



Technology & Power: Dennard Scaling



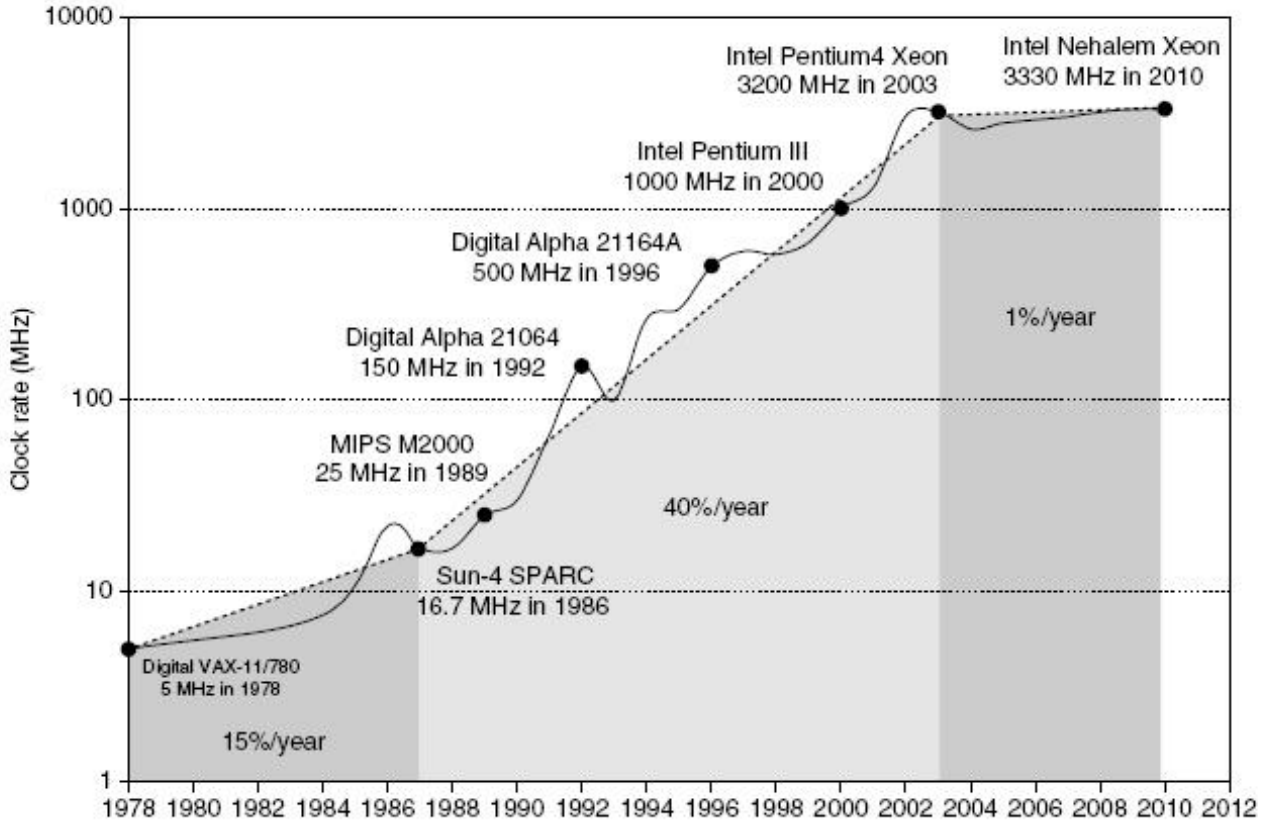
Energy scaling for fixed task is better, since more and faster transistors

Power & Energy

- **Dynamic Energy ∝ Capacitive Load × Voltage²**
  - 从0-1-0 或 1-0-1逻辑跃迁的脉冲能量
  - Capacitive Load =输出晶体管和导线的电容负载
  - 20年来晶体管供电电压已经从5V降到1V
- **Dynamic Power ∝**

  Capacitive Load × Voltage² × Frequency Switched

Power

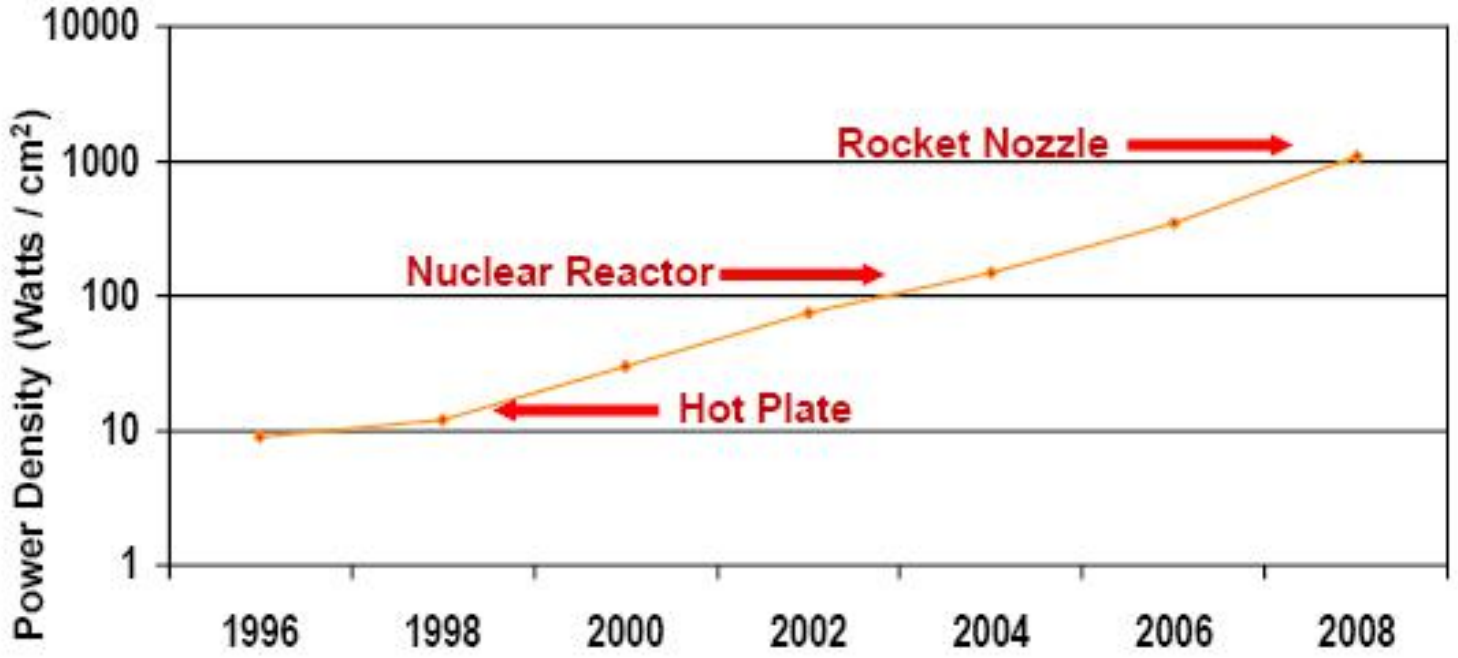


- **Intel 80386 consumed ~ 2 W**
- **3.3 GHz Intel Core i7 consumes 130 W**
- **Heat must be dissipated from 1.5 x 1.5 cm chip**
- **This is the limit of what can be cooled by air**

Limiting Force: Power Density

Moore's Law Extrapolation:

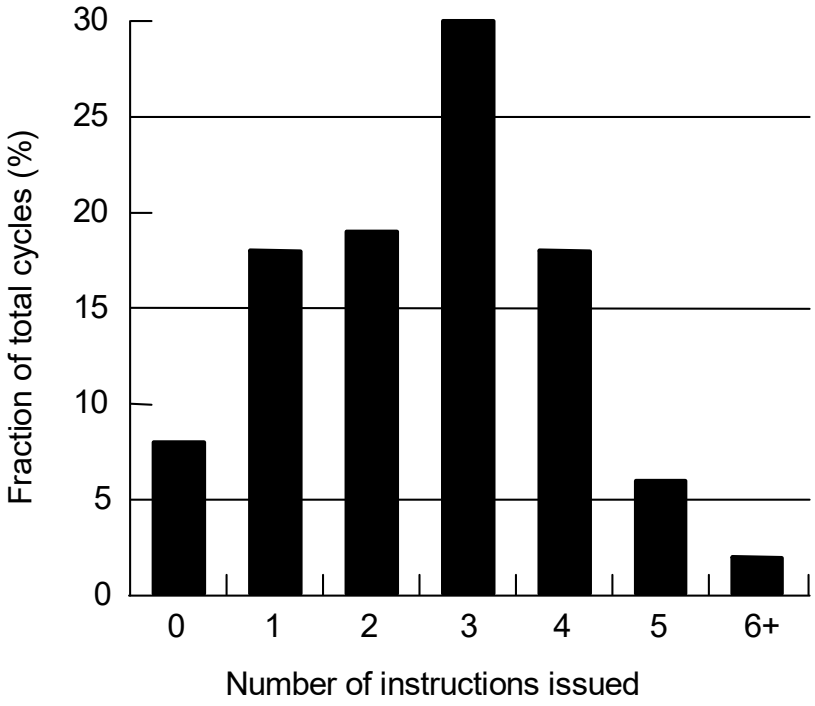
Power Density for Leading Edge Microprocessors

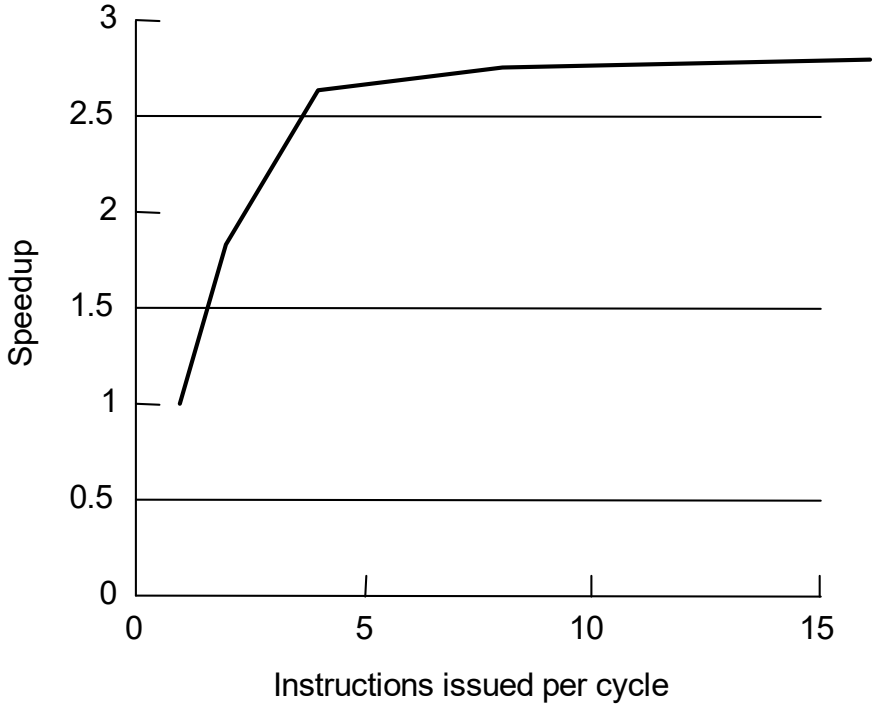


Power Density Becomes Too High to Cool Chips Inexpensively

Source: Shekhar Borkar, Intel Corp

How far will ILP go?

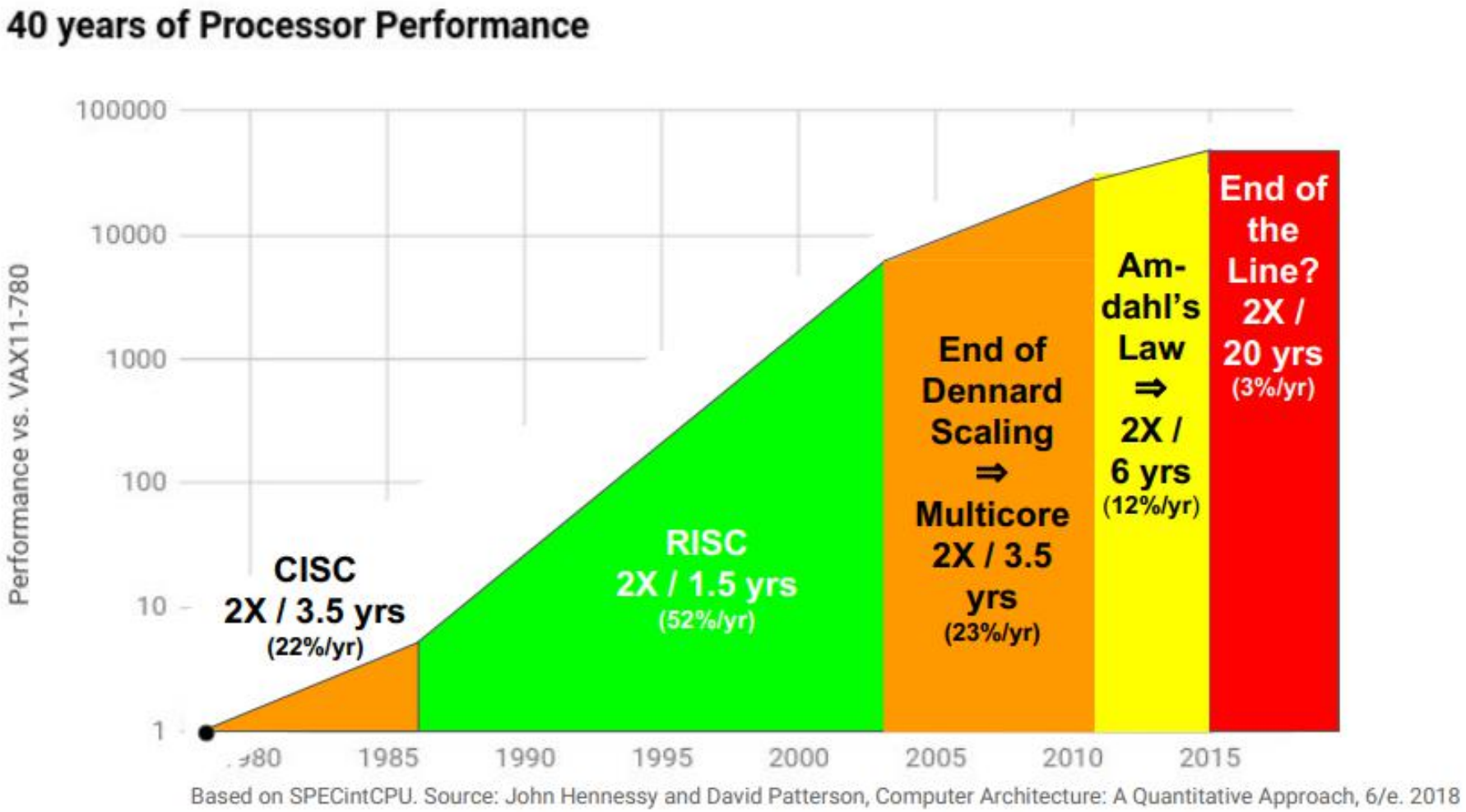




理想超标量模型执行下有限的ILP：无限的资源和存取带宽、完善的预取和重命名机制，但是使用现实的Cache。

结果：90%的时间每个时钟周期最多只能并行执行4条指令.

End of Growth of Single Program Speed?



Architecture: Increase in Parallelism

- **1985年以前位级(4-8-16位)并行（Bit level parallelism)**
  - 32-bit 处理器后性能提升变慢
  - 90年代采用64-bit 、128-bit 及更高位 通过向量处理提高性能
  - 重大的拐点：32位处理器和缓存集成在一个芯片
- **80年代中到90年代后期主要发展指令级并行（Instruction Level Parallelism)**
  - 流水线、RISC、编译技术的进步→ 挖掘指令级并行
  - 片上cache及功能部件的增加-> 超标量执行
  - 更精巧的技术: 乱序执行和硬件投机执行
- **现状：线程级并行和片上多处理器（thread level parallelism and chip multiprocessors)**
  - 线程级并行超越了指令集并行
  - 在单个芯片中部署多个处理器及互联结构
  - 在处理器芯片内多个线程并行执行

小结-计算机系统分类

- **计算机系统**
  - 个人移动设备 (PMD)
    - Emphasis on energy efficiency and real-time
  - 桌面计算（Desktop Computing）
    - Emphasis on price-performance
  - 服务器（Servers）
    - Emphasis on availability, scalability, throughput
  - 集群/仓储级计算机（Clusters / Warehouse Scale Computers）
    - Emphasis on availability and price-performance
  - 嵌入式计算机（Embedded Computers）
    - Emphasis: price

小结：计算机系统设计方面的巨大变化

- **在过去的50年，Moore’s law和Dennard scaling(登纳德缩放比例定律)主宰着芯片产业的发展**
  - Moore 65年预测：晶体管数量随着尺寸缩小按接近平方关系增长
  - Dennard 74年预测：晶体管尺寸变小，功耗会同比变小（相同面积下功耗不变）
  - 工艺技术的进步可在不改变软件模型的情况下，持续地提高系统性能/能耗比
- **最近10年间，工艺技术的发展受到了很大制约**
  - Dennard scaling over (supply voltage ~fixed)
  - Moore’s Law (cost/transistor) over?
  - Energy efficiency constrains everything
  - ……
- **功耗问题成为系统结构设计必须考虑的问题**
- **软件设计者必须考虑:**
  - Parallel systems
  - Heterogeneous systems

有关体系结构的新旧观念

- **Old Conventional Wisdom: Power is free, Transistors expensive**
- **New CW: "Power wall" Power expensive, Transistors free**

- **Old CW: 通过编译、体系结构创新来增加指令级并行 (Out-of-order, speculation, VLIW, …)**
- **New CW: "ILP wall" 挖掘指令级并行的收益越来越小**

- **Old CW: 乘法器速度较慢，访存速度比较快**
- **New CW: "Memory wall" 乘法器速度提升了，访存成为瓶颈 (200 clock cycles to DRAM memory, 4 clocks for multiply)**

- **Old CW: 单处理器性能2X / 1.5 yrs**
- **New CW: Power Wall + ILP Wall + Memory Wall = Brick Wall**
  - **单处理器性能 2X / 5(?) yrs**

  ⇒ 芯片设计的巨大变化: multiple "cores" (2X processors per chip / ~ 2 years)
    - 越简单的处理器越节能

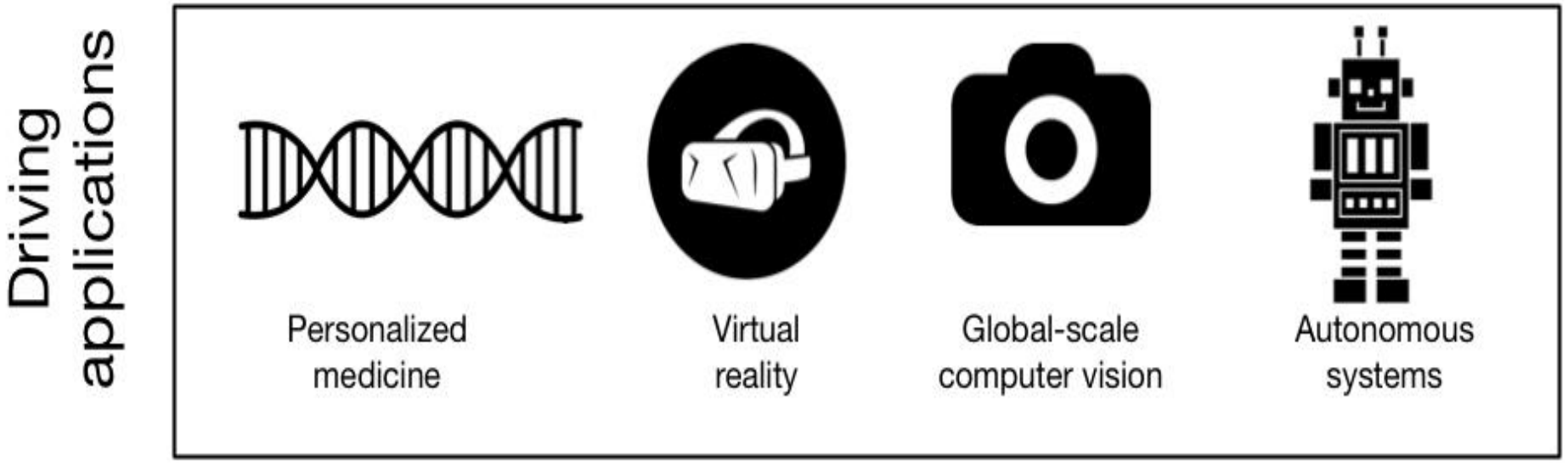
1.2 体系结构的发展历史、现状及趋势



历史 现状 趋势

1. Ceze L , Hill M D , Wenisch T F; “Arch2030: A Vision of Computer Architecture Research over the Next 15 Years” ; 2016; (https://research.cs.wisc.edu/multifacet/papers/ccc16_arch2030.pdf).
2. A community white paper; “21st Century Computer Architecture”; Computing Community Consortium; May 25, 2012; (http://cra.org/ccc/docs/init/21stcenturyarchitecturewhitepaper.pdf)
3. John Hennessy and David Patterson; “A New Golden Age for Computer Architecture: Domain-Specific Hardware/Software Co-Design, Enhuanced Security, Open Instruction Sets, and Agile Chip Development”; June 4, 2018; https://www.youtube.com/watch?v=3LVeEjsn8Ts

21st 新的应用层出不穷



机器学习是系统的关键负载（重要应用）

21世纪体系结构发展

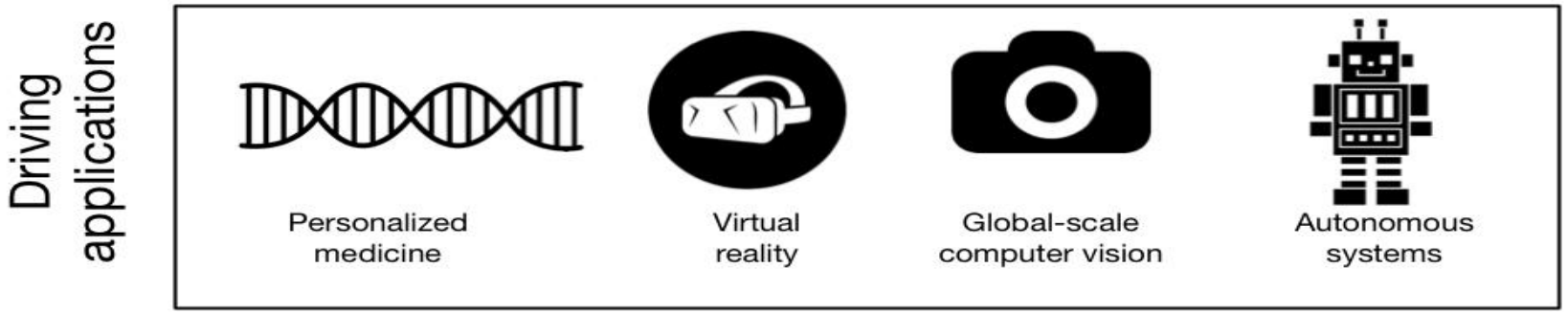
21st Century Comp Architecture

20th Century	21st Century	
Single-chip in stand-alone computer	**Architecture as Infrastructure:** Spanning sensors to clouds Performance + security, privacy, availability, programmability, …	**Cross-Cutting:** Break current layers with new interfaces
Performance via invisible instr.-level parallelism	**Energy First** • Parallelism • Specialization • Cross-layer design	
Predictable technologies: CMOS, DRAM, & disks	**New technologies** (non-volatile memory, near-threshold, 3D, photonics, …) Rethink: memory & storage, reliability, communication	

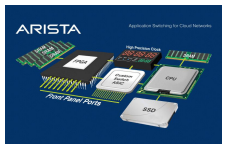
16

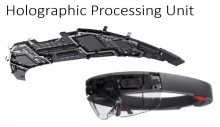
A community white paper; "21st Century Computer Architecture"; Computing Community Consortium; May 25, 2012;

2012以来的发展状况

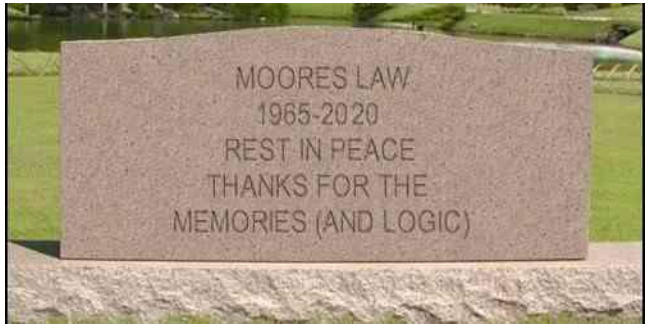


- 机器学习是系统的关键负载（重要应用）
- 专用加速器已经大规模使用





- 云计算无处不在
- 摩尔定律正走向失效得到普遍接受



现代体系结构发展的机遇

Observation	*Implications for next 15 year*
1. Specialization gap	Democratize HW design: tools and open source designs
2. Ubiquitous cloud: innovation abstraction	Cloud model provides practical deployment path for new architectures
3. 3D stacking is real	Opportunities for new architectures and integration models
4. Getting “closer to physics”	Need for more adventurous architectures
5.Machine learning as key app. component	New architectures are enablers: need real collaboration with core ML community

现代体系结构发展的机遇

- 硬件设计“大众化”
- 云计算模型的可扩放和虚拟化特性，可以透明、低成本地提供硬件创新的环境
- 3D集成提供了一个新的可伸缩性维度
- 新的物理层技术可能会催生全新的体系结构诞生
- ……

硬件设计“大众化”

- **Open Architectures**
  - 软件技术的进步激发体系结构创新
  - 为什么有开放的编译器、操作系统而没有开放的ISA
  - RISC Five 第五代Berkeley RISC
- **Domain Specific Languages and Architecture**
  - 提高性能的路径：Domain Specific Architectures(DSAs)
  - 根据应用特征调整体系结构来实现更高的效率
    - 对于特定的领域，更有效的挖掘计算并行性
    - 对于特定的领域，更有效的利用内存带宽
    - 降低不必要的精度
  - 并不仅针对一个专门的应用(ASIC), 而是通过执行软件适应某一领域。
    - 例如机器学习芯片。寒武纪芯片、TPU芯片
- **Agile Hardware Development**

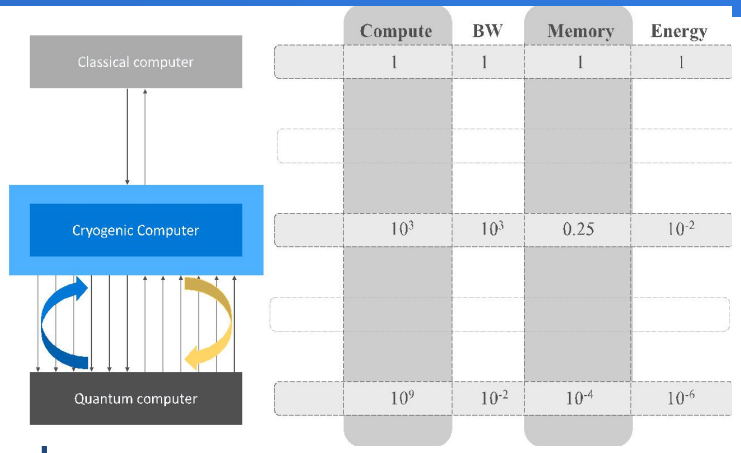
Getting Closer to Physics （全新体系结构）

New memories and devices

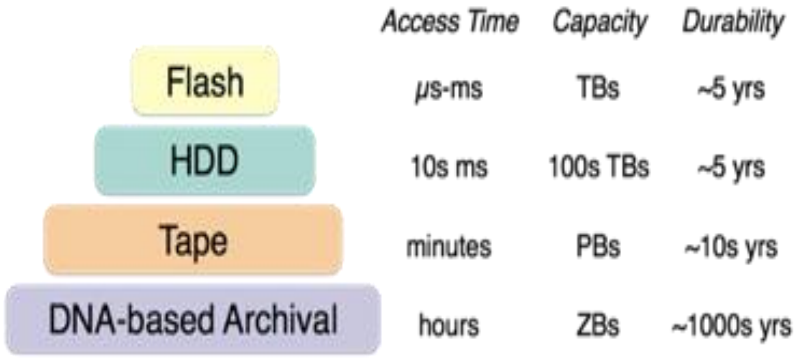
Carbon nanotubes

Quantum computing and superconducting logic

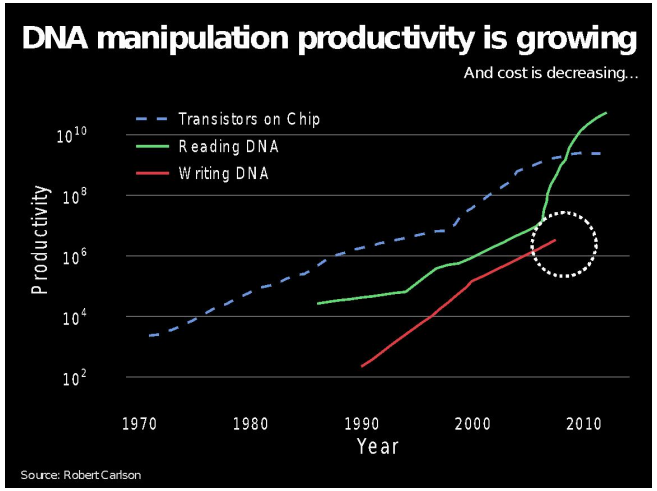
Borrowing from biology



[Doug Carmean, ISCA'16 Keynote]



[Bornholt et al.]



计算机安全现状及挑战

- **虽然目前有一些保护措施，但没有很好使用**
  - 域、环等机制
  - 似乎作用不大，并且开销较高
- **过去希望：软件能够消除攻击路径**
  - 程序验证
  - 内核态、用户态
- **Spectre & meltdown漏洞: 利用推断执行机制=>时序攻击**
- **许多微体系架构存在这类漏洞，这些漏洞是体系结构定义的bug**
- **需要CA2.0来避免利用这类漏洞的攻击**
- **硬件必须有助于安全系统的建立**

主观题 10分

设置

请谈谈你对降低硬件设计门槛的看法。我们可以从哪些方面努力来降低硬件设计的难度、成本等?

正常使用主观题需2.0以上版本雨课堂

作答

Acknowledgements

- **These slides contain material developed and copyright by:**
  - John Kubiatowicz (UCB)
  - Krste Asanovic (UCB)
  - John Hennessy (Standford)and David Patterson (UCB)
  - Chenxi Zhang (Tongji)
  - Muhamed Mudawar (KFUPM)
  - Onur Mutlu（ETH Zürich）
- **UCB material derived from course CS152、CS252、CS61C**
- **KFUPM material derived from course COE501、COE502**
- **https://people.cs.clemson.edu/~mark/hist.html**
- **https://research.cs.wisc.edu/multifacet/papers/ccc16_arch2030_outbrief17_03**