

COMPUTER ORGANIZATION

Lecture 1 Course Introduction

2024 Spring

Why to Learn Computer Organization?

- Embarrassing if you are a student in CS and can't make sense of the following terms: DRAM, SRAM, pipelining, cache hierarchies, I/O, virtual memory, ...
- Embarrassing if you are a student in CS and can't decide which processor to buy: 3 GHz P4 or 2.5 GHz Athlon (this course helps us reason about performance/power), ...
- First step for chip designers, compiler/ OS writers
- Knowledge of the hardware will help you write better programs

Must a Programmer Care about Hardware?

- Must know how to reason about program performance and energy
- CPU Performance: if we understand how CPU process data, we can enhance the computation efficiency
- Memory management: if we understand how/where data is placed, we can help ensure that relevant data is nearby
- Thread management: if we understand how threads interact, we can write smarter multi-threaded programs
- I/O management

What you have learned?

- Binary numbers
- Read and write basic C/Java programs
- Understand the steps in compiling and executing a program
- Digital Circuit, Logic design:
 - Logical equations, schematic diagrams
 - Combinational vs. sequential logic
 - Finite state machines (FSMs)

What you will learn?

- Major content
 - Basic parts of a computer (processor, memory, disk, etc.)
 - Principles of computer architecture: CPU datapath and control unit design
 - Assembly language programming in RISC-V
 - Memory hierarchies and design
 - I/O organization and design
- Course goals
 - To learn the organizational structures that determine the capabilities and performance of computer systems
 - To understand the interactions between the computer's architecture and its software
 - To understand cost performance trade-offs

Key Topics

- Introduction (Chapter 1)
 - Basic terms
 - Moore's Law, power wall
 - Core ideas in computer architecture
- Processors (Chapter 2-4)
 - Assembly language (Chapter 2)
 - Computer arithmetic (Chapter 3)
 - Pipelining (Chapter 4)
- Memory (Chapter 5)
- Parallel Processors (Chapter 6)

The content is useful and important

- Computer organization principles are everywhere
 - Embedded computer vs. general-purpose computers:
 - Cellphone, Digital Camera, MP3 music player, Industrial process control
- Complex system design
 - How to partition a problem
 - Functional Spec → Control & Datapath → Physical implementation
 - Modern CAD tools
- Both EEs and CSEs need this information in almost all jobs

Course Information



群名称:CS202-2024S
群 号:528170601

- Course website: Blackboard
- Instructor:
 - Dr. Yuhui BAI (baiyh@sustech.edu.cn)
 - Office: 411 College of Engineering South
 - Office hour: Thursday 14:00-16:00 (by appointment)
- Lecture
 - 10:20-12:10 Mon., Conf. room, Scientific building #1
- Lab
 - 14:00 -15:50 Mon., 511, Lecture Hall #3 (WANG Wei)
 - 14:00 -15:50 Mon., 510, Lecture Hall #3 (WANG Qing)
 - 14:00 -15:50 Tue., 511, Lecture Hall #3 (WANG Wei)
 - 14:00 -15:50 Tue., 510, Lecture Hall #3 (WANG Qing)
 - 14:00 -15:50 Tue., 509, Lecture Hall #3 (BAI Yuhui)
 - 14:00 -15:50 Wed., 510, Lecture Hall #3 (WANG Qing)

Course Information

- Tentative schedule

WEEK	LECTURE	DATE	TOPIC
1	Lecture #1	Feb. 19, 2024	Introductions
2	Lecture #2	Feb. 26, 2024	RISC-V ISAs: Basics
3	Lecture #3	Mar. 4, 2024	RISC-V ISAs: Procedure Call
4	Lecture #4	Mar. 11, 2024	RISC-V ISAs: Addressing
5	Lecture #5	Mar. 18, 2024	Performance
6	Lecture #6	Mar. 25, 2024	Arithmetic
7	Lecture #7	Apr. 1, 2024	Floating Point Arithmetic
8	Lecture #8	Apr. 8, 2024	The Processor
9	Mid-term	Apr. 15, 2024	TBD (Mid-term exam covers Lecture #1—#8)
10	Lecture #9	Apr. 22, 2024	The Pipeline
11	Lecture #10	Apr. 29, 2024	Instruction-Level Parallelism
12	Lecture #11	May 6, 2024	Memory Hierarchy
13	Lecture #12	May. 13, 2024	Memory Hierarchy(cont.)
14	Lecture #13	May. 20, 2024	Memory Hierarchy(cont.)
15	Lecture #14	May. 27, 2024	Parallel Processors
16	Review	Jun. 3, 2024	TBD (Final exam covers Lecture #9—#14)




Course Information

- 30% Mid-term examination
 - tentatively scheduled at week 9, Saturday afternoon
- 30% Final examination
- 30% Lab
- 5~10% Homework
- 0~5% In-class Quiz
- Note:
 - Submit the commitment letter on Blackboard system before March 11st(week4), end of day

Labs

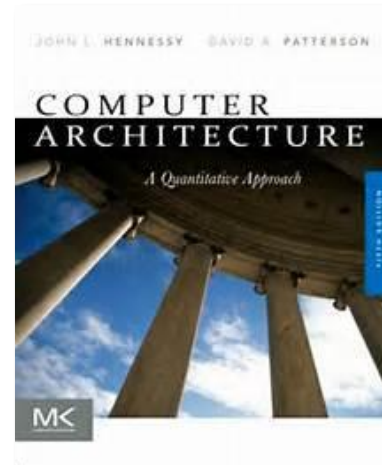
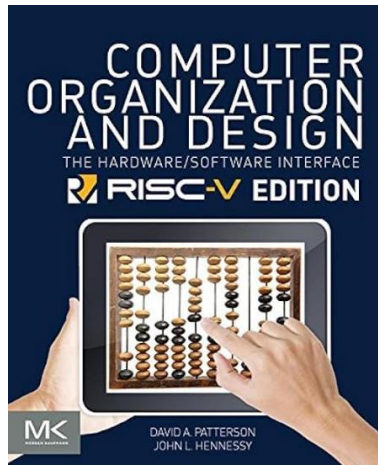
- Labs are a key portion of the class
- Highly recommended to find your partner as soon as possible.

Toolkits used in our Labs

Task	Tool kits
Learn and practice RISC-V (a type of Assembly language)	<p>➤ Rars (rars_27a7c1f)</p> 
Design and implement an CPU	<p>➤ Vivado</p>  <p>➤ FPGA based Development Board EGO1</p> 
Test the CPU with program(s) , both of which are based on RISC-V	<p>➤ Assembler (Rars)</p> <p>➤ Uart Tools</p> <p>➤ Vivado</p> <p>➤ FPGA based Development Board(EGO1)</p>

Course Information

- Textbook:
 - Computer Organization & Design, the Hardware/Software Interface, **RISC-V edition**. D. A. Patterson and J. L. Hennessy
 - The Textbook uses RV64, in class we learn RV32
- Reference book:
 - Computer Architecture - a quantitative approach, Hennessy and Patterson, 5th edition
 - Computer Organization & Design, the Hardware/Software Interface, 5th(MIPS) edition. D. A. Patterson and J. L. Hennessy



Patterson and Hennessy



- Turing award 2017

For pioneering a systematic, quantitative approach to the design and evaluation of computer architectures with enduring impact on the microprocessor industry.



David A. Patterson
Professor of UC Berkeley
Distinguished Engineer at Google

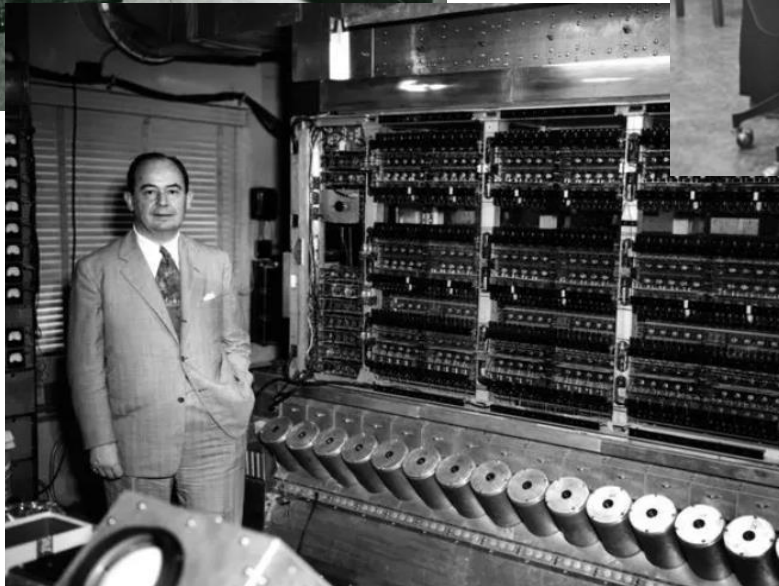
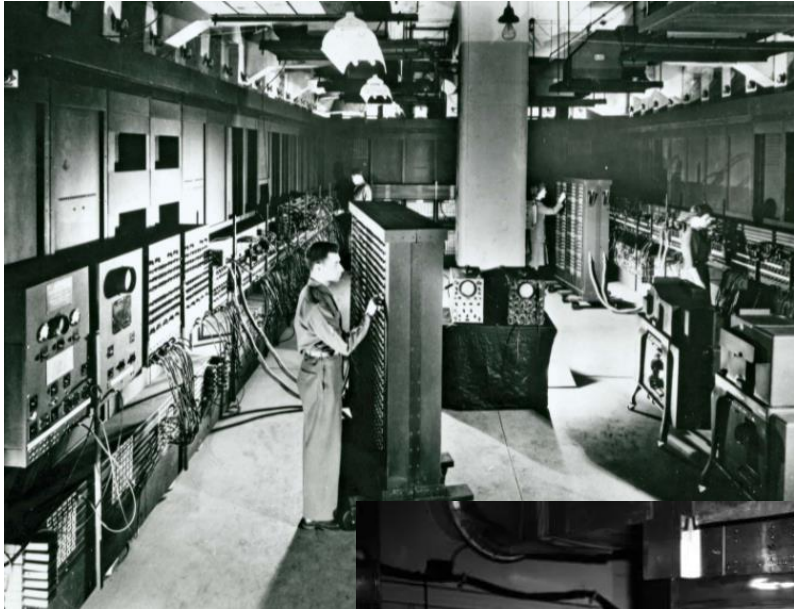


John L. Hennessy
President of Stanford University
Chairman of Alphabet

The Generations of Computers

- First Generation (1940s - 1950s)
 - **Vacuum Tubes**
- Second Generation (1950s - 1960s)
 - **Transistors**
- Third Generation (1960s - 1970s)
 - **Integrated Circuits**
- Fourth/Now Generation (1970s - Present)
 - **Microprocessors**
 - **Artificial Intelligence**

Old School Computers



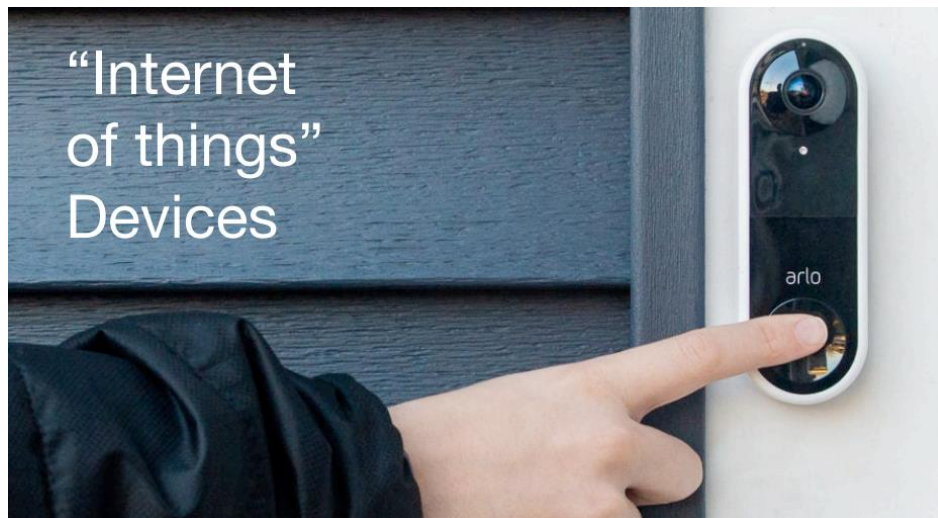
New School Computers

From the Small...



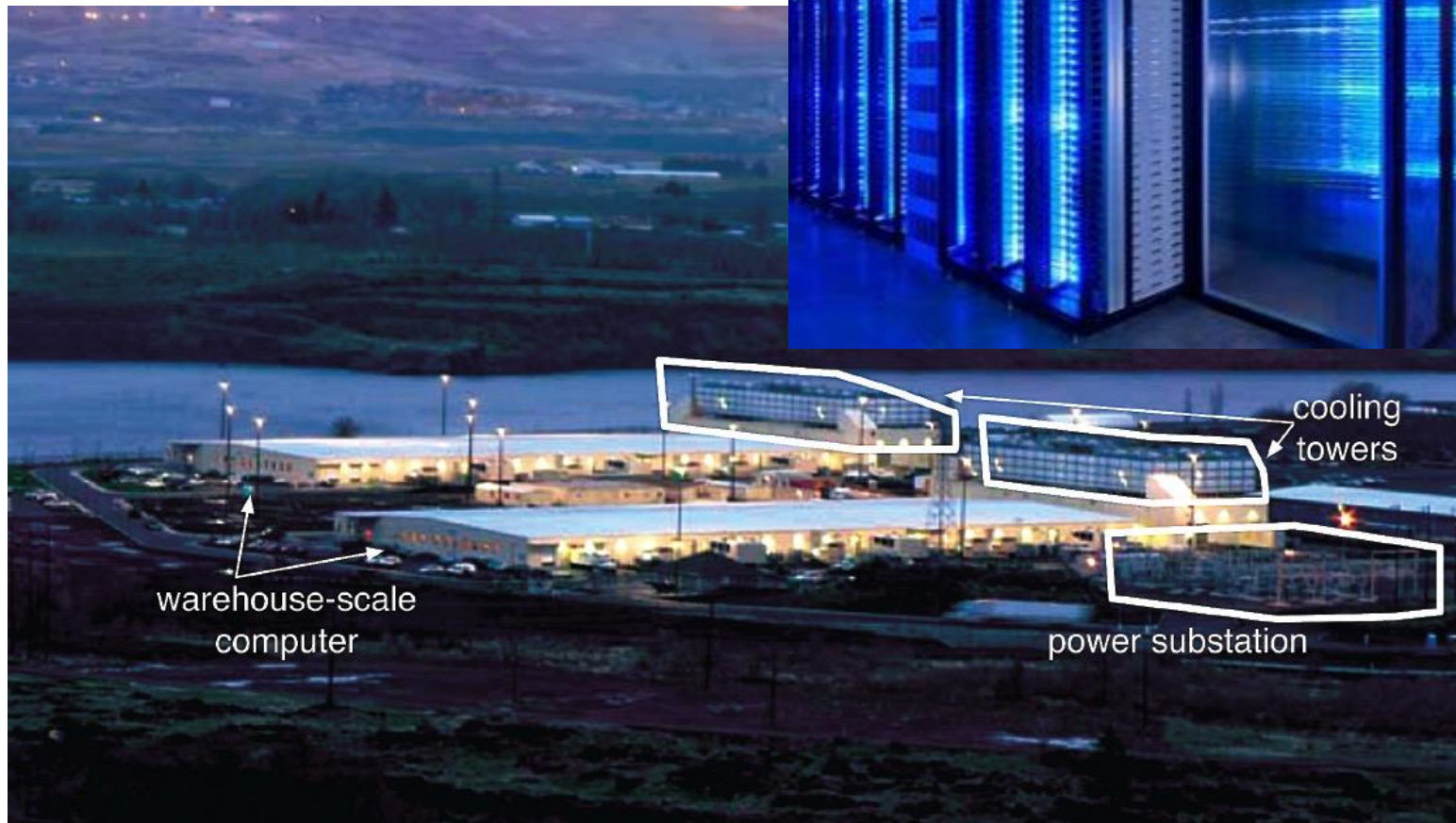
New School Computers

To the very small...



New School Computers

To the big...



Classes of Computers

- Personal computers
 - Computers designed for use by an individual
 - General purpose, variety of software
 - Subject to cost/performance tradeoff
- Server computers
 - Computers used for running larger programs for multiple users, often simultaneously
 - Network based
 - High capacity, performance, reliability
 - Range from small servers to building sized

Classes of Computers

- Supercomputers
 - High-end scientific and engineering calculations
 - Highest capability but represent a small fraction of the overall computer market
 - <https://www.top500.org/lists/top500>
- Embedded computers
 - Hidden as components of systems
 - For running one predetermined application or collection of software.
 - Stringent power/performance/cost constraints

Evolvment of Computers

- Flop: float point operation
- K M G ...?



The PostPC Era

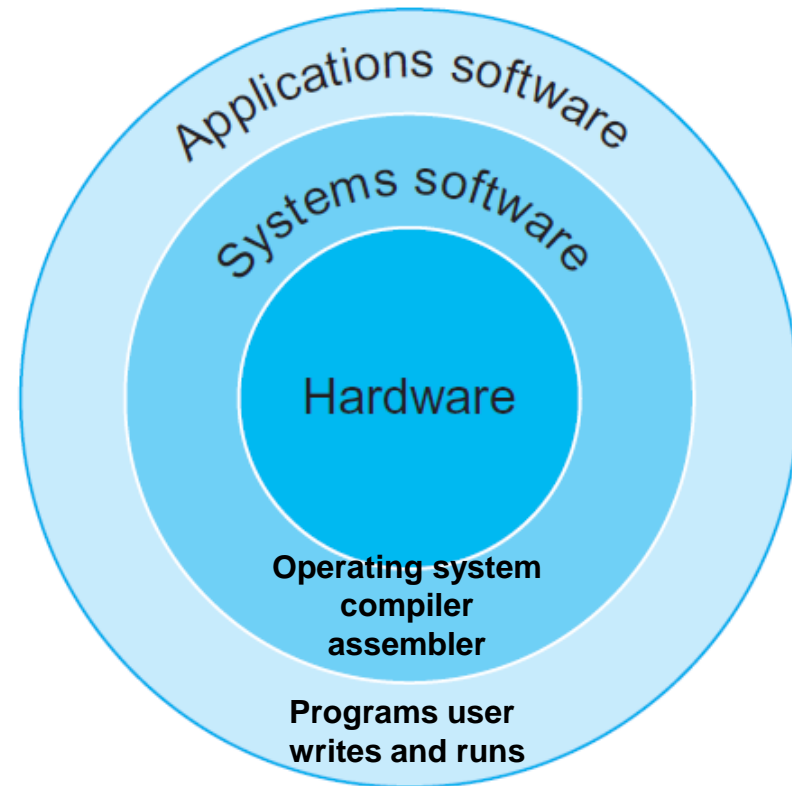
- Personal Mobile Device (PMD)
 - Battery operated
 - Connects to the Internet
 - Hundreds of dollars
 - Smart phones, tablets, electronic glasses
 - Internet of Things
- Cloud computing
 - Warehouse Scale Computers (WSC)
 - Software as a Service (SaaS)
 - Portion of software run on a Personal Mobile Device and a portion run in the Cloud
 - Data Centers (Amazon and Google)

New Computer Architecture for AI era

- AI and big data requires new computer architecture
 - More Suitable for deep learning
 - High requirement on parallel
 - Low energy
- From CPU to GPU, TPU...
- AI Chips
 - SmartPhone
 - AlphaGo
 - Autonomous Driving
 - ChatGPT
- Dozens of companies dive in this area:
 - Google, NVIDIA, 华为, 百度, 地平线, 寒武纪, 燧原
 - OpenAI...

The Concept of a Computer

- Application software
 - Written in high-level language
- System software
 - Compiler: translates HLL code to machine code
- Operating System:
 - Handling input/output
 - Managing memory and storage
 - Scheduling tasks & sharing resources
- Hardware
 - Processor, memory, I/O controllers



Levels of Program Code

- C program compiled into assembly language and then assembled into binary machine language.
- High-level language
 - Level of abstraction closer to problem domain
 - Provides for productivity and portability
- Assembly language
 - Textual representation of instructions
- Machine language
 - Hardware representation
 - Binary digits (bits)
 - Encoded instructions and data

High-level
language
program
(in C)

```
swap(size_t v[], size_t k)
{
    size_t temp;
    temp = v[k];
    v[k] = v[k+1];
    v[k+1] = temp;
}
```

Compiler

Assembly
language
program
(for RISC-V)

```
swap:
    slli x6, x11, 3
    add  x6, x10, x6
    ld   x5, 0(x6)
    ld   x7, 8(x6)
    sd   x7, 0(x6)
    sd   x5, 8(x6)
    jalr x0, 0(x1)
```

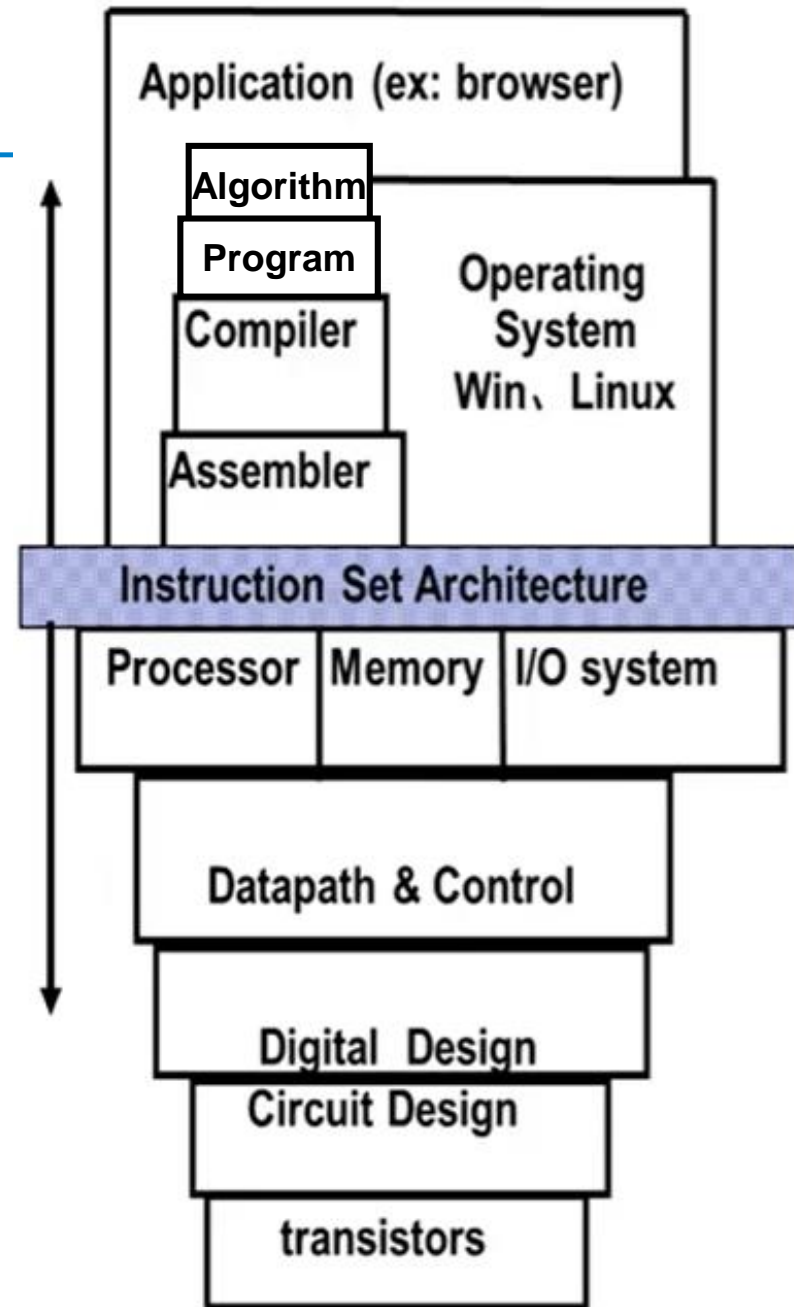
Assembler

Binary machine
language
program
(for RISC-V)

```
00000000001101011001001100010011
00000000011001010000001100110011
00000000000000110011001010000011
00000000100000110011001110000011
00000000011100110011000000100011
00000000010100110011010000100011
0000000000000000100000001100111
```

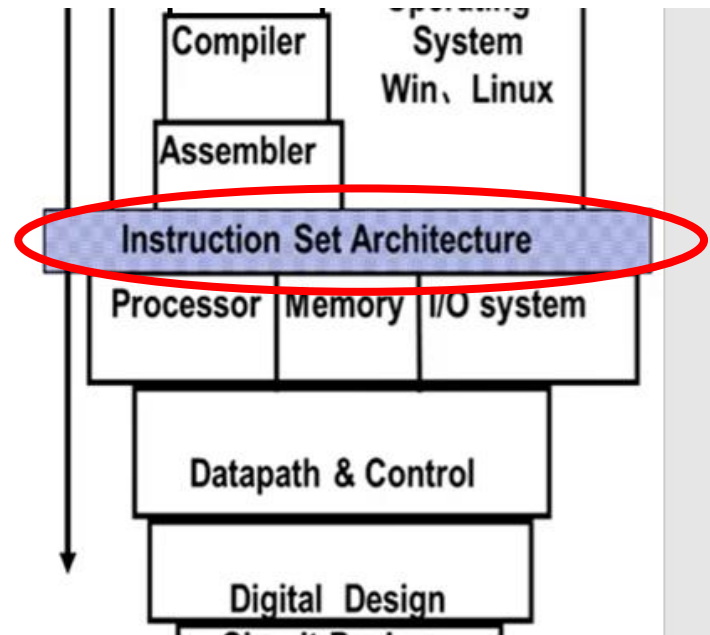
Abstractions

- Abstraction helps us deal with complexity
 - Hides lower-level details
- Instruction Set Architecture (ISA) or Computer Architecture
 - The hardware/software interface
 - Includes instructions, registers, memory access, I/O, and so on
- Operating system hides details of doing I/O, allocating memory from programmers



Instruction Set Architecture (ISA)

- A set of assembly language instructions (ISA) provides a link between software and hardware.
- Given an instruction set, software programmers and hardware engineers work more or less independently.
- Common types of ISA: RISC, CISC
- Examples:
 - IBM370/X86 (CISC)
 - **RISC-V** (RISC)
 - MIPS (RISC)
 - ARM (RISC)



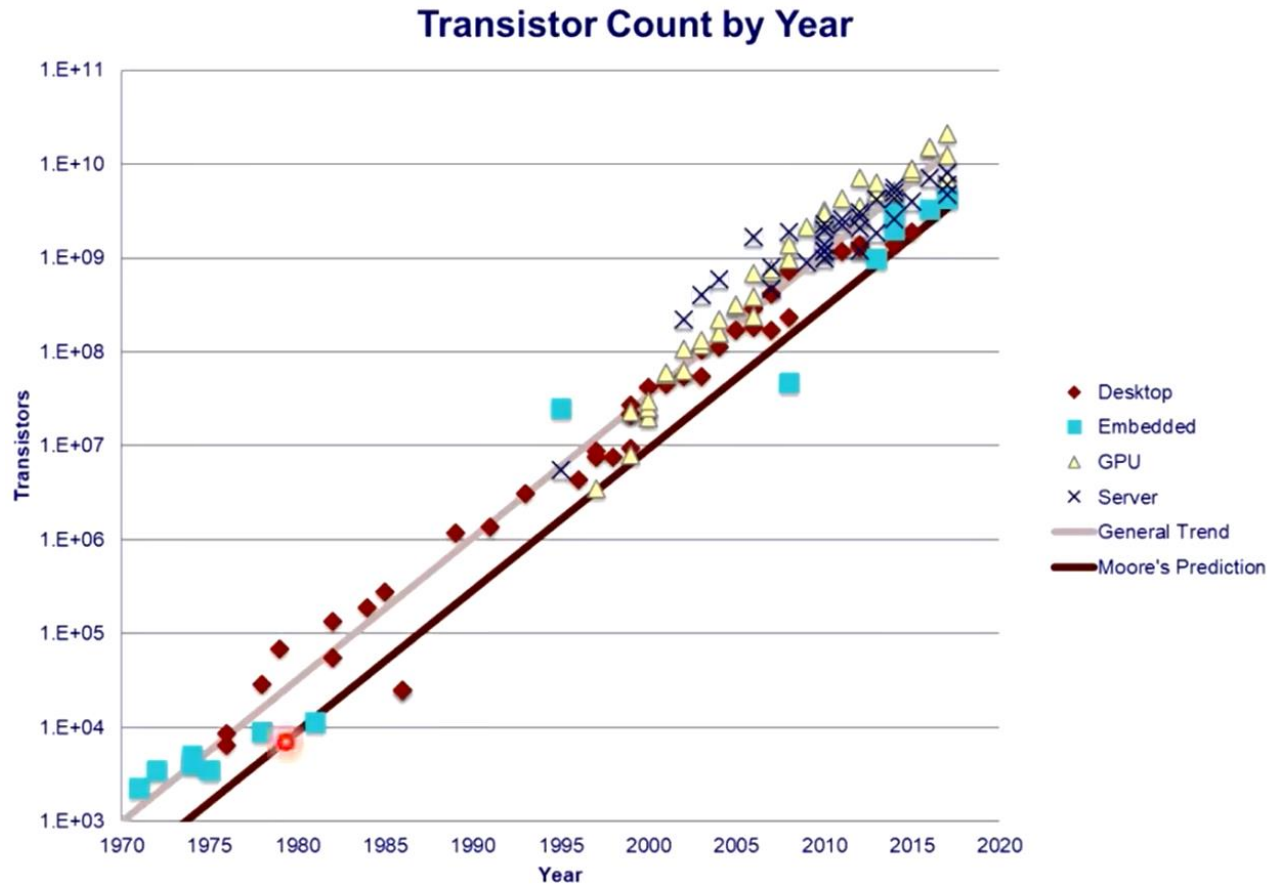
Eight Great Ideas

- Design for **Moore's Law**
- Use **abstraction** to simplify design
- Make the **common case fast**
- Performance via **parallelism**
- Performance via **pipelining**
- Performance via **prediction**
- **Hierarchy** of memories
- **Dependability** via redundancy



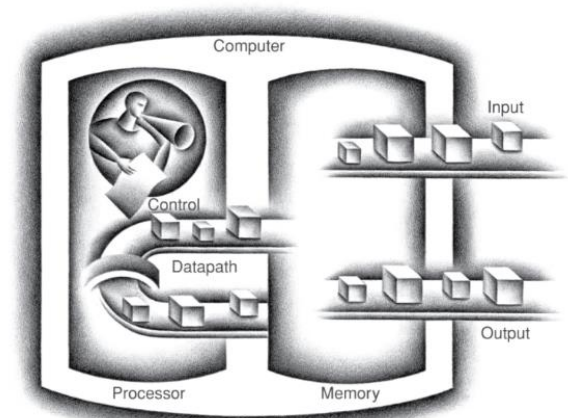
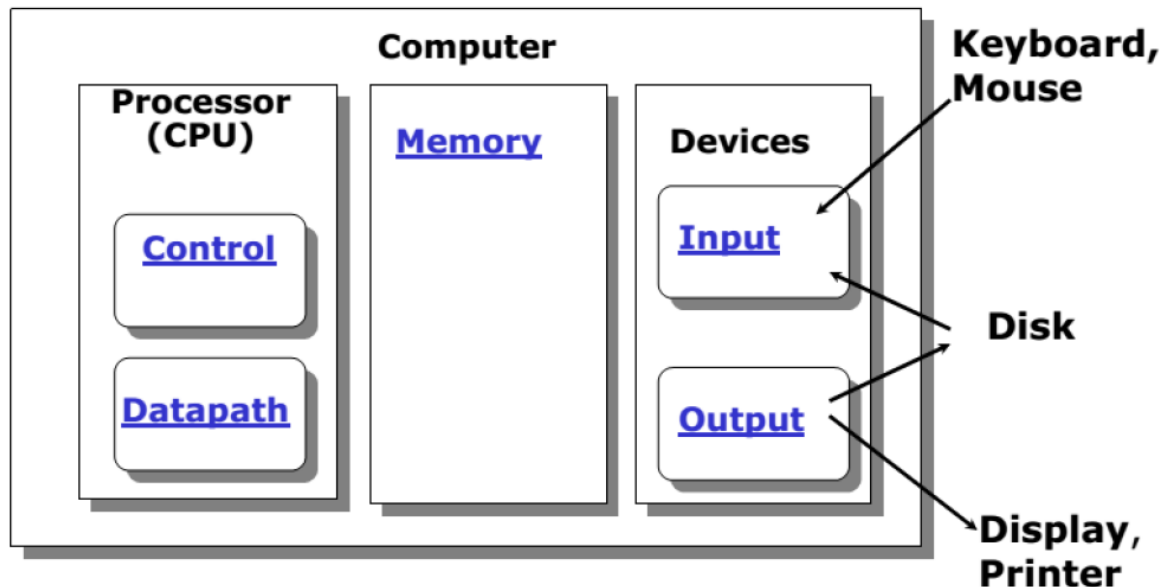
Moore's law

- **Prediction** made by Gordon Moore in 1965 that the number of transistors per silicon chip doubles every 18 to 24 months.
- Is moore's law still valid?



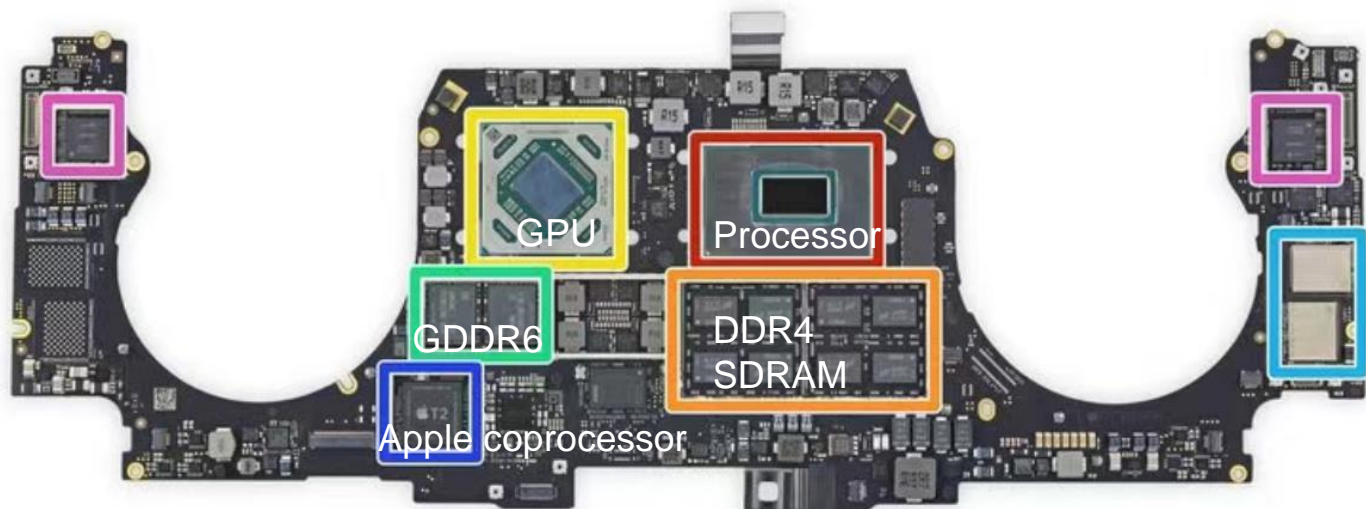
Components of a Computer

- Same components for all kinds of computer:
 - Input Device, Output Device, Memory, Processor (Control, Datapath)
 - Von Neumann Architecture vs. Harvard Architecture



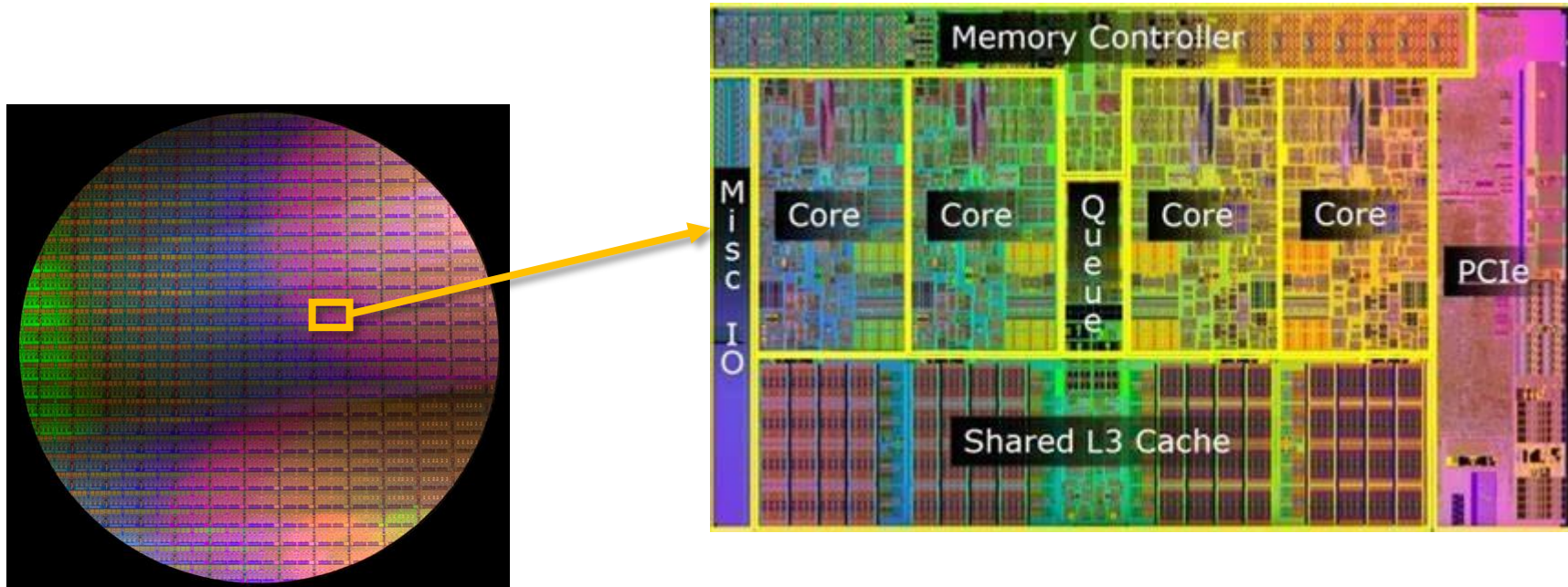
Teardown of MacBook

- 16" LED-backlit IPS Retina display
- Keyboard and Touch Bar
- 2.6 GHz 6-core Intel Core i7
- 16 GB of 2666 MHz DDR4 SDRAM
- 512 GB SSD
- 100 Watt-hour battery
- Speaker and microphone



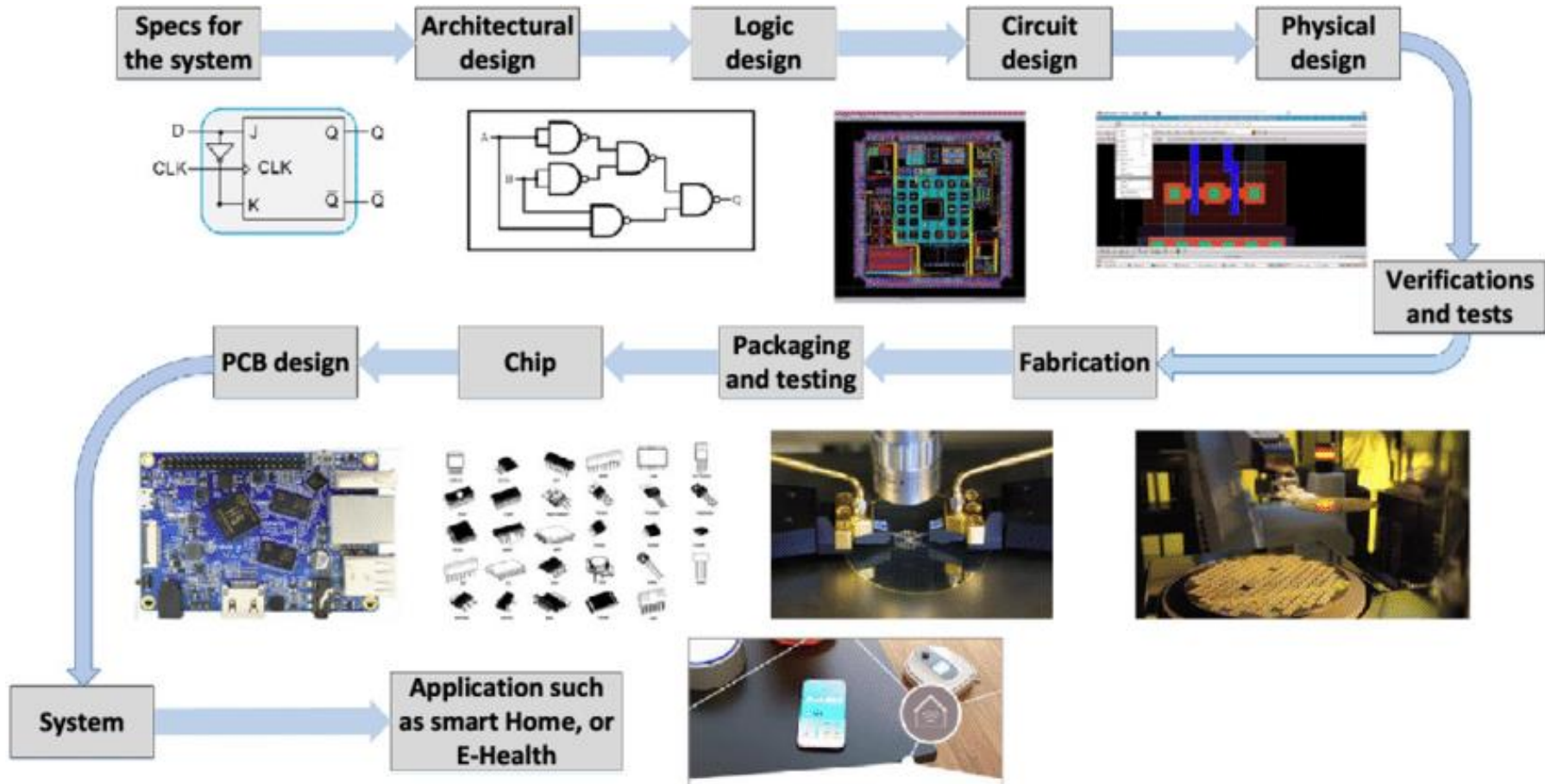
Inside the Processors

- Datapath:
 - performs operations on data
- Control:
 - sequences datapath, memory, I/O
- Cache memory:
 - small fast Static RAM memory for immediate access to data



Intel's Core i7 wafer and Die map

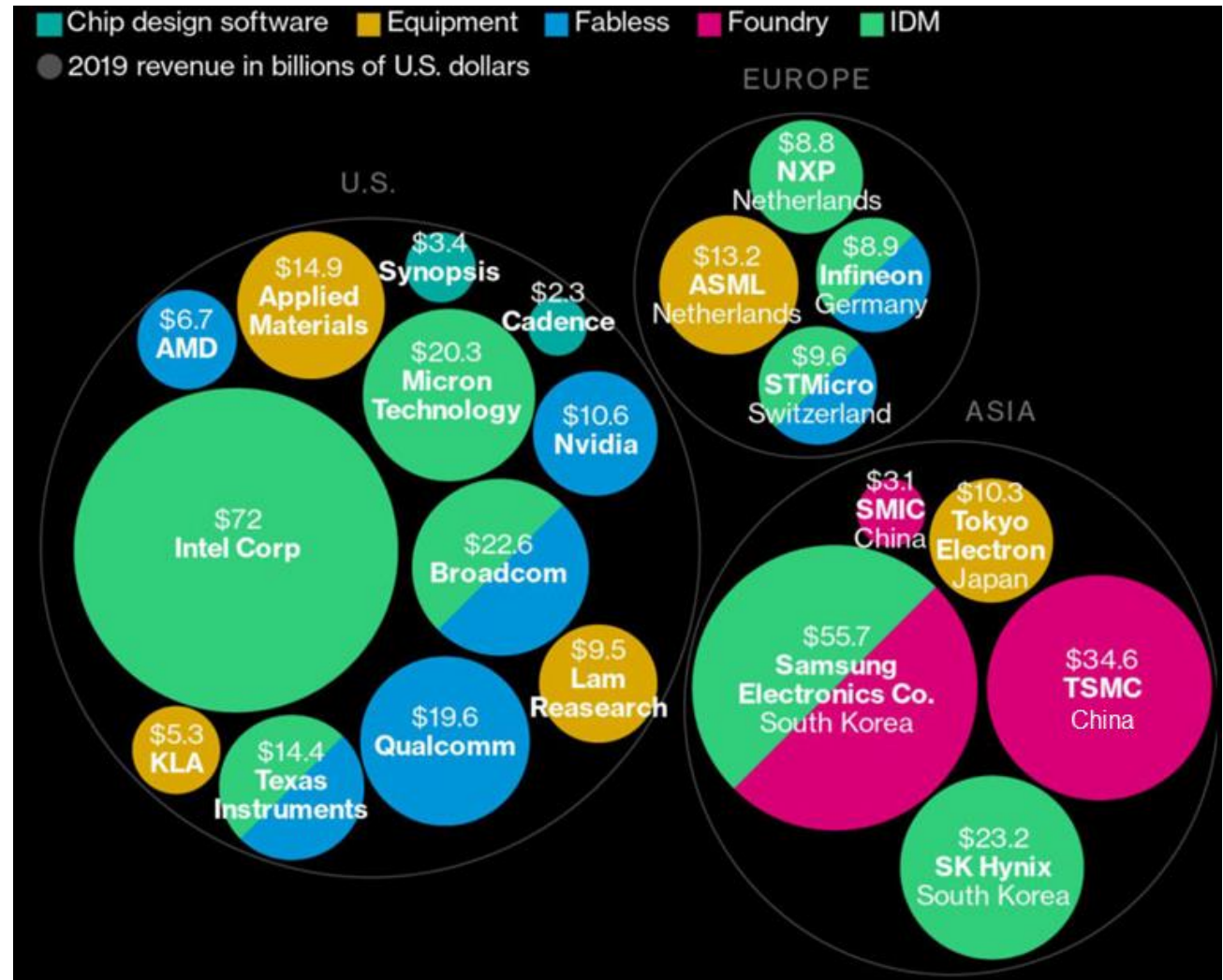
Semiconductor manufacturing process chain



Chip Industry Choke Points

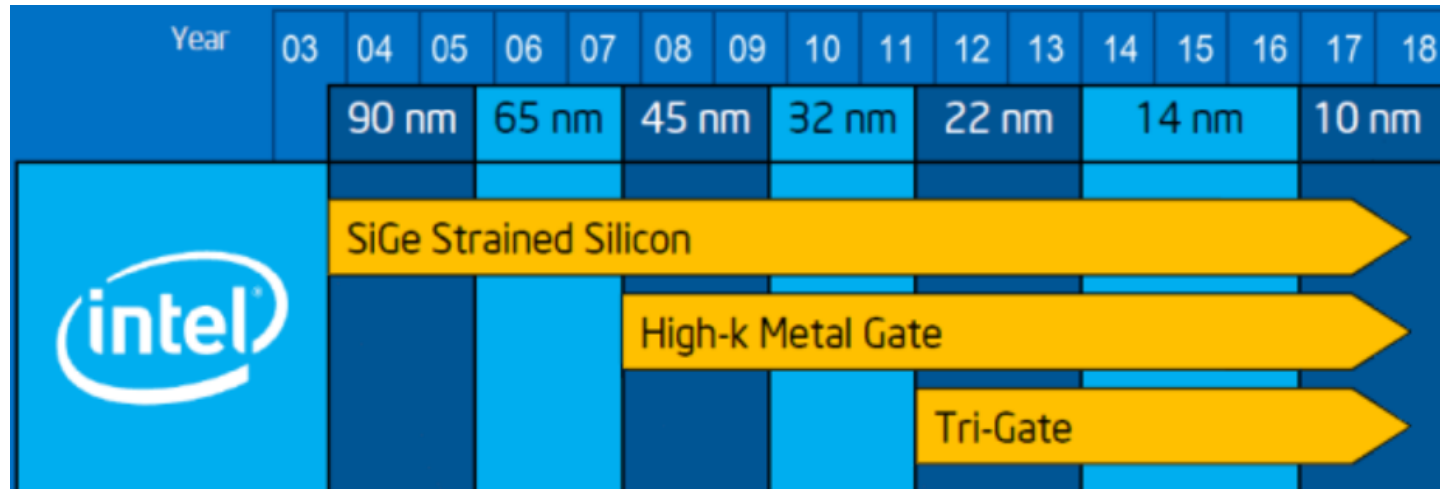
- Key players in chip industry

- Intel
- AMD
- Qualcomm
- Samsung
- TSMC
- Broadcom
- Nvidia
- ASML
- ...



Processor Technology Trends

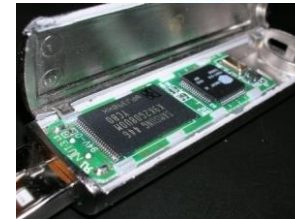
- Shrinking of transistor sizes: 90nm(2004) -> 45nm(2008) -> 22nm(2012) -> 10nm(2017) ...



- Transistor density increases by 35% per year and die size increases by 10-20% per year... functionality improvements!
- Transistor speed improves linearly with size (complex equation involving voltages, resistances, capacitances)
- Wire delays do not scale down at the same rate as transistor delays

Storage

- Volatile main memory
 - Loses instructions and data when power off
- Non-volatile secondary memory
 - Magnetic disk
 - Flash memory
 - Optical disk (CDROM, DVD)

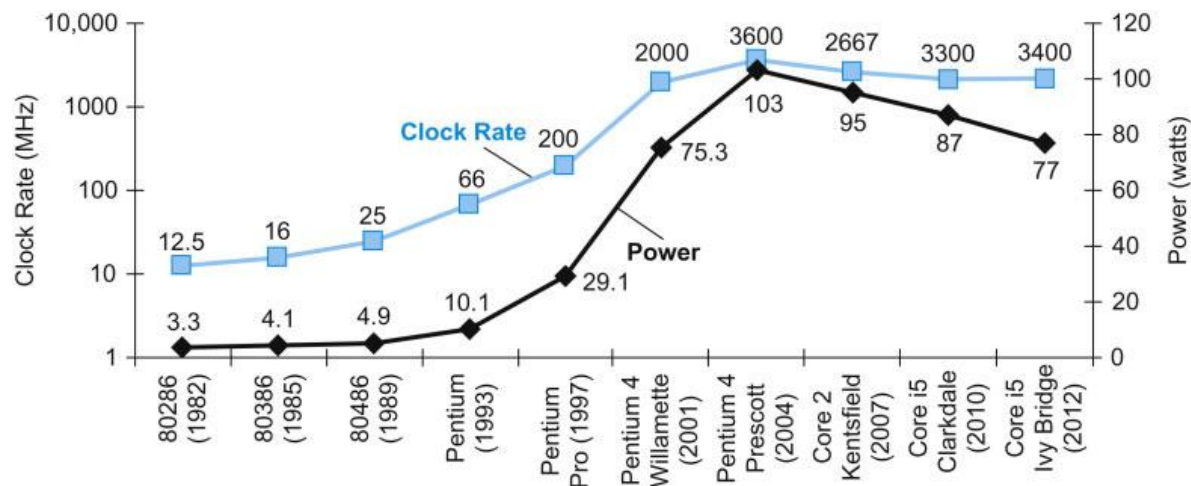


Memory and I/O Technology Trends

- DRAM density increases by 40-60% per year, latency has reduced by 33% in 10 years (the memory wall!), bandwidth improves twice as fast as latency decreases
- Disk density improves by 100% every year, latency improvement similar to DRAM
- Networks: primary focus on bandwidth; 10Mb → 100Mb in 10 years; 100Mb → 1Gb in 5 years

Power Consumption Trends

- Dynamic power
- Voltage and frequency are somewhat constant now, while capacitance per transistor is decreasing and number of transistors (activity) is increasing
- Leakage power is also rising (function of #trans and voltage)



“龙芯杯”全国大学生计算机系统能力培养大赛

NSCSCC

- NSCSCC（大赛官网：nscsc.com）
 - 由教育部高等学校计算机类专业教学指导委员会和系统能力培养研究专家组共同发起，以学科竞赛推动专业建设和计算机领域创新人才培养体系改革、培育我国高端芯片及核心系统的技术突破与产业化后备人才为目标，面向高校大学生举办的全国性大赛。
- 比赛分**团队赛**和**个人赛**共4个赛道：MIPS/LoongArch团队赛、MIPS/LoongArch个人赛。
- MIPS/LoongArch团队赛
 - 开发支持32位MIPS或LA32R基准指令集的微型计算机系统。
 - 初赛成绩 = 功能测试 得分 + 性能测试 得分
 - 决赛成绩 = $40\% \times \text{基准测试程序 跑分} + 20\% \times \text{自定义指令 的实现得分} + 40\% \times \text{系统展示及 答辩}$
 - 加分项：启动操作系统、实现加速器、设计可演示的应用等
- MIPS/LoongArch个人赛
 - 开发支持32位MIPS或LA32R基准指令集的简易计算机系统。
 - 完成三级功能测试（最多22条指令），支持SRAM、UART，运行监控程序。
 - 初赛成绩 = 功能测试 得分 + 性能测试 得分
 - 决赛成绩 = $70\% \times \text{基准测试程序 跑分} + 30\% \times \text{现场编程题 得分}$
- 比赛重要时间点（以2023年为例）

