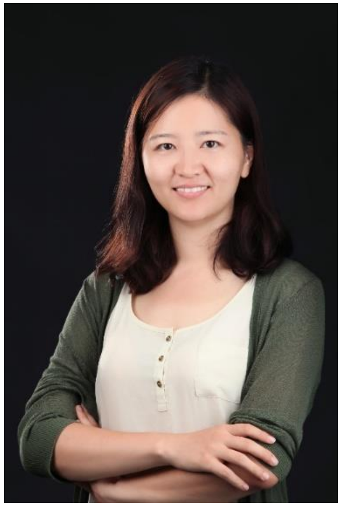
CS202: COMPUTER ORGANIZATION

Lecture 1

Course Introduction

张 进



- 2000年-2004年，清华大学电子系学士
- 2004年-2006年，清华大学电子系硕士
- 2006年-2009年，香港科技大学计算机系博士
- 2009年-2014年，香港科技大学计算机系研究助理教授
- 2014年至今，南方科技大学计算机系助理教授、副教授/博士生导师

研究方向：移动计算，智能物联网，无线感知，可穿戴设备与移动健康

- ➢ 在IOTJ、TWC、TMC、TPDS、Mobicom、Infocom、MobiHoc、ICDCS等国际顶级期刊会议发表论文90余篇
- ➢ 近五年承担和参与国家、广东省和深圳市科研项目10余项，包括：国家自然科学基金（青年，面上），科技部重点研发计划，广东省重点实验室，广东省普通高校重点实验室，广东省创新团队，深圳市工程实验室、深圳市学科布局项目
- ➢ 带领的研究团队：研究助理教授1名、博士10名、硕士6名、工程师2名、本科生20余名

Course Introduction

- Today’s topics:
  - Why computer organization is important
  - Course information
  - Background and computer abstraction

Why to Learn Computer Organization?

- Embarrassing if you are a student in CS and can’t make sense of the following terms: DRAM, 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 MIPS
  - 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

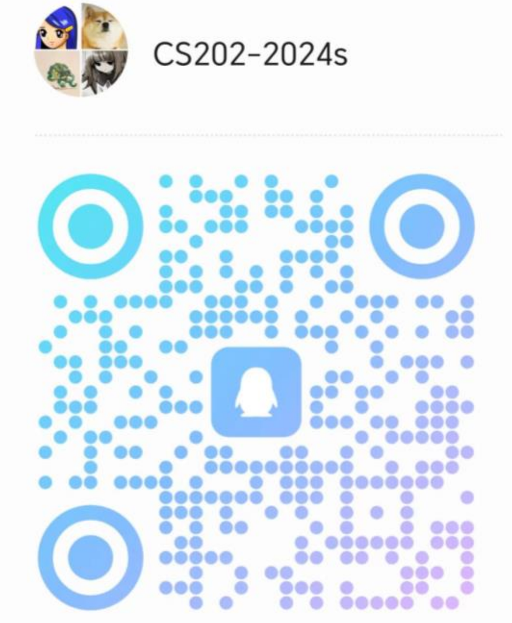
- Introduction (Chapter 1)
  - Basic terms
  - Moore’s Law, power wall
  - Core ideas in computer architecture
- Processors (Chapter 2-4)
  - Assembly language (Chapter 2)
  - Computer arithmetic (Chapter 3)
  - Pipelining (Chapter 4)
- Memory (Chapter 5)
- Parallel Processors (Chapter 6)

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

- Course Materials:
  - Blackboard system: Computer Organization Spring 2024
- QQ group: 628058404
- Lectures:
  - Friday 14:00-15:50 (3rd Teaching Building 305)
- Instructor:
  - Prof. Jin Zhang (zhangj4@sustech.edu.cn)
  - Office: 414, CoE South
  - Office hour: Monday 14:00-16:00



CS202-2024s

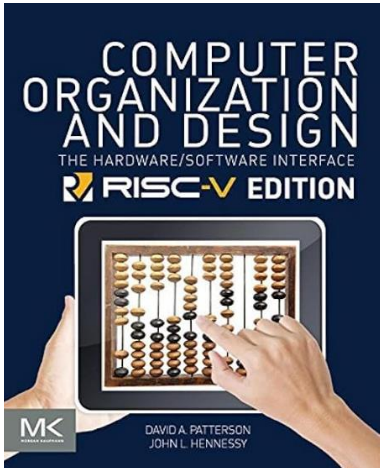
Course Information

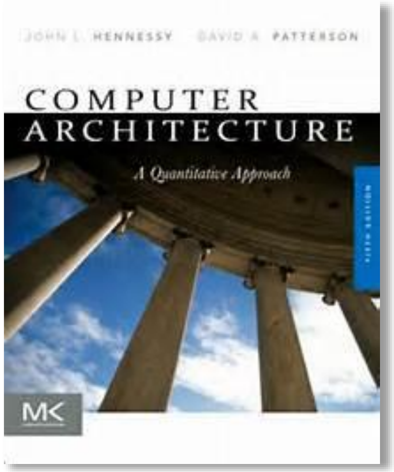
- Tentative schedule

WEEK	LECTURE	DATE	TOPIC
1	Lecture #1	Feb. 23, 2024	Introductions
2	Lecture #2	Mar. 1, 2024	RISC-V ISAs: Basics
3	Lecture #3	Mar. 8, 2024	RISC-V ISAs: Procedure Call
4	Lecture #4	Mar. 15, 2024	RISC-V ISAs: Addressing
5	Lecture #5	Mar. 22, 2024	Performance
6	Lecture #6	Mar. 29, 2024	Arithmetic
7	Lecture #7	Apr. 7, 2024	Floating Point Arithmetic
8	Lecture #8	Apr. 12, 2024	The Processor
9	Mid-term	Apr. 19, 2024	TBD (Mid-term exam covers Lecture #1—#8)
10	Lecture #9	Apr. 26, 2024	The Pipeline
11	Break	May 3, 2024	Labors Day Holiday
12	Lecture #10	May 10, 2024	Instruction-Level Parallelism
13	Lecture #11	May. 17, 2024	Memory Hierarchy
14	Lecture #12	May. 24, 2024	Memory Hierarchy(cont.)
15	Lecture #13	May. 31, 2024	Memory Hierarchy(cont.)
16	Lecture #14	Jun. 7, 2024	Parallel Processors

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 AND DESIGN
THE HARDWARE/SOFTWARE INTERFACE
RISC-V EDITION
MK
DAVID A. PATTERSON
JOHN 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

Recommended Reading

- Code: The Hidden Language of Computer Hardware and Software, Charles Petzold, 2000.《编码的奥秘》，《编码——隐匿在计算机软硬件背后》

- Computer Systems: A Programmer’s Perspective, R. E. Bryant, D. R. O’Hallaron, 3rd Edition, Pearson, 2015.《深入理解计算机系统》

Other Readings

- 《浪潮之巅 》《数学之美》吴军
- 《创新者：一群技术狂人和鬼才程序员如何改变世界》沃尔特·艾萨克森 （《史蒂夫·乔布斯传》、《爱因斯坦传》、《本杰明·富兰克林传》）
- 《沸腾十五年》《沸腾新十年》林军 《激荡四十年》
- 《图灵和ACM图灵奖》
- 《黑客与画家》硅谷创业之父Paul Graham
- 《人月神话》《信息简史》
- KK三部曲 《失控》《科技想要什么》 《必然》

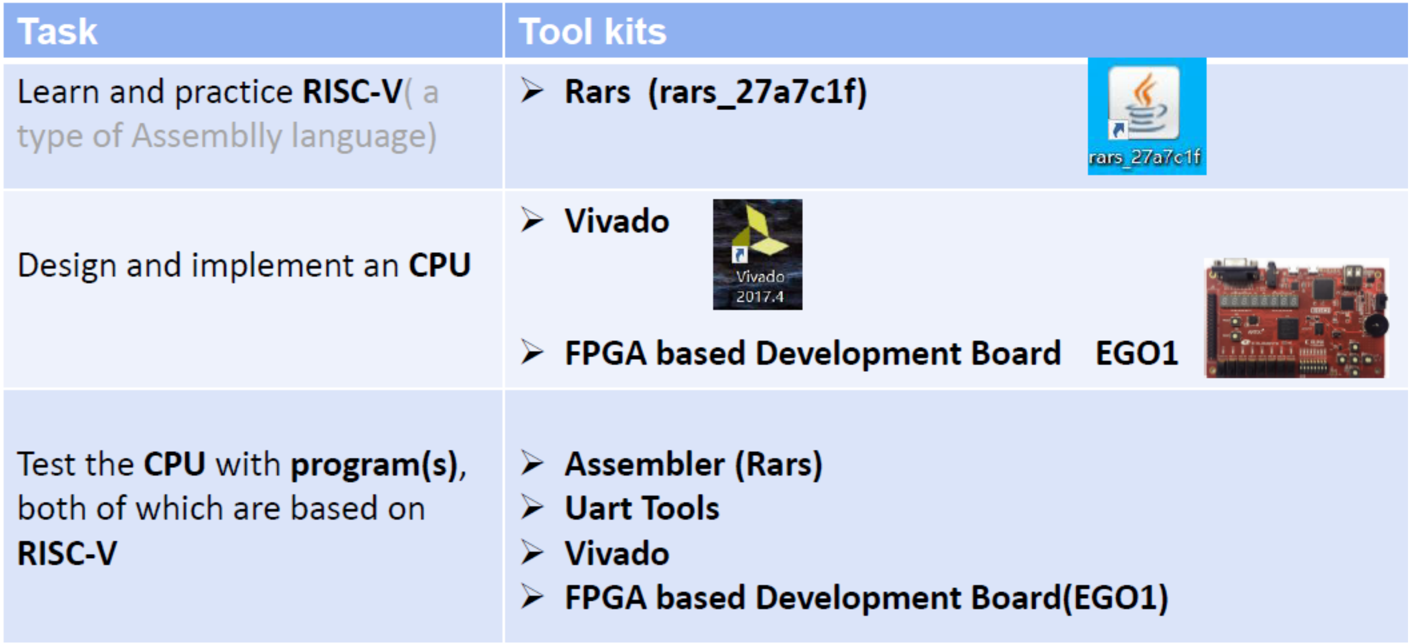
Course Information

- Assessment:
  - Theoretical part:
    - Final exam (~30%)
    - Midterm exam (~30%)
    - Assignments and Attendance (~10%)
  - Lab part (~30%)
- Please Submit on time
  - all the assignments and reports should be submitted in the BlackBoard system, late submission will not be accepted in the system.

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 Assemblly language)	➢ **Rars (rars_27a7c1f)**
Design and implement an **CPU**	➢ **Vivado** ➢ **FPGA based Development Board EGO1**
Test the **CPU** with **program(s)**, both of which are based on **RISC-V**	➢ **Assembler (Rars)** ➢ **Uart Tools** ➢ **Vivado** ➢ **FPGA based Development Board(EGO1)**

rars_27a7c1f
Vivado
2017.4

Rules about Plagiarism

- No Plagiarism is allowed
  - If plagiarism on homework or project is found for the first time, the plagiaristic part is graded as 0 and warning is given to the students
  - If plagiarism is found for the second time, the course is graded as 0
  - For lab/project report, any sentence that is copied from other paper or article should cites the original source as the reference, otherwise, the report is considered as plagiarism
- Submit the commitment letter on Blackboard system before homework #1

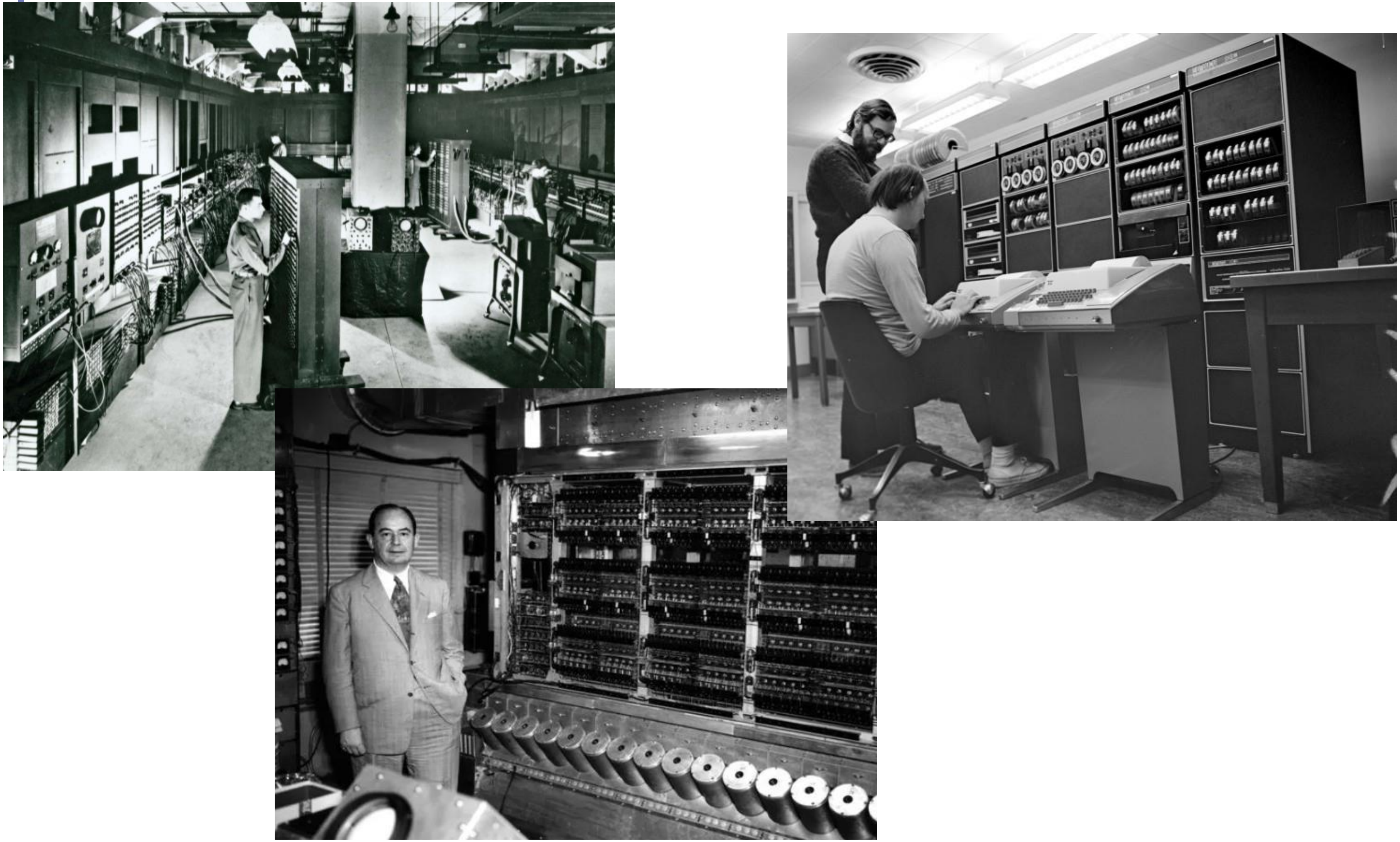
Tips for Attending the Lecture

- To get the best use of lecture
  - interactive
  - ask whenever you have question, interrupt whenever you want
  - ask immediately after the class if you are shy
  - give me suggestions and feedback frequently
- Get the main idea in class, read the details after class

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

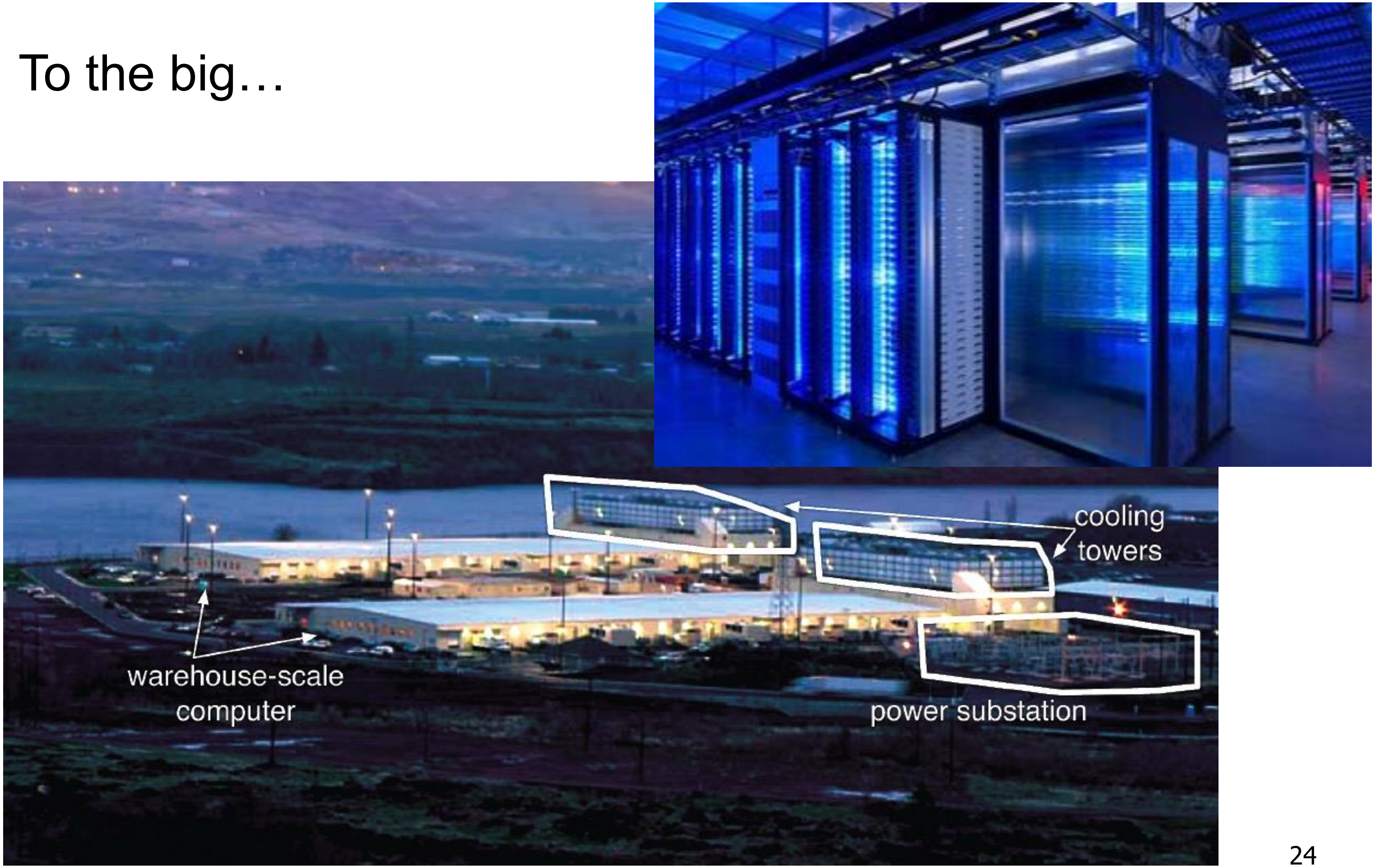
From the Small…

New School Computers

To the very small...

New School Computers

To the big…



Classes of Computers

- Personal computers
  - General purpose, variety of software
  - Subject to cost/performance tradeoff
- Server computers
  - 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

- Embedded computers
  - Hidden as components of systems
  - Stringent power/performance/cost constraints

Evolvement of Computers

Petaflop
Teraflop
Gigaflop

Flop: float point operation

K M G …?

The PostPC Era

- 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
  - Amazon and Google
- Data centers
  - Millions of computers connected by off-the-shelf networking devices

Google Data Centers

The PostPC Era

- Personal Mobile Device (PMD)
  - Battery operated
  - Connects to the Internet
  - Hundreds of dollars
  - Smart phones, tablets, electronic glasses

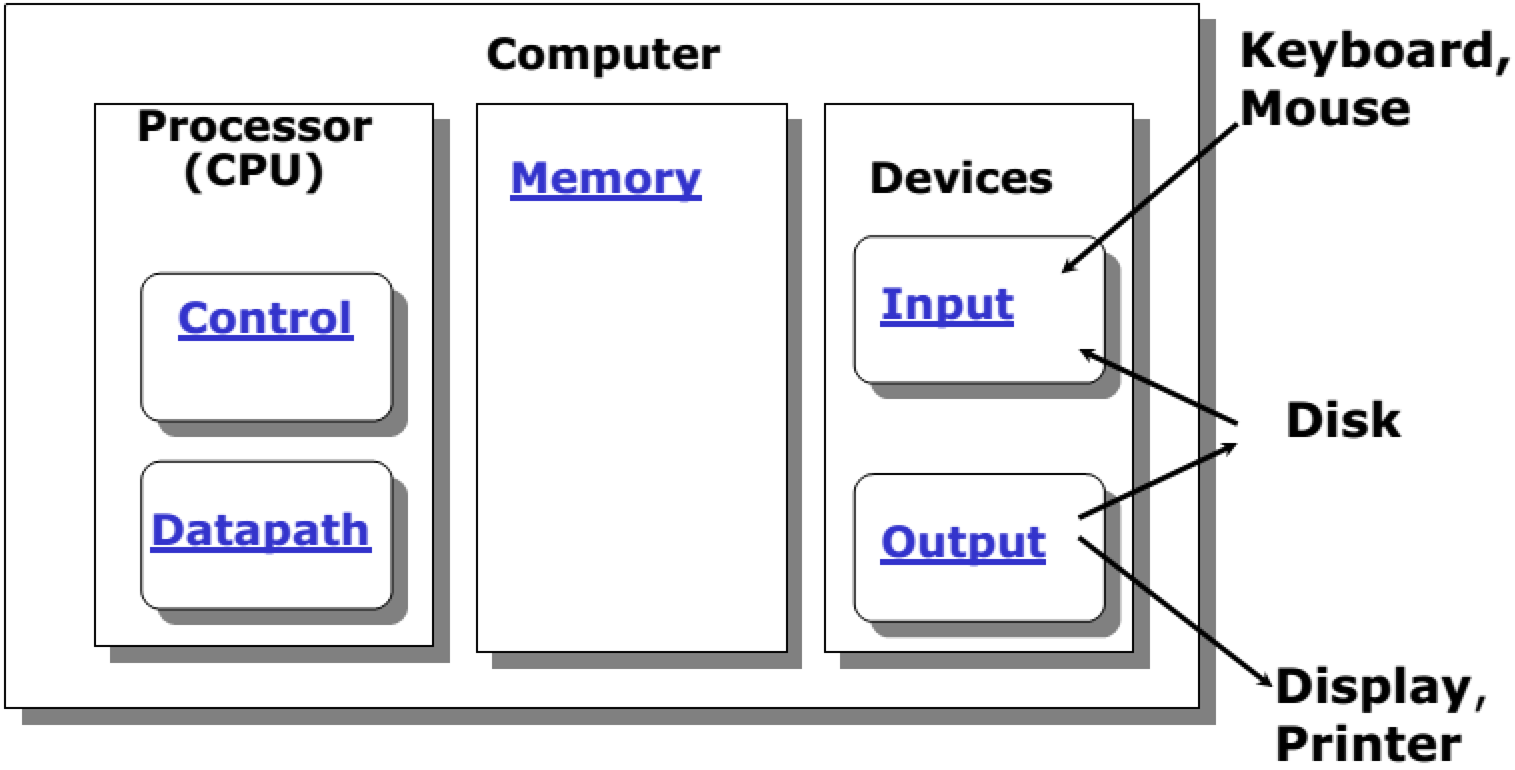
Smart Devices

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、地平线、寒武纪、燧原
  - OpenAI

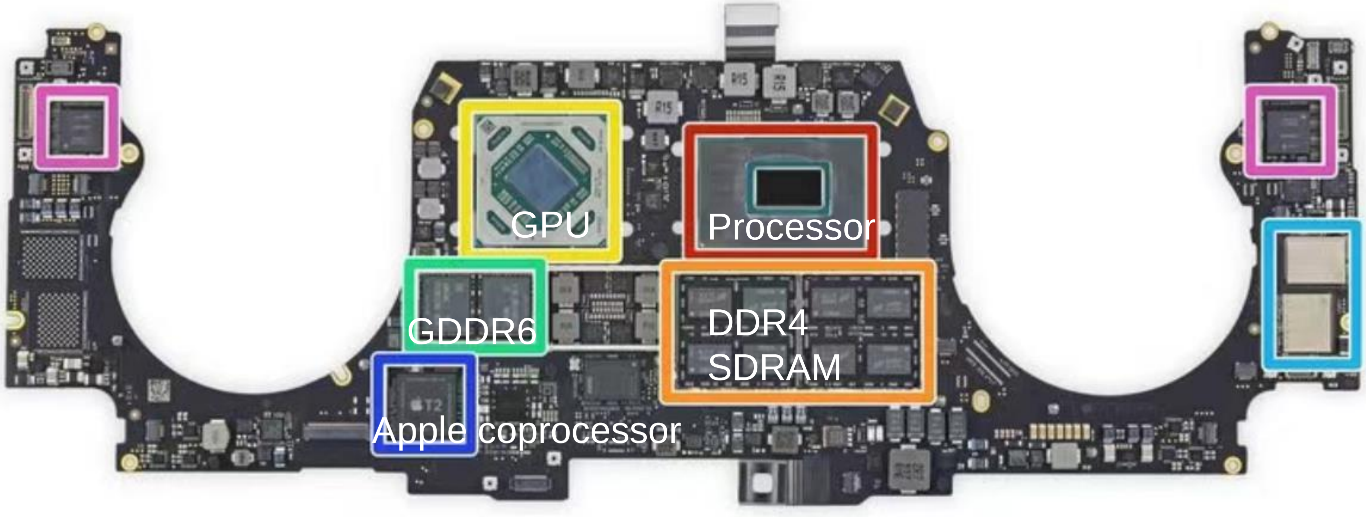
Components of a Computer

- Same components for all kinds of computer:
  - Input Device, Output Device, Memory, Processor (Control, Datapath)



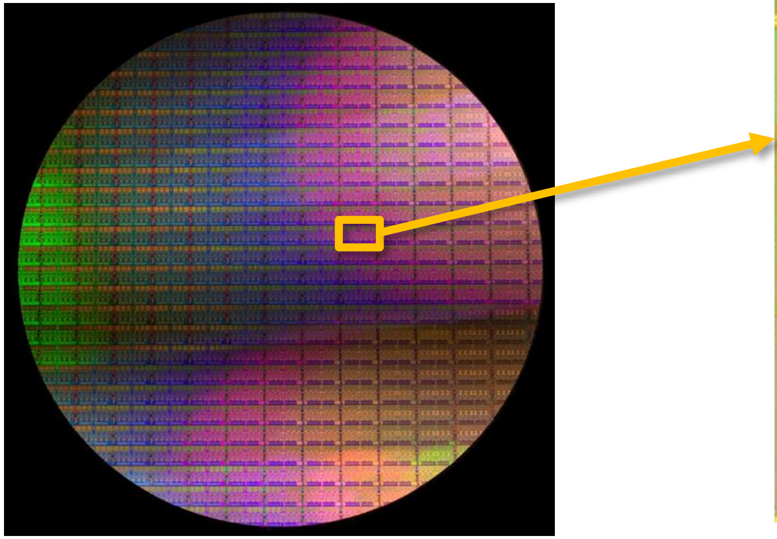
Teardown of MacBook

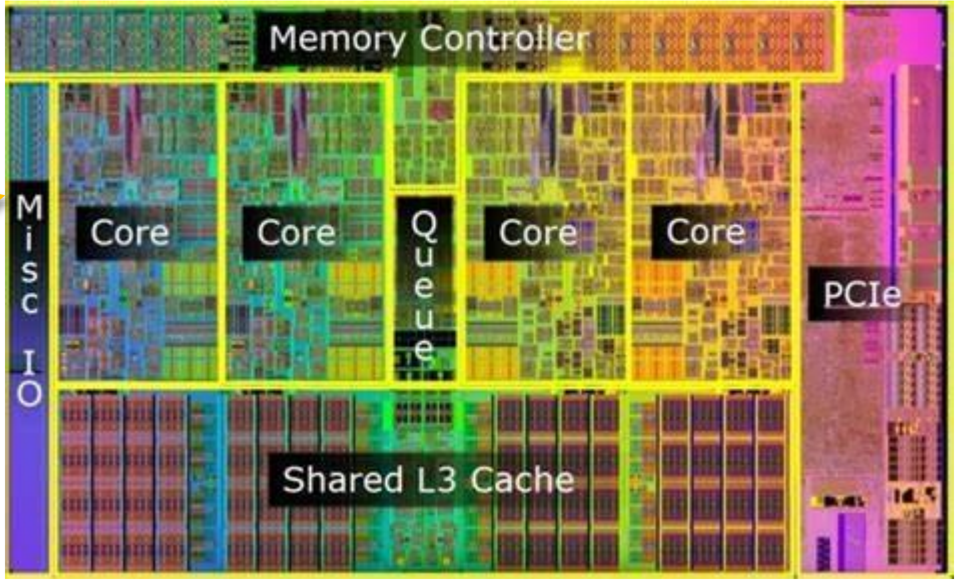
- 16" LED-backlit IPS Retina display
- Keyboard and Touch Bar
- 2.6 GHz 6-core Intel Core i7
- 16 GB of 2666 MHz DDR4 SDRAM
- 512 GB SSD
- 100 Watt-hour battery
- Speaker and microphone



Inside the Processors

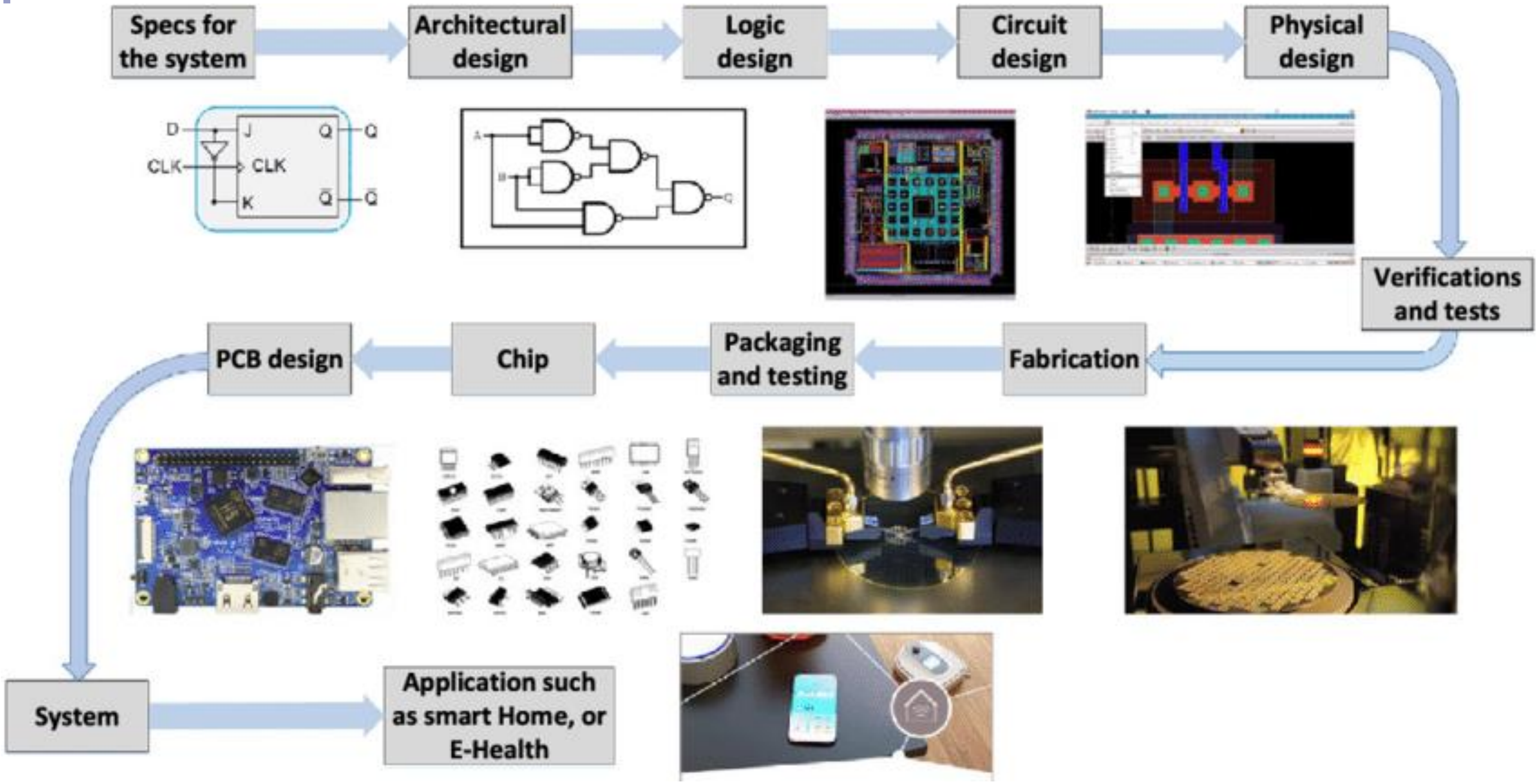
- Datapath:
  - performs operations on data
- Control:
  - sequences datapath, memory, I/O
- Cache memory:
  - small fast Static RAM memory for immediate access to data





Intel's Core i7 wafer and Die map

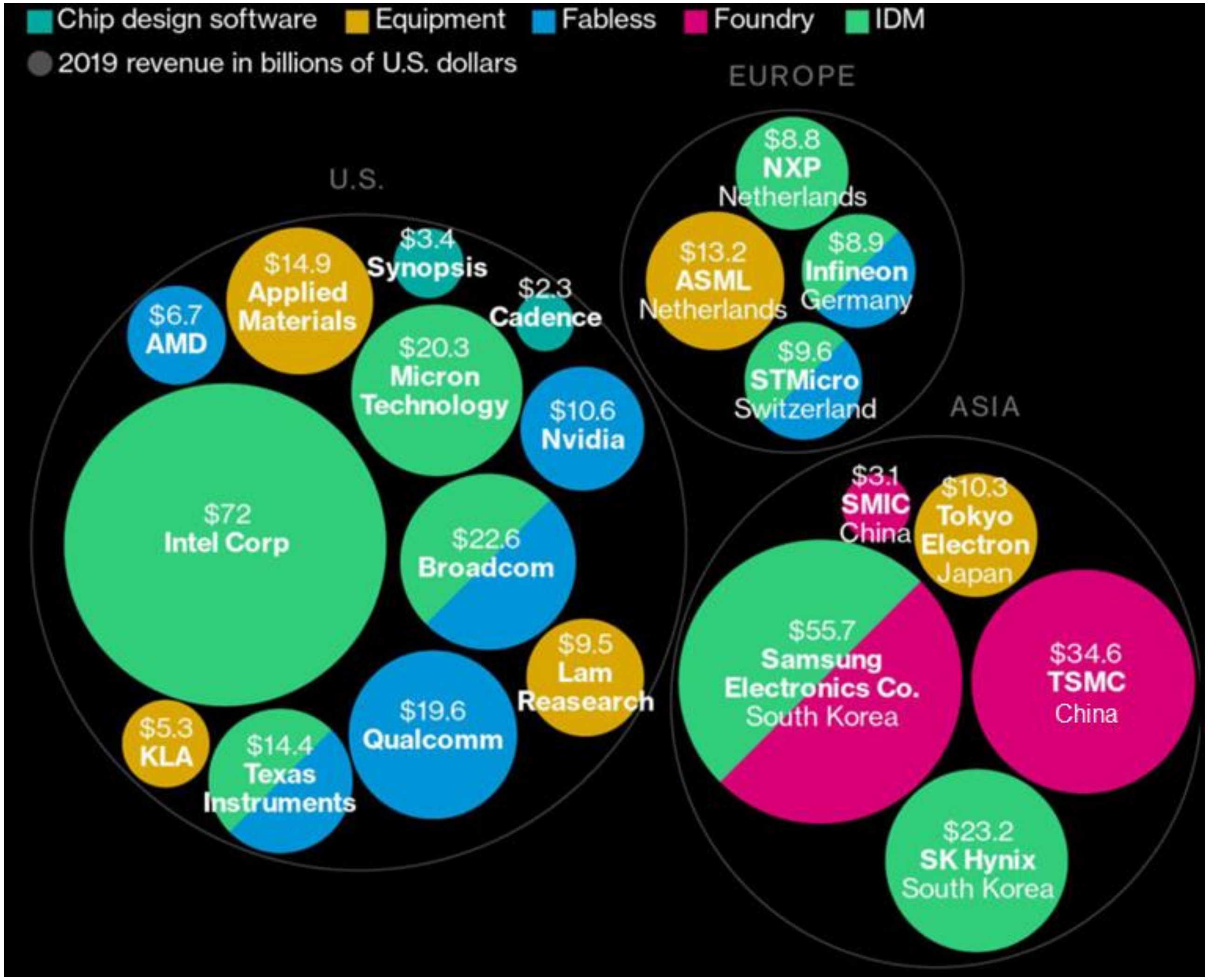
Semiconductor manufacturing process chain



Specs for the system → Architectural design → Logic design → Circuit design → Physical design → Verifications and tests → Fabrication → Packaging and testing → Chip → PCB design → System → Application such as smart Home, or E-Health

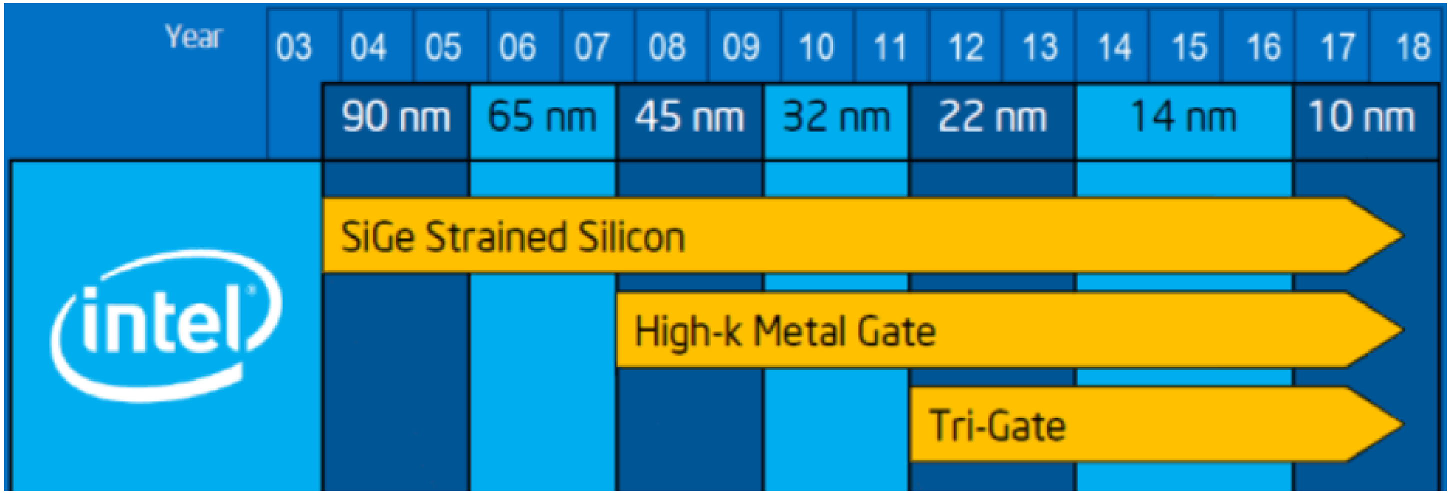
Chip Industry Choke Points

- Key players in chip industry
  - Intel
  - AMD
  - Qualcomm
  - Samsung
  - TSMC
  - Broadcom
  - Nvidia
  - ASML
  - …



Processor Technology Trends

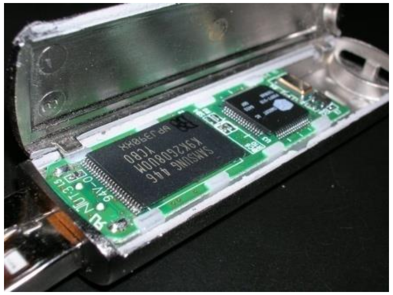
- Shrinking of transistor sizes: 90nm(2004) → 45nm(2008) → 22nm(2012) → 10nm(2017) …



- 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

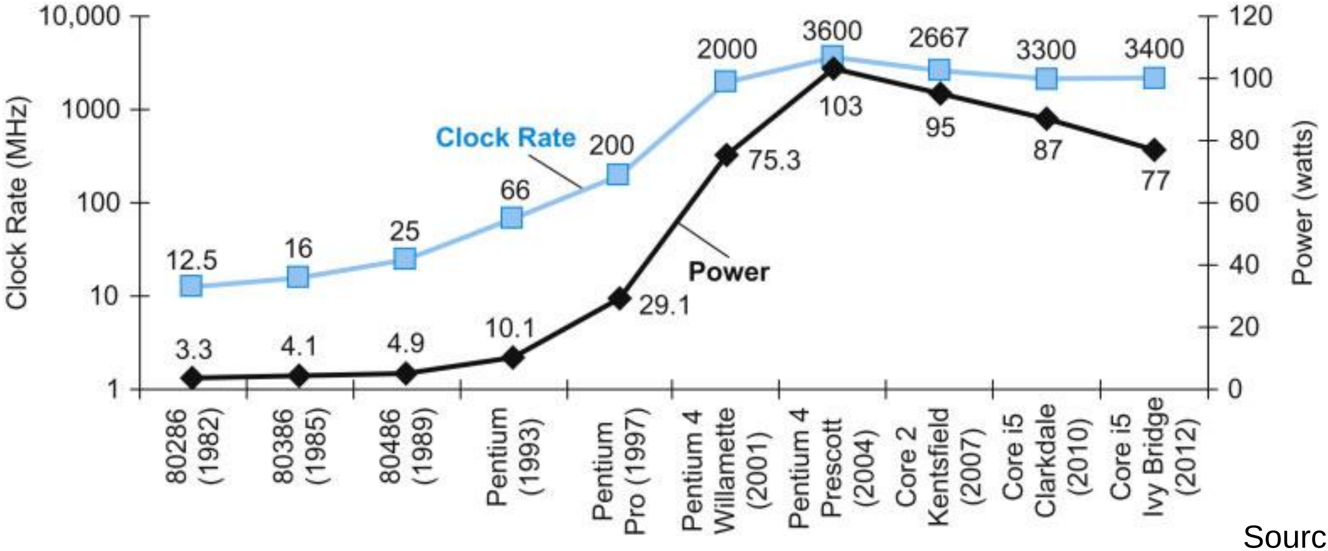
- DRAM density increases by 40-60% per year, latency has reduced by 33% in 10 years (the memory wall!), bandwidth improves twice as fast as latency decreases

- 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

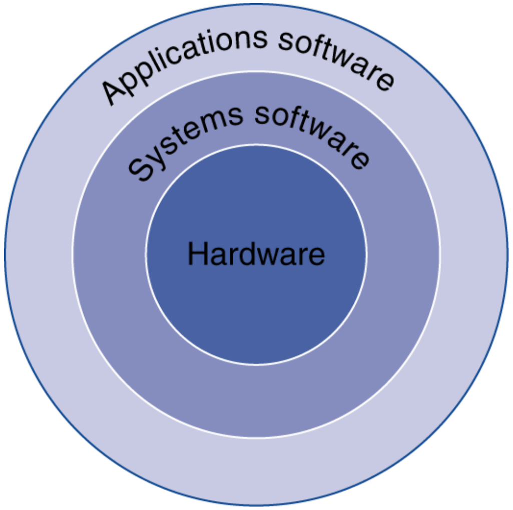
Power Consumption Trends

- Dyn power ∝ activity x capacitance x voltage2 x frequency
- 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)



Source: H&P Textbook

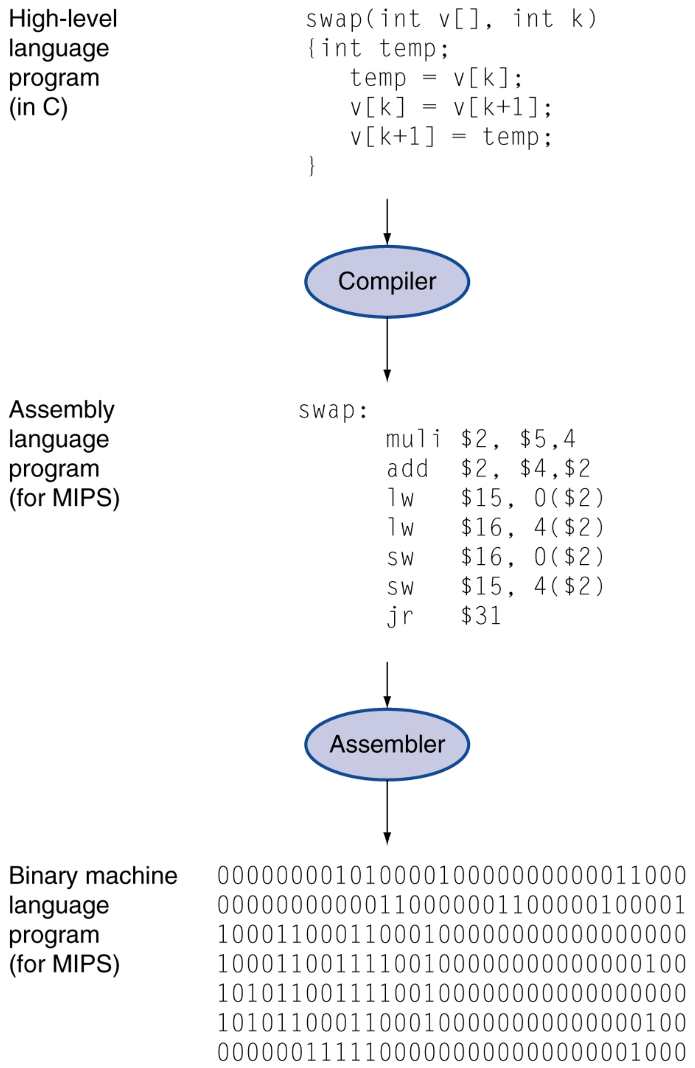
Between Your Program and Hardware



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

Levels of Program Code

- High-level language
  - Level of abstraction closer to problem domain
  - Provides for productivity and portability
- Assembly language
  - Textual representation of instructions
- Hardware representation
  - Binary digits (bits)
  - Encoded instructions and data


High-level language program (in C)
Compiler
Assembly language program (for MIPS)
Assembler
Binary machine language program (for MIPS)

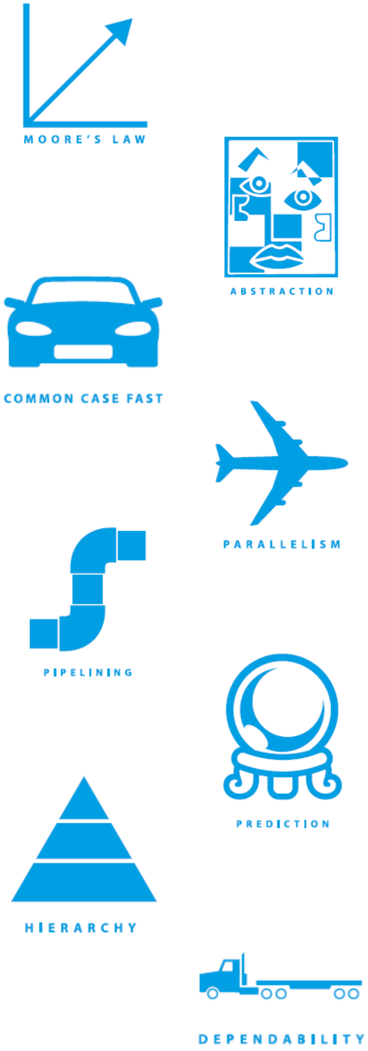
```
swap(int v[], int k)
{int temp;
   temp = v[k];
   v[k] = v[k+1];
   v[k+1] = temp;
}
```

```
swap:
      muli $2, $5,4
      add  $2, $4,$2
      lw   $15, 0($2)
      lw   $16, 4($2)
      sw   $16, 0($2)
      sw   $15, 4($2)
      jr   $31
```

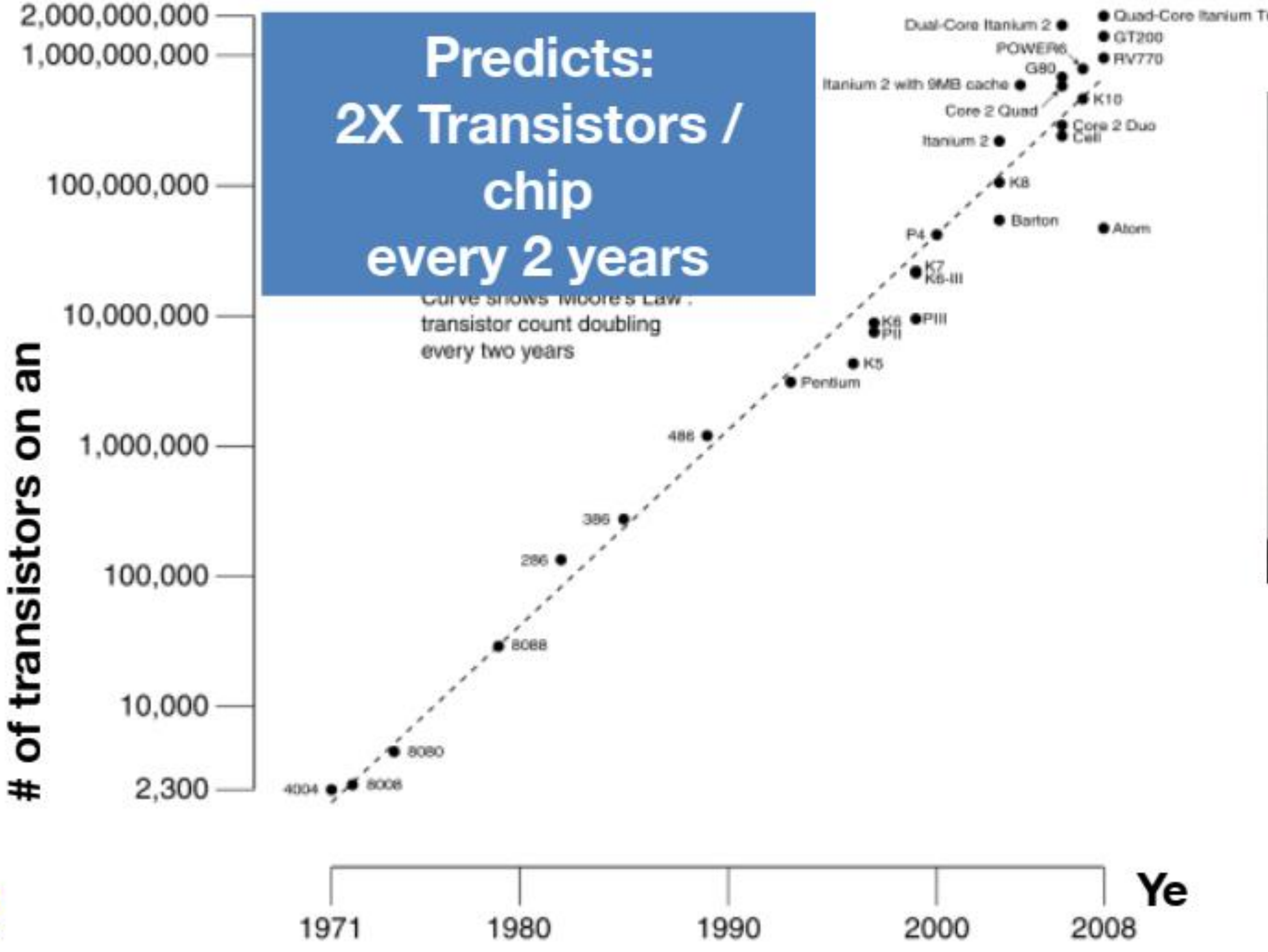
```
00000000101000010000000000011000
00000000000110000001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

Eight Great Ideas

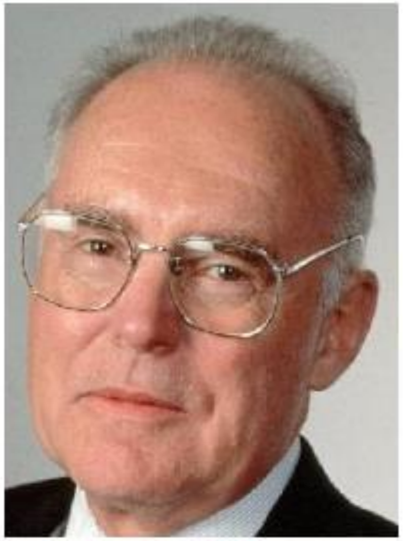
- Design for ***Moore's Law***
- Use ***abstraction*** to simplify design
- Make the ***common case fast***
- Performance *via* ***parallelism***
- Performance *via* ***pipelining***
- Performance *via* ***prediction***
- ***Hierarchy*** of memories
- ***Dependability*** *via* redundancy



Moore’s Law

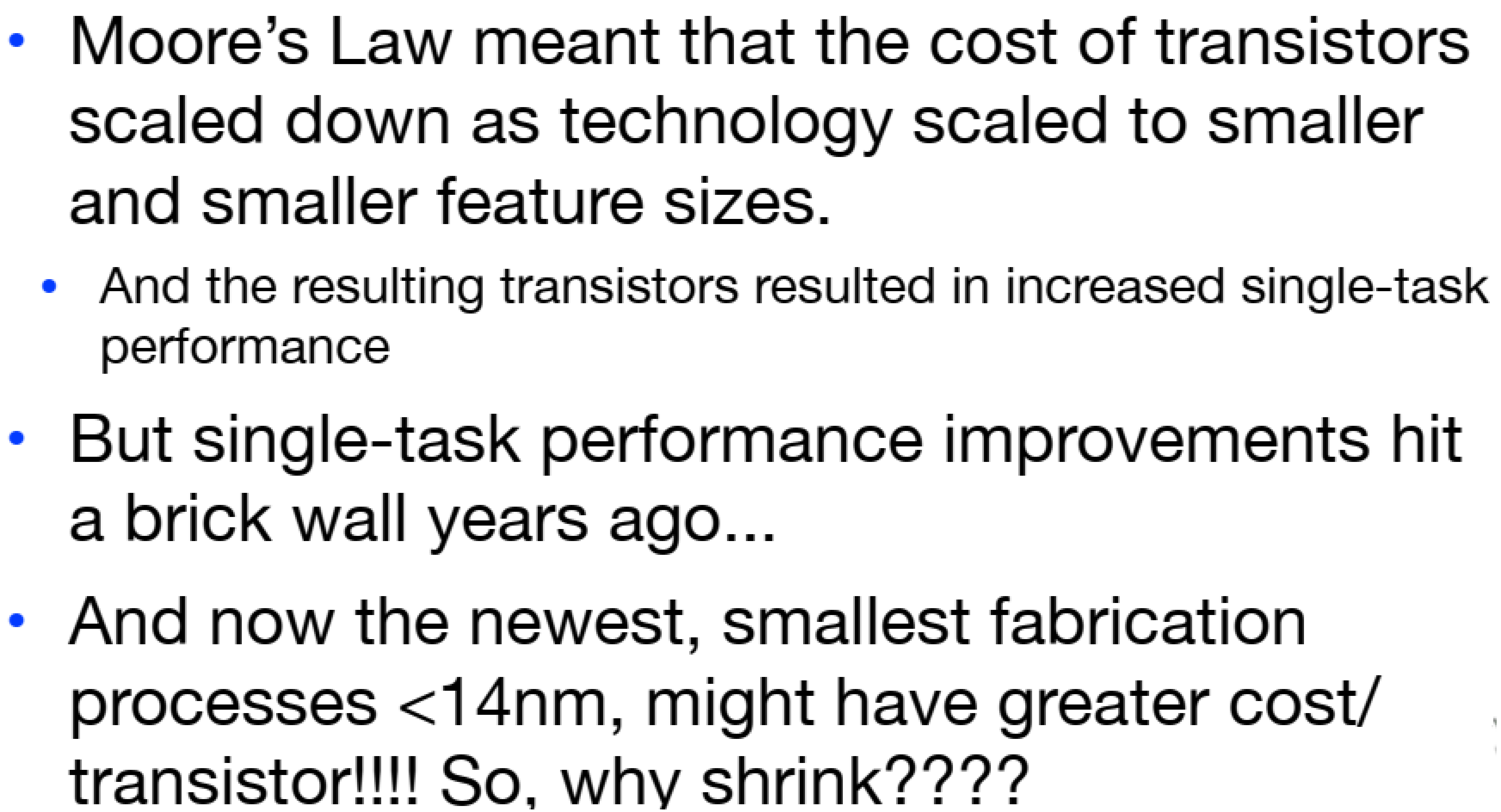


Predicts:
2X Transistors / chip
every 2 years



Gordon Moore
Intel Cofounder
B.S. Cal 1950!

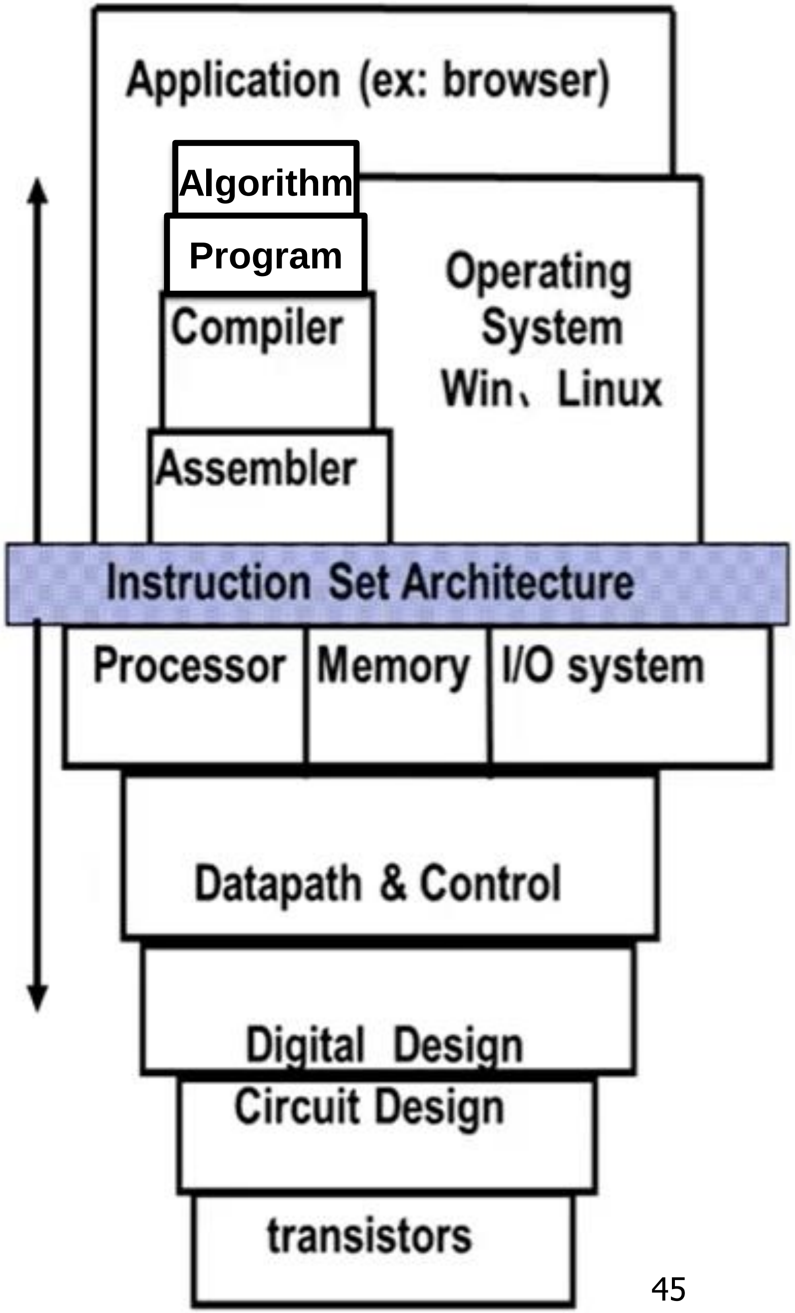
Post Moore's Law



- Moore's Law meant that the cost of transistors scaled down as technology scaled to smaller and smaller feature sizes.
  - And the resulting transistors resulted in increased single-task performance
- But single-task performance improvements hit a brick wall years ago...
- And now the newest, smallest fabrication processes <14nm, might have greater cost/transistor!!!! So, why shrink????

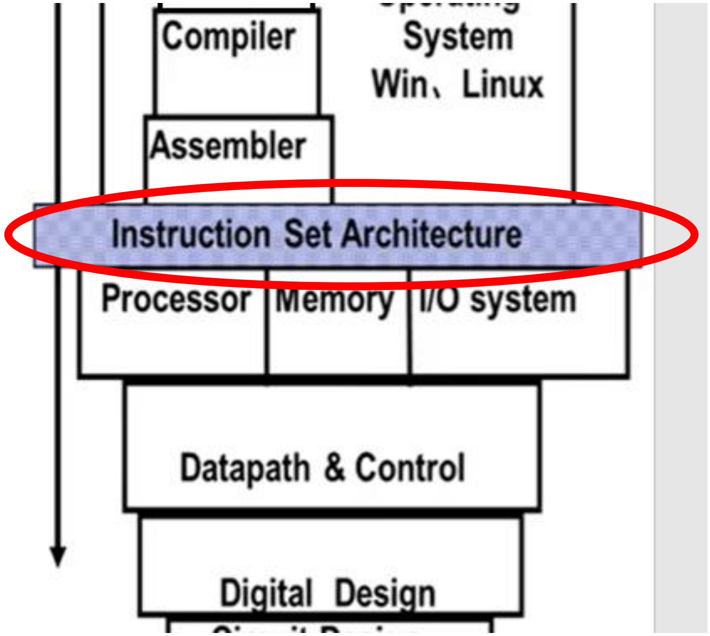
Abstractions

- 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

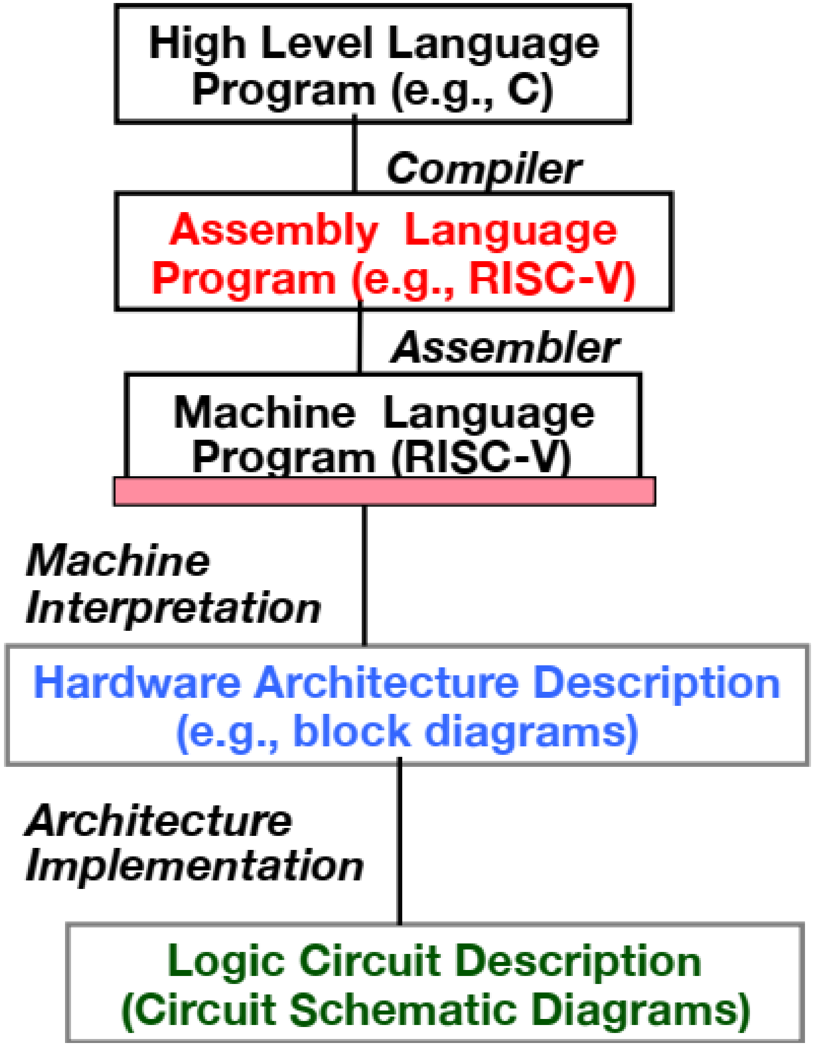


Instruction Set Architecture (ISA)

- 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)



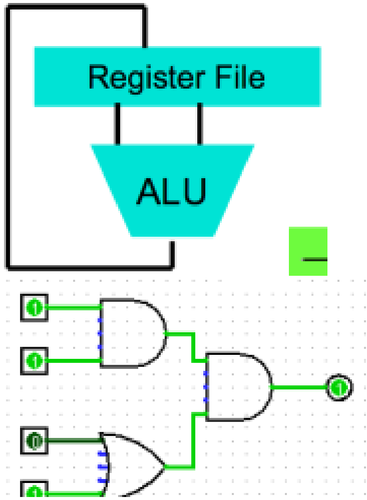
Abstraction


High Level Language Program (e.g., C)
Compiler
Assembly Language Program (e.g., RISC-V)
Assembler
Machine Language Program (RISC-V)
Machine Interpretation
Hardware Architecture Description (e.g., block diagrams)
Architecture Implementation
Logic Circuit Description (Circuit Schematic Diagrams)

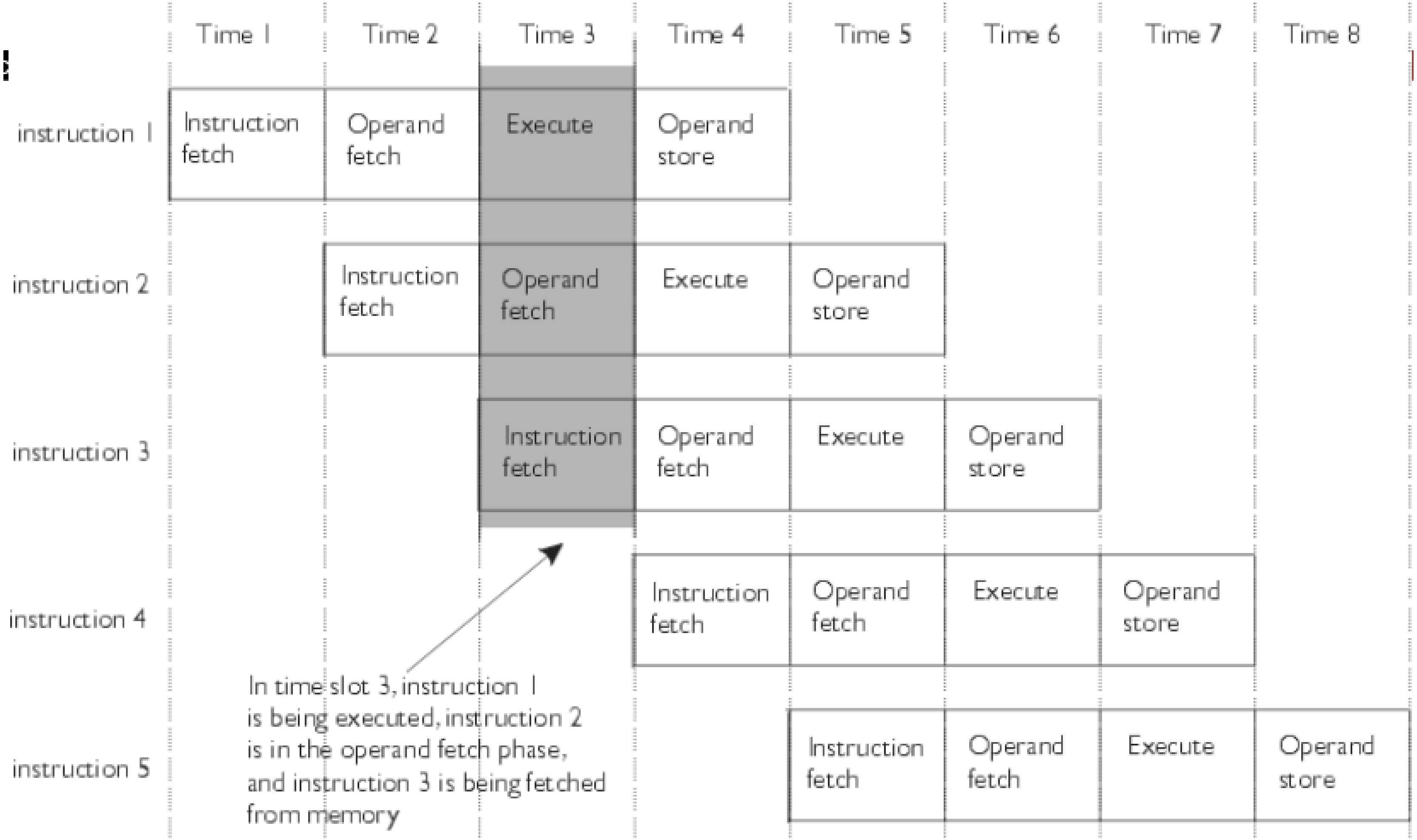
```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

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

```
0000 1001 1100 0110 1010    )0
1010 1111 0101 1000 0000    .0
1100 0110 1010 1111 0101    )1
0101 1000 0000 1001 1100    .1
```



Pipeline



	Time 1	Time 2	Time 3	Time 4	Time 5	Time 6	Time 7	Time 8
instruction 1	Instruction fetch	Operand fetch	Execute	Operand store				
instruction 2		Instruction fetch	Operand fetch	Execute	Operand store			
instruction 3			Instruction fetch	Operand fetch	Execute	Operand store		
instruction 4				Instruction fetch	Operand fetch	Execute	Operand store	
instruction 5					Instruction fetch	Operand fetch	Execute	Operand store

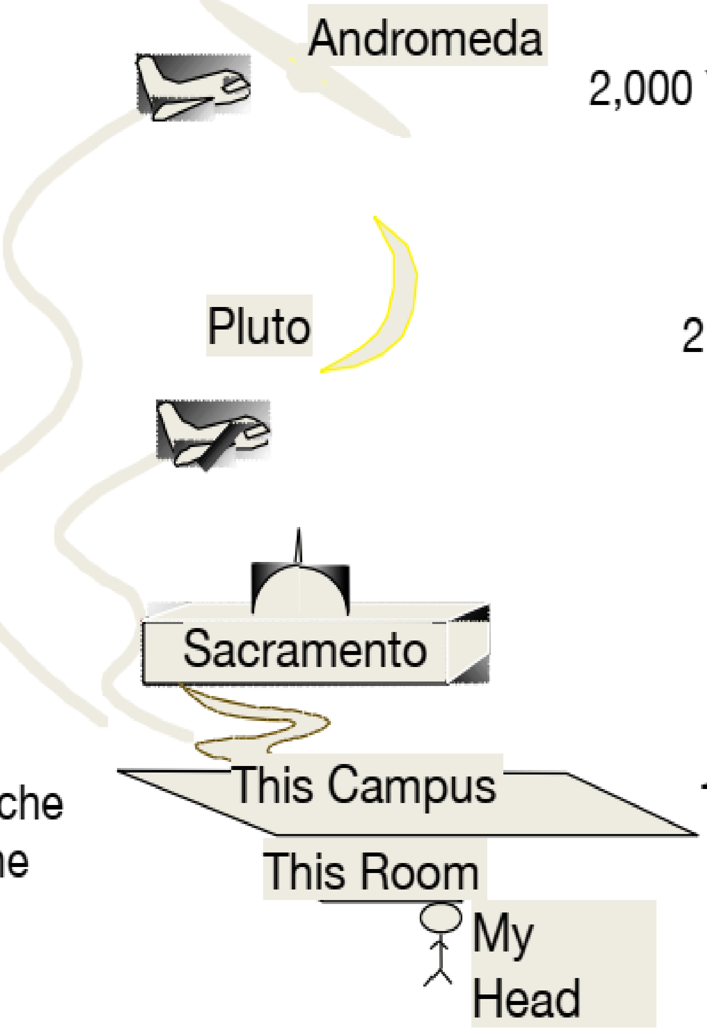
In time slot 3, instruction 1 is being executed, instruction 2 is in the operand fetch phase, and instruction 3 is being fetched from memory

Parallelism

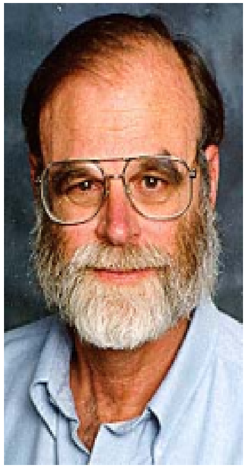
- Raspberry Pi 4 B+
  - Quad core processor at 1.5 GHz
    - Each core is 3-issue, out-of-order superscalar
  - Plus GPU, 1-4 GB RAM, 2x USB3, 2x USB2, Gigabit Ethernet, 2x HDMI...
  - $35-55
    - Nick is working on a board to turn one of these to power a fully autonomous, vision-guided drone
- Compare with a Cray-1 from 1975:
  - 8 MB RAM, 80 MHz processor, 300MB storage, $5M+
- Or modern high end servers:
  - You can get a 2u server which supports 4 processors
    - And each processor can have 20+ cores, so 80 processor cores!

Jim Gray's Storage Latency Analogy

- How far away is the Data?

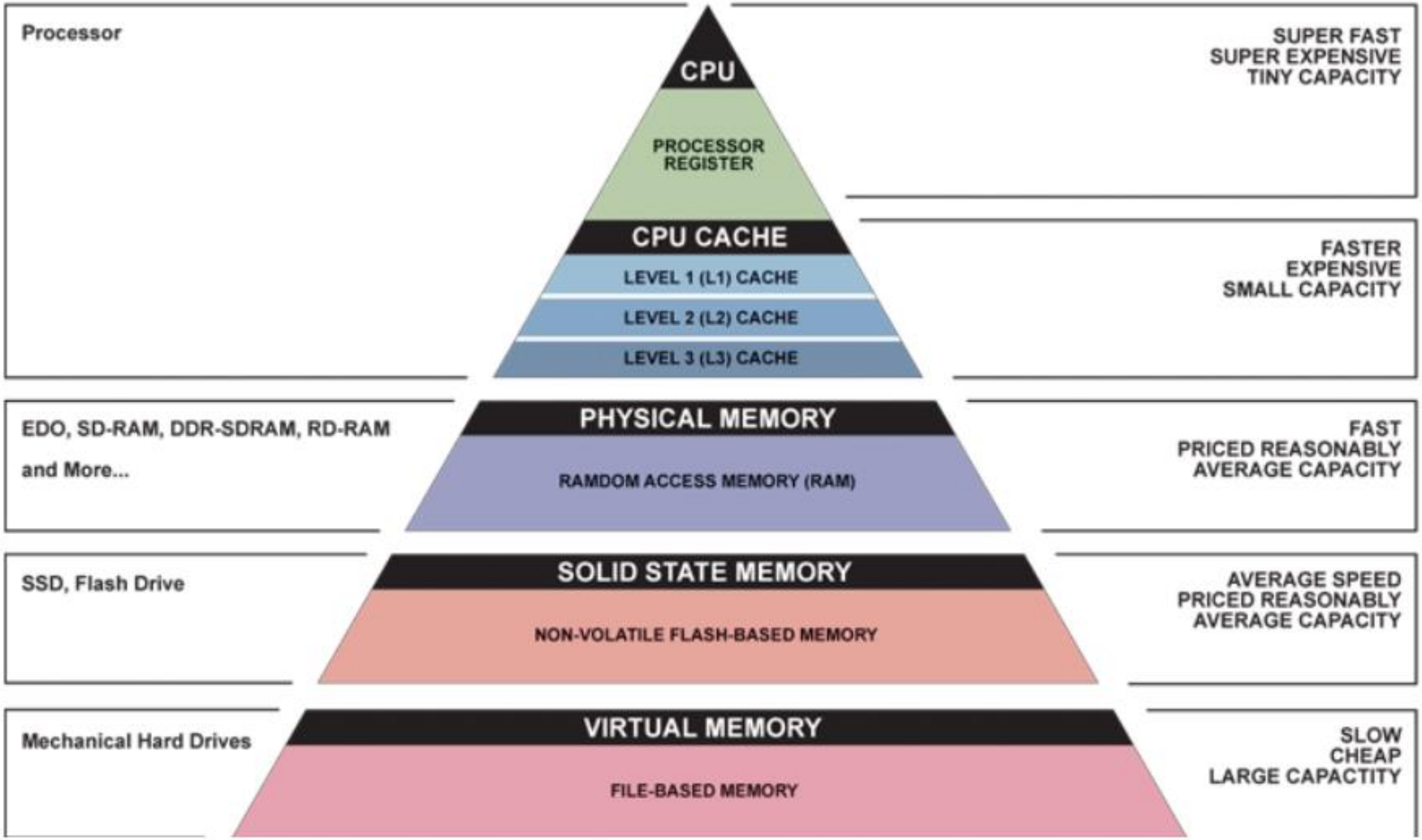


10^9	Tape Robot	Andromeda	2,000 Years
10^6	Disk	Pluto	2 Years
		Sacramento	1.5 hr
100	Main Memory		
10	On Board Cache	This Campus	10 min
2	On Chip Cache	This Room	
1	Registers	My Head	1 min
(ns)			



Jim Gray
Turing Award
B.S. Cal 1966
Ph.D. Cal 1969!

Memory Hierarchy



Fails Happen, so?

- 4 disks/server, 50,000 servers
- Failure rate of disks: 2% to 10% / year
  - Assume 4% annual failure rate
- On average, how often does a disk fail?

a) 1 / month

b) 1 / week

c) 1 / day

d) 1 / hour

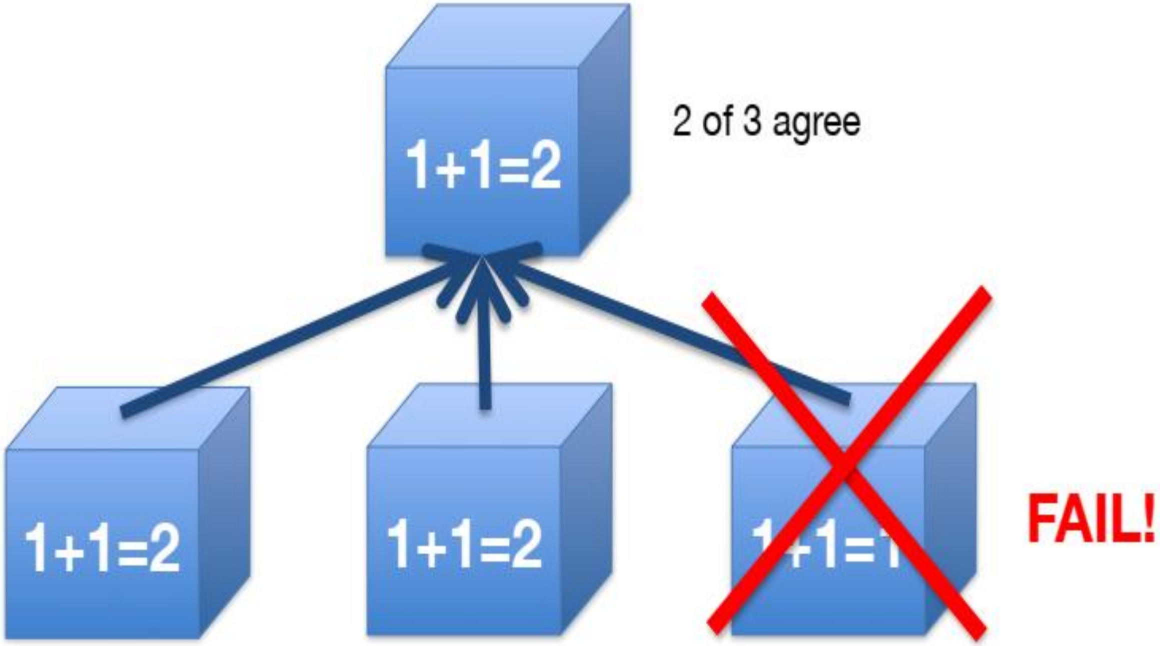
50,000 x 4 = 200,000 disks

200,000 x 4% = 8000 disks fail

365 days x 24 hours = 8760 hours

Reliability via Redundancy

Redundancy so that a failing piece doesn’t make the whole system fail



Increasing transistor density reduces the cost of redundancy