Jin-Soo Kim
(*jinsoo.kim@snu.ac.kr*)

Systems Software &
Architecture Lab.

Seoul National University

Fall 2020

Introduction to Computer Architecture

Chap. 1.1 – 1.5, 1.7 – 1.8

Computer Systems









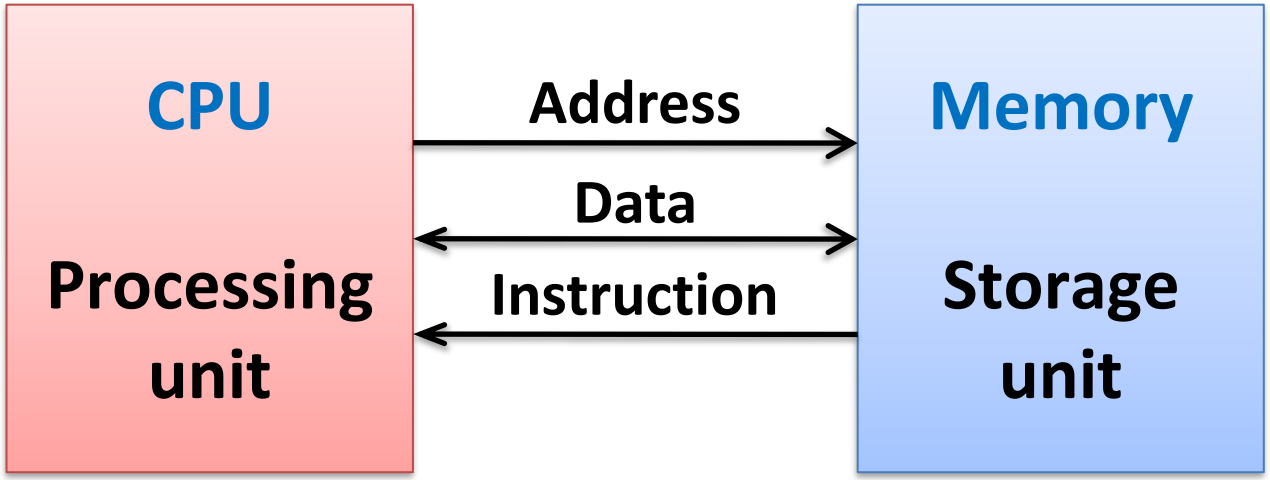

Classes of Computers

- Personal computers
  - General purpose, variety of software
  - Subject to cost/performance tradeoff
- Supercomputers
  - High-end scientific and engineering calculations
  - Highest capability but represent a small fraction of the overall computer market
- Server computers
  - Network based
  - High performance, capacity, reliability
  - Range from small servers to large data centers
- Embedded computers
  - Hidden as components of systems
  - Stringent power/performance/cost constraints

A computer is a  machine.

von Neumann Architecture

- By John von Neumann, 1945

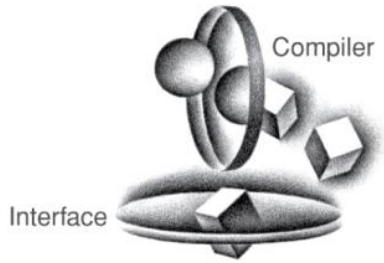


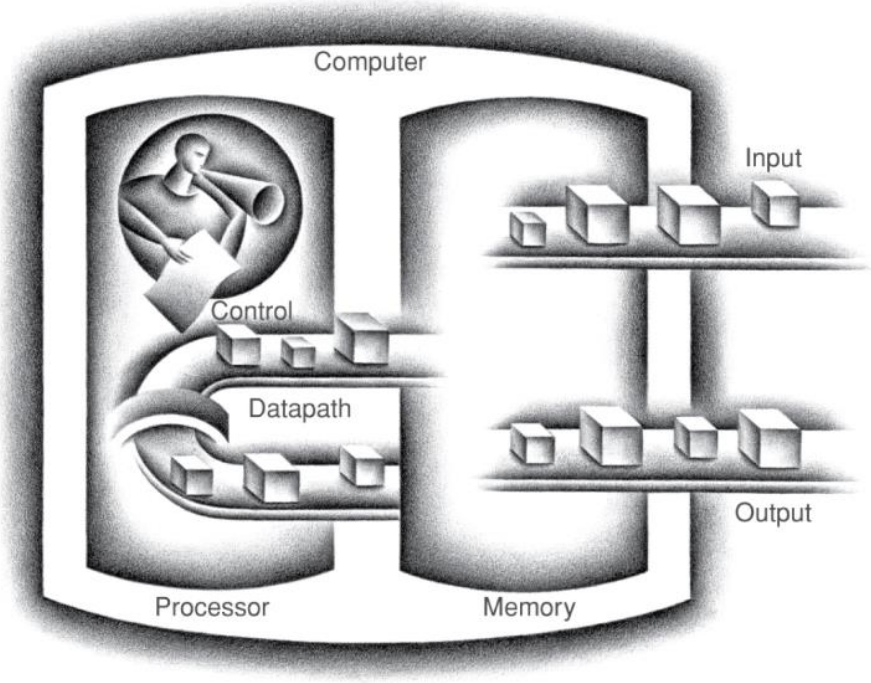
Data movement
Arithmetic & logical ops
Control transfer

Byte addressable array
Code + data (user program, OS)
Stack to support procedures

Components of a Computer

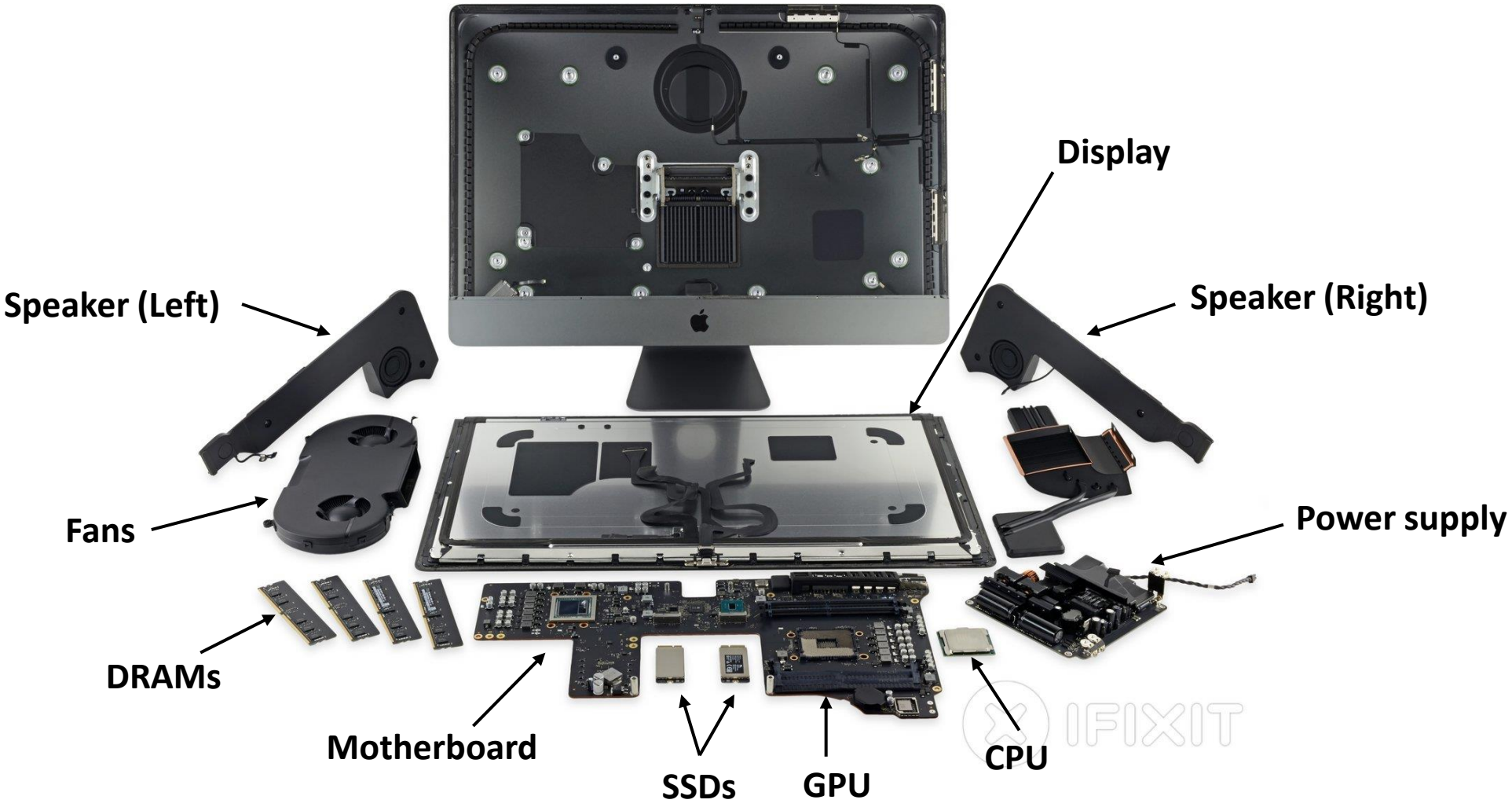
- CPU: Control + Datapath
- Memory
- I/Os
  - GPUs
  - User-interface devices:
    Display, keyboard, mouse, sound, …
  - Storage devices:
    HDD, SSD, CD/DVD, …
  - Network adapters:
    Ethernet, 3G/4G/5G, WiFi, Bluetooth, …





- Same components for all kinds of computer

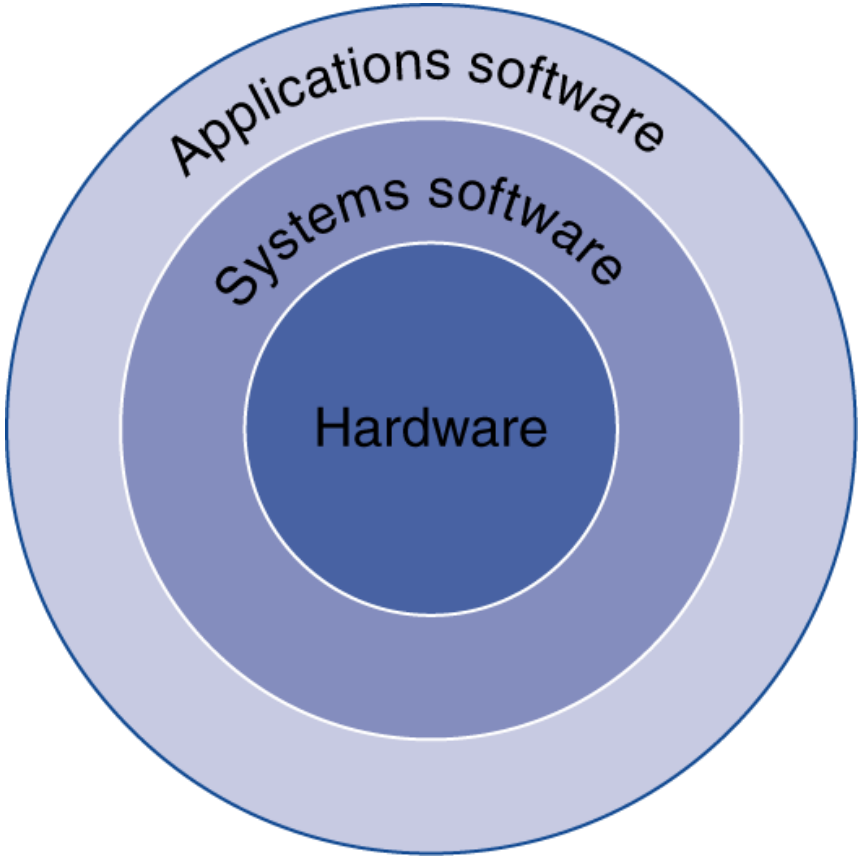
Opening the Box (iMac Pro)



Source: https://www.ifixit.com/Teardown/iMac+Pro+Teardown/101807

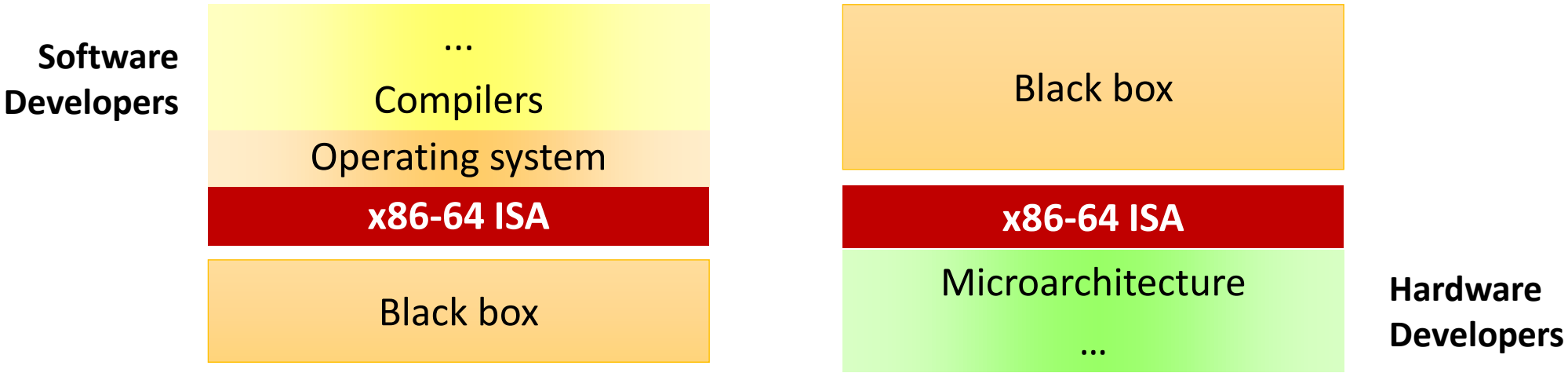
Below Your Program

- Application software
  - Written in high-level language
- 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



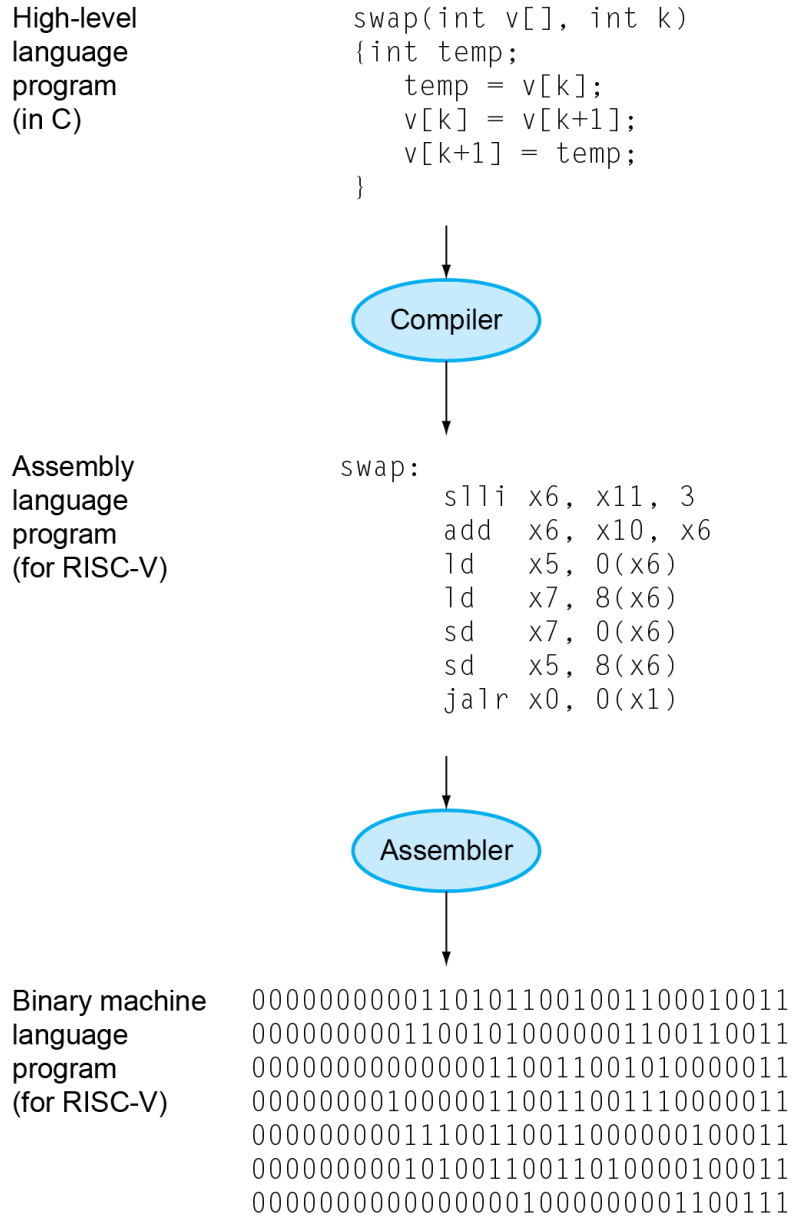
Instruction Set Architecture (ISA)

- The hardware/software interface
  - Hardware abstraction visible to software (OS, compilers, …)
  - Instructions and their encodings, registers, data types, addressing modes, etc.
  - Written documents about how the CPU behaves
  - e.g., All 64-bit Intel CPUs follow the same x86-64 (or Intel 64) ISA



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
  - For humans
- Machine language
  - Hardware representation
  - Binary digits (bits)
  - Encoded instructions and data



What Happened:

104 cabinets
(76 computes,
8 switches,
20 disks)

9298 cores

150m^2

1997

ASCI Red at Sandia

1.3 TF/s, 850 KW

2017

Cavium ThunderX2

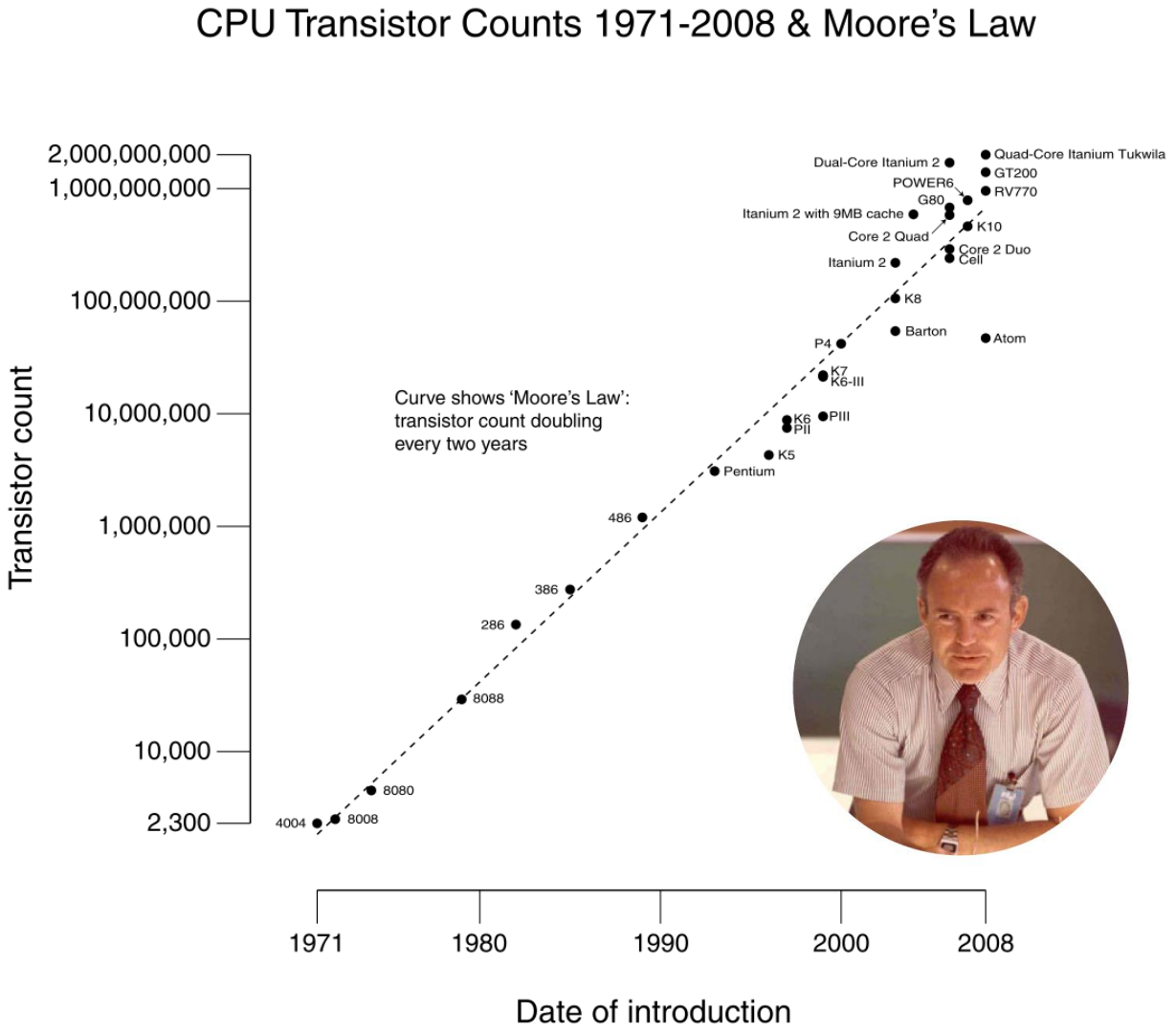
~ 1.1 TF/s, ~ 0.2 KW

3.5 orders of magnitude

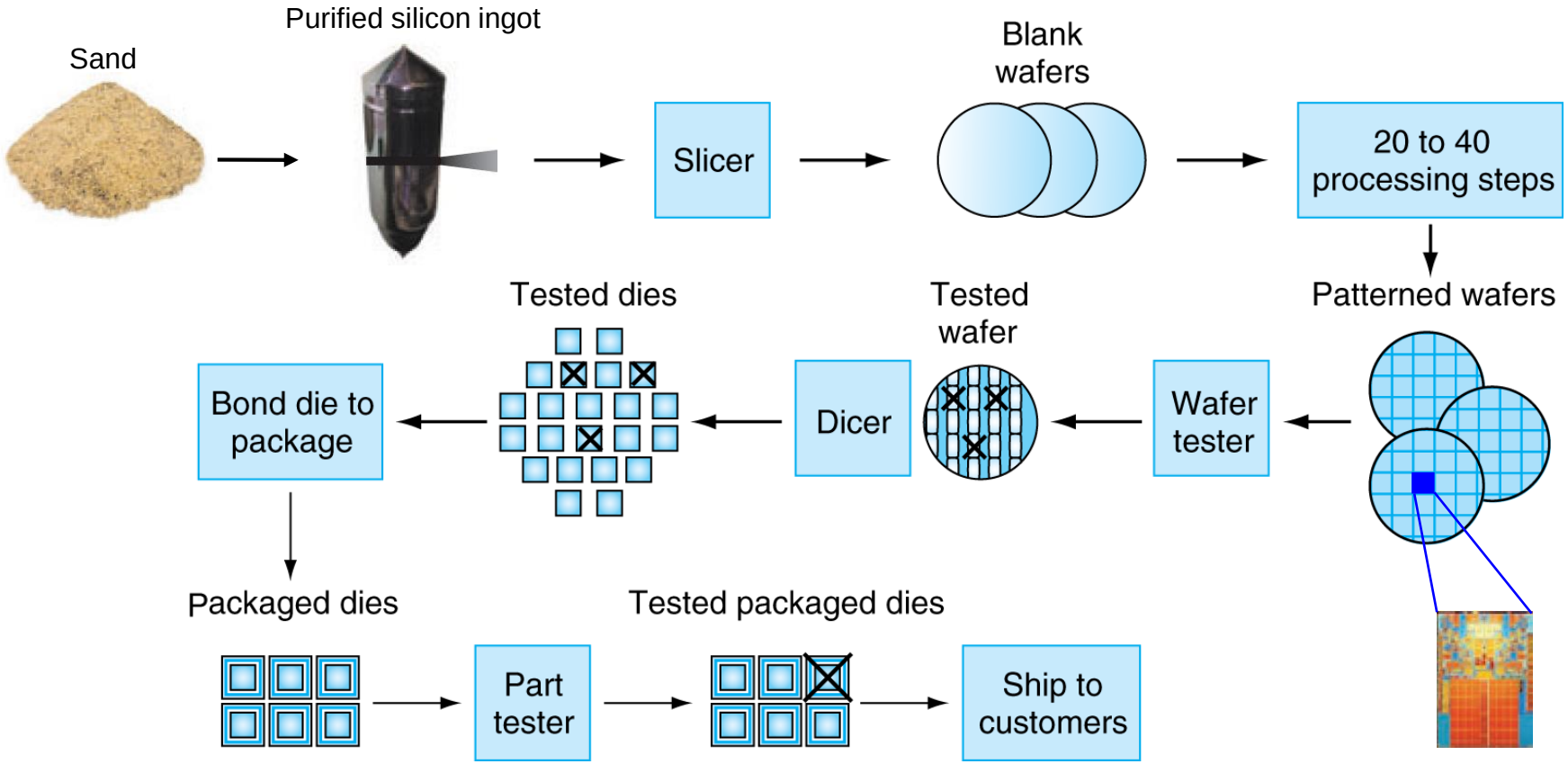
Source: David Keyes, "Algorithmic Adaptations to Extreme Scale Computing," ATPESC, 2018.

The Moore's Law

- *"The number of transistors incorporated in a chip will approximately double every 24 months"*
  (Gordon Moore, Intel Co-founder, 1965)
- Makes novel applications feasible
  - WWW, search engines,
  - Smartphones, VR/AR,
  - AI, Self-driving cars,
  - Human genome project, …
- Computers are pervasive

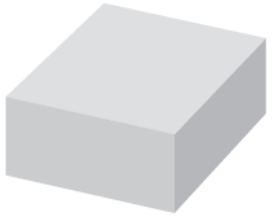


From Sand to Circuits

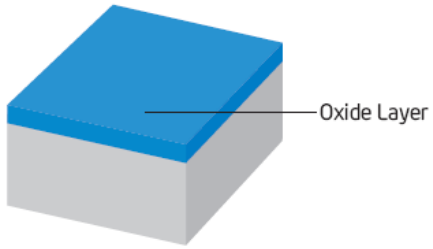


Yield: proportion of working dies per wafer

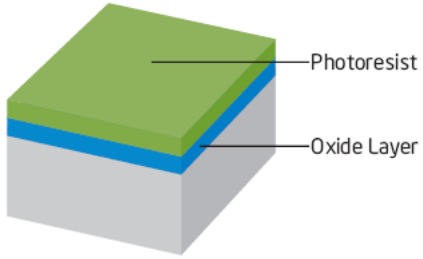
Processing ICs



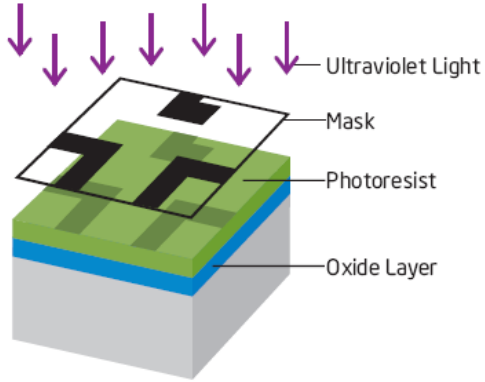
1. Start with a partially processed die on a silicon wafer.



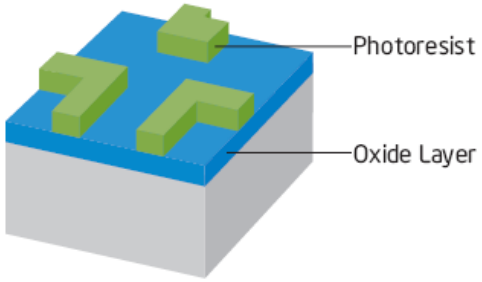
2. Deposit oxide layer.



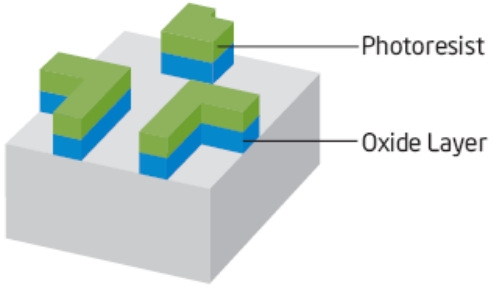
3. Coat with photoresist.



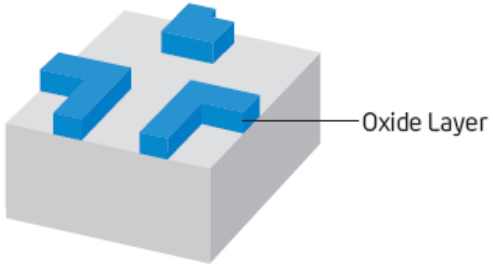
4. Position mask and flash ultraviolet light.



5. Rinse with solvent.



6. Etch with acid.



7. Remove remaining photoresist.

Source: Intel

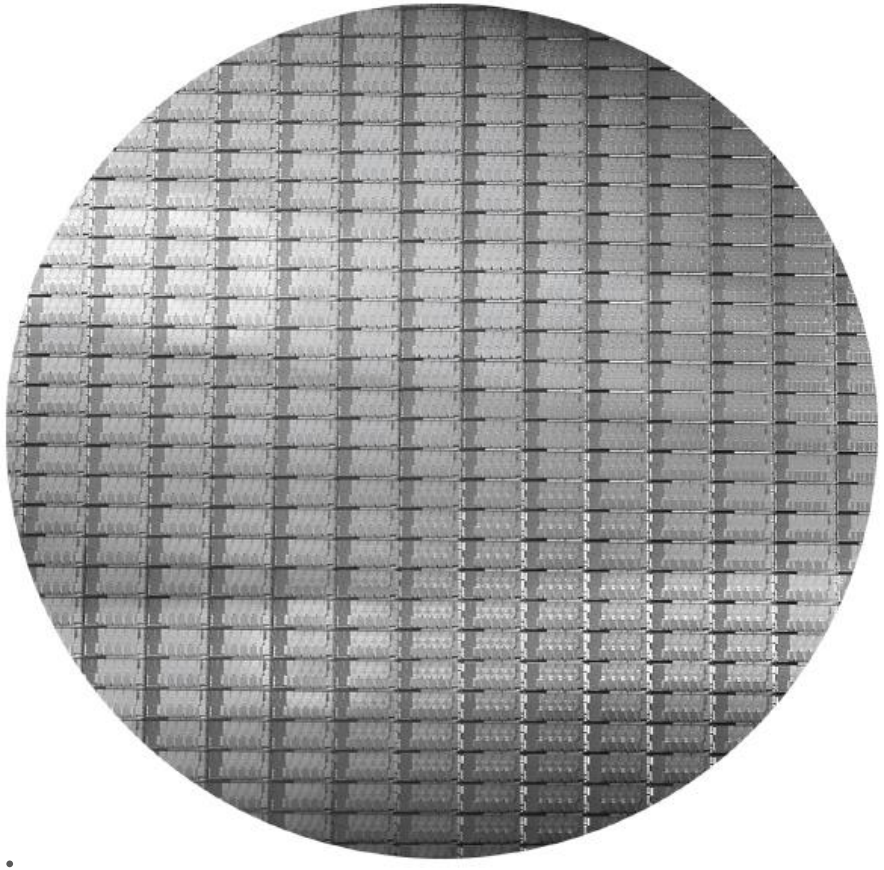
Intel Core i7 Wafer

- 12-inch (300mm) wafer, 280 chips, 32nm technology
- Each chip is 20.7 x 10.5 mm

$$\textbf{Cost per die} = \frac{\textbf{Cost per wafer}}{\textbf{Dies per wafer} \times \textbf{Yield}}$$

$$\textbf{Dies per wafer} \approx \textbf{Wafer area}/\textbf{Die area}$$

$$\textbf{Yield} = \frac{1}{\left(1 + \left(\textbf{Defects per area} \times \frac{\textbf{Die area}}{2}\right)\right)^2}$$



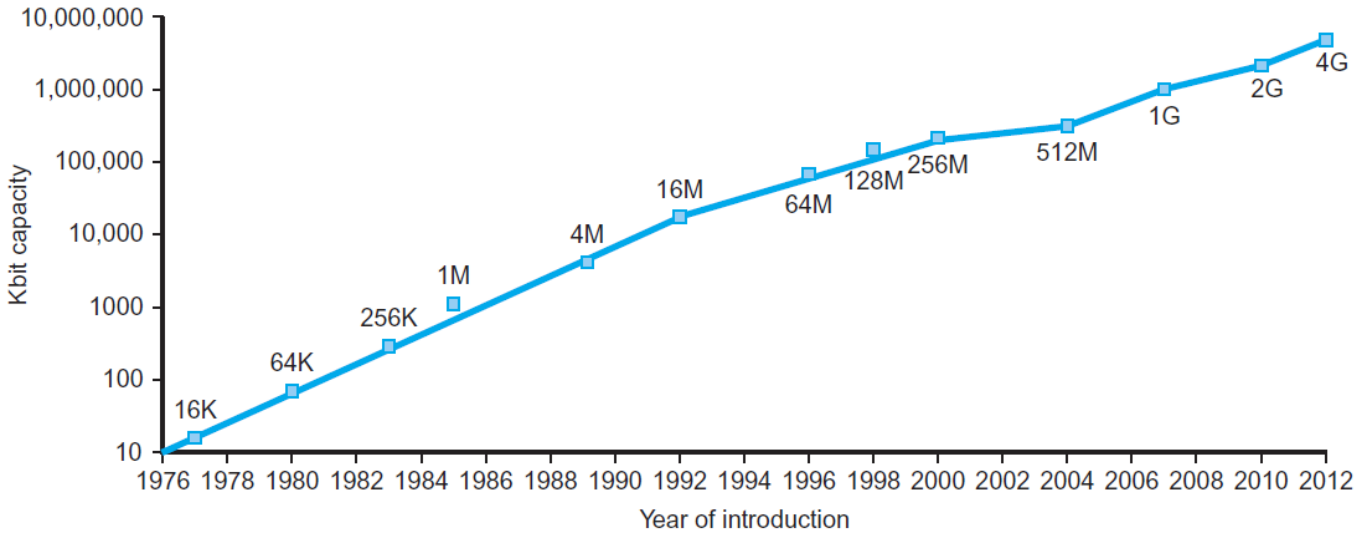
- Wafer cost and area are fixed
- Defect rate determined by manufacturing process
- Die area determined by architecture and circuit design

Technology Trends

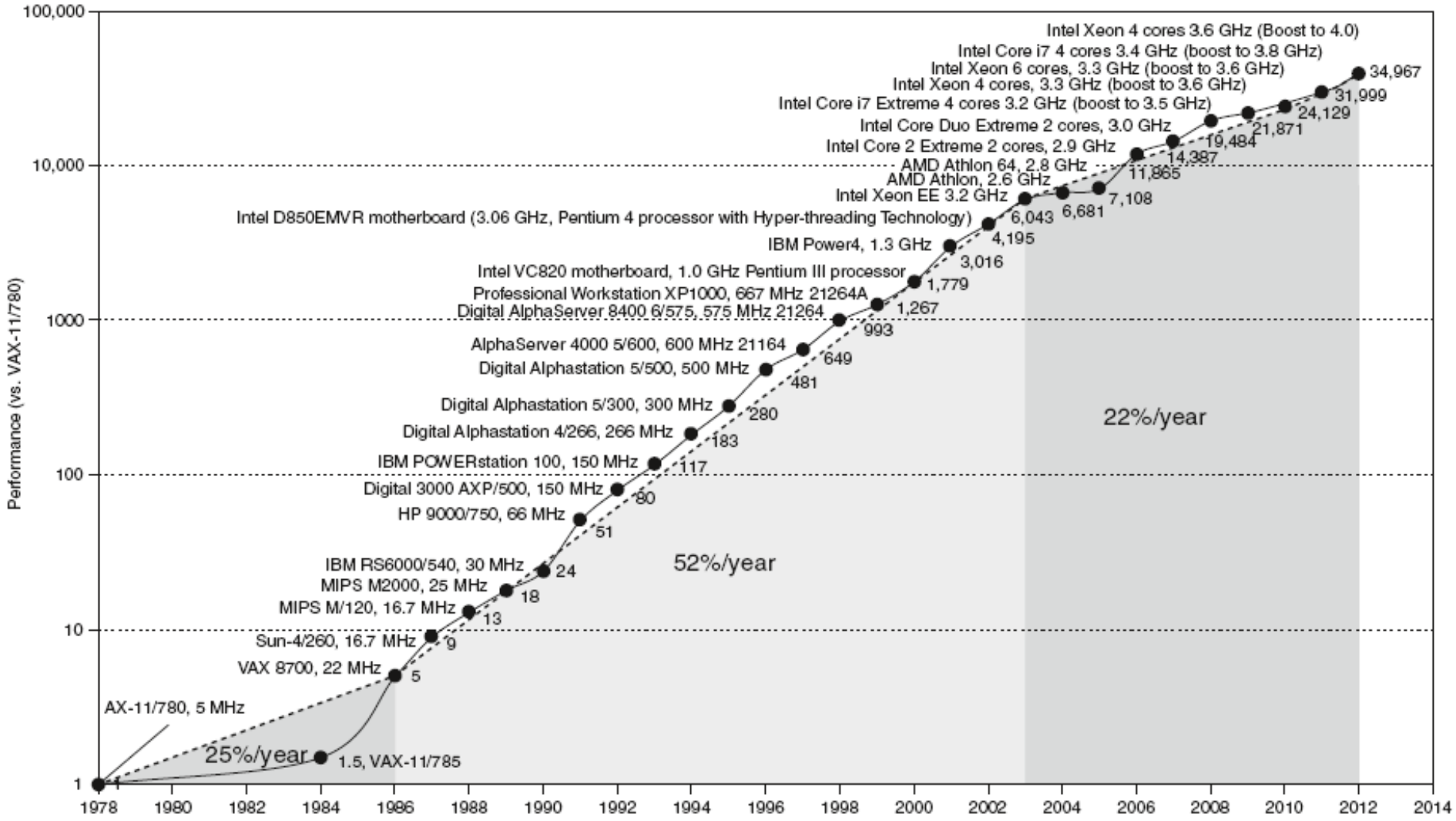
- CPU
  - Logic capacity: ~ 30% / year
  - Clock rate: ~ 20% / year

Year	Technology	Relative performance/cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit (IC)	900
1995	Very large scale IC (VLSI)	2,400,000
2013	Ultra large scale IC	250,000,000,000

- Memory
  - DRAM capacity: ~ 60% / year (4x every 3 years)
  - DRAM speed: ~ 10% / year



Uniprocessor Performance



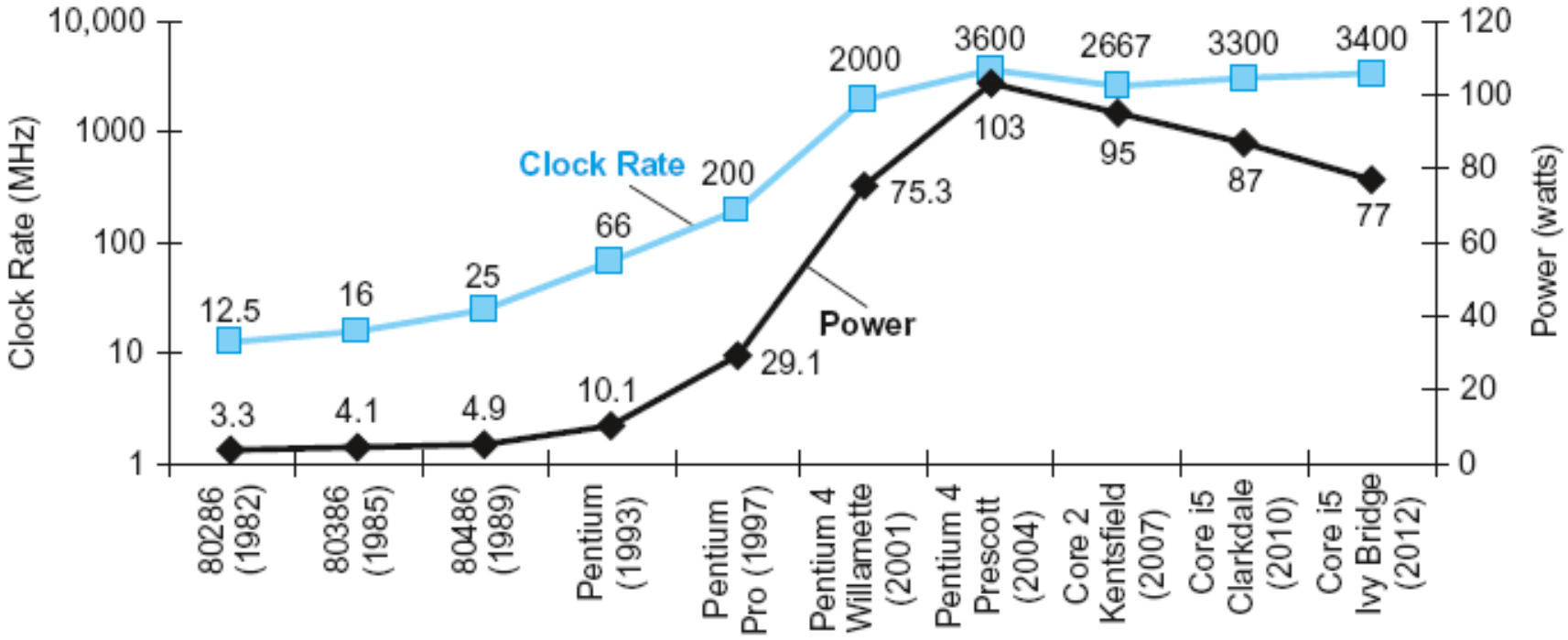
Architecture: Exploiting Parallelism

- Instruction level parallelism (ILP)
  - Pipelining
  - Superscalar
  - Out-of-order execution
  - Branch prediction
  - VLIW (Very Long Instruction Word)
- Data level parallelism (DLP)
  - SIMD / Vector instructions
- Task level parallelism (TLP)
  - Simultaneous multithreading (Hyperthreading)
  - Multicore

Power Trends

- In CMOS IC technology,

$$Power = Capactive\ Load \times Voltage^2 \times Frequency$$



Reducing Power

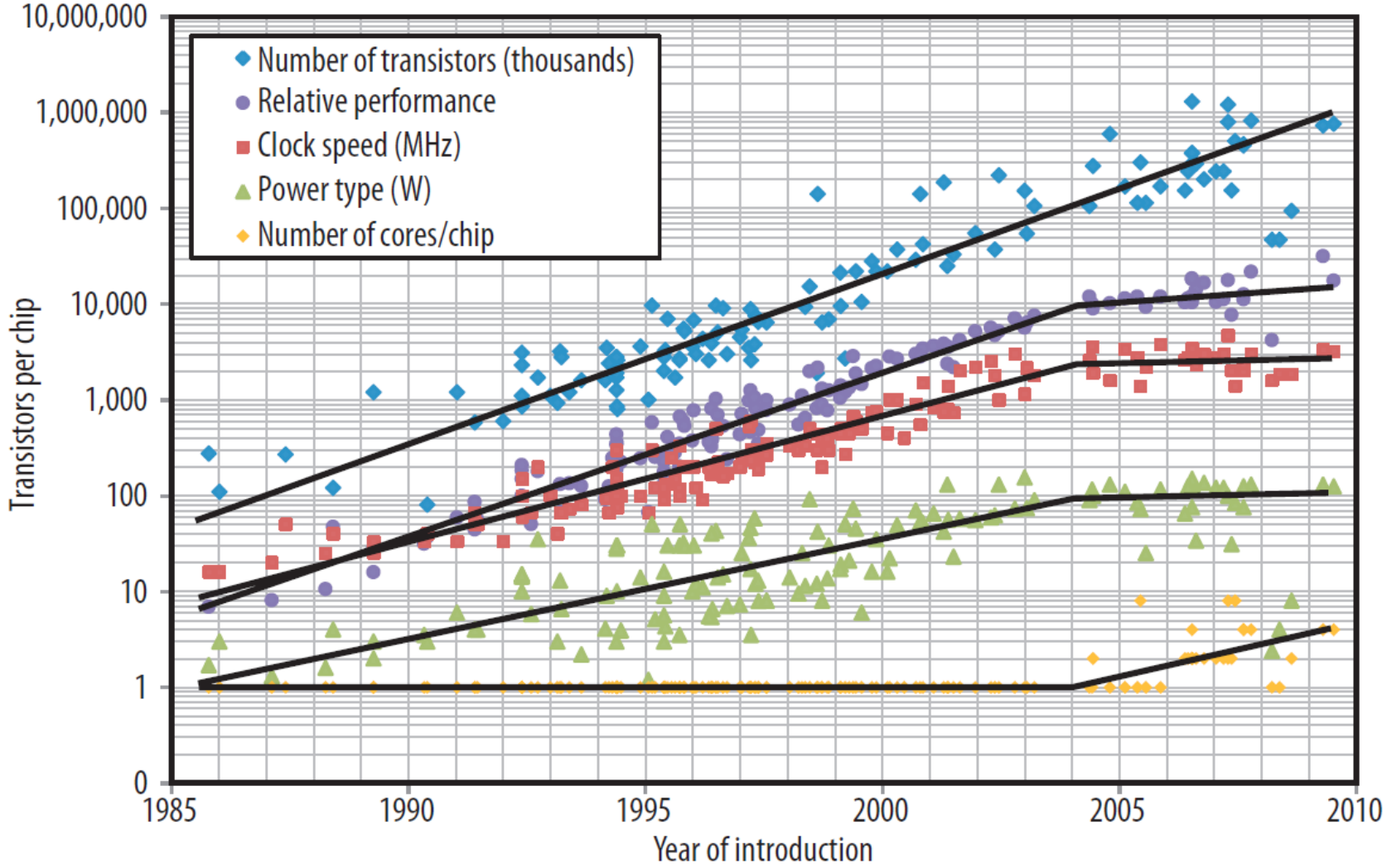
- Suppose a new CPU has
  - 85% of capacitive load of old CPU
  - 15% voltage and 15% frequency reduction

$$\frac{P_{new}}{P_{old}} = \frac{C_{old} \times 0.85 \times (V_{old} \times 0.85)^2 \times F_{old} \times 0.85}{C_{old} \times V_{old}^{\ 2} \times F_{old}} = 0.85^4 = 0.52$$

- The power wall
  - We can't reduce voltage further
  - We can't remove more heat
- How else can we improve performance?

The Shift to Multicores

- ILP wall
  - Control dependency
  - Data dependency
- Memory wall
  - Memory latency improved by 10% / year
  - Cache shows diminishing returns
- Power wall

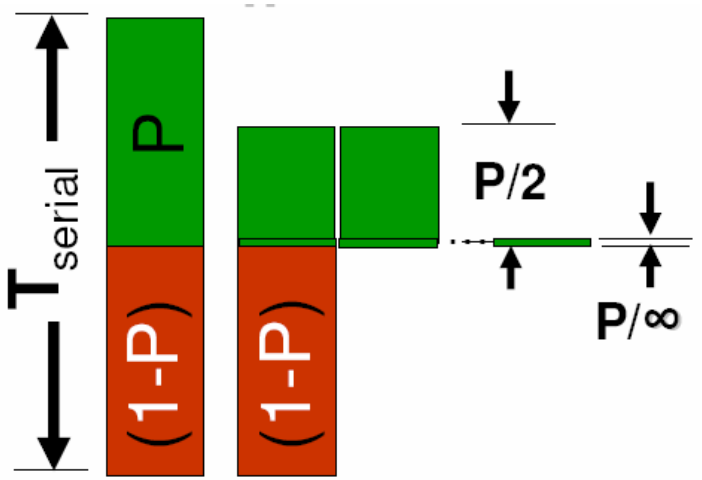


Source: IEEE Computer, 2011

Amdahl's Law

- The theoretical upper limit of speed up is limited by the serial portion of the code

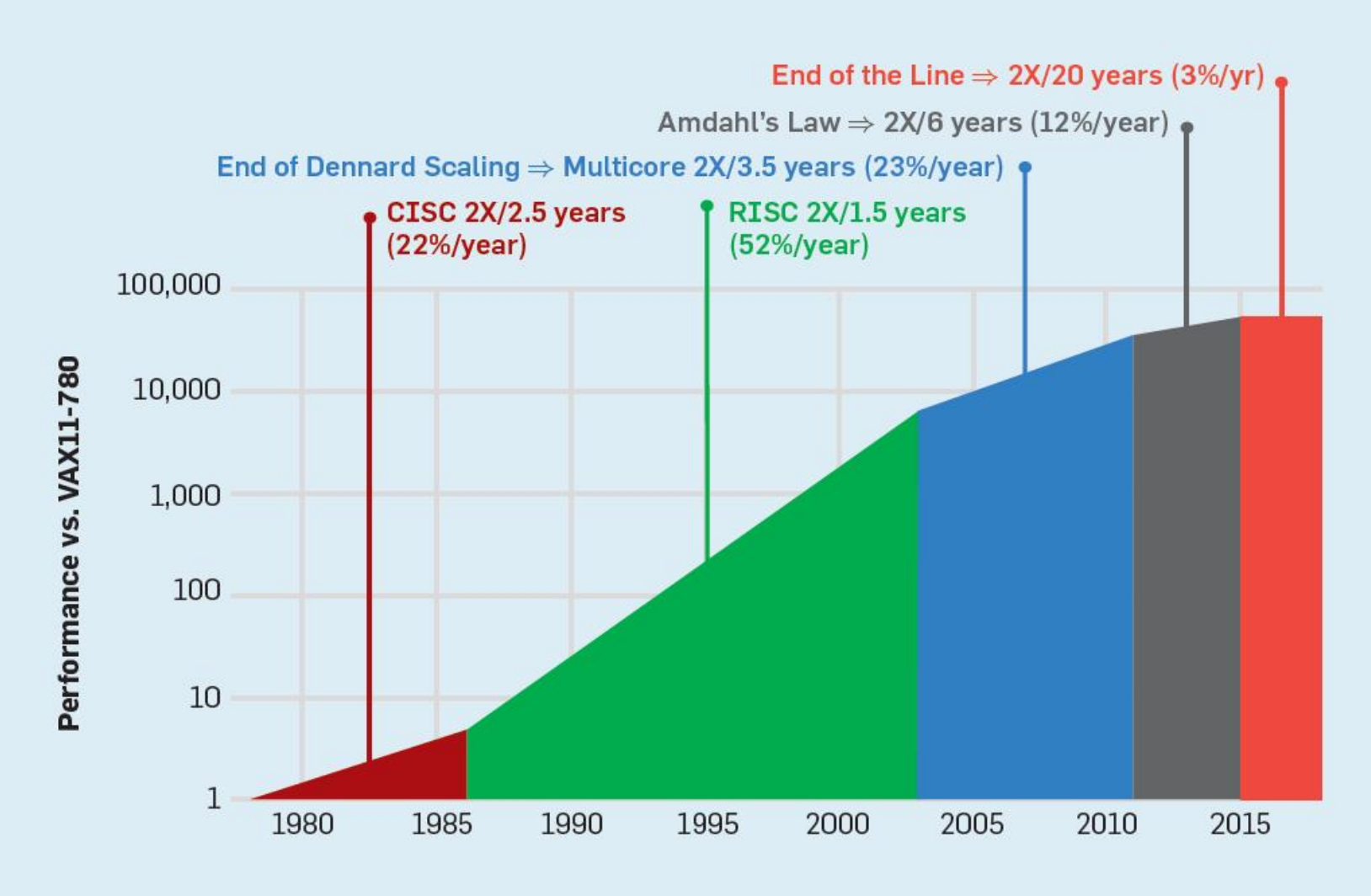
$$Speedup = \frac{1}{(1-P) + \frac{P}{n}}$$



  - $S = (1 - P)$: the time spent executing the serial portion
  - n: the number of processor cores

- Corollary: make the common case fast

Summary: A New Golden Age?



End of Moore's law

A New Golden Age for Computer Architecture (CACM, Feb. 2019)

Source: J. L. Hennessy and D. A. Patterson, "A New Golden Age for Computer Architecture", CACM, 2019.