Introduction and Motivation

ENCS5331: Advanced Computer Architecture

Fall 2024/2025

Instructor: Dr. Ayman Hroub

Special Thanks to Dr. Muhamed Mudawar (KFUPM) and Dr. Onur Mutlu (ETHZ) for most of the Slides

Course Topics

- Introduction Motivation
- Review
- Superscalar Processors
- Memory Subsystem
- Bus Architectures
- Parallel Processors and Thread Level Parallelism
- Data Level Parallelism
- Domain Specific Architectures (DSAs)
- Virtual Memory
- Introduction to Near/In Memory Computing and Emerging Memory Technologies
- Introduction to Hardware Security

Grading Scheme

Assessment Type	Weight
Two Paper Review Assignments	10%
Midterm Exam	20%
Term Paper	30%
Final Exam	40%
Total	**100**

Learning Outcomes (1)

- ❖ Understand superscalar processors
- ❖ Understand the processor's memory hierarchy design options and performance optimization.
- ❖ Understand parallel computing architectures, cache coherency and bus architectures
- ❖ Understand data level parallelism (DLP) and thread-level parallelism (TLP)
- ❖ Design Domain Specific Architectures (DSA's)

Learning Outcomes (2)

- ❖ Understand the modern trends in computer architecture and technology
- ❖ Understand the principles and technologies of in/near memory computing
- ❖ Aware of computer architecture research community, top journals, top conferences, and research trends
- ❖ Conduct research in computer architecture and write a research paper

Textbook

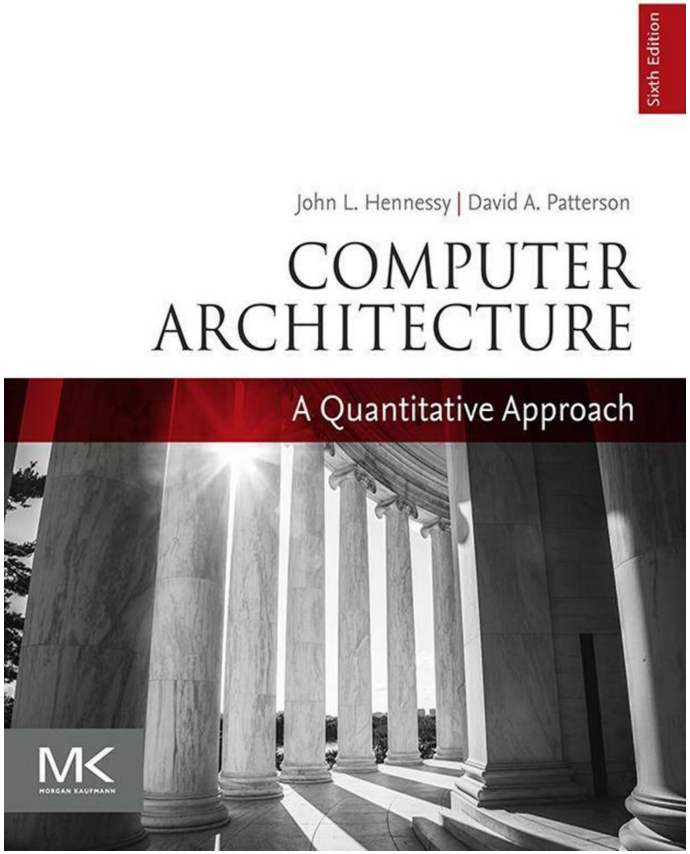
❖ Computer Architecture

A Quantitative Approach

✧ Sixth Edition

✧ John Hennessy & David Patterson

✧ Morgan Kaufmann Publishers, 2019



Other References

- ❖ Computer Organization and Design: The Hardware/Software Interface
  - ✧ Author(s) John L. Hennessy and David A. Patterson
  - ✧ Publisher Morgan Kaufmann
  - ✧ Edition 6th Edition (2021)
- ❖ Research Papers from the top conferences and journals
- ❖ Handouts/presentations provided by the instructor

What is the Relation between Teaching and Research?

- ❖ Teaching drives research
- ❖ Research drives teaching

Four Key Directions

- Fundamentally Secure/Reliable/Safe Architectures

- Fundamentally Energy-Efficient Architectures
  - Memory-centric (Data-centric) Architectures

- Fundamentally Low-Latency and Predictable Architectures

- Architectures for AI/ML, Genomics, Medicine, Health

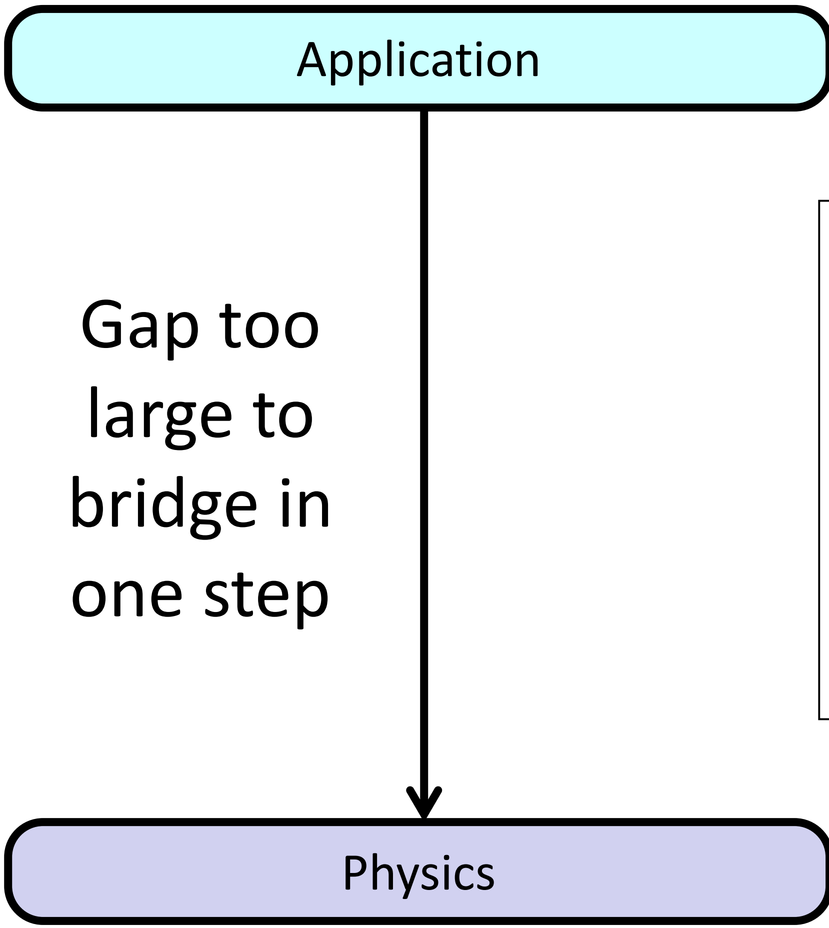
Why Do We Do Computing?

- To Solve Problems
- To Gain Insight (Richard Hamming)
- To Enable a Better Life & Future

How Does a Computer Solve Problems?

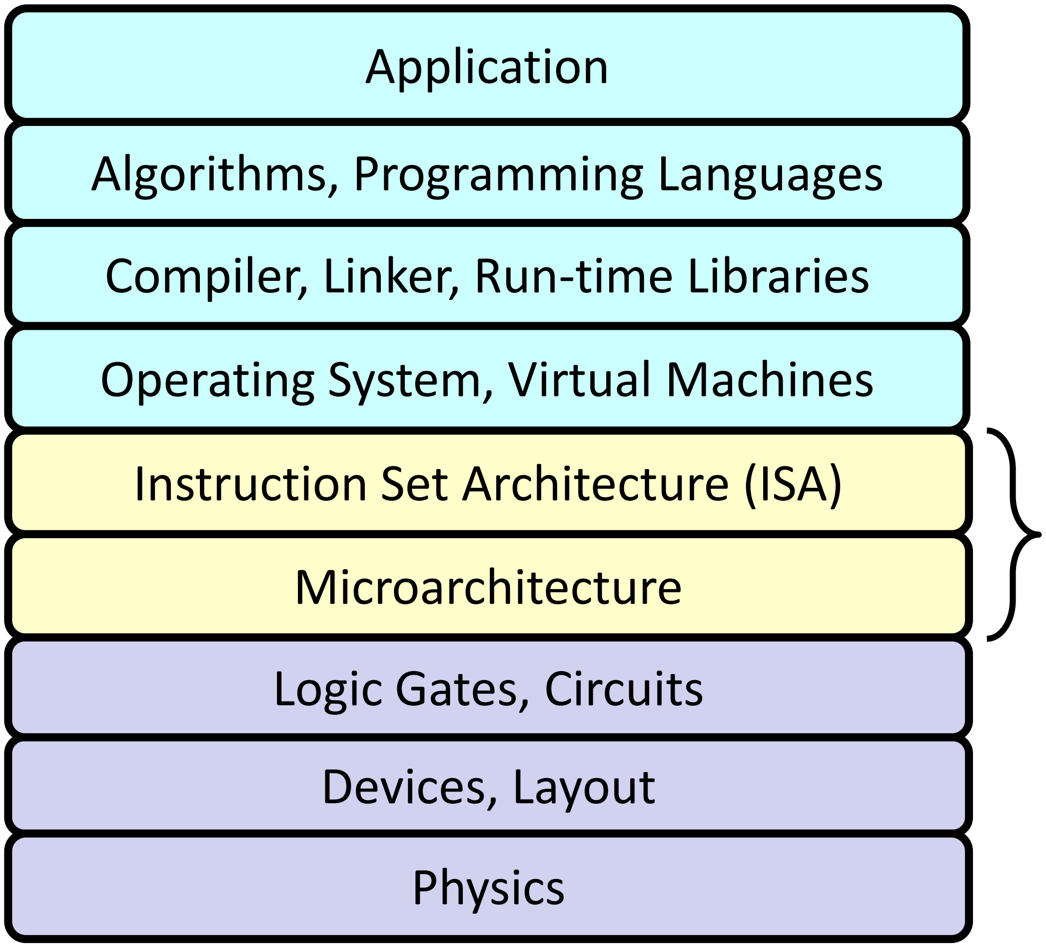
- Orchestrating Electrons

Abstraction Layers in Computing



How do we ensure that application programs are solved correctly by electrons?

Abstraction Layers in Computing



Focus of this course
+ Impact on
Software and Compiler

So, What is Computer Architecture?

Let's Start with Some Puzzles

a.k.a. Computer Architecture resembles Building Architecture

What Is This?

Source: https://www.flickr.com/photos/tambako/2286064777/in/photostream/

Gare do Oriente, Lisbon

Source: By Martín Gómez Tagle - Lisbon, Portugal, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=13764903

What About This?

Computer Architecture Definition

- Original Definition:

  The attributes of a [computing] system as seen by the programmer, i.e., the conceptual structure and functional behavior, as distinct from the organization of the data flows and controls, the logic design, and the physical implementation. (Amdahl, Blaaw, and Brooks, 1964)

  Today, this is known as Instruction-Set Architecture

Other Definitions

- Computer architecture is a set of rules and methods that describe the functionality, organization, and implementation of computer systems **[Cornell University]**

- Computer architecture is the science and art of designing computer platforms (hardware, system software, and programming model)

ISA vs. Microarchitecture

- Instruction Set Architecture (ISA)
  - **Class of ISA:** register-memory or register-register architectures
  - Programmer visible state (Register and Memory)
  - **Addressing Modes:** how memory addresses are computed
  - Data types and sizes for integer and floating-point operands
  - Instructions, encoding, and operation
  - Exception and Interrupt semantics
- Microarchitecture / Organization
  - Tradeoffs on how to implement the ISA for speed, energy, cost
  - Pipeline width and depth, cache organization, peak power, bus width, execution order, memory hierarchy, etc.

The Power of Abstraction

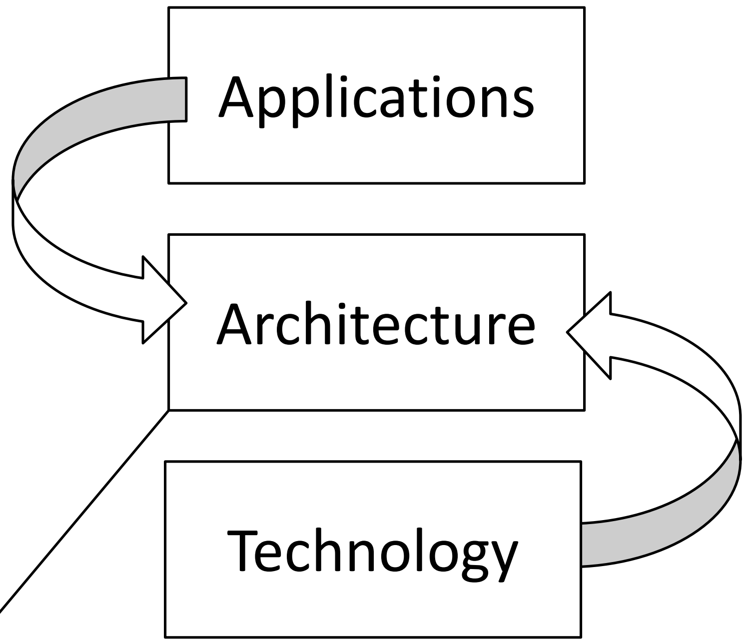
- Abstraction:
  - A higher level only needs to know about the interface to the lower level, not how the lower level is implemented
  - **Example:** a high-level language programmer does not need to know what the ISA is and how a computer executes instructions
- Abstraction improves productivity
  - No need to worry about decisions made in underlying levels
  - **Example:** programming in Java vs. C vs. assembly vs. binary vs. specifying control signals of each transistor every cycle
- Then, why would you want to know what goes on underneath or above?

Crossing the Abstraction Layers

- As long as everything goes well, not knowing what happens underneath or above is not a problem.
- What if
  - The program you wrote is running slow?
  - The program you wrote consumes too much energy?
- What if
  - The hardware you designed is too hard to program?
  - The hardware you designed is too slow because it does not provide the right primitives to the software?
- One goal of this course is to understand how a processor works underneath the software layer and how decisions made in hardware affect the programmer

Architecture Continually Changing

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



Improved technologies make new architectures possible

Compatibility: Ability of a new architecture to run older applications. Cost of software development makes compatibility a major force in market.

Changing Definition

- Computer Architecture's Changing Definition
- 1950s to 1960s:
  - Computer Architecture Course = Computer Arithmetic
- 1970s to mid 1980s:
  - Computer Architecture Course = Instruction Set Design,

    Especially ISA appropriate for compilers
- 1990s until today:
  - Computer Architecture Course =

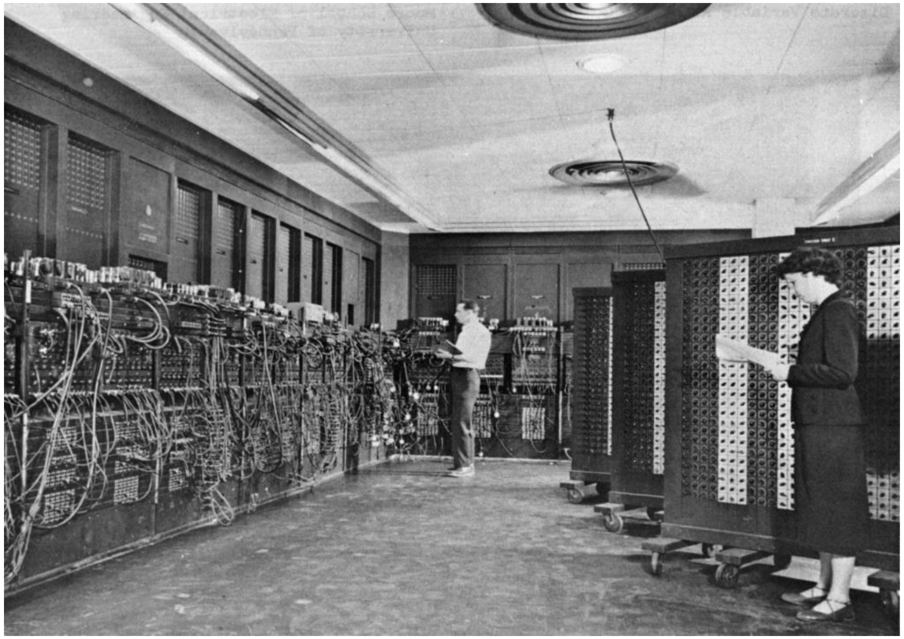
    Design of CPU, memory system, I/O system, Multiprocessors

Again, What is Computer Architecture?

Application
Algorithms, Programming Languages
Compiler, Linker, Run-time Libraries
Operating System, Virtual Machines
Instruction Set Architecture (ISA)
Microarchitecture
Logic Gates, Circuits
Devices, Layout
Physics

In its broadest definition, computer architecture is the *design, organization, and implementation of a computing system* that allows us to execute software applications efficiently using available manufacturing technologies, meeting price, power, and performance goals.

Computers Then …



ENIAC : Electronic Numerical Integrator And Computer

Built by Eckert and Mauchly at the University of Pennsylvania (1943-45)

First general-purpose electronic computer.

Cost \$500,000 (\$6,300,000 today)

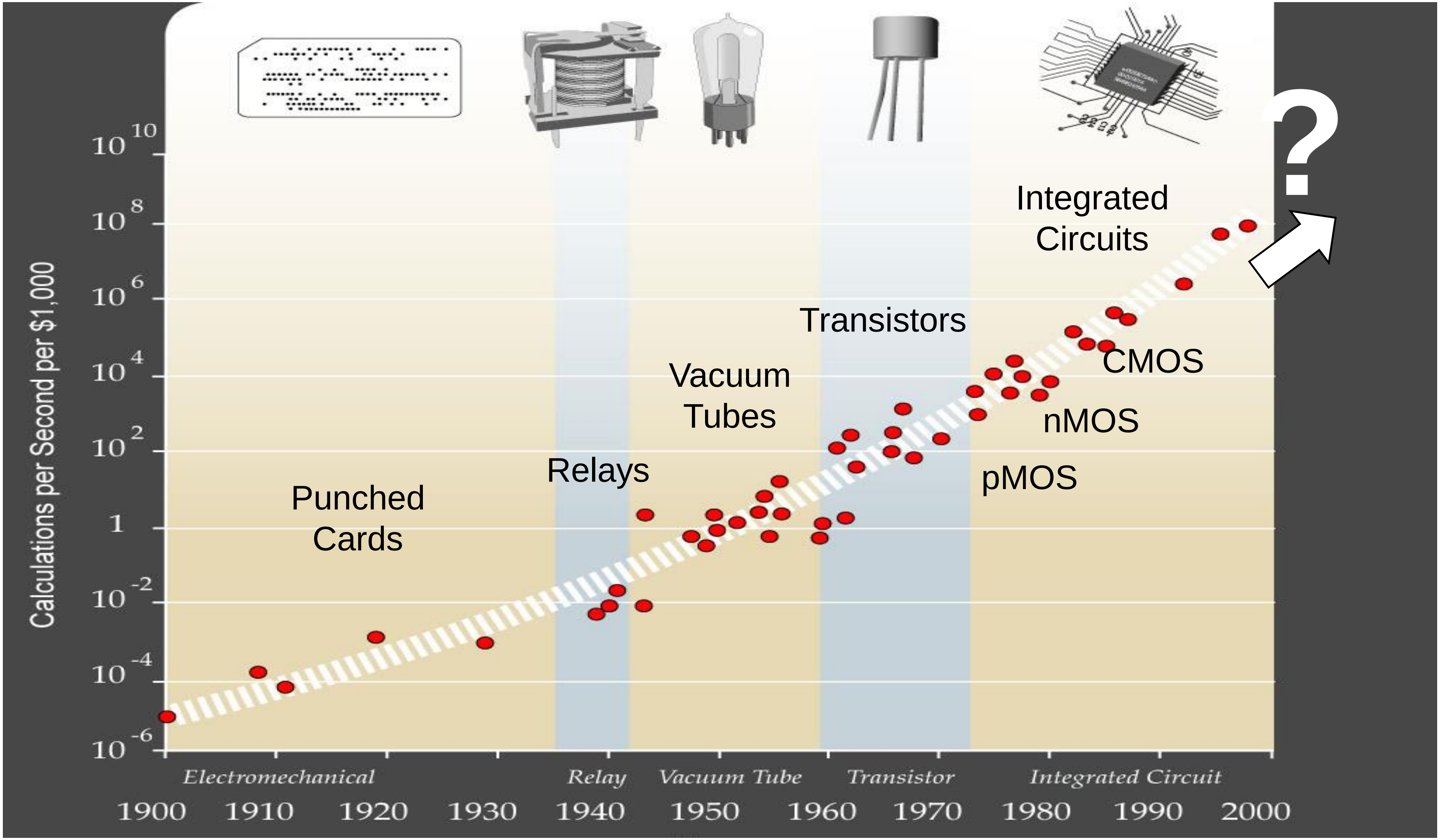
Around 18000 vacuum tubes, 7200 diodes, 1500 relays, 70000 resistors, 10000 capacitors, and around 5 million hand-soldered joints.

30 tons, 167 square meters, consumed 150 kw of power

Addition = 200 µs, division = 6 ms, read-in 120 cards per minute

Not very reliable (one vacuum tube fails about every two days)

Major Technology Inventions



What Next? … Beyond CMOS

- Spin wave computing
- Phase Change Memory (PCM)
- Resistive memories
- Photonics and optical computing
- Superconducting computing
- Graphene nanoribbons
- Molecular electronics
- Quantum computing
- Neuromorphic computing
- FeFET Transistors
- etc.

Classes of Computers (1)

Different Platforms, Different Goals

Source: http://www.sia-online.org (semiconductor industry association)

Different Platforms, Different Goals

Source: https://iq.intel.com/5-awesome-uses-for-drone-technology/

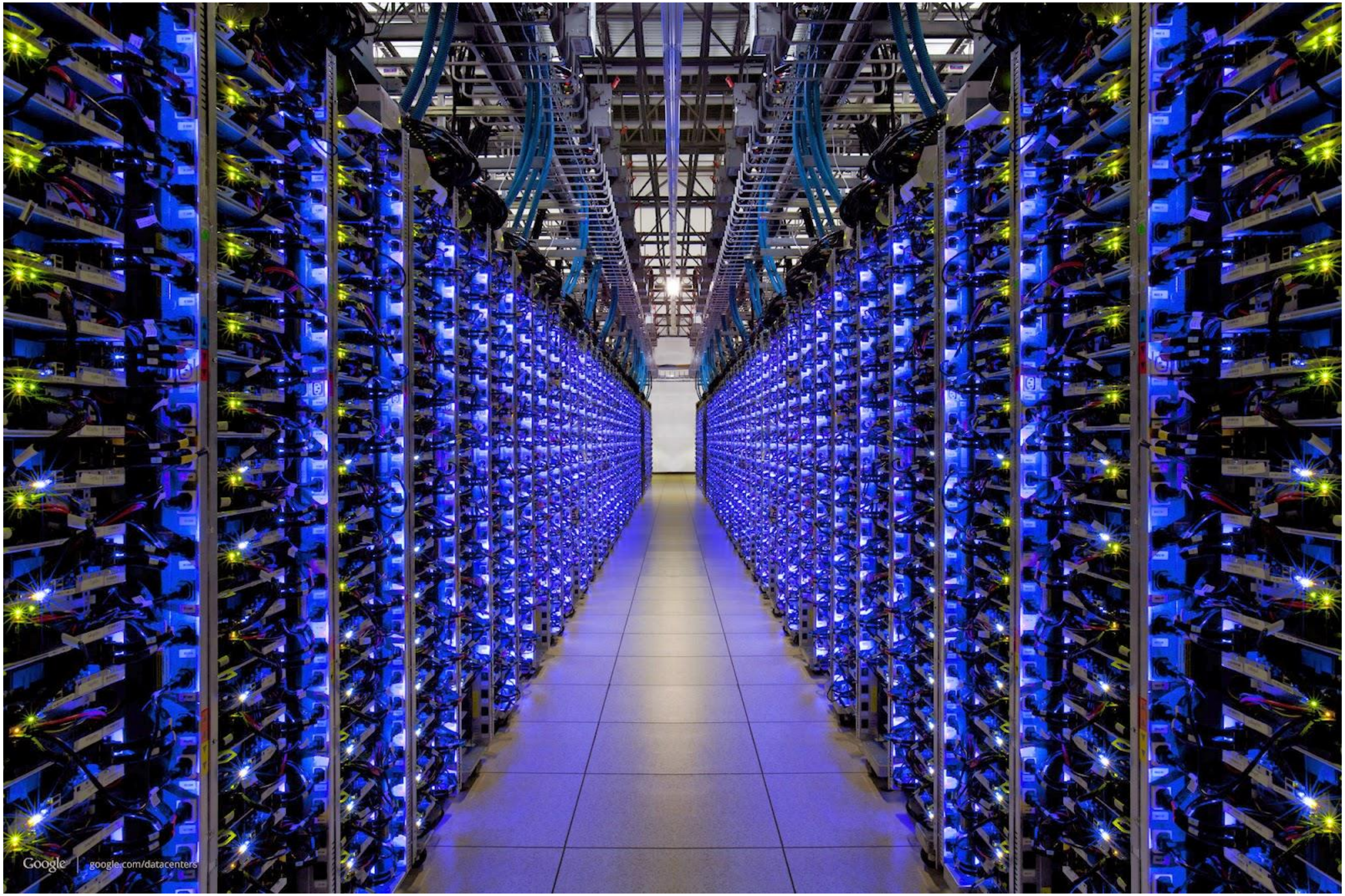
Different Platforms, Different Goals

Source: https://taxistartup.com/wp-content/uploads/2015/03/UK-Self-Driving-Cars.jpg

Different Platforms, Different Goals

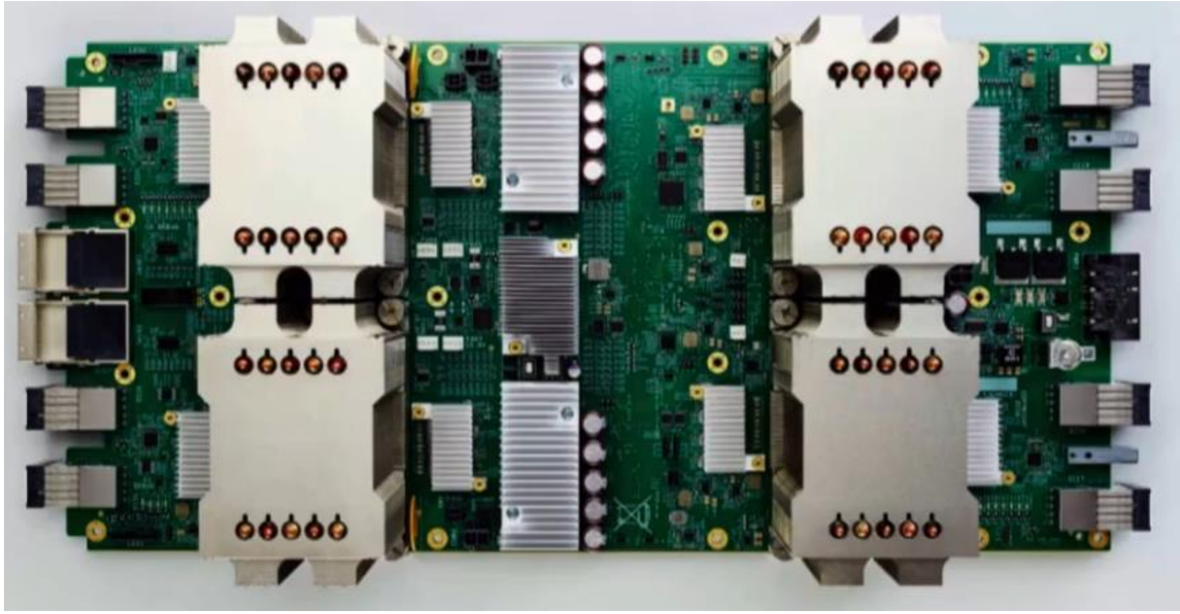
Source: https://fossbytes.com/wp-content/uploads/2015/06/Supercomputer-TIANHE2-china.jpg

Different Platforms, Different Goals



Source: http://datacentervoice.com/wp-content/uploads/2015/10/data-center.jpg

Google TPU Generation II (2017)



https://www.nextplatform.com/2017/05/17/first-depth-look-googles-new-second-generation-tpu/

4 TPU chips
vs 1 chip in TPU1

High Bandwidth Memory
vs DDR3

Floating point operations
vs FP16

45 TFLOPS per chip
vs 23 TOPS

Designed for training
and inference
vs only inference

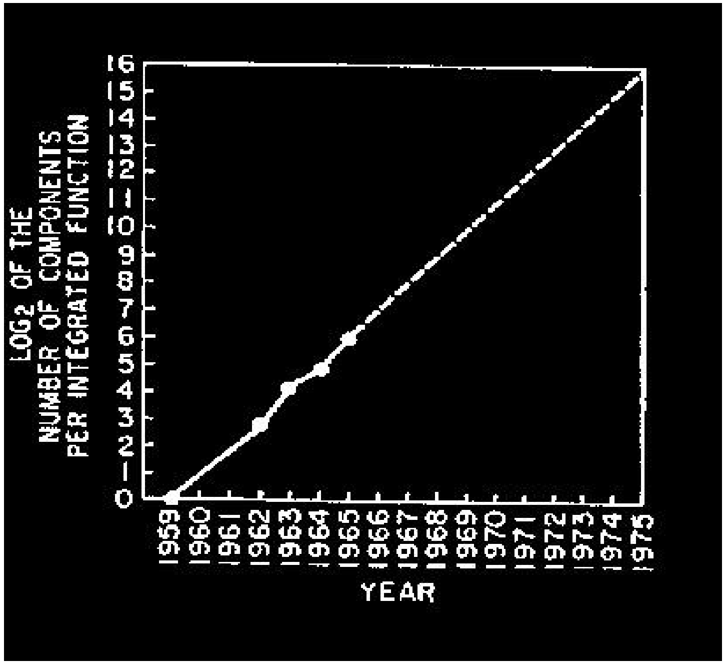
Classes of Computers (2)

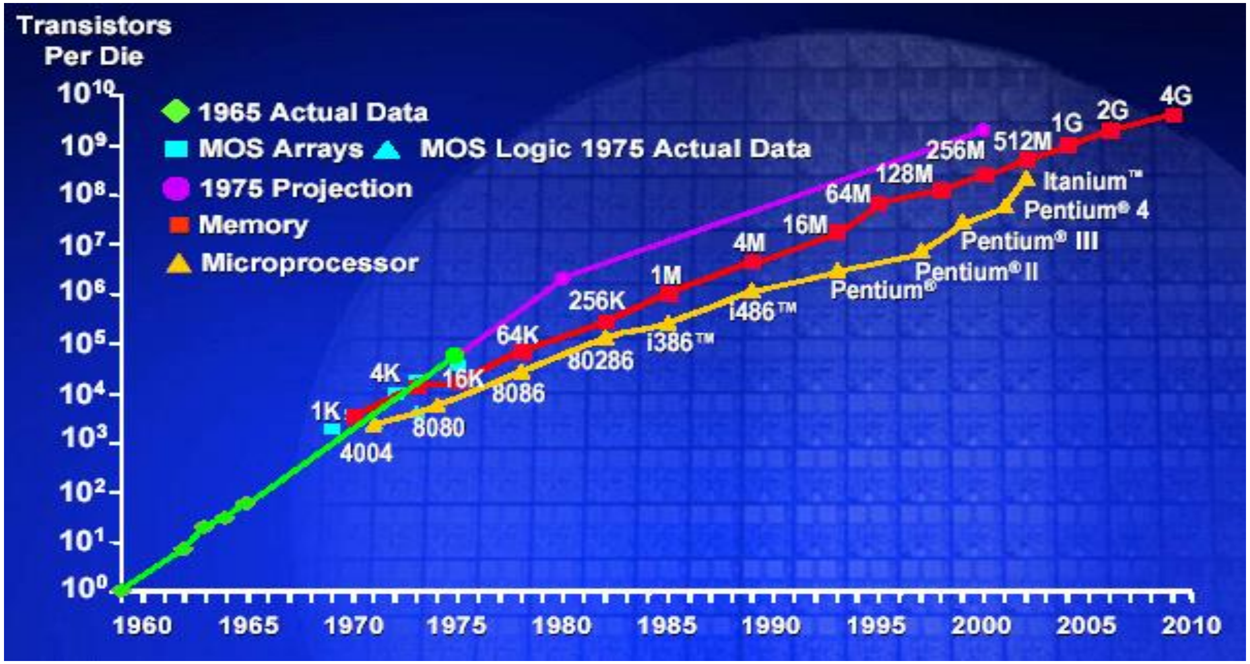
- Personal Mobile Devices
  - Cell phones and tablet computers, battery-operated
  - Emphasis on energy efficiency, real-time performance, and cost
  - Use of flash memory storage for energy and size requirement
- Desktop Computers
  - Battery-operated laptop computers to high-end workstations
  - Emphasis on price-performance and graphics performance
- Servers
  - Backbone of large-scale enterprise computing
  - Emphasis on availability, scalability, and performance
  - Failure of a server is more catastrophic than a desktop computer

Classes of Computers (3)

- ❖ Clusters / Warehouse Scale Computers
  - ✧ Clusters are collections of servers connected by a network
  - ✧ Example: search engine, social networking, online shopping
  - ✧ Used for “Software as a Service (SaaS)”
  - ✧ Emphasis on availability and price-performance
  - ✧ Supercomputers are related but the emphasis is on floating-point performance and fast internal networks
- ❖ Embedded Computers / Internet of Things (IoT)
  - ✧ Found everywhere: printers, networking devices, autonomous cars, flying UAVs, game players, digital cameras, TVs, etc.
  - ✧ Widest spread of processing power and cost

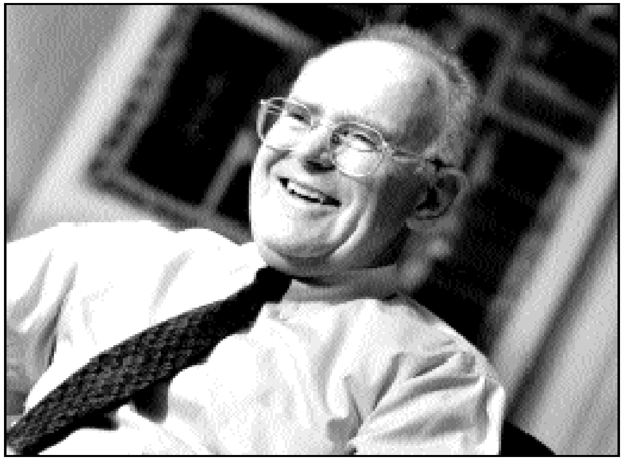
Moore's Law





Cramming More Components onto Integrated

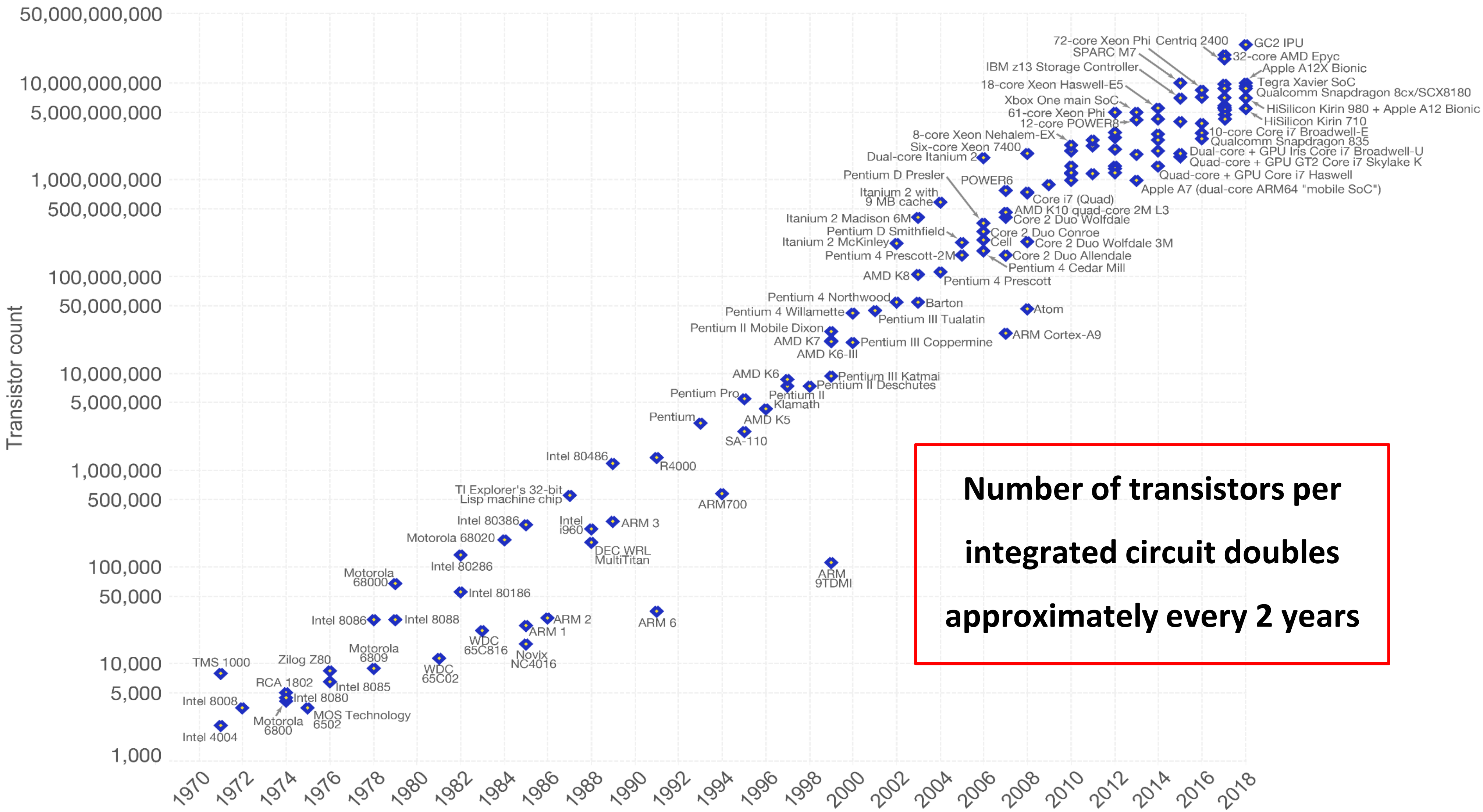
Circuits, Gordon Moore, Electronics, 1965



Number of transistors per integrated circuit

doubles every N months (12 ≤ N ≤ 24)

Moore's Law: Transistor Count on IC chips



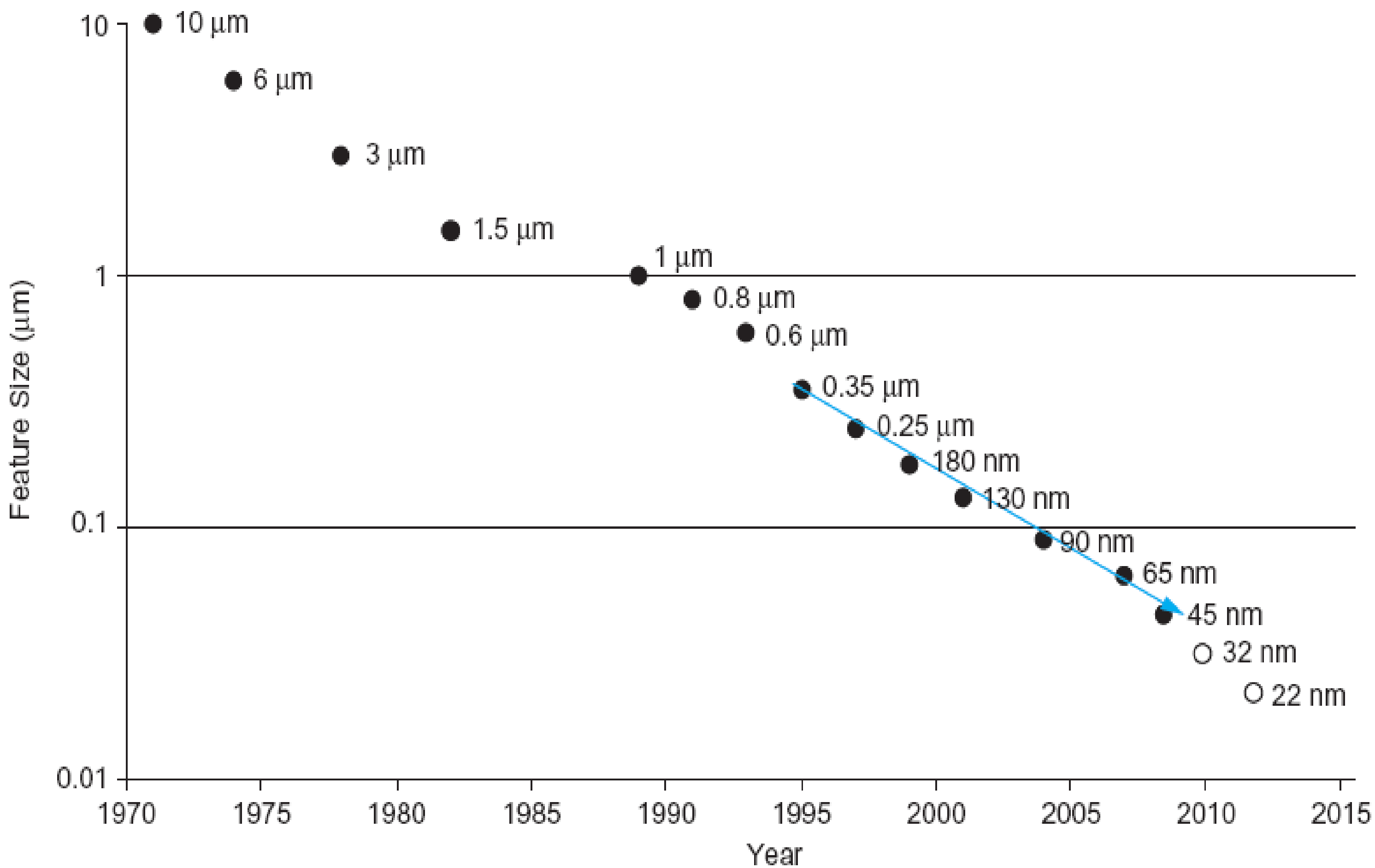
Number of transistors per integrated circuit doubles approximately every 2 years

Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

Feature Size Scaling

❖ Feature Size: minimum gate length of a transistor (source to drain)

- ✧ Transistors become cheaper
- ✧ Transistors become faster and lower power
- ✧ **Wires do not improve and may get worse**

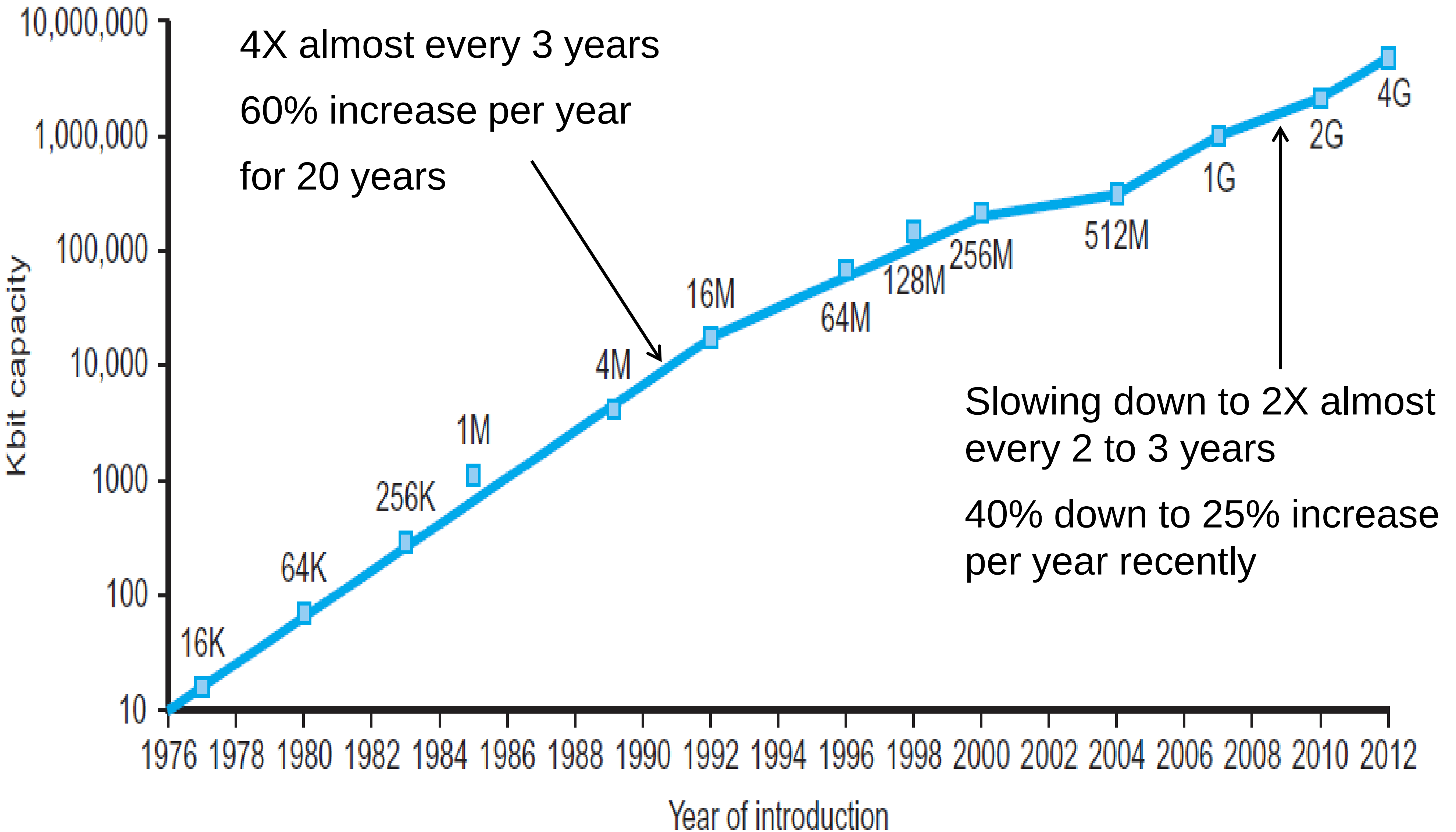


Source: Weste and Harris, CMOS VLSI Design Lecture Slides

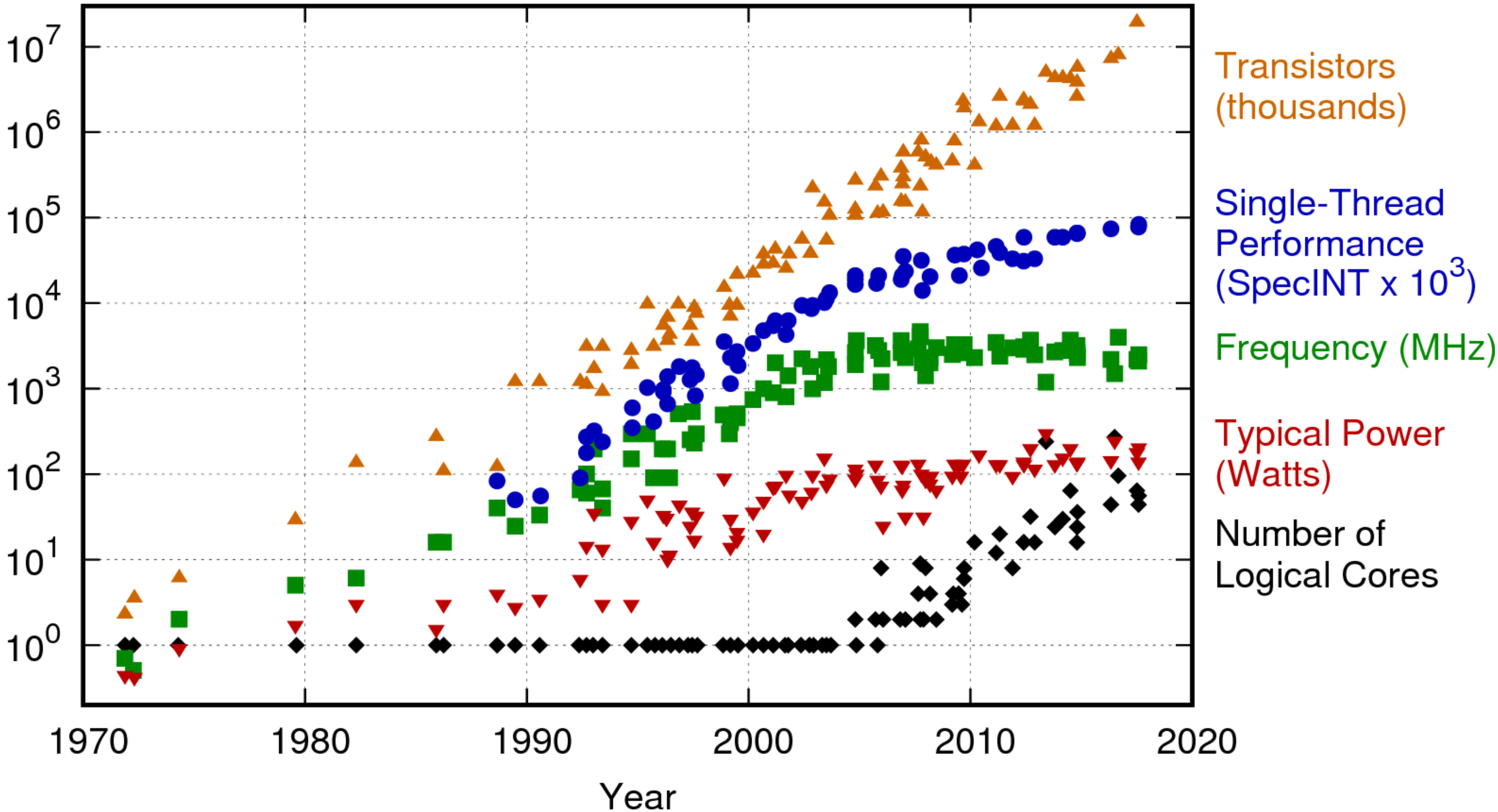
Technology Trends

- Integrated circuit technology
  - Transistor density: increases 35% per year (Moore's Law)
  - Die size: increases 10% to 20% per year (less predictable)
  - Integration overall: increases 40% to 55% per year
- DRAM capacity: 40% down to 25% per year (2000 to 2014)
  - **Recently slowing down:** 8 Gbits (2014), 16 Gbits (2019), 32 Gbits ???
- Flash storage capacity: increases 50% to 60% per year
  - 8X to 10X cheaper per bit than DRAM
- Magnetic disk capacity: 40% per year (2004 to 2011)
  - **Recently slowing down to only 5% per year**
  - 10X cheaper per bit than Flash
  - 100X cheaper per bit than DRAM

Growth of Capacity per DRAM Chip



Microprocessor Trend



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

Example of a Multicore Processor

IBM Power8 Processor

22 nm, 650 mm2 die

12 cores

16 exec pipe per core

Cache Hierarchy

32 KB Instruction Cache

64 KB Data Cache

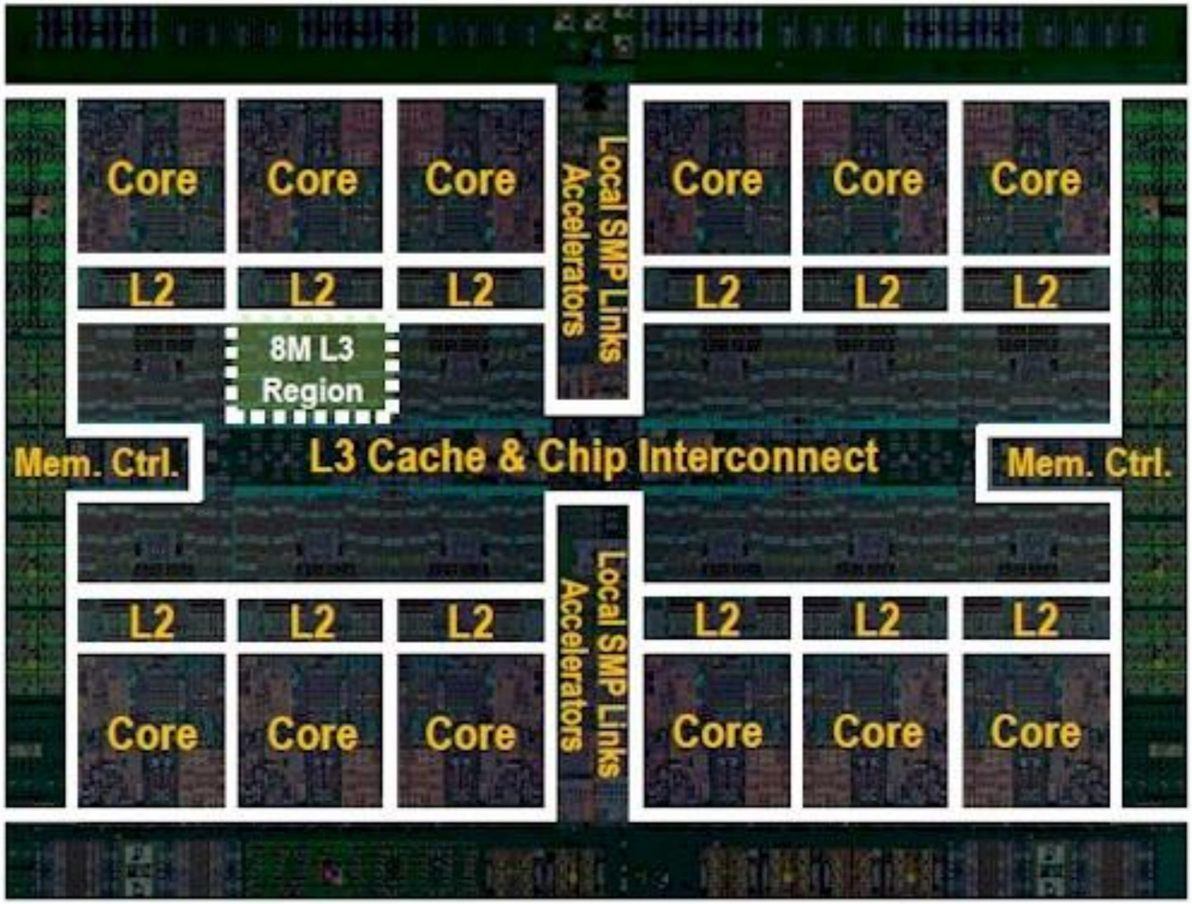
512 KB L2 per core

96 MB eDRAM shared L3

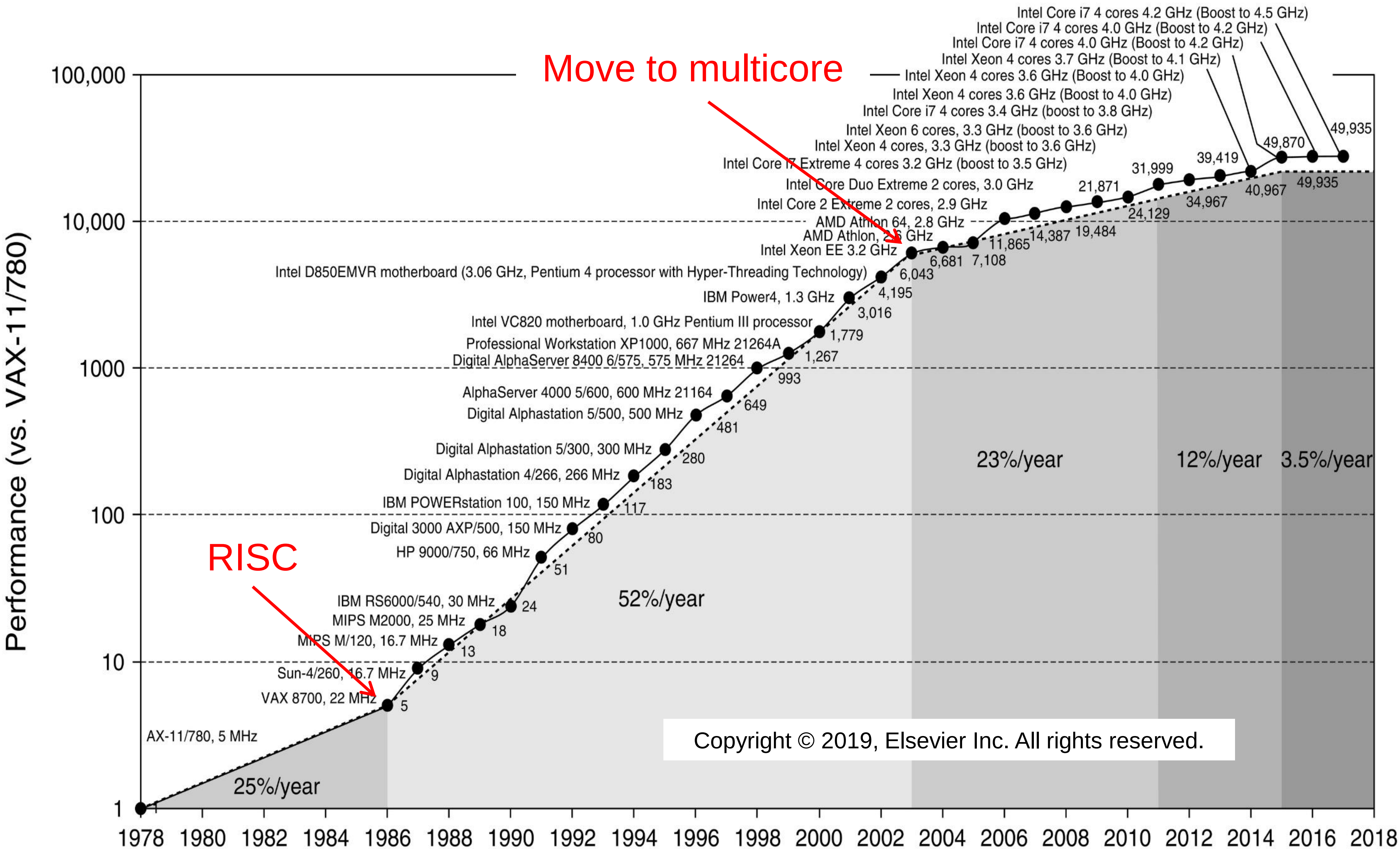
Memory Interface

Dual Memory controllers

230 GB/sec bandwidth



Processor Performance over 40 Years

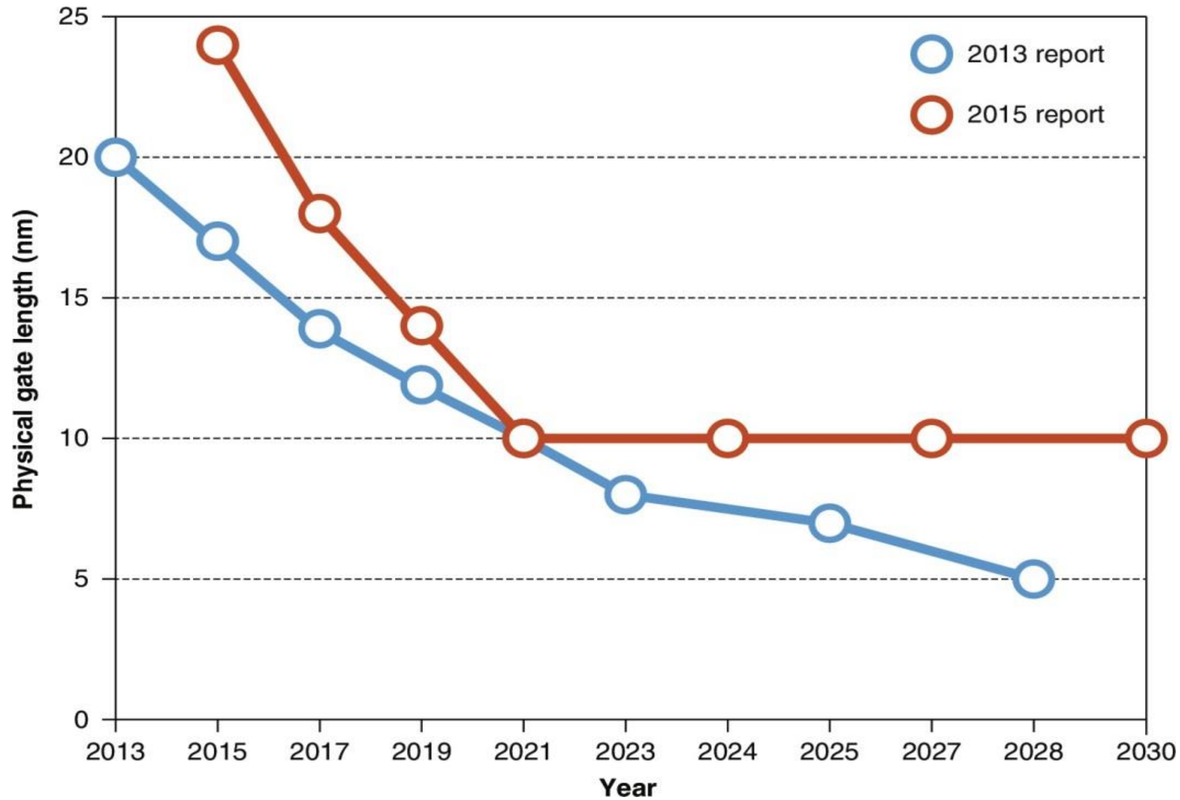


The End of the Uniprocessor Era

- Late 1970's saw the emergence of the microprocessor
- New RISC architectures appeared in early 1980's
  - RISC = Reduced Instruction Set Computer
- RISC architectures raised performance using
  - Instruction Level Parallelism through pipelining & multiple-issue
  - Cache memory reduced memory latency
  - Instruction execution pipelines and caches improved clock rates
  - 17 years of improved performance at a rate of ~50% per year
- Single processor performance dropped in 2003
  - Lack of more Instruction Level Parallelism (ILP) to exploit efficiently
  - Power dissipation put a limit on higher clock frequencies

The End of Moore's Law

- If Moore's law was to continue until 2050, engineers have to build transistors that are smaller than a hydrogen atom!
- Moore's law might end before 2030. However, transistors will keep getting better and more energy efficient.
- Only four companies can build state of the art chips:
  - Global Foundries
  - Intel
  - Samsung
  - TSMC



Parallelism in Computer Architecture

Computer Architecture can exploit parallelism in four ways:

1. Instruction-Level Parallelism (ILP)
   - ✧ Through pipelining and multiple instruction issue each cycle
   - ✧ Limited ILP in real programs, difficult to exploit efficiently
2. Data-Level Parallelism (DLP)
   - ✧ Single instruction operates on multiple data (SIMD)
   - ✧ Vector architectures and graphics processing units (GPUs)
3. Thread-Level Parallelism (TLP)
   - ✧ Can execute parallel threads on multiple cores (or same core)
   - ✧ Parallel threads communicate through shared memory
4. Request-Level Parallelism
   - ✧ Parallelism among independent processes specified by the OS

Conventional Wisdom in Computer Architecture

- Old Conventional Wisdom: Power is free, Transistors are expensive.
- New Conventional Wisdom: "Power wall" Power is expensive, Transistors are free (Can put more on chip than can afford to turn on)
- Old CW: Multiplication is slow, Memory access is fast
- New CW: "Memory wall" Memory is slow, multiplication is fast (200 clock cycles to DRAM memory, 4 clock cycles for multiplication)
- Old CW: Sufficiently increasing Instruction Level Parallelism via compilers, pipelining, out-of-order execution, speculation, …
- New CW: "ILP wall" law of diminishing returns on more HW for ILP
- Old CW: Uniprocessor performance 2X in 1.5 years
- New CW: Uniprocessor performance now 2X in ~7 years

  Change in chip design: switching to multiple "cores"
- Simpler processors are more energy efficient than complex ones

And in Conclusion …

- ❖ Computer Architecture is more than ISAs and RTL
- ❖ It is about the interaction of hardware and software
- ❖ Design of appropriate abstraction layers
- ❖ Computer architecture is shaped by technology and applications
  - ✧ History provides lessons for the future
- ❖ Going from sequential to parallel computing
  - ✧ Requires innovation in many fields, including computer architecture

Research Proposal Outline

- The Problem: What is the problem you are trying to solve
  - Define clearly.
- Novelty: Why has previous research not solved this problem? What are its shortcomings?
  - Describe/cite all relevant works you know of and describe why these works are inadequate to solve the problem.
- Idea: What is your initial idea/insight? What new solution are you proposing to the problem? Why does it make sense? How does/could it solve the problem better?
- Hypothesis: What is the main hypothesis you will test?
- Methodology: How will you test the hypothesis/ideas? Describe what simulator or model you will use and what initial experiments you will do.
- Plan: Describe the steps you will take. What will you accomplish by Milestone 1, 2, 3, and Final Report? Give 75%, 100%, 125% and moonshot goals.

All research projects can be and should be described in this fashion.