Computer Architecture / Security / Quantum / Where to Go Next

Yipeng Huang

Rutgers University

April 25, 2024

Computer Architecture

- ***Instruction level parallelism***
- Limitations of instruction level parallelism

- Data level parallelism
- Limitations of data level parallelism

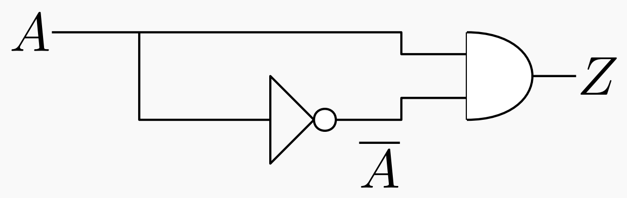
- Thread level parallelism
- Limitations of thread level parallelism

- Accelerators
- Quantum

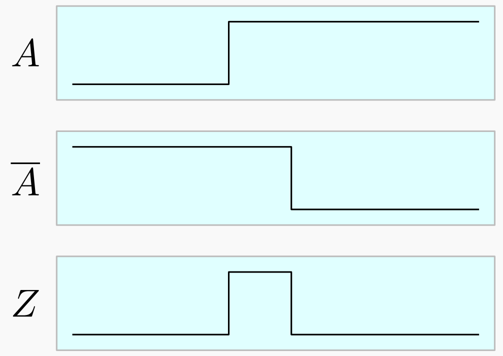
Synchronization. Clock signals.

Timing

Circuit:

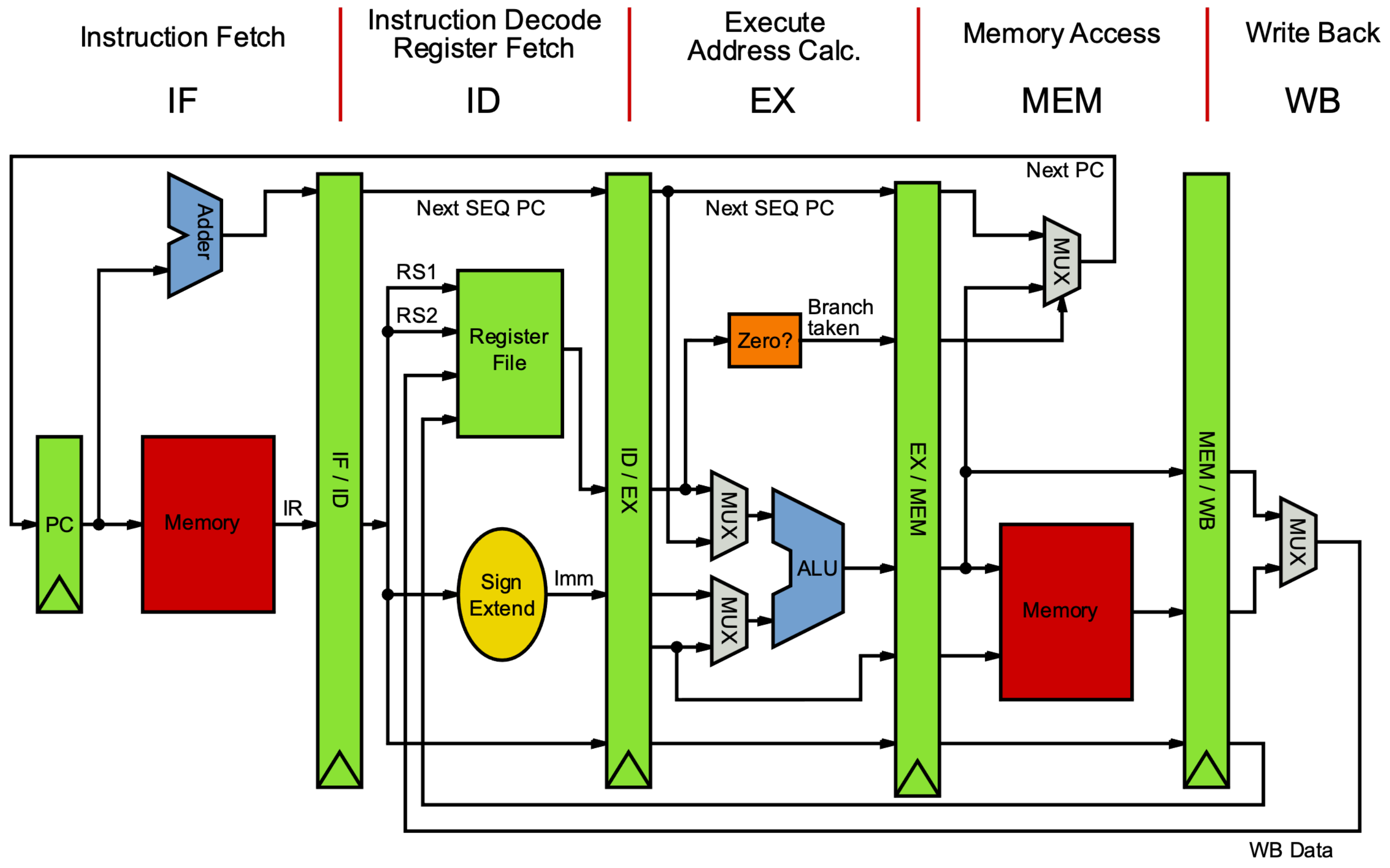


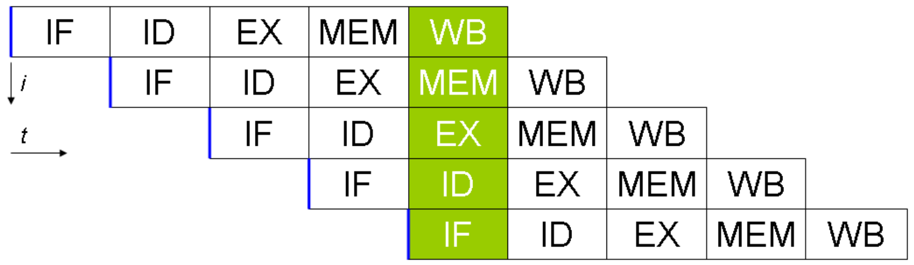
Voltages over time:



(A AND NOT A) should always evaluate to FALSE, need a clock signal to denote when values are valid, and to ignore transients.

The journey of an assembly instruction: pipeline





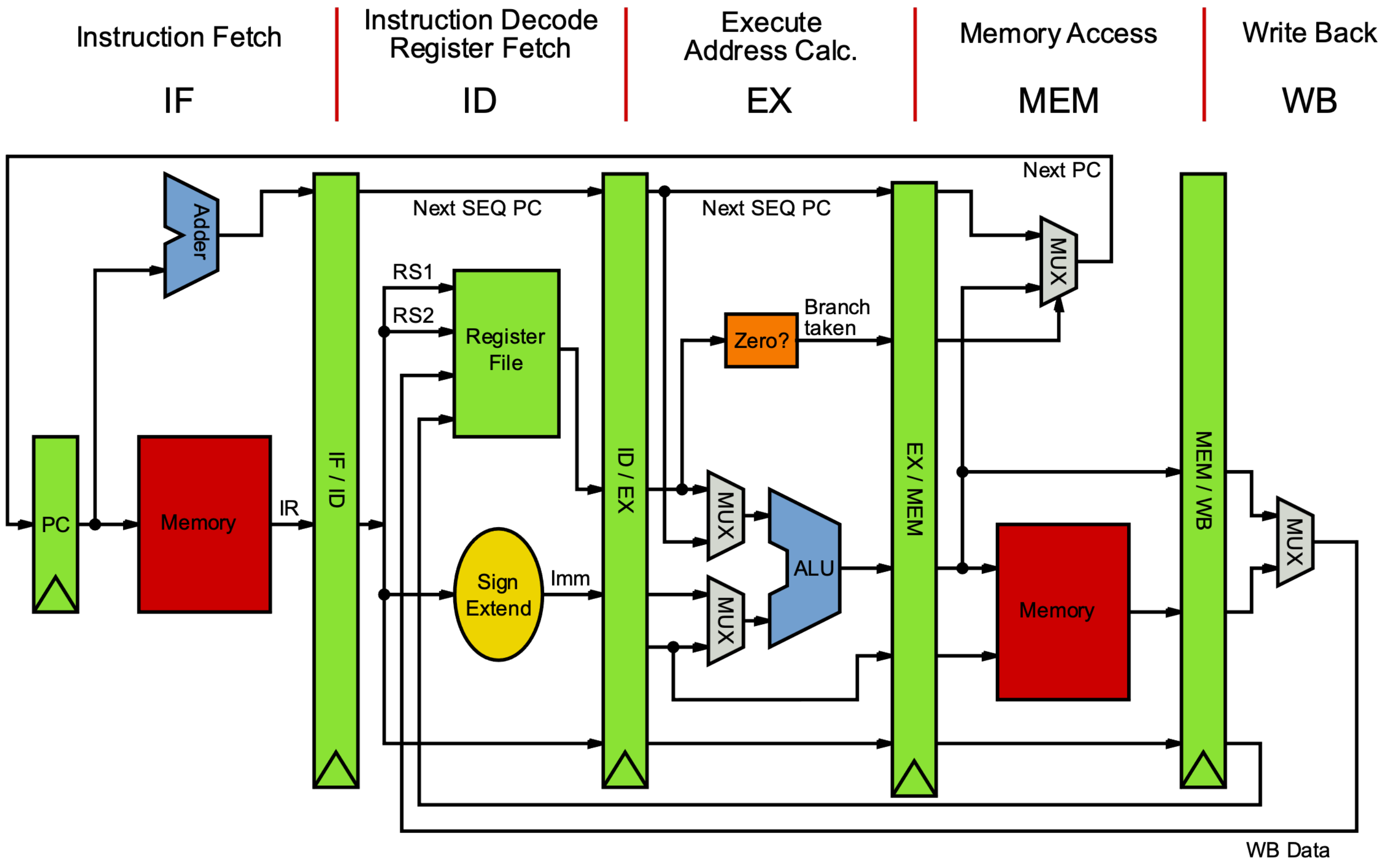
IF	ID	EX	MEM	WB				
	IF	ID	EX	MEM	WB			
		IF	ID	EX	MEM	WB		
			IF	ID	EX	MEM	WB	
				IF	ID	EX	MEM	WB

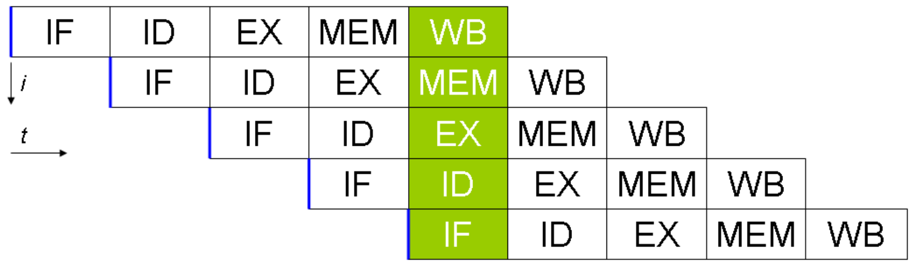
An assembly instruction needs multiple stages of combinational and sequential logic to execute.

At any given moment, several assembly instructions are in-flight.

Image sources: Wikimedia.

The journey of an assembly instruction: pipeline





IF	ID	EX	MEM	WB				
	IF	ID	EX	MEM	WB			
		IF	ID	EX	MEM	WB		
			IF	ID	EX	MEM	WB	
				IF	ID	EX	MEM	WB

Where does the cache logic in setAssociative.v live?

Where does the logic in binSub.v live?

What is the length of the combinational logic of binSub.v?

What determines the clock speed?

Microarchitectural support for Instruction Set Architectures

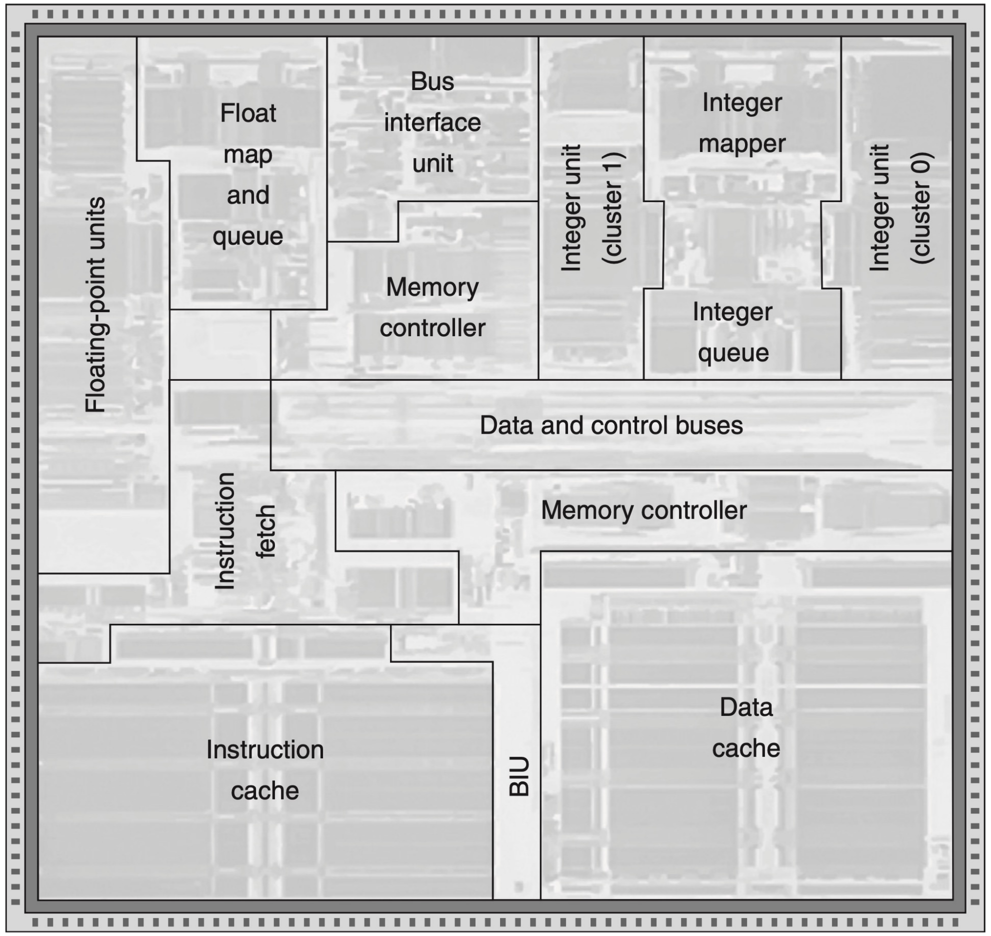
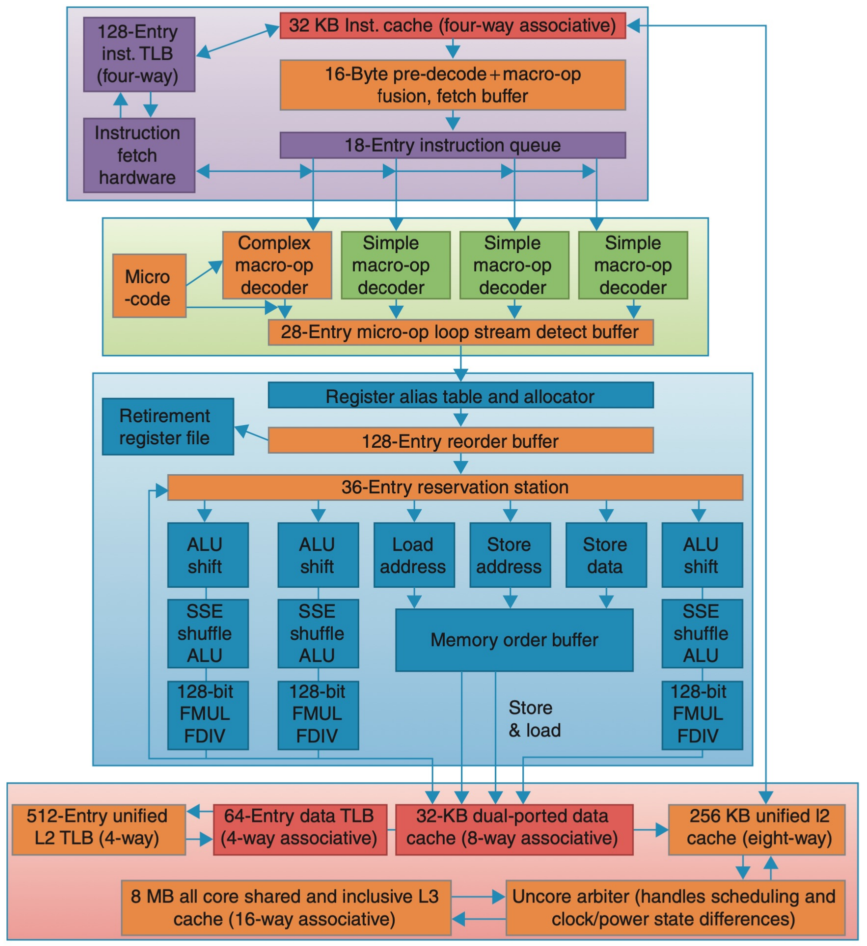


Figure 2.33 Floorplan of the Alpha 21264 [Kessler 1999].

Hennessy & Patterson. Computer Architecture: A Quantitative Approach

Instruction Level Parallelism: modern trickery



128-Entry inst. TLB (four-way)
32 KB Inst. cache (four-way associative)
16-Byte pre-decode+macro-op fusion, fetch buffer
Instruction fetch hardware
18-Entry instruction queue
Micro -code
Complex macro-op decoder
Simple macro-op decoder
Simple macro-op decoder
Simple macro-op decoder
28-Entry micro-op loop stream detect buffer
Register alias table and allocator
Retirement register file
128-Entry reorder buffer
36-Entry reservation station
ALU shift
ALU shift
Load address
Store address
Store data
ALU shift
SSE shuffle ALU
SSE shuffle ALU
Memory order buffer
SSE shuffle ALU
128-bit FMUL FDIV
128-bit FMUL FDIV
Store & load
128-bit FMUL FDIV
512-Entry unified L2 TLB (4-way)
64-Entry data TLB (4-way associative)
32-KB dual-ported data cache (8-way associative)
256 KB unified l2 cache (eight-way)
8 MB all core shared and inclusive L3 cache (16-way associative)
Uncore arbiter (handles scheduling and clock/power state differences)

Figure 3.41 The Intel Core i7 pipeline structure shown with the memory system components. The total pipeline depth is 14 stages, with branch mispredictions costing 17 cycles. There are 48 load and 32 store buffers. The six independent functional units can each begin execution of a ready micro-op in the same cycle.

- Superscalar instruction fetch
- Out-of-order instruction retirement
- Speculative execution

Hennessy & Patterson. Computer Architecture: A Quantitative Approach

Computer Architecture

- Instruction level parallelism
- ***Limitations of instruction level parallelism***

- Data level parallelism
- Limitations of data level parallelism

- Thread level parallelism
- Limitations of thread level parallelism

- Accelerators
- Quantum

Limitations of microarchitecture trickery: security vulnerabilities

- See Prof. Mark Hill (U. Wisconsin) slides on Meltdown and Spectre

- Lipp, Moritz, et al. "Meltdown: Reading kernel memory from user space." *27th {USENIX} Security Symposium ({USENIX} Security 18)*. 2018.

Computer Architecture

- Instruction level parallelism
- Limitations of instruction level parallelism

- ***<u>Data level parallelism</u>***
- ***<u>Limitations of data level parallelism</u>***

- Thread level parallelism
- Limitations of thread level parallelism

- Accelerators
- Quantum

Data Level Parallelism

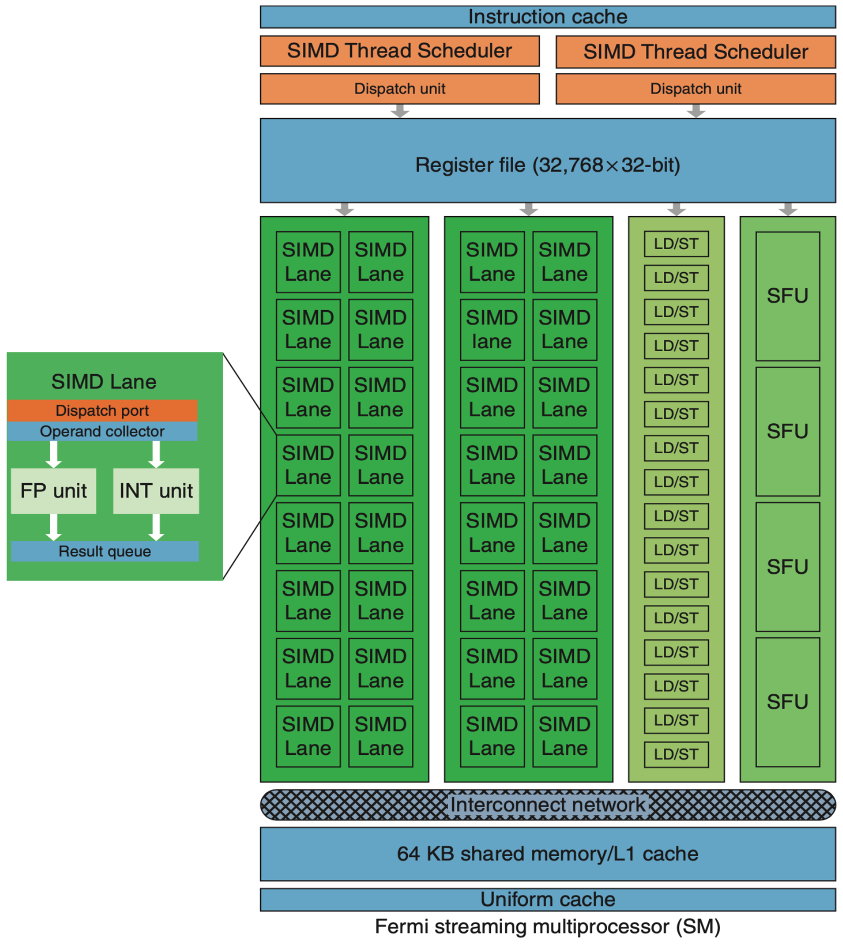
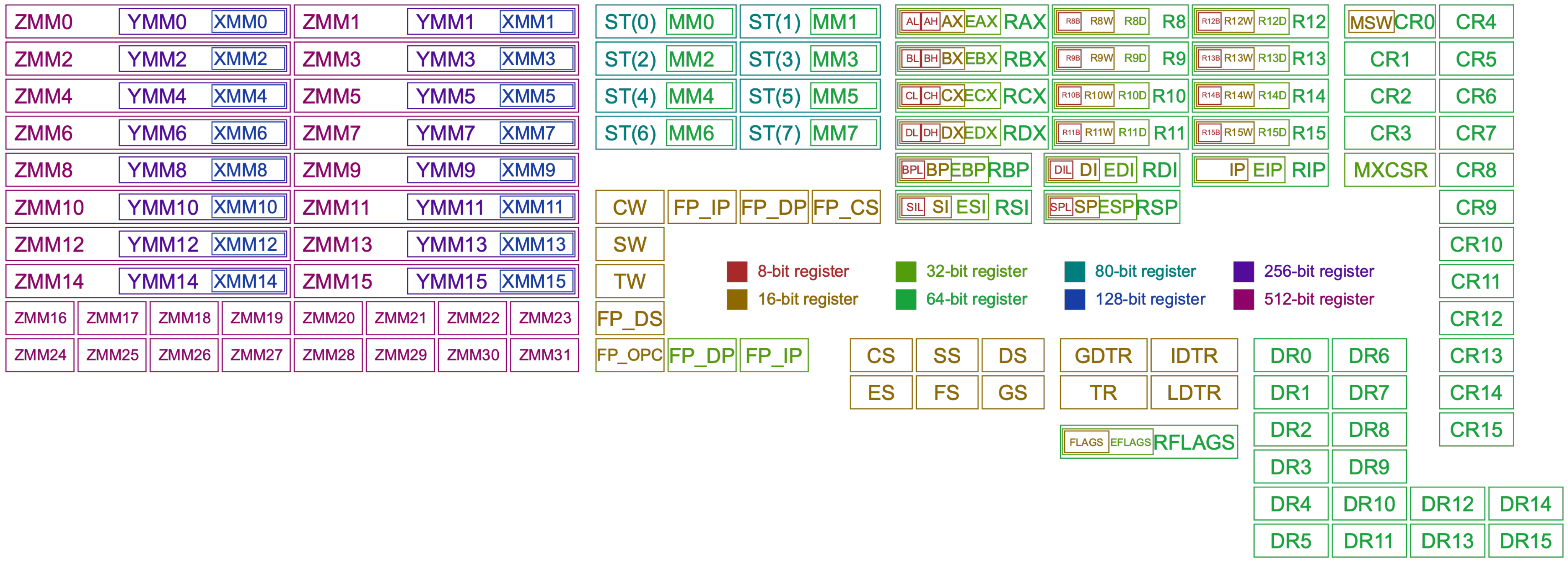


Figure 4.20 Block diagram of the multithreaded SIMD Processor of a Fermi GPU. Each SIMD Lane has a pipelined floating-point unit, a pipelined integer unit, some logic for dispatching instructions and operands to these units, and a queue for holding results. The four Special Function units (SFUs) calculate functions such as square roots, reciprocals, sines, and cosines.

Hennessy & Patterson. Computer Architecture: A Quantitative Approach

Data Level Parallelism



Wikimedia

Computer Architecture

- Instruction level parallelism
- Limitations of instruction level parallelism

- Data level parallelism
- Limitations of data level parallelism

- ***Thread level parallelism***
- ***Limitations of thread level parallelism***

- Accelerators
- Quantum

Dennard Scaling, Moore's Scaling, Power Wall

For a few decades, shrinking transistors drove clock speeds higher.

Dennard Scaling: Shrinking transistors also use less power.

Win-win.

Circa 2005, Dennard Scaling hit physical limitations.

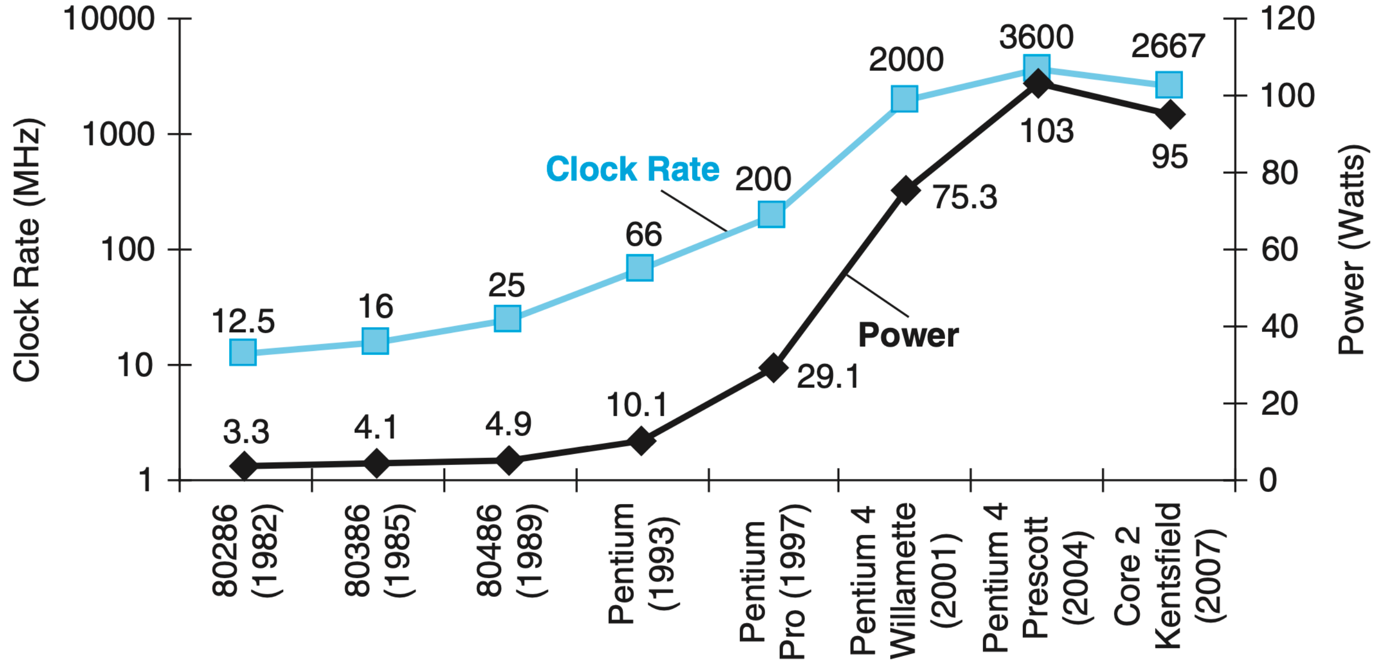
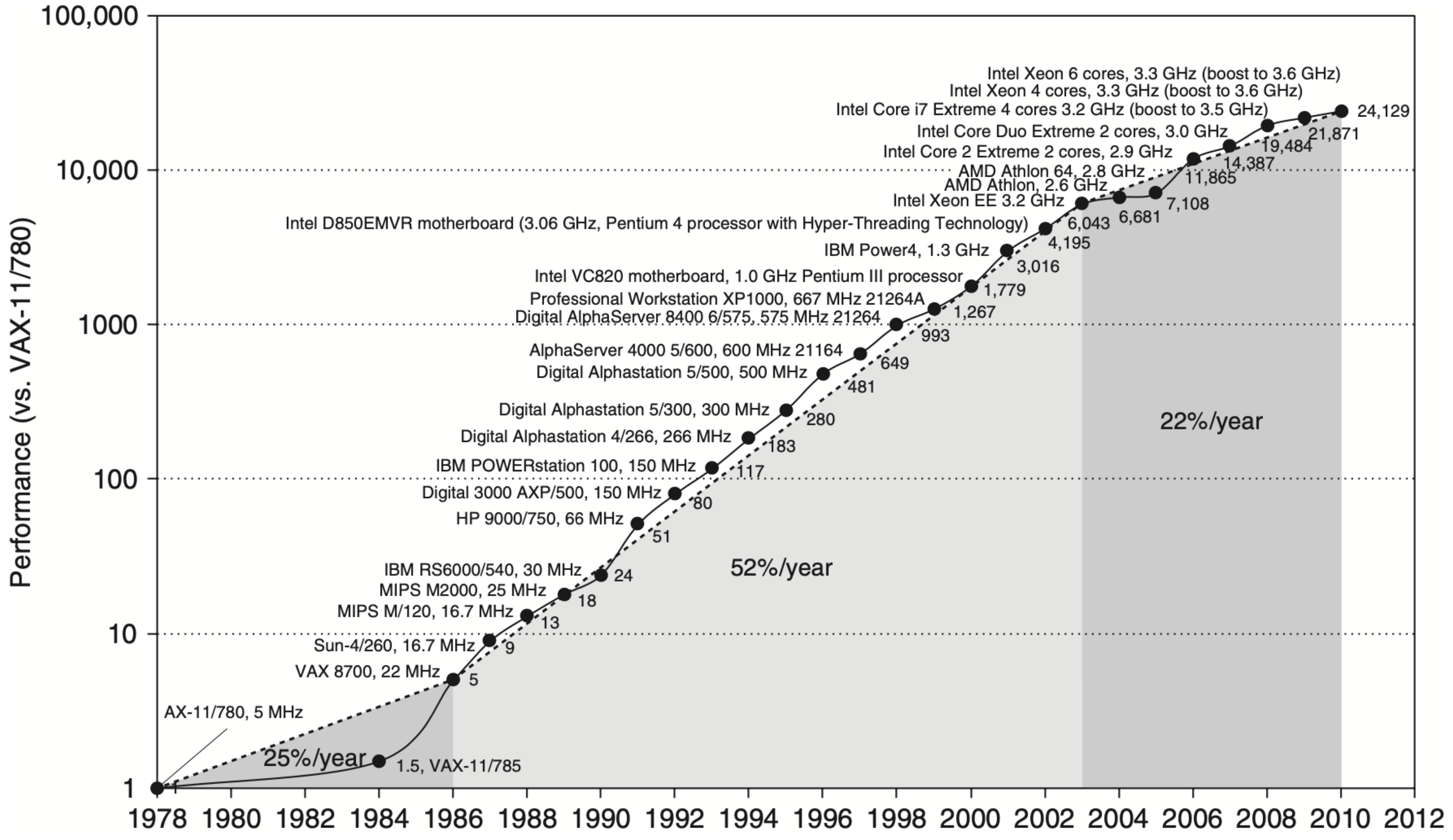


FIGURE 1.15 Clock rate and Power for Intel x86 microprocessors over eight generations and 25 years. The Pentium 4 made a dramatic jump in clock rate and power but less so in performance. The Prescott thermal problems led to the abandonment of the Pentium 4 line. The Core 2 line reverts to a simpler pipeline with lower clock rates and multiple processors per chip. Copyright © 2009 Elsevier, Inc. All rights reserved.

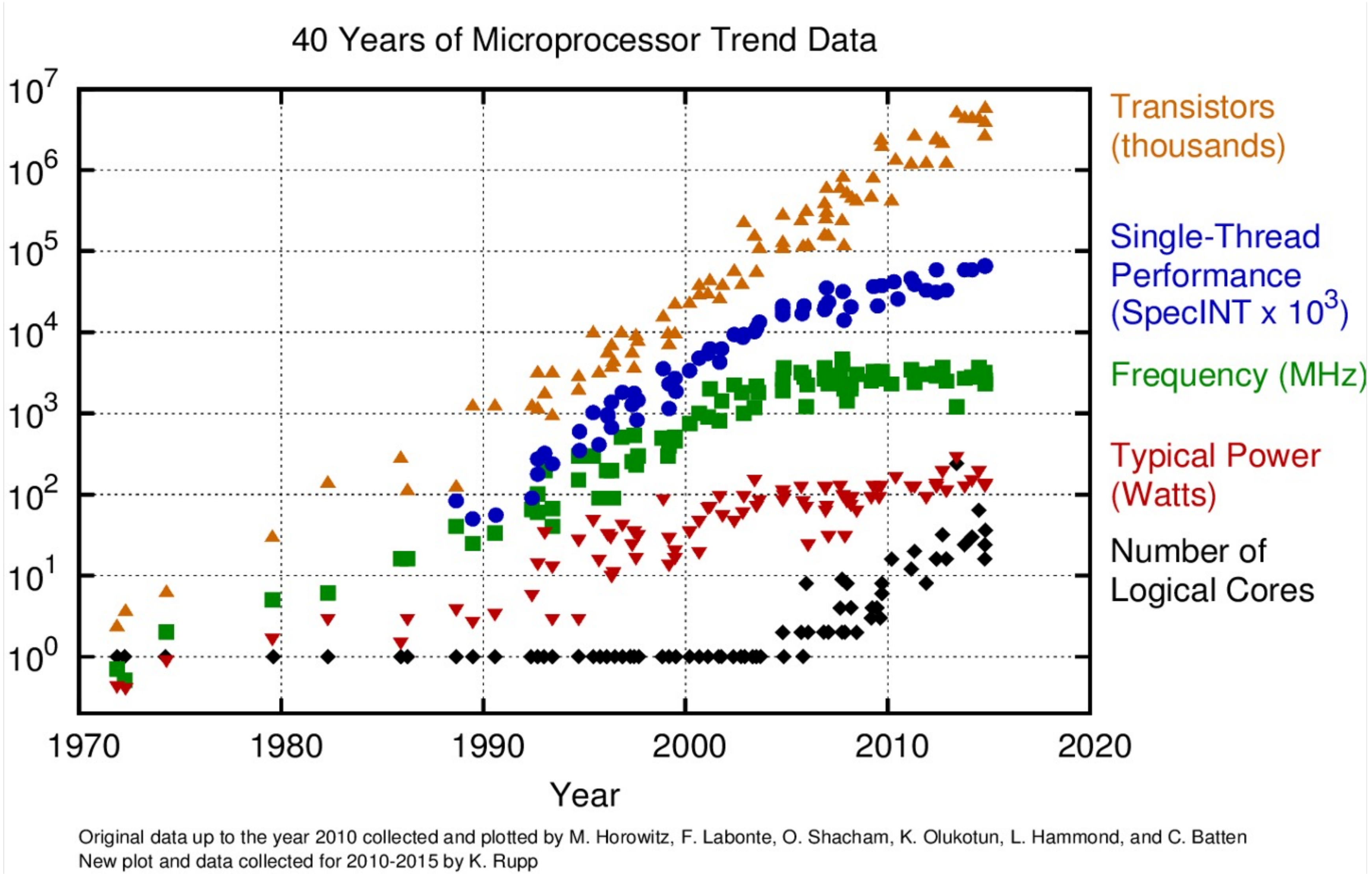
Patterson & Hennessy. Computer Organization and Design: The Hardware/Software Interface.

Drivers of CPU performance: scaling and architecture



Hennessy & Patterson. Computer Architecture: A Quantitative Approach

With end to Dennard's scaling, the continuation of Moore's law provided parallelism instead



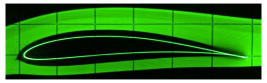
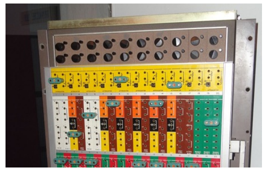
Computer Architecture

- Instruction level parallelism
- Limitations of instruction level parallelism

- Data level parallelism
- Limitations of data level parallelism

- Thread level parallelism
- Limitations of thread level parallelism

- ***Accelerators***
- Quantum

	1940s	1950s	1960s	1970s	1980s	1990s	2000s	2010s
Analog continuous-time computing	Analog computers for rocket and artillery controllers.	Analog computers for field problems. 	1st transistorized analog computer. 	Analog-digital hybrid computers.	…			
Digital discrete-time computing	Turing's *Bomba*.	1st transistorized digital computer.	Moore's law projection for transistor scaling.	Dennard's scaling for transistor power density.	VLSI democratized.	FPGAs introduced.	End of Dennard's scaling. CPUs go multicore.	Cloud FPGAs: Microsoft Catapult. Amazon F1.
	Stored program computer.	Microprogramming.	Instruction set architecture.	Reduced instruction set computers.		GPUs introduced.	Nvidia introduces CUDA.	ASICs: Google TPUs. DE Shaw Research Anton.

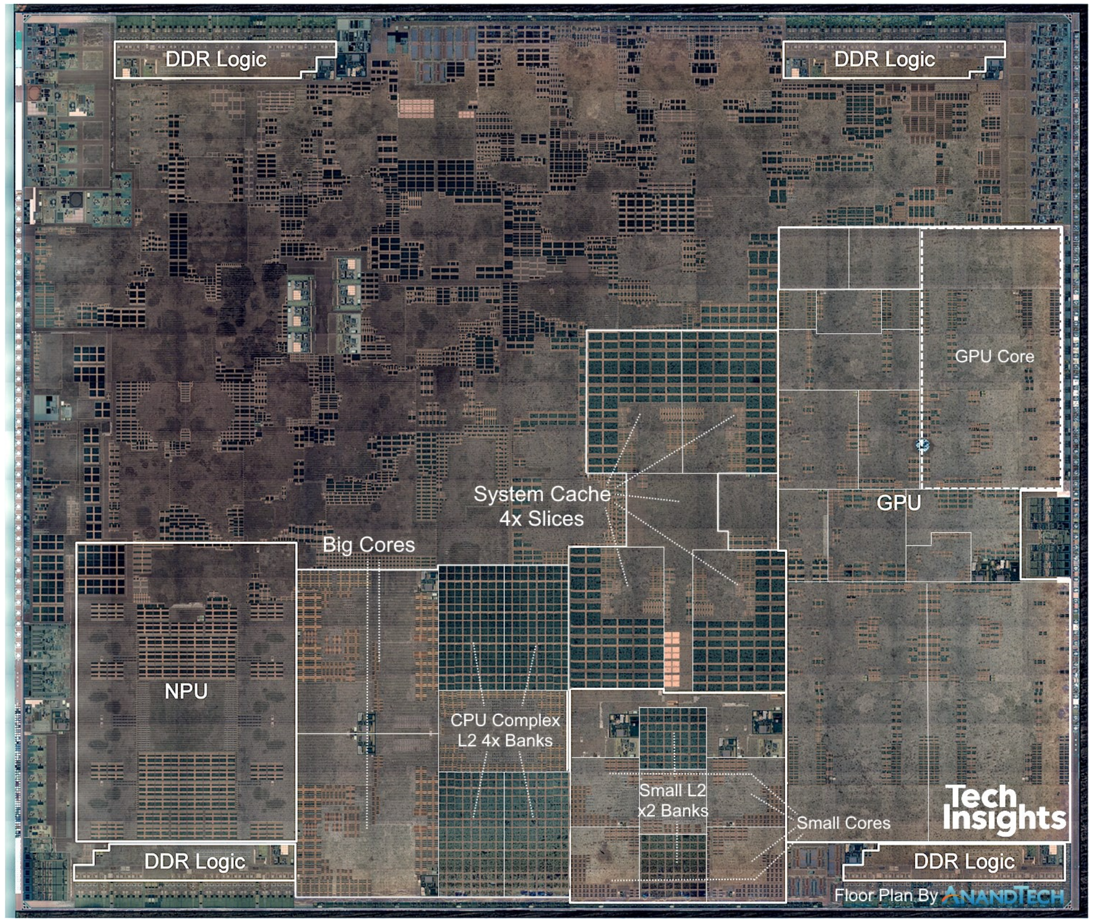
Heterogenous architectures

Transistor scaling and architectural abstractions drive digital revolution, make analog alternatives irrelevant

Scaling challenges drive heterogenous architectures

Image sources: AnalogMuseum.org, ComputerHistory.org

Accelerators



Computer Architecture

- Instruction level parallelism
- Limitations of instruction level parallelism

- Data level parallelism
- Limitations of data level parallelism

- Thread level parallelism
- Limitations of thread level parallelism

- Accelerators
- ***<u>Quantum</u>***

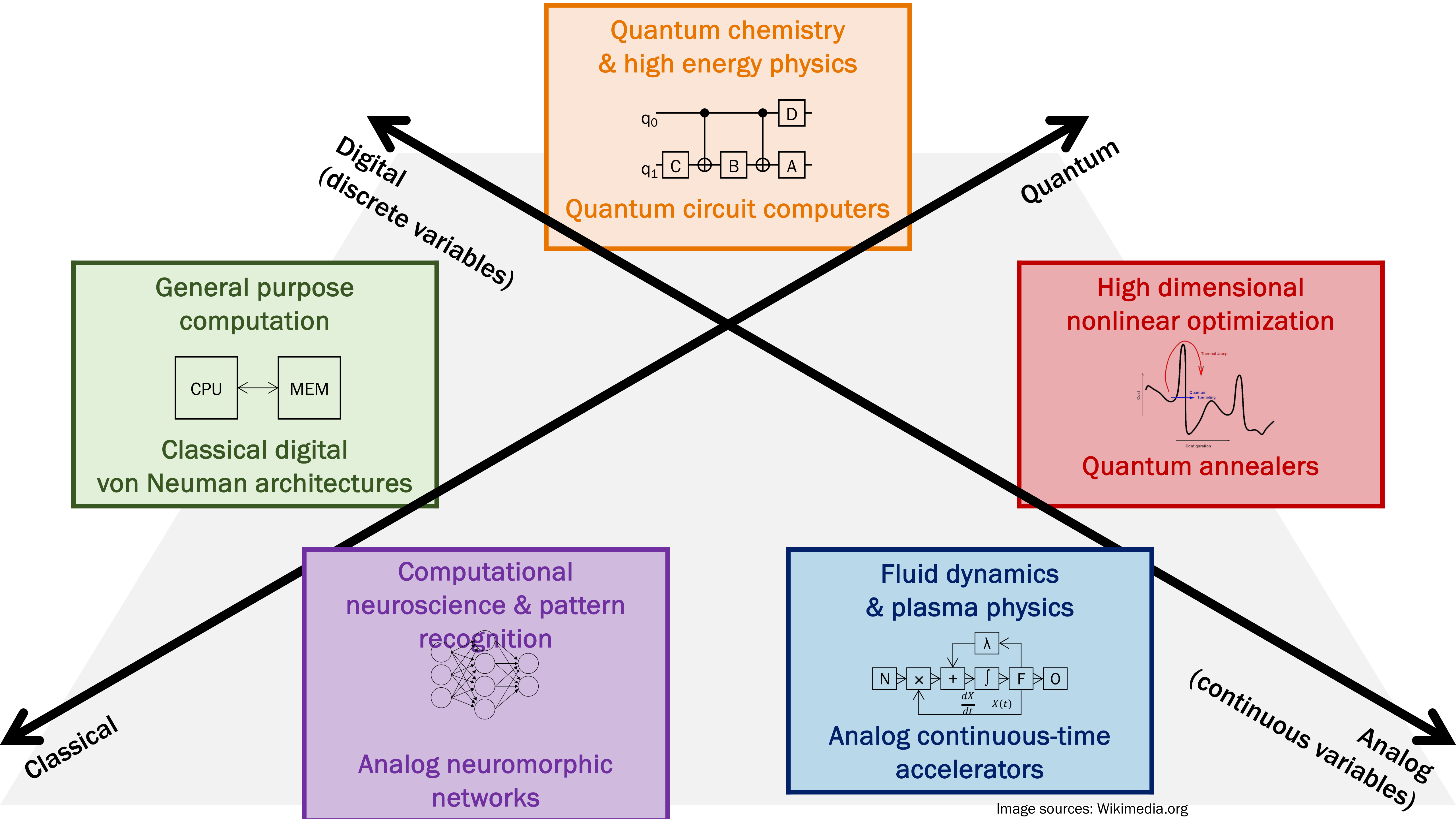


Image sources: Wikimedia.org

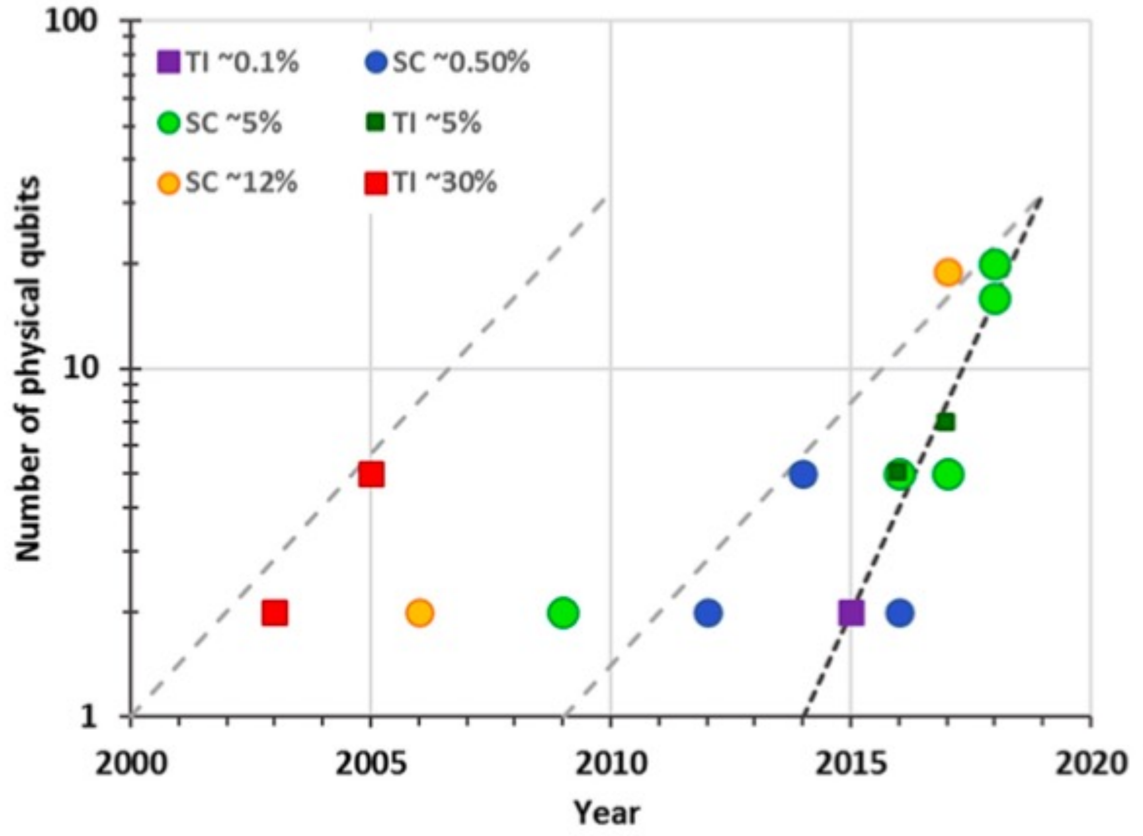


FIGURE 7.2 The number of qubits in superconductor (SC) and trapped ion (TI) quantum computers versus year; note the logarithmic scaling of the vertical axis. Data for trapped ions are shown as squares and for superconducting machines are shown as circles. Approximate average reported two-qubit gate error rates are indicated by color; points with the same color have similar error rates. The dashed gray lines show how the number of qubits would grow if they double every two years starting with one qubit in 2000 and 2009, respectively; the dashed black line indicates a doubling every year beginning with one qubit in 2014. Recent superconductor growth has been close to doubling every year. If this rate continued, 50 qubit machines with less than 5 percent error rates would be reported in 2019. SOURCE: Plotted data obtained from multiple sources [9].

Quantum Computing Progress and Prospects. National Academies Press.

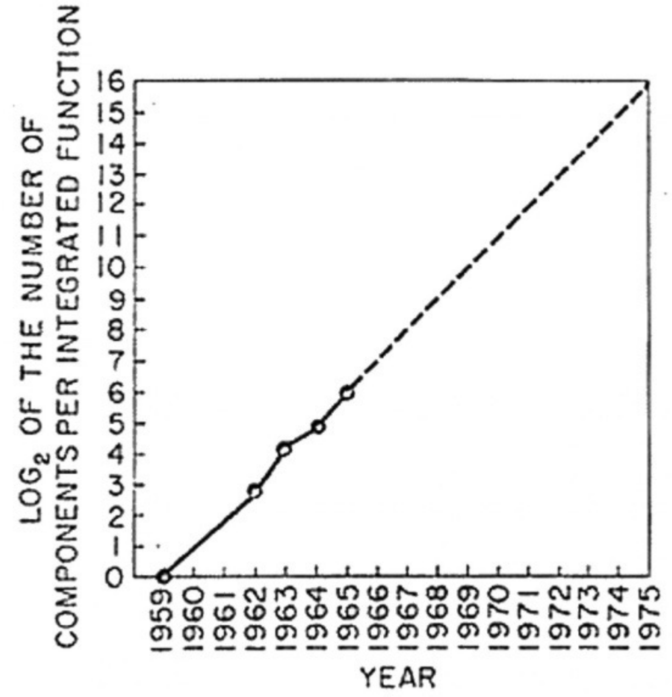


Fig. 2 Number of components per integrated function for minimum cost per component extrapolated vs time.

Intel

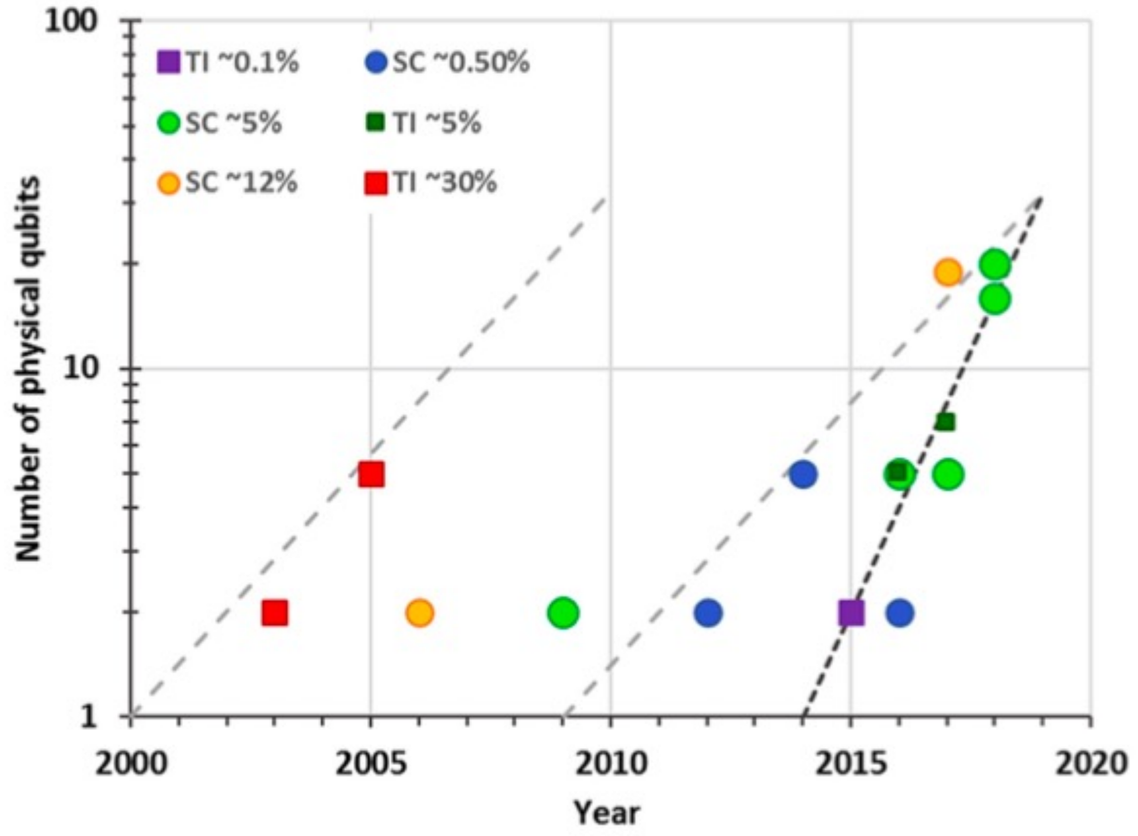


FIGURE 7.2 The number of qubits in superconductor (SC) and trapped ion (TI) quantum computers versus year; note the logarithmic scaling of the vertical axis. Data for trapped ions are shown as squares and for superconducting machines are shown as circles. Approximate average reported two-qubit gate error rates are indicated by color; points with the same color have similar error rates. The dashed gray lines show how the number of qubits would grow if they double every two years starting with one qubit in 2000 and 2009, respectively; the dashed black line indicates a doubling every year beginning with one qubit in 2014. Recent superconductor growth has been close to doubling every year. If this rate continued, 50 qubit machines with less than 5 percent error rates would be reported in 2019. SOURCE: Plotted data obtained from multiple sources [9].

Quantum Computing Progress and Prospects. National Academies Press.

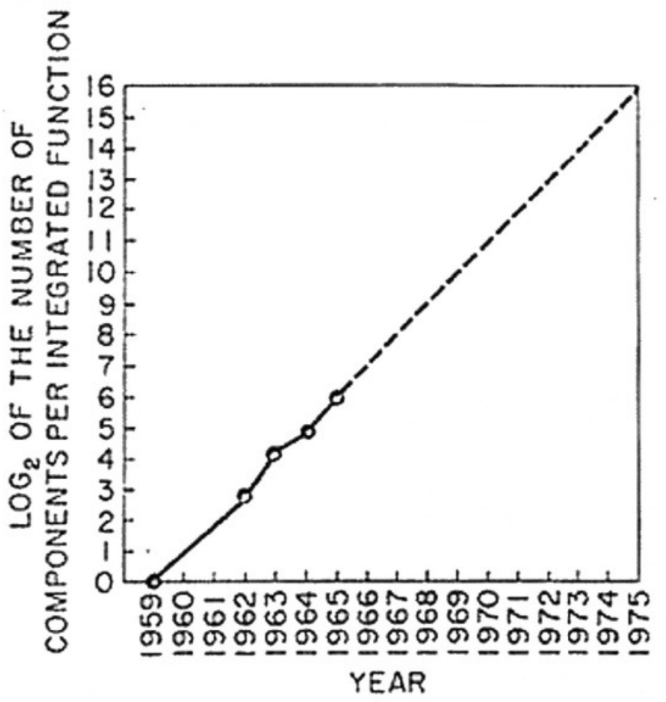
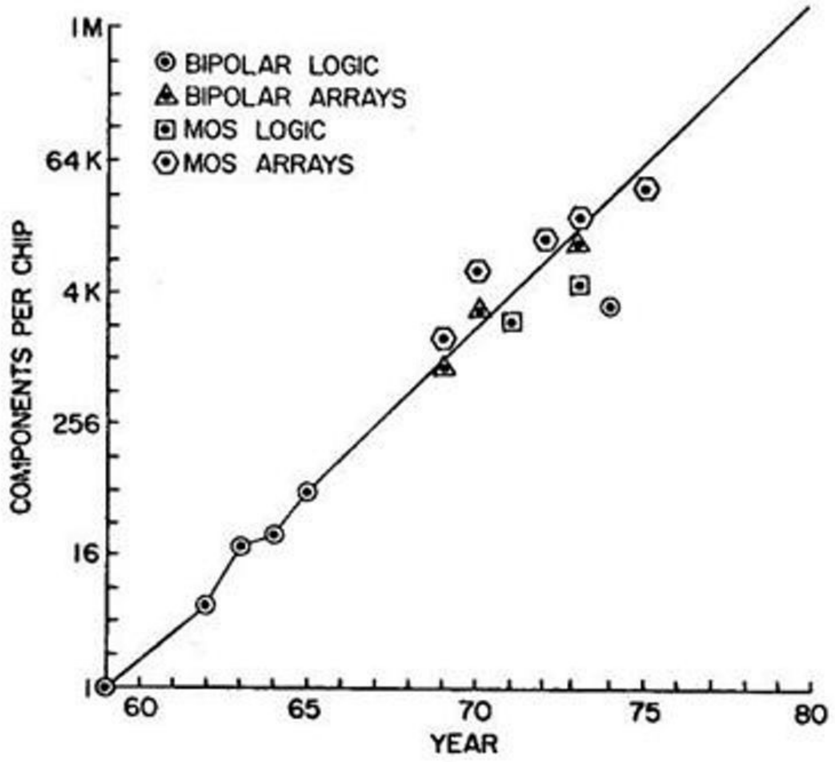


Fig. 2 Number of components per integrated function for minimum cost per component extrapolated vs time.

Intel



Computer History Museum

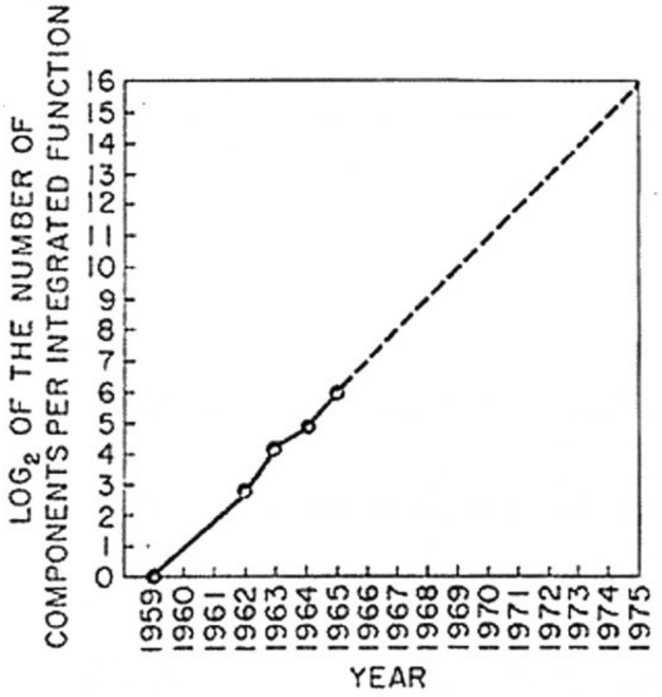


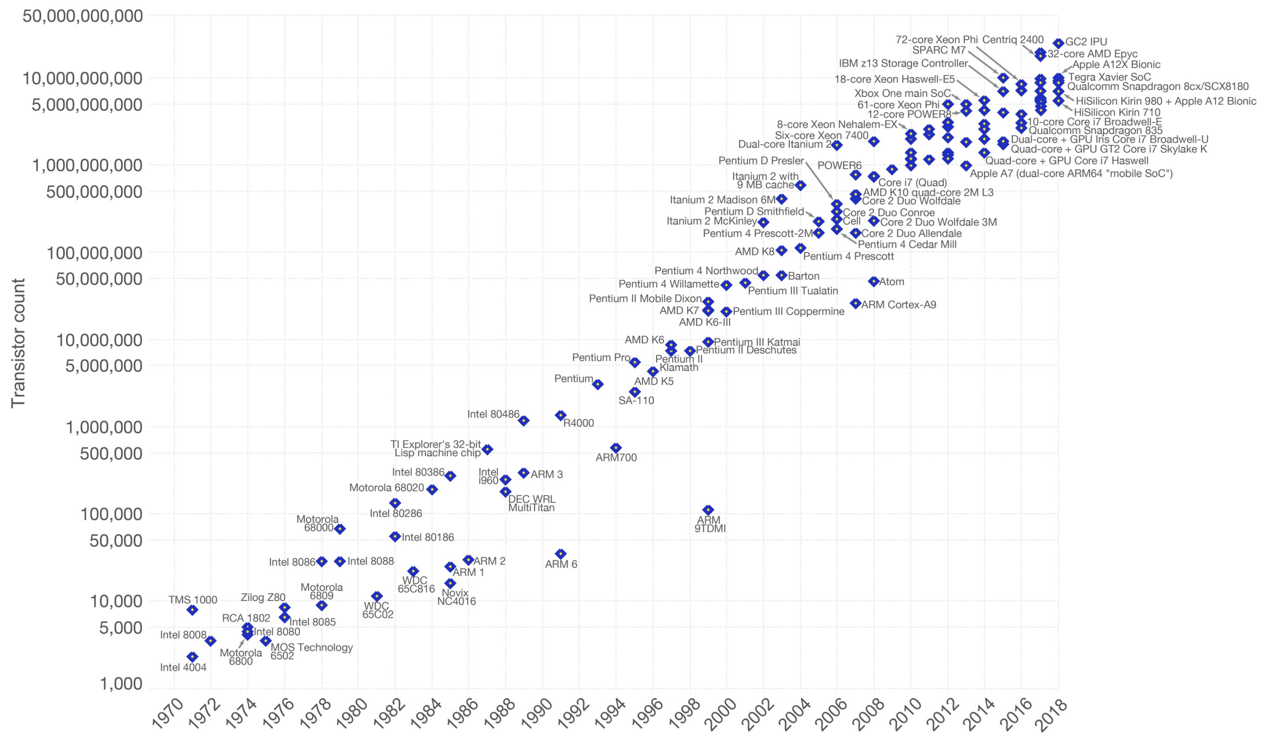
Fig. 2 Number of components per integrated function for minimum cost per component extrapolated vs time.

Intel

Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

Our World in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.



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.

Wikipedia

How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. Gidney and Ekerå. 2019.

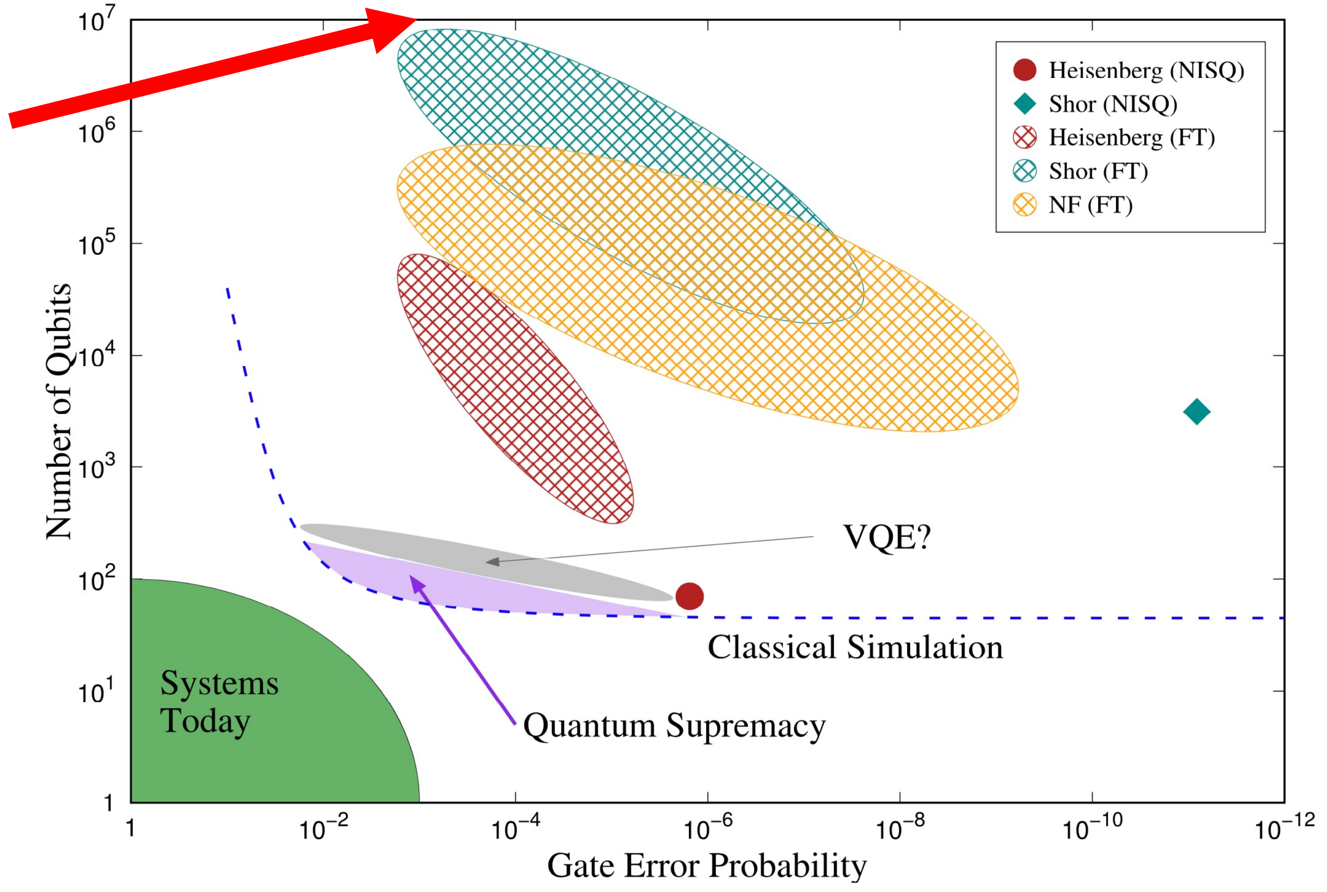


Fig. 2. ***Performance space of quantum computers, measured by the error probability of each entangling gate in the horizontal axis (roughly inversely proportional to the total number of gates that can be executed on a NISQ machine), and the number of qubits in the system in the vertical axis. Blue dotted line approximately demarcates quantum systems that can be simulated using best classical computers, while the green colored region shows where the existing quantum computing systems with verified performance numbers lie (as of September 2018). Purple shaded region indicates computational tasks that accomplish the so-called "quantum supremacy," where the computation carried out by the quantum computer defies classical simulation regardless of its usefulness. The different shapes illustrate resource counts for solving various problems, with solid symbols corresponding to the exact entangling gate counts and number of qubits in NISQ machines, and shaded regions showing approximate gate error requirements and number of qubits for an FT implementation (not pictured are the regions where the error gets too close to the known fault-tolerance thresholds): cyan diamond and shaded region correspond to factoring a 1024-bit number using Shor's algorithm [14], magenta circle and shaded region represent simulation of a 72-spin Heisenberg model [20], and orange shaded region illustrates NF simulation [21].***

An Outlook for Quantum Computing. Maslov et al.

A guide to the rest of the CS:APP textbook

CS:APP Chapter	Rutgers CS / ECE course in AY24-25 for further study
4. Processor Architecture	ECE 563 Computer Architecture I
5. Optimizing Program Performance	CS 214 Systems Programming / CS 415 Compilers
7. Linking	CS 415 Compilers
8. Exceptional Control Flow	CS 416 Operating Systems Design
9. Virtual Memory	CS 214 Systems Programming / CS 416 Operating Systems Design
10. System-Level I/O	CS 416 Operating Systems Design
11. Network Programming	CS 352 Internet Technology
12. Concurrent Programming	CS 214 Systems Programming / ECE 451 Parallel & Distributed Computing

https://classes.rutgers.edu//soc/#keyword?keyword=QUANTUM&semester=92024&campus=NB&level=U,G

2024 Fall: ECE 493. Soljanin. Quantum Computing Algorithms. (seniors only).
https://emina.flywheelsites.com/teaching/

2025 Spring: Physics 421. Roy. An Introduction to Quantum Computing.
https://www.physics.rutgers.edu/ugrad/421/

What my class is about

<u>Graduate seminar</u> on latest developments in <u>quantum computer engineering</u>

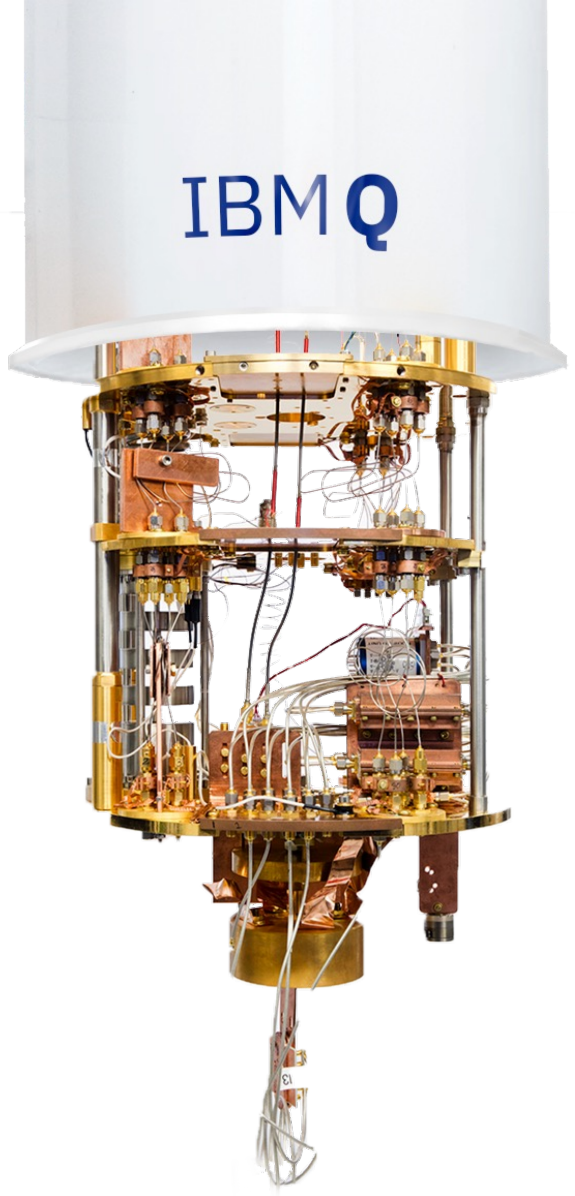
What is quantum computer engineering??

- realizing <u>quantum algorithms</u>
- on <u>prototype quantum computers</u>

—a rapidly growing field!!

Goals of the course:

- explore open-source tools for using quantum computers
- read and discuss recent developments
- build foundation for you to pursue research or to be experts in industry



Preview of the syllabus

- A systems view of quantum computer engineering
- Near-term intermediate-scale quantum algorithms
- Programming frameworks
- Emerging languages and representations
- Claims and counter claims for quantum advantage
- Extracting success
- Prototypes

Very important to help develop next iteration of this course.

- https://sirs.ctaar.rutgers.edu/blue