

Computer Architecture / Quantum Computing / Where to Go Next

Yipeng Huang

Rutgers University

December 9, 2025

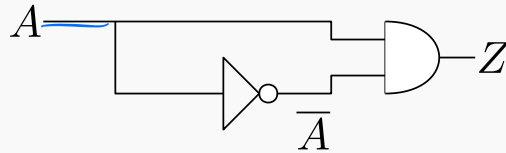
Computer Architecture

- **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

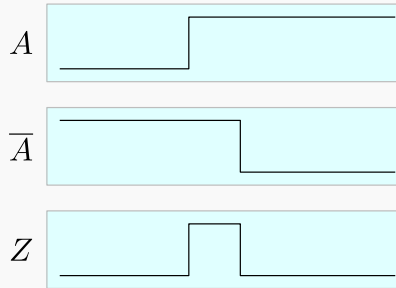
Synchronization. Clock signals.

Timing

Circuit:



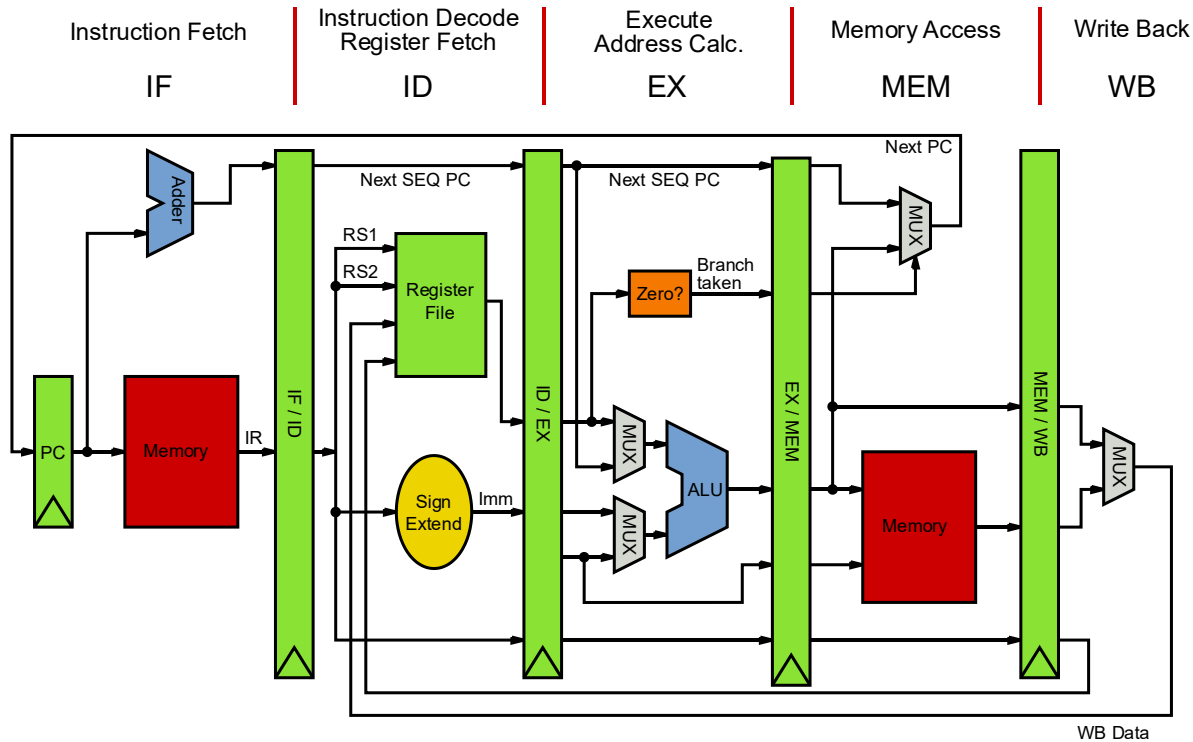
Voltages over time:



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

A	\bar{A}	$A \wedge \bar{A}$
0	1	0
1	0	0

The journey of an assembly instruction: pipeline



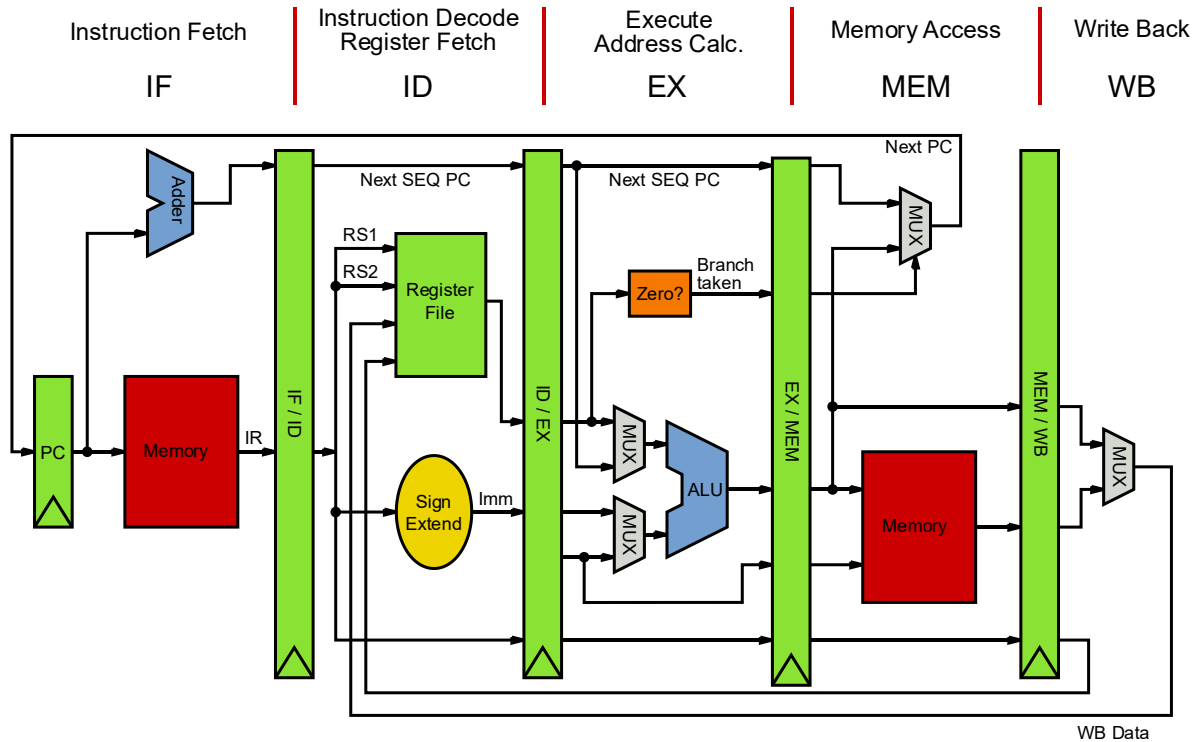
	IF	ID	EX	MEM	WB					
i		IF	ID	EX	MEM	WB				
t			IF	ID	EX	MEM	WB			
				IF	ID	EX	MEM	WB		
					IF	ID	EX	MEM	WB	

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					
i		IF	ID	EX	MEM	WB				
t			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

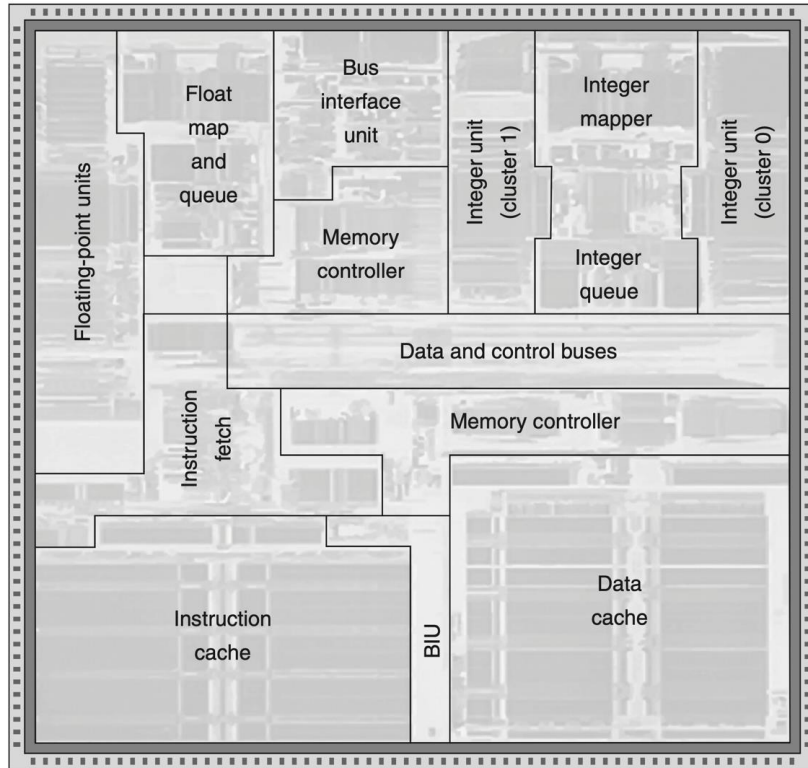
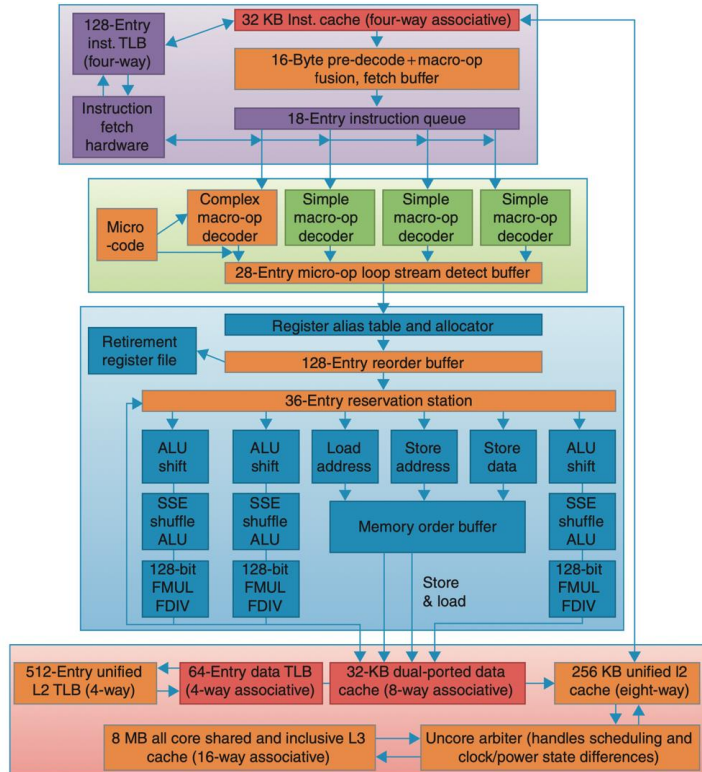


Figure 2.33 Floorplan of the Alpha 21264 [Kessler 1999].

Instruction Level Parallelism: modern trickery



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

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.

Computer Architecture

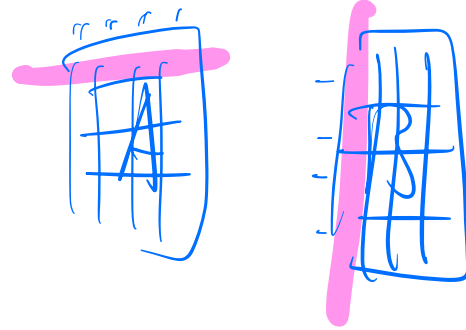
- Instruction level parallelism
- **Limitations of instruction level parallelism**
- Data level parallelism
- Limitations of data level parallelism
- Thread level parallelism
- Limitations of thread level parallelism
- Accelerators
- Quantum

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
- **Data level parallelism**
- **Limitations of data level parallelism**
- Thread level parallelism
- Limitations of thread level parallelism
- Accelerators
- Quantum



Data Level Parallelism

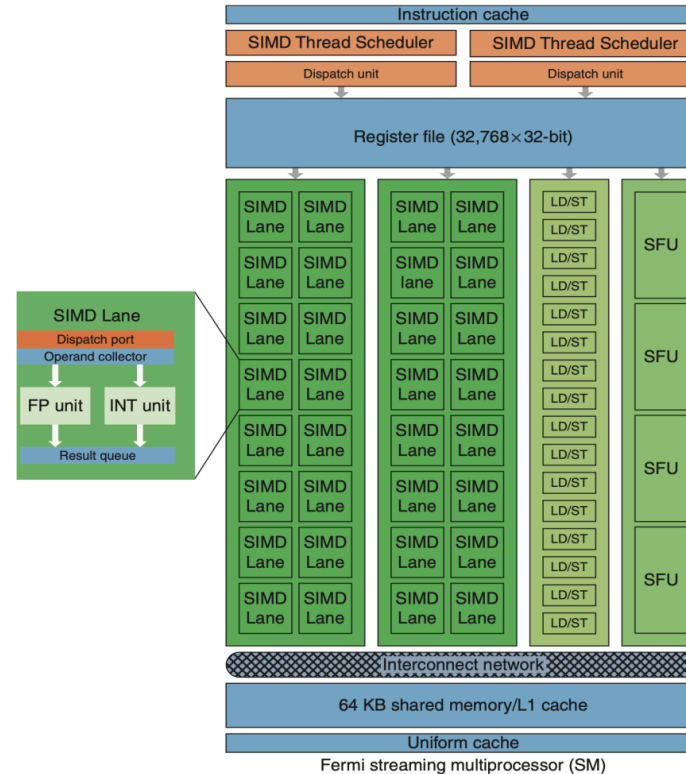
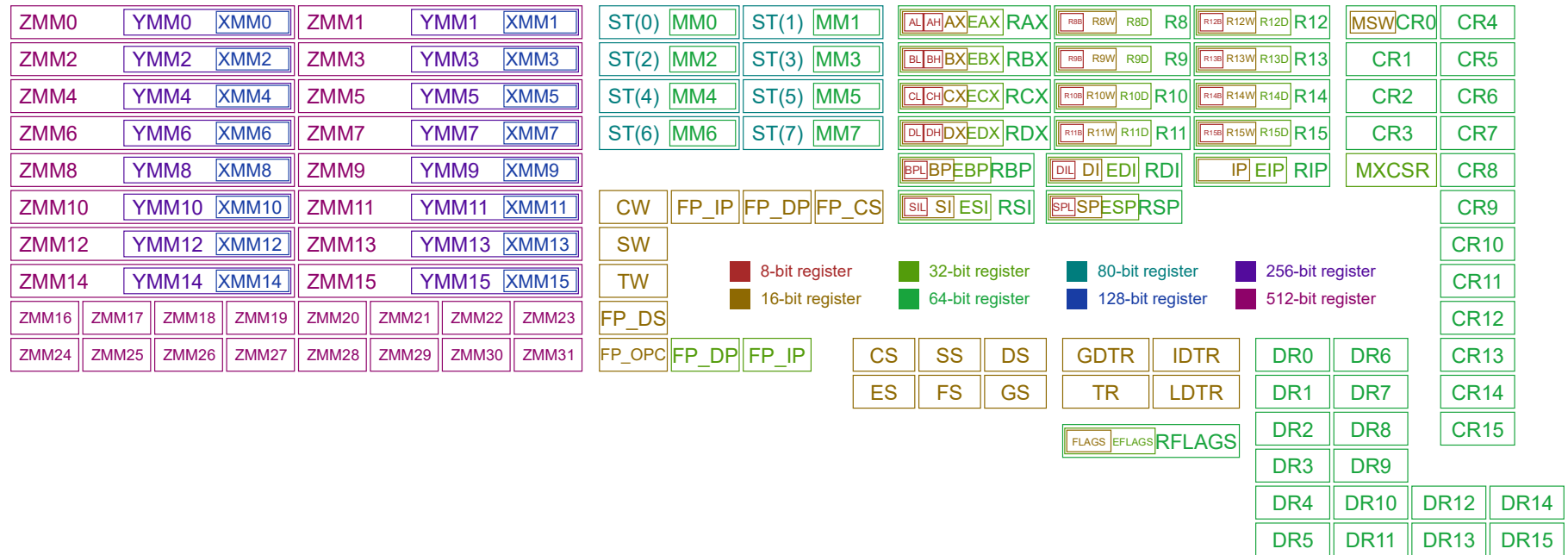


Figure 4.20 Block diagram of the multithreaded SIMD Processor of a Fermi GPU. Each SIMD Lane has a pipelined floating-point unit, a pipelined integer unit, some logic for dispatching instructions and operands to these units, and a queue for holding results. The four Special Function units (SFUs) calculate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

Data Level Parallelism



Computer Architecture

- Instruction level parallelism
- Limitations of instruction level parallelism

- Data level parallelism
- Limitations of data level parallelism

not all workloads linear algebra

- **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.

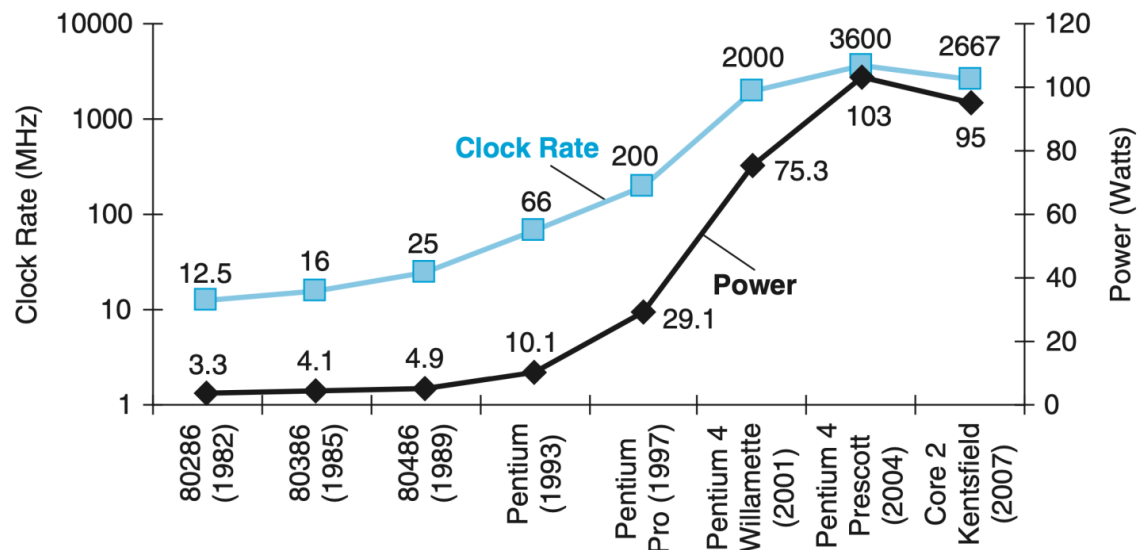
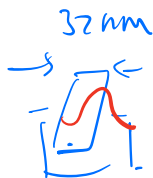
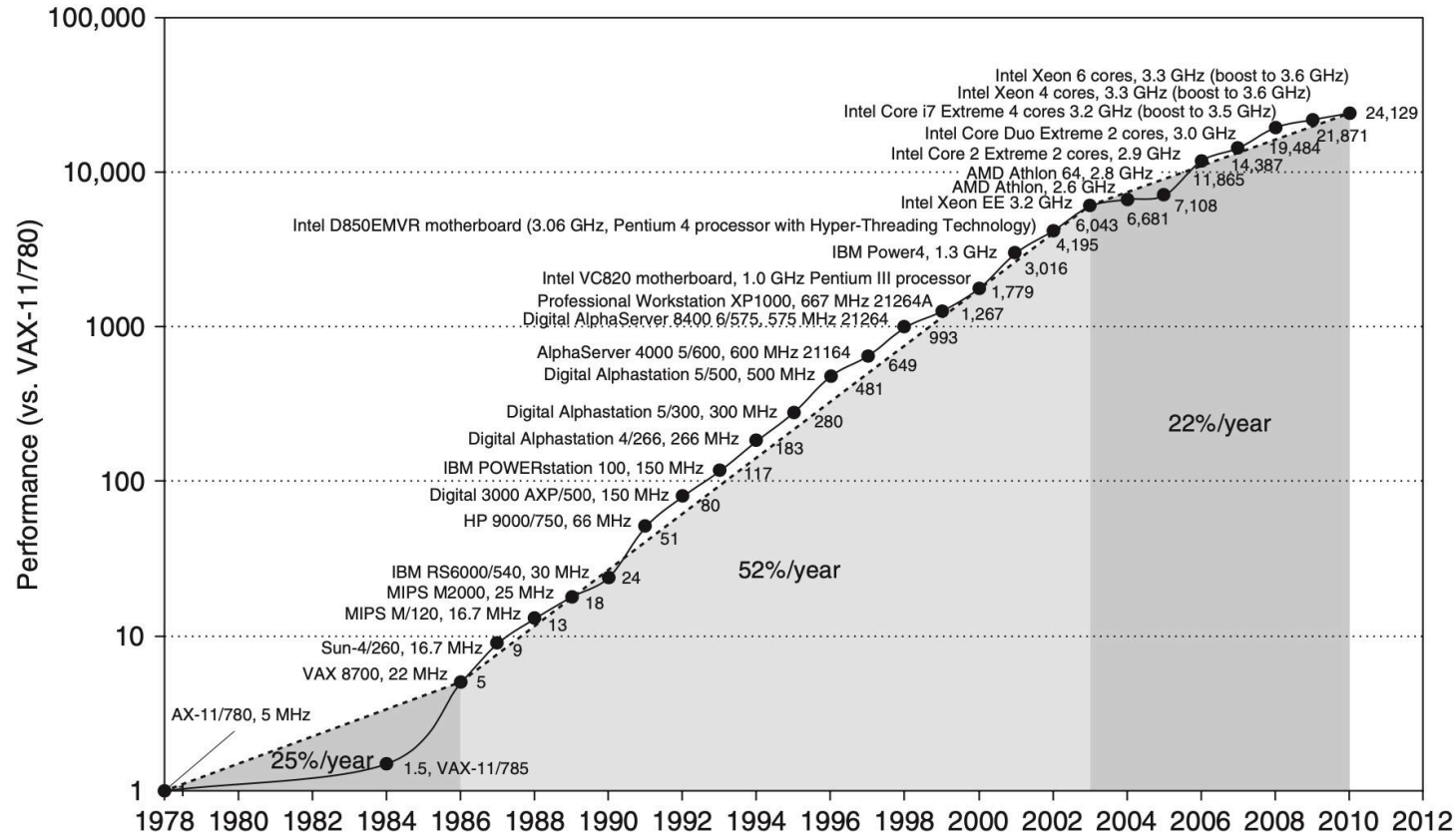
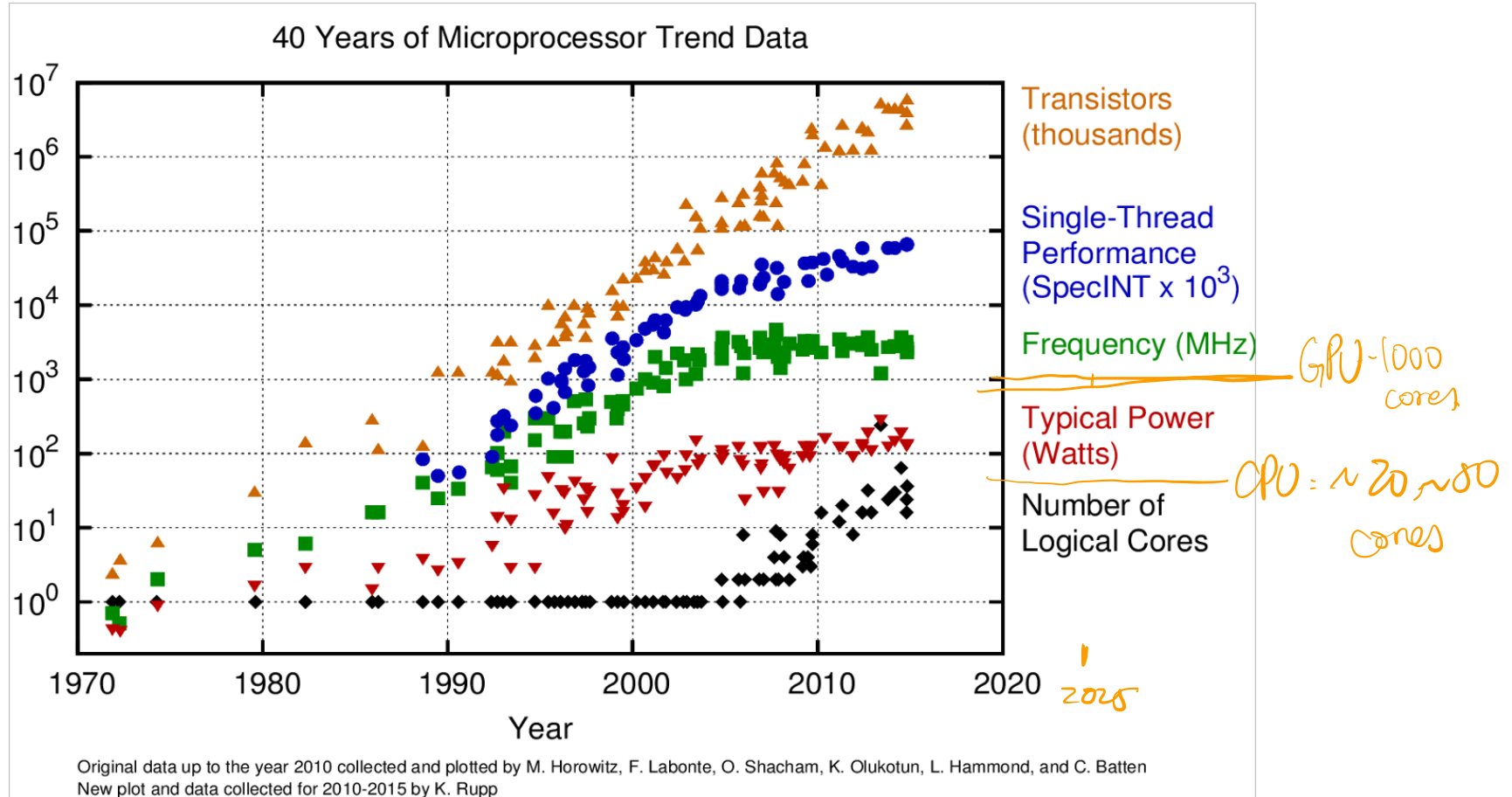


FIGURE 1.15 Clock rate and Power for Intel x86 microprocessors over eight generations and 25 years. The Pentium 4 made a dramatic jump in clock rate and power but less so in performance. The Prescott thermal problems led to the abandonment of the Pentium 4 line. The Core 2 line reverts to a simpler pipeline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

Drivers of CPU performance: scaling and architecture

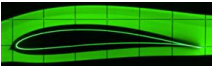



With end to Dennard's scaling, the continuation of Moore's law provided parallelism instead

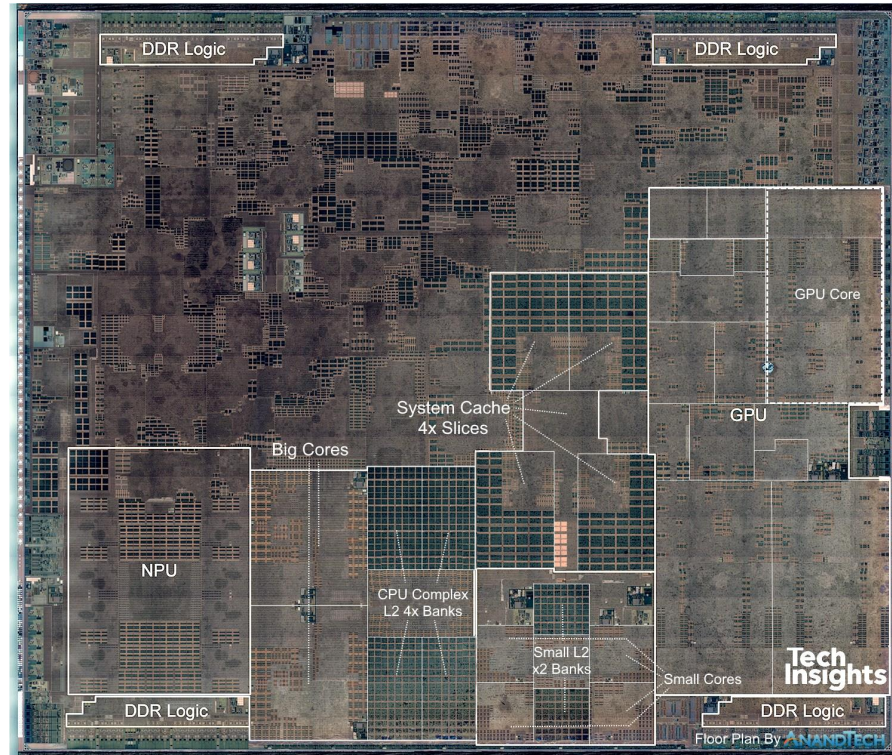


Computer Architecture

- 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

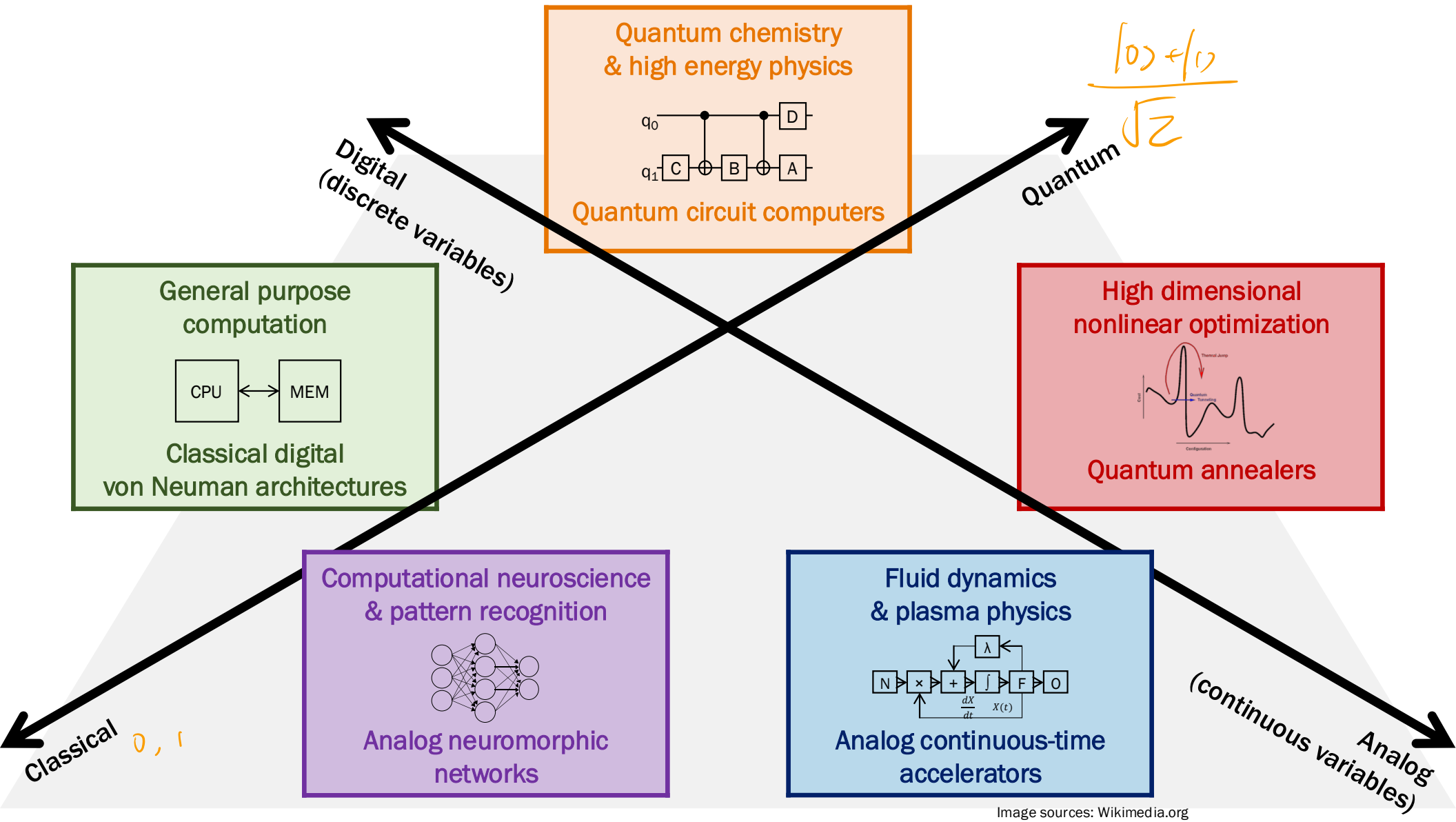
	1940s	1950s	1960s	1970s	1980s	1990s	2000s	2010s
Analog continuous-time computing	Analog computers for rocket and artillery controllers.	Analog computers for field problems. 	1 st transistorized analog computer. 	Analog-digital hybrid computers.	...			
Digital discrete-time computing	Turing's <i>Bomba</i> . Stored program computer.	1 st transistorized digital computer. Microprogram ming.	Moore's law projection for transistor scaling. Instruction set architecture.	Dennard's scaling for transistor power density. Reduced instruction set computers.	VLSI democratized.	<div> <div>FPGAs introduced.</div> <div>GPUs introduced.</div> <div>Heterogenous architectures</div> </div>		
						End of Dennard's scaling. CPUs go multicore.	Cloud FPGAs: Microsoft Catapult. Amazon F1.	ASICs: Google TPUs. DE Shaw Research Anton.
<div> Transistor scaling and architectural abstractions drive digital revolution, make analog alternatives irrelevant </div>						<div> Scaling challenges drive heterogenous architectures </div>		

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
- Quantum



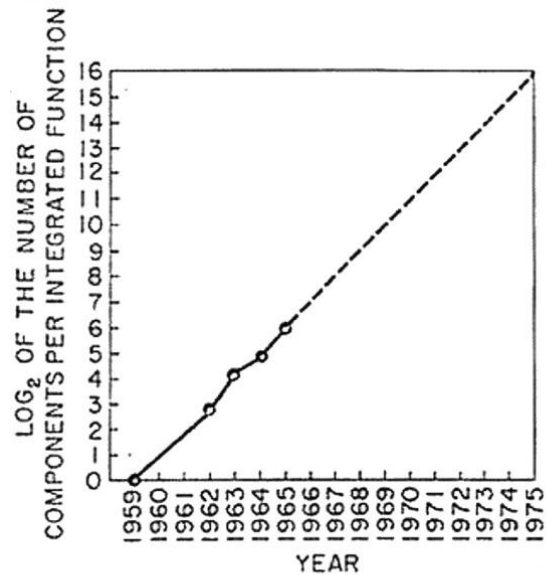
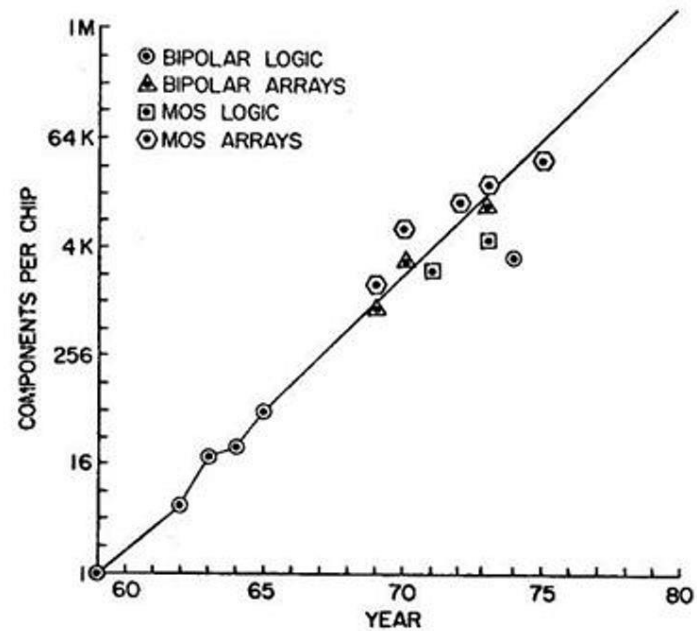


Fig. 2 Number of components per integrated function for minimum cost per component extrapolated vs time.

Intel



Computer History Museum

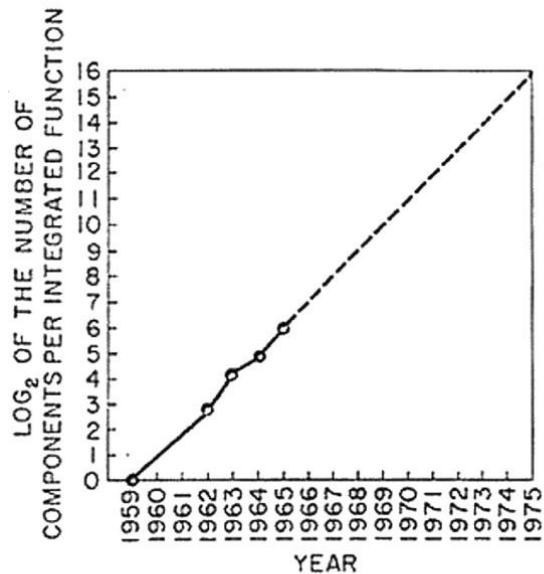
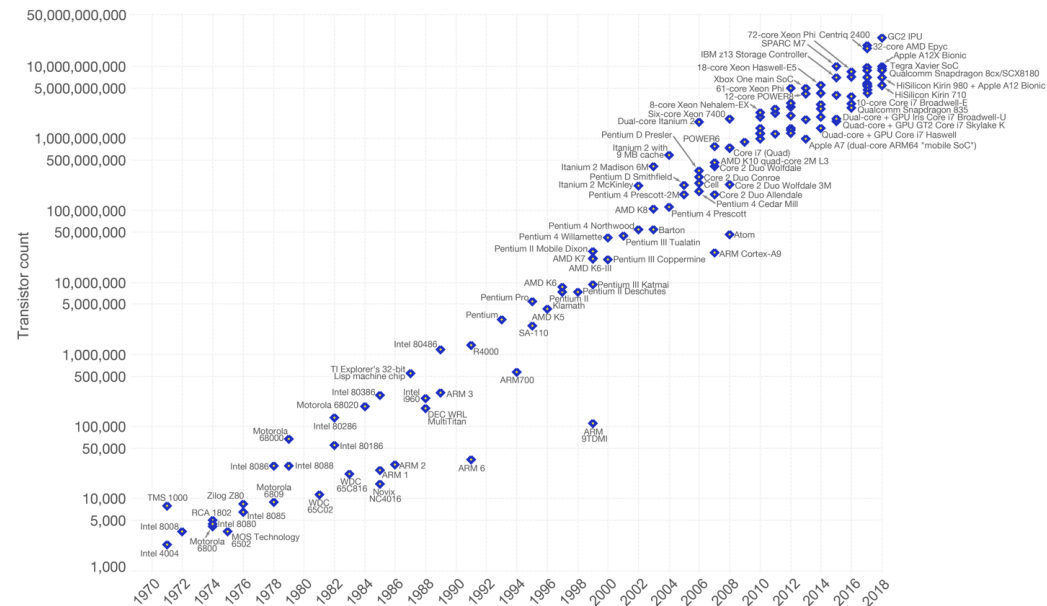


Fig. 2 Number of components per integrated function for minimum cost per component extrapolated vs time.

Intel

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

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.

Licensed under CC-BY-SA by the author Max Roser.

Wikipedia

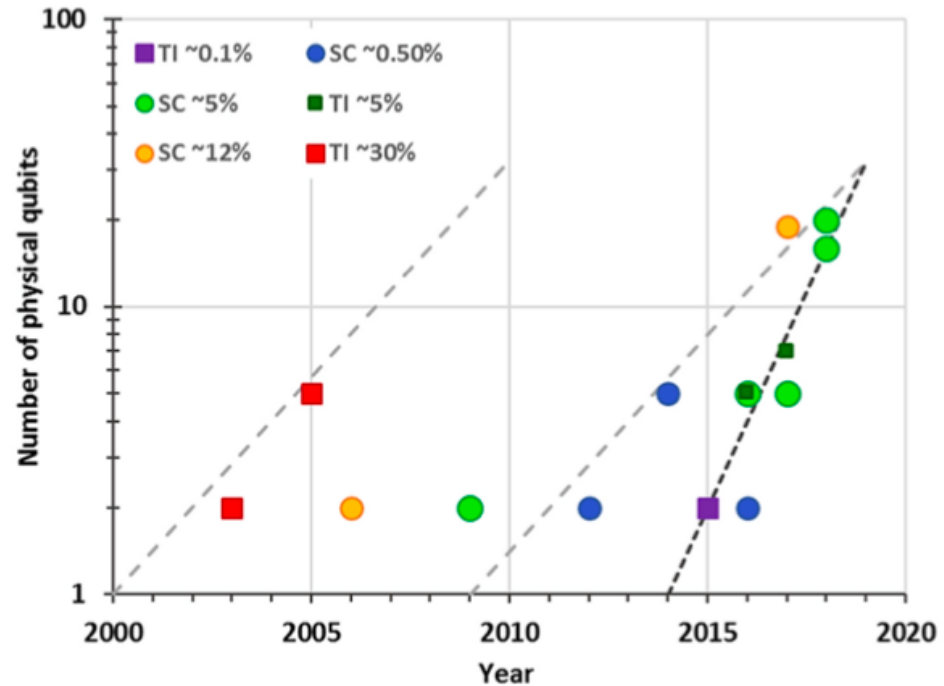


FIGURE 7.2 The number of qubits in superconductor (SC) and trapped ion (TI) quantum computers versus year; note the logarithmic scaling of the vertical axis. Data for trapped ions are shown as squares and for superconducting machines are shown as circles. Approximate average reported two-qubit gate error 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].

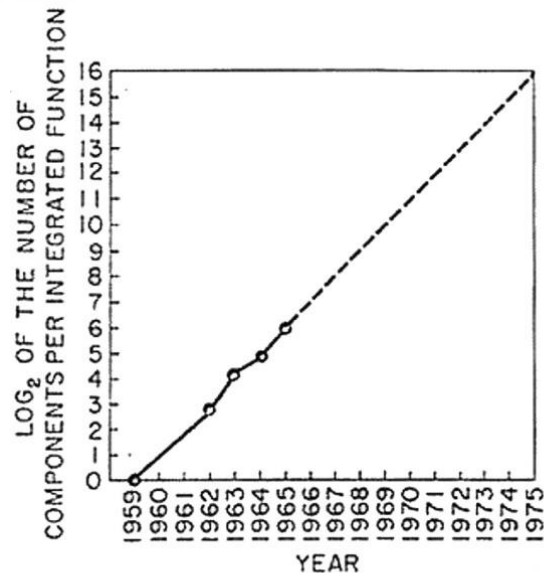


Fig. 2 Number of components per integrated function for minimum cost per component extrapolated vs. time.

Intel

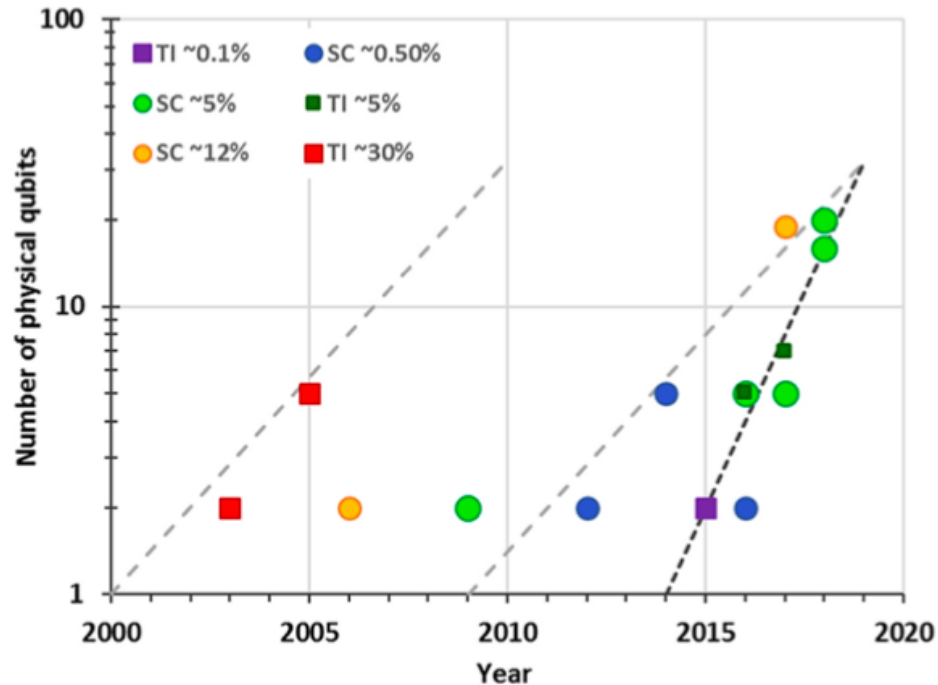


FIGURE 7.2 The number of qubits in superconductor (SC) and trapped ion (TI) quantum computers versus year; note the logarithmic scaling of the vertical axis. Data for trapped ions are shown as squares and for superconducting machines are shown as circles. Approximate average reported two-qubit gate error 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].

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

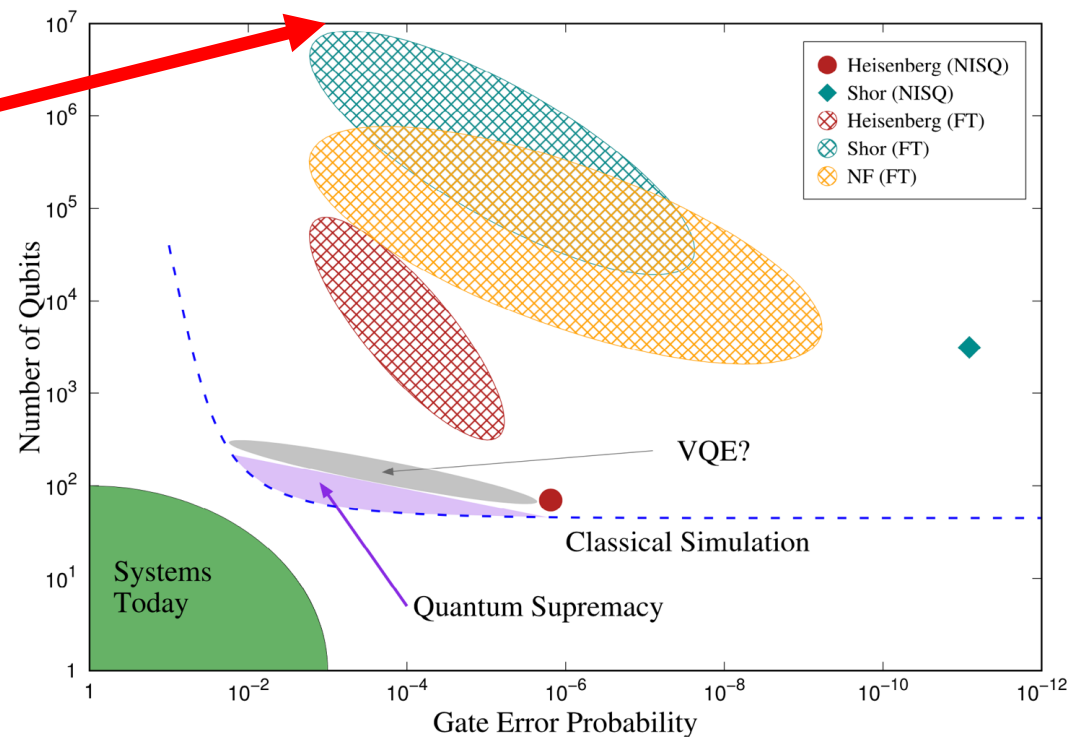


Fig. 2. Performance space of quantum computers, measured by the error probability of each entangling gate in the horizontal axis (roughly inversely proportional to the total number of gates that can be executed on a NISQ machine), and the number of qubits in the system in the vertical axis. Blue dotted 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].

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=12026&campus=NB,NK,CM&level=U,G>

2026 Spring: ECE 493/558. Soljanin. Quantum Computing and Information Systems. (seniors only).

<https://emina.flywheelsites.com/teaching/>

2026 Spring: Physics 421. Roy. An Introduction to Quantum Computing.

<https://www.physics.rutgers.edu/ugrad/421/>

What my class is about

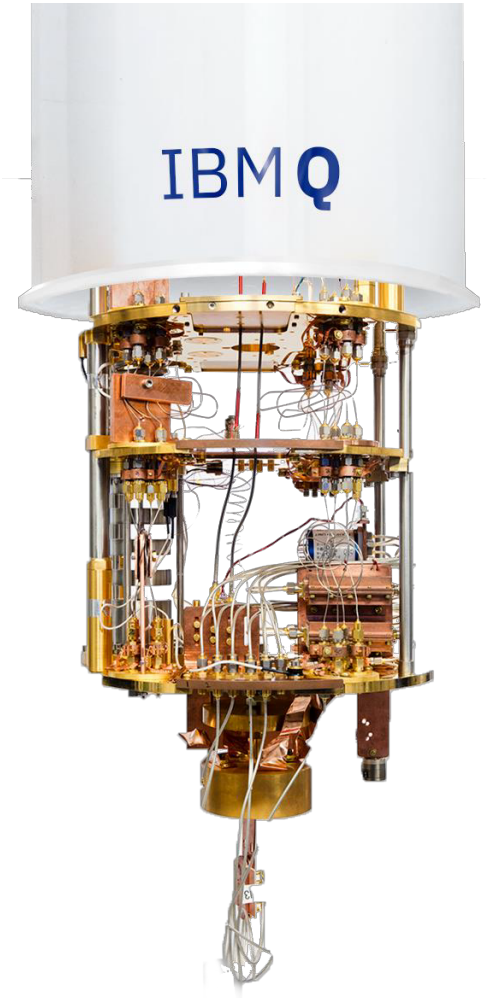
Graduate seminar on latest developments in quantum computer engineering

What is quantum computer engineering??

- realizing quantum algorithms
 - on prototype quantum computers
- 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>