A New Golden Age for Computer Architecture

David Patterson
UC Berkeley and Google
February 2023

Outline

- Computer Architecture 1960-2010 (10 minutes)
- Computer Architecture Today (10 minutes)
- Domain Specific Architectures for Machine Learning (5 minutes)
- Reducing CO2 emissions of Machine Learning (5 minutes)
- Conclusion
- Q&A

Computer Architecture 1960-2010

Instruction Set Architecture?

- Software talks to hardware using a **vocabulary**
  - Words called ***instructions***
  - Vocabulary called ***instruction set architecture (ISA)***

- Most important interface since determines software that can run on hardware
  - Software is distributed as instructions

IBM Compatibility Problem in Early 1960s

By early 1960's, *IBM had 4 incompatible lines of computers!*

701 → 7094
650 → 7074
702 → 7080
1401 → 7010

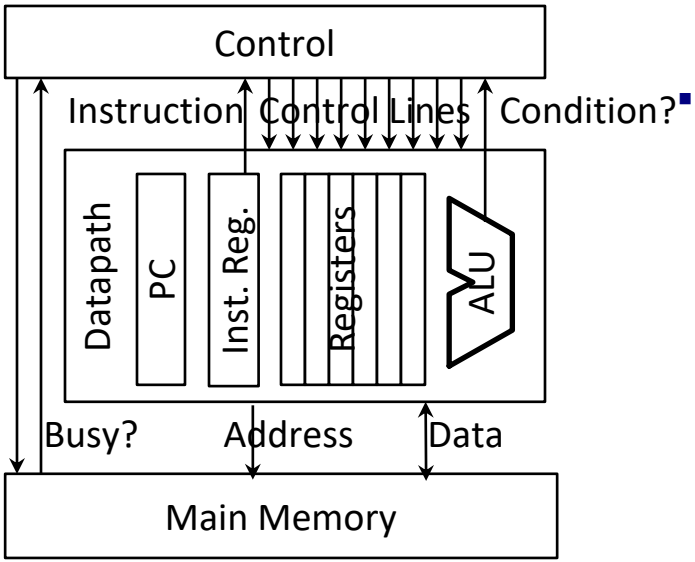
Each system had its own:

- Instruction set architecture (ISA)
- I/O system and Secondary Storage:
  magnetic tapes, drums and disks
- Assemblers, compilers, libraries,...
- Market niche: business, scientific, real time, ...

IBM System/360 – one ISA to rule them all

Control versus Datapath

- Processor designs split between *datapath*, where numbers are stored and arithmetic operations computed, and *control*, which sequences operations on datapath
- Biggest challenge for computer designers was getting control correct



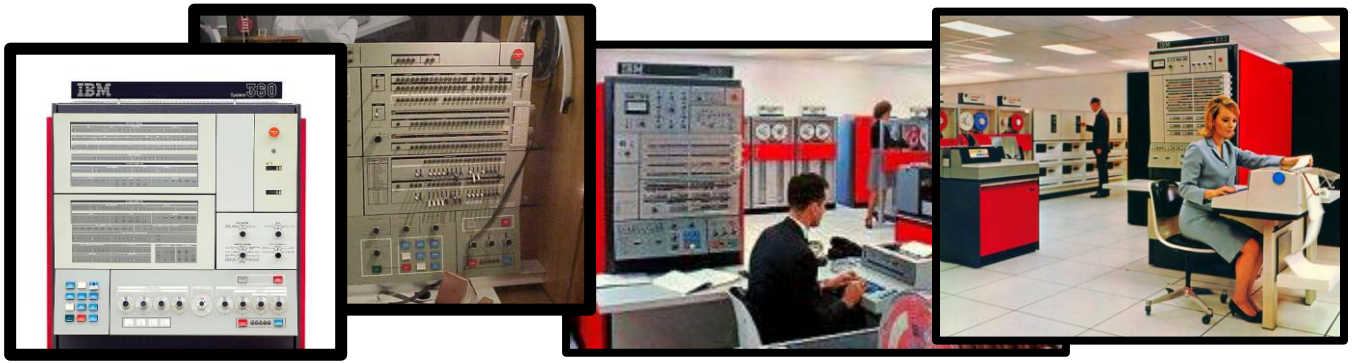
- **Maurice Wilkes** invented the idea of *microprogramming* to design the control unit of a processor*

  - Logic expensive vs. ROM or RAM
  - ROM cheaper and faster than RAM
  - *Control design now programming*

* "Micro-programming and the design of the control circuits in an electronic digital computer," M. Wilkes, and J. Stringer. *Mathematical Proc. of the Cambridge Philosophical Society,* Vol. 49, 1953.

Microprogramming in IBM 360

Model	M30	M40	M50	M65
Datapath width	8 bits	16 bits	32 bits	64 bits
Microcode size	4k x 50	4k x 52	2.75k x 85	2.75k x 87
Clock cycle time (ROM)	750 ns	625 ns	500 ns	200 ns
Main memory cycle time	1500 ns	2500 ns	2000 ns	750 ns
Price (1964 $)	$192,000	$216,000	$460,000	$1,080,000
Price (2023 $)	$1,860,000	$2,090,000	$4,450,000	$10,460,000



Fred Brooks, Jr.

IC Technology, Microcode, and CISC

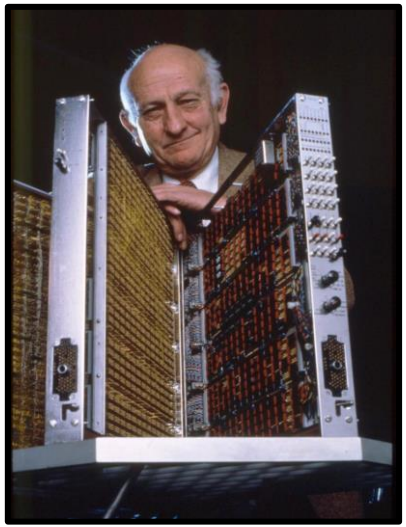
- Logic, RAM, ROM all implemented using same transistors
- Semiconductor RAM ~ same speed as ROM
- With Moore's Law, memory for control store could grow
- Since RAM, easier to fix microcode bugs
- Allowed more complicated ISAs (CISC)
- Minicomputer (TTL server) example:
  - Digital Equipment Corp. (DEC)
  - VAX ISA in 1977
- 5K x 96b microcode

Microprocessor Evolution

- Rapid progress in 1970s, fueled by advances in MOS technology, imitated minicomputers and mainframe ISAs
- “Microprocessor Wars”: compete by adding instructions (easy for microcode), justified given assembly language programming
- Intel iAPX 432: Most ambitious 1970s micro, started in 1975
  - 32-bit capability-based, object-oriented architecture, custom OS written in Ada
  - Severe performance, complexity (multiple chips), and usability problems; announced 1981
- Intel 8086 (1978, 8MHz, 29,000 transistors)
  - “Stopgap” 16-bit processor, 52 weeks to new chip
  - ISA architected in 3 weeks (10 person weeks) assembly-compatible with 8 bit 8080
- IBM PC 1981 picks Intel 8088 for 8-bit bus (and Motorola 68000 was late)
- Estimated PC sales: 250,000
- Actual PC sales: 100,000,000 ⇒ 8086 “overnight” success
- Binary compatibility of PC software ⇒ bright future for 8086

Analyzing Microcoded Machines 1980s

- UNIX proves even operating systems can be written in HLL
- Compilers now source of measurements
- John Cocke group at IBM
  - Worked on a simple pipelined processor, 801 minicomputer (ECL server), and advanced compilers inside IBM
  - Ported their compiler to IBM 370, only used simple register-register and load/store instructions (similar to 801)
  - Up to 3x faster than existing compilers that used full 370 ISA!
- Emer and Clark at DEC in early 1980s*
  - Found VAX 11/780 average clock cycles per instruction (CPI) = 10!
  - Found 20% of VAX ISA ⇒ 60% of microcode, but only 0.2% of execution time!



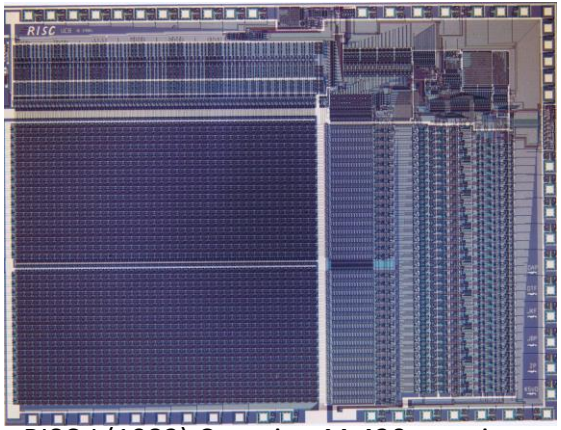
John Cocke

* "A Characterization of Processor Performance in the VAx-11/780," J. Emer and D.Clark, *ISCA*, 1984.

From CISC to RISC

- Use RAM for instruction *cache* of user-visible instructions
  - Contents of fast instruction memory change to what application needs now vs. ISA interpreter
- Use simple ISA
  - Instructions as simple as microinstructions, but not as wide
  - Enable pipelined implementations
  - Compiled code only used a few CISC instructions anyways

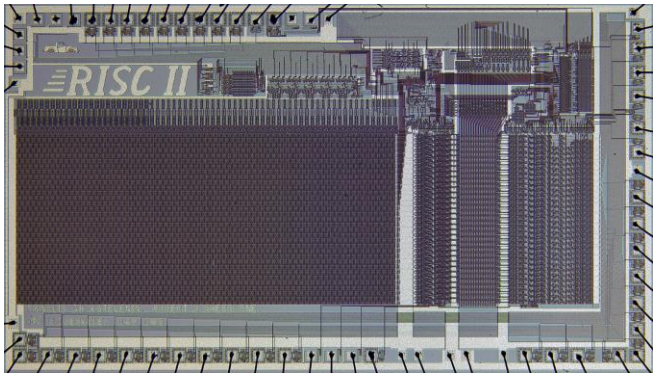
Berkeley and Stanford RISC Chips



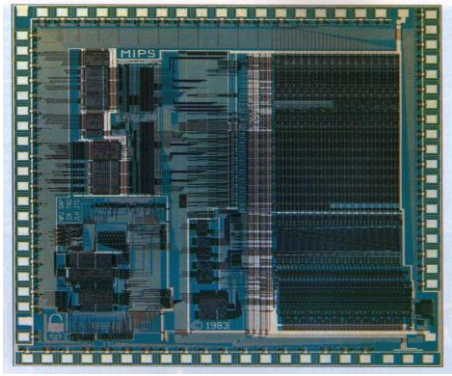
RISC-I (1982) Contains 44,420 transistors, fabbed in 5 µm NMOS, with a die area of 77 mm^2, ran at 1 MHz

Fitzpatrick, Daniel, John Foderaro, Manolis Katevenis, Howard Landman, David Patterson, James Peek, Zvi Peshkess, Carlo Séquin, Robert Sherburne, and Korbin Van Dyke. "A RISCy approach to VLSI." *ACM SIGARCH Computer Architecture News* 10, no. 1 (1982)

Hennessy, John, Norman Jouppi, Steven Przybylski, Christopher Rowen, Thomas Gross, Forest Baskett, and John Gill. "MIPS: A microprocessor architecture." In *ACM SIGMICRO Newsletter*, vol. 13, no. 4, (1982).



RISC-II (1983) contains 40,760 transistors, was fabbed in 3 µm NMOS, ran at 3 MHz, and the size is 60 mm^2



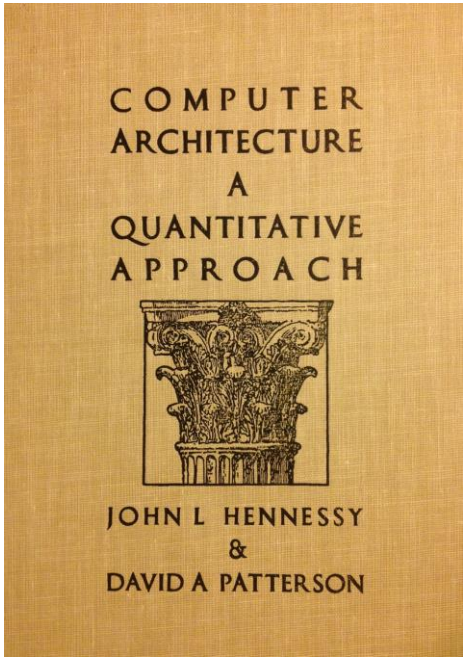
Stanford MIPS (1983) contains 25,000 transistors, was fabbed in 3 µm & 4 µm NMOS, ran at 4 MHz (3 µm), and size is 50 mm^2 (4 µm) (Microprocessor without Interlocked Pipeline Stages)

"Iron Law" of Processor Performance: How RISC can win

$$\frac{\text{Time}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} * \frac{\text{Clock cycles}}{\text{Instruction}} * \frac{\text{Time}}{\text{Clock cycle}}$$

- CISC executes fewer instructions / program (~ ¾ instructions)
  but many more clock cycles per instruction (~ 6x CPI)
  ⇒ RISC ~ 4x faster than CISC

"Performance from architecture: comparing a RISC and a CISC with similar hardware organization," Dileep Bhandarkar and Douglas Clark, *Proc. Symposium, ASPLOS*, 1991.



Computer Architecture Today

CISC vs. RISC Today

PC Era

- Hardware translates x86 instructions into internal RISC instructions
- Then use any RISC technique inside MPU
- > 350M / year !
- x86 ISA eventually dominates servers as well as desktops

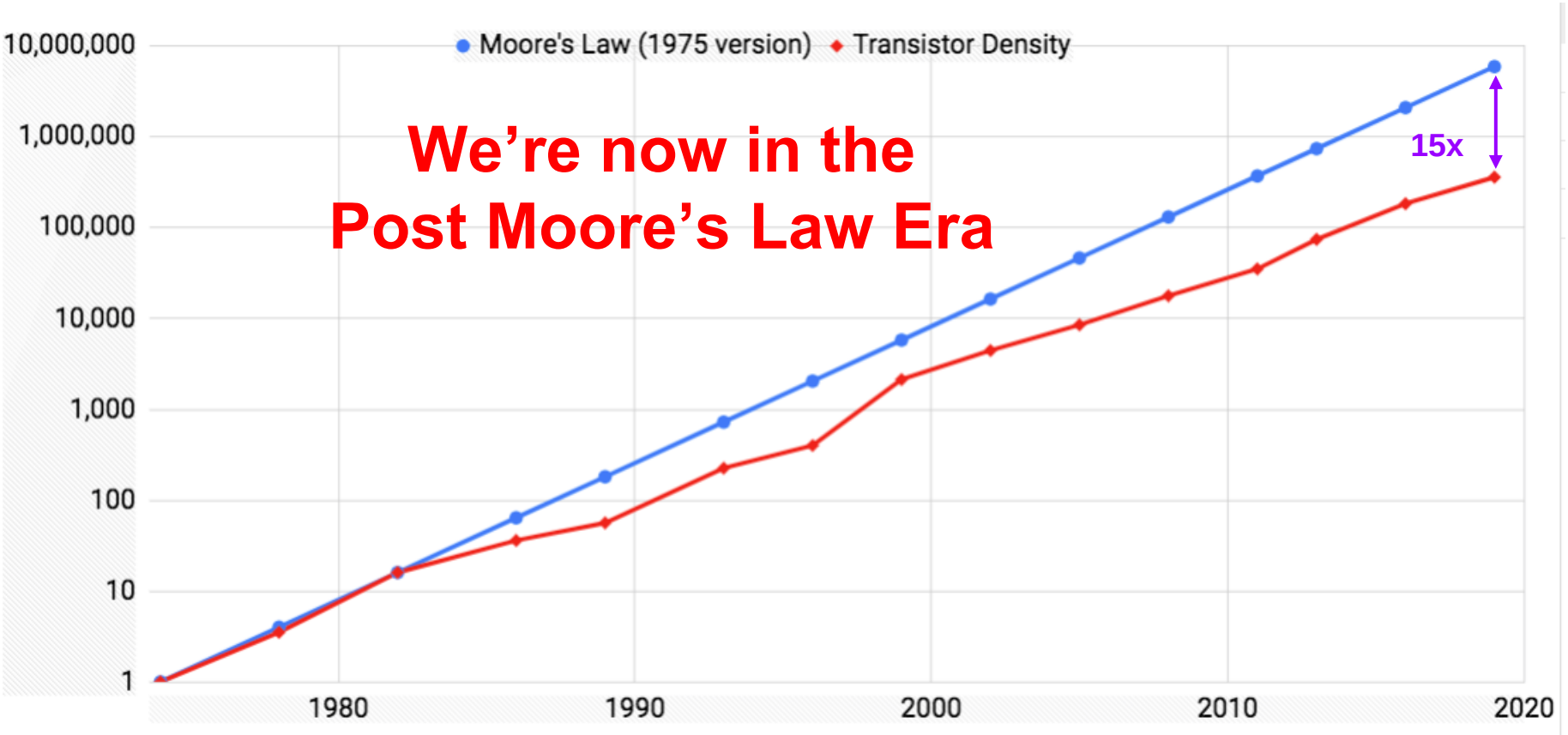
PostPC Era: Client/Cloud

- IP in SoC chip vs. MPU chip
- Value die area, energy as much as performance
- >25B total / year in 2022
- 99% Processors today are RISC
- RISC-V a free and open ISA (billions of cores per year in 2022)

Lessons from RISC vs CISC

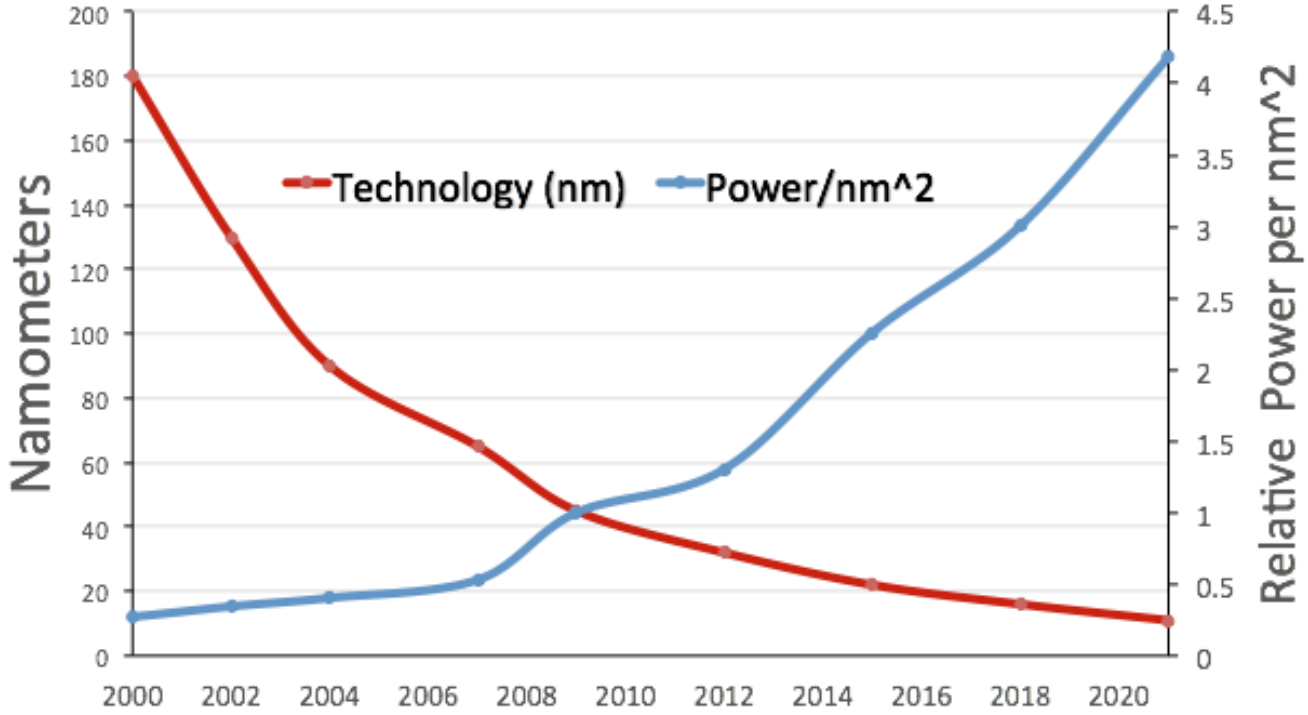
- Less is More
  - It’s harder to come up with simple solutions, but they accelerate progress
- Importance of the software stack vs the hardware
  - If compiler can’t generate it, who cares?
- Importance of good benchmarks
  - Hard to make progress if you can’t measure it
  - For better or for worse, benchmarks shape a field
- Take the time for a quantitative approach vs rely on intuition to start quickly

Moore’s Law Slowdown in Intel Processors



Moore, Gordon E. "No exponential is forever: but ‘Forever’ can be delayed!"
Solid-State Circuits Conference, 2003.

Technology & Power: Dennard Scaling

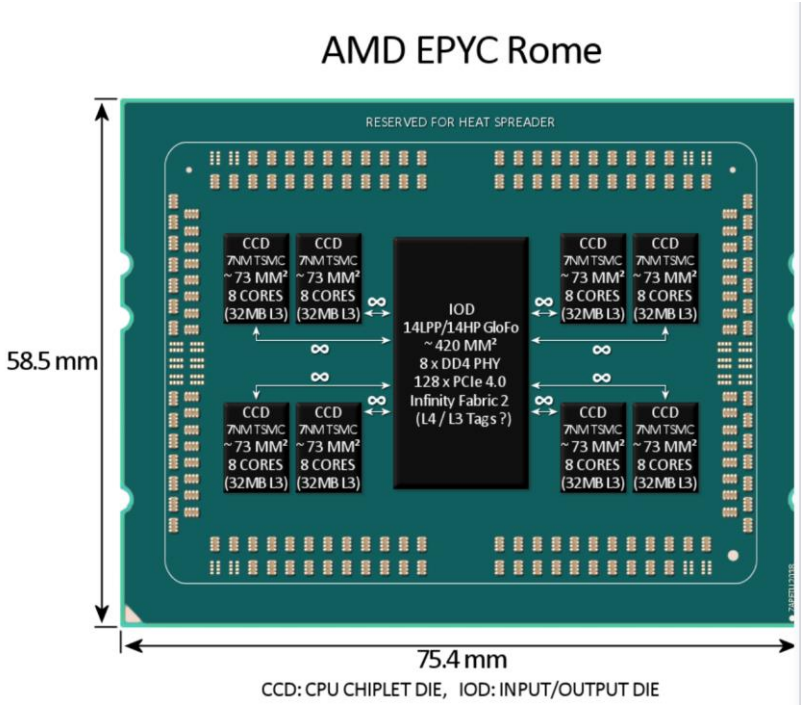


Power consumption based on models in "Dark Silicon and the End of Multicore Scaling," Hadi Esmaelizadeh, *ISCA,* 2011

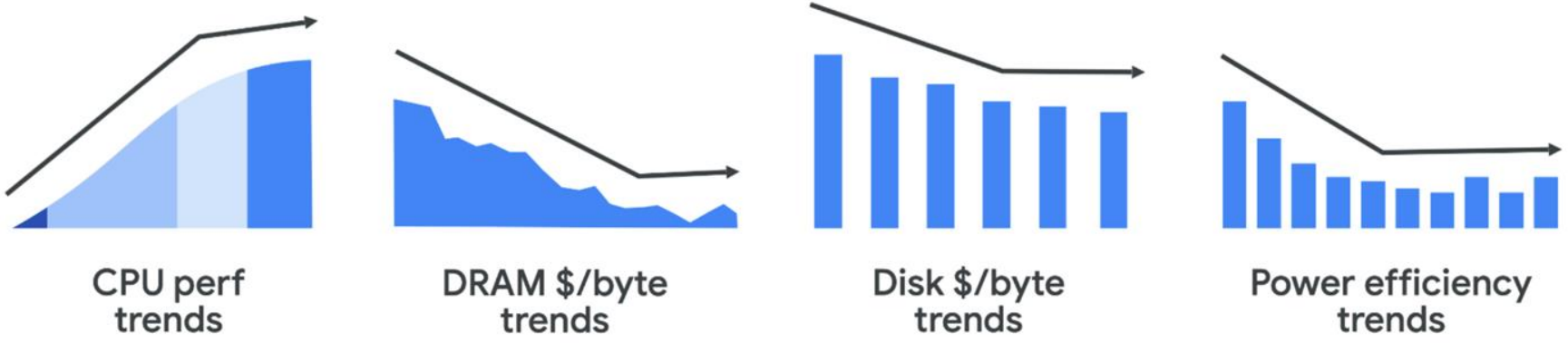
Energy scaling for fixed task is better, since more and faster transistors

Bespoke Chiplet Solution

- As Moore's Law diminishes, semiconductor wafer costs rising faster than performance gains from latest technology
- Instead of increasingly larger chips in latest technology, use clever packaging and smaller chips ("chiplets"), some in older technologies
  - AMD EPYC Rome: 1 I/O chiplet in 12 nm + ≤ 8 core complex chiplets in 7 nm
  - Intel Sapphire Rapids: 4 chiplets, each with a subset of cores and IOs
- Save money (smaller chips have higher yield, some use old tech) *and* allows bigger systems (more transistors)
- More Cores, but cores no faster
  - May Improve throughput, not latency
  - Have to scale memory bandwidth too



Technology plateaus



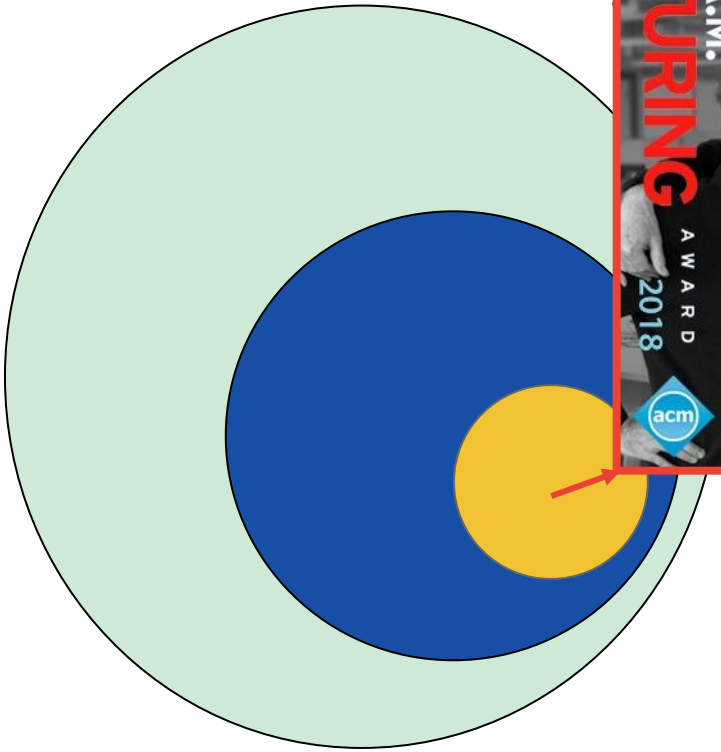
Domain Specific Architectures for Machine Learning

Exciting New Frontier for Computer Architects

- Doubling transistor count but no longer at fixed cost or fixed power
- From 2x every two years to 2x every 10 years
- Slowing Moore's Law & Dennard Scaling ⇒ *Domain Specific Architectures* (*DSA*)
  - Do few things very well, but do everything poorly
- What to accelerate?

Cloud to AI

artificial intelligence



…om up

Techniques to learn from labeled data
Inspired by neurons in brain
From explainable machine learning
to hard-to-explain machine learning

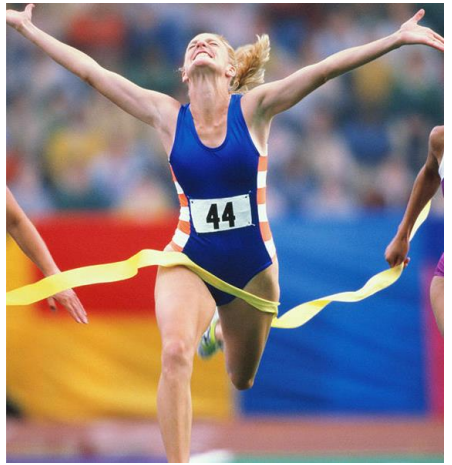
Observation: Programming a computer ***to be clever*** is harder than programming it ***to learn to be clever***

More computational power needed
(just as Moore’s Law is winding down!)

Tensor Processing Units (TPU) Origin Story

- 2013: Prepare for success-disaster of new DNN apps
  - Scenario with 100M users speaking to phones 3 minutes per day: If only CPUs, need double times whole data center fleet!
- Goal: Custom hardware to reduce the Total Cost of Ownership (TCO) of DNN inference phase by 10x
  - Must run existing apps developed for CPUs and GPUs
- Very short development cycle
  - Started TPU v1 project 2014, running in datacenter 15 months later: Architecture invention, compiler invention, hardware design, build, test, deploy

Reasons for TPU v1 Success



- 1 large 2D multiplier vs many smaller 1D units
  - Matrix multiplies benefit from 2D HW
- Narrower data types vs 32-bit Fl Pt
  ⇒ more efficient computation / memory
- TPU v1 drops CPU/GPU features (caches, branch predictors)
  ⇒ saves area & energy
  ⇒ reuse transistors for domain-specific on-chip memory
- Announced to world May 16, 2016
  - *"We've been running TPUs inside our data centers for more than a year, and have found them to deliver an order of magnitude better-optimized performance per watt for ML."*[27]

The Launching of “1000 Chips”

- Intel acquires DSA chip companies
  - Nervana: ($0.4B) August 2016
  - Movidius: ($0.4B) September 2016
  - MobilEye: ($15.3B) March 2017
  - Habana: ($2.0B) December 2019
- Alibaba, Amazon build inference chips
- >100 startups ($3B/yr) launch own bets
  - Coarse-Grained Reconfigurable Arch: SambaNova, ...
  - Analog computing: Mythic, …
  - Full silicon wafer computer: Cerebras, …
- Academia: TPUv1 paper ~4000 citations
- Most influential since RISC or Pentium Pro?

Helen of Troy
by Evelyn De Morgan

Dire Projections of Carbon Emissions for ML Training

Malthusian Predictions about ML Training

- Environmental cost to improve ML task (2024)?*
  "The answers are grim: Training such a model would cost US $100 billion and would produce as much carbon emissions as New York City does in a month. And if we estimate the computational burden of a 1 percent error rate, the results are considerably worse."

  Thompson, N.C., et al., October 2021.
  Deep Learning's Diminishing Returns: The Cost of Improvement is Becoming Unsustainable, *IEEE Spectrum*

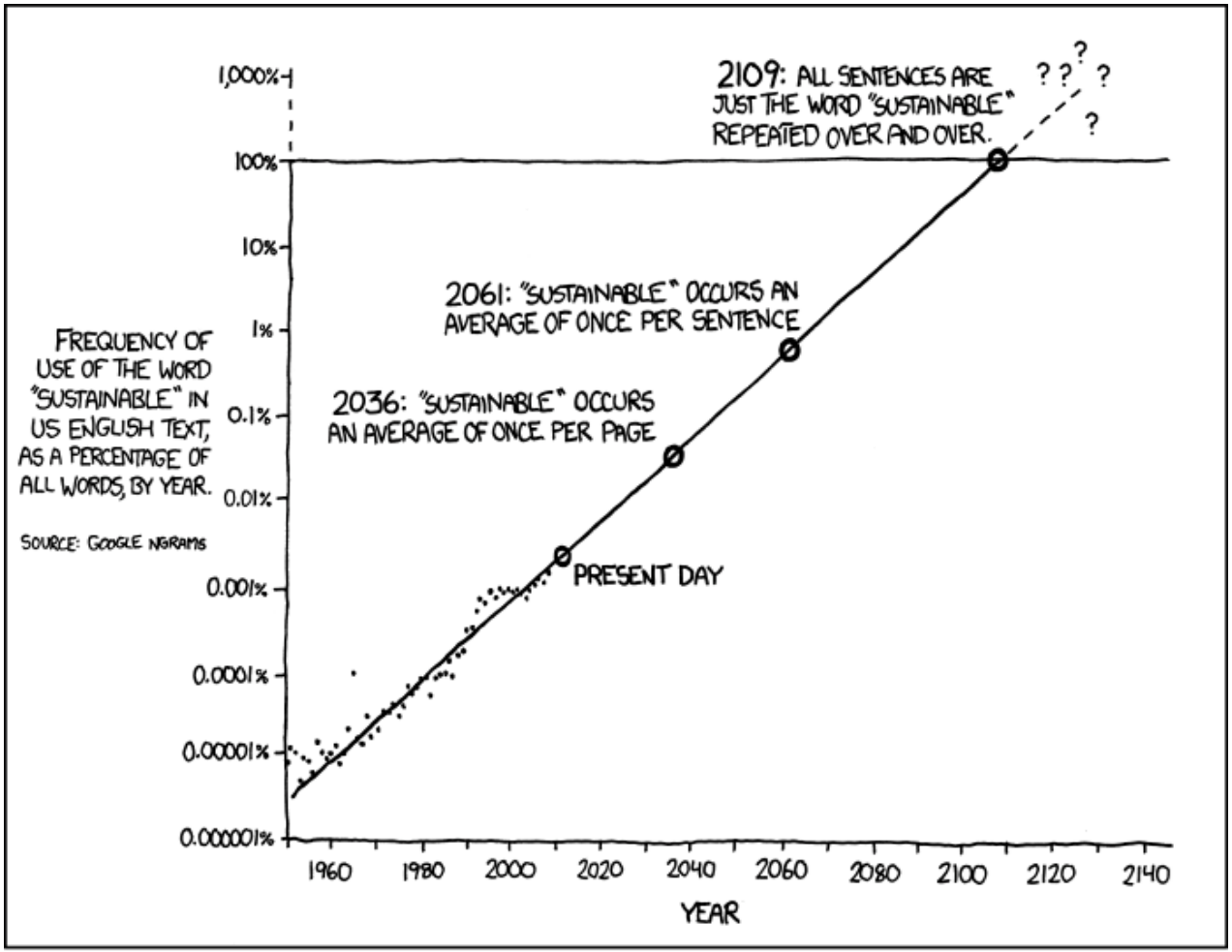
- *"In fact, by 2026, the training cost of the largest AI model predicted by the compute demand trend line would cost more than the total U.S. GDP."*
  [$20T]

  Lohn, J. and Musser, M., January 2022.
  AI and Compute—How Much Longer Can Computing Power Drive Artificial Intelligence Progress?
  Center for Security and Emerging Technology

* The ML task is object recognition using the Imagenet benchmark to reduce the error rate for an ML task* to a 5% from 11.5% today.

Google

https://xkcd.com/1007/



THE WORD "SUSTAINABLE" IS UNSUSTAINABLE.

How to document energy use and CO_2e emissions

KWh = Hours to train × Number of Processors × Average Power per Processor × PUE

- Many cloud companies publish quarterly PUE for all metros (e.g., Iowa, Oklahoma)
  - *Power Usage Effectiveness*: energy overhead "wasted" in datacenter (doesn't get to computers); if overhead is 50%, PUE = 1.5
- ML experts already know Hours to Train and Number of Processors
- Average Power per Processor:
  - Measure power while running

tCO_2e = KWh × tCO_2e per KWh

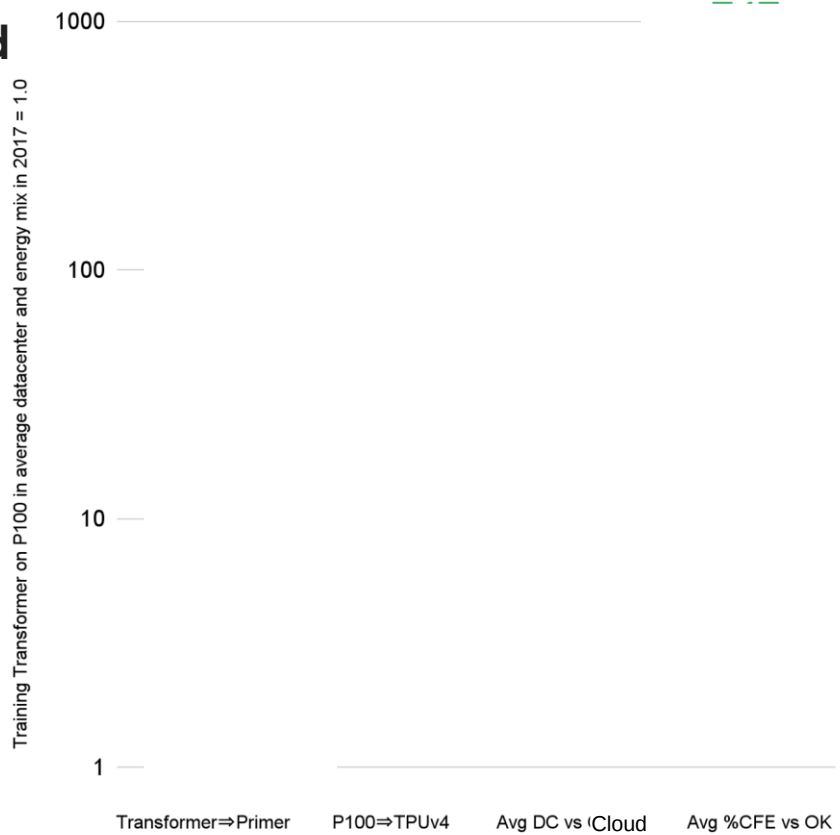
- Ask datacenter operator for energy cleanliness: tCO_2e per KWh
  - Varies 10x by location

Good News: Reduce energy 100x, CO_2e 1000x!

CO_2 equivalent emissions (CO_2e) include greenhouse gases

Energy efficiency in ML can be improved by 4 (multiplicative) best practices "4Ms of ML Energy Efficiency"

1. **Model. Transformer (2017) to Primer (2021) is 4x**
2. **Machine. P100 (2017) to newest GPU/TPU (2021) is 14x**
3. **Mechanization (datacenter efficiency). PUE from on premise average to Cloud average is 1.4x**
4. **Maps (geographic location, energy source). Avg %Carbon Free Energy (2017) to Oklahoma %CFE is 9x (2021)**



Thanks to Cliff Young for 4M mnemonic!

Putting it all together: The Supercomputer Fugaku

- Rather than use off-the-shelf CPUs or GPUs, designed a custom chip based on RISC (A64FX)
  - ~160,000 nodes, ~400 racks, ~440 PetaFLOPS/s 64b Fl PT, ~ 1 ExaFLOPS/s 32b, 30 MegaWatts
- Fastest in all 4 high performance computing benchmarks for 4 times
  - Top500, Graph500, HLP-AI, High Performance Conjugate Gradients (HPCG)
- Fast MLPerf HPC v1.0 ML training benchmark (CosmoFlow)
- Increased performance 100x but power increased only 3x ⇒ 33x in perf/Watt
  - Improved Machine of the 4Ms

Conclusion and Lessons Learned

- Moore's Law is slowing, Dennard scale is dead
- Chiplet allow more cores per package,
  but transistors not cheaper, helps only some applications
- More dramatic impact possible for DSAs than General Purpose CPUs
- Architects must learn a wider range of topics for DSAs,
  from algorithms to packaging technologies
  - As opposed to running SPEC benchmarks on a software CPU simulator
- DNN models still growing and changing fast
  - Running DNNs well potentially has large commercial impact
- Lower CO2e of training big models if follow best practices: Best **m**odel, Best **m**achine, Best **m**echanization (data center efficiency), Best **m**ap (clean energy)