# A New Golden Age for Computer Architecture:

## History, Challenges, and Opportunities

David Patterson

UC Berkeley and Google

December 5, 2018

*Full Turing Lecture:*

*https://www.acm.org/hennessy-patterson-turing-lecture*

# Lessons of last 50 years of Computer Architecture

1. *Software advances can inspire architecture innovations*
2. *Raising the hardware/software interface creates opportunities for architecture innovation*
3. *Ultimately the marketplace settles architecture debates*

# 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 |

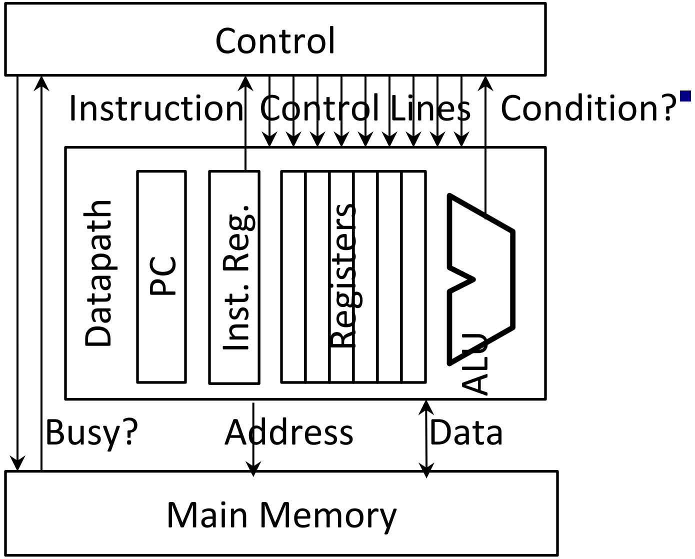
Each system had its own:

- 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



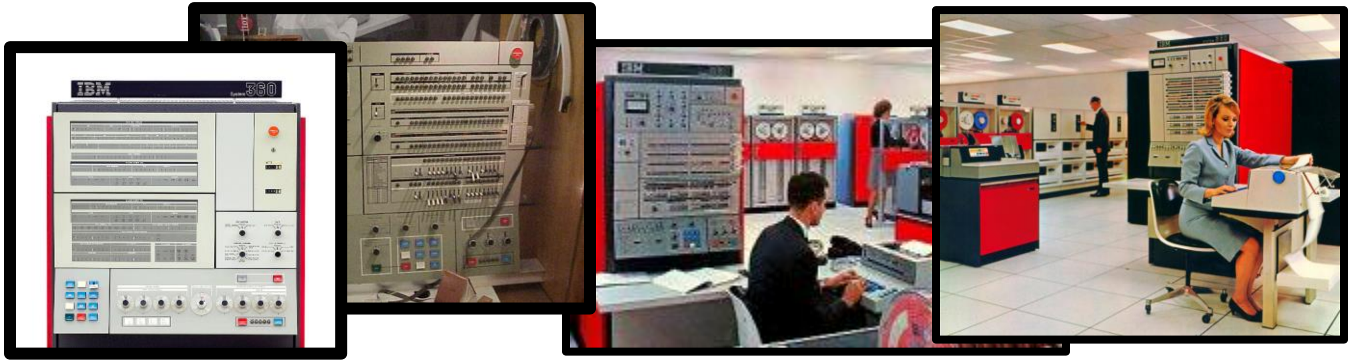
- **Maurice Wilkes** invented the idea of *microprogramming* to design the control unit of a processor*

  - 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

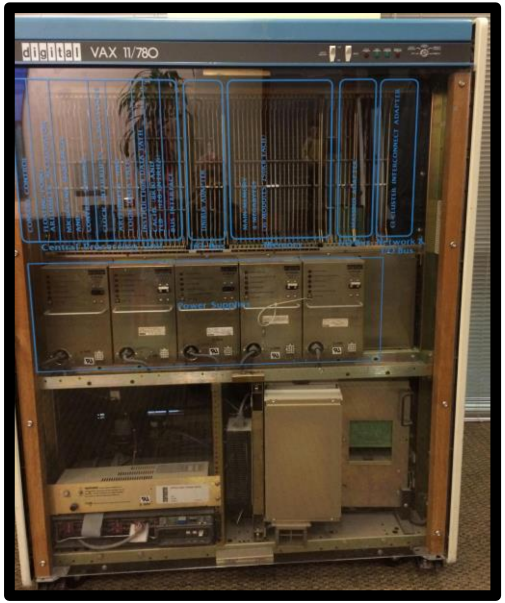
| Model | M30 | M40 | M50 | M65 |
|---|---|---|---|---|
| Datapath width | 8 bits | 16 bits | 32 bits | 64 bits |
| Microcode size | 4k x 50 | 4k x 52 | 2.75k x 85 | 2.75k x 87 |
| Clock cycle time (ROM) | 750 ns | 625 ns | 500 ns | 200 ns |
| Main memory cycle time | 1500 ns | 2500  ns | 2000 ns | 750 ns |
| Price (1964 $) | $192,000 | $216,000 | $460,000 | $1,080,000 |
| Price (2018 $) | $1,560,000 | $1,760,000 | $3,720,000 | $8,720,000 |



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

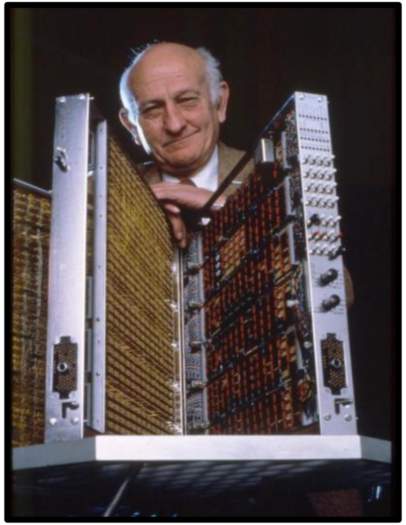


# Microprocessor Evolution

- 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

- *HW/SW interface rises from assembly to HLL programming*
  - 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!
- Patterson after ‘79 DEC sabbatical: repair microcode bugs in microprocessors?**
  - What’s magic about ISA interpreter in Writable Control Store? Why not other programs?



**John Cocke**

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

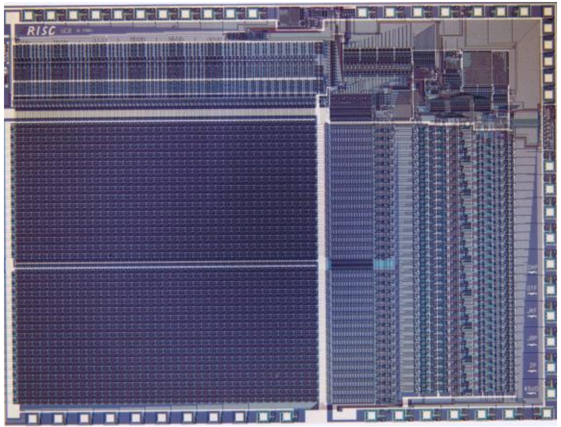
** “RISCy History,” David Patterson, May 30, 2018, Computer Architecture Today Blog

# From CISC to RISC

- Use RAM for instruction *cache* of user-visible instructions
  - *Software concept: Compiler vs. Interpreter*
  - 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
- Chaitin’s register allocation scheme* benefits load-store ISAs

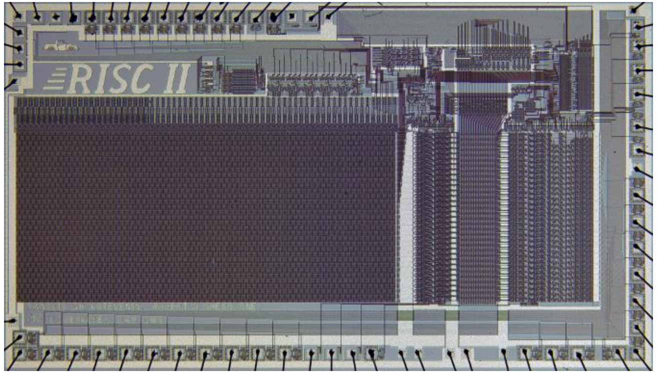
*Chaitin, Gregory J., et al. "Register allocation via coloring." *Computer languages* 6.1 (1981), 47-57.

# Berkeley and Stanford RISC Chips



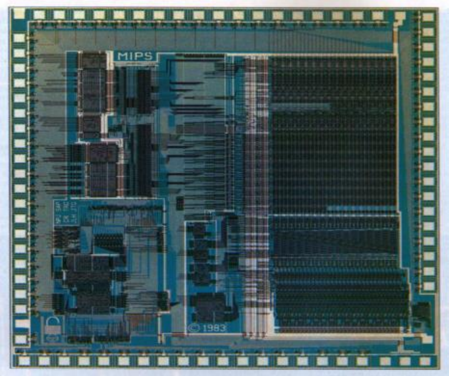
Fitzpatrick, Daniel, John Foderaro, Manolis Katevenis, Howard Landman, David Patterson, James Peek, Zvi Peshkess, Carlo Séquin, Robert Sherburne, and Korbin Van Dyke. "A RISCy approach to VLSI." *ACM SIGARCH Computer Architecture News* 10, no. 1 (1982)

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



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

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).



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)

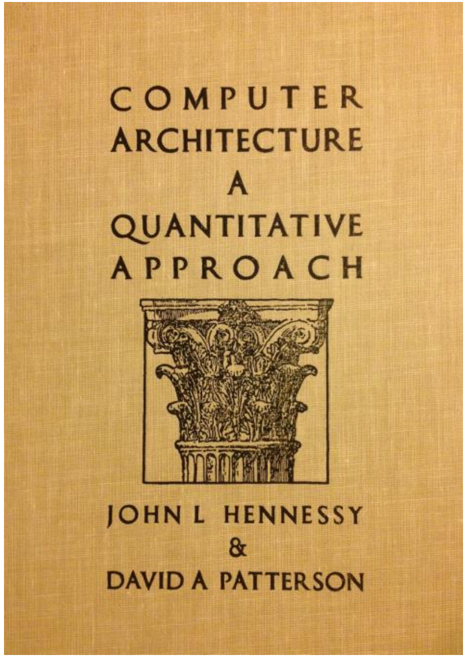
# “Iron Law” of Processor Performance: How RISC can win

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

- CISC executes fewer instructions / program (≈ 3/4X 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
A
QUANTITATIVE APPROACH

JOHN L HENNESSY
&
DAVID A PATTERSON

# CISC vs. RISC Today

**PC Era**

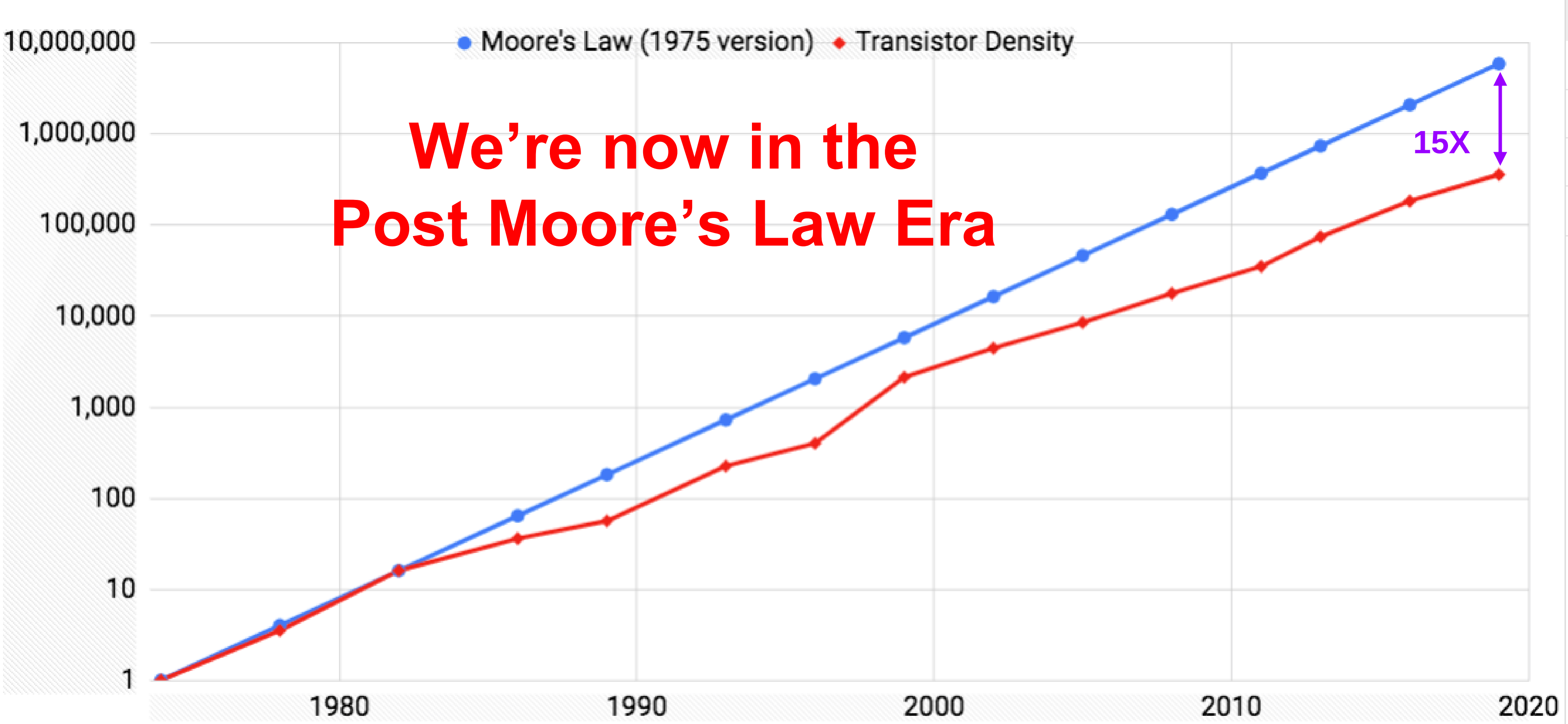
- Hardware translates x86 instructions into internal RISC instructions *(Compiler vs Interpreter)*
- Then use any RISC technique inside MPU
- > 350M / year !
- x86 ISA eventually dominates servers as well as desktops

**PostPC Era: Client/Cloud**

- IP in SoC vs. MPU
- Value die area, energy as much as performance
- > 20B total / year in 2017
- 99% Processors today are RISC
- *Marketplace settles debate*

*"A Decade of Mobile Computing", Vijay Reddi, 7/21/17, *Computer Architecture Today*

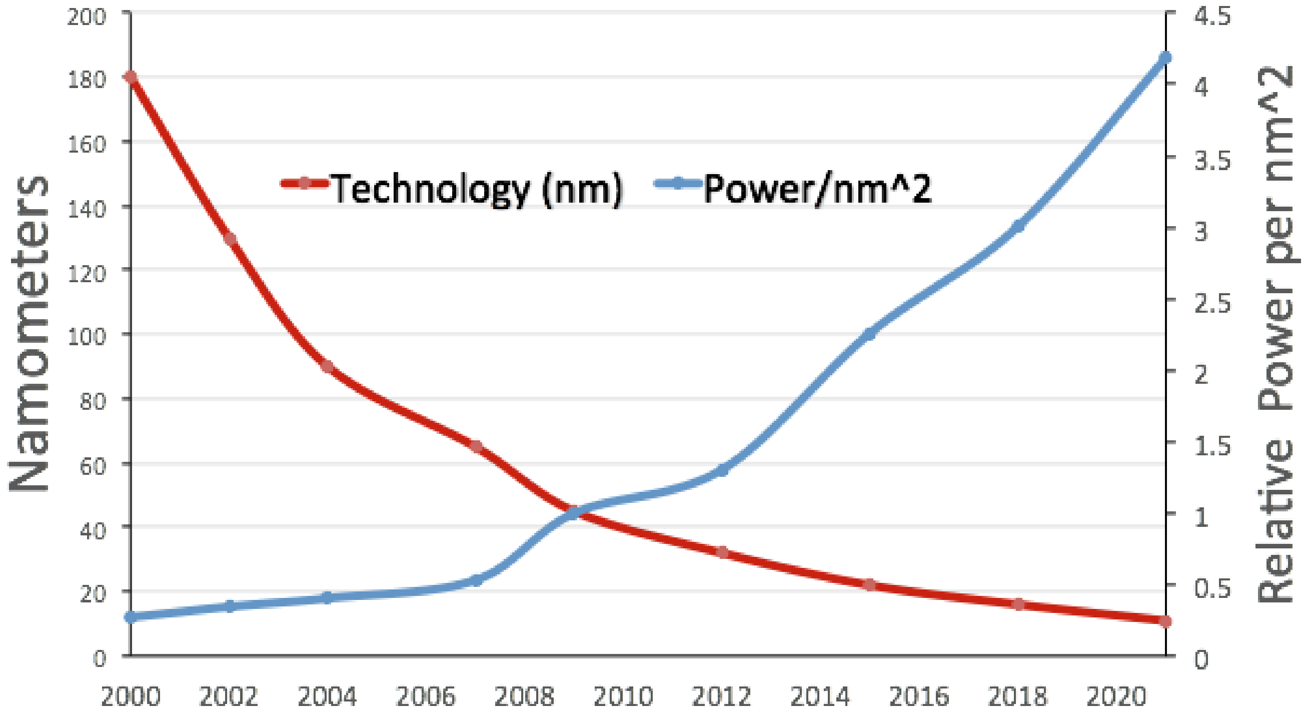
# Moore’s Law Slowdown in Intel Processes



**We’re now in the Post Moore’s Law Era**

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

# Technology & Power: Dennard Scaling

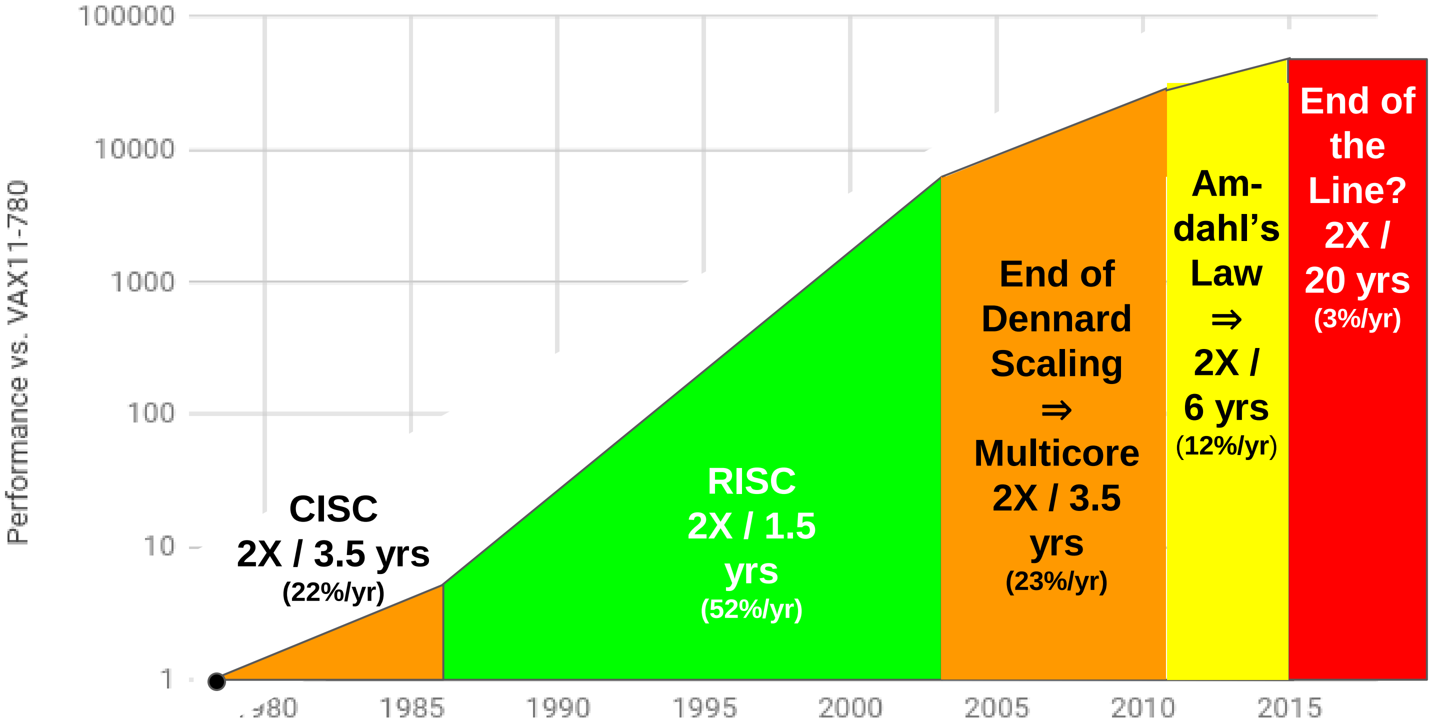


Power consumption based on models in "Dark Silicon and the End of Multicore Scaling," Hadi Esmaelizadeh, *ISCA*, 2011

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

# End of Growth of Single Program Speed?

40 years of Processor Performance



Apple A12 single thread performance (RISC ISA) = x86 Skylake single thread perf (SPEC), at much lower power, *Anandtech* 10/8/18

Based on SPECintCPU. Source: John Hennessy and David Patterson, Computer Architecture: A Quantitative Approach, 6/e. 2018

# Current Security Challenge

- Spectre: speculation ⇒ timing attacks that leak ≥10 kb/s
- More microarchitecture attacks on the way*
- Spectre is bug in computer architecture definition vs chip
- Need Computer Architecture 2.0 to prevent timing leaks**
- Software not yet secure ⇒ how can hardware help?

* "A Survey of Microarchitectural Timing Attacks and Countermeasures on Contemporary Hardware," Qian Ge, Yuval Yarom, David Cock, and Gernot Heiser, Journal of Cryptographic Engineering, April, 2018

** "A Primer on the Meltdown & Spectre Hardware Security Design Flaws and their Important Implications", Mark Hill, 2/15/18, Computer Architecture Today

# What Opportunities Left? (Part I)

- *Software advances can inspire architecture innovations*
- Why open source compilers and operating systems but not ISAs?

# What’s Different About RISC-V?
## (“RISC Five”, fifth UC Berkeley RISC)

- Free and Open
  - Anyone can use
  - More competition ⇒ More innovation
  - Pick ISA, then vendor

- For Cloud & Edge
  - From large to tiny computers

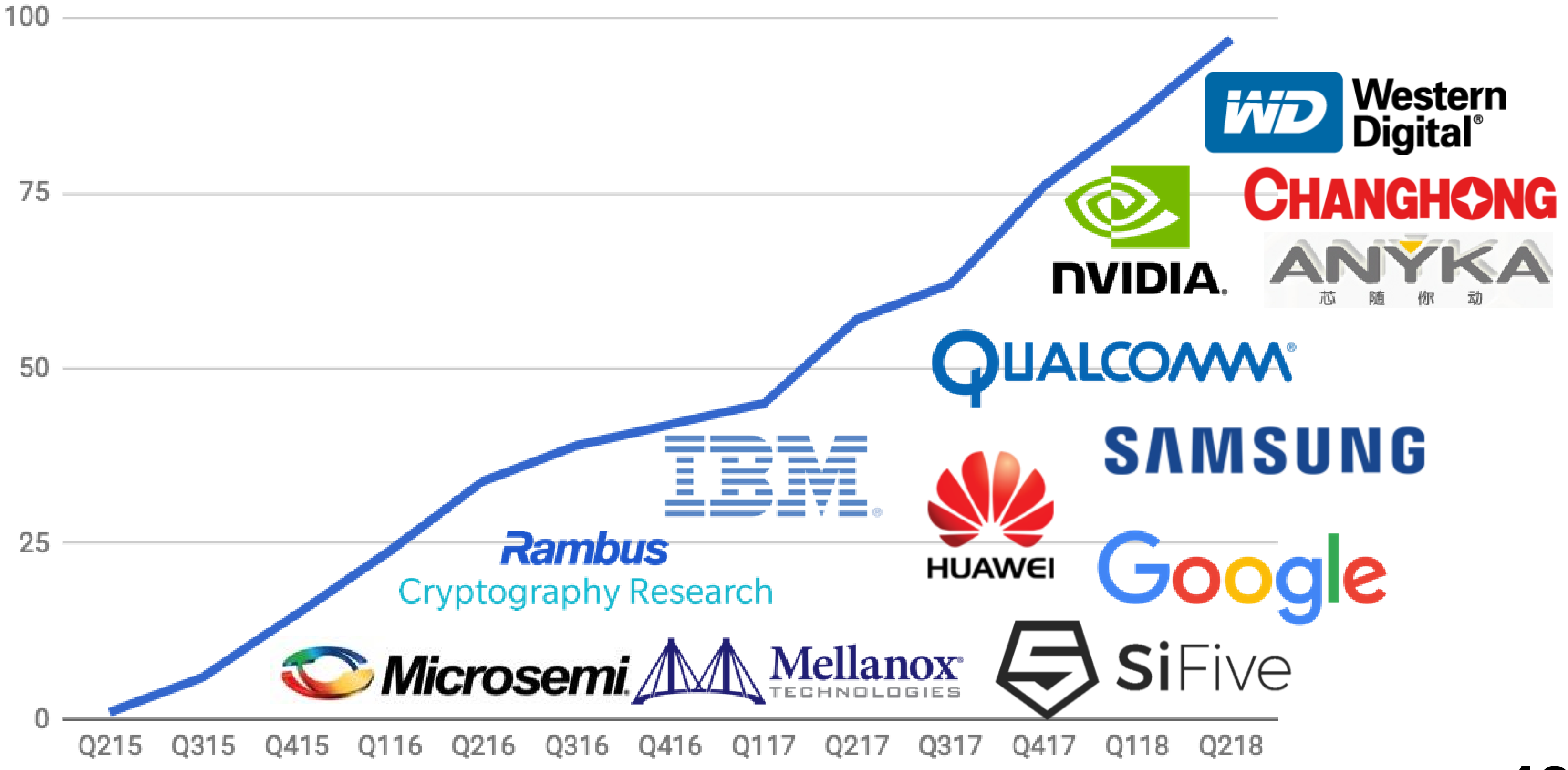
- Secure/Trustworthy
  - Design own secure core
  - Open cores ⇒ no secrets

- Simple, Elegant
  - 25 years later, learn from 1st gen RISCs*
  - Far simpler than ARM and x86
  - **Can add custom instructions**
  - **Input from software/architecture experts BEFORE finalize ISA**

- Community designed
  - RISC-V Foundation owns RISC-V ISA

* “How close is RISC-V to RISC-I?” David Patterson, 9/19/17, ASPIRE Blog

# Foundation Members since 2015



# Free & Open Instruction Set (ISA) vs Free & Open Source Hardware?

- Specifications
  - Instruction Set Architecture (for example, RISC-V)
  - Similar to Portable Operating System Interface (POSIX) standard in software
- Designs (“source code”)
  - RISC-V Rocket
  - Similar to Linux in software
- Products
  - OURS Pygmy chip
  - Similar to RedHat 7.5 in software

## 3 Types of Specifications or Designs

1. Free & Open
   - No fee, anyone can use
   - Can design it yourself, share with others, get from others
2. Licensable
   - Company owns, pay fee to use Can’t share with or get fro
3. Closed
   - Company owns, others cannot use

# Need Free & Open Specification To Have Free & Open Designs

| | *Specifications* | |
|---|---|---|
| Specifications | ***Free & Open Spec*** | |
| | ***Licensable Spec*** | |
| | ***Closed Spec*** | |

# Need Free & Open Specification To Have Free & Open Designs

| | | Designs (“Source”) | | | Products |
|---|---|---|---|---|---|
| | *Designs*<br>*Specifications* | **Free & Open Designs** | **Licensable Designs** | **Closed Designs** | |
| Specifications | **Free & Open Spec** | | | | |
| | **Licensable Spec** | | | | |
| | **Closed Spec** | | | | |

# Need Free & Open Specification
# To Have Free & Open Designs

| Specifications | | Designs (“Source”) | | | Products |
|---|---|---|---|---|---|
| | Designs<br>Specifications | Free & Open Designs | Licensable Designs | Closed Designs | |
| | Free & Open Spec | | | | |
| | Licensable Spec | | | | |
| | Closed Spec | | | 🟥 | Based on Close<br>Designs |

# Need Free & Open Specification
# To Have Free & Open Designs

Designs (“Source”)

Products

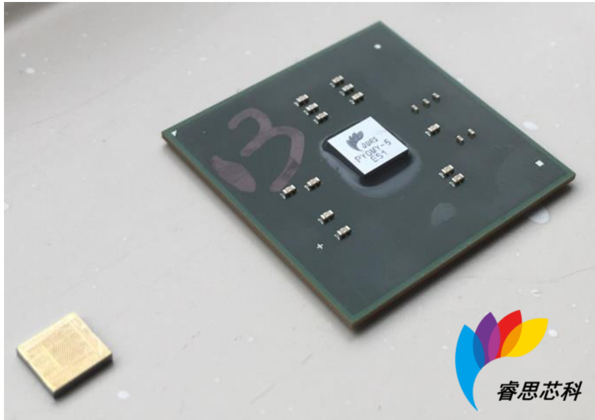
Specifications

| Designs<br>Specifications | Free & Open Designs | Licensable Designs | Closed Designs | |
|---|---|---|---|---|
| Free & Open Spec | | | | |
| Licensable Spec | | $5M + 4% | $25M | Based on Licen<br>or Closed |
| Closed Spec | | | 🟥 | Based on Close<br>Designs |

# Need Free & Open Specification
# To Have Free & Open Designs

Specifications

| | Designs (“Source”) | | | Products |
|---|---|---|---|---|
| *Designs*<br>*Specifications* | ***Free & Open Designs*** | ***Licensable Designs*** | ***Closed Designs*** | |
| ***Free & Open Spec*** | **“Open Source”** | 🟨 | 🟥 | Based on Free Open Licensed Closed |
| ***Licensable Spec*** | | 🟨 | 🟥 | Based on Licen or Closed |
| ***Closed Spec*** | | | 🟥 | Based on Close Designs |



OURS (睿思芯科) energy-efficient RISC-V AI Chip for IoT

## OURS Pygmy microprocessor

28nm HPC+ TSMC @ 600 MHz

From scratch to tapeout ~7 months

(Thanks to the RISC-V infrastructure)

Full RISC-V based heterogenous multicore architecture

64-bit control processor (RV64g)

~ 10mW active

12 energy-efficient AI engines based on custom RV vector extensions

INT8 : ~4 TOPS/watt

FP16 : ~0.35 TOPS/watt

1MB SRAM, LPDDR4 support

Retail price < ¥20 ($3)

# Security and Open Architecture

- Security community likes simple, verifiable (no trap doors), alterable, free and open architecture and implementations
- Equally important is number of people and organizations performing architecture experiments
  - Want all the best minds to work on security
- Plasticity of FPGAs + open source RISC-V implementations and SW ⇒ novel architectures can be deployed online, subjected to real attacks, evaluated & iterated in weeks vs years (even 100 MHz OK)
- RISC-V may become security exemplar via HW/SW codesign by architects and security experts

# What Opportunities Left? (Part II)

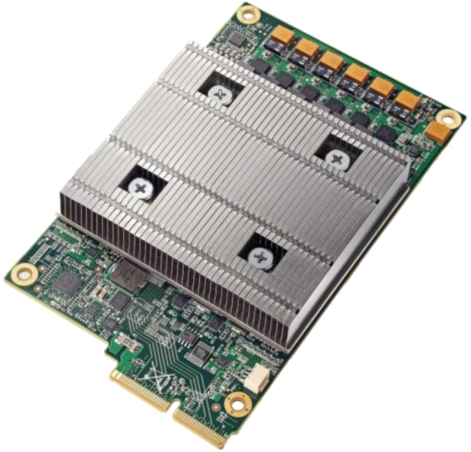
- Only performance path left is **Domain Specific Architectures (DSAs)**
  - Just do a few tasks, but extremely well
- Achieve higher efficiency by tailoring the architecture to characteristics of the domain
- Not one application, but a domain of applications
- Different from strict ASIC since still runs software

# Why DSAs Can Win (no magic) Tailor the Architecture to the Domain

- More effective parallelism for a specific domain:
  - SIMD vs. MIMD
  - VLIW vs. Speculative, out-of-order
- More effective use of memory bandwidth
  - User controlled versus caches
- Eliminate unneeded accuracy
  - IEEE replaced by lower precision FP
  - 32-64 bit integers to 8-16 bit integers
- Domain specific programming language provides path for software

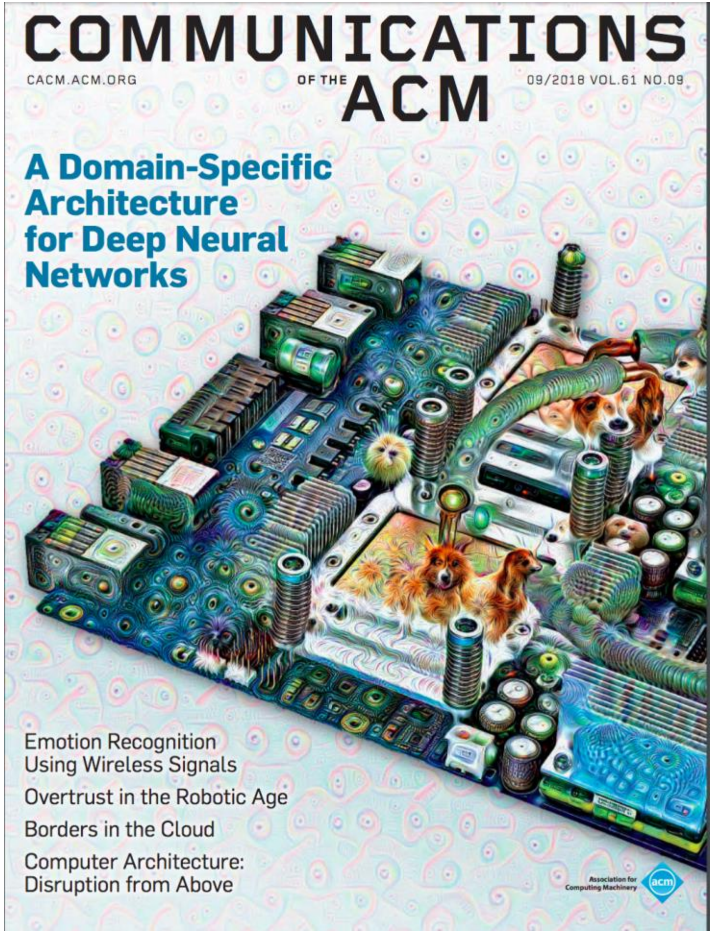
# Tensor Processing Unit v1 (Announced May 2016)

Google-designed chip for neural net **inference**



In production use for 3 years: used by billions on search queries, for neural machine translation, for AlphaGo victory over Lee Sedol, …

A Domain-Specific Architecture for Deep Neural Networks, Jouppi, Young, Patil, Patterson, *Communications of the ACM*, September 2018



COMMUNICATIONS OF THE ACM

CACM.ACM.ORG 09/2018 VOL.61 NO.09

A Domain-Specific Architecture for Deep Neural Networks

Emotion Recognition Using Wireless Signals

Overtrust in the Robotic Age

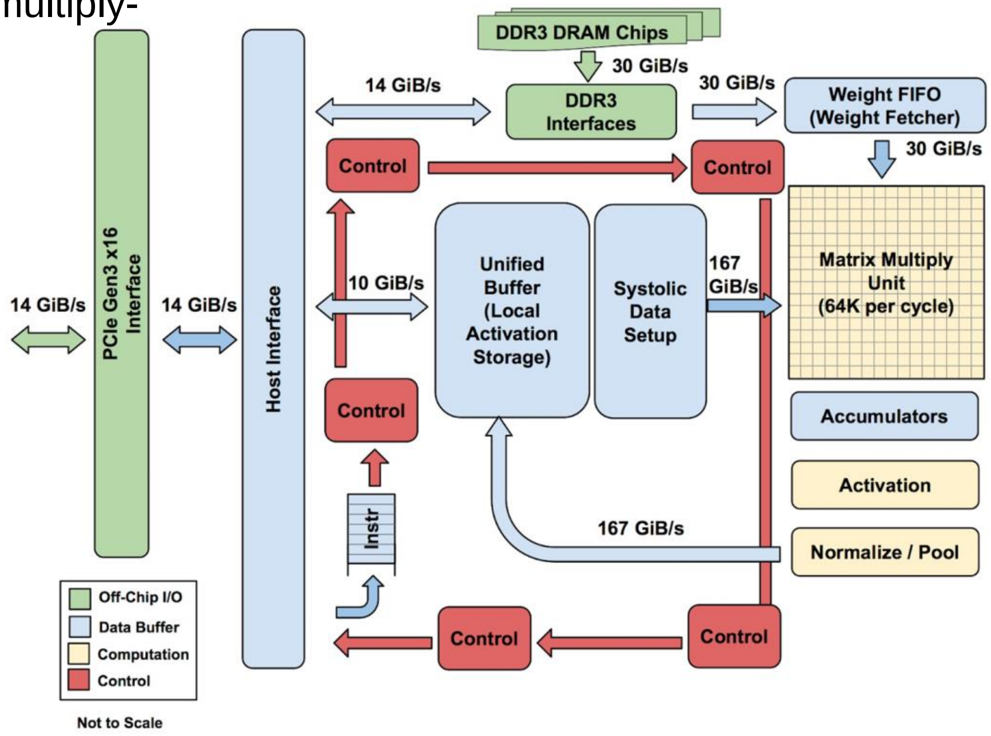
Borders in the Cloud

Computer Architecture: Disruption from Above

Association for Computing Machinery acm

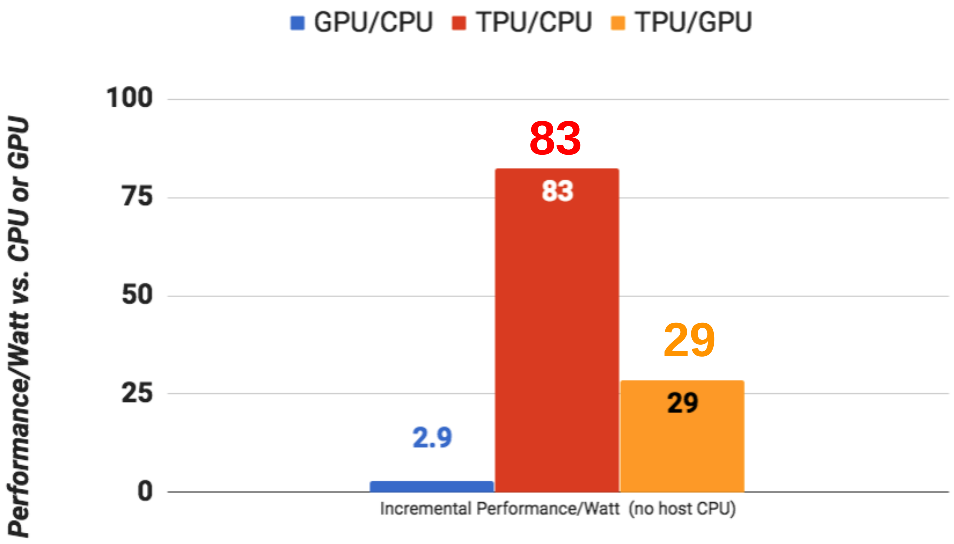
# TPU: High-level Chip Architecture

- The Matrix Unit: 65,536 (256x256) 8-bit multiply-accumulate units
- 700 MHz clock rate
- Peak: 92T operations/second
  - 65,536 * 2 * 700M
- >25X as many MACs vs GPU
- >100X as many MACs vs CPU
- 4 MiB of on-chip Accumulator memory + 24 MiB of on-chip Unified Buffer (activation memory)
- 3.5X as much on-chip memory vs GPU
- 8 GiB of off-chip weight DRAM memory



# Perf/Watt TPU vs CPU & GPU

Using production applications vs contemporary CPU and GPU



Measure performance of Machine Learning?

See MLPerf.org ("SPEC for ML")

- Benchmark suite being developed by
  23 companies and
  7 universities
- 1st Results November 2018

# Current Neural Network Architecture Debate

- Google TPU: 1 core per chip, large 2D multiplier, software controlled memory (instead of caches)
- Nvidia GPU: 80+ cores, many threads, many registers
- Microsoft FPGA: customize “hardware” to application
- Intel CPU: 30+ cores, 3 levels of caches, SIMD instructions
    - Also bought Altera that supplies Microsoft’s FPGAs
    - Also bought Nervana, Movidius, MobilEye to offer custom chip DSA
- > 45 startups with their own architecture bets

# Conclusion

- End of Moore's Law, Dennard Scaling, General Purpose Performance, Security Challenges require Innovation in Computer Architecture
- Open Instruction Set Architecture vs. Licensable or Closed Instruction Set Architecture
- Agile Hardware Development vs. Large Scale Hardware Development
- Domain Specific Architecture vs. General Purpose Architectures

A New
Comp. Architecture
Golden Age