A New Golden Age for Computer Architecture:

History, Challenges, and Opportunities

David Patterson
UC Berkeley and Google

August 22, 2019

Full Turing Lecture:
https://www.acm.org/hennessy-patterson-turing-lecture

1

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*

2

IBM Compatibility Problem in Early 1960s

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

701
650
702
1401

7094
7074
7080
7010

Each system had its own:

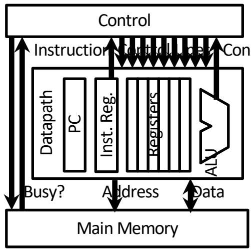
- Instruction set architecture (ISA)
- I/O system and Secondary Storage: magnetic tapes, drums and disks
- Assemblers, compilers, libraries,...
- Market niche: business, scientific, real time, ...

IBM System/360 – one ISA to rule them all

3

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.

4

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

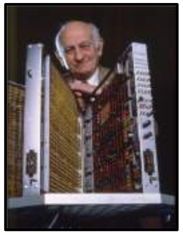
- Logic, RAM, ROM all implemented using same transistors
- Semiconductor RAM ≈ same speed as ROM
- With Moore's Law, memory for control store could grow
- Since RAM, easier to fix microcode bugs
- Allowed more complicated ISAs (CISC)
- Minicomputer (TTL server) example:
  - -Digital Equipment Corp. (DEC)
  - -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!



John Cocke

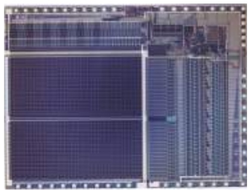
* "A Characterization of Processor Performance in the VAX-11/780," J. Emer and D.Clark, *ISCA*, 1984.

From CISC to RISC

- Use RAM for instruction *cache* of user-visible instructions
  - *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.

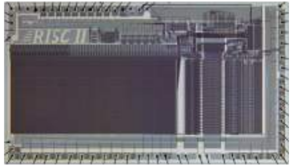
Berkeley and Stanford RISC Chips



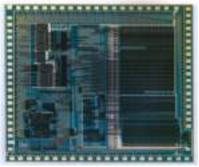
Fitzpatrick, Daniel, John Foderaro, Manolis Katevenis, Howard Landman, David Patterson, James Peek, Zvi Peshkess, Carlo Séquin, Robert Sherburne, and Korbin Van Dyke. "A RISCy approach to VLSI." *ACM SIGARCH Computer Architecture News* 10, no. 1 (1982)

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-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



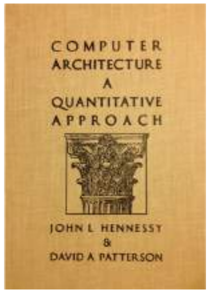
Stanford MIPS (1983) contains 25,000 transistors, was fabbed in 3 µm & 4 µm NMOS, ran at 4 MHz (3 µm), and size is 50 mm^2 (4 µm) (Microprocessor without Interlocked Pipeline Stages)

"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.



CISC vs. RISC Today

PC Era

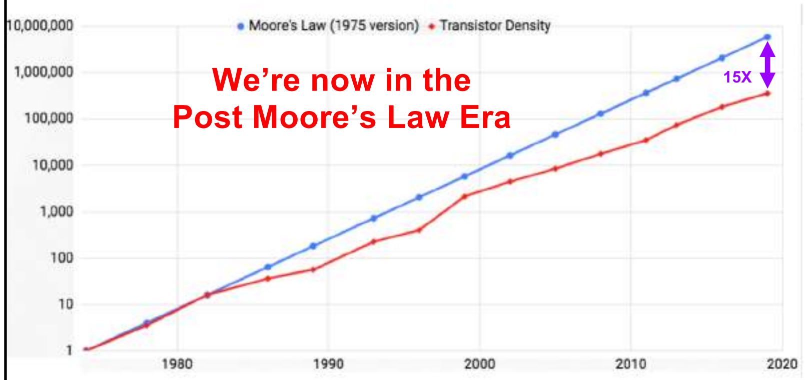
- Hardware translates x86 instructions into internal RISC instructions *(Compiler vs Interpreter)*
- Then use any RISC technique inside MPU
- > 350M / year !
- x86 ISA eventually dominates servers as well as desktops

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*

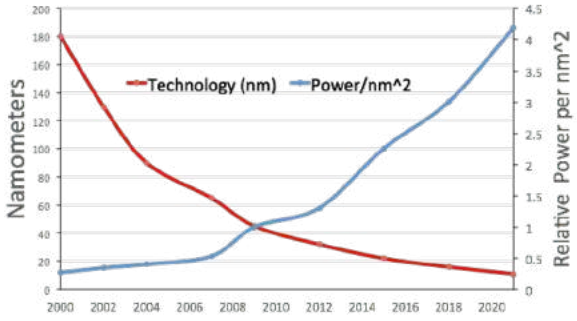
Moore's Law Slowdown in Intel Processors



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.*

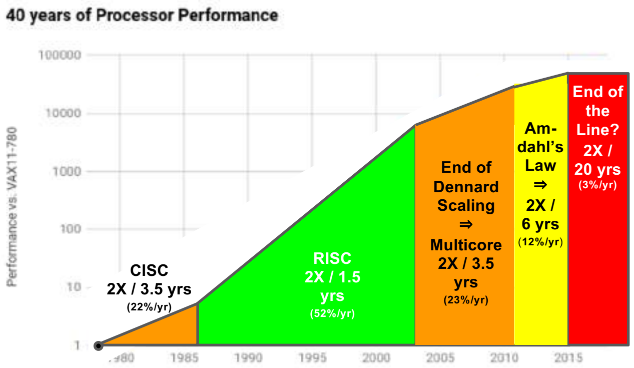
Technology & Power: Dennard Scaling



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?



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

Looks Bad!

"*What we have before us are some breathtaking opportunities disguised as insoluble problems*."

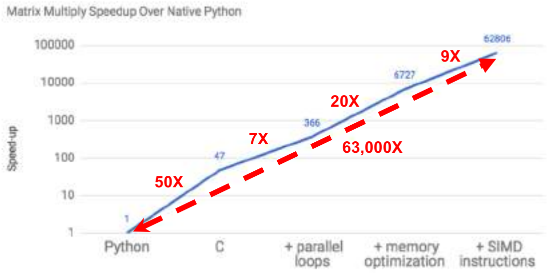
-John Gardner, 1965

What Opportunities Left? (Part I)

- SW-centric
  - Modern scripting languages are interpreted, dynamically-typed and encourage reuse
  - Efficient for programmers but not for execution
- HW-centric
  - Only path left is *Domain Specific Architectures*
  - Just do a few tasks, but extremely well
- Combination:
  - Domain Specific Languages & Architectures
  - Raises level of HW/SW Interface

What's the Opportunity?

Matrix Multiply: relative speedup to a Python version (on 18 core Intel CPU)



from: "There's Plenty of Room at the Top," Leiserson, et. al., *Science*, *to appear*.

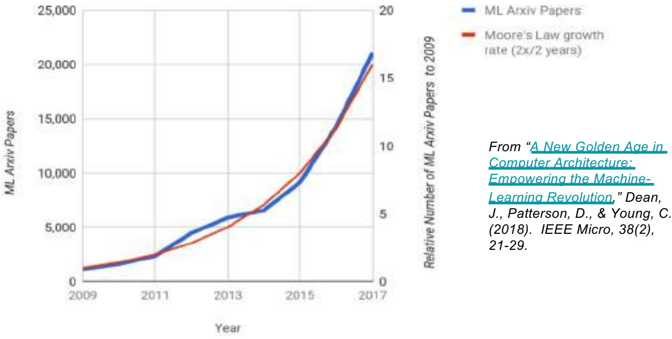
What Opportunities Left?

- 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

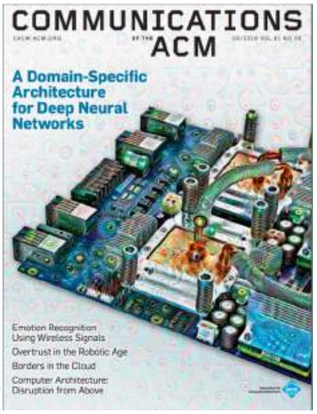
Deep learning is causing a machine learning revolution



From "A New Golden Age in Computer Architecture: Empowering the Machine-Learning Revolution," Dean, J., Patterson, D., & Young, C. (2018). IEEE Micro, 38(2), 21-29.

Tensor Processing Unit v1 (Announced May 2016)

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



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

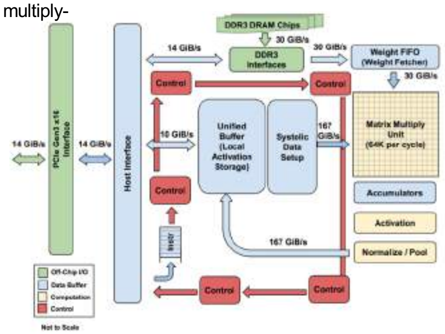
Computer Architecture: Disruption from Above

In production use for 4 years: used by billions on search queries, for neural machine translation, for AlphaGo match, …

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

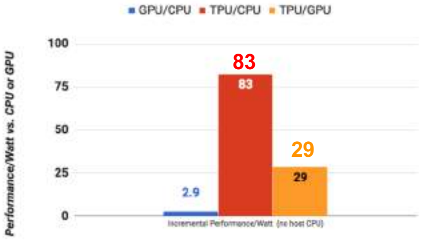
TPU: High-level Chip Architecture

- 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?

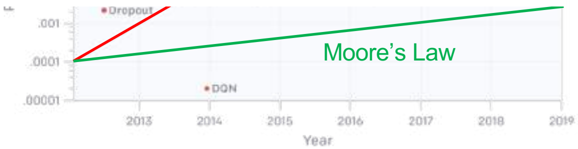
See MLPerf.org ("SPEC for ML")

- Benchmark suite being developed by 23 companies and 7 universities
- 1st Results Public 12/12/18

ML Training Trends

Moore's Law performance doubles every 18 months

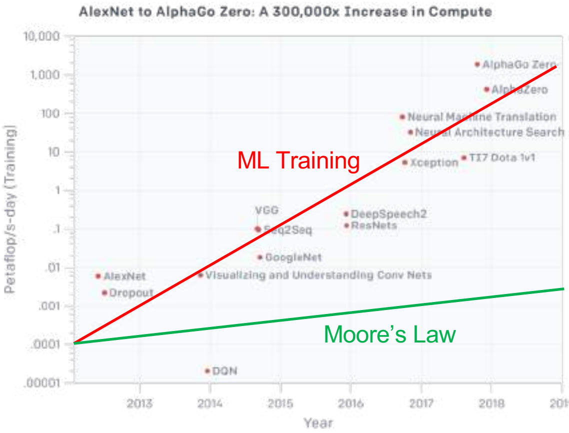
From "AI and Compute," Dario Amodei and Danny Hernandez, May 16, 2018



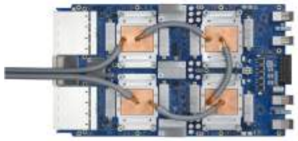
ML Training Trends

Since 2012,
AI training state of the art compute demand 10X per year!
(Moore's Law "only" 10X in 5 years)

From "AI and Compute," Dario Amodei and Danny Hernandez, May 16, 2018



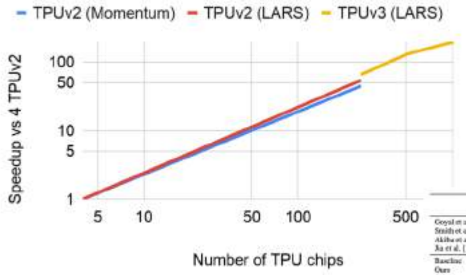
Training: TPUv2 (5/2017),TPUv3 (5/2018)



Peak: 11.5 PetaFLOP/s

Peak: >100 PetaFLOP/s

ResNet-50 Speedup: Batch Size, Optimizer, Accuracy



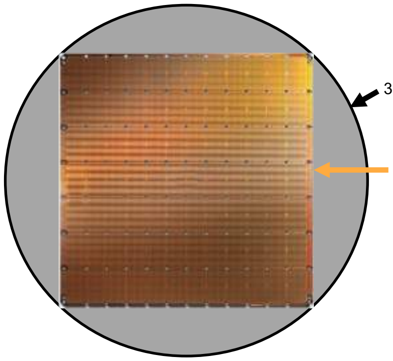
Ying, C., Kumar, S., Chen, D., Wang, T. and Cheng, Y., December 2018. Image Classification at Supercomputer Scale. *arXiv preprint arXiv:1811.06992.*

	Hardware	Chips	Batch	Optimizer	BN	Accuracy	Time
Goyal et al. [6]	P100	256	8192	Momentum	Local	76.3%	1 hour
Smith et al. [16]	TPU v2	128	8192 → 16384	Momentum	Local	76.1%	30 mins.
Akiba et al. [2]	P100	1024	32768	RMS + Mom.	Local	74.9%	15 mins.
Jia et al. [10]	P40	1024	65536	LARS	Local	76.2%	8.7 mins.
Baseline	TPU v2	4	1024	Momentum	Local	76.3%	7.2 hours
Ours	TPU v2	256	16384	Momentum	Local	75.3%	9.7 mins.
Ours	TPU v2	256	32768	LARS	Local	76.3%	8.0 mins.
Ours	TPU v3	512	32768	LARS	Local	76.4%	3.3 mins.
Ours	TPU v3	1024	65536	LARS	Local	75.2%	1.8 mins.
Ours	TPU v3	1024	32768	LARS	Distributed	**76.3%**	**2.2 mins.**

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 (20MB registers), small multipliers, caches, scatter/gather & coalescing HW
- 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
- > 100 startups with their own architecture bets
- *#3. Ultimately the marketplace settles architecture debates*

Cerebus announces ML Training "Chip" 8/19/19



300 mm (12 inch) wafer

215 x 215 mm
(8.5 x 8.5 inch)
"chip"

What Opportunities Left? (Part II)

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

RISC-V Origin Story

- UC Berkeley Research using x86 & ARM?
  - Impossible – too complex *and* IP issues
- 2010 started "3-month project" to develop own clean-slate ISA
  - Krste Asanovic, Andrew Waterman, Yunsup Lee, Dave Patterson
- 4 years later, released frozen base user spec

Why are outsiders complaining about changes of RISC-V in Berkeley classes?

33

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

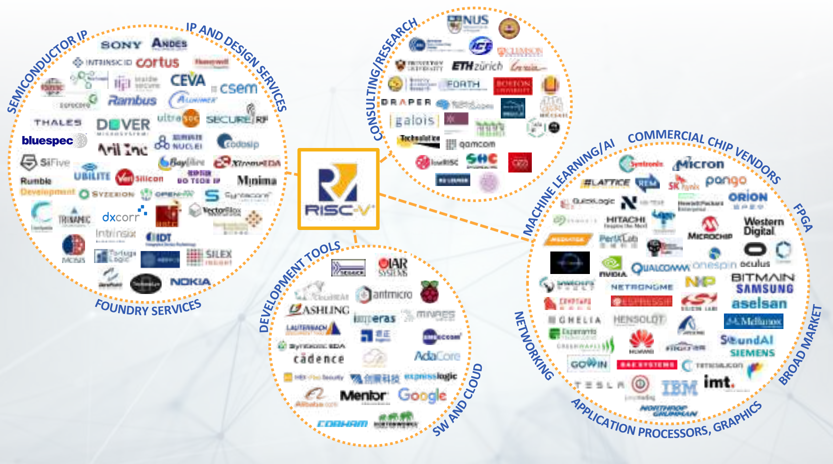
- For Cloud & Edge
  - From large to tiny computers
- Secure/Trustworthy
  - Design own secure core
  - Open cores ⇒ no secrets

- 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**

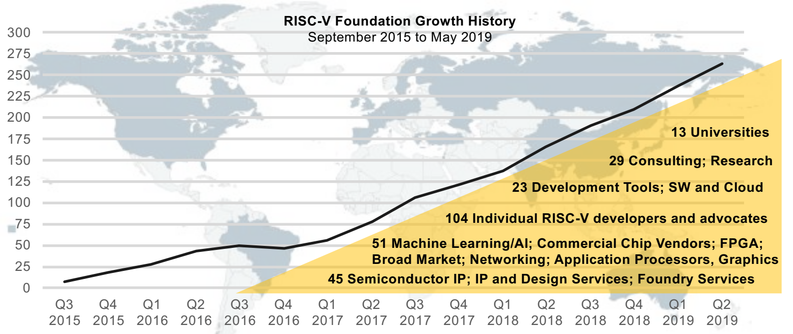
- Community evolves
  - RISC-V Foundation owns RISC-V ISA

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

34



More than 300 RISC-V Members in 28 Countries Around the World

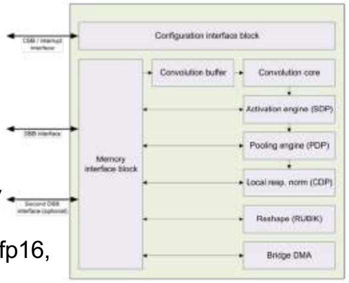


May 2019

36

NVDLA: An Open DSA and Implementation

- NVDLA: NVIDIA Deep Learning Accelerator for DNN Inference
- Free & Open: All SW, HW, and documentation on GitHub
- Scalable, configurable design
  - Each block operates independently or in pipeline to bypass memory
  - Data type configurable: int8, int16, fp16,
  - 2D MAC array configurable: 8 to 64 x 4 to 64
  - Size scales 6X (0.5 - 3mm^2), power scales 15X (20 - 300 mW)
- RISC-V core as host (optional)



Security and Open Architecture

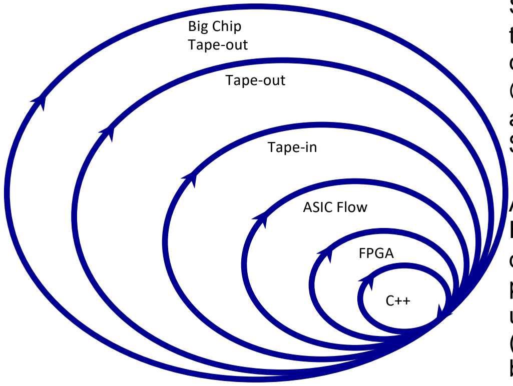
- Security community likes simple, verifiable (no trap doors), alterable, free and open architecture and implementations
- Equally important is number of people and organizations performing architecture experiments
  - Want all the best minds to work on security
- Plasticity of FPGAs + open source RISC-V 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 III)

- *Software advances can inspire innovations*
- Agile: small teams do short development between working but incomplete prototypes and get customer feedback per step
- Scrum team organization
  - 5 - 10 person team size
  - 2 - 4 week sprints for next prototype iteration
- New CAD enables SW Dev techniques to make small teams productive via abstraction & reuse => **Agile Hardware Development**

Agile Hardware Development Methodology

Lee, Y., Waterman, A., Cook, H., Zimmer, B., Keller, B., Puggelli, A., ... & Chiu, P. F. (2016). "An agile approach to building RISC-V microprocessors." *IEEE Micro*, *36*(2), 8-20.



Small chip tape-out 100 chips 1x1mm @ 28nm is affordable at $14,000!

AWS FPGA F1 instance ⇒ develop new prototypes using cloud (nothing to buy)

Lessons of last 50 years of Computer Architecture

1. *Software advances can inspire architecture innovations*
   - ○ Microprogramming - control as SW
   - ○ RISC, x86 ISA - (Hardware) translator vs interpreter
   - ○ Open Architectures & Implementations
   - ○ Agile Hardware Development
2. *Raising the HW/SW interface enables arch. opportunities*
   - ○ Assembly to HLL ⇒ RISC
   - ○ HLL to Domain Specific Language⇒DSA
3. *Ultimately the marketplace settles architecture debates*
   - ○ Losers: 432
   - ○ Winners: IBM S/360, 8086 (PC Era), RISC (Post PC Era)
   - ○ Open vs Proprietary ISA (RISC-V vs ARM): Too soon to tell
   - ○ ML DSA (SIMD vs GPU vs TPU vs FPGA vs startups): Too soon to tell

Questions?

COMMUNICATIONS OF THE ACM
CACM.ACM.ORG
A New Golden Age for Computer Architecture
Agriculture Technology
Monitoring Noise Pollution
The Computational Sprinting Game
Blockchain from a Distributed Computing Perspective

Quantum Computing to the Rescue?

- Google, IBM, Microsoft pursuing Quantum Computing
- Physics, Math, Theory results are beautiful
- For Cloud, not Client
- #1 Recommendation of Quantum Workshop May 2018:*

First and foremost, there is an overarching need for new Quantum Computing algorithms that can make use of the limited qubit counts and precisions available in the foreseeable future. Without a "killer app" or at least a useful app runnable in the first ten years, progress may stall.

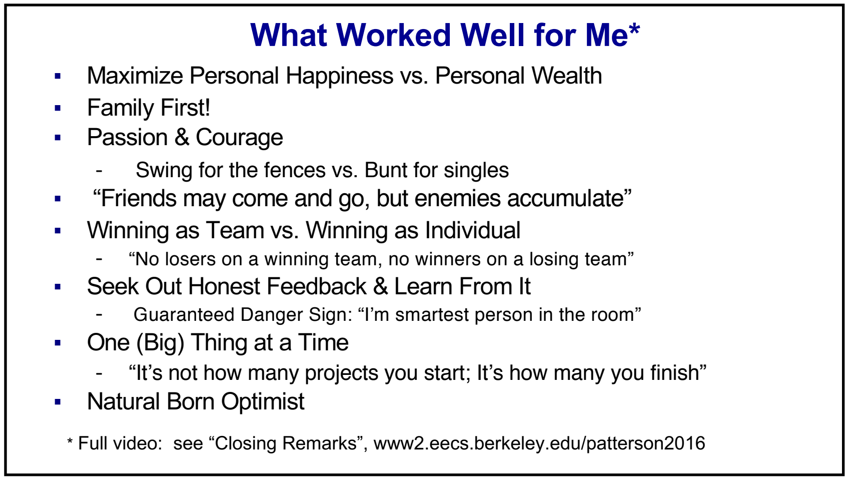
* "Next Steps in Quantum Computing: Computer Science's Role," May 22-23, 2018, Washington D.C., Computing Community Consortium

Quantum Computing to the Rescue?

- *Quantum Computing - Progress and Prospects**
  - 12/2018 consensus study from National Academies
- "*Significant technical and financial issues remain towards building a large, fault-tolerant quantum computer and one is unlikely to be built within the coming decade."*

  Gwynne, Peter. (2019). "Practical quantum computers still at least a decade away." *Physics World*. 32. 9-9. 10.1088/2058-7058/32/1/14.

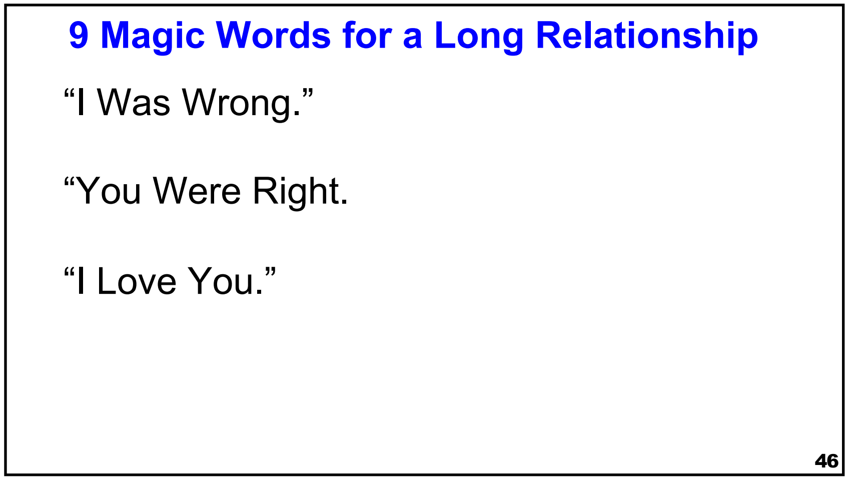
*Mark Horowitz (Chair, NAE, Stanford, EE), Alán Aspuru-Guzik (U. Toronto, Chemistry), David Awschalom (NAE & NAS, U. Chicago, Physics), Robert Blakley (Citigroup), Dan Boneh (NAE, Stanford, CS), Susan Coppersmith (NAS, U. Wisconsin, Physics), Jungsang Kim (Duke, Physics & CS), John Martinis (UCSB & Google), Margaret Martonosi (Princeton, CS), Michele Mosca (U. Waterloo, Math & Physics), William Oliver (MIT, Physics), Krysta Svore (Microsoft), Umesh Vazirani (NAE, Berkeley, CS), National Academies, Washington D.C. https://www.nap.edu/catalog/25196/quantum-computing-progress-and-prospects



What Worked Well for Me*

- Maximize Personal Happiness vs. Personal Wealth
- Family First!
- Passion & Courage
  - Swing for the fences vs. Bunt for singles
- "Friends may come and go, but enemies accumulate"
- Winning as Team vs. Winning as Individual
  - "No losers on a winning team, no winners on a losing team"
- Seek Out Honest Feedback & Learn From It
  - Guaranteed Danger Sign: "I'm smartest person in the room"
- One (Big) Thing at a Time
  - "It's not how many projects you start; It's how many you finish"
- Natural Born Optimist

* Full video: see "Closing Remarks", www2.eecs.berkeley.edu/patterson2016



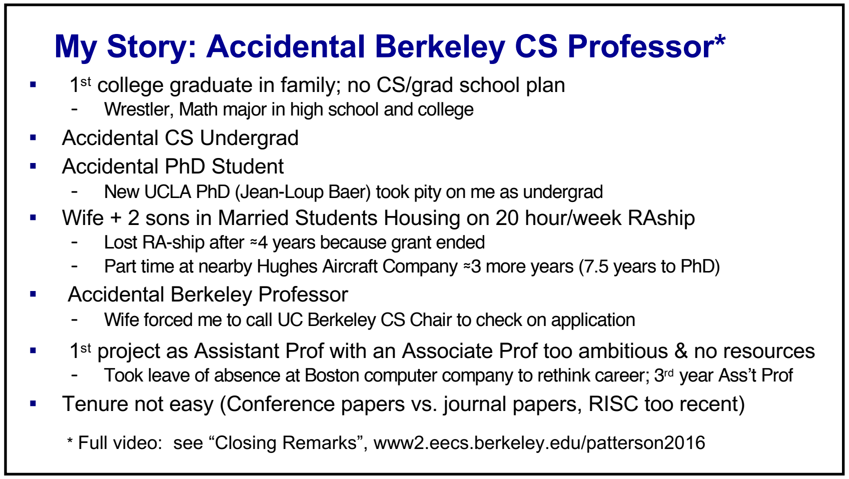
9 Magic Words for a Long Relationship

"I Was Wrong."

"You Were Right.

"I Love You."

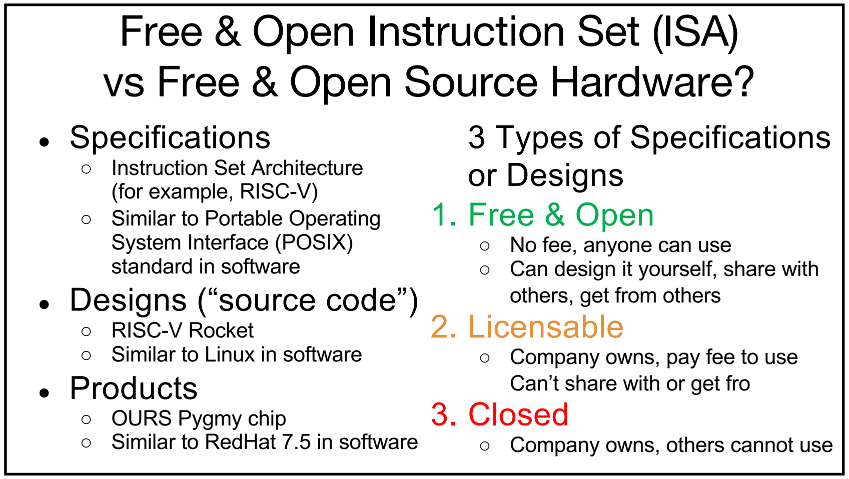
46



My Story: Accidental Berkeley CS Professor*

- 1st college graduate in family; no CS/grad school plan
  - Wrestler, Math major in high school and college
- Accidental CS Undergrad
- Accidental PhD Student
  - New UCLA PhD (Jean-Loup Baer) took pity on me as undergrad
- Wife + 2 sons in Married Students Housing on 20 hour/week RAship
  - Lost RA-ship after ≈4 years because grant ended
  - Part time at nearby Hughes Aircraft Company ≈3 more years (7.5 years to PhD)
- Accidental Berkeley Professor
  - Wife forced me to call UC Berkeley CS Chair to check on application
- 1st project as Assistant Prof with an Associate Prof too ambitious & no resources
  - Took leave of absence at Boston computer company to rethink career; 3rd year Ass't Prof
- Tenure not easy (Conference papers vs. journal papers, RISC too recent)

* Full video: see "Closing Remarks", www2.eecs.berkeley.edu/patterson2016



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

Specifications

	Designs (“Source”)			Products
Designs Specifications	**Free & Open Designs**	**Licensable Designs**	**Closed Designs**	
Free & Open Spec				
Licensable Spec				
Closed Spec				

Need Free & Open Specification To Have Free & Open Designs

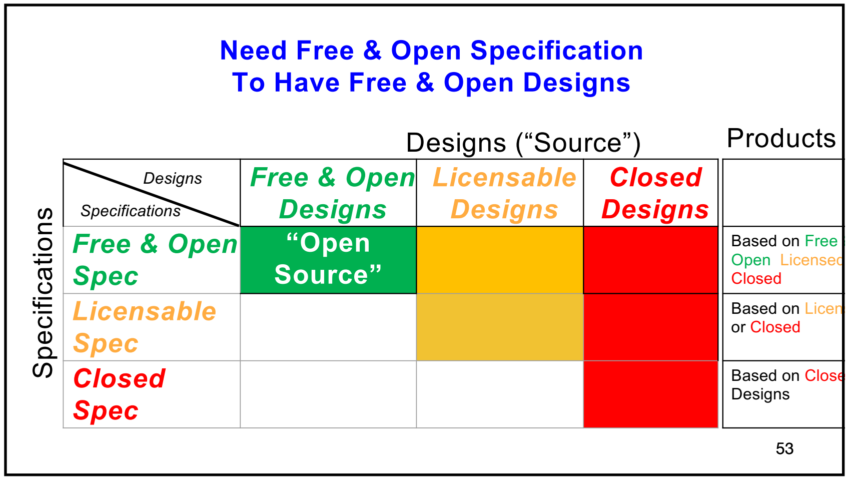
Specifications

	Designs (“Source”)			Products
Designs Specifications	**Free & Open Designs**	**Licensable Designs**	**Closed Designs**	
Free & Open Spec				
Licensable Spec				
Closed Spec			🟥	Based on Close Designs

Need Free & Open Specification To Have Free & Open Designs

Specifications

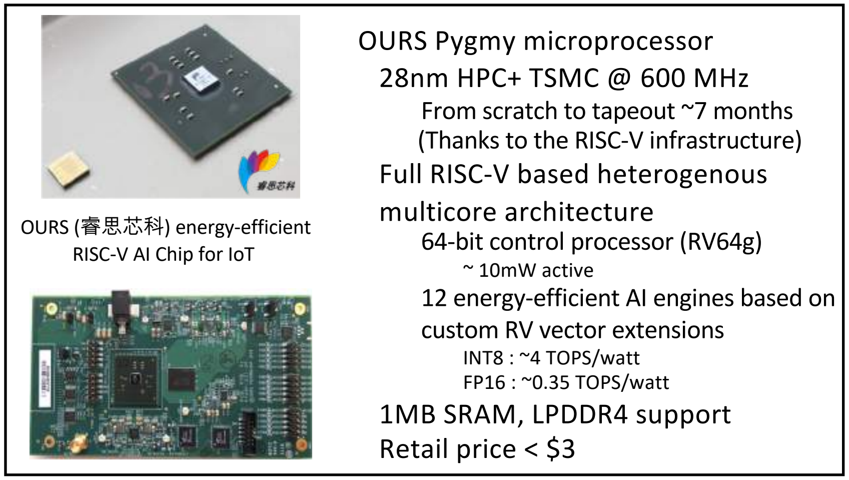
	Designs (“Source”)			Products
Designs Specifications	**Free & Open Designs**	**Licensable Designs**	**Closed Designs**	
Free & Open Spec				
Licensable Spec		$5M + 4%	$25M	Based on Licen or Closed
Closed Spec			🟥	Based on Close Designs



Need Free & Open Specification
To Have Free & Open Designs

		Designs ("Source")			Products
	Designs Specifications	Free & Open Designs	Licensable Designs	Closed Designs	
Specifications	Free & Open Spec	🟩 "Open Source"	🟨	🟥	Based on Free Open Licensed Closed
	Licensable Spec		🟨	🟥	Based on Licen or Closed
	Closed Spec			🟥	Based on Close Designs

53



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 < $3