A New Golden Age for Computer Architecture:
Domain-Specific Hardware/Software Co-Design, Enhanced Security, Open Instruction Sets, and Agile Chip Development

John Hennessy and David Patterson

In the 1980s, Mead and Conway[1] democratized chip design and high-level language programming surpassed assembly language programming, which made instruction set advances viable. Innovations like RISC, superscalar, multilevel caches, and speculation plus compiler advances (especially in register allocation) ushered in a Golden Age of computer architecture, when performance increased annually by 60%. In the later 1990s and 2000s, architectural innovation decreased, so performance came primarily from higher clock rates and larger caches. The ending of Dennard Scaling and Moore's Law also slowed this path; single core performance improved only 3% last year! In addition to poor performance gains of modern microprocessors, Spectre recently demonstrated timing attacks that leak information at high rates[2].

We're on the cusp of another Golden Age that will significantly improve cost, performance, energy, and security. These architecture challenges are even harder given that we've lost the exponentially increasing resources provided by Dennard scaling and Moore's law. We've identified areas that are critical to this new age:

1) Hardware/Software Co-Design for High-Level and Domain-Specific Languages

Advanced programming languages like Python and domain-specific languages like TensorFlow have dramatically improved programmer productivity by increasing software reuse and by raising the level of abstraction. Whereas compiler-architecture co-design delivered gains of about three in the 1980s for C compilers and RISC architectures, new advances could create compilers and *domain-specific architectures*[3] (DSAs) that deliver tenfold or more jumps[4] in this new Golden Age.

2) Enhancing Security

We've made tremendous gains in information technology (IT) in the past 40 years, but if security is a war, we're losing it. Thus far, architects have been asked for little beyond page-level protection and supporting virtual machines. The very definition of computer architecture ignores timing, yet Spectre shows that attacks that can determine timing of operations can leak supposedly protected data. It's time for architects to redefine computer architecture and treat security as a first class citizen to protect data from timing attacks, or at worst reduce information leaks to a trickle.

3) Free and Open Architectures and Open-Source Implementations

Progress on these issues likely will require changes to the instruction set architecture (ISA), which is problematic for proprietary ISAs. For tall challenges like these, we want *all* the best minds to work on them, not only the engineers who work for the ISA owners. Thus, a free and open ISA such as RISC-V can be a boon to researchers[5] because:

- Many people in many organizations can innovate simultaneously using RISC-V.

---

[1] Mead, Carver, and Lynn Conway. *Introduction to VLSI systems*. Addison-Wesley, 1980.

[2] Hill, Mark. "A Primer on the Meltdown & Spectre Hardware Security Design Flaws and their Important Implications," Computer Architecture Today Blog, February 15, 2018, https://www.sigarch.org/ a-primer-on-the-meltdown-spectre-hardware-security-design-flaws-and -their-important-implications/

[3] Hennessy, John L., and David A. Patterson. "Domain Specific Architectures" in *Computer architecture: a quantitative approach*. Sixth Edition, Elsevier, 2018.

[4] Jouppi, Norman P., Cliff Young, Nishant Patil, David Patterson, et al. "In-datacenter performance analysis of a Tensor Processing Unit." In *Proc. 44th Annual International Symposium on Computer Architecture*, pp. 1-12. ACM, 2017.

[5] Ceze, Luis, Mark Hill, Karthikeyan Sankaralingam, and Thomas Wenisch. "Democratizing Design for Future Computing Platforms," June 26, 2017, www.cccblog.org/2017/06/26/democratizing-design-for-future-computing-platforms/

- The ISA is designed for modularity and extensions.
- It comes with a complete software stack, including compilers, operating systems, and debuggers, which are open source and thus also modifiable.
- This modern ISA is designed to work for any application, from cloud-level servers down to mobile and IoT devices.
- RISC-V is driven by a 100-member foundation[6] that ensures its long-term stability and evolution.

Unlike the past, open ISAs are viable because many engineers for a wide range of products are designing SOCs by incorporating IP and because ARM has demonstrated that IP works for ISAs.

An open architecture also enables open-source processor designs for both FPGAs and real chips, so architects can innovate by modifying an existing RISC-V design and its software stack. While FPGAs run at perhaps only 100 MHz, that is fast enough to run trillions of instructions or to be deployed on the Internet to test a security feature against real attacks. Given the plasticity of FPGAs, the RISC-V ecosystem enables experimental investigations of novel features that can be deployed, evaluated, and iterated in days rather than in years. That vision requires more IP than CPUs, such as GPUs, neural network accelerators, DRAM controllers, and PCIe controllers[7]. The stability of process nodes due to the ending of Moore's Law make this goal easier than in the past. This necessity opens a path for architects to have impact by contributing open-source components much as their software colleagues do for databases and operating systems.

4) Agile Chip Development

As the focus of innovation in architecture shifts from the general-purpose CPU to domain-specific and heterogeneous processors, we will need to achieve major breakthroughs in design time and cost (as happened for VLSI in the 1980s). Small teams should be able to design chips, tailored for a specific domain or application. This will require that hardware design become much more efficient, and more like modern software design.

Unlike the "waterfall" development process of giant chips by large companies, Agile development process[8] allows small groups to iterate designs of working but incomplete prototypes for small chips. Fortuitously, the same programming language advances that improved reuse of software have been incorporated in recent hardware design languages, which makes hardware design and reuse easier. While one can stop at layout for a research paper, building real chips is inspiring for everyone in a project, and is the only way to verify important characteristics like timing and energy consumption. The good news is that today TMSC will deliver 100 small test chips in the latest technology for only $30,000[9]. Thus, virtually all projects can afford real chips as final validation of innovation as well as to enjoy the satisfaction of seeing your ideas work in silicon.

We believe the deceleration of performance gains for standard microprocessors, the opportunities in high-level, domain-specific languages and security, the freeing of architects from the chains of proprietary ISAs, and (ironically) the ending of Dennard scaling and Moore's law will lead to another Golden Age for architecture. Aided by an open-source ecosystem, agily developed prototypes will demonstrate advances and thereby accelerate commercial adoption. We envision the same rapid improvement as in the last Golden Age, but this time in cost, energy, and security as well in performance.

What an exciting time to be a computer architect!

---

[6] www.riscv.org.

[7] DARPA, Broad Agency Announcement, "Electronics Resurgence Initiative," September 13, 2017.

[8] Lee, Yunsup, Andrew Waterman, Henry Cook, Brian Zimmer, et al. "An agile approach to building RISC-V microprocessors." *IEEE Micro* 36, no. 2 (2016): 8-20.

[9] Patterson, David and Borivoje Nikolić, "Agile Design for Hardware, Parts I, II, III," *EE Times,* July 27 to August 3, 2015.

Biographical Information

John L. Hennessy

John L. Hennessy was President of Stanford University from 2000 to 2016. He is Director of the Knight-Hennessy Scholars Program at Stanford, a member of the Board of Cisco Systems and the Gordon and Betty Moore Foundation and Chairman of the Board of Alphabet Inc. Hennessy earned his Bachelor's degree in electrical engineering from Villanova University and his Master's and doctoral degrees in computer science from the State University of New York at Stony Brook.

Hennessy's numerous honors include the IEEE Medal of Honor, the ACM-IEEE CS Eckert-Mauchly Award (with Patterson), the IEEE John von Neumann Medal (with Patterson), the Seymour Cray Computer Engineering Award, and the Founders Award from the American Academy of Arts and Sciences. Hennessy is a Fellow of ACM and IEEE, and is a member of the National Academy of Engineering, the National Academy of Sciences and the American Philosophical Society.

David A. Patterson

David A. Patterson is a Distinguished Engineer at Google and serves as Vice Chair of the Board of the RISC-V Foundation, which offers an open free instruction set architecture with the aim to enable a new era of processor innovation through open standard collaboration. Patterson was Professor of Computer Science at UC, Berkeley from 1976 to 2016. He received his Bachelor's, Master's and doctoral degrees in computer science from the University of California, Los Angeles.

Patterson's numerous honors include the IEEE John von Neumann Medal (with Hennessy), the ACM-IEEE CS Eckert-Mauchly Award (with Hennessy), the Richard A. Tapia Award for Scientific Scholarship, Civic Science, and Diversifying Computing, and the ACM Karl V. Karlstrom Outstanding Educator Award. Patterson served as ACM President from 2004 to 2006. He is a Fellow of ACM, AAAS and IEEE, and was elected to the National Academy of Engineering and the National Academy of Sciences.