

Computer Architecture Today

Informing the broad computing community about current activities, advances and future directions in computer architecture.

Open-Source Hardware: Stone Soups and Not Stone Statues, Please

by Hadi Esmaeilzadeh and Michael Bedford Taylor on Dec 18, 2017 | Tags: Hardware, Open Source



Open-source software has become so entrenched in the software development ecosystem that major services (e.g., the Facebook cloud service) and products (e.g., the Android mobile operating system) rely almost exclusively upon open-source components and tools. This movement has reduced the cost, shortened time-to-market, distributed control across many corporations, and above all, opened the floodgates for the explosion of

startups, hastening innovation. Instagram was founded with \$500K and turned into a \$1 Billion acquisition with only 13 employees. If you look at Instagram's software stack, it is shaped like an iceberg, where above the waterline is their proprietary "innovation layer" code and below the waterline, invisible, is the remaining 90% of the code, written by others, in the form of open-source software, for instance Linux, Python, Apache and GNU.

If software is shaped like an iceberg, then hardware is surely shaped like an icy mountain. The innovation layer is at 30,000 feet, and below it is a giant monolith of closed source. In contrast to software, the hardware community, the titan Atlas, holding the entire IT industry and this very software ecosystem on its shoulders, has shrugged the open-source movement. As a result, hardware startups are few in number, and the rate of growth in this domain, excluding AI, is limited. Our innovations as a research community have a narrow pathway to adoption through a few big players that make hardware. Instead of developing artifacts that demonstrate viability of research ideas, we heavily rely on tweaks into simulators. Even in that realm, benchmarking has fallen apart. Many groups are developing their own domain-specific benchmarks since the suites that once made the cornerstone of our community are no longer representative. The dark cloud of diminishing benefits from traditional silicon scaling is looming and these conventional approaches do not seem to be stopping the slow-down of Moore's Law.

The Role of Dark Silicon, and Old Silicon. However, every cloud has a silver lining. The dark silicon phenomenon has caused a shift toward specialization and acceleration, opening the possibilities for rethinking the abstractions. Hopefully, the community will embrace the possibility of being freed from the constraints of one or two IP-encumbered ISAs that dictated the long-held border of hardware and software. Berkeley's RISC-V showed it is possible to come out of the shadows. The era of tweaking the microarchitecture under the thrall of these ISAs seem to be coming to an end. We can do more!!! The possibilities are growing exponentially as even software companies are integrating various forms of reconfigurable hardware in their datacenters, e.g., Microsoft's Catapult, Google's TPU, and Amazon's F1 instances. To fully exploit this unique opportunity, our proposition is to embrace open source hardware. The growing prevalence of FPGAs can be the gateway towards this new realm of development. FPGAs provide a lower-cost stepping stone to custom chip development. Chip manufacturing in older technology nodes, such as detailed in the recent ASPLOS Moonwalk paper on NRE, is another work-around that can significantly reduce the cost.

Platforms versus Prototypes. However, developing hardware prototypes and making it open source on its own will not start the waves of innovation. First, we need to provide a software stack that enables mapping emerging applications on these platforms, ensuring the utility of the open-source hardware. Second, it is essential to make the designs and implementation parametric and customizable, enabling extensibility and continued innovation. For example, we have taken initial steps in this direction by developing TABLA and DnnWeaver, that provide initial open-source hardware acceleration solutions for machine learning and deep neural networks that examine novel designs, compilers, domain-specific languages, and algorithmic hardware/software abstractions.

Recent Open Source Platforms. Rocket chip, and Pulpino provide configurable RISC-V implementations to the world of serial CPUs, while OpenPiton focuses on SPARC-based distributed cache-coherent shared memory manycore designs, and OpenCelerity on RISC-V heterogeneous manycore architectures. MIAOW, GPLGPU, and Nyuzi aim to fill the open-source gap for GPUs. However, the open-source vision extends beyond core design and its software stack. A complete hardware solution also requires analog and digital interfacing logic such as PLLs, high-speed I/O pads, DRAM controllers, ADC/DAC, as well as open source motherboards, socket definitions and packages, such as in BaseJump. We need our colleagues in the circuit design community to also open their innovations. The cost of CAD tools, the analog to gcc and gnu, is another big hurdle and the efforts such as Qflow, Yosys, Chisel, PyMTL, BaseJump STL, myHDL, FreePDK are leading the way.

Making Open Source a Centerpiece of the Academic Enterprise. Academics need to bear the sigil of moving beyond short-term business profit concerns, and break the ground for decades of innovation by making open sourcing hardware a centerpiece of their activities. We not only need to incentivize open sourcing hardware, but also need to make it an essential part of our curriculum. We can create a compelling case for hardware if we give our students experience with all of the open source pieces they need to bring their innovation layer to market. Then, we can retain the talent and the interest of young minds and create the next generation of hardware innovators. We need to come together as a community to support the open source initiatives that are sprouting here and there and make it mainstream.

Make Stone Soups, Not Stone Statues. Credit is the currency of the academic realm. But focusing on personal or institutional branding will lead to the end of any open source project. In order for a project to truly be open source over the long term, it must be treated as a stone soup: you contribute a rough prototype (the stone), and let go by slowly

stepping away as you convince the rest of the community that it is actually everybody's soup that they are contributing to. All code, whether commercial or open-source, software or hardware, requires money to develop and maintain. Although in the short-term, we may have temporary government grants that fund a project, and we can have graduate students hack on the first prototypes, ultimately, the cost of the project needs to be covered as a small side cost to each participant that does not require exclusive attention by large teams of graduate students centralized in one institution. To do this, the community needs to consolidate around a small number of projects (which will mean abandoning your own projects for others!) and the originators of the code need to replace themselves as maintainers and credit-earners within the broader community, so that cost is distributed in a sustainable way. As they say, "In order to love, you must let go."

About the Authors:

Hadi Esmaeilzadeh is an *Associate Professor* at the University of California, San Diego.

Michael Bedford Taylor is an *Associate Professor* at the University of Washington, Seattle.

Share this:





Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Name



Mark D. Hill · 4 months ago

See also a Computing Community Consortium whitepaper we wrote to advocate that the US government catalyze open-source hardware:

Democratizing Design for Future Computing Platforms,
Luis Ceze, Mark D. Hill, Karthikeyan Sankaralingam, Thomas F. Wenisch,
Computing Community Consortium, Jun 26, 2017.

arVIX: <https://arxiv.org/abs/1706....>

^ | ▾ · [Reply](#) · [Share](#) ▾



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Name

EDITOR: ALVIN R. LEBECK

Contribute to Computer Architecture Today

RECENT BLOG POSTS

Why Machine Learning Needs Benchmarks

A Brief Guide of xPU for AI Accelerators

A Common Standard to Fix Our Review Process (and oh, I was wrong about one thing)

The new life of SmartNICs

John Hennessy and David Patterson Share ACM Turing Award

ARCHIVES

May 2018 (1)

April 2018 (7)

March 2018 (6)

February 2018 (4)

January 2018 (7)

December 2017 (4)

November 2017 (4)

October 2017 (6)

September 2017 (6)

August 2017 (2)

July 2017 (4)

June 2017 (6)

May 2017 (6)

April 2017 (4)

March 2017 (2)

TAGS

[Accelerators](#) [ACM SIGARCH](#) [Architecture](#) [Awards](#) [Cache](#) [Conference](#) [CPU](#) [Databases](#)

[Datacenters](#) [Data fusion](#) [Debugging](#) [Discrimination](#) [Distributed Systems](#) [Diversity](#)

[Emerging Technology](#) [Harassment](#) [Hardware](#) [Industry](#) [IoT](#) [ISA](#) [Kalman filters](#) [Machine Learning](#)

[Measurements](#) [Memory](#) [Mobile](#) [Near Data Computing](#) [Networking](#) [Operating Systems](#) [Opinion](#)

[Parallelism](#) [People](#) [Performance](#) [Persistent](#) [Policy](#) [Programmability](#) [Programming](#) [Security](#)

[Shared-Nothing](#) [Smartphone](#) [Specialization](#) [Storage](#) [Systems](#) [Virtualization](#) [Virtual Memory](#)

[Vision](#)

SUBSCRIBE



RSS SUBSCRIBE

Get E-Mail Alerts, enter your Email:

[SUBSCRIBE »](#)

ACM SIGARCH

SIGARCH serves a unique community of computer professionals working on the forefront of computer design in both industry and academia. It is ACM's primary forum to interchange ideas about tomorrow's hardware and its interactions with software.

Join Us

Keep up-to-date with the latest technical developments, network with colleagues outside your workplace and get cutting-edge information, focused resources and unparalleled forums for discussions. [Learn more...](#)

About the Site

This site is maintained by volunteers working in many programs of ACM SIGARCH. We thank you for visiting! If you have questions about the site, please send a note to our [information director](#).

