Open-Source EDA:
If We Build It, Who Will Come?

Andrew B. Kahng

Departments of CSE and ECE, UC San Diego

La Jolla, CA 92093-0404 USA

abk@eng.ucsd.edu   https://vlsicad.ucsd.edu/~abk/

Abstract—The VLSI technology and scaling roadmap has always included Process technology (wrapped as "PDK"), VLSI designs themselves ("System Drivers"), and EDA technology ("Design Technology"). Today, we see an open-source foundry PDK, and we see a vibrant open-source hardware design ecosystem. But what about open-source EDA? The development of open-source EDA technology cannot be separated from the question, "If we build it, who will come?" Today's talk will try to provide some thoughts on this question. What is "it"? Who is "we"? Who is "who"? And in what ways will the "who" come to interact with open-source EDA?

I. INTRODUCTION

"If you build it, they will come" is a slight variation of the famous line that runs throughout "Field of Dreams" (the film adaptation of W. P. Kinsella's novel, *Shoeless Joe* [16]). In that line, "it" is the field where dreams are realized – and where curing the mistakes of the past brings about a brighter future.

Open-source EDA is truly a field of dreams.

As summarized in [13], a *culture* of open-source EDA brings many clear benefits. It enables the scientific method by bringing transparency and reproducibility to VLSI CAD research. By providing reusable "CAD-IP", it improves research efficiency, thus making the field more attractive. And, when available in an end-to-end flow, it enables research to be evaluated in industry-relevant, flow-scale settings.[1] These are dreams that have been with us for decades, whether as the MARCO GSRC Bookshelf of Fundamental CAD Algorithms in the 1990s [6] [23], or the more recent OpenROAD project [1], [2], [36], [39] in the DARPA IDEA program [26].

Recent years have seen greatly increased complexity of IC design in advanced process technologies. The skyrocketing cost, difficulty and risk of design have put silicon implementation out of the reach of system innovators. This crisis of design and innovation has brought renewed attention to the hardware design process itself, notably since 2017 as one of six main thrusts within the U.S. DARPA Electronics Resurgence Initiative [33].

Recent years have also seen tremendous energy put into reducing barriers – toward a "democratization of hardware

[1]Academic end-to-end flows to enable (industry-relevant) research are not necessarily comprised of open-source tools, but are also a well-established goal, e.g., under the leadership of the IEEE CEDA Design Automation Technical Committee (DATC) [7], [8], [11], [12], [27].

design". Today, we see an open-source foundry PDK and design enablement [35] [5], and we see a vibrant open-source hardware ecosystem [32], [34], [45]. But what about open-source EDA? The still-nascent development of open-source EDA technology cannot be separated from the question, "If we build it, who will come?" This paper gives some thoughts on this question, based on the past three years' experience with conception, proposal, and execution of the OpenROAD (Foundations and Realization of Open, Accessible Design) project [29] in the DARPA IDEA program: What is "it"? Who is "we"? Who is "who"? And in what ways will the "who" come to interact with open-source EDA?

II. WHAT IS "IT"

A central goal of the OpenROAD project has always been to deliver "critical mass and critical quality to seed a FOSS EDA ecosystem". This goal has been socialized with a wide range of potential users and stakeholders, at such forums as the DAC-2018 and DAC-2019 "birds of a feather" meetings on Open-Source Academic EDA Software [25], and the 2018 and 2019 workshops on Open-Source EDA Technology (WOSET) [30] – and in numerous public presentations. These discussions, along with the past two years of interactions within the DARPA IDEA program, have made it clear that open-source EDA can mean many things to many people.

Open-source EDA is a moving, many-faceted target.

In July 2018, the community wanted to see a complete RTL-to-GDS flow. In July 2019, with a flow and DRC-clean layout generation in foundry 65nm having been demonstrated, the community wanted to see an integrated tool and tape-out proofs. Today, with an integrated tool and at least one third-party tapeout, community inputs focus on advanced-node foundry support, PPA calibrations, overarching software approach, and a wide range of functionality (DFT, functional simulation, DRC/LVS engine, etc.) that go well beyond the already-ambitious scope of the project.

Even as open-source EDA presents a rapidly moving target, there is also a fundamental tension between OpenROAD's "no human in the loop" goal (no user commands needed, just as a driverless car requires no steering wheel) and many prospective users' typical "here are my 20 favorite Tcl commands that I'd like to see in OpenROAD" request. It has been necessary to continually clarify that while commercial EDA seeks to deliver ultimate quality of results (PPA), OpenROAD seeks to deliver ultimate ease of use. These are two different universes.

The ICCAD-2019 invited paper [13] summarized understanding to date of the "table stakes" and "unblocking milestones" needed to attract users to an open-source EDA tool. These included: (1) a unified tool that achieves a full RTL-to-GDS flow; (2) a shared netlist architecture for the tool that enables tight incremental optimization loops; (3) continuous build and integration within a strong software development methodology; and (4) proper open-source licensing to enable unfettered usage across research and commercial settings.

Currently, (1) and (2) are achieved through OpenROAD's integration onto OpenDB [22] [42], a new open-source physical implementation database that holds all essential data for the physical design creation flow (floorplan, global and detailed placement, CTS, and global and detailed routing) as well as timing and power analyses. The underlying data model of OpenDB is similar to that of the LEF/DEF exchange formats, or the well-known OpenAccess database [44]. The open-sourced OpenSTA [43] is intimately connected with OpenDB, such that both timing graph and physical design information are accessible to tools such as a sizing optimizer. The shared netlist data structure enables in-memory communication between tools and the speed improvements that make tight incremental optimization loops feasible.

Since the release of OpenDB, nearly 20 distinct projects in OpenROAD have been integrated into a single binary, referred to as the OpenROAD *top-level app*. Redundancies and inconsistencies such as multiple LEF/DEF readers and writers, as well as file-based or name-based communication between flow steps, have been eradicated. Instead, all projects utilize OpenDB's data structure, via C++ and Tcl APIs. Figure 1 gives a current view of OpenROAD's flow and "v1.0" tool.
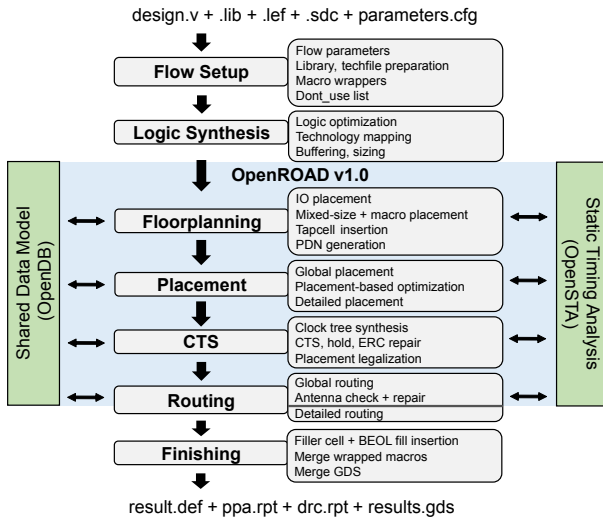

Fig. 1. OpenROAD flow and integrated tool architecture.

III. WHO IS "WE"

OpenROAD was originally proposed to be developed by Ph.D. students and post-docs at four universities. Separately, students and post-docs at a fifth university would serve in an "internal

design advisors" role (see [20]) that was envisioned to span product engineering, expert user testing, and corporate AE-like functions. The clear separation between "internal design advisors" and "tool developers" is built into OpenROAD to avoid improper use of commercial EDA tools. In particular, commercial EDA tools must be used to verify PPA and other calibrations that are required in the project's deliverables; such usage is made by design advisors, not tool developers.

Open-source EDA goes beyond academic research skillsets.

By the project's nine-month mark, it was clear that OpenROAD needed a dedicated, experienced EDA architect and technical manager from outside the existing team. Voluntary budget reallocations to enable hiring of such a technical project lead were initiated from *within* the OpenROAD team. During the project's second year, these reallocations along with non-DARPA gift funding enabled several industry veterans (Tom Spyrou, James Cherry, Matt Liberty) and additional consulting effort to be recruited into OpenROAD. This has brought much-needed technical leadership and know-how – spanning tool delivery and project management, infrastructure (DB, GUI, build/CI), and key engines (STA, RCX) – on an essentially full-time basis. In the context of this section: *"We" must include professional EDA software developers and architects.*[2]

Additional observations regarding the "We" are as follows.

Strong contributors have software skills *and* the right mindset. Several of OpenROAD's strongest tool developers have been undergraduate and graduate students from outside the U.S. The project has maintained connections with active academic research groups who share the vision of open-source EDA in the RTL-to-GDS space. These groups have been able to identify students who have strong software development skills and who are willing to join the project, particularly when this brings a source of support. Over the past two years, such students have taken on key tool development challenges as well as infrastructure tasks (Jenkins CI [24], measuring and improving test coverage, checking and reconciling Tcl naming [38], etc.); this has led to thesis topics as well as publications along the way. In the forthcoming ICCAD-2020 paper [10], a set of authors from Brazil document their experiences as developers for four separate tools within OpenROAD: global routing, clock tree synthesis, IO placement, and tapcell insertion. In general, the combination of software development skills and willingness to be mentored by industry veterans is very powerful. Physical design understanding is easier to grow than software development maturity.

"We" evolves with "It" (which depends on "Who"). The makeup of an open-source tool development team will depend on feature requirements and roadmap, as well as on the size and sophistication of the user population. At this stage of the OpenROAD project, these "We", "It" and "Who" aspects are

---

[2]Meeting aggressive timelines, and functionality and tapeout-capability requirements, is inconsistent with long learning curves, science experiments, or even publications as an end goal. With recent expansions of OpenROAD's scope for 2020-2022, the addition of EDA industry veterans to the team will almost certainly continue.

all very much in flux. This said, it is almost axiomatic that a successful open-source EDA project will see more demands for productization and user support. While success is always welcome, the reality is that it can make project involvement less attractive to academic researchers. This magnifies the need for experienced, full-time architects and developers.

Another reality is that the rigorous software methodology and code organization that improve software quality, maintainability and velocity of development (cf. Google's single-repository approach [19]) can create overheads and barriers – ranging from integration and testing to issues of credit assignment, authorship of publications, etc.[3] An open question is whether "arm's-length development and contribution" might take root in the academic world via such initiatives as the IEEE CEDA DATC Robust Design Flow noted above. For example, when an academic contest is framed in an industry-standard enablement, any winning entry would – if open-sourced by its creators – be available for integration into a more robust, monolithic open-source tool.

"We" must include several types of users. A misconception with open-source EDA is that "users can see the source code, so they can help figure out how to fix bugs or make enhancements." This statement holds for only a very small fraction of OpenROAD's users. In reality, an open-source EDA project such as OpenROAD requires several distinct types of "users" in addition to EDA developers, architects and software engineering infrastructure.

An open-source EDA project requires experienced, *tall-thin tool users* who understand SOC designs as well as their implementation through tapeout. Our "internal design advisors" fit this mold. Such users bring indispensable skills in scripting "make chip" flows, constructing test suites, and bringing up new designs in new enablements. And, experienced EDA tool users have contributed important parts of the OpenROAD flow, notably *pdngen* and the underlying logic of *tapcell*, which are both implemented in OpenDB Tcl.

Importantly, bringup of a new tool *also* requires *application engineers* (AEs), along with expert *power users* who will drive R&D and functional requirements. No successful EDA tool in the history of the field has ever existed without at least a year of intensive "taxicab mode" support delivered by field AEs to key "beta customer" power users. Both power users and AEs find fixes and workarounds and package these up for R&D (developers). Theirs is a much more active, problem-solving mindset compared to mainstream EDA users, who tend to file bugs and wait for fixes.[4] In OpenROAD, an experienced

"power user" and design services consultant was engaged to help accelerate the timeline to a 12nm SOC tapeout milestone. Going forward, it is likely that an additional industry veteran will be recruited to cover the project's current gaps in product engineering and applications engineering.

IV. WHO IS "WHO"

Open-source EDA tools such as Magic, SPICE, MIS/SIS, FastCap and Capo have been in use for decades. Chip tapeouts have been achieved with open-source flows such as qflow [46] and by companies such as efabless [47]. Beyond previous open-source EDA efforts in the RTL-to-GDS space, OpenROAD has an integrated database and timer, Tcl and python scripting interfaces, GUI, and the "feel" of a commercial back-end EDA tool. OpenROAD *additionally* aims for 24-hour no-human-in-the-loop automation that can directly produce a manufacturable layout database in commercial FinFET technology. But who might come to use OpenROAD in the future? Following are some preliminary thoughts.[5]

Open-source EDA is part of a movement.

The open hardware community will use OpenROAD. A vibrant open-source hardware ecosystem sparked by RISC-V has grown rapidly in recent years [32], [34], [45]. This past June, a SKY130 open-source foundry PDK and design enablement was announced by Google and SkyWater Technology Foundry [35]; the presentation [4] was viewed more than 10,000 times in its first week on YouTube. The OpenLANE flow [15] [21] [31] from efabless.com [47] is built on top of OpenROAD and achieved SKY130 tapeout of the "striVe" SOC in May 2020 (see Figure 2).[6] The coming year is likely to see a number of OpenLANE/OpenROAD SKY130 tapeouts made by researchers, makers and small companies. Broadly, open-source EDA can enable product innovators to perform system ideation and design space exploration in a more friction-free manner, with near-zero overhead.
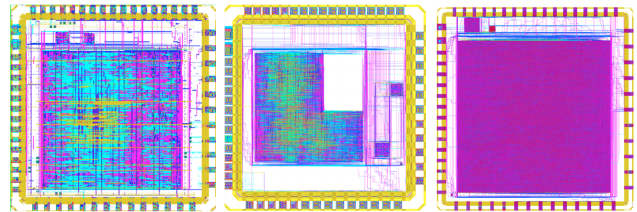

Fig. 2. Left to right: striVe, strive2 and strive3 SOC designs in SKY130 enablements (source: M. Kassem, efabless.com).

---

[3]This may well lead to more forks and fewer pull requests, i.e., a suboptimal level of "Contribute over Create" (Best Practice #1 in the FOSS102 slides of Ansell [3]). Within OpenROAD, students' natural preferences to maintain independent repositories have slowly faded as the benefits of integration and coding best practices, along with the overheads of keeping functionality in synch between 'integrated' and 'standalone' versions, are comprehended. Particularly during the past half year, the number of independent repositories and the number of submodules in the https://github.com/The-OpenROAD-Project/OpenROAD integrated app have markedly decreased.

[4]As noted in [37], as a non-commercial, academic project, it is not even possible for OpenROAD to receive testcases with bug reports as is the norm for commercial EDA tool suppliers.

[5]It is also important to understand who will *not* use OpenROAD for back-end SP&R implementation. Notably, academic design teams have low-cost licenses to leading-edge commercial tools, so have little reason to explore a raw and limited open-source tool. And, IC product organizations at the bleeding edge will never be able to use an open-source tool: only commercial EDA drives its technology to achieve ultimate quality of (PPA) results in the latest nodes. Recall "two different universes" above.

[6]A family of striVe variants has been designed by efabless.com in SKY130. The figure shows (i) striVe, with PicoRv32 and 1kB of synthesized "Logic RAM", in a high-density library; (ii) striVe2 which swaps in an OpenRAM dual-ported SRAM block; and (iii) striVe3 which uses the OSU scl cell library.

Teachers and trainers will use OpenROAD, particularly in contexts where leading-edge back-end implementation tools are unavailable (license limits, tool complexity, etc.). A tool such as OpenROAD is quite simple, yet it gives students insight into back-end database, engine integrations and tight incremental analysis-optimization loops, scripting interfaces, GUI, and other basic aspects of modern P&R tools. For VLSI CAD educators, the transparency of open source enables course assignments that directly delve into tool source code.

Academic EDA researchers will use OpenROAD to improve research efficiency. The reasons for this will only grow stronger over time: bulletproof interfaces, strong test coverage, a user and developer community on GitHub, integration of leading-edge academic methods for easy comparison, etc. Use of OpenROAD as a backplane for academic contests could also increase, for similar reasons.

Mixed-signal SOC designers could use OpenROAD in a "big-A, little-d" context. Digital content on the order of several thousand gates is beyond the reach of manual layout, but a tool such as OpenROAD could suffice to achieve tapeout.

Underserved designers, e.g., in small startups or in government facilities, may be unable to access commercial EDA licenses. Some users may require more complete ownership and control of a transparent chip implementation tool chain, e.g., for reasons of security and trust. Customization opportunities that open source affords may be needed to address rad-hard, 3DIC, trusted IC, or other design applications [14] [48].

V. Looking Back, Looking Ahead

Looking back over the past two years, two high-level takeaways emerge.

First, the passage of time and the accumulation of developed code mean that *some decisions are difficult to revisit*. In other words, "those ships have sailed". OpenROAD's database and timer, Jenkins, coding style, and other project aspects are not likely to change.[7]

Second, OpenROAD's open-source RTL-to-GDS development is challenged by a number of "tensions". (i) Stricter software engineering methodology and tighter integration within a single repository run counter to attracting community participation. (ii) Push-button tapeout capability in advanced-node foundry technology is a requirement that is difficult to "share" with external developers. (iii) Ph.D. research has a mismatch to the demands of EDA tool development and support.[8] More generally, development on an aggressive schedule runs counter to the intermittently available (exams, class projects, winter breaks, summer internships, other research topics, etc.) and

easily-decommitted nature of life in academia. (iv) The IDEA program objective is push-button RTL-to-GDS (a driverless flow that needs no steering wheel), while exploratory use cases and early-adopter users demand much more flexibility and controllability.

These realizations can inform a look ahead, as discussed next.

Looking ahead, two evolutionary directions are of particular interest for open-source EDA.

First, suitable high-value use cases for open-source EDA must be identified. Several are associated with the taxonomy of "Who" above. Beyond these, open-source EDA's low adoption cost and cloud scalability may be well-matched to the long-standing challenge of early design space exploration and pathfinding.

Figure 3 (top) is reproduced from the Design Chapter in the 2009 ITRS Roadmap [41]. The figure's message is that earlier knobs in the flow (system-level, architecture, RTL design) grow relatively more powerful over time. However, while design space exploration should ideally explore more powerful knobs more thoroughly (Ideal "DSE"), attention and iterations still tend to be biased toward the RTL-down flow (Today's "DSE"). This is because optimizing and exploring in early stages has limited value when the back end cannot be predicted accurately enough, and high-level decisions do not correlate to what can actually be closed and signed off. This bespeaks an inability to predict, and an inability to path-find – which are opportunities for leverage of scalable open-source EDA.
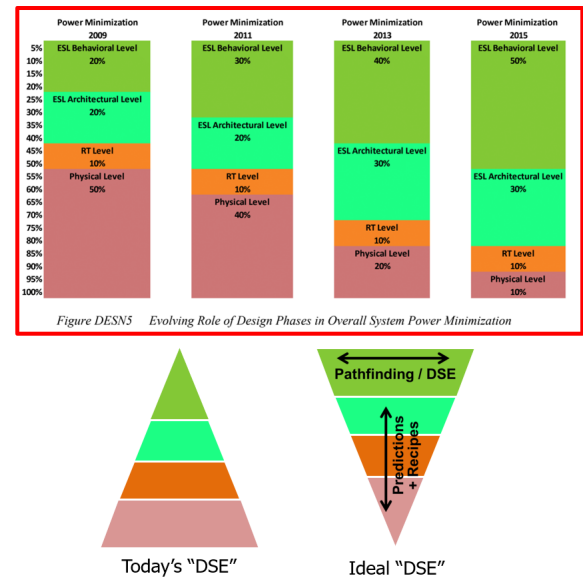

Fig. 3. Top: Growing impact of higher-level design stages on system-level power optimization with advancing technology and system complexity [41]. Bottom: Today's design space exploration (DSE) cannot accurately explore higher-level design stages (left), whereas an ideal DSE would spend effort where it can pay off the most (right).

Second, a sustainable open-source EDA based *ecosystem* must be created that comprehends utility functions and proper

---

[7]Contractual requirements are also constraining. E.g., OpenROAD's requirement to release permissively-licensed open source made the use of OpenAccess [44] impossible. More than a year was spent in pursuit of a database solution; this eventually led to the open-sourcing of Athena Design Systems code and our adoption of OpenDB.

[8]This can be the case even when the Ph.D. student has an EDA R&D career in mind, and despite open-source EDA enabling a new level of EDA and IC design job-readiness. At the same time, all IDEA program performers knew at the outset that "this is not research as usual", and that "the deliverable is working code, not papers".

incentivizations for *all* stakeholders – spanning IC and system companies, smaller design teams whose needs are ill-served today, open-source hardware and software communities, EDA researchers and professional societies, policy-making bodies and consortia, and commercial EDA vendors. For example, improved separations of "research" (driven by students, professors, enthusiasts and design companies) from "productization and support" (driven by EDA professionals and entrepreneurs who leverage open-source EDA technology to serve a bona fide market) will likely be welcome on multiple fronts.

In his recent 2020 DARPA ERI Summit plenary talk [17], the IDEA program manager, Mr. Serge Leef, posed the question: **Can DARPA *improve access* [to state-of-the-art EDA tools] and *fuel advances* through open-source EDA technologies?** Here, "improve access" was framed as a matter of *economics*, i.e., open-source tools together with cloud deployment can significantly reduce design non-recurring engineering (NRE) costs. And, "fuel advances" was framed as a matter of unleashing hardware *innovation*, i.e., open-source tools can lower barriers and accelerate advances in both hardware design and EDA technology. The overarching challenge is to transition open-source EDA technology from research lab to commercial impact – in a scalable, sustainable way.

Figure 4 presents key elements of three essential pillars that a commercial entity (the Transition "Operator" in the figure) must provide to achieve a scalable, sustainable business based on open-source EDA technology. These pillars – Productization, Support, and Business – exactly correspond to the gaps exposed during OpenROAD's open-source tool development in an academic environment.
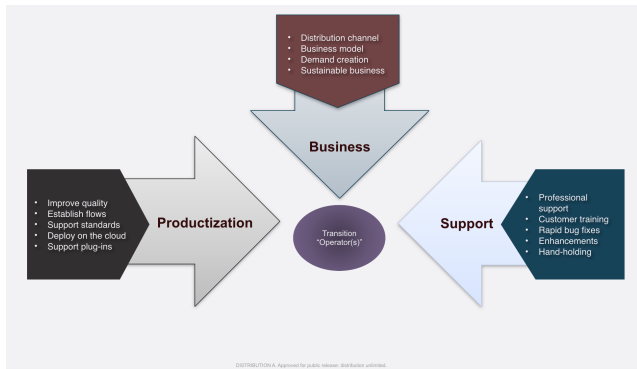
Fig. 4. A potentially scalable and sustainable framework for delivering open-source EDA technology to an IC designer market. (Source: Mr. Serge Leef, DARPA. August 2020.)

Last, if we view open-source EDA as a potential disruptive technology, then the canonical trajectory of adoption is as shown in Figure 5 [9] [28]. A disruptive technology will initially penetrate least-demanding and/or most-underserved market segments. These could correspond to the "Who" in Section IV above.
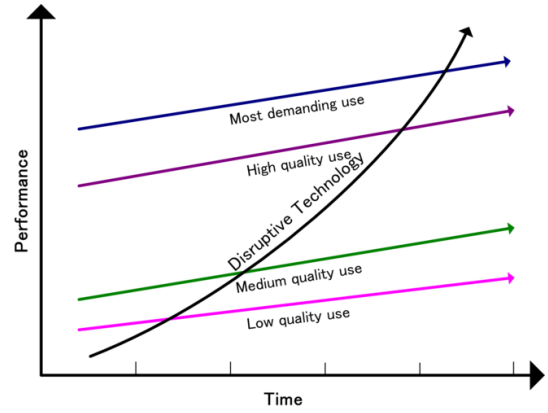
Fig. 5. The trajectory of disruptive innovation.

VI. Conclusion

A year ago, [13] considered open-source EDA as a *mirror* that enabled reflections on history, culture, and futures for the research community and the industry. Here, the question of "If we build it, who will come?" also leads to reflections and realizations.

Open-source EDA is truly a field of dreams.

Open-source EDA is a moving, many-faceted target.

Open-source EDA goes beyond academic research skillsets.

Open-source EDA is part of a movement.

The OpenROAD project is still absorbed by many challenges that surround the goal of "critical mass and critical quality to seed a FOSS EDA ecosystem". These challenges include software development infrastructure and robustness; growing available development resources; delivering tool enhancements to satisfy users; adding key missing functionality; improving quality of results; growing a community of users and contributors; and pursuing fundamental EDA research objectives while also helping to accelerate real-world innovation in silicon. As the project continues to execute toward its deliveries, we look forward to the conversations that will illuminate new, sustainable and scalable pathways for the transition of open-source EDA research into real-world impact. And, as developers of open-source EDA, we look forward to continued evolutions: the "It" that OpenROAD project members and many others will build; the community of "We" who contribute; and the "Who" who drive and use open-source EDA to ultimately unleash hardware innovation.

Open-source EDA is a journey.

References

[1] T. Ajayi, D. Blaauw, T.-B. Chan, C.-K. Cheng, V. A. Chhabria, D. K. Choo, M. Coltella, S. Dobre, R. Dreslinski, M. Fogaça, S. Hashemi, A. Hosny, A. B. Kahng, M. Kim, J. Li, Z. Liang, U. Mallappa, P. Penzes, G. Pradipta, S. Reda, A. Rovinski, K. Samadi, S. S. Sapatnekar, L. Saul, C. Sechen, V. Srinivas, W. Swartz, D. Sylvester, D. Urquhart, L. Wang, M. Woo and B. Xu, "OpenROAD: Toward a Self-Driving, Open-Source Digital Layout Implementation Tool Chain", *Proc. GOMACTech*, 2019, pp. 1105-1110.

[2] T. Ajayi, V. A. Chhabria, M. Fogaça, S. Hashemi, A. Hosny, A. B. Kahng, M. Kim, J. Lee, U. Mallappa, M. Neseem, G. Pradipta, S. Reda, M. Saligane, S. S. Sapatnekar, C. Sechen, M. Shalan, W. Swartz, L. Wang, Z. Wang, M. Woo and B. Xu, "Toward an Open-Source Digital Flow: First Learnings from the OpenROAD Project", *Proc. DAC*, 2019, pp. 76:1-76:4.

[3] T. Ansell, "FOSS 101" and "FOSS 102" presentations, July 2019. j.mp/eri19-foss101 and j.mp/eri19-foss102.

[4] T. Ansell, "Fully open source manufacturable PDK for a 130nm process", *FOSSi Dial-Up presentation*, June 30, 2020. https://youtu.be/EczW2IWdnOM.

[5] T. Ansell and M. Saligane, "The Missing Pieces of Open Design Enablement: A Recent History of Google Efforts", to appear in *Proc. ICCAD*, 2020.

[6] A. E. Caldwell, A. B. Kahng and I. L. Markov, "Toward CAD-IP Reuse: The MARCO GSRC Bookshelf of Fundamental CAD Algorithms", *IEEE Design and Test of Computers*, 2002, pp. 70-79.

[7] J. Chen, I. H.-R. Jiang, J. Jung, A. B. Kahng, V. N. Kravets, Y.-L. Li, S.-T. Lin and M. Woo, "DATC RDF-2019: Towards a Complete Academic Reference Design Flow", *Proc. ICCAD*, 2019, pp. 1-6.

[8] J. Chen, I. H.-R. Jiang, J. Jung, A. B. Kahng, V. N. Kravets, Y.-L. Li, S.-T. Lin and M. Woo, "DATC RDF-2020: Strengthening the Foundation for Academic Research in IC Physical Design", to appear in *Proc. ICCAD*, 2020.

[9] C. Christensen, *The Innovator's Dilemma*, Harvard Business Review Press, 1997.

[10] M. Fogaça, E. Monteiro, M. Danigno, I. Oliveira, P. Butzen and R. Reis, "Contributions to OpenROAD from Abroad: Experiences and Learnings", to appear in *Proc. ICCAD*, 2020.

[11] J. Jung and I. H.-R. Jiang and J. Chen and S.-T. Lin and Y.-L. Li and V. N. Kravets and G.-J. Nam, "DATC RDF: An Academic Flow from Logic Synthesis to Detailed Routing", *Proc. ICCAD*, 2018, pp. 37:1-37:4.

[12] J. Jung, P.-Y. Lee, Y.-S. Wu, N. K. Darav, I. H.-R. Jiang, V. N. Kravets, L. Behjat, Y.-L. Li and G.-J. Nam, "DATC RDF: Robust design flow database", *Proc. ICCAD*, 2017, pp. 872-873.

[13] A. B. Kahng, "Looking Into the Mirror of Open Source: Invited Paper", *Proc. ICCAD*, 2019.

[14] A. B. Kahng and F. Koushanfar, "Evolving EDA Beyond its E-Roots: An Overview", *Proc. ICCAD*, 2015, pp. 247-254.

[15] M. Kassem, T. Edwards and M. Shalan, "Building OpenLane: A 130nm OpenROAD-based Tapeout-Proven Flow". to appear in *Proc. ICCAD*, 2020.

[16] W. P. Kinsella, *Shoeless Joe*, Houghton Mifflin, 1982.

[17] S. Leef, "Open Source Accelerated Chip Design", plenary talk, *DARPA ERI Summit*, August 18, 2020.

[18] G. Moore, *Crossing the Chasm*, Harper Business Essentials, 1991.

[19] R. Potvin and J. Levenberg, "Why Google Stores Billions of Lines of Code in a Single Repository", *Comm. of the ACM*, 2016, pp. 78-87.

[20] A. Rovinski, T. Ajayi, M. Kim, G. Wang and M. Saligane, "Bridging Academic Open-Source EDA to Real-World Usability" *Proc. ICCAD*, 2020.

[21] M. Shalan, "OpenLane, A Digital ASIC Flow for SkyWater 130nm Open PDK", *FOSSi Dial-Up presentation*, July 28, 2020. https://www.youtube.com/watch?v=Vhyv0eq_mLU.

[22] T. Spyrou, "Open-Source EDA Challenges and Architecture", opening talk at DAC2019 Open-Source Academic EDA Software Birds-of-a-Feather meeting.

[23] (MARCO GSRC) VLSI CAD Bookshelf. http://vlsicad.eecs.umich.edu/BK/

[24] Jenkins website, https://jenkins.io/

[25] DAC 2019 Birds-of-a-Feather Meeting: Open-Source Academic EDA Software Continued, https://github.com/The-OpenROAD-Project/Birds-of-a-Feather-Open-Source-Academic-EDA-Software/wiki/DAC-2019-Birds-of-a-Feather:-Open-Source-Academic-EDA-Software

[26] DARPA IDEA, https://www.darpa.mil/program/intelligent-design-of-electronic-assets

[27] DATC, https://ieee-ceda.org/node/2591 and https://github.com/ieee-ceda-datc/RDF2019

[28] Wikipedia, "Disruptive Innovation", https://en.wikipedia.org/wiki/Disruptive_innovation

[29] OpenROAD, https://theopenroadproject.org/

[30] Workshop on Open-Source EDA Technology (WOSET), http://scale.engin.brown.edu/woset/

[31] efabless OpenLANE repository, https://github.com/efabless/openlane.

[32] Chips Alliance, https://chipsalliance.org/.

[33] DARPA Electronics Resurgence Initiative, https://www.darpa.mil/work-with-us/electronics-resurgence-initiative

[34] The Free and Open Source Silicon Foundation, https://fossi-foundation.org/.

[35] Google-Skywater, https://github.com/google/skywater-pdk.

[36] OpenROAD: Foundations and Realization of Open and Accessible Design, https://theopenroadproject.org

[37] "OpenROAD Flow Initial Information for Users", November 2018, https://theopenroadproject.org/openroad_event/openroad-flow-initial-information-for-users/

[38] "OpenROAD 'Safe Names' Conventions, v1.0", December 2019, https://theopenroadproject.org/openroad_event/new-openroad-safe-names-conventions-v1-0/

[39] OpenROAD GitHub Repository, https://github.com/The-OpenROAD-Project/OpenROAD.

[40] "International Roadmap for Devices and Systems 2018 Update More Moore", https://irds.ieee.org/images/files/pdf/2018/2018IRDS_MM.pdf

[41] "International Technology Roadmap for Semiconductors 2009 Design", https://www.dropbox.com/sh/ia1jkem3v708hx1/AAB1fo1HrYIKClJNk0dB7YrCa?dl=0&preview=Design.pdf

[42] OpenDB, https://github.com/The-OpenROAD-Project/OpenDB

[43] OpenSTA, https://github.com/The-OpenROAD-Project/OpenSTA

[44] OpenAccess, https://si2.org/openaccess/

[45] OpenHW Group, https://www.openhwgroup.org/

[46] Qflow. https://github.com/RTimothyEdwards/qflow

[47] Efabless.com efabless.com

[48] IEEE CEDA Design Automation Futures Workshop, October 2016. https://ieee-ceda.org/event/design-automation-futures-workshop-2016-dafw