Low-Power Design with Open-Source Hardware: Opportunities and Challenges

Vaibhav Verma, Xinfei Guo and Mircea R. Stan
Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22904
{vv8dn, xg2dt, mircea}@virginia.edu

***Abstract*—Reducing system power consumption has always been a primary design goal and the driving force behind a variety of products ranging from server-class chips to battery-operated IoT devices. Recently, the advent of open-source hardware (OSH) promises great portability, flexibility and customizability. Meanwhile, OSH also introduces new design challenges. We argue that with increasing demands from application space, the interaction between low-power design with open-source hardware offers a bright future in terms of development cost and enables broader collaborative efforts. In this paper, we will overview the opportunities and challenges in low power design with various levels of open-source hardware.**

***Index Terms*—Low-power, Open-source Hardware, Scaling, EDA, IoT**

I. Introduction

Design for low power or power-aware design are not new topics and innovations in this field have been achieved through decades of joint efforts by industry and academia. Even though advanced process technology such as FinFETs incrementally improve energy efficiency and leakage power, pushing power consumption further down [1] but power density in modern designs has been increasing at a much higher pace. This demands aggressive voltage scaling to meet thermal and power budgets but scaled supplies lead to dramatic loss of dynamic range in circuits complicating the low-power design process [2]. A major trend in application space is the emergence of connected edge devices implemented as System-on-Chip (SoC). These devices are used as sensors and monitors deployed in smart vehicles, implantable medical equipment, smart homes and so on. Such devices face stringent power constraints because they are powered by batteries or energy harvesters. Delivering such devices at volume requires innovative low-power design approaches to meet both the technology and application challenges.

General-purpose computing platforms are proving sub-optimal for processing emerging applications like big data and artificial intelligence. Open-source hardware (OSH) emerges as a perfect solution in this scenario to accelerate low-power design. Because OSH is flexible and freely available to the public, it democratizes access to custom silicon to solve these issues. Design approaches that involve OSH have the potential to shorten time to market and lead to low-cost customized silicon development [3]. High-level language driven design methodologies (e.g. High level synthesis) can make OSH even more portable and accessible [4], and open it to the "maker community". As pointed out in [5], open-source hardware is still underused compared to open-source software because of the complexity involved in the design process. However, we believe that with the increasing number of devices that are going to be deployed within a short time and the demands of developing domain-specific hardware [6], OSH will become an attractive hardware design approach.

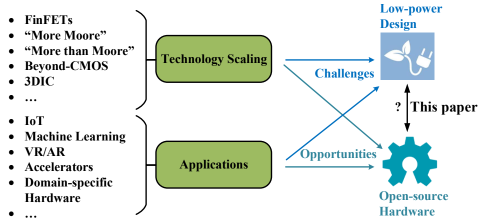

Fig. 1. Interaction between low-power design and open-source hardware (OSH): Both technology scaling and increasing application space push low power design and introduce new challenges but they also open the door for OSH development.

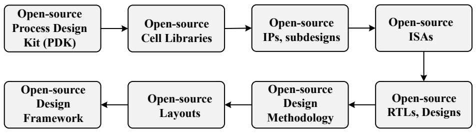
Open-source Process Design Kit (PDK)
Open-source Cell Libraries
Open-source IPs, subdesigns
Open-source ISAs
Open-source RTLs, Designs
Open-source Design Methodology
Open-source Layouts
Open-source Design Framework

Fig. 2. Hierarchies of Openness in Chip Design Process (roughly following the order of a typical IC design flow)

A promising direction for open-source hardware design will be in the field of low-power design. Through OSH's low cost of entry, designers will be able to exercise radical design techniques which would otherwise be considered too risky. With openly available hardware IP for part of the design process, more efforts can then be spent on exploring low-power design solutions. This paper aims to explore answers to following questions. 1) How existing OSH can help low-power design? 2) What are the potential challenges and limitations for low-power OSH? 3) How to move forward from current OSH landscape? To address these questions, we overview the current state-of-art in OSH and foresee how they can help with low power design. When these two domains interact, as shown in Figure 1, there are potential challenges to be addressed. We also pose open questions and potential challenges for OSH and low power community to tackle in the future.

II. Overview of Open-source Hardware ecosystem

Chip design process involves many sequential steps and interacts with many EDA tools which require numerous inputs. In this section, we briefly overview different levels of OSH following the order shown in Figure 2.

- **Open-source PDKs and Cell Libraries**: Over years, continuous efforts have been made to develop open-source PDKs that are independent of specific foundries but are representative enough for research and educational purposes. Such efforts were represented by FreePDK developed by academia that covers technology nodes such as 45nm bulk [7] and 15nm FinFET [8], and ASAP design kit that is developed for 7nm FinFET [9]. Associated open-source cell libraries are also developed to enable researchers to design complicated circuits with advanced technologies such as FinFETs.

- **Open-source IPs and Subdesigns**: Hierarchical design is a popular approach where design is divided into small modules or sub-designs depending on the complexity. Such a module, also called IP or subdesign, can be a small functional block, memory array or a custom design. OpenCore.org [10] has emerged as one of the largest platforms for sharing open-source IPs, sub-blocks such as digital filters, adders and so on. With the recent push for specialized architectures and acceleration hardware, the efforts of developing open-source subdesigns (such as process elements for neural network) have been increasing continuously.
- **Open-source ISAs**: Most of the modern ISAs, such as the ones developed by Intel, IBM or ARM, are proprietary and cannot be used by public without necessary licenses that cost time and money. Open-source ISAs such as openSPARC and RISC-V [11] are gaining interest from both academia and industry [12]. Open-source ISA allows independent researchers and developers to implement their own processors or modify open source implementations. Cost of a processor can be reduced significantly by avoiding licensing fees.
- **Open-source RTLs and Full Designs**: Open-source RTLs at a core level are sometimes referred as "soft core". The key idea is to release a whole functional design to public which can be pushed through the design flow directly. The OSH at the RTL level usually supports extension and customization, and these soft cores can serve as a perfect platform for researchers to exercise their research ideas. Some examples of widely used (but not limited to) Open-source RTL designs include openSPARC or designs on OpenCore.org [10] such as WISHBONE Bus or openMSP430. Many mature open-source RTL designs also provide necessary scripts or verification flow to simulate and validate the design with standard design methodology.
- **Open-source Design Methodology and EDA Tools**: One of the main obstacles of open-source development has been the lack of reliable design tools. Each EDA tool usually requires very high license fees. Open-source EDA tools might not be the ultimate option for silicon implementations, but we believe they can be very useful for the purpose of functional validation, estimation of metrics, and can be extremely helpful for educators or students who don't have access to expensive EDA tools. Huge progress has been made in this area in recent years. Massive online courses focused on educating future IC designers are available which use open-source design flow called VSDFLOW [13] and Qflow [14] that featured a set of open-source EDA tools, such as Yosys [15] for synthesize, Placement tools graywolf [16], Qrouter [17] for Routing, Opentimer [18] for timing analysis, Magic [19] for layout, and NGSpice [20] or eSIM [21] for circuit design, analysis and simulation. Open-source memory compiler OpenRAM proposed in [22] can be used to assist the array generation, characterization and verification of memory designs across multiple technology nodes and memory configurations. Ongoing efforts are trying to further optimize and validate the tools against industry standards for actual silicon implementation purposes.
- **Open-source Layouts**: Since layout is directly constrained by design rules specified for certain technology node, it is hard to transfer between technologies. Thus, it is less economic to make efforts for releasing the layout directly at a core level unless it is designed with open-source PDKs. Most efforts at this level of openness are limited to open-source cell library layouts or subdesign layouts.
- **Open-source Design Framework**: Design framework refers to a design environment or platform that allows design space explorations, prototyping or designing special-purpose hardware. Arduino [23] might be one of the earliest such open-source platforms for prototyping micro-controllers. They also provide necessary software libraries and tools. Many industry players have released their platforms to support wide range of hardware development efforts. For example, NVDLA [24] developed by NVidia provides a free and open architecture that promotes a standard way to design deep learning inference accelerators. Trillium [25] developed by Arm is machine Learning (ML) platform that enables advanced, efficient inference at the edge. Intel Edison Development Platform [26] provides rapid prototyping of IoT and wearable computing products. SiFive [27] also releases development boards for RISC-V based semi-custom chips. Facebook launched the The Open Compute Project (OCP) [28] to enable open-source high-performance computing. Most of these open-source design frameworks support scalable and highly configurable architectures that can be adapted to various applications ranging from IoT devices to high performance computing machines. Open-source can enable such massive collaborative efforts of industry, academia and individual contributors.

III. Low-power design meets Open-source Hardware

Open-source development of hardware offers numerous benefits that ease the development of specialized electronic circuits in general. Widely collaborative nature of open-source development can reduce the complexity of designing involved circuits and systems like mixers, PLL's and SRAM's etc. OSH can help to accelerate the pace of technological advancement by reducing the time spent on individual circuits and rather focusing on using a repository of proven and optimized components to design bigger systems aimed at making human lives

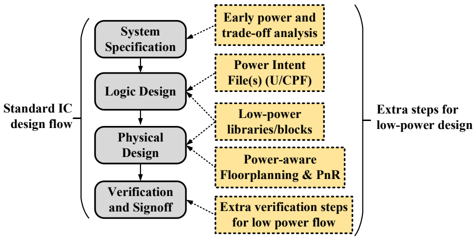
Standard IC design flow
System Specification
Logic Design
Physical Design
Verification and Signoff
Early power and trade-off analysis
Power Intent File(s) (U/CPF)
Low-power libraries/blocks
Power-aware Floorplanning & PnR
Extra verification steps for low power flow
Extra steps for low-power design

Fig. 3. An abstracted view of a typical IC design flow and extra steps required for supporting low power features

easier. Low-power hardware design is one such area where we can take full advantages offered by OSH. As shown in Figure 3, the design of low-power SoC usually requires extra effort compared to standard digital systems in terms of defining the power intent of the design and developing various power modules like power gates, level-shifters, isolation cells etc. On top of that, power-up and power-down sequence and strategies have to be defined and verified for different power states. All these can be done by either using standard languages like UPF or CPF, supported by leading synthesis tools, along with logic RTL design or by designing custom circuits to push the limits of voltage/frequency scaling. These tasks add extra complexity to the design cycle. Complexity leads to additional development time for schematics, physical design and verification of power-aware circuits. This extra time and complexity also translates into higher costs in terms of Non-recurring engineering (NRE) costs for specialized designers for low-power design.

Open-source development can help to counter these issues. Complexity and design time can be significantly reduced by using silicon-proven circuits and techniques from the open-source community. A major portion of design time is spent in conceptualizing, simulating and verifying complex circuits. Hence, already tested designs can serve as first point in design cycle rather than starting from scratch and thus speeding up the whole cycle. Multiple designers across the globe can work together to solve major issues in power-hungry blocks and suggest improvements to achieve the most efficient designs in the fastest time possible. The efficient designs can then be packaged as hardened IPs which can be used by other similar projects to further reduce design complexity and time. Thus, designers can build on top of each other's designs rather than reinventing the wheel all over again. This can also save tremendous design cost by reducing both the NRE design time cost and by decreasing the probability of silicon failure as a result of tested IP's.

IV. Open-source Challenges in Low-power Design

Opportunities always come with challenges. Figure 4 summarizes extra steps that are necessary for supporting low power features at each level of open-source hardware.

- **PDK/IP level** - Low power techniques such as multi-V_{th} require multiple versions of transistor models. Widely-used techniques such as power-gating, clock-gating and voltage islands will require special cells like header/footer cells, clock-gating cells, retention registers and isolation cells to be present in open-source standard cell libraries. Blocks such as SRAM usually need additional custom effort to support the low power features and this brings additional challenges to open-source IP development.

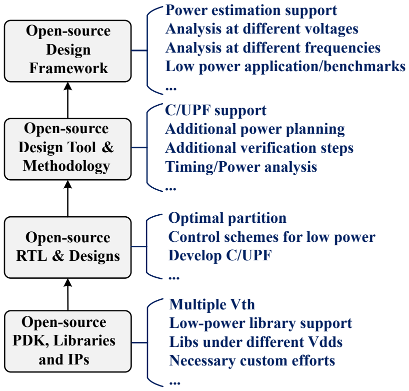
Open-source Design Framework
Power estimation support
Analysis at different voltages
Analysis at different frequencies
Low power application/benchmarks
...
Open-source Design Tool & Methodology
C/UPF support
Additional power planning
Additional verification steps
Timing/Power analysis
...
Open-source RTL & Designs
Optimal partition
Control schemes for low power
Develop C/UPF
...
Open-source PDK, Libraries and IPs
Multiple Vth
Low-power library support
Libs under different Vdds
Necessary custom efforts
...

Fig. 4. Summary of extra efforts that are necessary for supporting low power design at each level of open-source hardware.

- **RTL level** - Designing a big system like a processor or accelerator at the RTL level can be a daunting and time-consuming task. Making it low power can be even more difficult as many features need to be incorporated in RTL as well as the design process. Examples are control schemes for power gating, clock gating, the optimal partition for leakage reduction and so on. Developing open-source RTLs to support low power thus require additional iterations with support for UPF/CPF.
- **Tool and EDA level** - Open-source synthesis and place and route EDA tools should be able to support low power implementations with UPF/CPF and the core algorithm should also take power optimization into account. Power planning tools need to support additional power planning effort and verification tools should recognize all the power features. Timing analysis and power analysis tools need to identify the power hungry blocks and guide designers to optimize power at various design stages.
- **Design Framework level** - Power needs to be accurately modeled in these tools for predicting a reasonable budget. During design space exploration, besides the voltage and clock frequency, low power features need to be incorporated as key design knobs. For example, system level scheduler is necessary for managing the power gating or clock gating. Accordingly, low-power specific benchmarks that are developed from applications are required to evaluate the potential designs within these open-source frameworks.

Despite all these challenges, we believe that hope is not lost towards a brighter OSH future. The first step in solving these issues is to identify them and then direct dedicated efforts towards solving them. Low-power design community is slowly coming to the realization about the importance of open-sourcing hardware designs to accelerate growth and innovation in a rather sluggish industry. Efforts like RISC-V [11] and Arduino [23] are gaining widespread interests in low power applications such as IoT and embedded systems. PULP [29] is another open-source low-power processing project with multiple silicon proven open systems. Low-power design with open-source hardware can result in a good balance of design time, cost and design goals.

V. Future Low-power Open-source Hardware

It is clear that open-source hardware development in the low power domain is still at a young age. But it has been gaining speed in recent years and huge progress in this area has been made. In this section, we envision how community can work together to tackle above discussed challenges moving forward.

- **Maintaining a healthy low-power open-source hardware ecosystem**. As it is well known that hardware development, by nature, incurs multiple entities and steps that range from manufacturing and design to deployment. It is very challenging to maintain a healthy ecosystem similar to what open-source software has done [30]. Thus, clear licenses that account for hardware design are necessary. It requires the community to work together to come up with clear definitions of each term and emphasize their importance.
- **Developing a uniform framework to glue OSH and Non-OSH**. Although there have been increasing efforts of developing open-source tools for hardware design, it is still far from being widely used for building reliable products. Hence, proprietary IPs play key roles in optimizing the hardware performance and power. We believe that mixing OSH and non-OSH can be a promising approach in near future. Our design experience shows that combining some levels of open-source hardware with necessary proprietary tools, PDKs and libraries can shorten the design cycle and reduce the design effort. All of these require good interfaces and validated flows to integrate OSH with conventional Non-OSH approach. Examples of such interfaces can be well-documented scripts, tool descriptions and validated design experiences.
- **Developing reusable, customizable and modular open-source hardware**. We envision that domain-specific open-source hardware will become more attractive in future. As end market becomes more diverse, customizable hardware will keep increasing, thus open hardware that can support it will be promising [3].
- **Enabling open-source SW/HW co-design**. Hardware and software co-design has become necessary in today's world and OSH should adopt best practices of open-source software (OSS). High-level descriptions of hardware can help to bring both sides together. For example, high-level synthesis (HLS) introduced by FPGA community is becoming integral. FPGAs allow to mimic software development process in hardware design [31]. Thus, OSH and OSS developers should take advantage of these platforms to develop more sustainable open-source environment and solve emerging power issues together.

VI. Conclusion

As low-power design becomes more complex, we believe that open-source hardware can cut down both design cost and time dramatically. It will empower low-power design by giving freedom to experiment and push boundaries for extreme power efficient designs. Many challenges in open-source hardware are being solved and open-source tools and components are being updated and validated in silicon regularly. We envision that this design style will catch wind and soon become the standard hardware design method where designers openly contribute latest circuits, PDKs, methodologies and other hardware components to publicly available repositories with appropriate licenses. This will help low-power designers to focus more on rapid innovation and solving challenges just like software rather than redesigning the same circuits which have already been there earlier.

Acknowledgment

This work was funded in part by DARPA under the PERFECT program and by CRISP, one of six SRC JUMP Centers. The views, opinions and/or findings expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. government.

References

[1] X. Guo *et al.*, "Back to the Future: Digital Circuit Design in the FinFET Era," *JLPEA*, vol. 13, no. 3, pp. 338–355, 2017.
[2] A. Fahim, "Challenges in low-power analog circuit design for sub-28nm CMOS technologies," in *Low Power Electronics and Design (ISLPED), 2014 IEEE/ACM International Symposium on*, pp. 123–126, IEEE, 2014.
[3] Y. Lee *et al.*, "An agile approach to building RISC-V microprocessors," *IEEE Micro*, vol. 36, no. 2, pp. 8–20, 2016.
[4] Y. Lee, "RISC-V Rocket Chip SoC Generator in Chisel," in *Online slides: https://riscv. org/wp-content/uploads/2015/01/riscv-rocket-chip-generator-workshop-jan2015. pdf*, 2015.
[5] G. Gupta *et al.*, "Kickstarting Semiconductor Innovation with Open Source Hardware," *Computer*, vol. 50, no. 6, pp. 50–59, 2017.
[6] D. Patterson, "50 Years of computer architecture: From the mainframe CPU to the domain-specific tpu and the open RISC-V instruction set," in *ISSCC*, pp. 27–31, IEEE, 2018.
[7] J. E. Stine *et al.*, "FreePDK: An open-source variation-aware design kit," in *Microelectronic Systems Education, 2007. MSE'07. IEEE International Conference on*, pp. 173–174, IEEE, 2007.
[8] K. Bhanushali and W. R. Davis, "FreePDK15: an open-source predictive process design kit for 15nm FinFET technology," in *Proceedings of the 2015 Symposium on International Symposium on Physical Design*, pp. 165–170, ACM, 2015.
[9] L. T. Clark *et al.*, "ASAP7: A 7-nm finFET predictive process design kit," *Microelectronics Journal*, vol. 53, pp. 105–115, 2016.
[10] "OpenCores." https://opencores.org/.
[11] "RISC-V." https://riscv.org/.
[12] D. Patterson, "Reduced Instruction Set Computers Then and Now," *Computer*, vol. 50, no. 12, pp. 10–12, 2017.
[13] K. P. Ghosh and A. K. Ghosh, "Technology mediated tutorial on RISC-V CPU core implementation and sign-off using revolutionary EDA management system (EMS)VSDFLOW," in *CSTIC, 2018 China*, pp. 1–3, IEEE, 2018.
[14] "Qflow." http://opencircuitdesign.com/qflow/.
[15] C. Wolf *et al.*, "Yosys-a free Verilog synthesis suite," in *Proceedings of the 21st Austrian Workshop on Microelectronics (Austrochip)*, 2013.
[16] "graywolf." https://github.com/rubund/graywolf.
[17] "qrouter." http://opencircuitdesign.com/qrouter/.
[18] T.-W. Huang and M. D. Wong, "Opentimer: A high-performance timing analysis tool," in *ICCAD*, pp. 895–902, IEEE Press, 2015.
[19] J. K. Ousterhout *et al.*, "The magic VLSI layout system," *IEEE Design & Test of Computers*, vol. 2, no. 1, pp. 19–30, 1985.
[20] "Ngspice." http://ngspice.sourceforge.net/.
[21] "esim." https://esim.fossee.in/.
[22] M. R. Guthaus *et al.*, "OpenRAM: An open-source memory compiler," in *Proceedings of the 35th International Conference on Computer-Aided Design*, p. 93, ACM, 2016.
[23] "Arduino project home." https://www.arduino.cc/.
[24] "Nvdla." http://nvdla.org/.
[25] "Project Trillium - Arms Machine Learning (ML) platform." https://developer.arm.com/products/processors/machine-learning.
[26] "The Intel Edison Development Platform." https://software.intel.com/en-us/iot/hardware/discontinued.
[27] "SiFive." https://www.sifive.com/.
[28] "The Open Compute Project (OCP)." https://www.opencompute.org/.
[29] "PULP." https://www.pulp-platform.org/.
[30] G. Gupta *et al.*, "Open-source hardware: Opportunities and challenges," *arXiv preprint arXiv:1606.01980*, 2016.
[31] E. Rubow, "Open source hardware." http://cseweb.ucsd.edu/classes/fa08/cse237a/topicresearch/erubow_tr_report.pdf, 2008. UCSD Working Paper.