EDITOR **DIRK RIEHLE**
Friedrich Alexander-University of Erlangen Nürnberg;
dirk.riehle@fau.de

Open Source Hardware

Frank Hannig and Jürgen Teich, Friedrich-Alexander University Erlangen-Nürnberg

Hardware that can be manufactured from free and open source descriptions has gained a lot of momentum. This article gives a general introduction, focusing on electronics and integrated circuits, corresponding open ecosystems and organizations, and highlights benefits and challenges.

Open source hardware (OSH) refers to physical components generated by a decentralized design and development model encouraging open collaboration. OSH is an analogy to open source software. OSH usually intends that information about the hardware is readily identifiable so that others can make it. OSH refers to hardware designs whose specifications, construction instructions, and documentation can be publicly accessed, modified, and used by others. Hardware generally can refer to everything that can be built. Hence, it also applies to the do-it-yourself culture and its maker subculture in the broadest sense. The maker scene is technology oriented with a major interest in but not restricted to electronics, mechanics, and robotics.

Open source software (OSS) is free, that is, everybody has the freedom to download or copy, use, study, and distribute it, with or without modifications for free. So, one requires only a computer and a respective runtime system to execute the software. However, additional tools such as editors and compilers are required for inspection, modification, and translation. Yet most of these software tools are also available as open source.

Sometimes OSH is also referred to as free and open source hardware (FOSH). However, hardware is physical and cannot just be downloaded or replicated. Producing hardware requires physical resources and also costs money. Manufacturing a product requires a bill of materials (that is, a list of required raw materials, components, and tools), which obviously has a specific price. Thus, here freedom refers to hardware made from a free design, for example, mechanical drawings, schematics, a layout data of printed

Digital Object Identifier 10.1109/MC.2021.3099046
Date of current version: 24 September 2021

FROM THE EDITOR

One of the amazing things about open source is that it is not just about open source software (OSS) code only any longer. The open source movement has brought open collaboration and open licensing into all kinds of domains, ranging from wikis and Wikipedia through open access to this month's column on open source hardware (OSH). The authors of this article, Hannig and Teich, provide us with an overview of a deep and rich topic that might well warrant its own column. We'll learn both about fundamental principles and inspiring examples and how OSH is a valid sibling to OSS. Enjoy, and as always be safe and healthy! — *Dirk Riehle*

circuit boards (PCB) or integrated circuits, source codes given in a hardware description language (HDL), and a bill of materials.

As already mentioned, the term *hardware* can be interpreted very broadly. The whole range of hardware is considered by the Open Source Hardware Association (OSHWA), a nonprofit organization that provides principles, an elaborated definition of OSH, best practices, and arranges events around open source hardware.

In the following, we just focus on electronic hardware.

ARDUINO, AN OSH SUCCESS STORY

The Arduino project began more than 15 years ago by a few students. The project offers anyone interested in getting creative around electronics and microcontroller programming the opportunity to develop working prototypes that connect the physical and digital worlds quickly. Applications include artistic installations, music, games, toys, robotics, smart homes, smart manufacturing, and smart agriculture.

Arduino consists of both OSH and OSS. The software comprises an integrated development environment and the Arduino programming language, which are rather C/C++ library functions to simplify the programming. Regarding the hardware, all schematic diagrams and design files (layout, parts list, and so on) for manufacturing the PCB are freely available. While the construction plans are open source, one should note that the principal functionality of such boards depends on closed source components (for example, proprietary chips) and thus also on their availability.

One could list many more very flourishing open source tinker electronic platforms, minicomputers, and Internet of Things (IoT) devices. However, in this article, we want to focus on the heart of every computer, its processor. Thanks to the continuous miniaturization of semiconductors during the last half century, it is possible to integrate complex systems consisting of millions to billions of transistors into one silicon chip. Such highly integrated circuits, known as a system-on-a-chip (SoC), are used in mobile computing and many embedded devices. SoCs integrate one or more processors cores, possibly different types, memories, interconnects, perhaps embedded graphics processing, audio and video decoders/encoders, digital signal processing components, artificial intelligence (AI) accelerators, radio modems, and sensors, interfaces to access off-chip memories and other peripherals. These components are referred to as intellectual property (IP) cores.

IP CORES—THE INPUT ARTIFACTS OF SILICON MANUFACTURING

An IP core denotes a predesigned and reusable functional block or blueprint used within a semiconductor chip design. It contains the IP of the developer or manufacturer. These building blocks allow for a modular design of SoCs and thus avoid redundant development work and costs. IP cores can be owned and used by an individual or licensed to others. Many companies in the chip design business are fabless and solely focus on developing and licensing IP cores. Prominent examples include Arm Limited, known as Arm (see www.arm.com), and Imagination Technologies, as well as electronic design automation (EDA) companies such as Cadence Design Systems and Synopsys. IP cores can be divided into soft and hard.

A *soft IP core* is typically a synthesizable register-transfer level (RTL) model specified in an HDL or the form of a generic netlist. Soft IP cores are technology independent and can be flexibly reused and modified in their functionality. They are soft because they must be compiled (synthesized) and mapped to a specific semiconductor technology node on the way to the chip.

A *hard IP core* denotes a low-level implementation specific to a given semiconductor technology. These hard macros are commonly provided as a physical layout (for example, GDSII format). At this low level, a design can hardly be changed. Yet the required chip area, performance, and power can be determined very precisely.

IS OSH SOMETHING NEW?—FROM HACKING INDIVIDUALS TO LARGE ORGANIZATIONS

The first freely available processor cores date back to the turn of the millennium when two RISC CPU architectures followed the OSH movement. They are LEON-1 [based on the SPARC V8 instruction set architecture (ISA)] and the OpenRISC 1000 core. Both were developed in one-person projects, and their sources were released as HDL code together with a corresponding C compiler. At the same time, the OpenRISC developer also founded OpenCores, a web-based portal for open source IP cores for application-specific integrated circuit and field-programmable gate array (FPGA) design. Rather than being an open collaboration platform,

it provides releases of IP core designs developed by individual enthusiasts.

Within the last 10 years, several other nonprofit associations and umbrella organizations have been founded. One OpenCores offspring is the Free and Open Source Silicon (FOSSi) Foundation, which coined the term "open source silicon" and promotes and supports the open design of digital hardware and related ecosystems. The foundation operates the LibreCores website, a gateway to free and open source soft IP cores. FOSSi is run by volunteers and financed by donations and sponsors. Other nonprofit open hardware organizations are financed through membership models, for example, the Open Compute Project (OCP), which is a foundation that shares designs of computer infrastructure and data center products. OCP defines openness broadly, ranging from fully open source to compliance with existing open interfaces.

RISC-V International, formerly known as RISC-V Foundation, is a nonprofit organization with the mission to standardize and promote the free and open RISC-V ISA[1] together with its hardware/software design ecosystem. The RISC-V community is rapidly growing.[2] By April 2021, RISC-V International had more than 1,500 members, including hundreds of renowned companies and research institutes. A study by Semico Research forecasts the market will consume more than 60 billion RISC-V CPU cores by 2025.

Finally, the CHIPS Alliance, a nonprofit organization hosted by the Linux Foundation, develops and hosts high-quality OSH projects related to silicon devices and FPGAs. Organized in several working groups, the alliance covers the entire hardware/software design ecosystem, including verified IP cores (for example, processors, interconnects, accelerators, analog components, and mixed-signal blocks) and SoC designs and open source EDA tools for design and verification. Many projects are centered around RISC-V, and there is a strong collaboration with RISC-V International. The CHIPS Alliance is backed by contributors like Google, Intel, Western Digital, SiFive, Alibaba, Esperanto Technologies, Antmicro, and many others.

ECOSYSTEMS FOR OSH

From its idea to the final silicon product, there is a long way to go with many intermediate steps. Consequently, designing SoCs requires an entire open ecosystem. Its integral parts include the following:

- *Open specifications and open standards*: In the context of processors, it is an abstract model of its programmer's interface, the so-called ISA. Several open and royalty-free ISAs exist: the mentioned SPARC ISA by Sun Microsystems and OpenRISC ISA, MIPS ISA, Power ISA that roots in IBM's development, and RISC-V ISA (which is remarkably designed in academia from the ground up for being shared open source).
- *Description languages*: Design entry and the specification of soft IP cores are dominated by the IEEE-standardized hardware description languages VHDL; Verilog; its superset SystemVerilog with enhancements towards verification; and System C, which is, rather, a system-level modeling language. Apart from the object orientation in the latter two HDLs, the programming language concepts are quite old-fashioned, but being widespread, these HDLs are also the most supported by design tools.

  Recently, several open source HDLs have emerged from academia. They all follow the same principle of being embedded in a host programming language. For being adopted in tool flows (see the next bullet, EDA tools), these open source HDLs all provide translators to VHDL, Verilog, or SystemVerilog.

  The hardware community propels open source a lot. Yet, it is challenging to determine quantitatively how large the hardware share of open source is overall. Looking at GitHub, projects in HDLs make up only a tiny fraction. One reason might be that only established HDLs are named, but embedded HDLs only appear under their host languages (for example, Scala or Python). At least SystemVerilog and Verilog have made it into the top 50 languages on GitHub within the last two years. However, both contribute less than 0.07% to GitHub's overall pull requests (according to GitHut 2.0, https://madnight.github.io/githut/).
- *EDA tools*: EDA is a generic term for software tools for the design of electronic systems. Still, we again focus on integrated circuits in the following. EDA tools can be viewed top-down, that is, from design entry down to physical implementation. Coarsely, this trajectory of refinement stages can be divided into front-end design (among others consisting of specification, high-level design, RTL coding, and functional verification) and back-end design (for example, logic synthesis, floorplanning, place and route, and preparation for manufacturing). The result of a front-end design can be a verified soft IP core, whereas the design of a hard IP core also requires stages in the back end. OpenROAD/OpenLANE[3] are collections that stitch many tools together to an entire back end design flow from RTL to GDSII output. The momentum and coverage of open source EDA are also reflected in the *IEEE Design & Test* special issue of the same name.[4]
- *Process design kits* are a set of files used within the semiconductor industry to model a fabrication process.
- *Software stack for processors* includes compilers (for example,

GCC or LLVM), operating systems (for example, Linux), and emulators and virtualization (for example, QEMU).

A remarkable milestone in offering a complete manufacturing chain for open source silicon has been set up by a partnership of Google, the SkyWater foundry, and efabless, a service provider for custom silicon designs. Google sponsors multiproject wafer (MPW) shuttles in SkyWater's 130-nm mixed-signal CMOS technology. Both a corresponding process design kit and the OpenLANE tool stack are fully open source. Moreover, closed source projects can also employ the entire open source design flow, making prototypes and low-volume chip productions affordable at a price of about US$10,000 per project.

In the following, we elaborate on the major benefits but also challenges of OSH.

BENEFITS OF OSH

- *Innovation*: Collaborative OSH design can reduce cost by reuse, lower time to market, and improve quality. Paired with the free competition of a large developer base, irrespective of designing proprietary or open cores, this drives innovation and can stimulate product differentiation through novel custom chip designs.
- *Democratization*: Hardware expertise becomes available for everyone without cost for licensing or extensive training. Customized hardware can become affordable for more devices, which is especially important for low-cost IoT devices.
- *Security*: Security has been often achieved through obscurity, that is, one keeps something closed. However, security hackers try to find and exploit weaknesses in a computer system. OSH also allows the "good guys" to detect vulnerabilities and to fix them quickly.
- *Transparency and trust*: Using proprietary hardware often comes with uncertainty, whether there is some built-in backdoor (for example, in encryption) or how much data are collected. Moreover, the evolution of technology enables the design of very complex hardware/software systems. Their sheer complexity and nebulosity can be scary and lead to a trust crisis. Open source offers transparency and traceability, and can thus ultimately lead to trust in hardware technology.
- *Research and education* can be closer tied to real-world hardware architectures when building upon full OSH/OSS stacks.

CHALLENGES AND OUTLOOK

- *Legal aspects*: Many challenges are linked to legal aspects, including the following questions:
  - Does OSH require specific licensing models? Just a few remarks: Noncommercial licenses are problematic since money is always involved in hardware production. Permissive software licenses fit well with OSH. Yet, a challenge here is due to the different terminologies used in the software and hardware worlds. For example, FOSSi's Solderpad hardware license is based on a slightly modified Apache license 2.0 and solves the terminology issue by providing a kind of dictionary. The situation is more difficult in copy-left licenses that utilize the copyright to impose obligations on licensees. However, copyright does not cover the production and distribution of hardware (for more details, see reciprocal variants of CERN's Open Hardware Licence v2).
  - Where should nonprofit OSH organizations be anchored to not being impaired by political disruption? What happens to OSH in the case of export control regulations?
- *Protection of proprietary data*: Traditionally, semiconductor foundries require proprietary preparation and manufacturing steps. They and the EDA industry are protectionist and usually share their trade secrets (technology data, standard cell libraries, tools, IP cores) only under a nondisclosure agreement and costly licenses. One challenge is protecting closed source IP without compromising user access to OSH. Another question is how protected semiconductor IP can coexist within an open hardware design platform? These are not only legal issues again but also impose technological challenges such as encryption or watermarking of IP cores.
- *Reducing costs*: Relieving hardware design from expensive tool and IP licenses is a first important milestone. Yet, the fixed cost for taping out chips in 20-years-old technology (130 nm) is relatively expensive. Here, the question is how service providers (for example, the MOSIS Service, EUROPRACTICE, or efabless) for MPW fabrication and OSH can make low-volume productions in state-of-the-art technology affordable. How should service providers deal with vendor IDs, since they cannot be obtained free of charge?
- *Business models*: How should EDA companies and foundries be encouraged to participate more actively in the OSH ecosystem? For example, they could contribute the central part of a hardware product open source, while offering additional features as closed source hardware or provide (cloud) services for EDA and open hardware design.

- *EDA*: How can productivity, complexity, interoperability, and diversification be further improved? Architecture description languages and integration tools to customize and assemble synthesizable SoCs, infer simulators, verification, and a corresponding software stack can tackle these issues. The RISC-V-centric Chipyard framework is a recent example. Raising the abstraction level of design entry through domain-specific languages and corresponding tools to generate domain-specific accelerators [5] and heterogeneous processors can stimulate agility and raise productivity.

In conclusion, we believe that rapidly growing open communities and developments will soon overcome the above challenges, and OSH will pave the way to a plethora of new fantastic electronic devices.

REFERENCES

1. K. Asanović and D. A. Patterson, "Instruction sets should be free: The case for RISC-V," Univ. of California, Berkeley, Tech. Rep. UCB/EECS-2014-146, 2014.
2. S. Higginbotham, "The rise of RISC," *IEEE Spectr.*, vol. 55, no. 8, p. 18, 2018.
3. A. Ghazy and M. Shalan, "OpenLANE: The open-source digital ASIC implementation flow," in *Proc. Workshop on Open-Source EDA Technol. (WOSET)*, 2020, Art. no. 21.
4. S. Reda, L. Stok, and P.-E. Gaillardon, Guest Eds., "Special issue on open-source EDA," *IEEE Des. Test.*, vol. 38, no. 2, 2021.
5. O. Reiche, M. A. Özkan, R. Membarth, J. Teich, and F. Hannig, "Generating FPGA-based image processing accelerators with Hipacc," in *Proc. Int. Conf. Comput. Aided Des. (ICCAD)*, 2017, pp. 1026–1033. doi: 10.1109/ICCAD.2017.8203894

FRANK HANNIG is a lecturer and senior researcher in the Department of Computer Science, Friedrich-Alexander University Erlangen-Nürnberg, Erlangen, 91058, Germany. Contact him at frank.hannig@fau.de.

JÜRGEN TEICH is a full professor and chair of Hardware/Software Co-Design in the Department of Computer Science, Friedrich-Alexander University Erlangen-Nürnberg, Erlangen, 91058, Germany. Contact him at juergen.teich@fau.de.

Digital Object Identifier 10.1109/MC.2021.3108826