

Teaching Integrated Circuit Design with Open Source Electronic Design Automation Tools and Open Process Design Kits

Piotr Mierzwiński, Bartosz Dec

Institute of Microelectronics and Optoelectronics, Faculty of Electronics and Information Technology
Warsaw University of Technology, Warsaw, Poland
piotr.mierzwiński@pw.edu.pl, bartosz.dec@pw.edu.pl

Abstract—This extended abstract presents a course on integrated circuit design with open-source electronic design automation tools and Open Process Design Kits. It connects device, digital and analog circuit, and physical-design knowledge in a transparent flow reproducible beyond a closed laboratory. Lectures, guided laboratories and repository exercises cover requirements, hardware description, simulation, synthesis, physical design, verification and basic full-custom work. The first implementation used a controlled Linux environment, a Verilog–Verilator–Yosys–LibreLane digital flow and selected analog exercises with Open PDK resources. Results indicate that open tools complement commercial EDA by supporting independent practice, exposing intermediate files and reports, and making debugging visible. Limitations include analog-tool configuration complexity and the need for prepared images, diagnostic scripts and reference projects. The work concludes that open flows can strengthen practical microelectronics education when introduced as reproducible, carefully supported laboratories.

Keywords—integrated circuit design; open-source electronic design automation; open process design kit; microelectronics education; RTL-to-GDSII; laboratory teaching

I. MOTIVATION AND AIM

Integrated circuit design education must connect device physics, circuit design, hardware description, verification, physical implementation and sign-off into one engineering process. The reported course makes this connection explicit through an open, reproducible flow. It does not replace industrial EDA environments, but complements them with a transparent setting in which students inspect files, constraints, reports and intermediate artefacts without confidentiality barriers.

II. COURSE AND LABORATORY METHOD

The course follows the logic of an integrated circuit design project. Lectures introduce open-source software and hardware, licensing constraints, requirements engineering, hardware description languages, process design kit contents, layout, floorplanning, placement, clock-tree synthesis, routing, finishing and tape-out preparation. The laboratory component was intentionally scoped to basic digital and analog flows.

In the digital part students implemented Verilog designs, simulated them with Verilator, synthesized them with Yosys and executed an automated physical flow in LibreLane. Tasks included simple counters and an arithmetic and logic block with an APB5 interface. In the analog part students moved through schematic entry, simulation, layout, design rule checks

and layout-versus-schematic checks for simple full-custom cells and selected blocks.

III. OBSERVATIONS FROM THE FIRST IMPLEMENTATION

The first implementation confirmed that students value practical contact with design artefacts: source code, testbenches, configuration files, reports, layout files, netlists and verification logs. Educational value came not only from successful runs, but also from configuration errors, violated constraints, failing checks and iterative debugging. Open tools let students repeat part of the work outside the laboratory and compare the openness of this flow with the maturity of commercial tools. The main difficulty was analog-tool and library configuration. A prepared x86 Linux environment reduced this risk, while private installations required additional support.

IV. BROADER ECONOMIC CONTEXT

Beyond education, a complete open-source IC design toolchain has economic significance. It lowers the perceived entry barrier to semiconductor design and shows that early prototyping, experimentation and small demonstrator projects do not always require expensive commercial licenses. This matters for graduates considering start-ups, own IP blocks or small-scale design activities. Foundries and technology providers releasing Open PDKs may also broaden their future customer base by making their technologies accessible to students, researchers and SMEs. Such access can encourage SMEs to explore semiconductor solutions and supports a more inclusive innovation ecosystem.

V. CONCLUSIONS

Open-source EDA and Open PDKs can bridge theory and industrial practice. They make the IC design process observable, reproducible and discussable in class. Future work should focus on packaged environments, diagnostic scripts, reference repositories, team projects and systematic evaluation based on questionnaires and student artefacts

REFERENCES

- [1] The OpenROAD Project, "OpenROAD documentation," accessed Apr. 30, 2026. [Online]. Available: <https://openroad.readthedocs.io/>
- [2] IHP GmbH, "IHP Open PDK documentation," accessed Apr. 30, 2026. [Online]. Available: <https://ihp-open-pdk-docs.readthedocs.io/>
- [3] OS-EDA, "Open-source EDA course," GitHub repository, accessed Apr. 30, 2026. [Online]. Available: <https://github.com/OS-EDA/Course>
- [4] IHP GmbH, "IHP Analog Academy," GitHub repository, accessed Apr. 30, 2026. [Online]. Available: <https://github.com/IHP-GmbH/IHP-AnalogAcademy>