GDSFactory: An Open-Source Python Library for Chip Design and Simulation

Joaquin Matres,[1,*] Simon Bilodeau,[2] Niko Savola,[3,4] Marc de Cea,[3] Wai Kwan Yeung,[5] Erman Timurdogan,[6] Jan David Fischbach,[7] Helge Gehring,[8] Lucas Grosjean,[9] Yannik Mahlau,[10] Floris Laporte,[1] Sebastian Goeldi,[1] and Troy Tamas[1]

[1]*GDSFactory, 650 Castro St Ste 120 PMB 98035, Mountain View, CA 94041, USA*
[2]*Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544, USA*
[3]*Taara Connect, Inc., Sunnyvale, CA 94089, USA*
[4]*Department of Applied Physics, Aalto University, P.O. Box 13500, FI-00076 Aalto, Finland*
[5]*Department of Physics, University of Oxford, Oxford, UK*
[6]*Lumentum, San Jose, CA, USA*
[7]*Institute of Nanotechnology, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany*
[8]*Institute of Physics and Center for Nanotechnology, University of Münster, Münster D-48149, Germany*
[9]*Department of Optics, FEMTO-ST Institute, 25000 Besançon, France*
[10]*Institute of Information Processing, Leibniz University Hannover, Germany*

[*]jmatres@gdsfactory.com

Abstract: We present GDSFactory, an open-source Python library for integrated circuit design automation supporting photonics, analog, quantum, and MEMS applications. The platform provides unified workflows for full chip development: layout design and simulations (device and circuit via S-parameter analysis).

1 Introduction

Photonic wafer fabrication cycles can take multiple months and many iterations to resolve Design-for-X (DfX) challenges, including manufacturing, test, and reliability. These iterations can add cost and delay customer delivery and ultimately result in loss of business. Software for design and verification cycles are fast and inexpensive but can be limited in capability to match ground truth. GDSFactory bridges this gap by providing a comprehensive Python API for chip development, including layout design, device simulation, circuit simulation, and verification [1].

In contrast to traditional logic-driven electronic design flows [2], integrated photonics requires freeform parametric geometries and tight integration between layout and electromagnetic simulation. GDSFactory addresses this with scriptable parametric cells (PCells), hierarchical assembly, routing, and seamless interfaces to industry-standard simulators while leveraging Python's extensive scientific ecosystem [3] (Fig. 1).

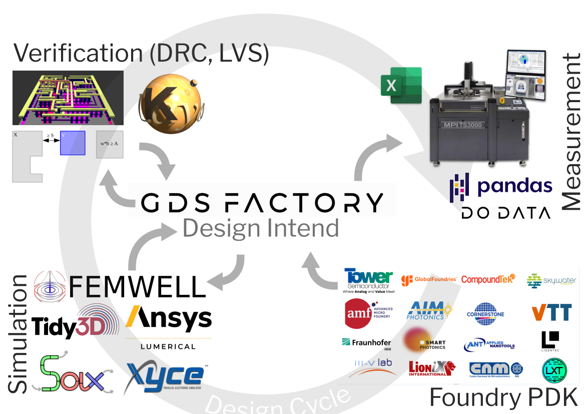

Figure 1. GDSFactory workflow: GDSFactory spans the entire design cycle, with a variety of Foundry PDKs available it is both easy to construct circuits reusing proven devices and to realize conceptually new designs. The generated component layouts can be used in various simulators for exploration, optimization and validation. GDSFactory tightly integrates with KLayout, leveraging its advanced design rule checks (DRC) and layout versus schematic (LVS) capabilities. For characterization of the fabricated devices GDSFactory provides rich metadata compatible to commercial wafer probers, including the position and orientation of fiber-to-waveguide couplers.

2 Layout Design

GDSFactory implements cells as Python functions returning a `Component` class with polygons, electrical and optical port metadata, and convenience methods for export and plotting. Using KLayout's C++ geometry engine [4], users define parametric cells through a functional programming approach where the `@gf.cell` decorator handles caching to eliminate redundant regeneration. Port metadata enables automatic routing via various routing functions that define routes between ports or groups of ports.

3 Device Simulation

GDSFactory's gplugins repository [5] provides unified interfaces to a growing list of simulators by reusing the core layout abstractions (Components, Layerstacks, Ports). For instance, Components can be meshed via GMSH [6] (through a wrapper [7]) for cross-sectional or 3D analysis (Fig. 2) For rigorous electromagnetic verification required by photonics, the library supports full-wave analysis via both Finite-Difference Time Domain (FDTD) and Finite Element Method (FEM) approaches. The FDTD method is supported through open-source backends including MEEP [8] and Luminescent AI, as well as solvers commonly used in industry such as Tidy3D [9] (cloud-based GPU acceleration) and Lumerical FDTD. A plugin for the open-source fdtdx solver is currently in development [10]. Available FEM-based and multiphysics solvers include Femwell [11] for waveguide mode analysis and thermal simulation, Palace [12] for RF and microwave simulations, and MEOW [13] for eigenmode expansion. For lower-level modeling, TCAD simulations are also supported through DEVSIM [14] for semiconductor device physics and Sentaurus for advanced process simulation.

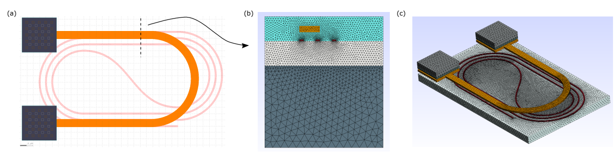

Figure 2. GDSFactory meshing: (a) heater layout, (b) cross-sectional mesh, (c) 3D mesh.

4 Circuit Simulation

Circuit-level simulation enables system-scale photonic design. GDSFactory facilitates this through netlist extraction, which enables compositions of device-level simulations. For example, we have used SAX [15] for differentiable S-parameter circuit simulation using JAX, which supports automatic differentiation for gradient-based optimization, Monte Carlo analysis for yield estimation, and wavelength-dependent S-parameter interpolation from FDTD results. VLSIR [16] provides SPICE netlist export for mixed photonic-electronic simulation.

5 Process Design Kits

Open-source PDKs include GlobalFoundries 180nm, SkyWater 130nm [17], VTT 3 µm SOI, SiEPIC, Cornerstone, IHP, Luxtelligence, and Quantum-RF-PDK. These foundry process design kits are critical for enabling GDSFactory to fit easily into existing development flows while matching ground-truth data. Commercial PDKs available through GDSFactory+ subscription [18] include AIM Photonics, AMF, CompoundTek, Fraunhofer HHI, Smart Photonics, Tower PH18, OpenLight, III-V Labs, LioniX, Ligentec, Lightium, and QCI. The generic PDK follows standard layer conventions [19] for cross-foundry compatibility.

6 Conclusion

GDSFactory provides a unified Python-driven workflow spanning layout, device simulation, and circuit simulation. The tight integration between device solvers and circuit simulators enables rapid design iteration from component to system level, and its programmatic, open nature enables modern agentic workflows [20]. The library is freely available at `https://github.com/GDSFactory/GDSFactory`.

References

[1] J. Matres *et al.*, "GDSFactory," GitHub (2024), `https://github.com/GDSFactory/GDSFactory`.
[2] W. Bogaerts *et al.*, "Silicon photonics circuit design: methods, tools and challenges," Laser Photon. Rev. **12**, 1700237 (2018).
[3] J. Matres, "Awesome Photonics," GitHub (2024), `https://github.com/joamatab/awesome_photonics`.
[4] M. Köfferlein, "KLayout," `https://www.klayout.de/`.
[5] J. Matres *et al.*, "gplugins," GitHub (2024), `https://github.com/gdsfactory/gplugins`.

[6] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities," *Int. J. Numerical Meth. Engng*, vol. 79, no. 11, pp. 1309–1331, 2009. [Online]. Available: https://doi.org/10.1002/nme.2579.

[7] S. Bilodeau *et al.*, "Meshwell," GitHub (2026), https://github.com/simbilod/meshwell.

[8] A. F. Oskooi *et al.*, "MEEP: A flexible free-software package for electromagnetic simulations by the FDTD method," Comput. Phys. Commun. **181**, 687–702 (2010).

[9] Flexcompute Inc., "Tidy3D," https://www.flexcompute.com/tidy3d/.

[10] Y. Mahlau *et al.*, "A flexible framework for large-scale FDTD simulations: open-source inverse design for 3D nanostructures," in *Photonic and Phononic Properties of Engineered Nanostructures XV*, vol. 13377, pp. 40–52 (2025), https://github.com/ymahlau/fdtdx.

[11] H. Gehring *et al.*, "Femwell," GitHub (2023), https://github.com/HelgeGehring/femwell.

[12] AWS, "Palace: 3D Finite Element Solver for Computational Electromagnetics," GitHub (2024), https://github.com/awslabs/palace.

[13] F. Laporte, "MEOW," GitHub (2024), https://github.com/flaport/meow.

[14] J. E. Sanchez, "DEVSIM: A TCAD Semiconductor Device Simulator," *Journal of Open Source Software*, vol. 7, no. 70, p. 3898, Feb. 2022. [Online]. Available: https://doi.org/10.21105/joss.03898

[15] F. Laporte, "SAX," GitHub (2023), https://github.com/flaport/sax.

[16] D. Fritchman *et al.*, "VLSIR" GitHub (2024), https://github.com/vlsir/vlsir.

[17] SkyWater Technology Foundry and Google, "SkyWater Open Source PDK," GitHub (2023), https://github.com/google/skywater-pdk.

[18] GDSFactory, "GDSFactory+," https://gdsfactory.com/.

[19] L. Chrostowski and M. Hochberg, *Silicon Photonics Design* (Cambridge, 2015).

[20] A. Sharma *et al.*, "AI agents for photonic integrated circuit design automation," *APL Machine Learning*, vol. 3, no. 4, Dec. 2025. doi: 10.1063/5.0300741.