Abstract: We present gdsfactory, an open-source Python library for integrated circuit design automation supporting photonics, analog, quantum, and MEMS applications with unified design, simulation, verification, and validation workflows.

1 Introduction

Hardware iterations typically require months and millions of dollars, while software iterations cost hundreds of dollars and take hours. Gdsfactory bridges this gap by providing a comprehensive Python API for chip development, including layout design, verification, and validation [1].

Unlike constrained logic-driven electronic design flows [2], integrated photonics requires freeform parametric geometries. Gdsfactory addresses this with scriptable parametric cells (PCells), hierarchical assembly, and routing—all in Python's extensive scientific ecosystem.

2 Design Capabilities

Gdsfactory implements cells as Python functions returning a `Component` class with polygons, port metadata, and convenience methods. Using a C++-based gdstk backend [3], users define PCells with a functional programming approach:

```
import gdsfactory as gf

@gf.cell
def mzi_with_bend(radius=10):
    c = gf.Component()
    mzi = c.add_ref(gf.components.mzi())
    bend = c.add_ref(
        gf.components.bend_euler(radius=radius))
    bend.connect('o1', mzi['o2'])
    return c
```

The `@gf.cell` decorator handles caching, eliminating redundant regeneration. Port metadata enables automatic routing via `get_route` and `get_bundle` functions that connect components following `CrossSection` specifications (Fig. 1).

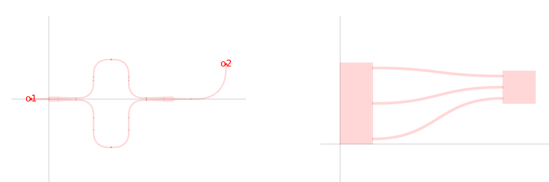

Figure 1. (a) MZI with Euler bend showing port connections. (b) Routed $n \times n$ components using S-bends.

3 Simulation and Verification

Gdsfactory's gplugins repository interfaces with simulators by reusing layout abstractions. Device-level exports include: FDTD solvers (MEEP [4], Tidy3D, Lumerical), mode solvers (Femwell, MPB), EME (MEOW), and TCAD (DEVSIM, Sentaurus). Components can be meshed via GMSH for cross-sectional or 3D analysis (Fig. 2).

Circuit-level simulation uses netlist extraction with SAX for S-parameter analysis [5] and VLSIR for Spice formats. Design rule checking integrates with KLayout through an extensible API.

4 Process Design Kits

Open-source PDKs include GlobalFoundries 180nm, SkyWater 130nm [6], VTT 3 µm SOI, and SiEPIC. Commercial PDKs under NDA include AIM, AMF, TowerSemi, IMEC, and HHI. The generic PDK follows standard layer conventions [7] for cross-foundry compatibility.

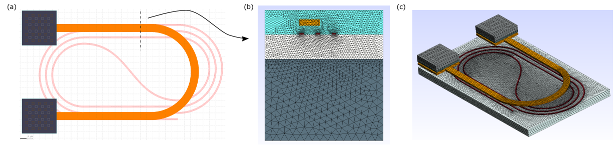

Figure 2. Gdsfactory meshing: (a) heater layout, (b) cross-sectional mesh, (c) 3D mesh.

5 Conclusion

Gdsfactory provides a unified Python-driven workflow for chip design, verification, and validation. Its integration with simulators, open-source PDKs, and extensible architecture accelerates hardware development across photonics, quantum, and analog applications. 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] L. H. Gabrielli, "gdstk," GitHub (2023), https://github.com/heitzmann/gdstk.

[4] 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).

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

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

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