PHIDL: Intuitive GDS layout and CAD geometry creation for Python

Adam N. McCaughan*, Alexander N. Tait, Sonia M. Buckley, Jeff T. Chiles, Sae Woo Nam
National Institute of Standards and Technology, Boulder, CO 80305
*Correspondence to: adam.mccaughan@nist.gov

Computer-aided design (CAD) has become a critical element in the creation of nanopatterned structures and devices. In particular, with the increased adoption of easy-to-learn programming languages like Python there has been a significant rise in the amount of lithographic geometries generated through scripting and programming. However, there are currently unaddressed gaps in usability for open-source CAD tools -- especially those in the GDSII design space -- that prevent wider adoption by scientists who might otherwise benefit from scripted design. Many GDS tools exist [1][2], but they tend to fall in one of two categories: (1) GUI-based layout tools with ad-hoc scripting interfaces, or (2) full-featured Cadence-style layout software which requires 30 lines of boilerplate/overhead code just to draw a simple ring. For example, constructing relations between adjacent geometries is often much more difficult than necessary -- spacing a resonator structure a few micrometers from a readout structure often requires manually-coding the placement arithmetic. While inconveniences like this can be overcome by writing custom functions, they are often significant barriers to entry for new students or those less familiar with programming.

To help streamline the design process and reduce barrier to entry for scripting designs, we have developed PHIDL, an open-source GDSII-based CAD tool for Python 2 and 3 based on gdspy [3]. PHIDL has a large library of premade geometry functions ready for use, including basic shapes, text, lithographic test structures, boolean operations, grow/shrink operations, placement and automatic packing algorithms, superconducting nanowire optimal curves, and photonic waveguide routing. Additionally, as shown in Fig. 1, PHIDL allows the user to define ports on their geometries, which can be used to easily snap shapes together like Lego blocks. It also includes a built-in geometry viewer which can display the geometry of any cell, polygon, port, or other geometric object and has proven to be extremely useful for debugging design scripts.

To minimize the learning curve, we have written a thorough and descriptive tutorial for PHIDL which allows new users to get acquainted quickly. In addition, we have also created geometry function reference documentation, as well as an online interactive notebook which can be used to try out Python and PHIDL using only a web browser. It can be installed easily from the Python package repository using "pip", and the source can be viewed or downloaded from GitHub (https://github.com/amccaugh/phidl).

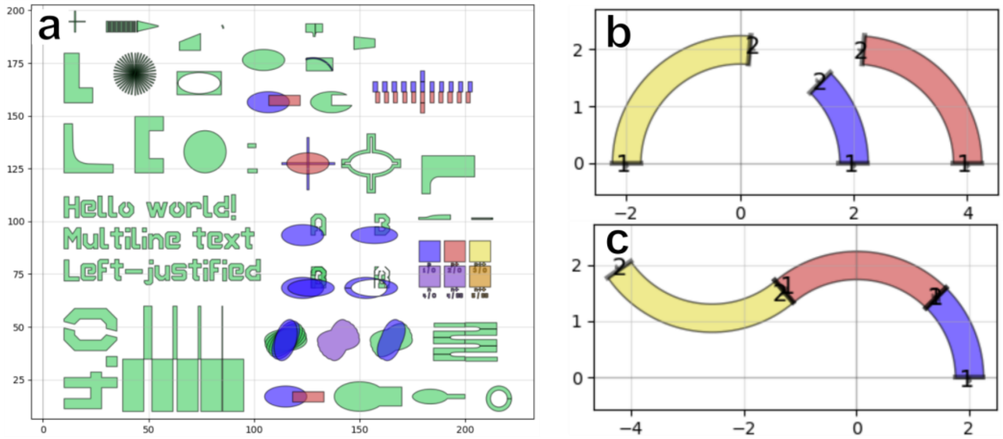

Fig. 1. Geometry library and shape manipulation in PHIDL. **(a)** Examples of shapes and functions available in the PHIDL geometry library including basic shapes, text, layer-alignment calipers, resolution tests, boolean operations, and grow/shrink operations. These shapes were placed within the rectangular area using a built-in packing algorithm. **(b,c)** Example of shape manipulation using ports. Three arcs with ports on each end are assembled in a simple two-line command.

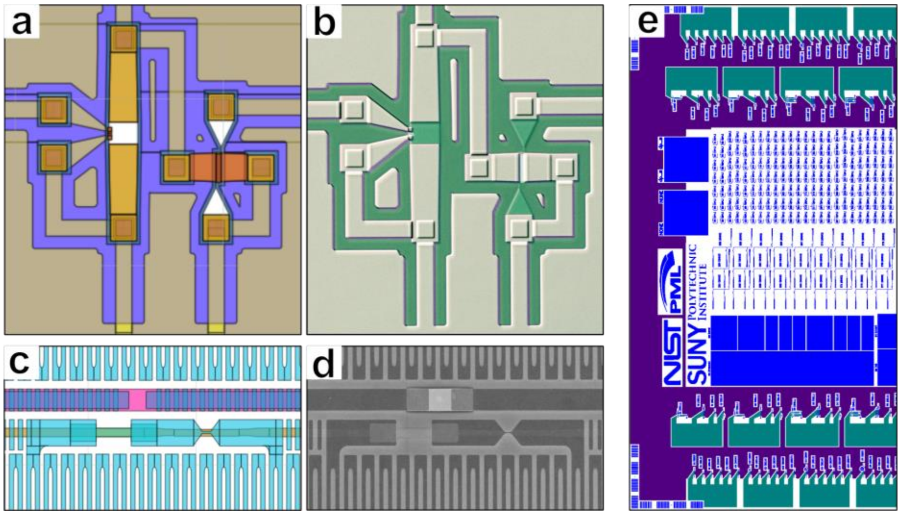

Fig. 2. Example layouts and fabricated layouts made with PHIDL **(a)** Layout of a superconducting-nanowire electronic circuit based on thermal switches [4] with 7 independent layers, and **(b)** resulting microscope image of fabricated device. **(c)** Layout of a multilayer superconducting single photon detector device and **(d)** scanning electron micrograph of the fabricated device. **(e)** Photonic integrated circuit layout made in PHIDL with over 95,000 polygons and 6.8 million points.

[1] K. C. Balram *et al.*, “The Nanolithography Toolbox,” *J. Res. Natl. Inst. Stand. Technol.*, vol. 121, p. 464, Oct. 2016.

[2] W. Bogaerts, P. Dumon, E. Lambert, M. Fiers, S. Pathak, and A. Ribeiro, “IPKISS: A parametric design and simulation framework for silicon photonics,” in *The 9th International Conference on Group IV Photonics (GFP)*, 2012, pp. 30–32.

[3] L. H. Gabrielli, “gdspy.” [Online]. Available: https://github.com/heitzmann/gdspy/.

[4] A. N. McCaughan *et al.*, “A superconducting thermal switch with ultrahigh impedance for interfacing superconductors to semiconductors,” *Nat. Electron.*, Sep. 2019.