LibreLane

Leo Moser

HeiChips 2025

Background: OpenLane

- 2020: First open-source Process Design Kit by Skywater: sky130
- Problem: The PDK was free, but there was no compatible end-to-end open-source flow
- Solution: OpenLane
  - Developed by Efabless Corporation
  - Free & Open-source
  - Batteries-included & Easily configurable
- Hundreds of designs on the Open MPW shuttles

Problems with OpenLane: The Architecture

- **Developed on a very strict deadline**
  - lots of architectural shortcuts taken, no proper abstraction
- Running **parts** of the flow (more common than you think)
  - extremely buggy
- No standard way of reporting data about the design
  - area, power consumption, clock speed…
- Difficult to add custom steps to the flow (integrate other tools)
- Tcl - “stringly-typed” language

Introducing LibreLane

Open-source digital design flow infrastructure

What is LibreLane?

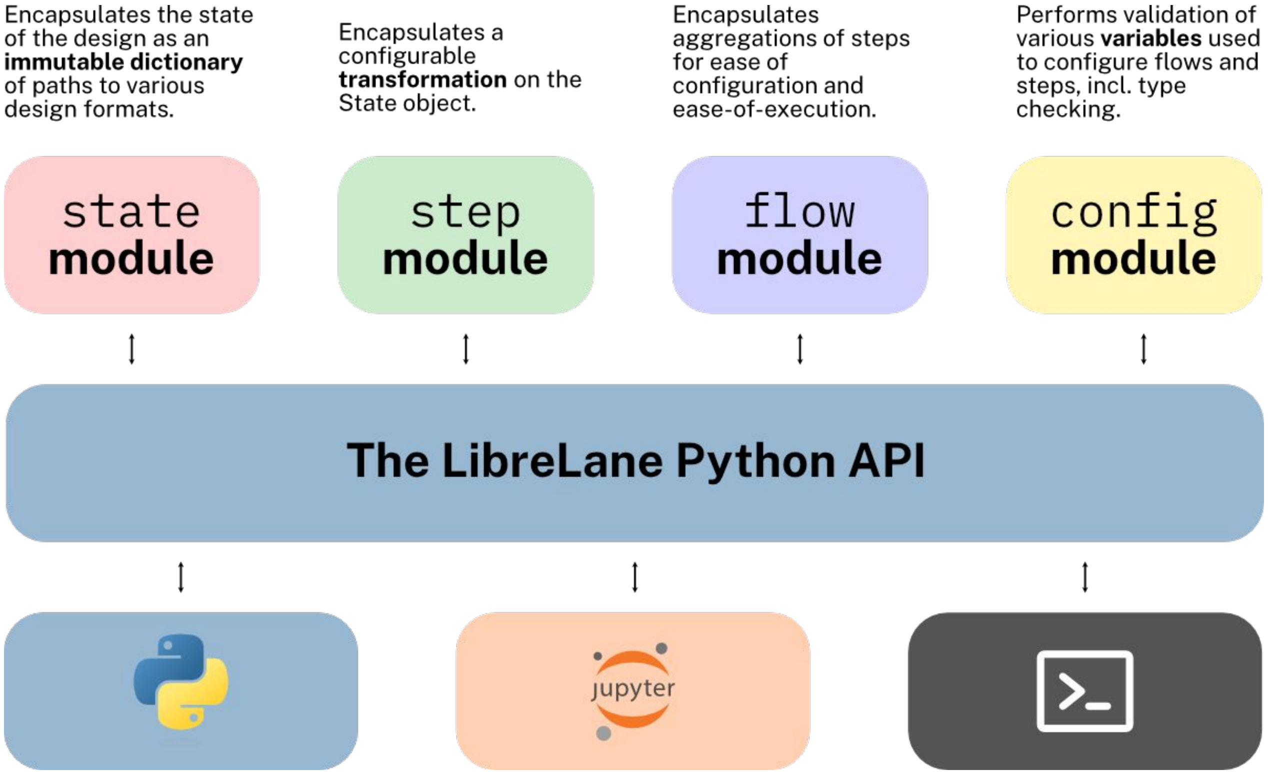
- A ground-up rewrite of OpenLane
    - Started at Efabless as “OpenLane 2”
    - (Mostly) backwards-compatible with OpenLane 1
- Fully customizable **Python**-based Flow
- All tools included and ready to download and run:
    - Natively using a package manager called Nix
    - Containerized using Docker
    - In your browser using Colab
- **Community-driven!**
    - **https://matrix.to/#/#librelane:fossi-chat.org**

https://colab.research.google.com/github/librelane/librelane/blob/main/notebook.ipynb

Technical Design Goals

- **Retain the simplicity of OpenLane**
  - Ultimately, it is still simple to configure your chip and get from Verilog to GDSII
- **Offer robust packaging**
  - Not just build-scripts: something that can replicate the environment in various places, and not only using containerization and/or emulation, but natively
- **Modularity and API Access**
  - Adheres to a simple object-oriented architecture
  - While still supporting OpenLane config files using the "Classic" flow, you can very easily implement your own flows

Architectural Overview



Encapsulates the state of the design as an **immutable dictionary** of paths to various design formats.

Encapsulates a configurable **transformation** on the State object.

Encapsulates aggregations of steps for ease of configuration and ease-of-execution.

Performs validation of various **variables** used to configure flows and steps, incl. type checking.

`state` **module**

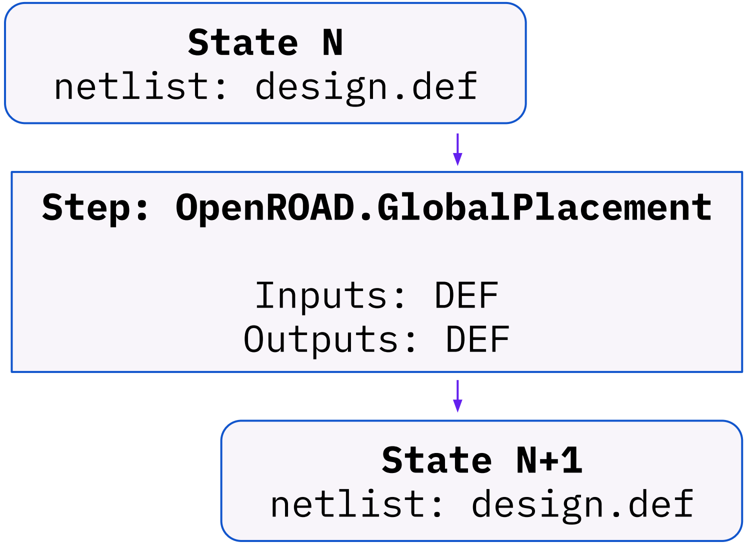
`step` **module**

`flow` **module**

`config` **module**

The LibreLane Python API

LibreLane: State



- Each step consumes a state and generates a new state
- Each state is explicitly saved as a JSON file
- DesignFormat:
  Mapping from view to file

The LibreLane Classic Flow

- **The default flow of LibreLane**
  - Re-implementation of the OpenLane flow using the Python API
- **Key enhancements:**
  - Improved handling of Macros
    - MACROS configuration variable listing macro views and instances
    - Power pins for macros may be declared in RTL with an `ifdef
  - Parallelized STA, including multi-corner extraction and analysis
  - Additional DRC with KLayout
  - Basic SystemVerilog, VHDL support
  - Better tool warning capture

Simple Example

Counter

Simple Example: Counter

config.yaml

```yaml
DESIGN_NAME:   counter
VERILOG_FILES: dir::src/*.sv
CLOCK_PORT:    clk_i
CLOCK_PERIOD:  10 # ns
```

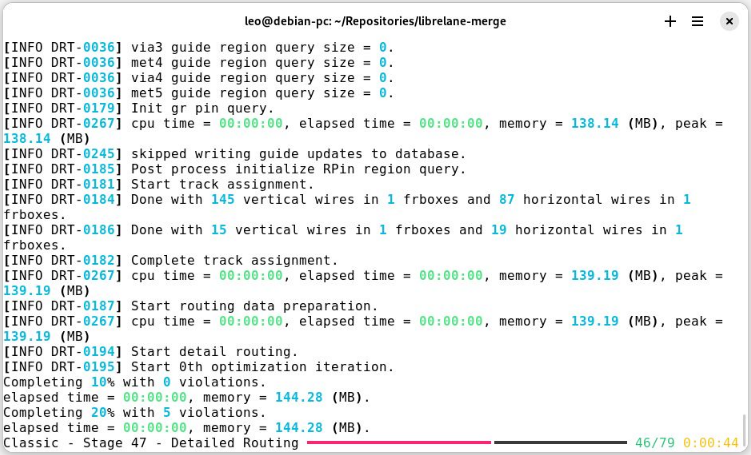
CLI

```
$ librelane config.yaml
```

counter.sv

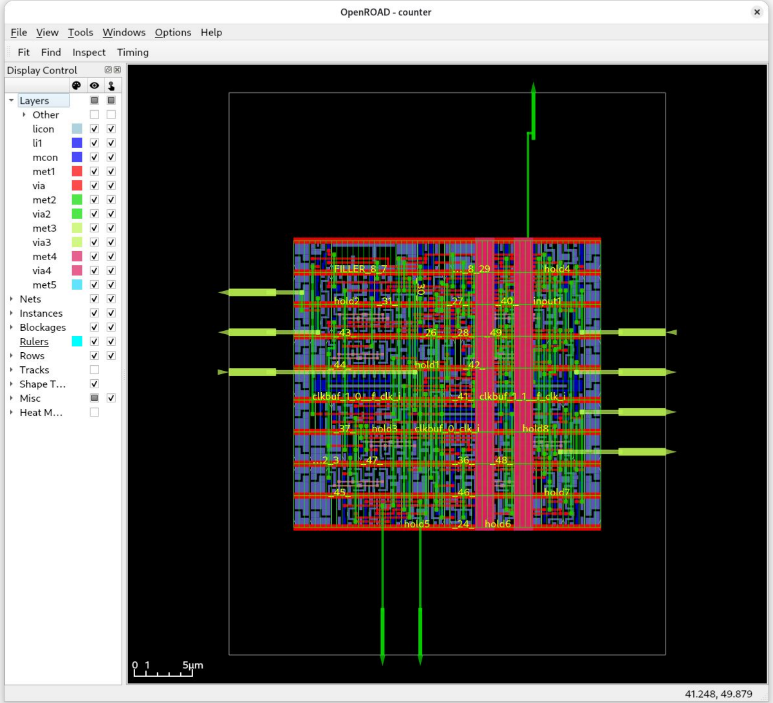
```systemverilog
module counter (
    input  logic       clk_i,
    input  logic       rst_ni,
    output logic [7:0] count_o
);

    always_ff @(posedge clk_i) begin
        if (!rst_ni) begin
            count_o <= '0;
        end else begin
            count_o <= count_o + 1;
        end
    end

endmodule
```

Simple Example: Counter



leo@debian-pc: ~/Repositories/librelane-merge

```
[INFO DRT-0036] via3 guide region query size = 0.
[INFO DRT-0036] met4 guide region query size = 0.
[INFO DRT-0036] via4 guide region query size = 0.
[INFO DRT-0036] met5 guide region query size = 0.
[INFO DRT-0179] Init gr pin query.
[INFO DRT-0267] cpu time = 00:00:00, elapsed time = 00:00:00, memory = 138.14 (MB), peak =
138.14 (MB)
[INFO DRT-0245] skipped writing guide updates to database.
[INFO DRT-0185] Post process initialize RPin region query.
[INFO DRT-0181] Start track assignment.
[INFO DRT-0184] Done with 145 vertical wires in 1 frboxes and 87 horizontal wires in 1
frboxes.
[INFO DRT-0186] Done with 15 vertical wires in 1 frboxes and 19 horizontal wires in 1
frboxes.
[INFO DRT-0182] Complete track assignment.
[INFO DRT-0267] cpu time = 00:00:00, elapsed time = 00:00:00, memory = 139.19 (MB), peak =
139.19 (MB)
[INFO DRT-0187] Start routing data preparation.
[INFO DRT-0267] cpu time = 00:00:00, elapsed time = 00:00:00, memory = 139.19 (MB), peak =
139.19 (MB)
[INFO DRT-0194] Start detail routing.
[INFO DRT-0195] Start 0th optimization iteration.
Completing 10% with 0 violations.
elapsed time = 00:00:00, memory = 144.28 (MB).
Completing 20% with 5 violations.
elapsed time = 00:00:00, memory = 144.28 (MB).
Classic - Stage 47 - Detailed Routing                                46/79 0:00:44
```

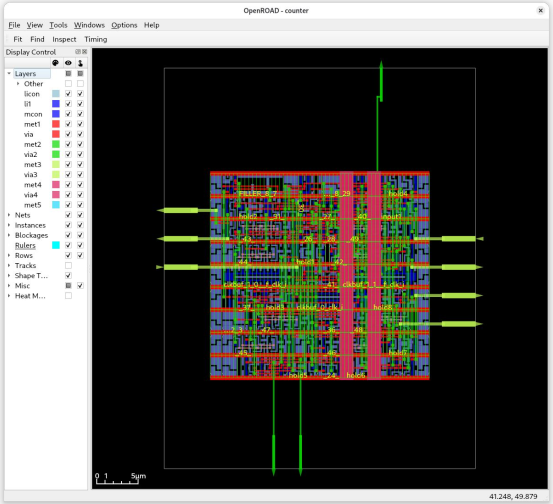


Officially Supported PDKs

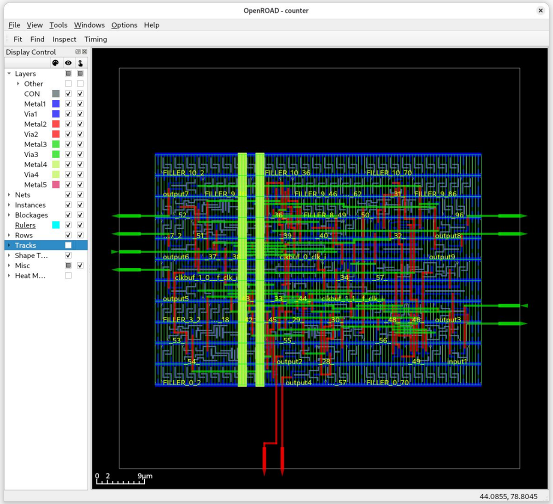
- Open source PDKs
  - sky130 — 2.0.0
  - gf180mcu — 2.0.0
  - ihp-sg13g2 — 3.0.0 (upcoming)
- IHP Open PDK
  - Initial LibreLane support merged!
  - Pre-release in ciel:
    - https://github.com/fossi-foundation/ciel
- Select the PDK
  - `export PDK`
  - pass “`--pdk`” as argument

Officially Supported PDKs

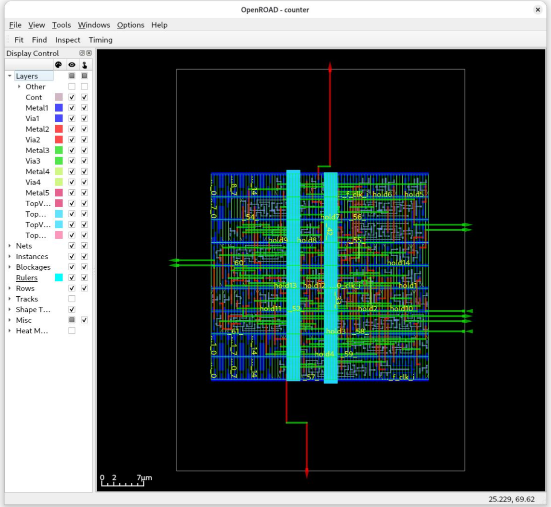
sky130A



gf180mcuD



ihp-sg13g2



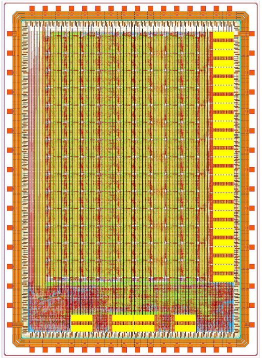
Advanced Usage

- Customizing Flows
    - From the config file
    - From Python
- Custom Steps
- Plugin system: Extensibility
    - “Back at Efabless”
        - DFT plugin
        - Synopsys plugin
    - Greyhound
        - FABulous plugin

LibreLane Success Stories

Greyhound

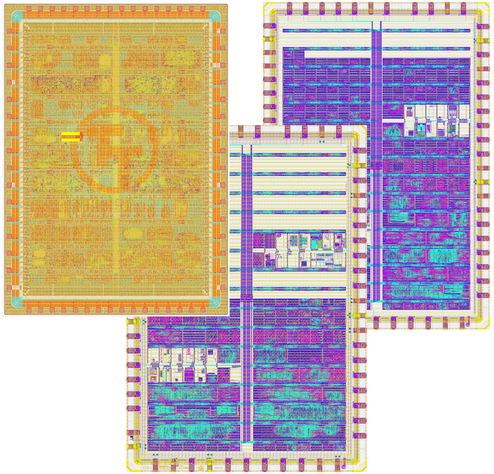
ihp-sg13g2



https://github.com/mole99/greyhound-ihp

Tiny Tapeout

sky130 & ihp-sg13g2



https://tinytapeout.com

Frigate

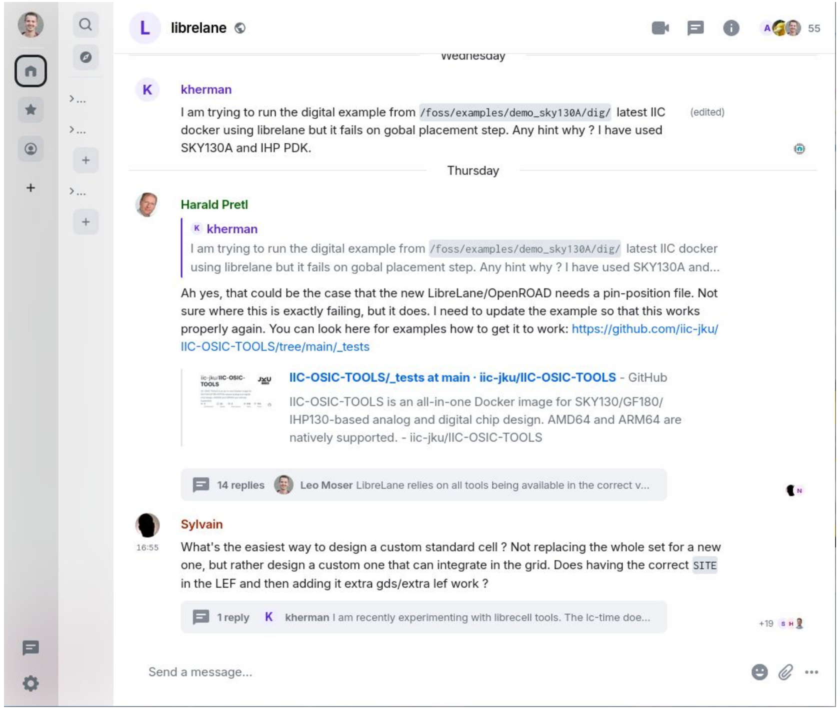
sky130

…and many more

Looking to the Future (Community-driven)

- Main development @ AUC Open Hardware Lab
  - Same primary authors, basically
- Legal stewardship of the FOSSi Foundation
  - Also handling hosting the repositories and binaries
- **Aim:** Community-driven development
  - Many individuals and companies contributing to and benefitting from the same tool
- More PDKs! More plugins! More tools!

Community



Wednesday

kherman

I am trying to run the digital example from `/foss/examples/demo_sky130A/dig/` latest IIC docker using librelane but it fails on gobal placement step. Any hint why ? I have used SKY130A and IHP PDK. (edited)

Thursday

Harald Pretl

> kherman
> I am trying to run the digital example from `/foss/examples/demo_sky130A/dig/` latest IIC docker using librelane but it fails on gobal placement step. Any hint why ? I have used SKY130A and...

Ah yes, that could be the case that the new LibreLane/OpenROAD needs a pin-position file. Not sure where this is exactly failing, but it does. I need to update the example so that this works properly again. You can look here for examples how to get it to work: https://github.com/iic-jku/IIC-OSIC-TOOLS/tree/main/_tests

IIC-OSIC-TOOLS/_tests at main · iic-jku/IIC-OSIC-TOOLS - GitHub

IIC-OSIC-TOOLS is an all-in-one Docker image for SKY130/GF180/IHP130-based analog and digital chip design. AMD64 and ARM64 are natively supported. - iic-jku/IIC-OSIC-TOOLS

14 replies **Leo Moser** LibreLane relies on all tools being available in the correct v...

Sylvain 16:55

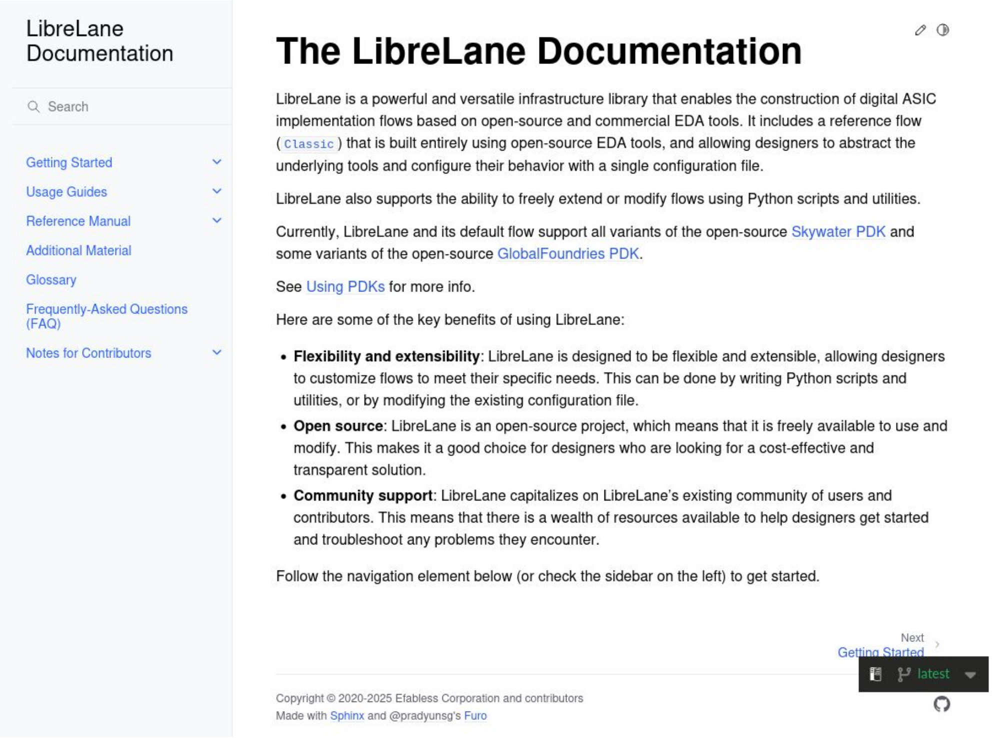
What's the easiest way to design a custom standard cell ? Not replacing the whole set for a new one, but rather design a custom one that can integrate in the grid. Does having the correct `SITE` in the LEF and then adding it extra gds/extra lef work ?

1 reply K **kherman** I am recently experimenting with librecell tools. The lc-time doe...

- **FOSSi Chat**
  - Community for Free and Open Source Silicon
  - Hosted and administered by the FOSSi Foundation
  - https://fossi-chat.org

The FOSSi Foundation.

Documentation



The LibreLane Documentation

LibreLane is a powerful and versatile infrastructure library that enables the construction of digital ASIC implementation flows based on open-source and commercial EDA tools. It includes a reference flow (`Classic`) that is built entirely using open-source EDA tools, and allowing designers to abstract the underlying tools and configure their behavior with a single configuration file.

LibreLane also supports the ability to freely extend or modify flows using Python scripts and utilities.

Currently, LibreLane and its default flow support all variants of the open-source Skywater PDK and some variants of the open-source GlobalFoundries PDK.

See Using PDKs for more info.

Here are some of the key benefits of using LibreLane:

- **Flexibility and extensibility**: LibreLane is designed to be flexible and extensible, allowing designers to customize flows to meet their specific needs. This can be done by writing Python scripts and utilities, or by modifying the existing configuration file.
- **Open source**: LibreLane is an open-source project, which means that it is freely available to use and modify. This makes it a good choice for designers who are looking for a cost-effective and transparent solution.
- **Community support**: LibreLane capitalizes on LibreLane's existing community of users and contributors. This means that there is a wealth of resources available to help designers get started and troubleshoot any problems they encounter.

Follow the navigation element below (or check the sidebar on the left) to get started.

Next
Getting Started

Made with Sphinx and @pradyunsg's Furo

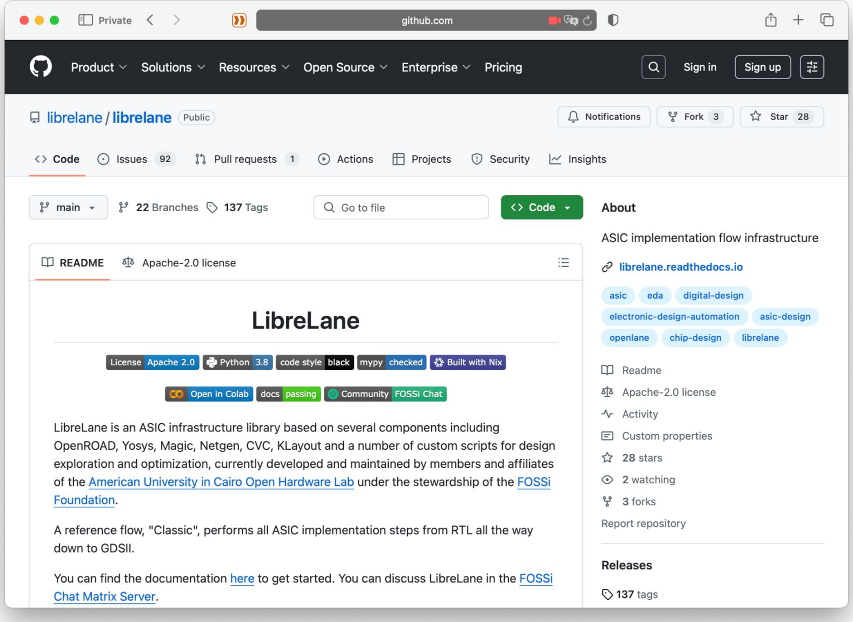
https://librelane.readthedocs.io

- Nix and Docker based installation
- Usage guides
- Full flow, steps and variables documentation
- LibreLane API

Thank you!

Give us a star on GitHub!

github.com/librelane/librelane



LibreLane

LibreLane is an ASIC infrastructure library based on several components including OpenROAD, Yosys, Magic, Netgen, CVC, KLayout and a number of custom scripts for design exploration and optimization, currently developed and maintained by members and affiliates of the American University in Cairo Open Hardware Lab under the stewardship of the FOSSi Foundation.

A reference flow, "Classic", performs all ASIC implementation steps from RTL all the way down to GDSII.

You can find the documentation here to get started. You can discuss LibreLane in the FOSSi Chat Matrix Server.

About

ASIC implementation flow infrastructure

librelane.readthedocs.io