LibreLane

Leo Moser, HeiChips 2026, Heidelberg

What is LibreLane?

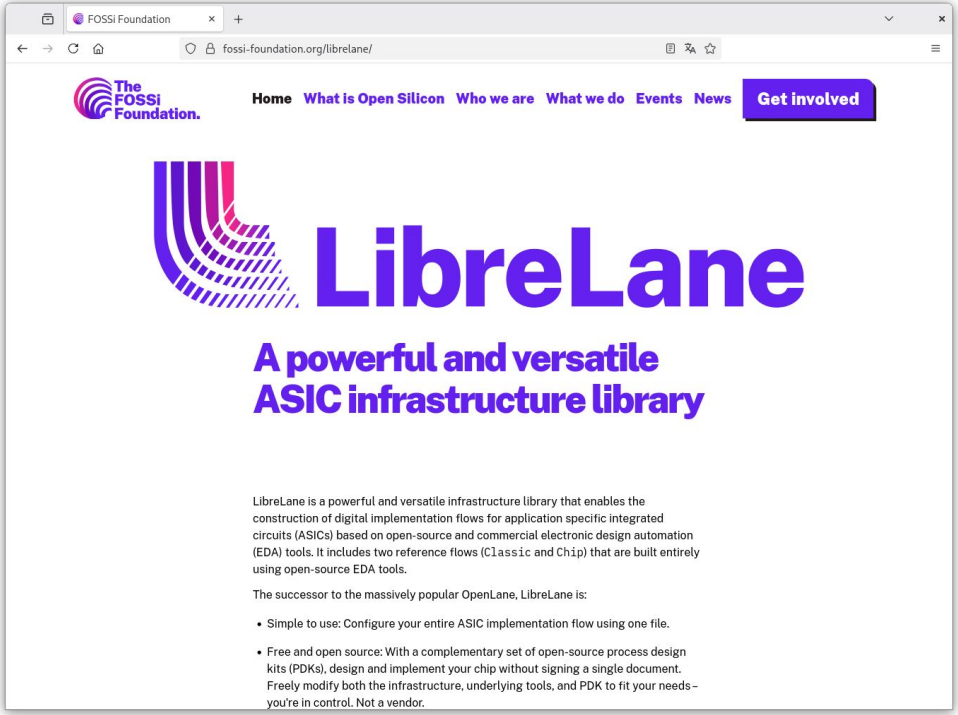
- A ground-up rewrite of OpenLane
  - Started at Efabless as “OpenLane 2”
  - (Mostly) backwards-compatible with OpenLane 1
- Fully customizable **Python**-based Flow
- All tools included and ready to download and run:
  - Natively using a package manager called Nix
  - Containerized using Docker
  - 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

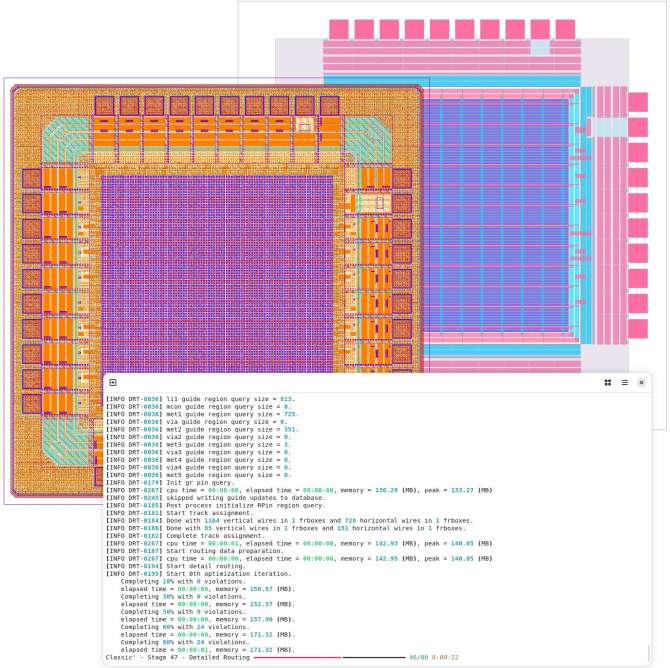
Brand New Website!



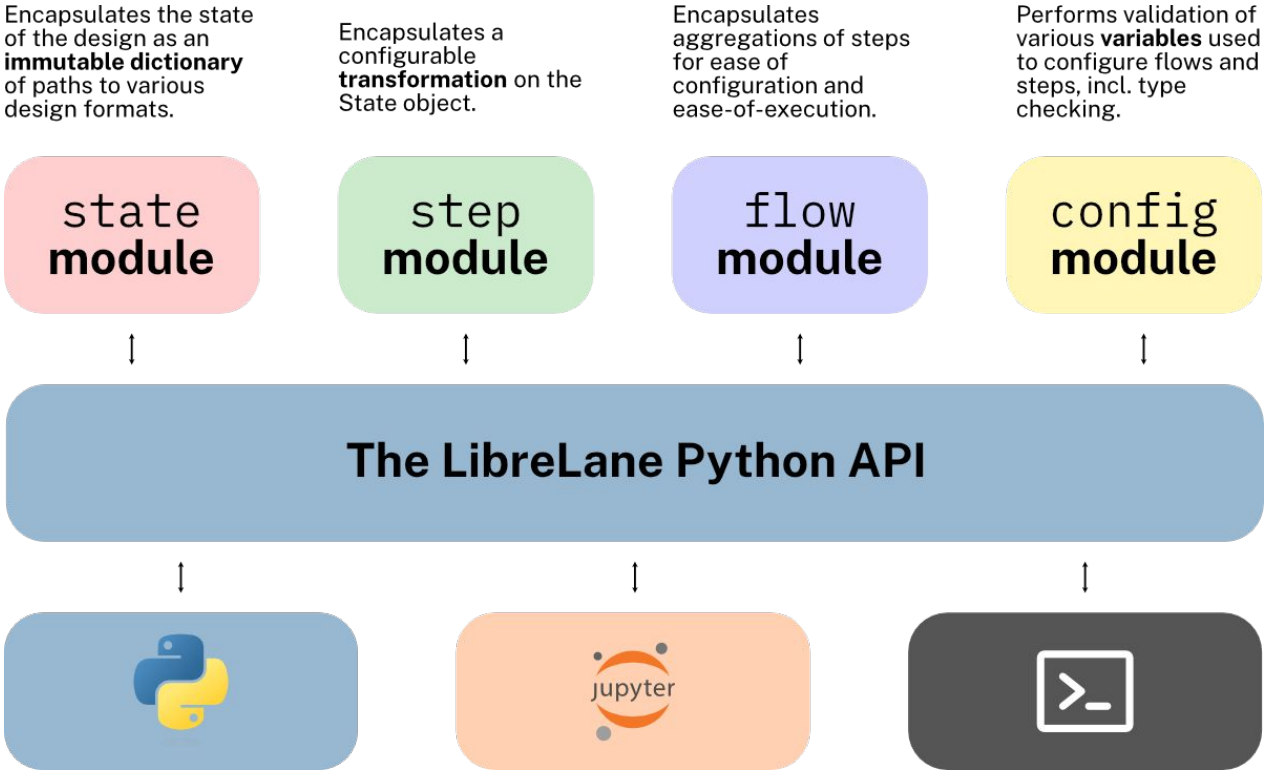
https://librelane.org

LibreLane 3.0 Features

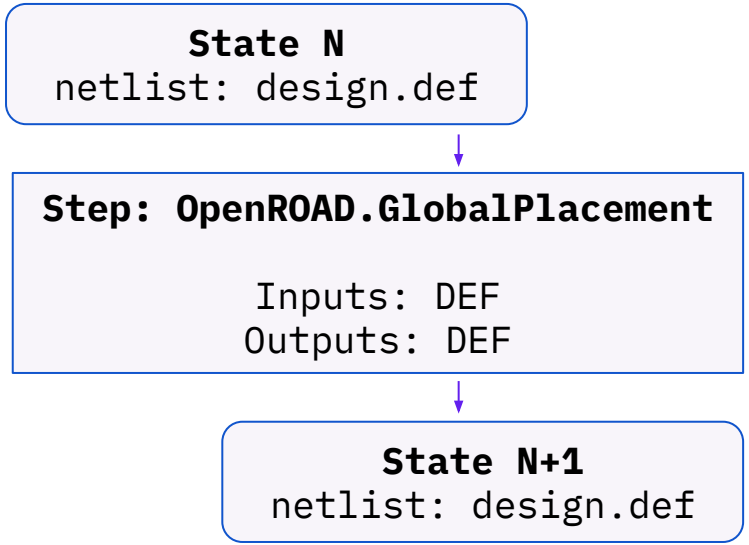
- The `Chip` Flow
  - New steps and full chip templates
- Custom Flow Updates
- Improved Plugin Support
  - Difetto DFT Plugin
  - FABulous (e)FPGA Plugin
- Misc. LibreLane Improvements



Architectural Overview



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

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

CLI

```
$ librelane config.yaml
```

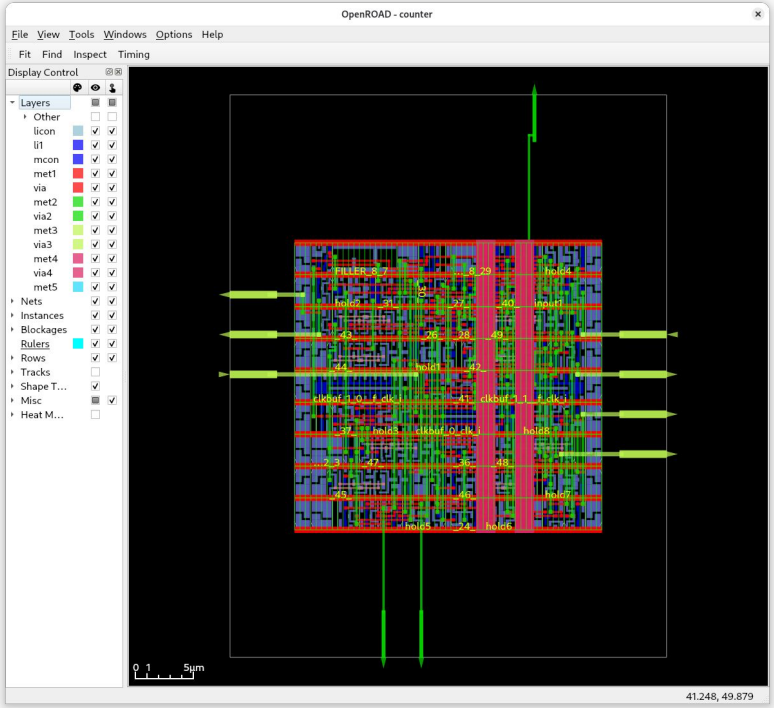
counter.sv

```
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

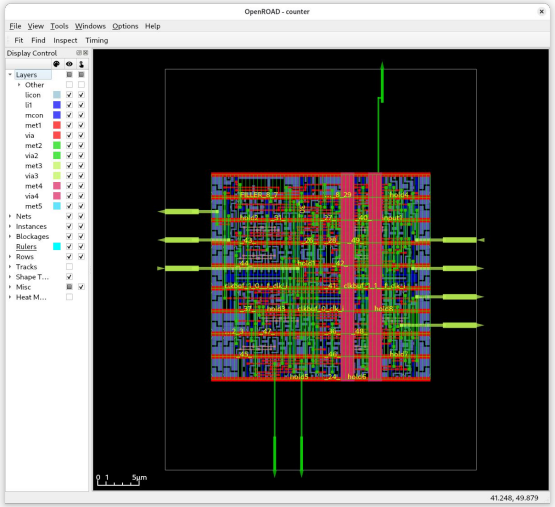
leo@debian-pc: ~/Repositories/librelane-merge

```
[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
```

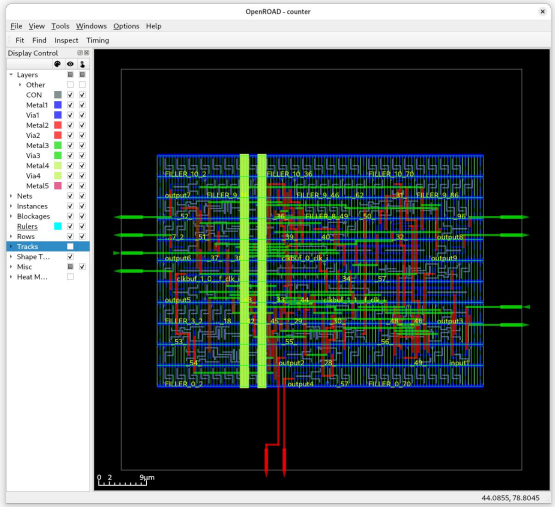


Officially Supported PDKs

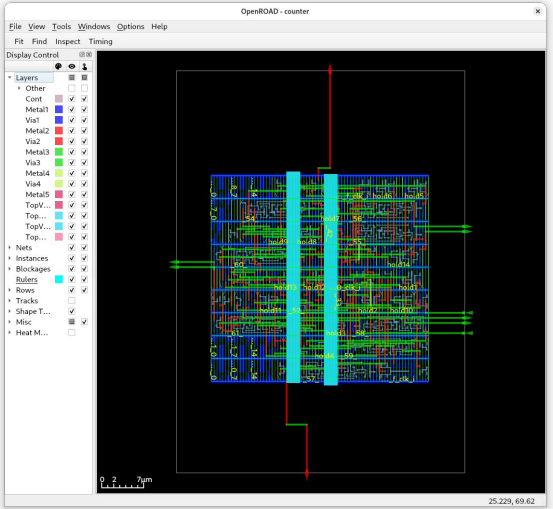
sky130A



gf180mcuD

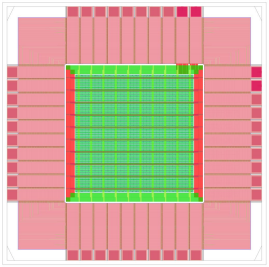
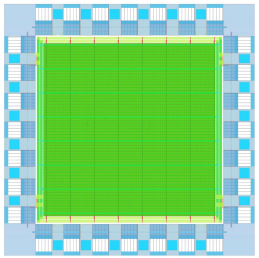
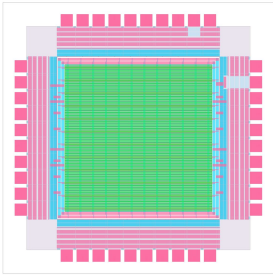


ihp-sg13g2



Officially Supported PDKs

- Open source PDKs
  - sky130
  - gf180mcu
  - ihp-sg13g2
- LibreLane templates:
  - full-chip template from IHP (ihp-sg13g2)
  - full-chip templatefrom wafer.space (gf180mcuD)
- Plugin Support
  - Difetto DFT Plugin
  - FABulous (e)FPGA Plugin

LibreLane

- Website

  https://librelane.org
- Repo

  https://github.com/librelane/librelane
- Docs

  https://librelane.readthedocs.io
- Chat

  https://matrix.to/#/#librelane:fossi-chat.org

Thank You!

https://librelane.org