

Precision Innovations Inc.

www.precisioninno.com

DAC 2024

Tom Spyrou

CEO

tspyrou@precisioninno.com



PII products and solutions based on OpenROAD provide Application Specific Integrated Circuit (ASIC) and System on Chip (SoC) design teams with an open source, no-human-in-loop, 24-hour place and route solution. It was initially developed with funding from DARPA under the Intelligent Design of Electronic Assets (IDEA) program. Precision Innovations is the primary industrial developer and is commercializing this technology.

Precision Innovations Overview

Company/Team Overview

- Principal industrial developer and integrator of the OpenROAD RTL to GDS ASIC/SoC Development solution
- Founded 2019
- Electronic Design Automation (EDA) veterans – 20 employees / contractors
- HQ – San Diego, CA.

History

- DARPA award for OpenROAD
- Now funded by users purchasing professional support and funding specific development

Technology overview

- OpenROAD is an EDA solution for hardware implementation of ASIC and SoC chips
- Supports RTL to GDS on designs down to 12nm
- A new popular use model is fast estimation / new design feasibility analysis on any node
- Professional Support Solutions available

Why Professional Support

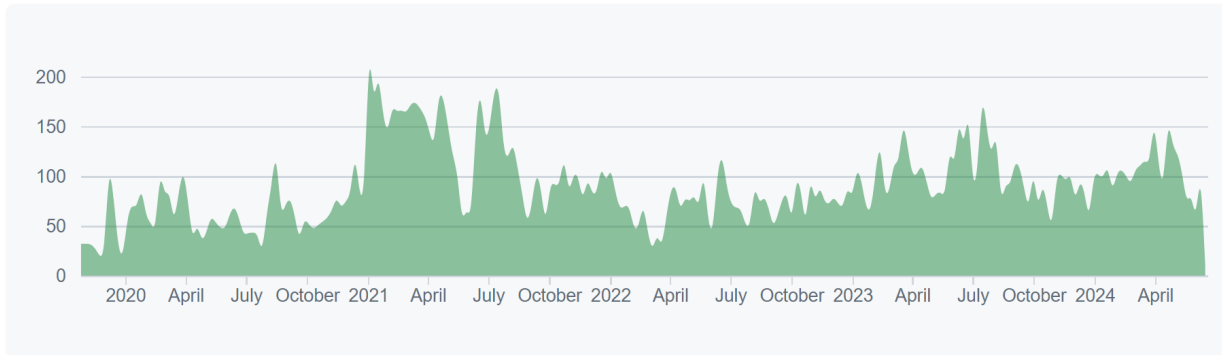
- Accelerated response time on issues
- Be-spoke feature development
- Secure regression testing on your design style

OpenROAD has gone viral with users and contributors!

- We have passed 20K commits!

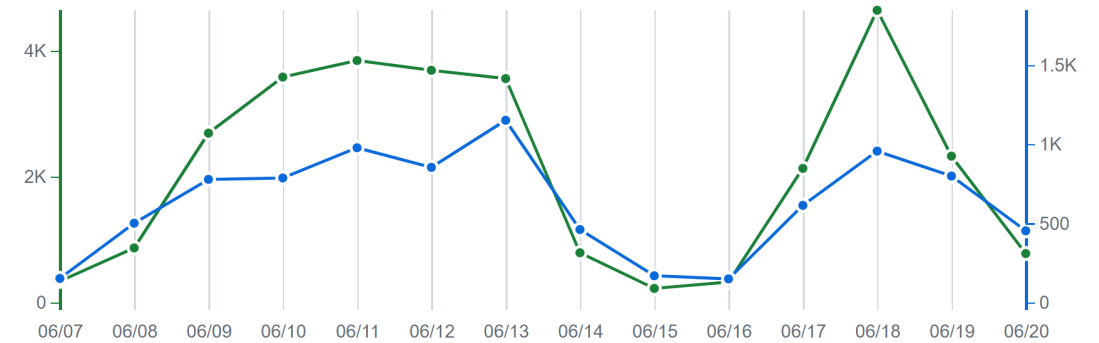
Oct 27, 2019 – Jun 20, 2024

Contributions to master, line counts have been omitted because commit count exceeds 10,000.



- OpenROAD is now used in too many places to track
- Users include large companies, small companies, researchers, and students of all ages
- Public documentation and the ability to use search engines to solve problems is compelling

Git clones



29,739 Clones

7,107 Unique cloners

May 20, 2024 – June 20, 2024

Period: 1 month

Overview

85 Active pull requests

47 Active issues

74

Merged pull requests

11

Open pull requests

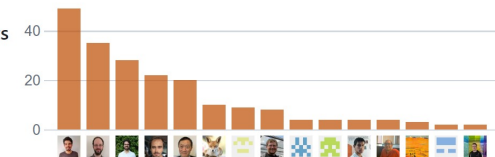
32

Closed issues

15

New issues

Excluding merges, **19 authors** have pushed **209 commits** to master and **209 commits** to all branches. On master, **276 files** have changed and there have been **13,600 additions** and **7,558 deletions**.



74 Pull requests merged by 19 people

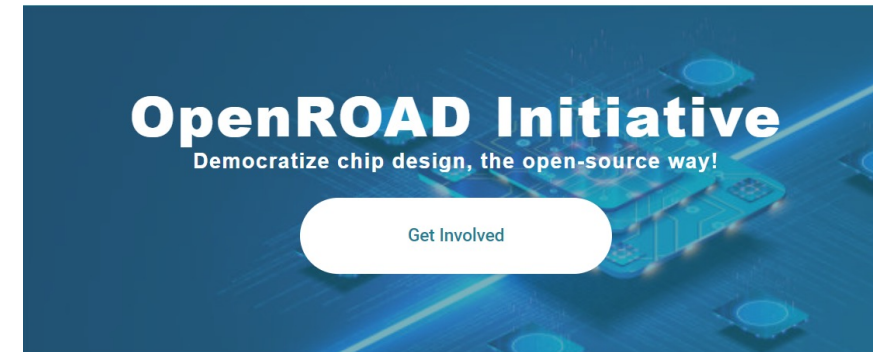
The OpenROAD Initiative 501(c)(3) Nonprofit

- What is The OpenROAD Initiative www.openroadinitiative.org
 - A 501(c)(3) public charity for the benefit of open-source EDA and related technology
- Founders
 - Professor Andrew Kahng and Tom Spyrou
 - Founded Feb 2023
- Goals
 - Continued development of OpenROAD and all open-source EDA
 - Development of open source PDKs both manufacturable and Proxy
 - Creation of education material teaching chip design
 - Training of new Chip Designers and new EDA developers on the full flow
 - Open issue discussion which is internet searchable encouraging new participation
- Who would contribute
 - Any organization interested in the continued development of the open design ecosystem

The OpenROAD Initiative 501(c)(3) Nonprofit

- We see a parallel between Linux and Open-Source EDA
- The OpenROAD initiative would like to be the shepherd of Open-Source EDA
- Commercial companies like PII need to exist around the open-source code to provide professional services and high-quality releases
- When PII has several competitors, that will mean that the effort is successful
- There needs to be a neutral entity to control the technical direction
- This includes trusted developers who can accept merges and ensure architectural integrity
- We are now actively looking for funding and for the initial partners
- These are important decision which will take time
- Partners will come from companies which are heavily using open source EDA

<https://openroadinitiative.org/>



Exploring LLMs in OpenROAD to Enhance User Experience and Foster Collaboration

GSOC OpenROAD 2024 LLM Projects
OR Assistant: Leveraging Open EDA Datasets

An OpenROAD internship project by **Aviral Kaintura** and
Palaniappan R

- Enhance user experience in OpenROAD by leveraging LLMs for common questions and problems
- Contribute towards a common dataset for OpenROAD based dataset for research in AI-based, open-source chip design
- **OR Assistant focuses on solving common and frequently occurring problems**
 - **Installation and Troubleshooting Assistance:** Provide quick and accurate solutions to common installation issues and troubleshooting steps.
 - **Easy Access to Existing Resources:** Summarize relevant information from datasets based on OpenROAD documentation, man pages, error logger and online resources to provide concise responses to queries

OR Assistant

How can I do macro placement?

To perform macro placement, use the following syntax:

```
macro_placement
  [-halo {halo_x halo_y}]
  [-channel {channel_x channel_y}]
  [-fence_region {lx ly ux uy}]
  [-snap_layer snap_layer_number]
  [-style corner_wax_wl|corner_min_wl]
```

- **-halo**: Specifies the keep out distance from the macro, in the X and Y directions, to the standard cell row.
- **-channel**: Specifies the channel width between macros, in the X and Y directions.
- **-fence_region**: Specifies the fence region, in the form of `lx ly ux uy`, where `(lx, ly)` is the lower left corner and `(ux, uy)` is the upper right corner.
- **-snap_layer**: Specifies the snap layer number.
- **-style**: Specifies the placement style, which can be either `corner_wax_wl` or `corner_min_wl`.

Sources: <https://openroad-flow-scripts.readthedocs.io/en/latest/user/FlowVariables.html>
<https://openroad.readthedocs.io/en/latest/main/src/mpl/README.html> <https://openroad-flow-scripts.readthedocs.io/en/latest/tutorials/FlowTutorial.html> <https://openroad.readthedocs.io/en/latest/main/src/mpl2/README.html>

Help me install OpenROAD-flow-scripts

Installing OpenROAD-flow-scripts

Option 1: Using Docker

1. Install Docker for an efficient user experience.
2. Install OpenROAD-flow-scripts using Docker as follows:

```
docker run --rm -it -u $(id -u ${USER}):$(id -g ${USER}) -v $(pwd)/flow:/Op
```

Option 2: Installing Dependencies and Building Manually

1. Install dependencies by following the helper script:

```
sudo ./etc/DependencyInstaller.sh
```

Note: This requires sudo access and may affect your environment.

2. Alternatively, install dependencies locally using setup.sh from OpenROAD-flow-scripts:

```
git clone --recursive https://github.com/The-OpenROAD-Project/OpenROAD-flow
cd OpenROAD-flow-scripts
sudo ./setup.sh
```

3. Build OpenROAD-flow-scripts:

```
./build_openroad.sh --local
```

4. Verify installation by cloning the OpenROAD-flow-scripts repository non-recursively:


```
git clone https://github.com/The-OpenROAD-Project/OpenROAD-flow-scripts.git
```


5. Export path variables accordingly.

Option 3: Using Prebuilt Binaries

1. Download prebuilt binaries from Precision Innovations Github releases for your distribution.
2. Run the install command based on your platform (e.g., for Ubuntu 20.04):

```
sudo apt install ./openroad_2.0_amd64-ubuntu20.04.deb
```


 How can I specify I/O pin layout?

 You can use the `set_io_pin_constraint` command to specify I/O pin layout.


This command sets region constraints for pins according to the pin direction or the pin name.


You can use the `-direction` option to specify the pin direction, the `-pin_names` option to specify a list of pin names, the `-region` option to specify the region where the pins should be placed, and the `-mirrored_pins` option to specify a list of pins that will be symmetrically placed in the vertical or the horizontal edges.

Additionally, you can use the `-group` and `-order` options to specify that the pins should be placed together and in ascending x/y position, respectively.

Sources: 

Feedback

 What does the `-switch_cell` argument in PDN do while doing power switch insertion?

 The `-switch_cell` argument in PDN defines the name of the coarse grain power switch cell to be used wherever the `stdcell` rail connects to the rest of the power grid. The mesh layers are associated with the unswitched power net of the voltage domain, whereas the `stdcell` rail is associated with the switched power net of the voltage domain. The placement of a power switch cell connects the unswitched power mesh to the switched power rail through a power switch defined by the `define_power_switch_cell` command.

Check out our prototype here,
<http://34.16.183.224:8503/>

Collaboration: Fostering OpenROAD-based AI

- Collaborate with NYU/ ASU
 - EDA Corpus : <https://github.com/OpenROAD-Assistant/EDA-Corpus>
 - Contribute and test a common dataset
 - Foster LLM-based research using OpenROAD
 - Integrate OR Assistant to common interfaces for a better user experience
 - GUI, GH etc
 - Expand flow-orchestration and generation scripts
 - Autotuning, ORFS custom flows
 - The Possibilities are endless!!
-
- Open-source tools are ideal for training new EDA tool users
 - Today's new software users expect to be able to internet search for answers not use proprietary manuals and interfaces which are cumbersome
 - Training LLMs to answer questions for new users in the open will foster much more learning
 - This is **not possible** with the current restrictions on closed source tools