

ece 260c.

Intro, Logistics and OpenROAD

Prof. Andrew Kahng

ECE 260C A00, SP25: “VLSI Special Topics”

- This version of 260C will explore emerging open-source VLSI (digital, RTL-to-GDS) implementation software, giving you a chance to understand the internals of the tools you use.
- **Course materials will center on the OpenROAD tool and the IHP130 open-source PDK.**

UC San Diego TODAY

RESEARCH PEOPLE CAMPUS & COMMUNITY ARTS & CULTURE VOICES

Open-source Semiconductor Chip Design Tool Celebrates Success

UC San Diego-led OpenROAD project democratizes silicon innovation for students and chip designers alike



A team led by engineers at UC San Diego successfully created and made available an open-source tool that is lowering barriers of access and democratizing silicon innovation worldwide.

October 16, 2024

In 2018, a team of the nation's leading semiconductor experts from industry and

Story by:
Katherine Connor -
kconnor@ucsd.edu

Scope

- Explore the expanding field of Open-Source VLSI CAD Software
 - Learn how to use OpenROAD and Yosys as an alternative to commercial software like Cadence Innovus
- Learn how to apply complex scripting to achieve better results
 - Take advantage of open-source by making deeper in-source changes
 - Build design space exploration flows commonly demanded in industry
- Understand the internals and algorithms that drive implementation
- Apply learned methodologies in the context of System-on-Chip design
- **Note: This is a brand-new class (!) with assumed prerequisites (!)**

Logistics

- Lectures are Tues/Thurs from 5:00 pm to 6:20 pm in Center Hall 216
 - Attendance is Mandatory. At the end of the lecture, you must complete Lecture Participation in Canvas > Quizzes. This is 20% of your grade. Regrades available.
 - The topics and lecturer for each lecture can be seen on the Course Schedule
- There are 5 labs and a Final Project
 - Labs are 60% of your grade (12% each)
 - The Final Project is 20% of your grade.
- Stay up to date on the Schedule and check Canvas Announcements !!!
 - Canvas Home > Course Schedule
 - Office Hours are also on Canvas
 - **Course staff: Andrew Kahng (instructor), Bodhi Pramanik (TA), Davit Markarian (volunteer)**

Logistics Cont'd

- Labs are assigned at the end of lecture – Lab 0 today
 - Check the Course Schedule for due dates.
 - On the due date, they can be turned in as late as 2:59 AM with no penalty. After, no submissions will be accepted.
 - Regrades will be available.
- Your Final Project will be assigned in Week 5
 - In a small team, you will improve upon an existing SoC
 - More details will follow when assigned.
- This course has no final exam
 - Instead, the Final Project is due on the assigned exam date at midnight.
- Please review the Syllabus on Canvas

What is Open-Source EDA ?

- EDA tools that satisfy the definition of *open source*
 - [Tim Ansell, Open Source 101, 2019](#)
- Freely usable, freely modifiable, and shareable
- License permits free redistribution, creation of derived works, and use by anyone for any purpose, in a technology-neutral manner
 - Transformative! → Linux, Android, RISC-V, TensorFlow, ...
- Permissive licenses: BSD, MIT, Apache2.0 ...
- Share-alike / “copyleft”: GPL-2 or GPL-3 ...

What is **open source**?

Two parts

Industry standard definition:



Code released

under an

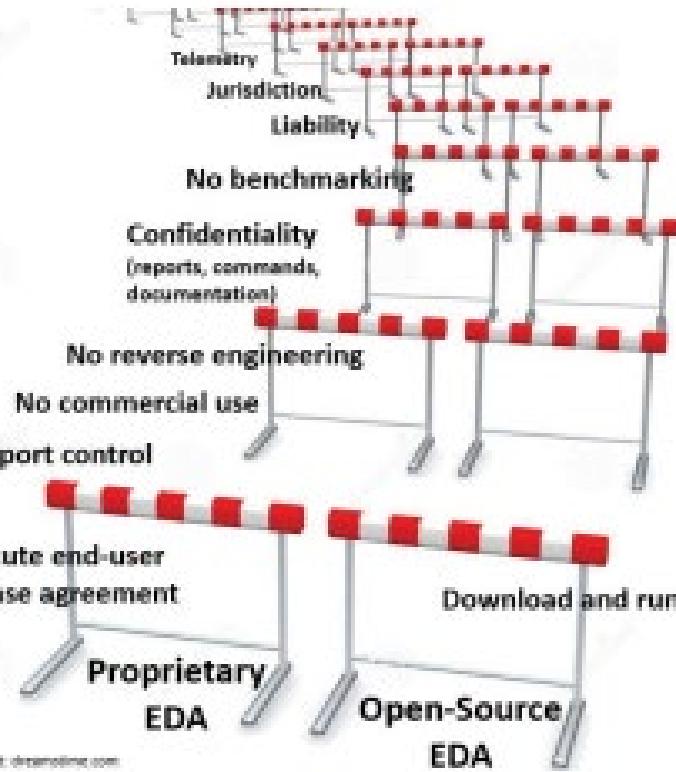


open source license

j.mp/eri19-foss101

Closed-Source EDA (e.g., in ECE 260B)

- Can you share your Tcl script with another user?
- Can you share code or write a tool that reads the same command syntax?
- Can you share the tool's output (gds, logfile, ...)
- Can you compare it with another tool ("benchmarking")?
- Can you use the tool output to make a chip startup?
- Can you share any of the tool's documentation?
- Can you upload the user guide or a logfile into ChatGPT?
- ... **(No! See the EULA that your university executed.)**



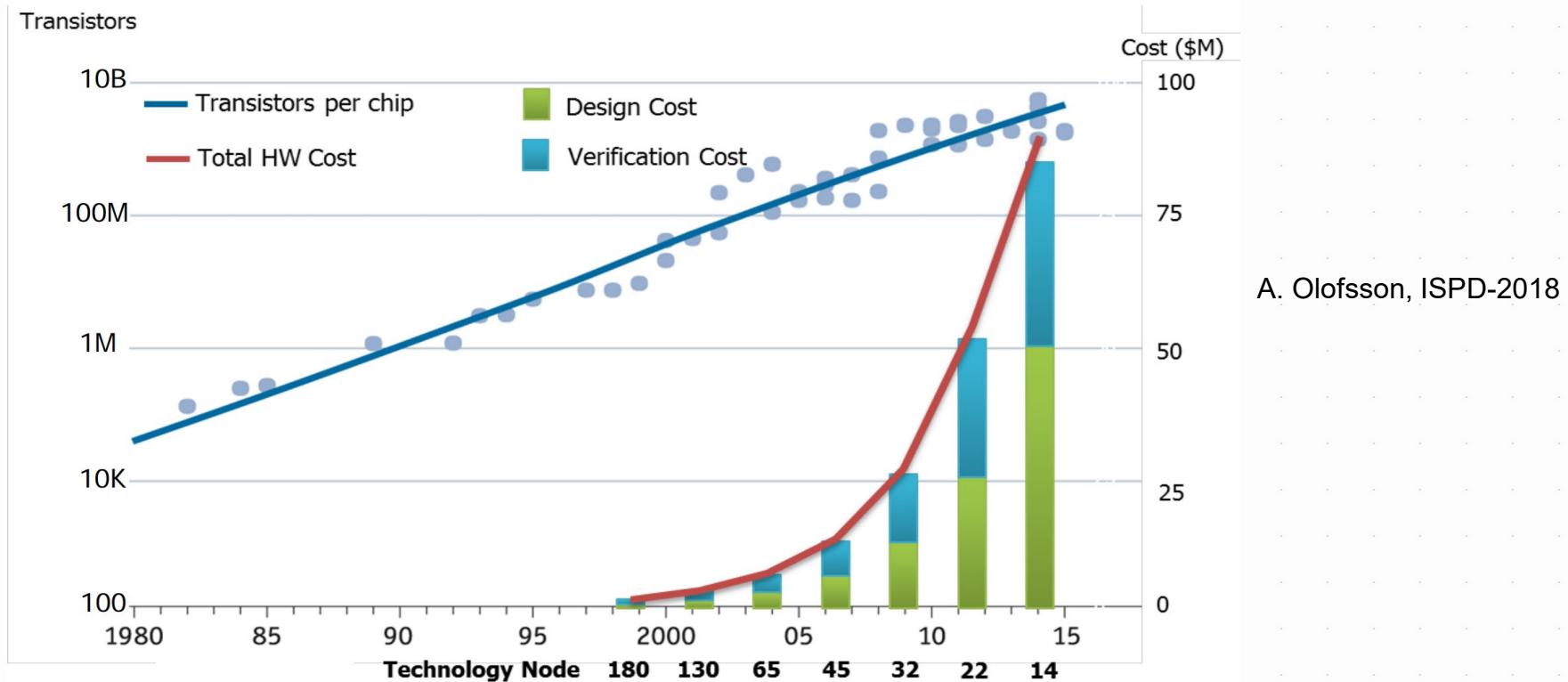
"A Mixed Open-Source and Proprietary EDA Commons for Education and Prototyping", ICCAD-2022. ([.pdf](#))

About OpenROAD

“Foundations and Realization of Open, Accessible Design”

The Crisis of Hardware Design

- ASIC design in advanced nodes: barriers of Cost, Expertise, Risk



- Innovators can't evaluate PPAC metrics of their design ideas

ASIC Design with Proprietary EDA: Need \$\$\$, Experts

- Very sophisticated tools with 1000's of commands
- Tool supplier focus: performance, power, area
- Large teams of expert users, many manual steps
- Long project schedules
- Significant project risks

OpenROAD: June 2018 – December 2023

- “Foundations and Realization of Open, Accessible Design”
- Funded by U.S. DARPA as part of the Electronics Resurgence Initiative (ERI). (UCSD = prime contractor)
- Mission: Democratize IC Design, Boost HW & EDA Innovation
 - Revitalize EDA
 - Contract: Deliver an Open-Source, RTL-to-GDS EDA system
 - 24-hour, no-human-in-loop, tapeout-clean layout in FinFET nodes



QUALCOMM arm

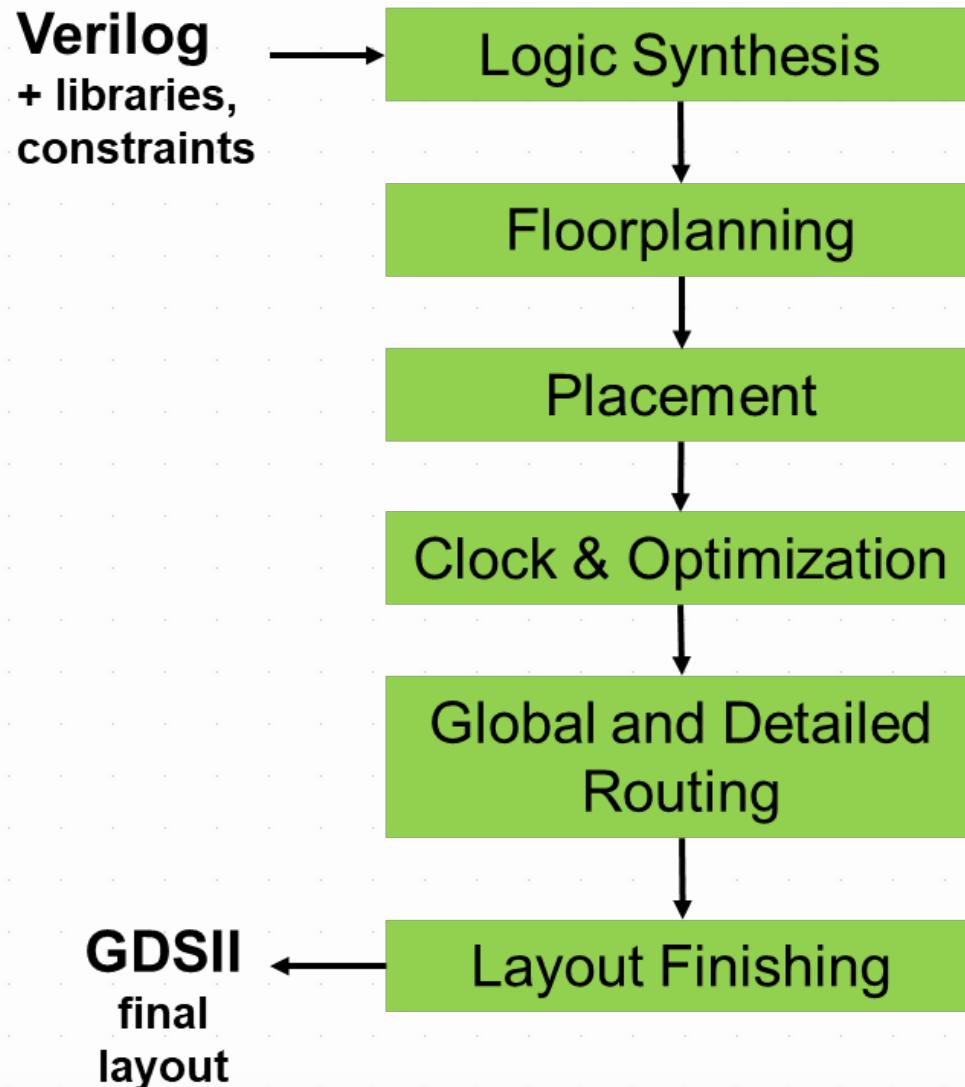


<https://precisioninno.com/>

OpenROAD: No Humans, 24 Hours

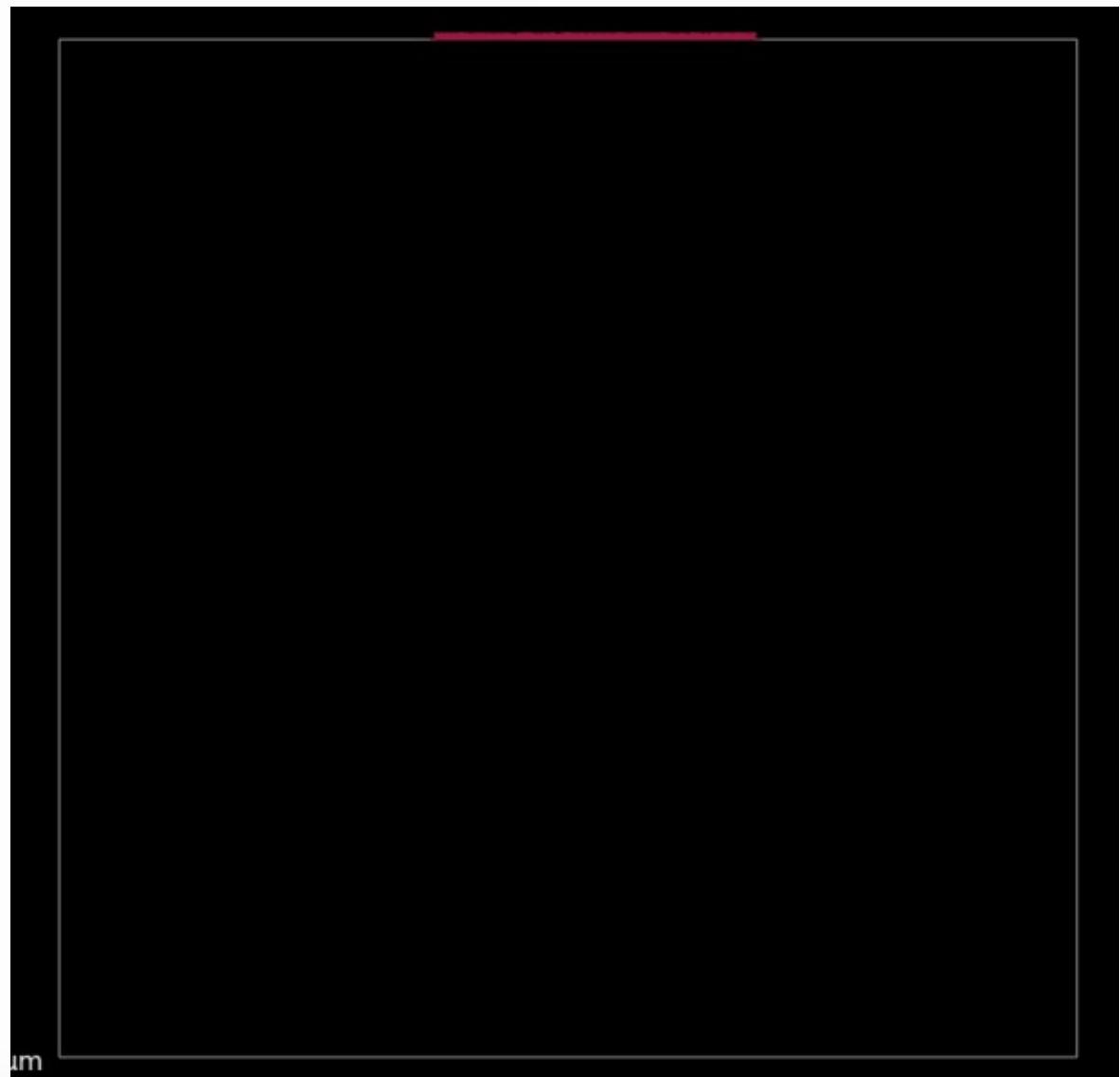
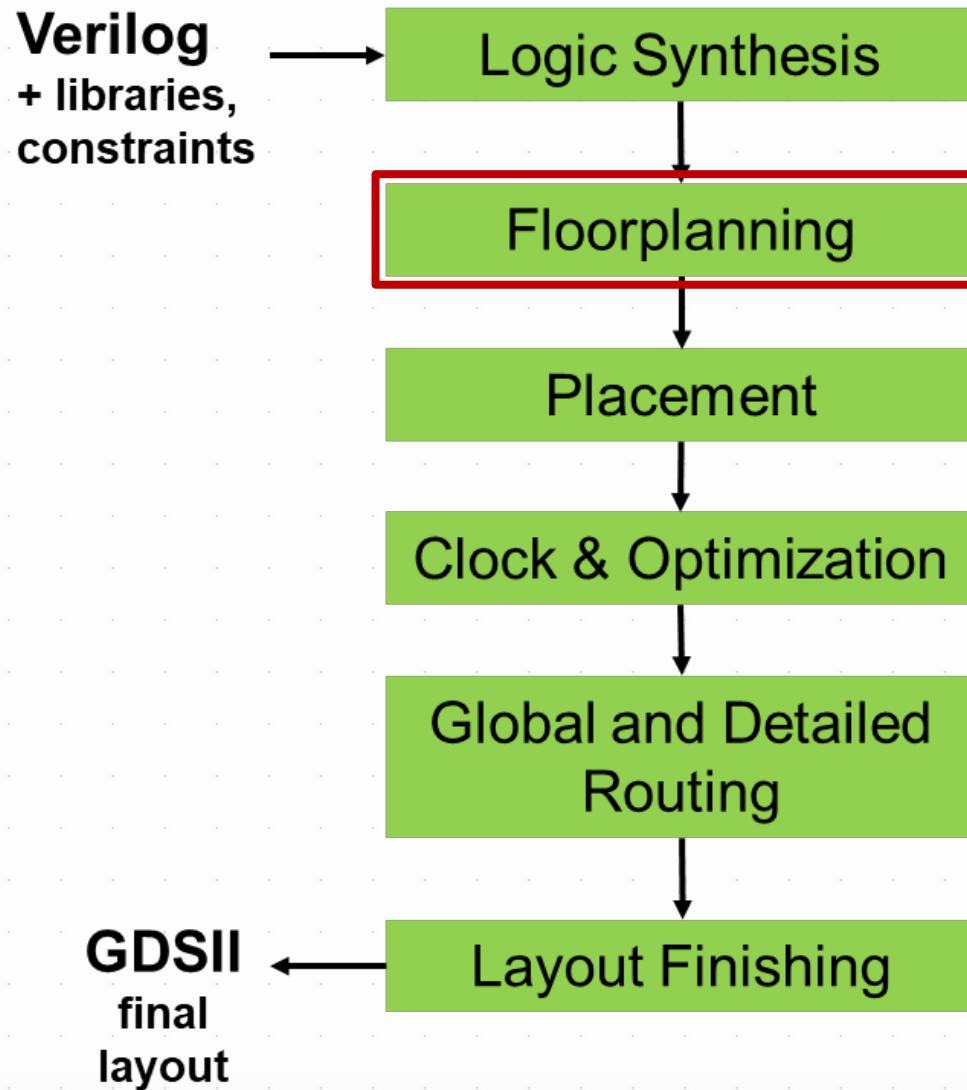
- FOCUS: Ease of use and runtime
- Directly attack the crises of design and innovation
 - Schedule barrier: **RTL-to-GDS** in 24 hours
 - Expertise barrier: No-human-in-loop, tapeout GDS
 - Cost barrier: Open source (and, runs in 24 hours)
- Unleash system innovation and design innovation
- Enable tool customization to system, application needs

RTL-to-GDS Chip Implementation Flow

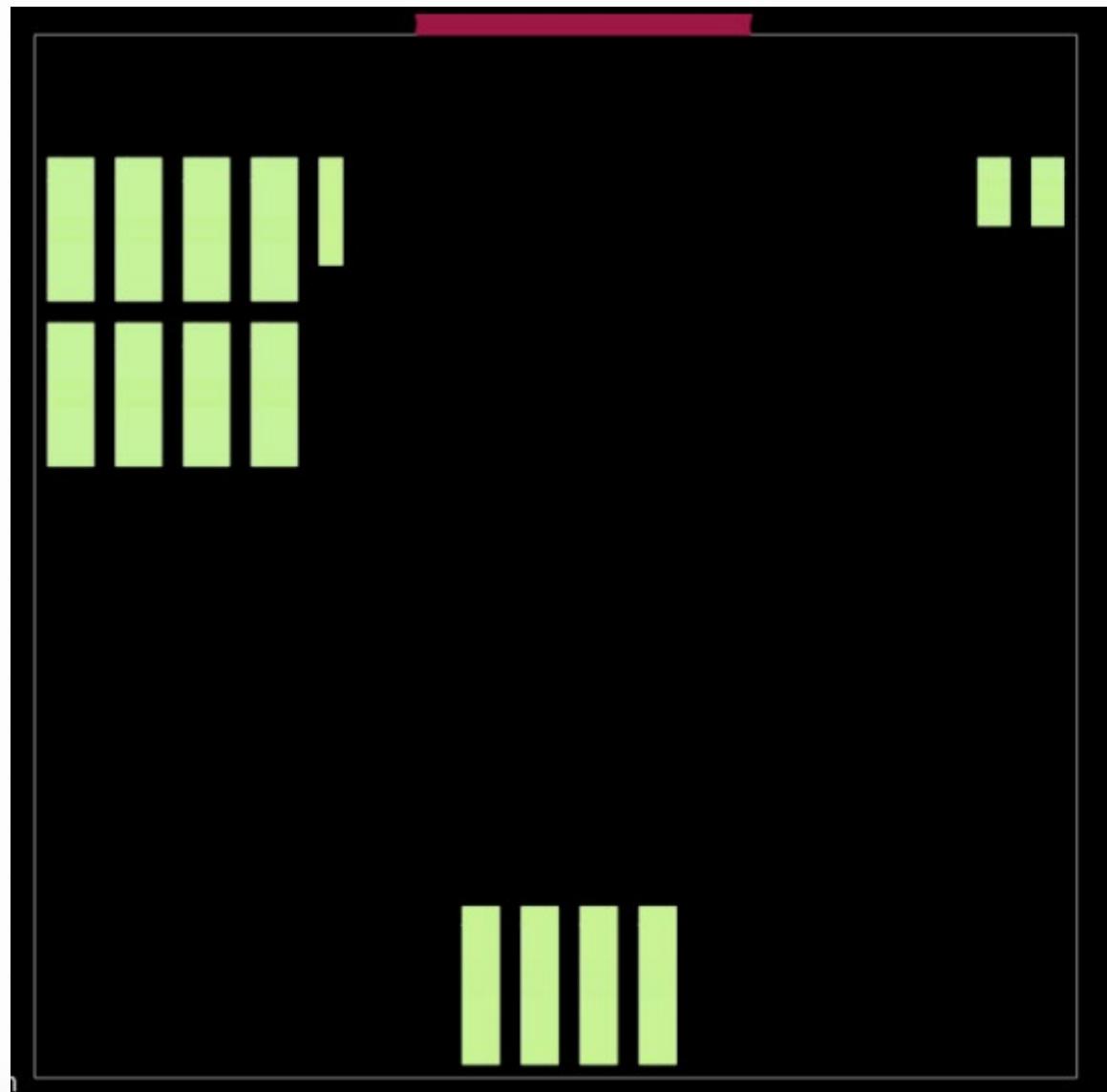
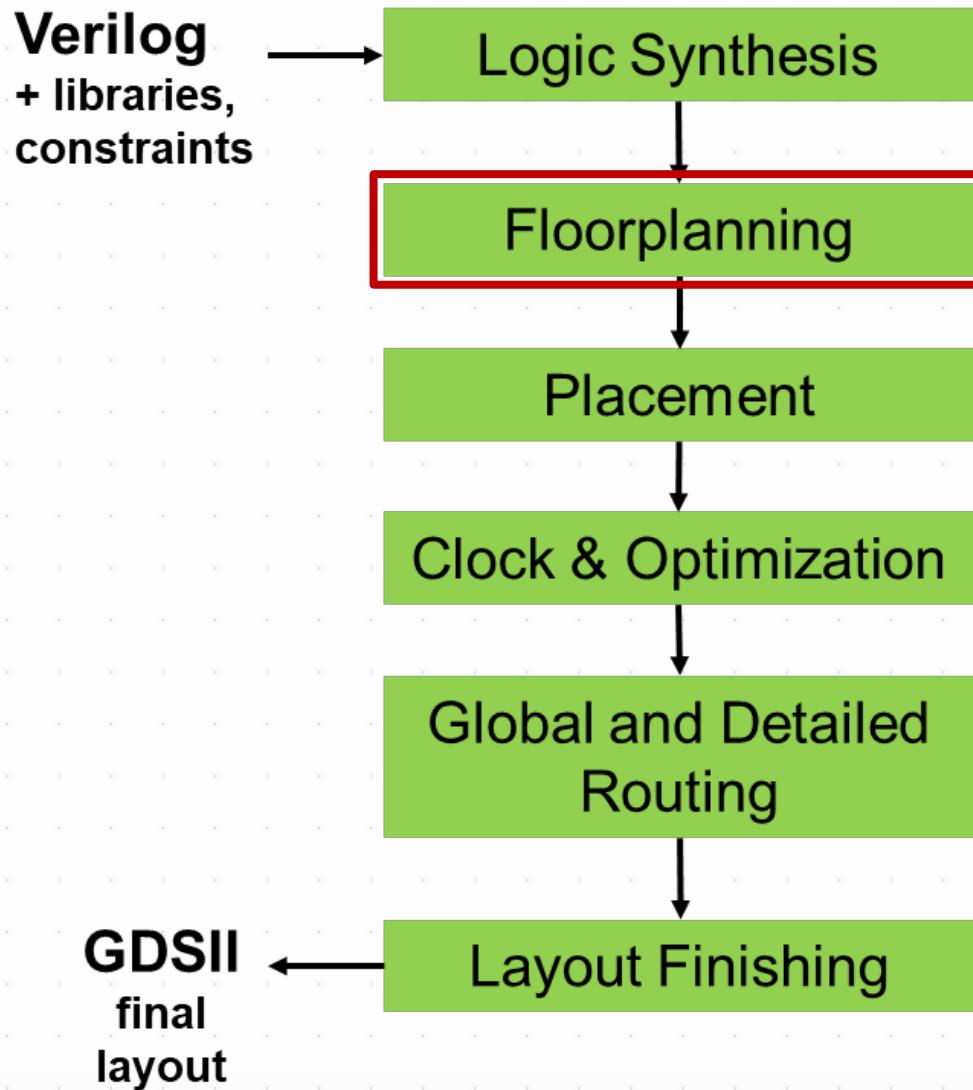


IO Placement

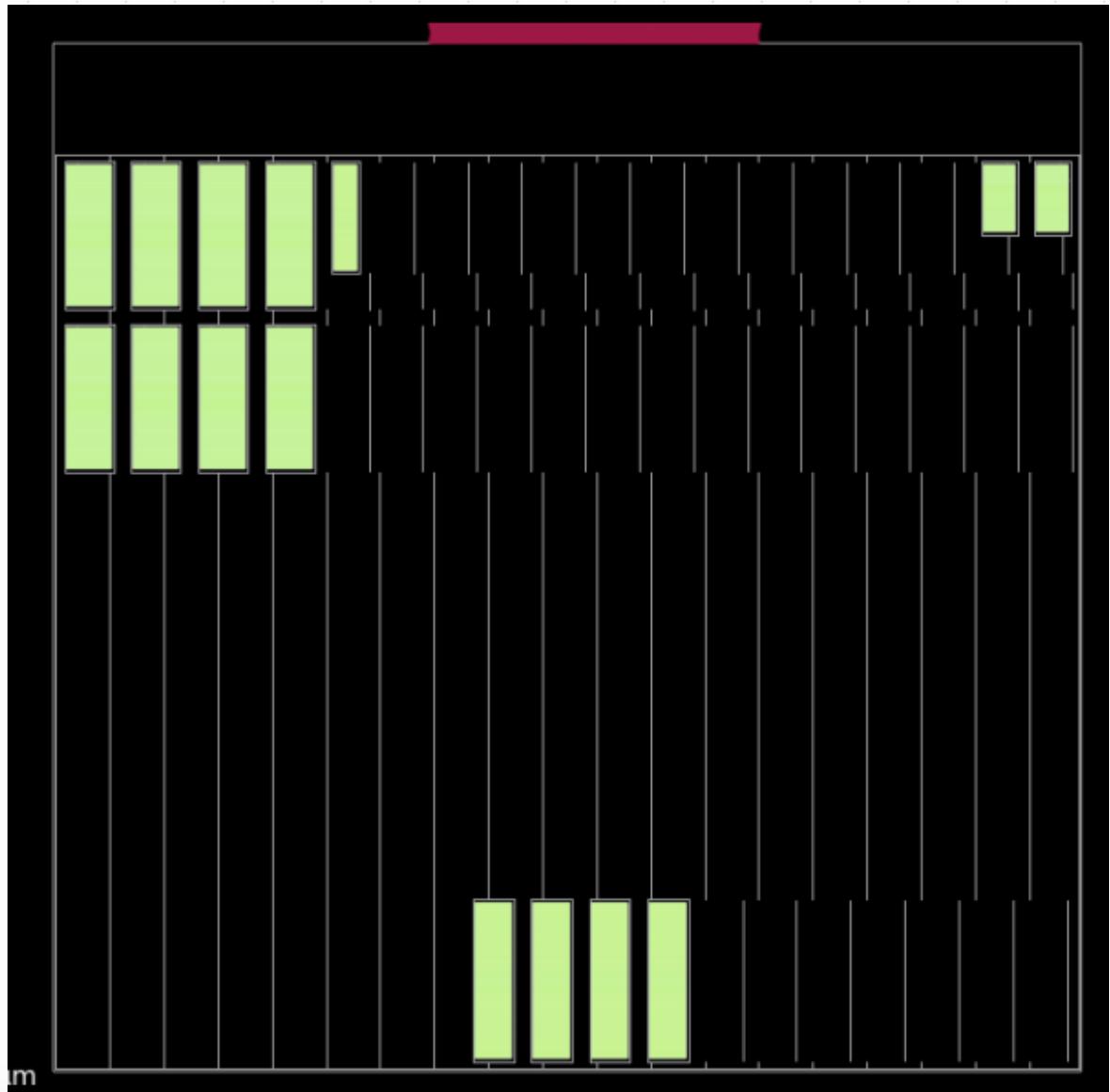
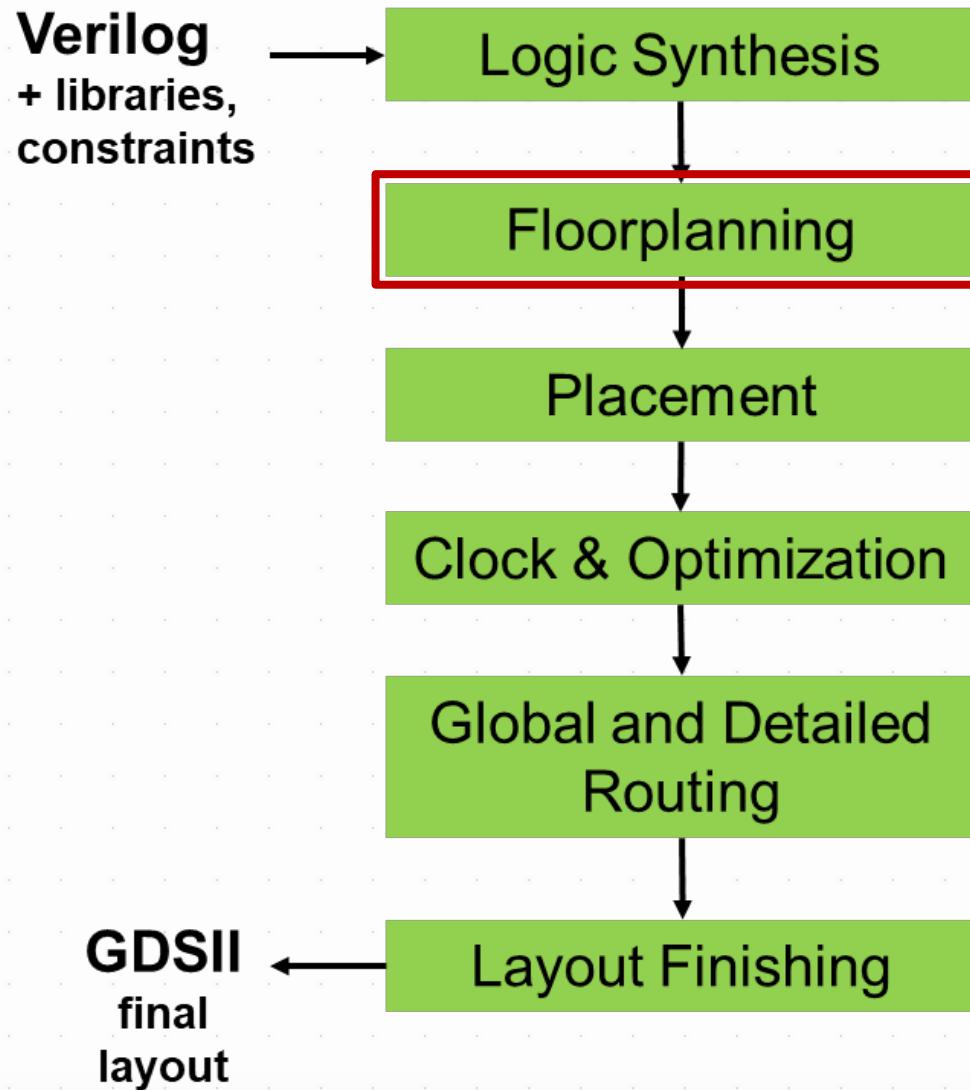
(Example: foundry 12nm RISC core, “coyote”)



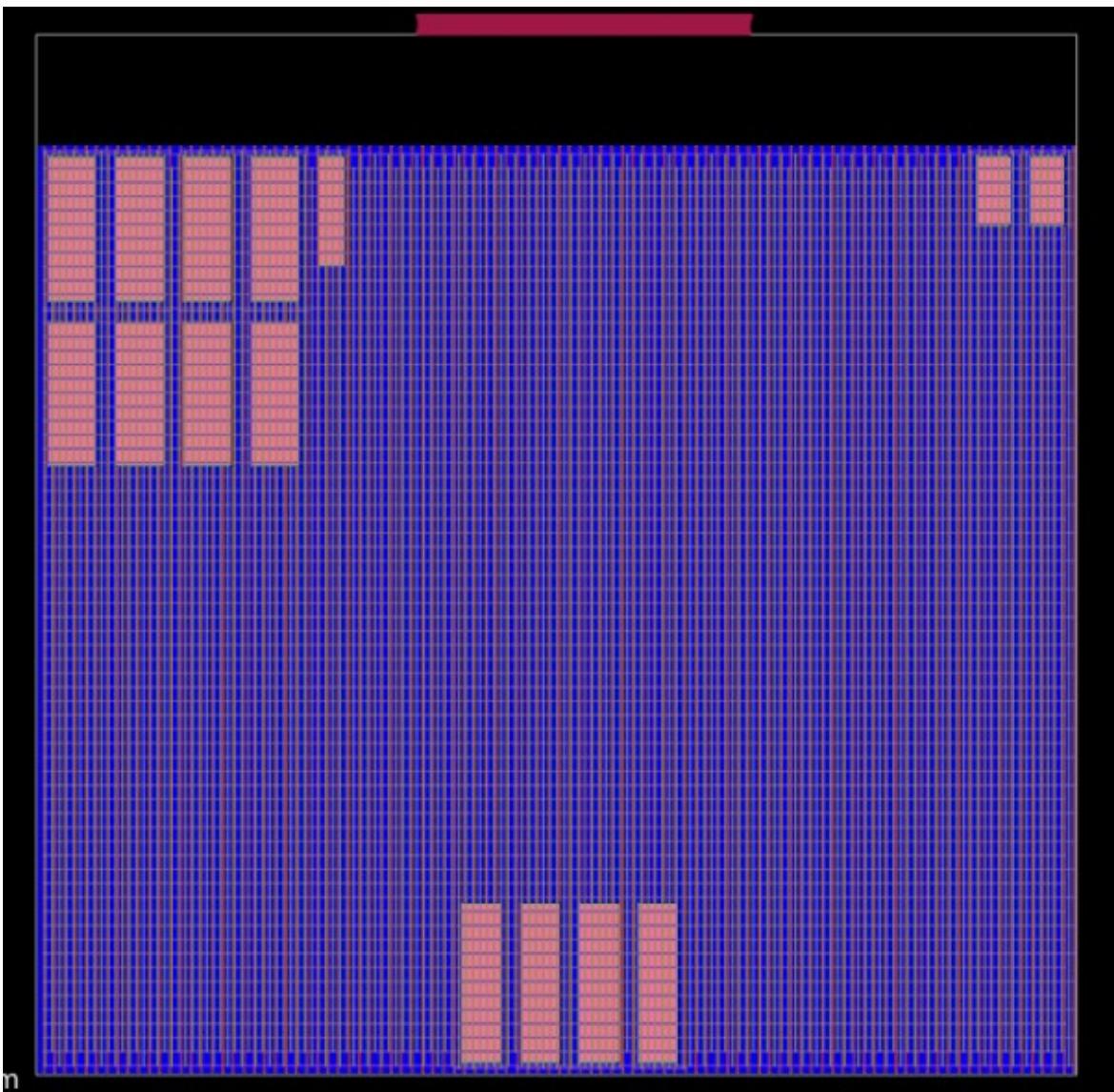
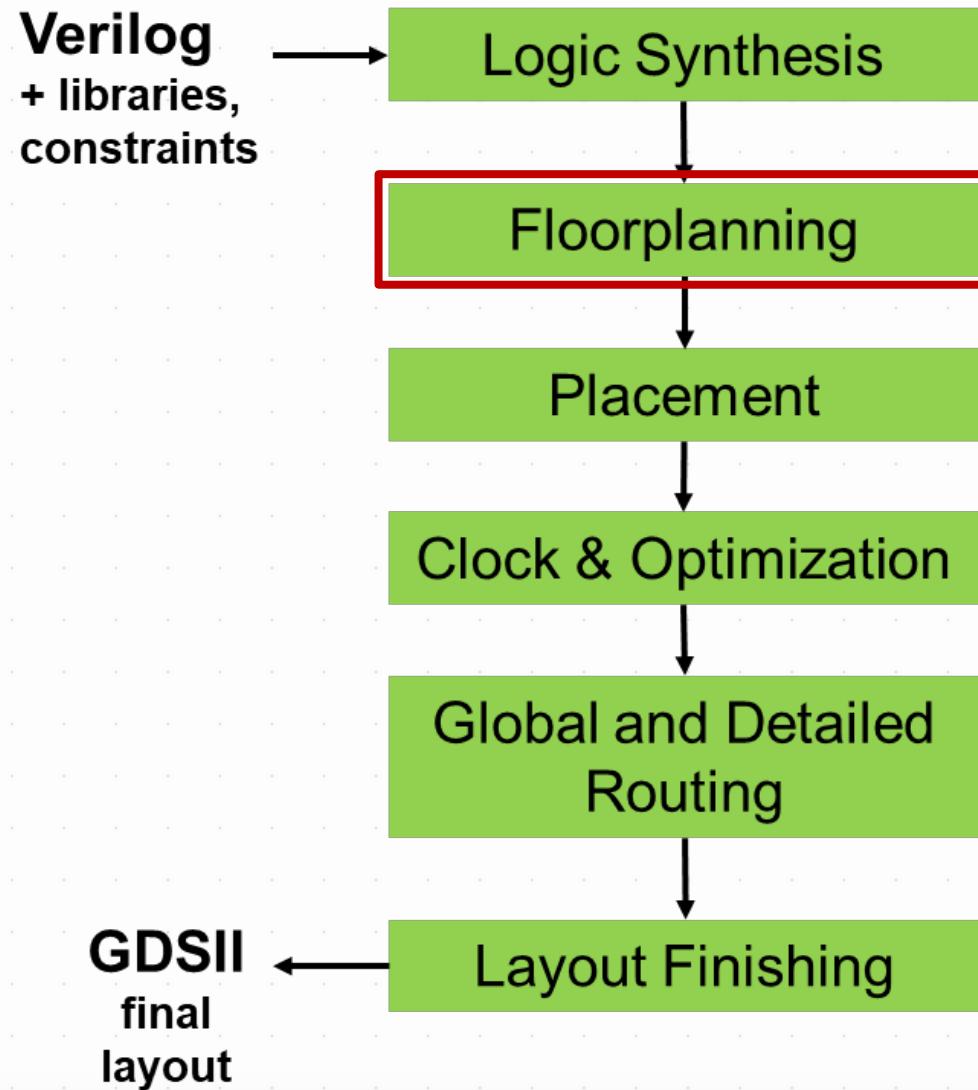
Macro Placement



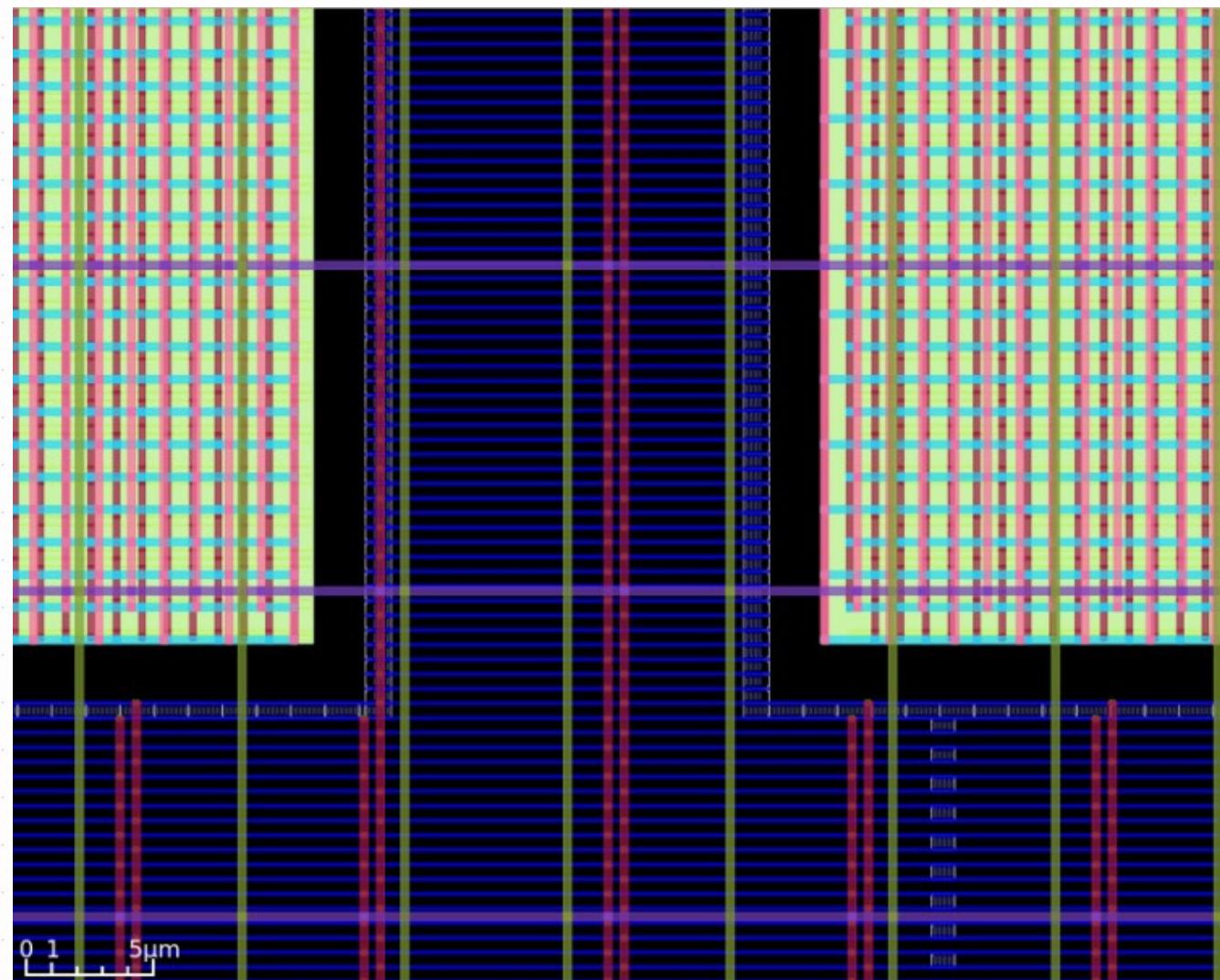
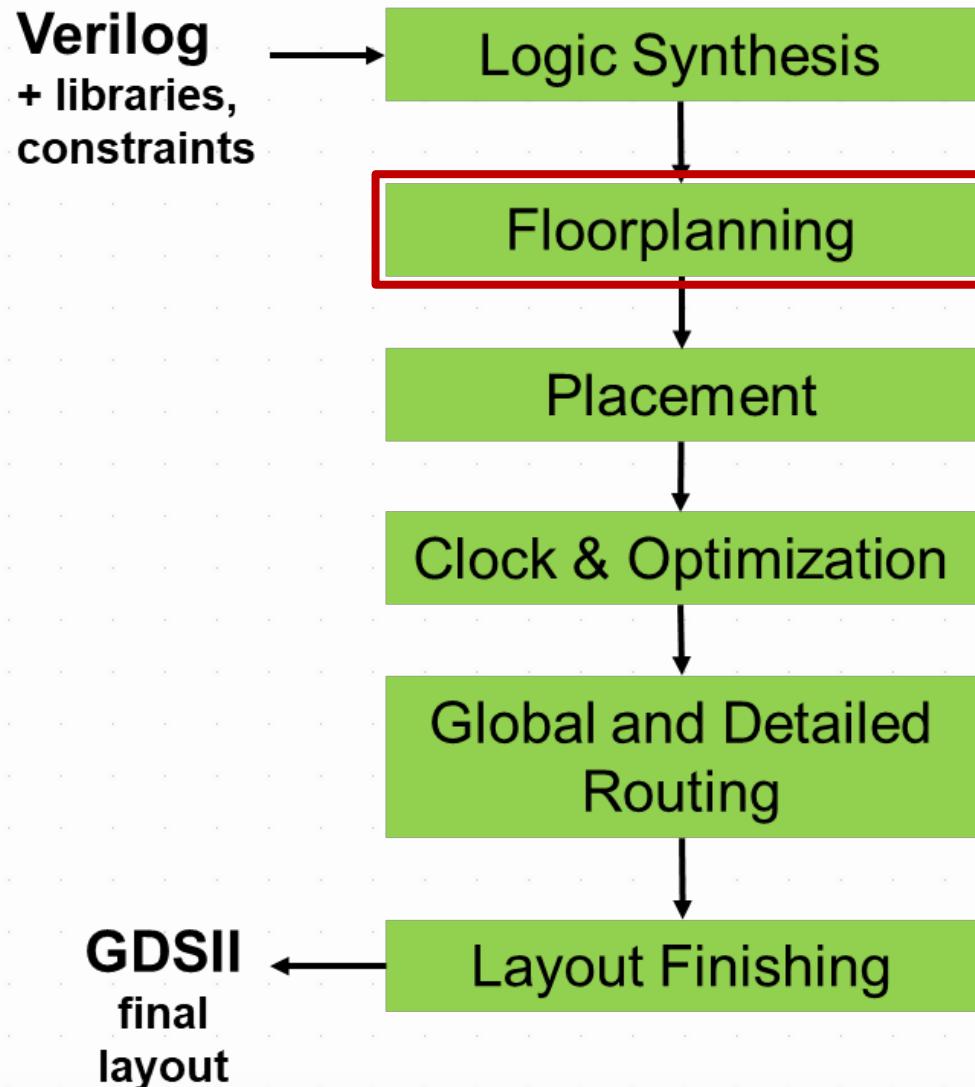
Tapcell (Well Tap) Insertion



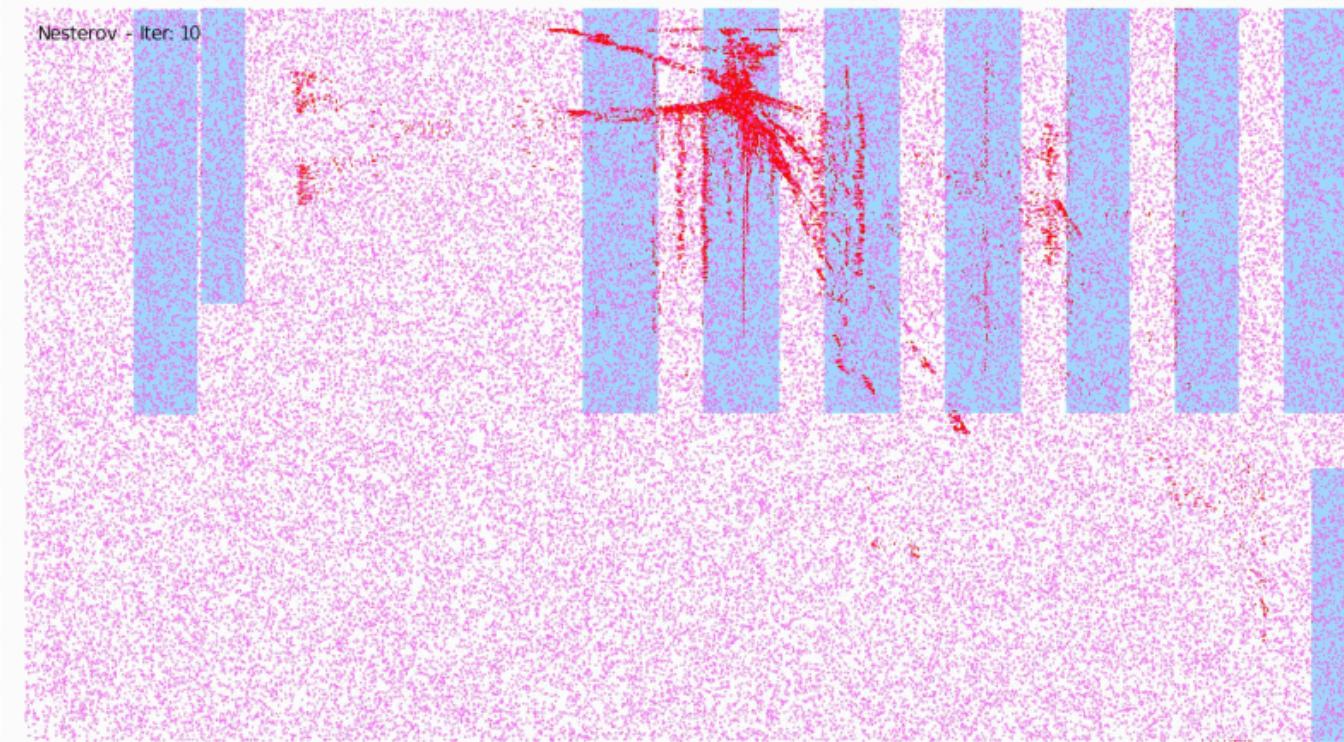
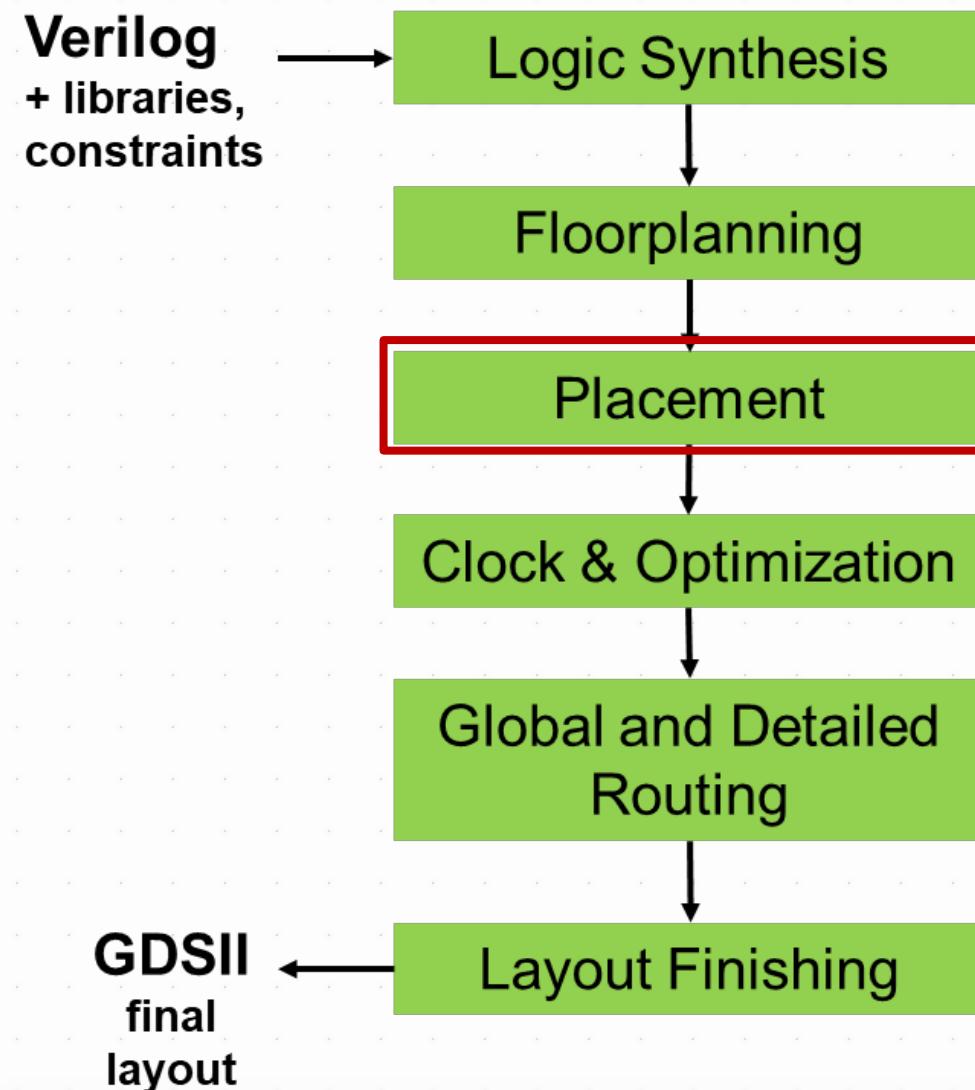
Power Delivery Network



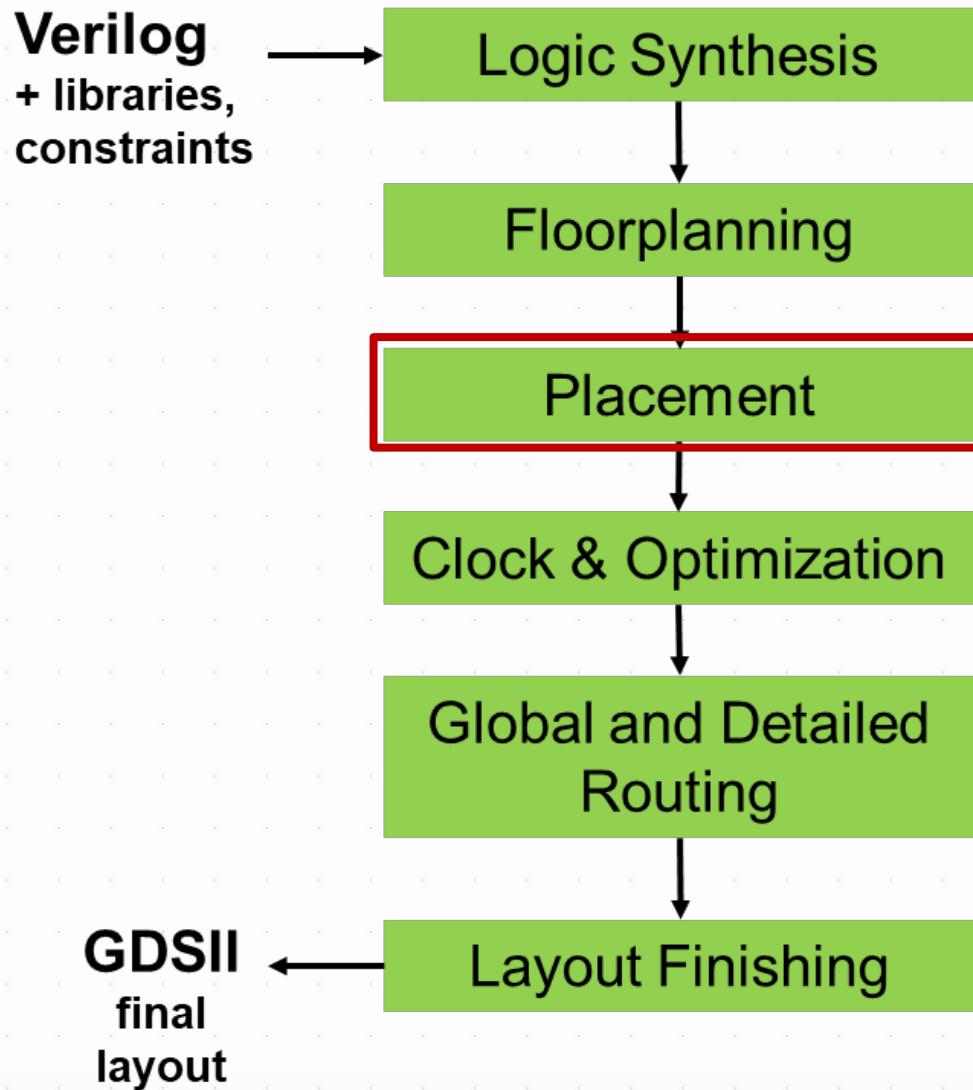
Power Delivery Network – Zoom-In



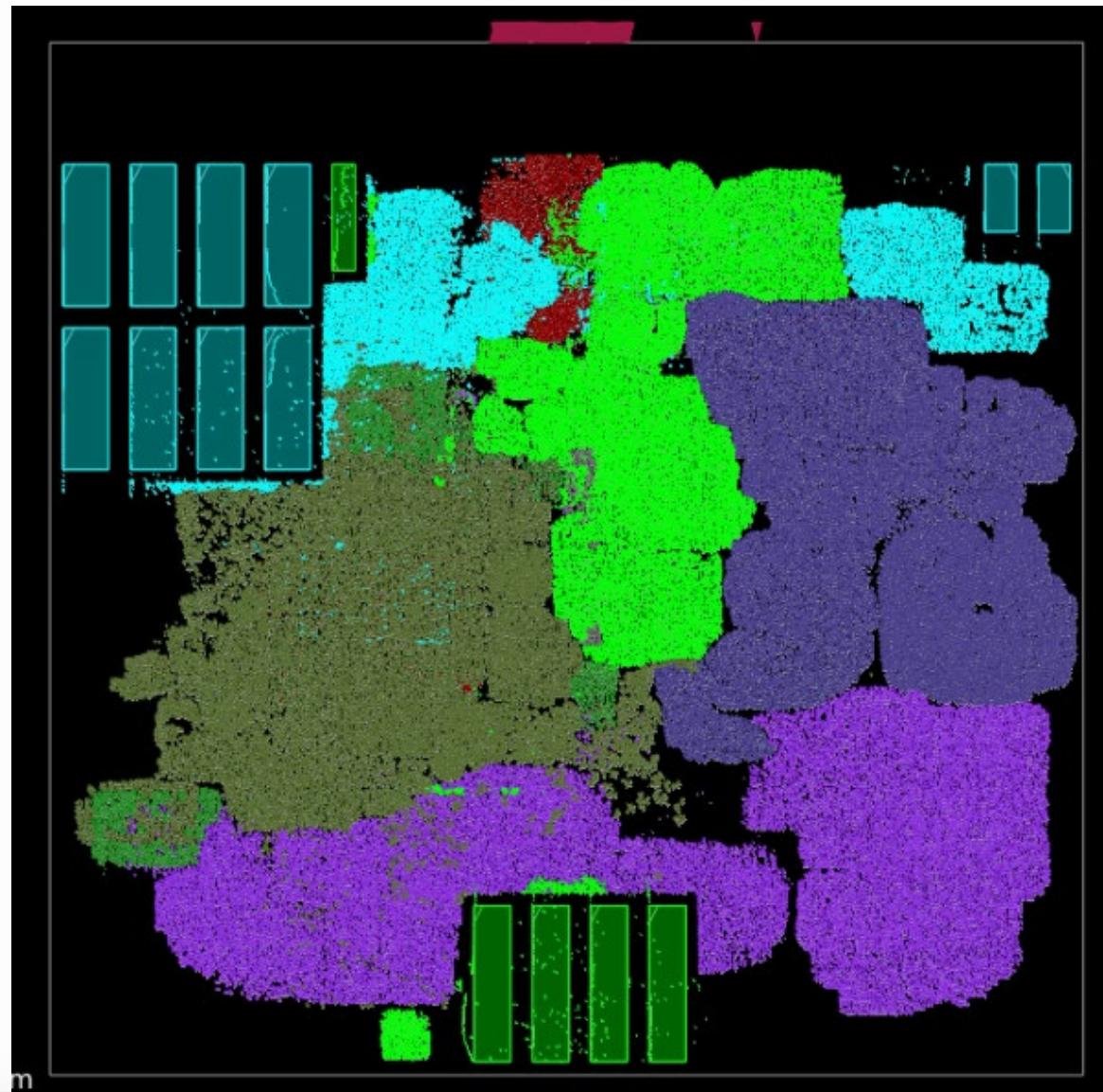
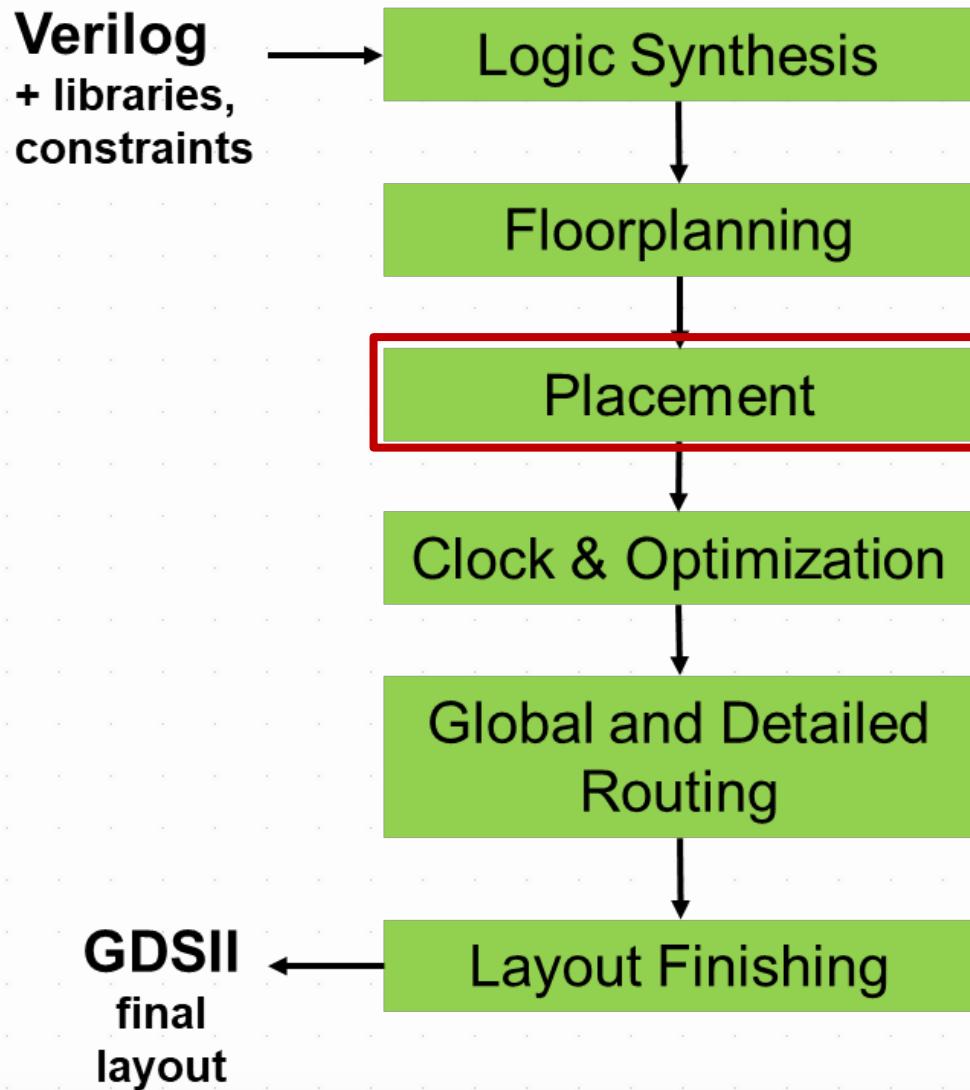
Global Placement



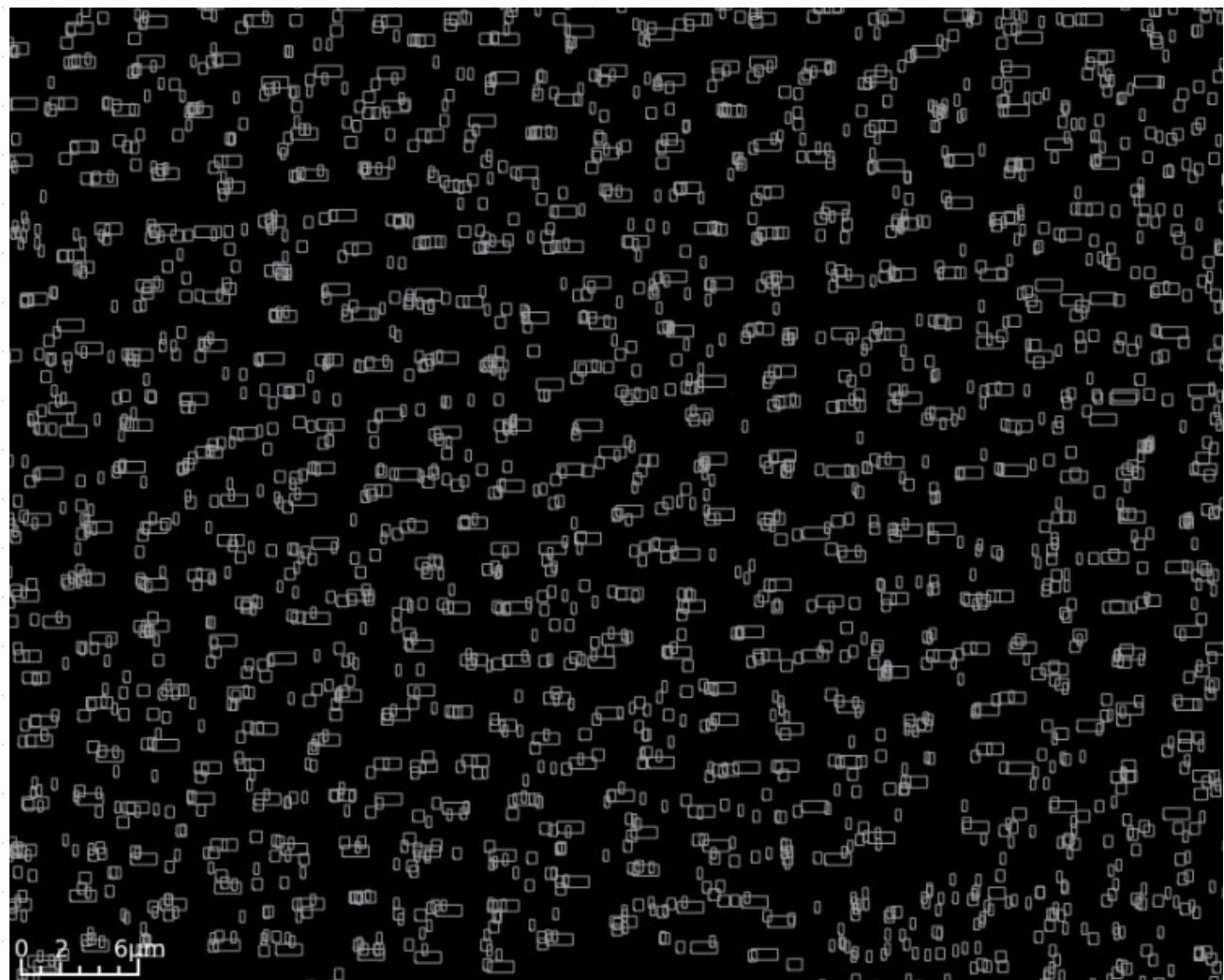
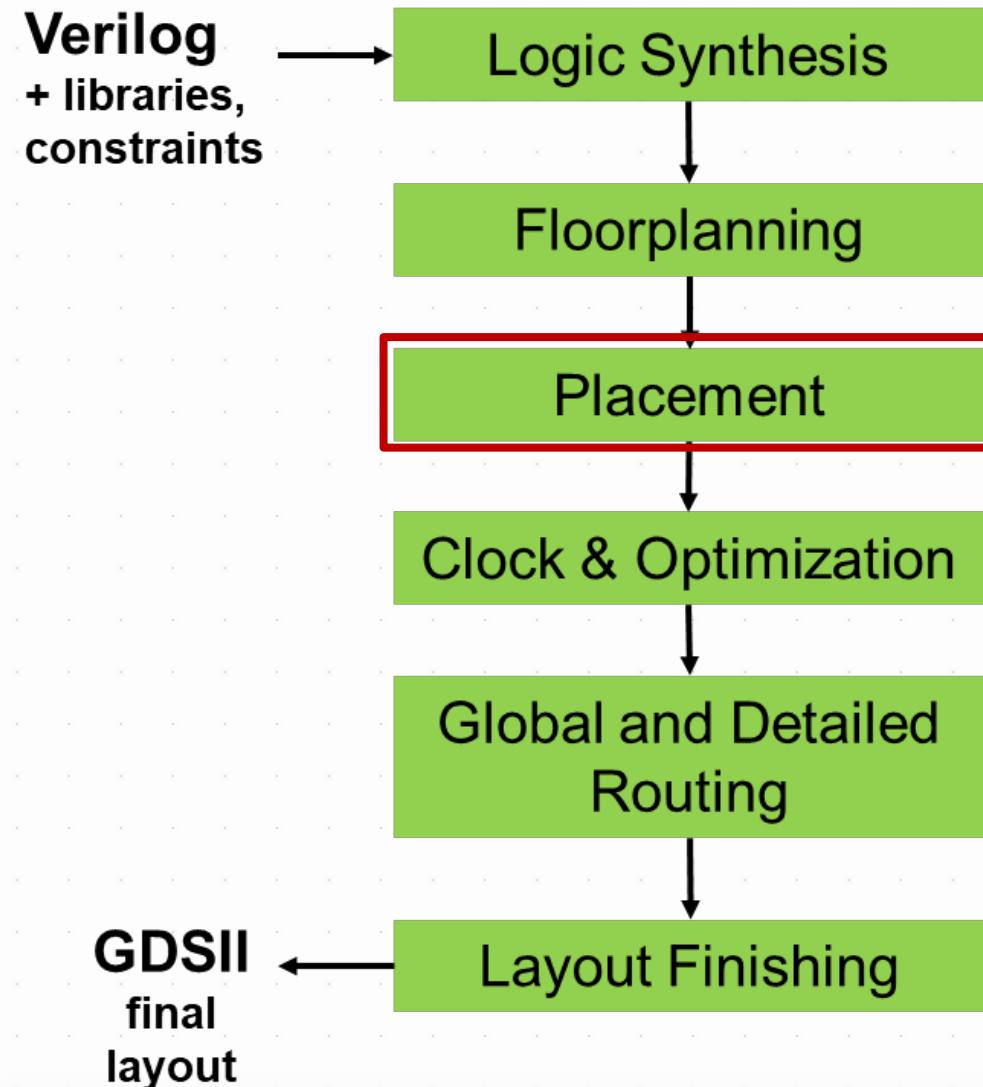
Global Placement



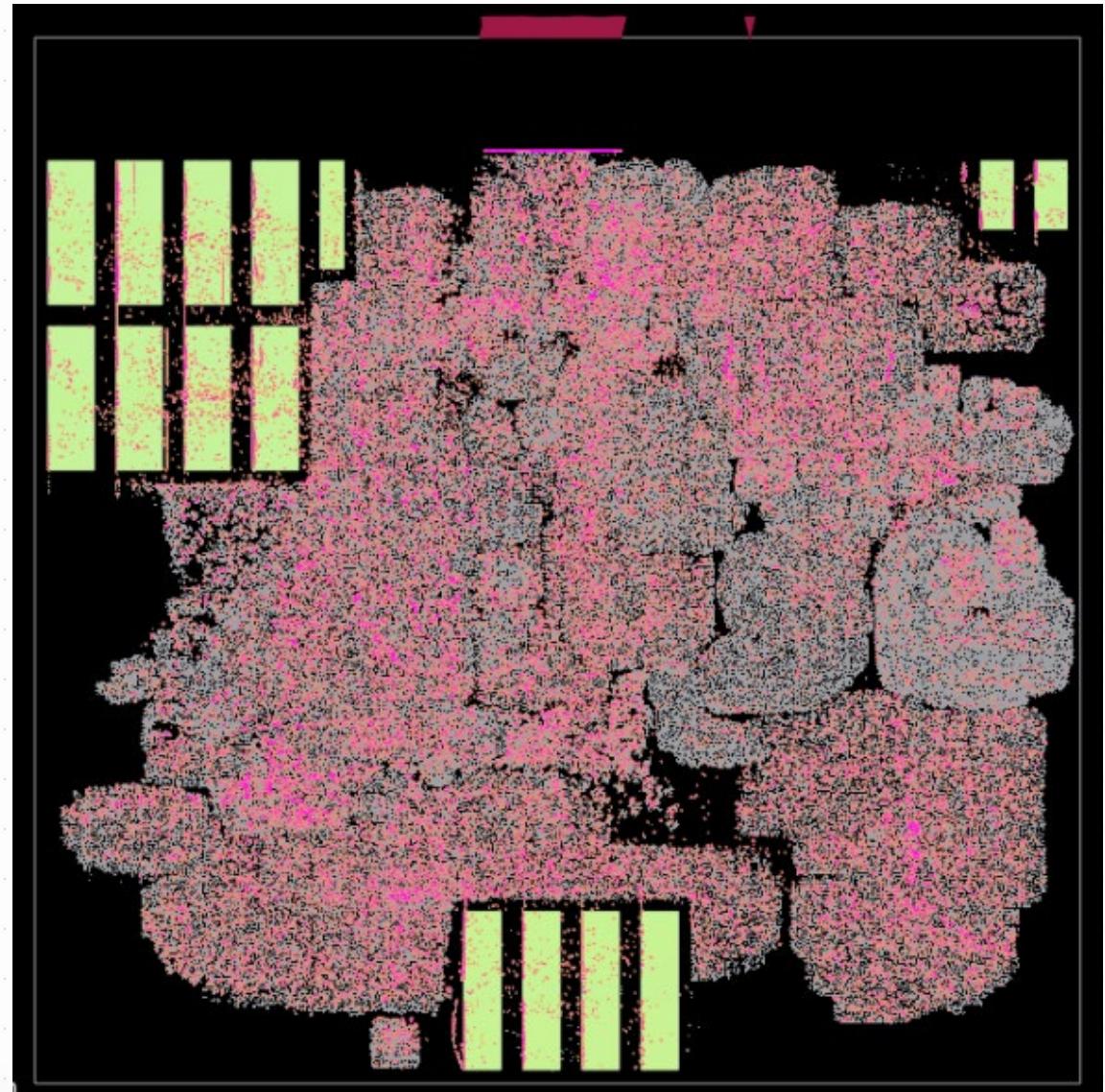
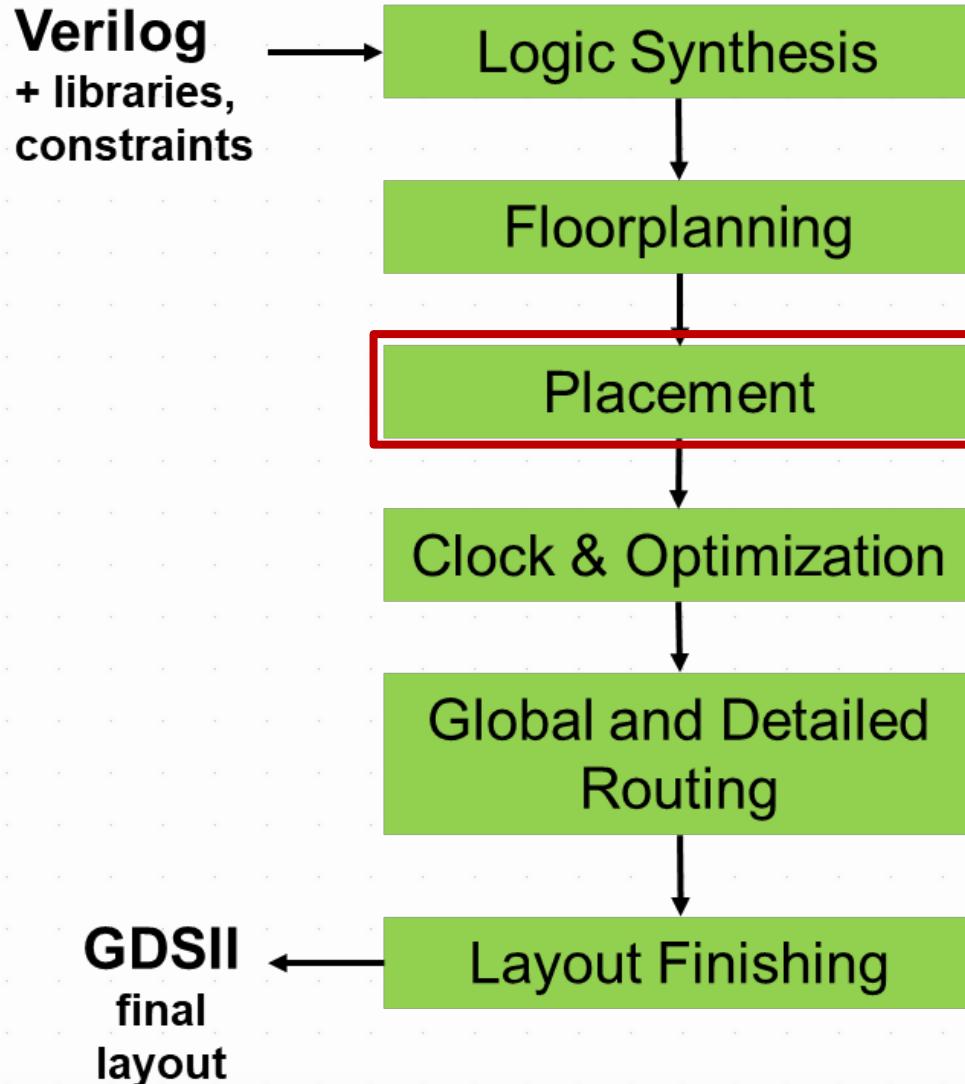
Global Placement w/Partially Expanded Hierarchy



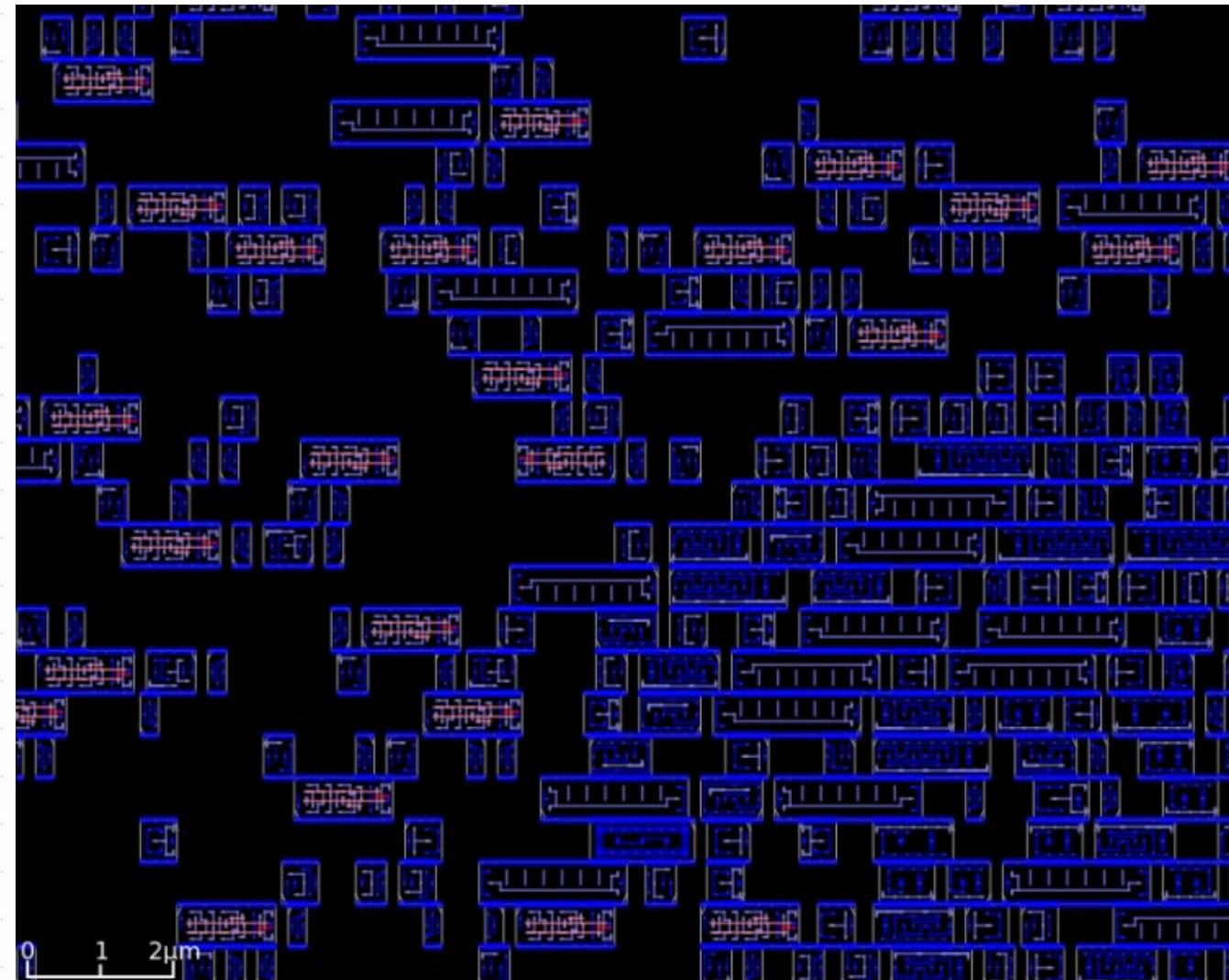
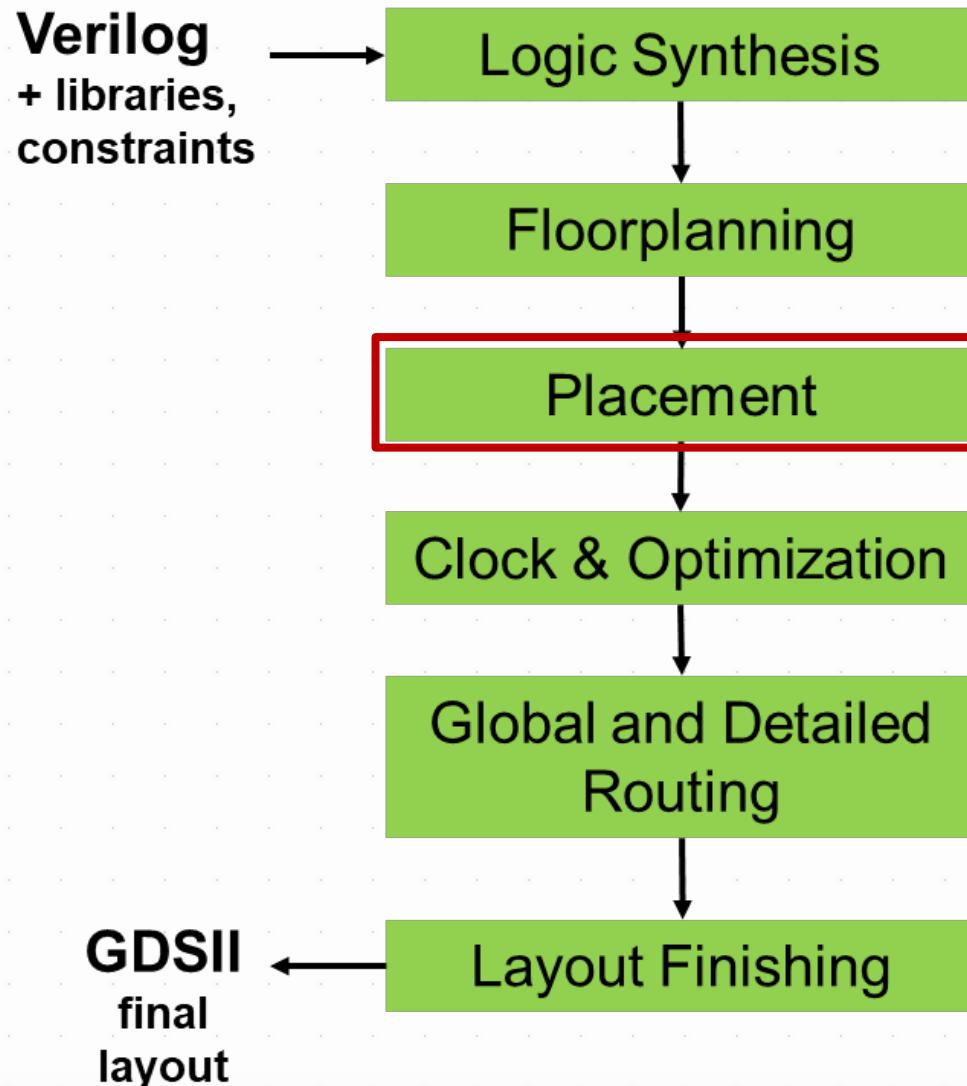
Global Placement Zoom-In



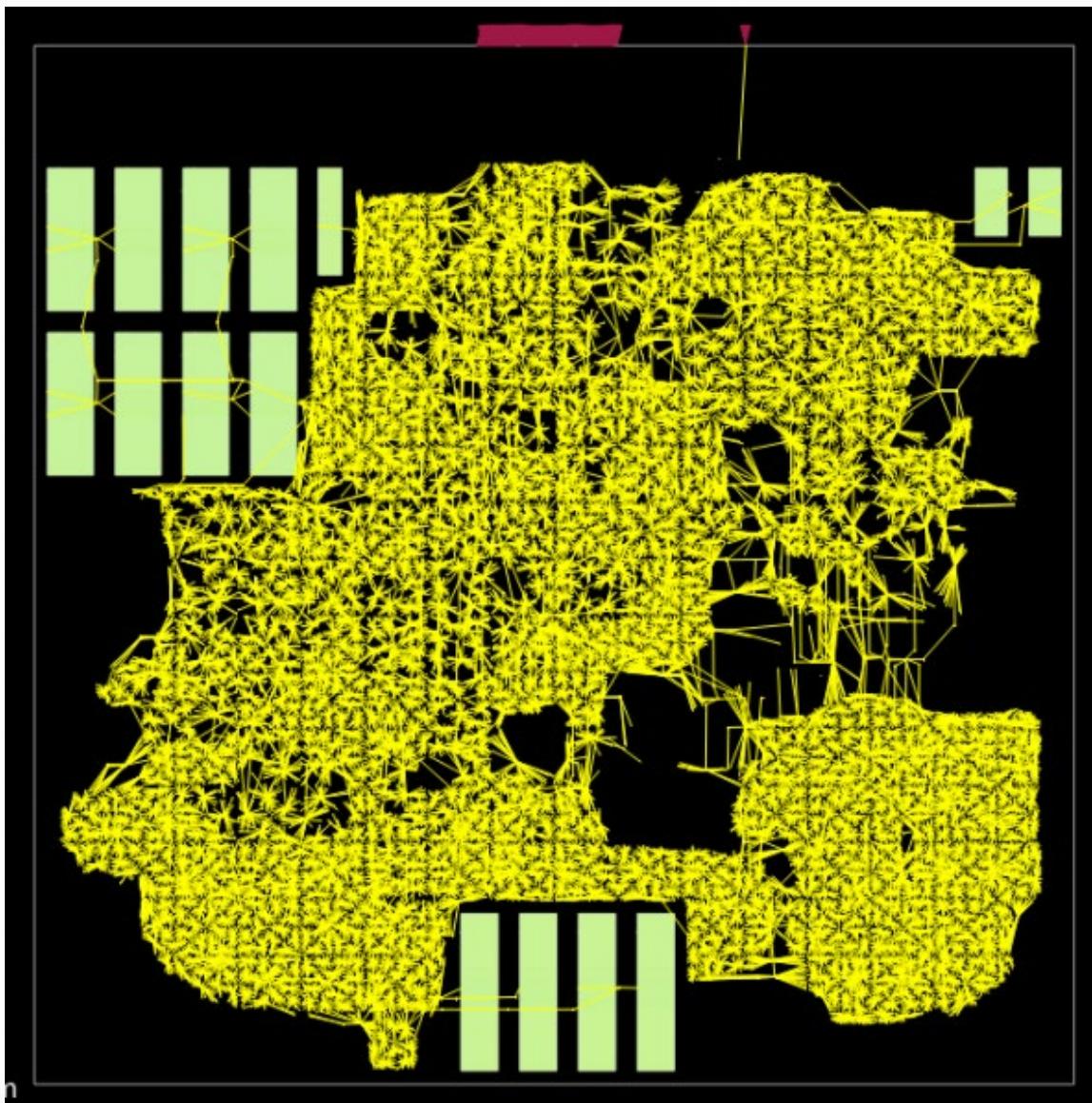
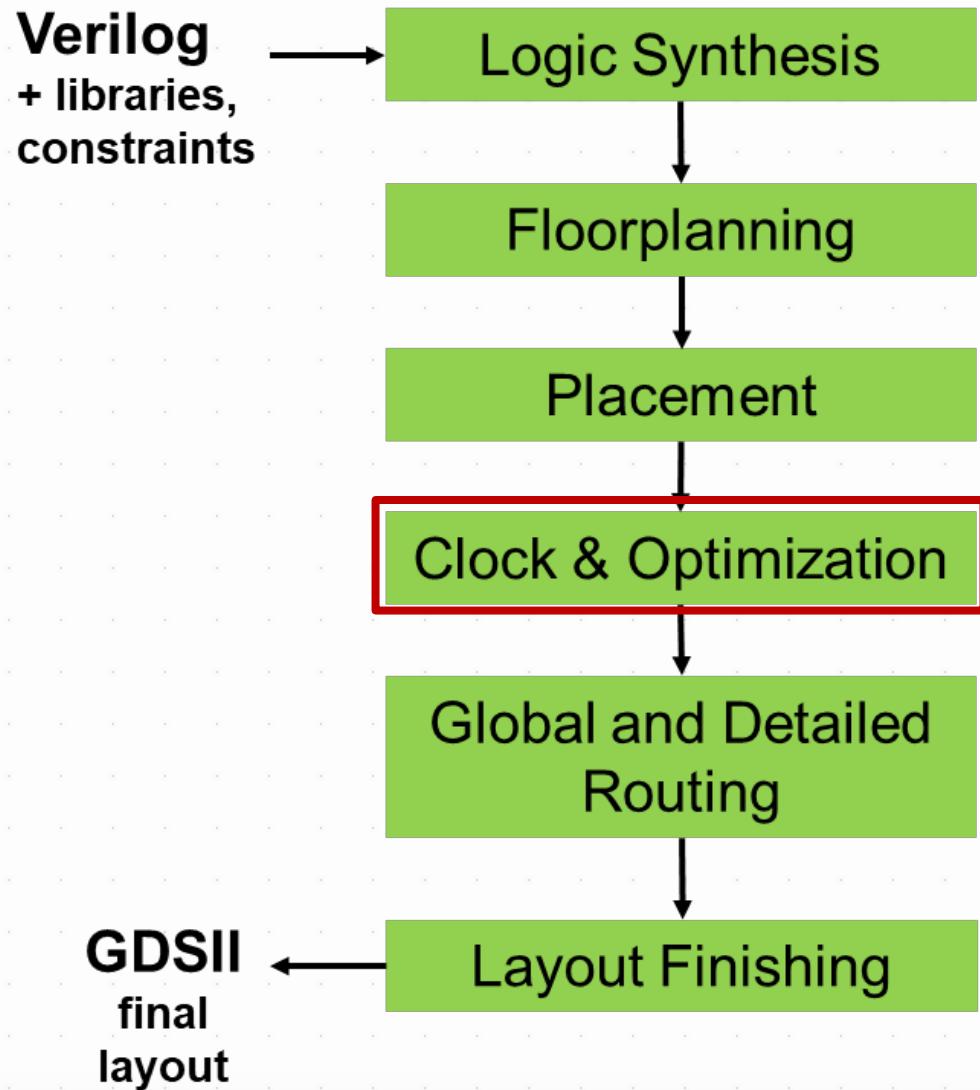
Sizing and Buffering (electrical rules)



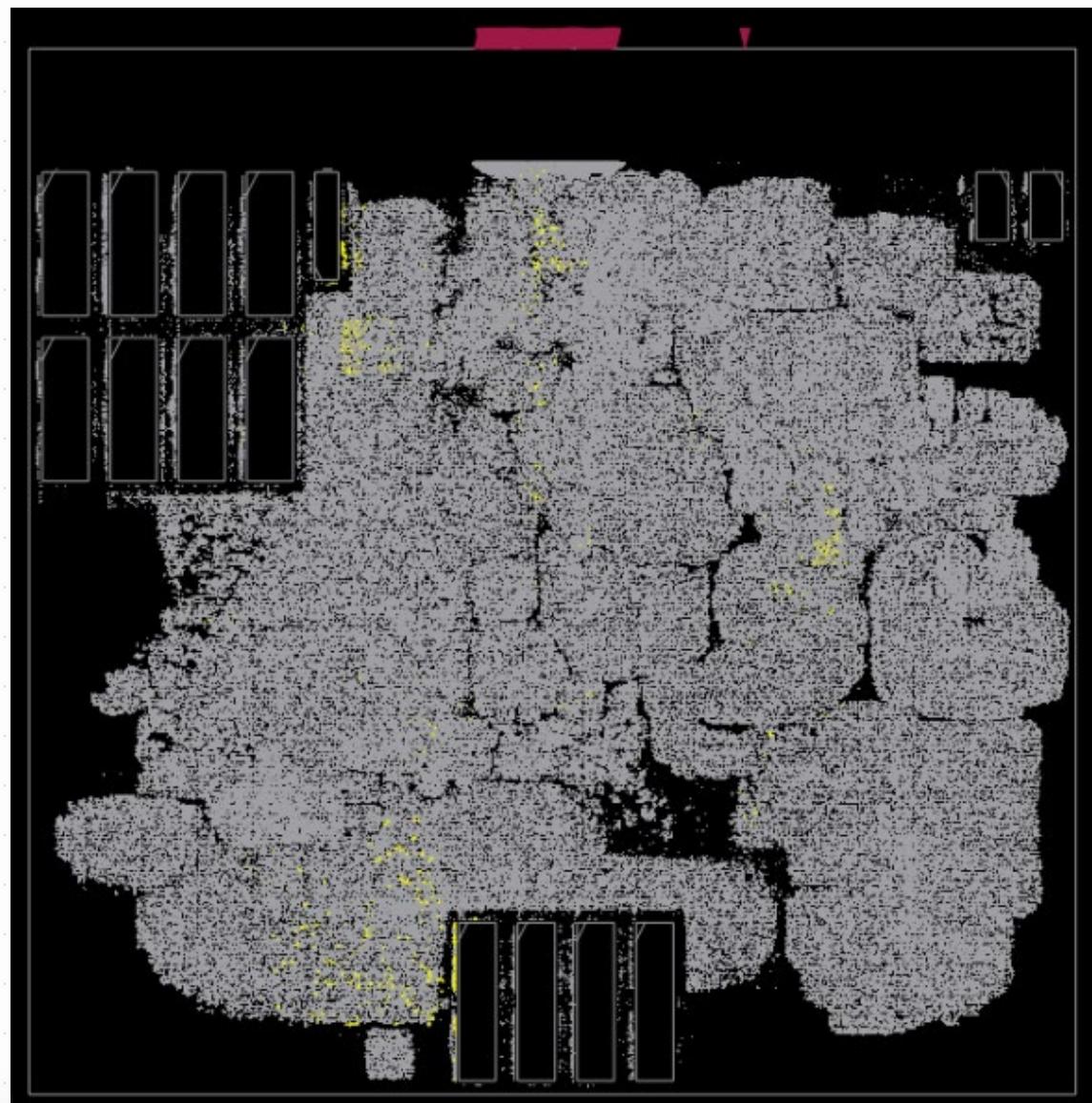
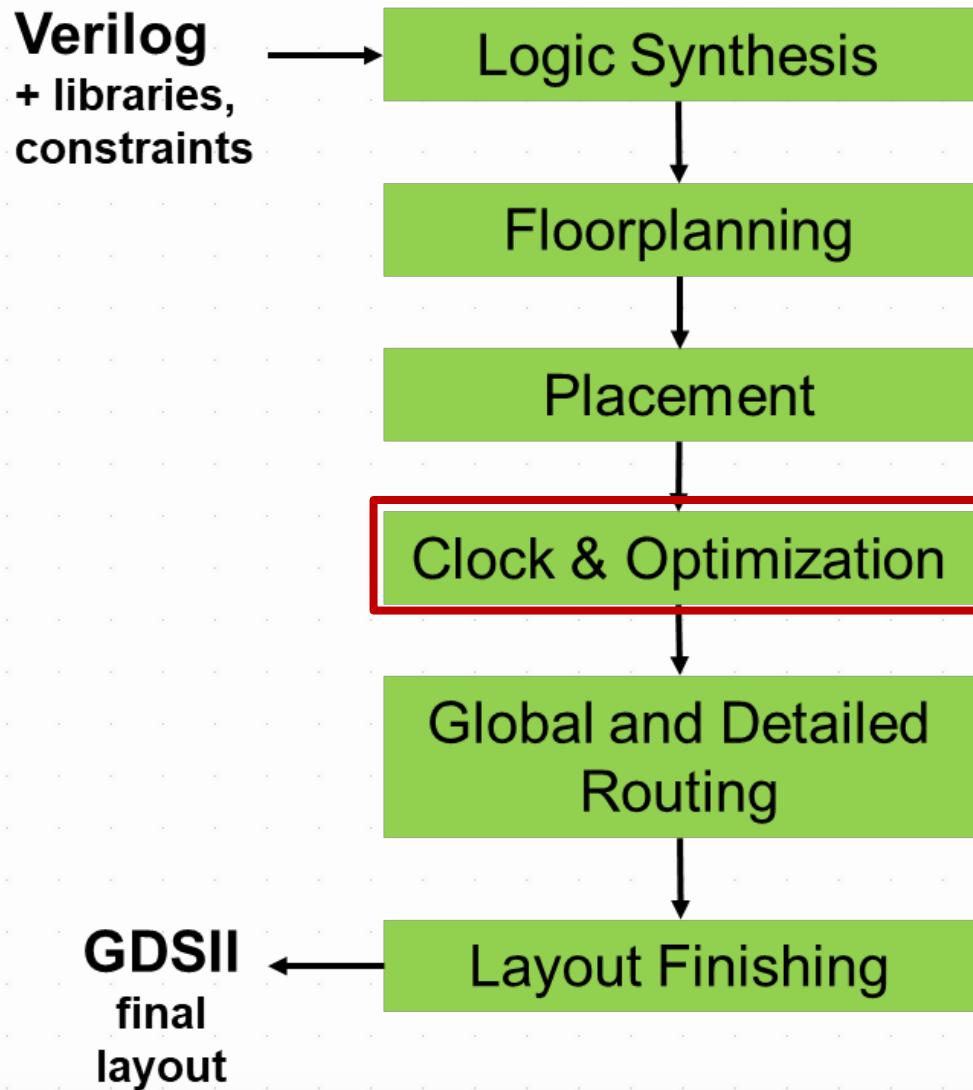
(Legalized) Detailed Placement Zoom-In



Clock Tree Synthesis

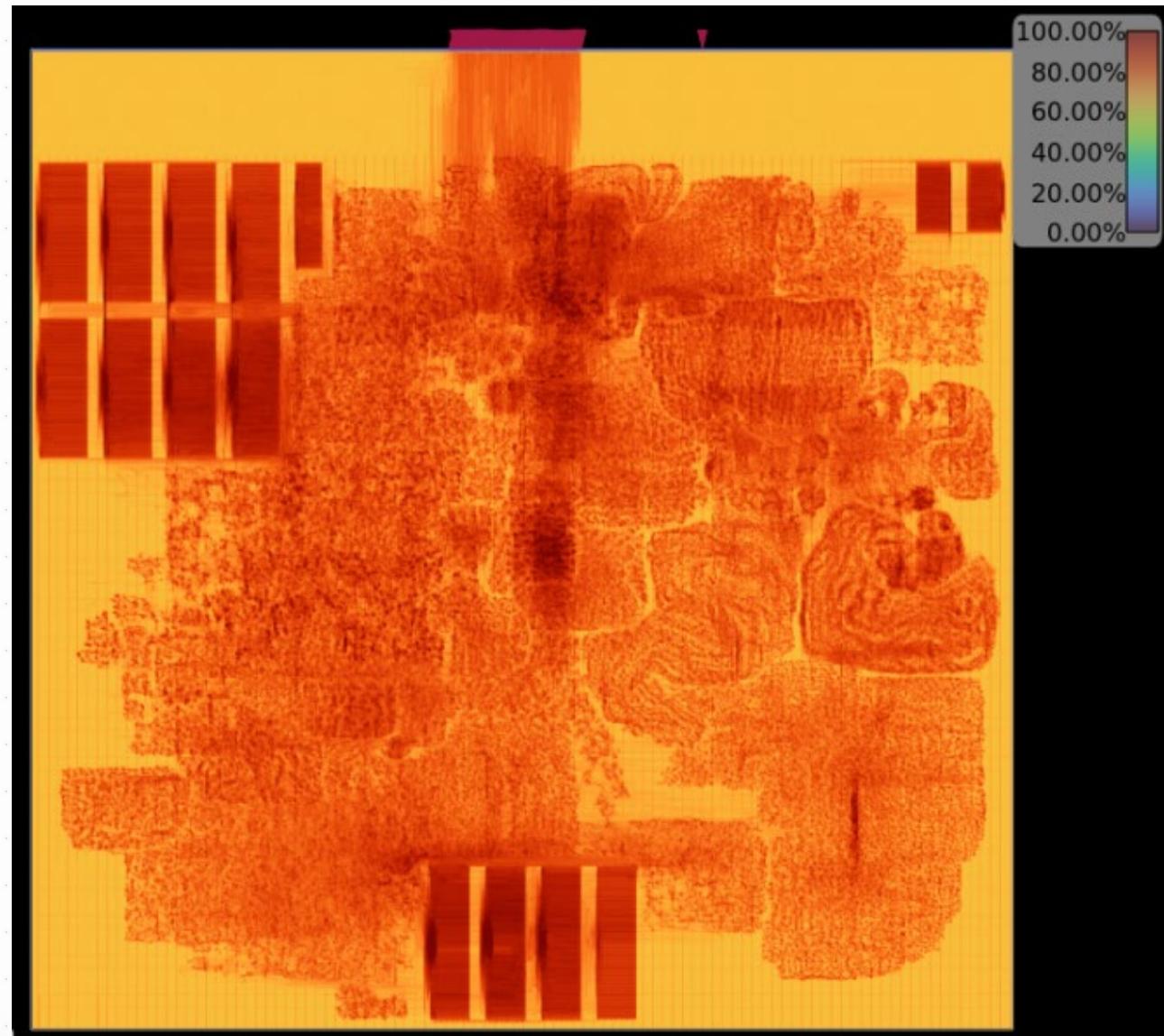
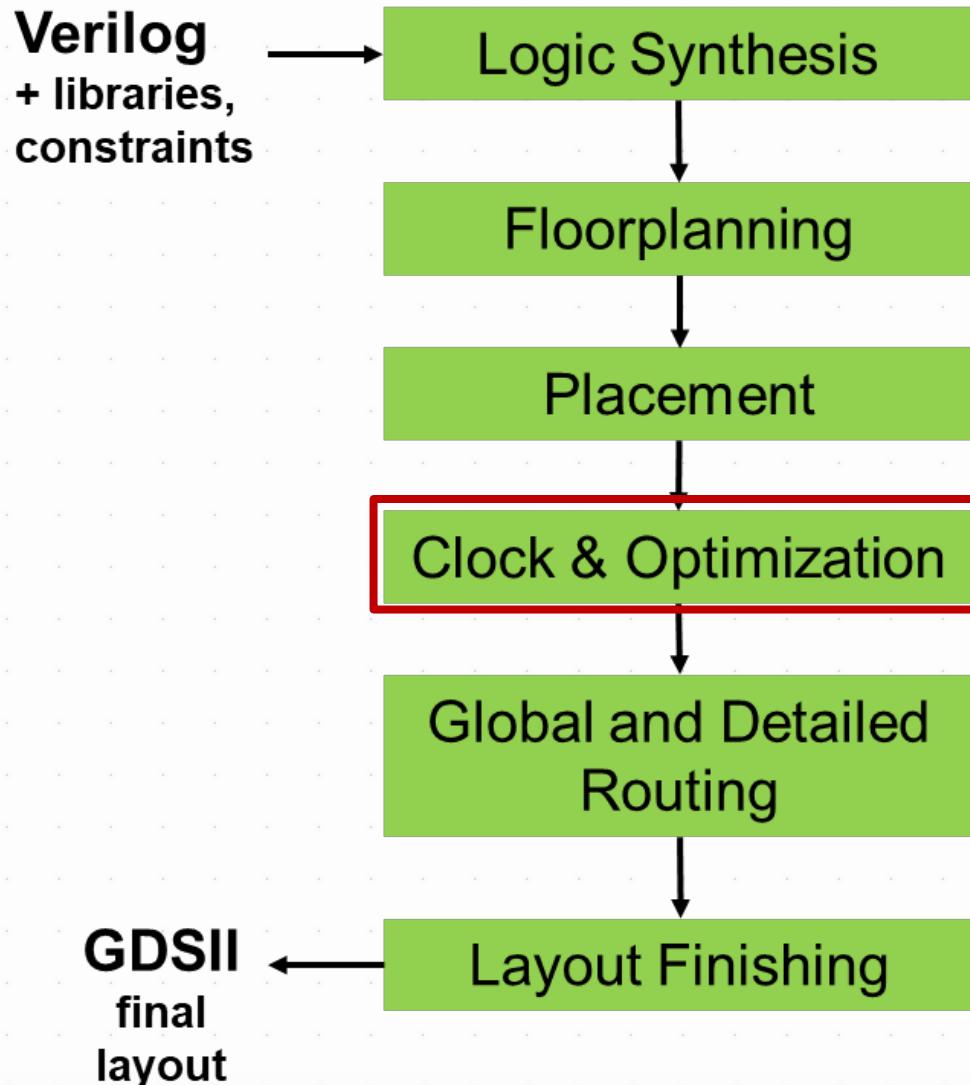


Hold Fix Buffers

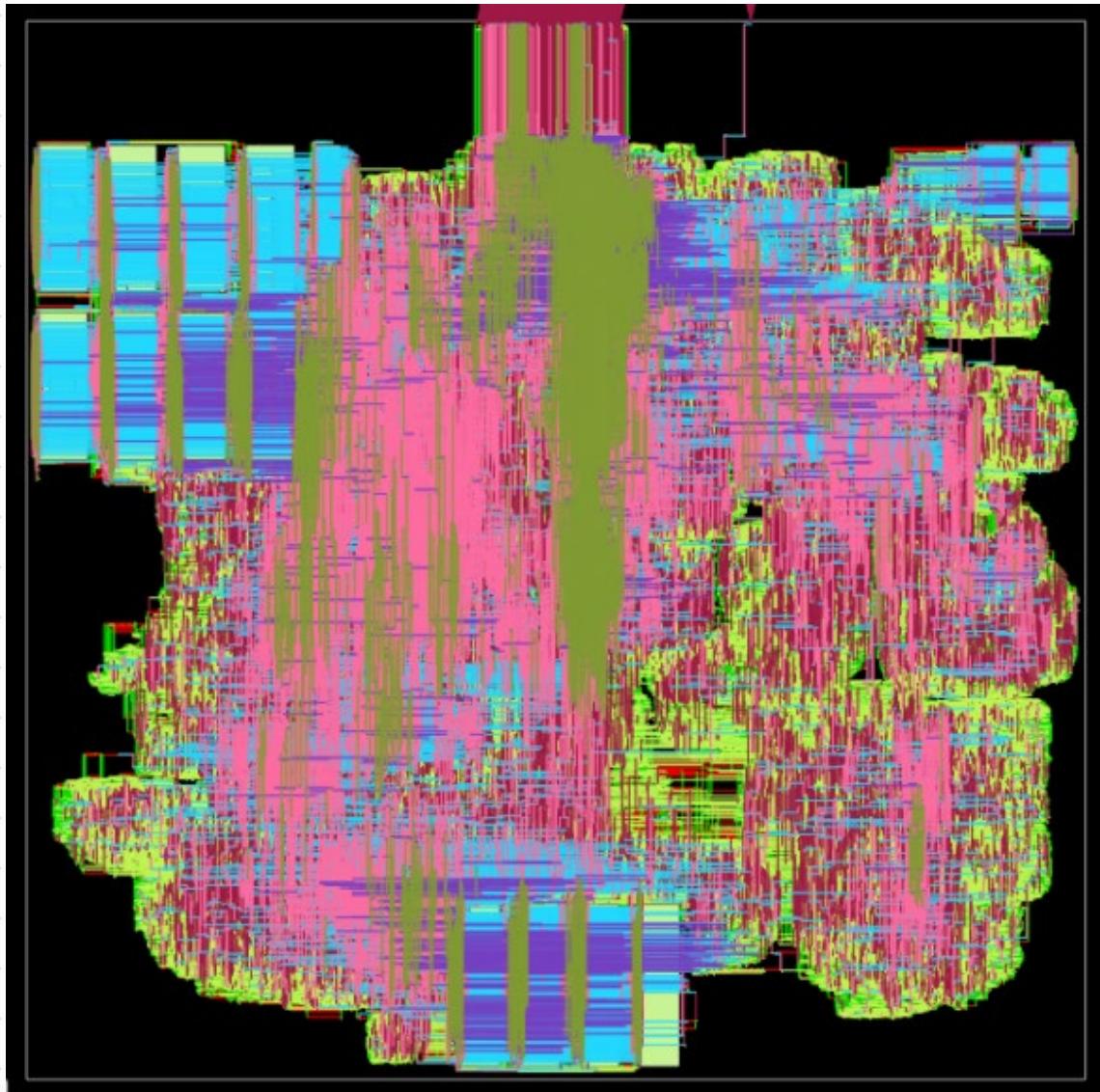
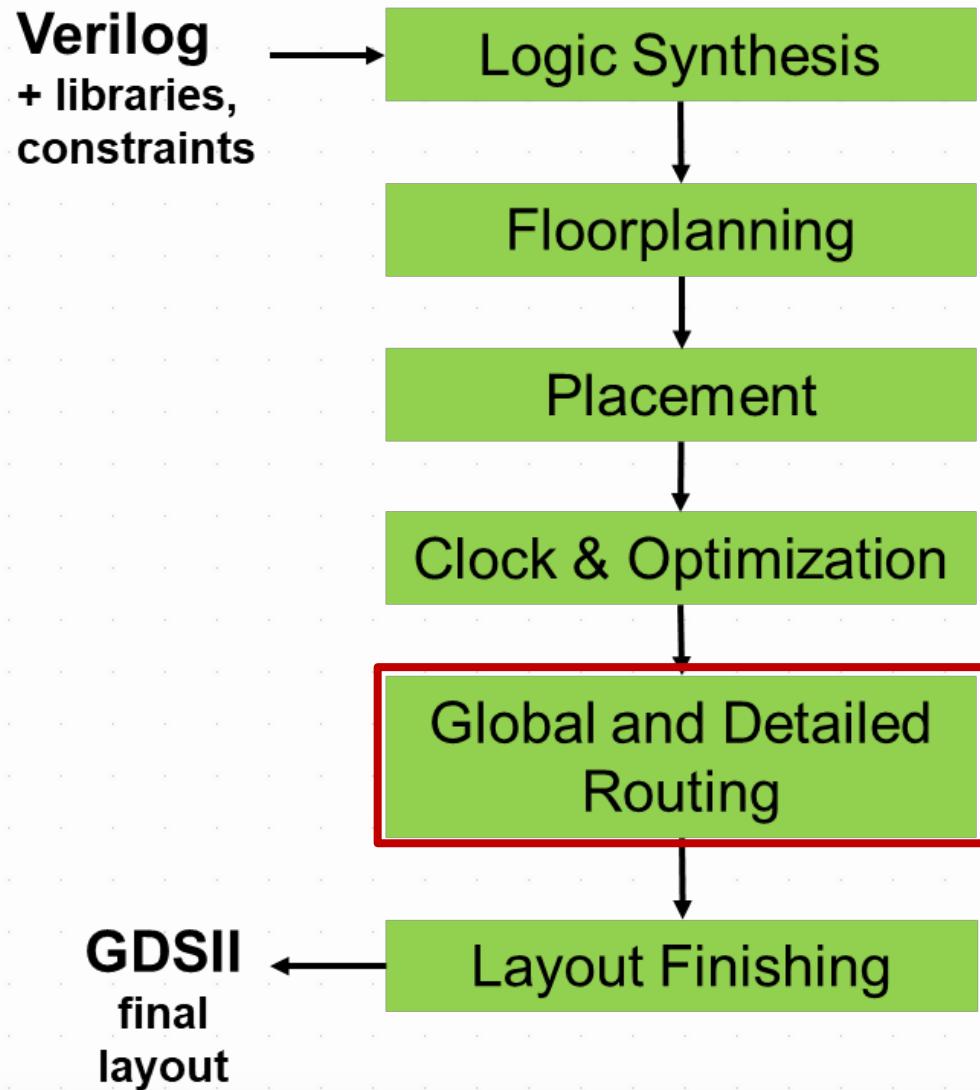


Congestion Map

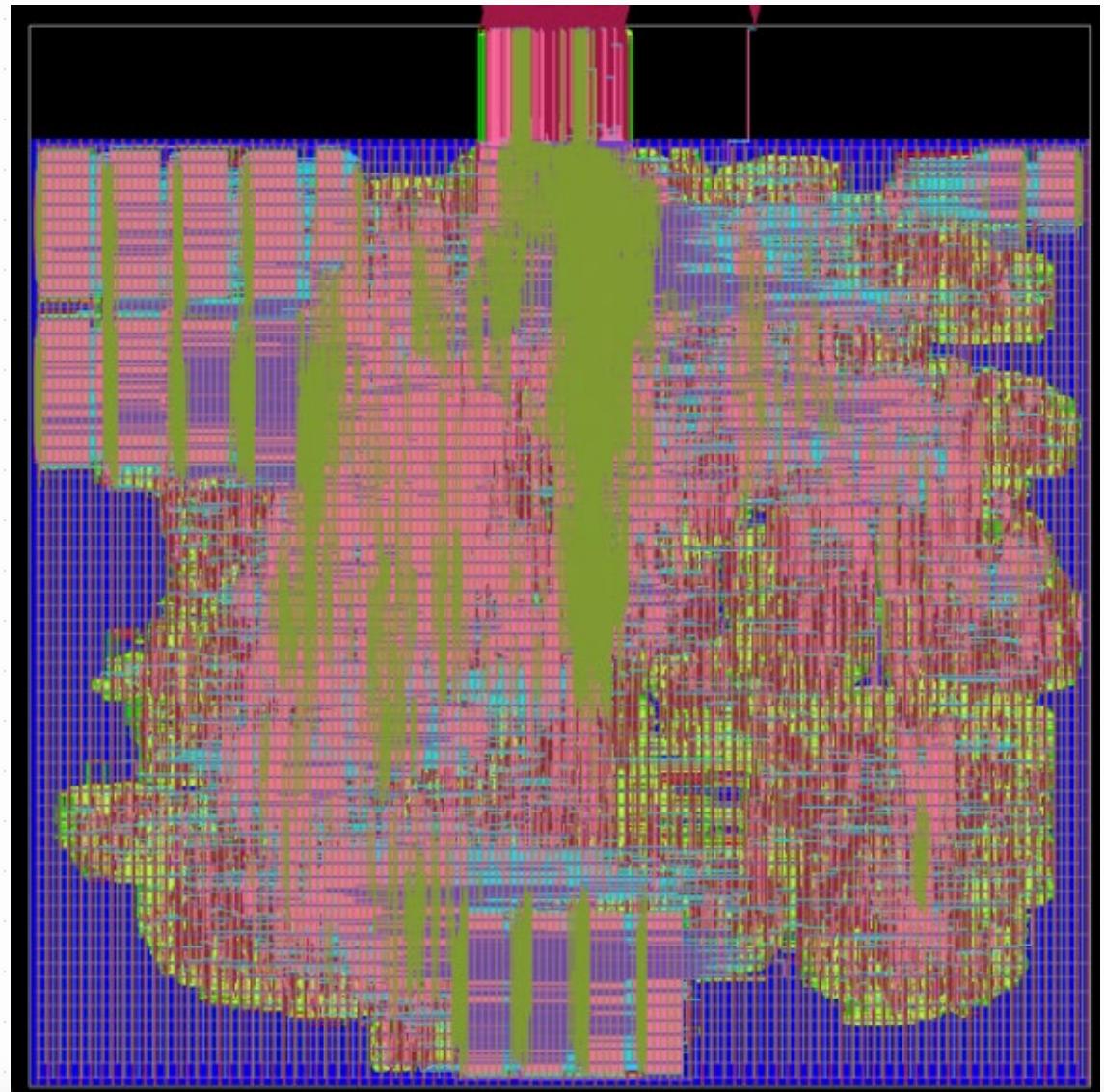
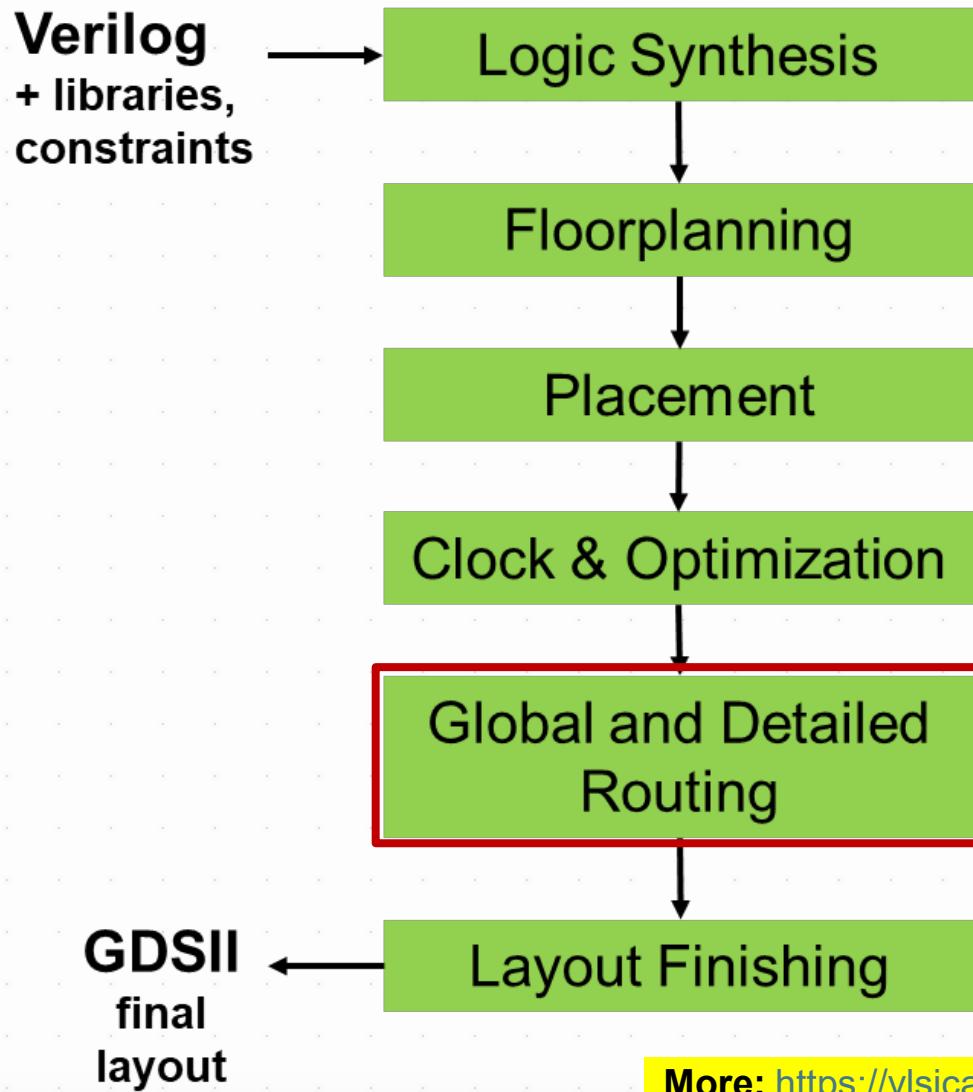
Is this final placement routable?



Final (Detailed) Routing

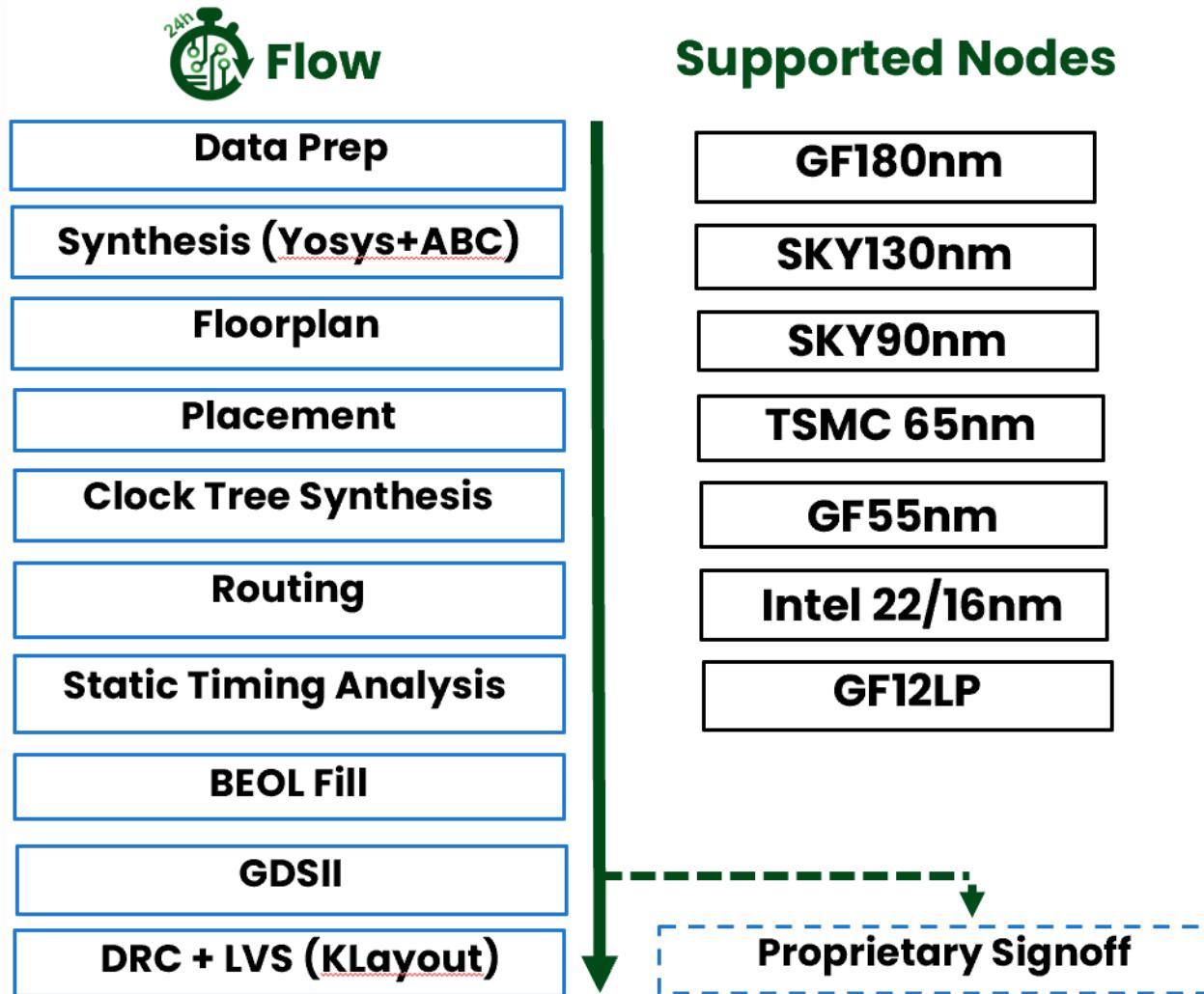


Final (Detailed) Routing, With PDN Shown



More: <https://vlsicad.ucsd.edu>
<https://theopenroadproject.org>

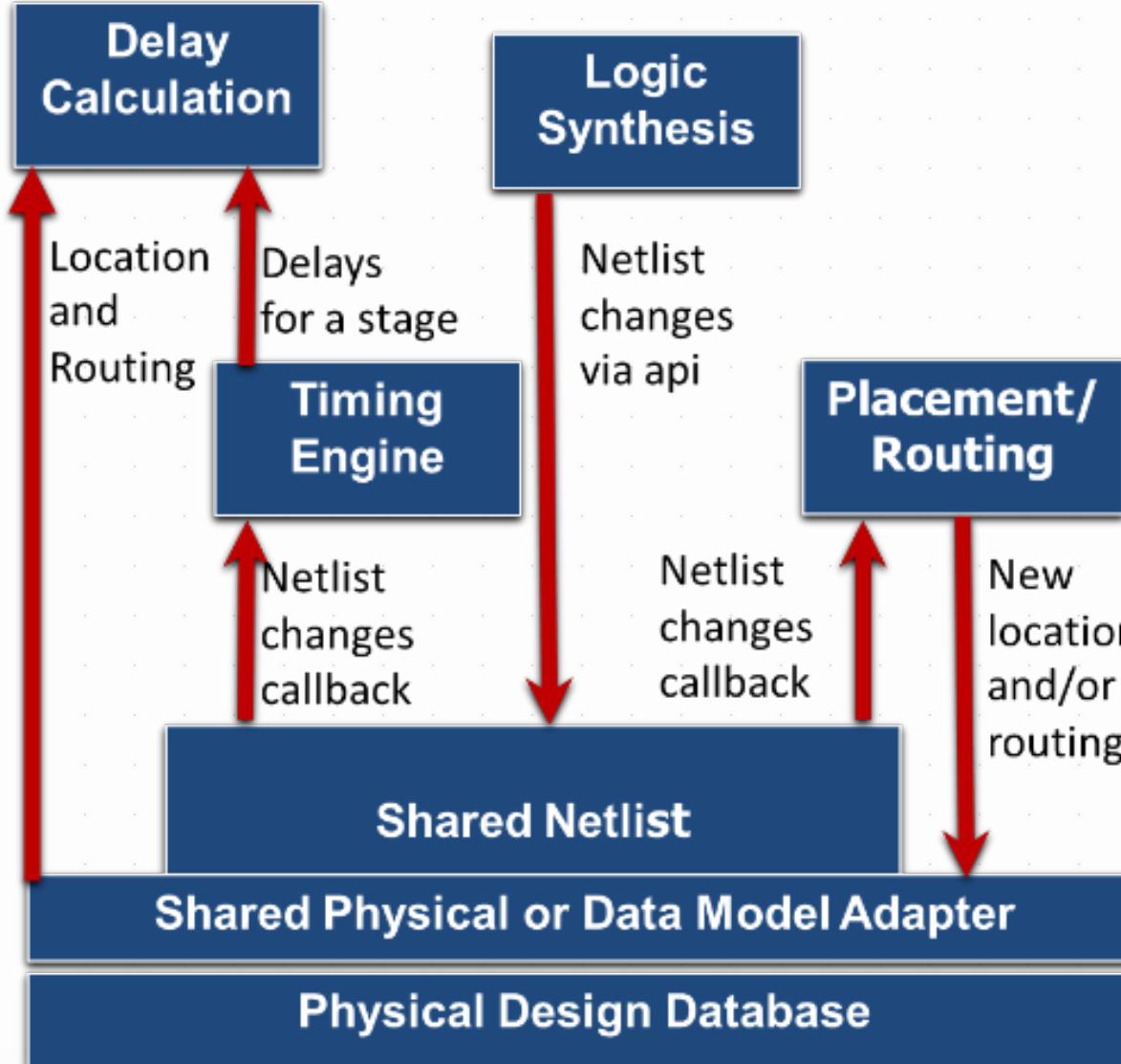
Recent Status



- **Functionality:** 800+ tapeouts at 180-12nm
 - IHP130 recently supported → this class
- **Community:** OpenROAD app has >28K commits from 130+ contributors
- **Education and Workforce:** from high school to graduate level, extension
- **Researchers**
- **Small R&D teams, startups**

Industrial Strength Incremental Architecture: Built to Last

Further notes: [link](#) [link](#) [link](#)



OpenROAD Availability

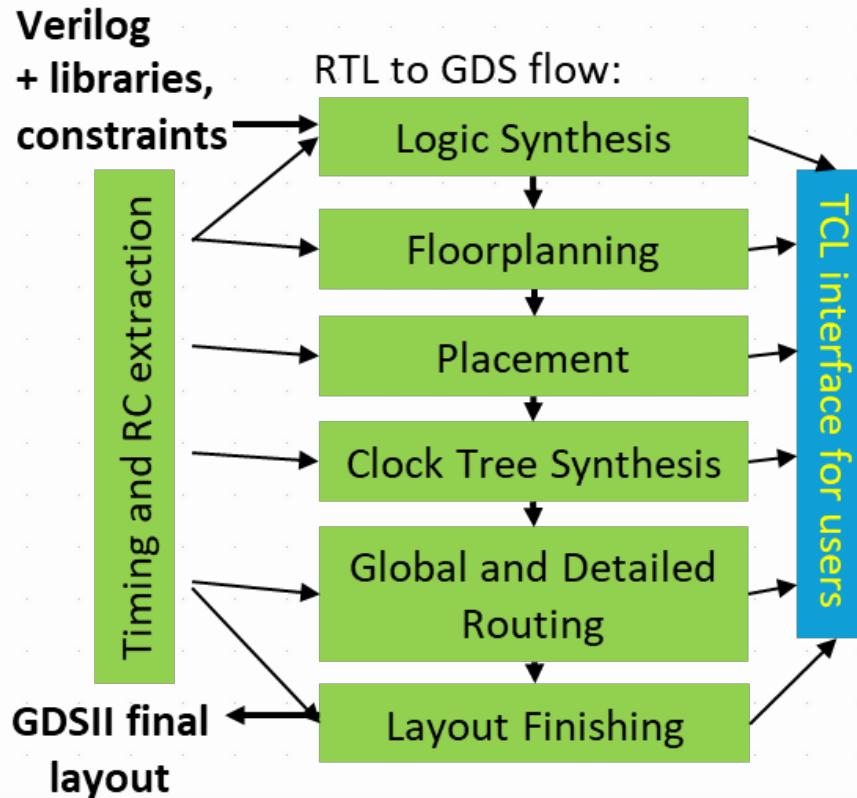
<https://openroad.readthedocs.io/en/latest/main/README.html>

- The Project on GitHub
 - <https://github.com/The-OpenROAD-Project>
- The Flow
 - Automated full flow, built using tool components that are created for automation
 - <https://github.com/The-OpenROAD-Project/OpenROAD-flow-scripts>
- The Top-level Application
 - An integrated EDA tool focused on full automation
 - <https://github.com/The-OpenROAD-Project/OpenROAD>
- More!
 - Documentation: <https://openroad.readthedocs.io/en/latest/main/README.html>
 - Slack: <https://skywater-pdk.slack.com/archives/C0161A4A59V>
 - OpenTapeout video:
<https://www.youtube.com/watch?v=wvPZREaP7E0&t=2652s>

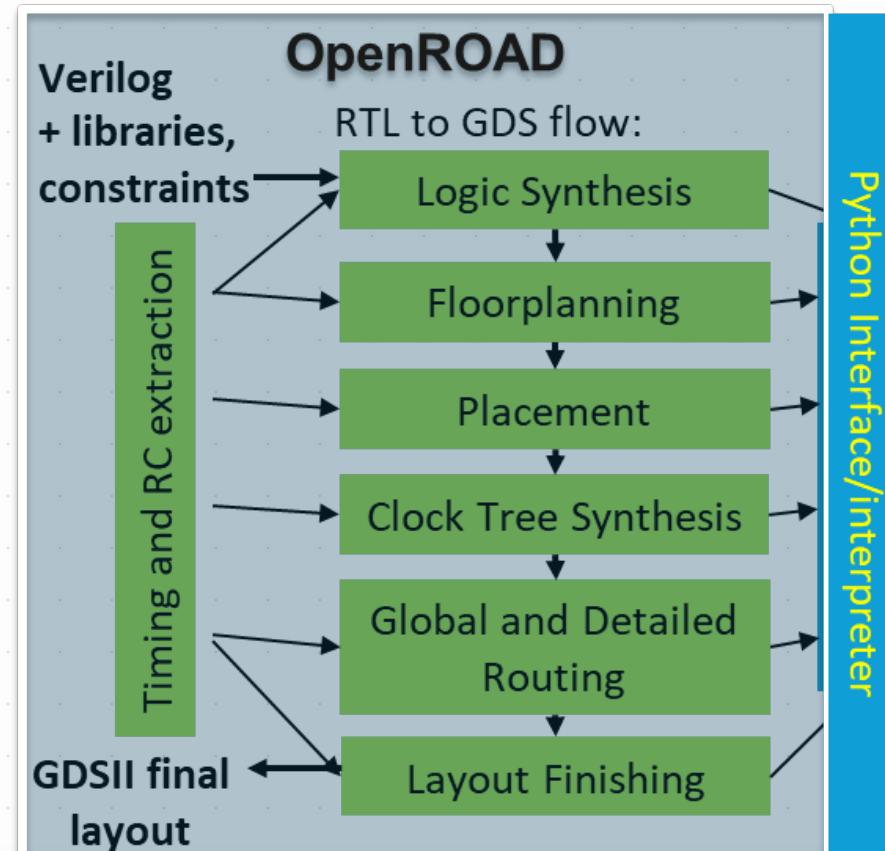
*BUT: What does OpenROAD
(and, OS EDA) bring to the table?*

Open-source EDA Enables AI/ML for EDA, IC Design

What do designers/users work with?

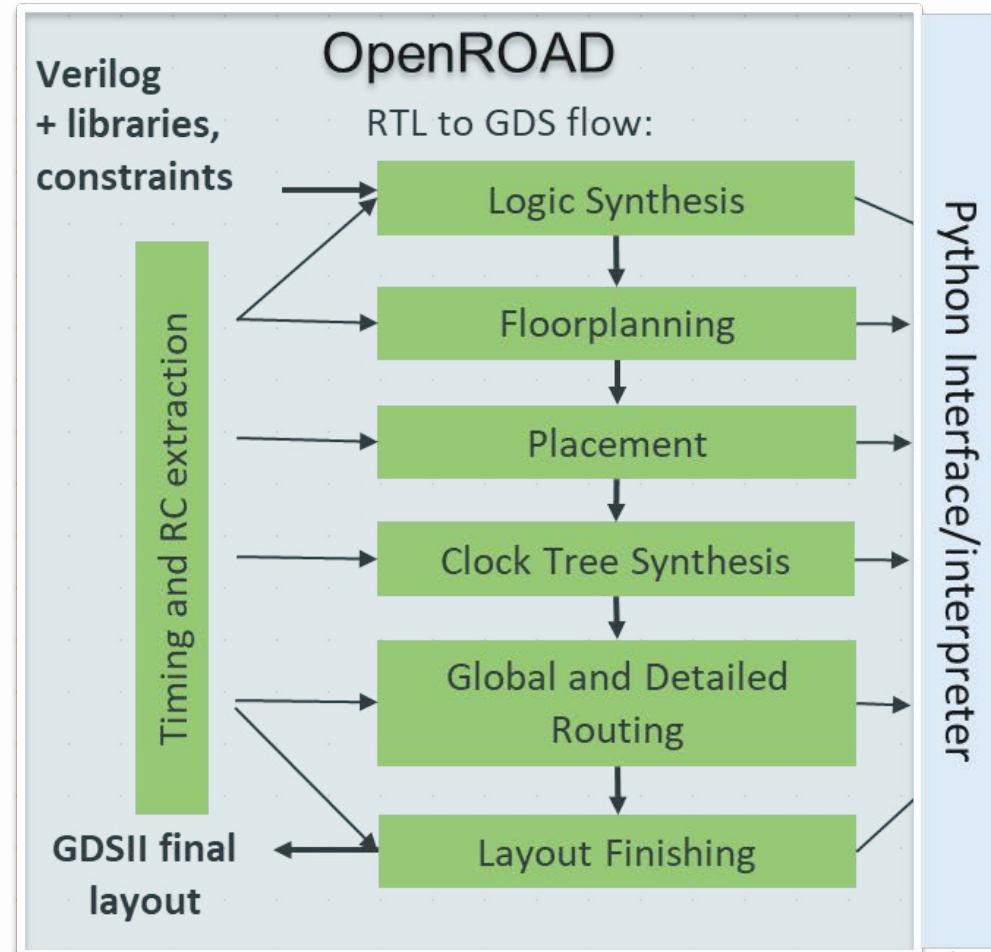


What can they also work with?



Enables an ML for EDA playground

OpenROAD as an ML for EDA “Playground”

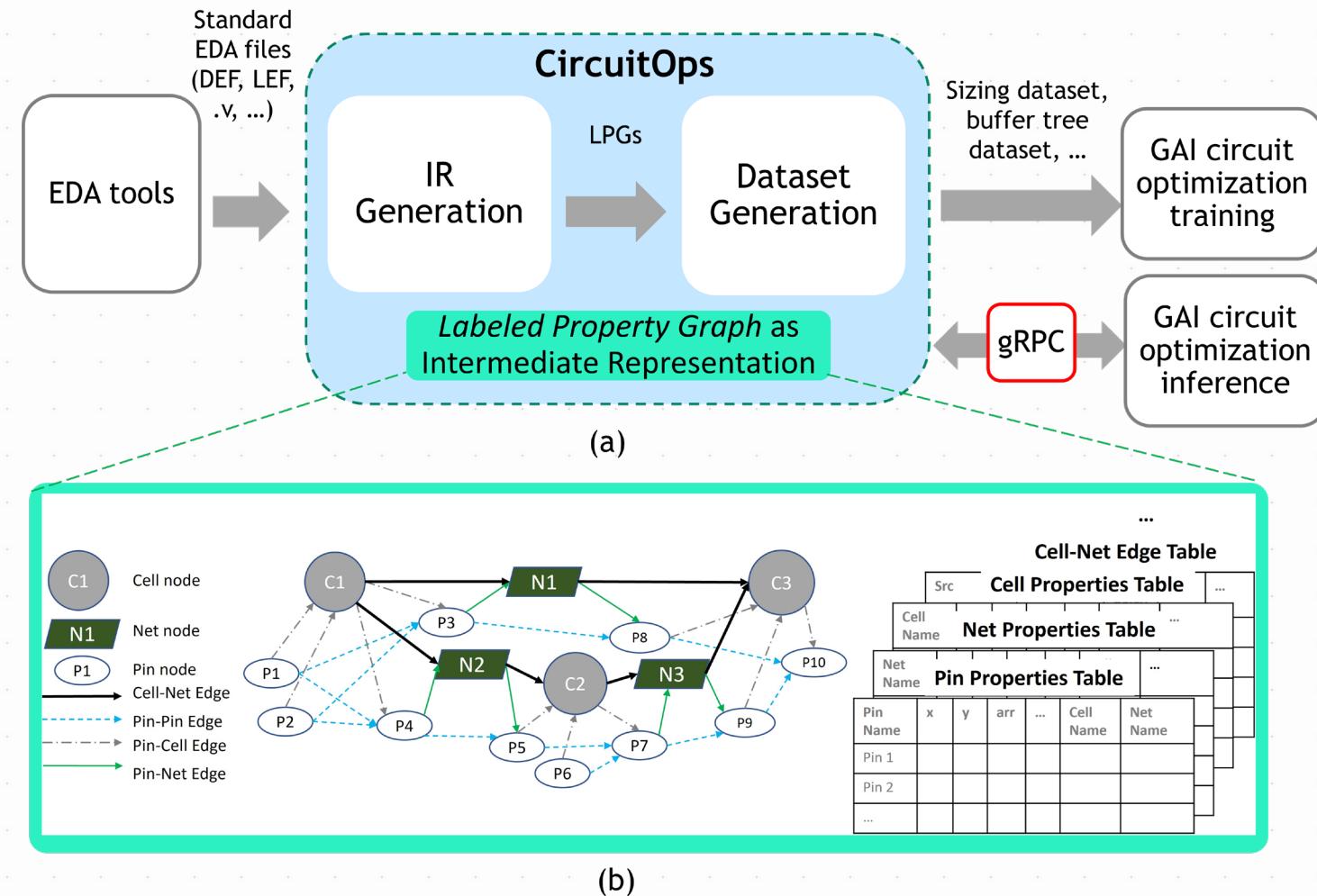


Enables an ML for EDA playground

- **User friendly data formats**
 - Standard ML-friendly data representation formats
- **Python APIs on existing EDA tools**
 - Enable data generation in ML-friendly format
- **Callbacks and EDA database writebacks from the ML environment**
 - Node and edge transformation to automatic EDA tool python API translation for novice EDA tool users

Nvidia CircuitOps Data Format

- ML-friendly data representation format
- Intermediate representation of EDA data as labeled property graphs (LPG) represented as deep graph library (DGL) object or graph tool which integrates easily with PyTorch
- Each node has associated relational tables that store node features, e.g., pin slack, transition, etc.



V. A. Chhabria, W. Jiang, A. B. Kahng, R. Liang, H. Ren, S. S. Sapatnekar and B.-Y. Wu, "OpenROAD and CircuitOps: Infrastructure for ML EDA Research and Education", ([.pdf](#)), Proc. IEEE VLSI Test Symposium, April 2024.

ML-centric APIs

1) `all_slacks = ord.get_property(list_pins, "rise_slack")`

where `all_slacks` is a numpy array

2) `graph_design = ord.get_netlist(list_insts, properties)`

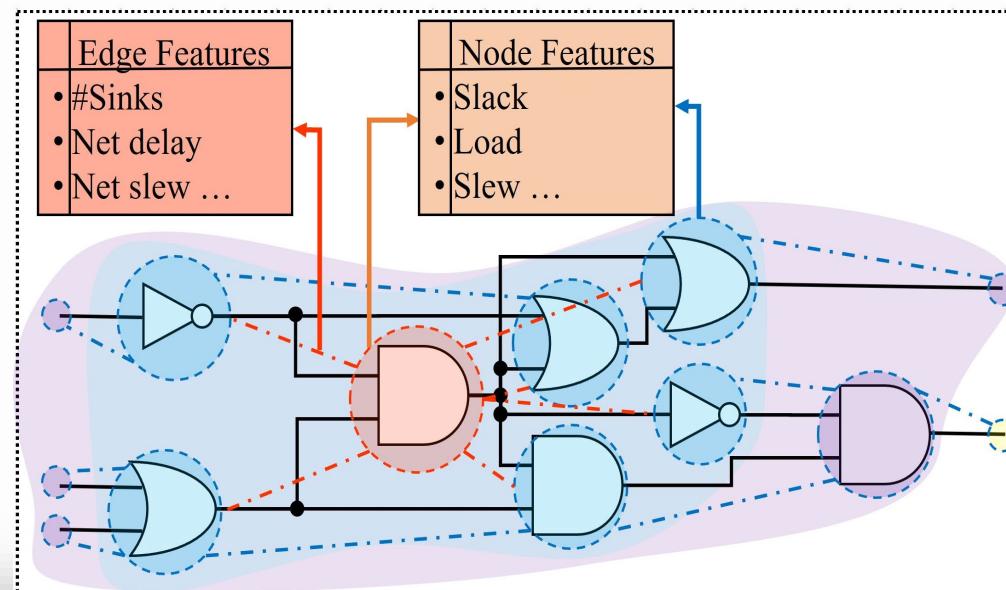
where `graph_design` is a DGL graph object where all nodes are instances in `list_insts` annotated with `properties` as node/edge features

3) `cong_map = ord.get_map(map=congestion, resolution=1um)`

where `cong_map` is a 2D numpy array representing a heat map

Examples of image-based
ML data extraction

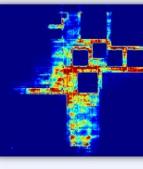
Examples of graph-based data
extraction: node, edge features



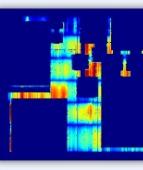
Congestion



DRC violations



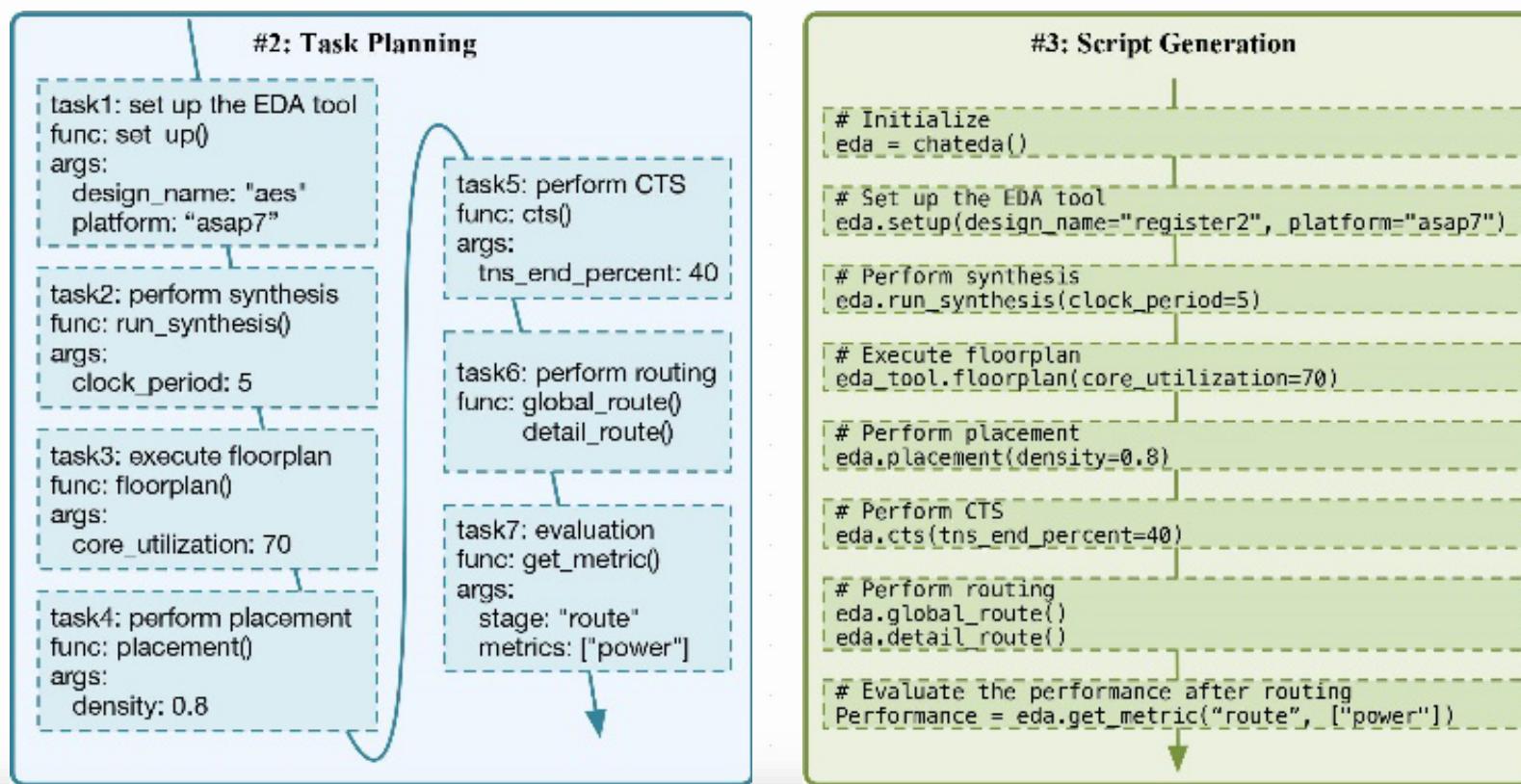
IR drop



LLM-based EDA Agent: ChatEDA

#1. User Requirement

For the design named “aes” on the platform “asap7”, please perform synthesis with a clock period of 5, followed by floorplan with a core utilization of 70%. Then, execute placement with a density of 0.8. Next, proceed with CTS to fix 40% of violating paths. Finally, evaluate the performance after routing using “power” metric.

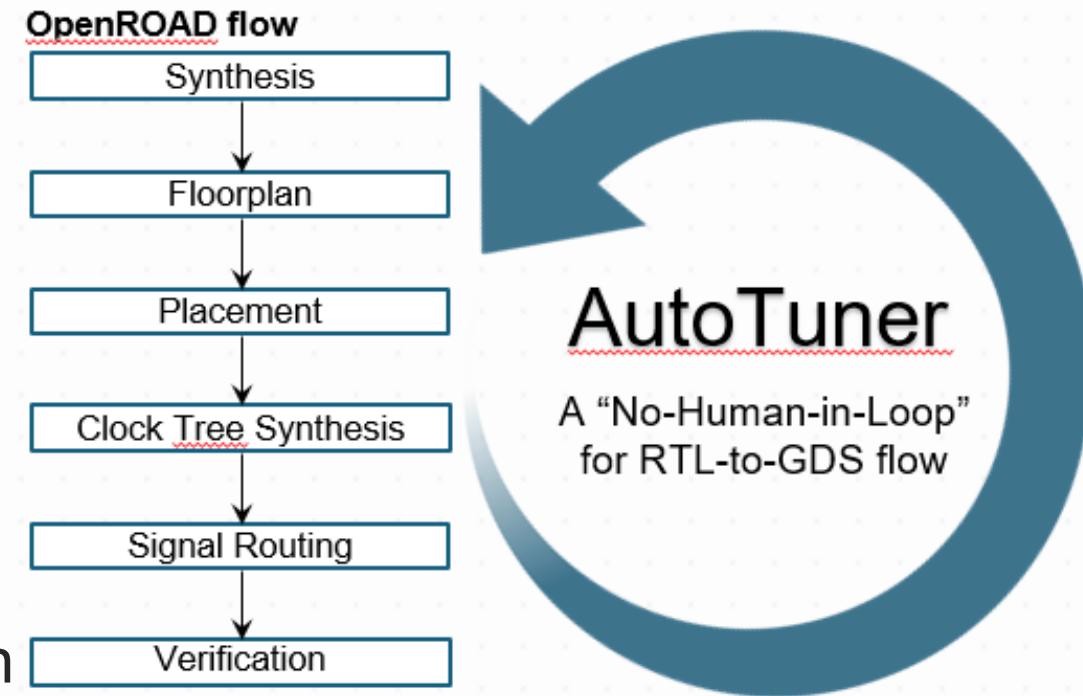


Workflow

1. (user) natural language input
2. (ChatEDA) task planning
3. (ChatEDA) script generation
4. (OpenROAD) task execution

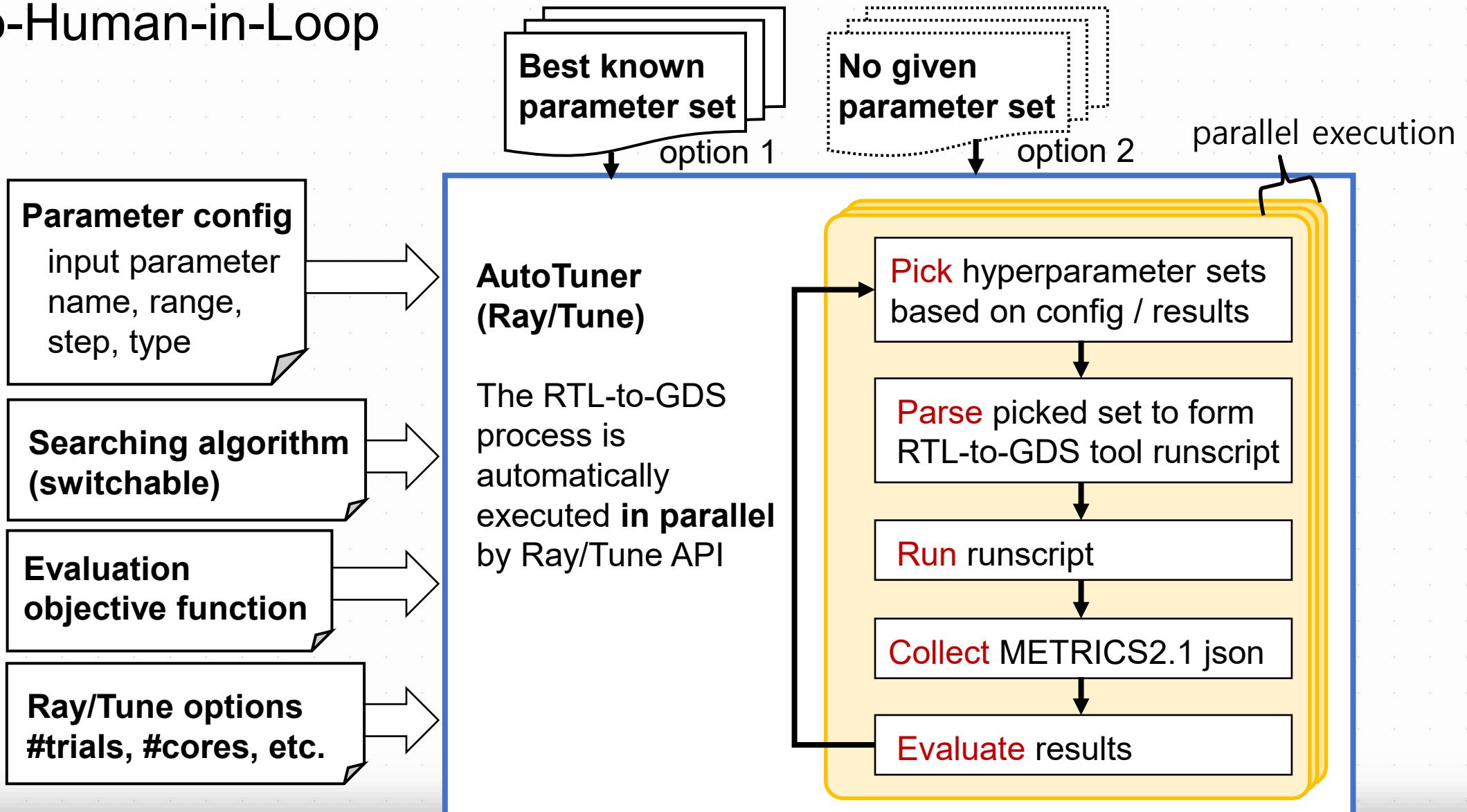
“No-Human-in-Loop” Flow Parameter Autotuning

- Automatic, iterative tuning to improve user-defined **score** within a given hyperparameter range space
- Interface: Python packages Ray/Tune
- Algorithms: HyperOpt, PBT, Optuna, Nevergrad, Ax, random search
- Advantages:
 - No need for pre-existing big data
 - Fewer trials needed than with, e.g., grid search
 - Powerful parallelization management: core/thread management, external server usage, visualization
 - Good outcomes within reasonable schedule, compute budgets



Flow Parameter AutoTuner – Architecture

- No-Human-in-Loop



Hyperparameter Space: SkyWater 130HD, ibex

- Tech: Skywater 130nm HD
- Design: ibex
- Tested Algorithms
 - HyperOpt
- Hyperparameter config <name, type, minmax, step>

Assuming full factorial combinations,
1,058,298,150
= #possible combinations!

```
1 {  
2     "GP_PAD": {"type": "int", "minmax": [0,4], "step": 1 },  
3     "DP_PAD": {"type": "int", "minmax": [0,4], "step": 1 },  
4     "LAYER_ADJUST": {"type": "float", "minmax": [0.1,0.7], "step": 0 },  
5     "PLACE_DENSITY": {"type": "float", "minmax": [0.1,1.0], "step": 0 },  
6     "FLATTEN": {"type": "int", "minmax": [0,1], "step": 1 },  
7     "PINS_DISTANCE": {"type": "int", "minmax": [1,3], "step": 1 },  
8     "CTS_CLUSTER_SIZE": {"type": "int", "minmax": [10,40], "step": 1 },  
9     "CTS_CLUSTER_DIAMETER": {"type": "int", "minmax": [80,120], "step": 1 },  
10    "GR_OVERFLOW": {"type": "int", "minmax": [1,1], "step": 0 }  
11 }
```

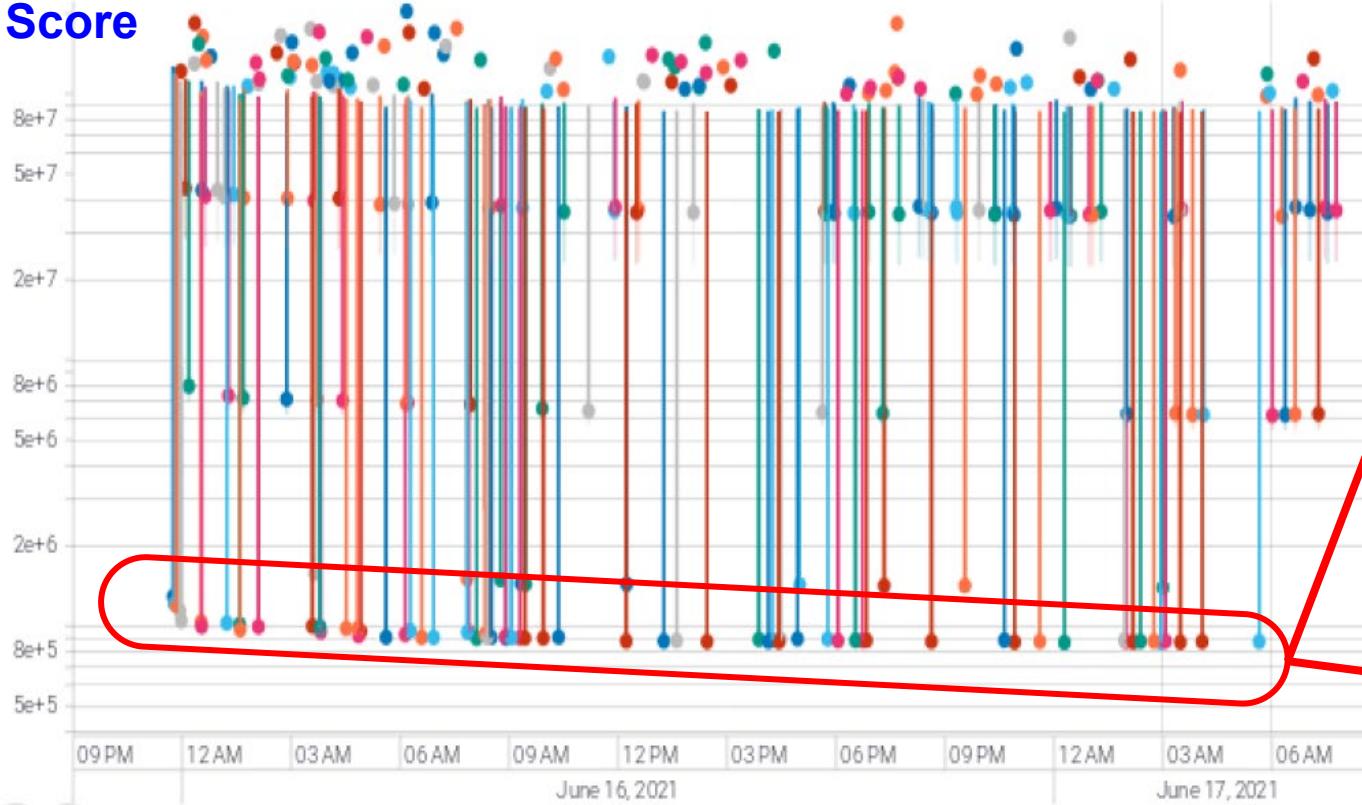
- When type is int and step = 0, it means constant value
- When type is float and step = 0, it means continuous range

TensorBoard Visualization: SkyWater 130HD, ibex

- GUI integration with **TensorBoard**
- Score results versus Wall Time

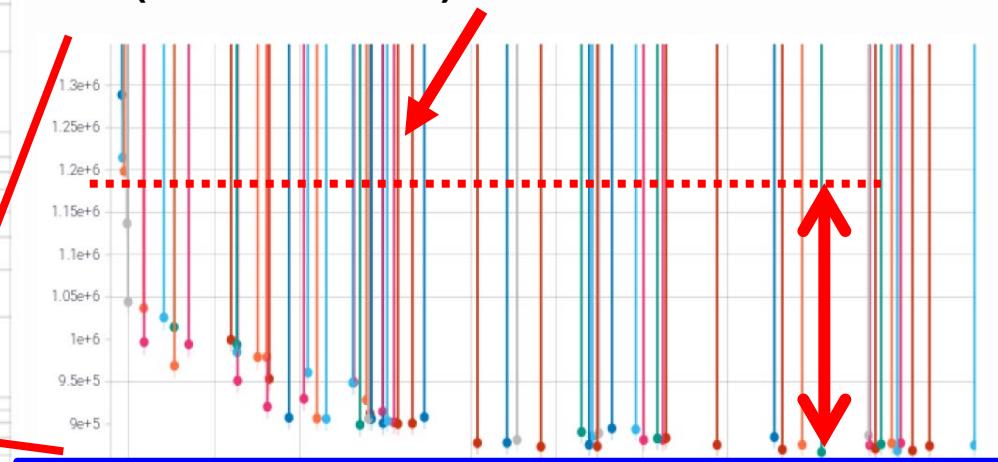
User-defined

Score



Dots = trials

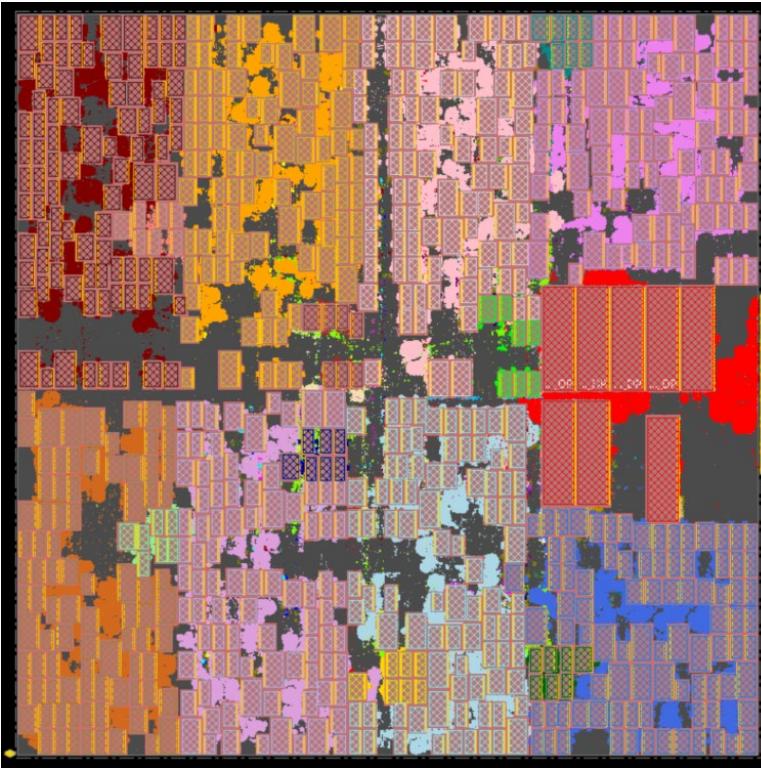
Default flow score = 1,174,346
Our Best Score = 855,373
(370 trials in total 500 #trials)
(less is better)



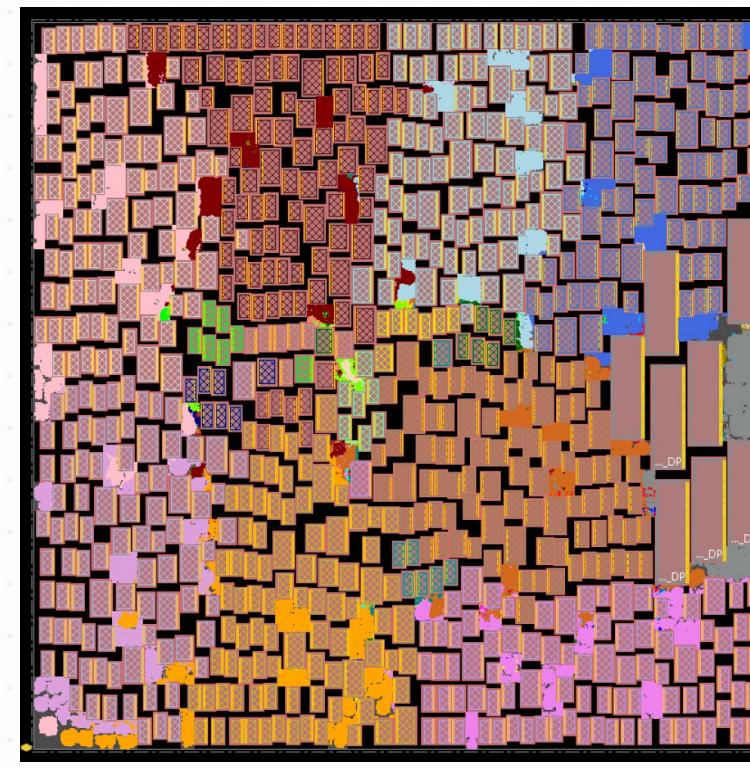
Improvement

WL 1003801um → 843258um (**-16%**)
Effective CP 20.935ns → 16.185 ns (**-23%**)
Total power 0.024 W → 0.0133 W (**-45%**)

High-quality Engines: Hier-RTLMP Macro Placer



Hier-RTLMP (postRoute)



Comm Macro Placer (postRoute)

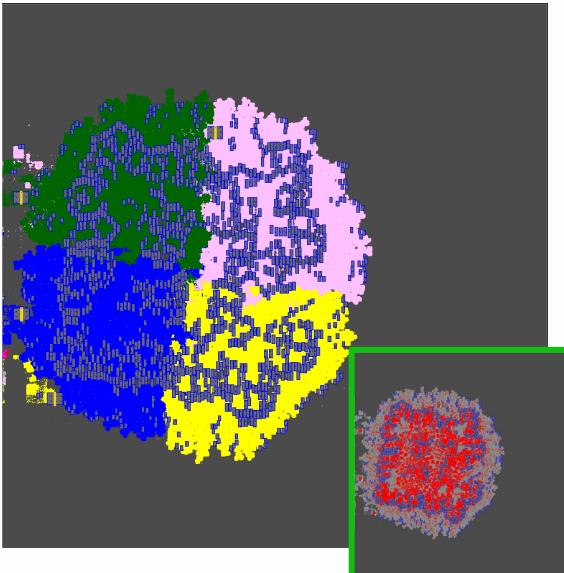
TABLA01 AI accelerator
in GF 12nm, **760** macros

Macro Placer	Std Cell Area (mm ²)	Power (mW)	WNS (ns)	TNS (ns)
Hier-RTLMP	0.160	640	-0.085	-0.417
Comm	0.165	689	-0.370	-92.246

High-quality Engines: GPU-Accelerated Global Placer

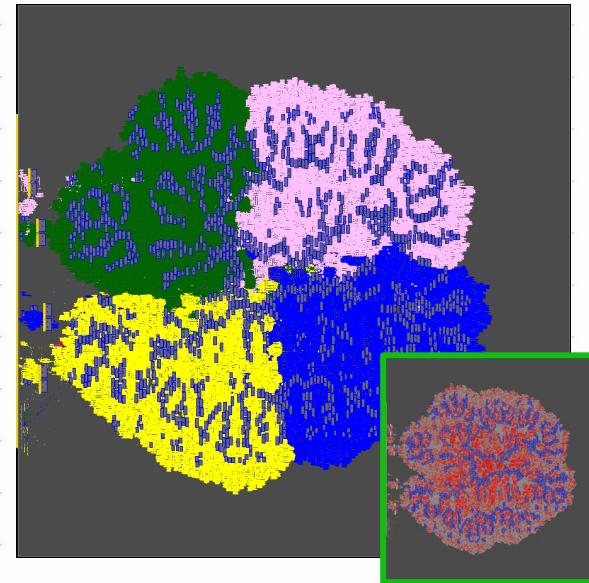
	Runtime (s)	HPWL (m)	eGR WL (m)
RePIAce	65381	325	404
Comm	24561	414	478
DG-RePIAce	1808	327	407

OpenROAD RePIAce



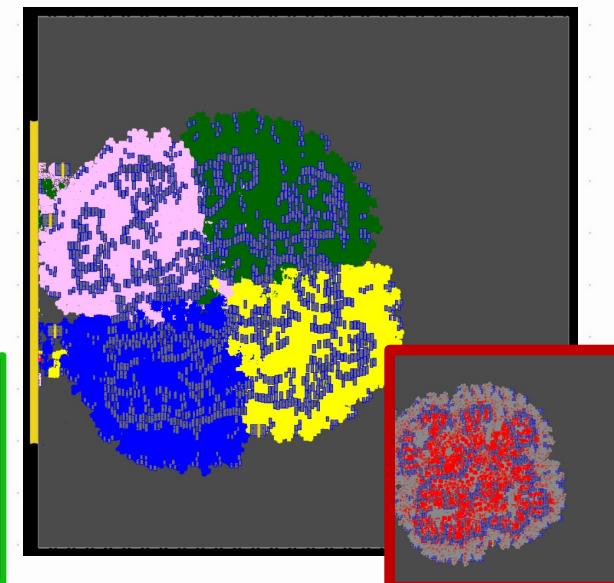
2.79% H + 2.04% V

Commercial



1.51% H + 1.01% V

DG-RePIAce



2.37% H + 1.68% V

Testcase: MemPool Cluster, ETH Zurich (**9.5M cells, 1296 macros in NG45**)



- Main
- Overview
- Infrastructure
 - Metrics List
 - Dictionary
- Publications
 - Papers
 - Presentations
 - GSRC Presentations
- Codes
- Theses
- Links
 - Conferences & Workshops
 - Softwares
 - Others

The METRICS Initiative

Recent Updates

- [Survey \(1st draft\)](#) for design quality and productivity ([the multiple-choice version](#))
- [Reduced survey \(2nd draft\)](#) for design quality and productivity that is distributed at June 2001 GSRC workshop
- [Updated survey \(3rd draft\)](#) for design quality and productivity that reflects the discussion at June 2001 GSRC workshop
- [Workshop notes](#) for METRICS discussion at June 2001 GSRC workshop
- [List of prediction/estimator models](#) enabled by METRICS System NEW
- [DAC02 Birds-of-a-Feather meeting summary](#) (June 12, 2002) NEW

- **METRICS 1.0** (1999; DAC00, ISQED01)
 - “Measure to Improve” <http://vlsicad.ucsd.edu/GSRC/metrics>
- **METRICS 2.0 (WOSET-2018)** was proposed as an update of **METRICS 1.0**
- **METRICS2.1** is proposed as a standard, with a concrete realization in OpenROAD

METRICS2.1: Standard Naming !

<https://github.com/ieee-ceda-datc/datc-rdf-Metrics4ML>

- **Problem:** “Tower of Babel” (names, formats that are all different and proprietary)
- **Solution:** “METRICS”
 - General and extensible
 - Syntax and semantics to support future addition of new metrics
- **No ambiguity!!!**
 - Any desired measurement must map to a unique METRICS2.1 metric
 - Every METRICS2.1 metric must map to a unique interpretation as a measurement
 - Two-way mapping is crucial to avoid future confusion
- Can also capture the same metric at different stages of the design flow
- **Free, open and frictionless – agnostic to EDA provider**

METRICS2.1 Examples

<https://github.com/ieee-ceda-datc/datc-rdf-Metrics4ML>

Sample metrics

Metric	Description
<i>timing_setup_wns</i>	Setup worst negative slack in the design
<i>timing_setup_wns_clock:clk_a</i>	Setup worst negative slack for clock “clk_a” in the design
<i>timing_setup_wns_analysis_view:slow</i>	Setup worst negative slack for analysis view “slow”
<i>power_total</i>	Total power consumption
<i>power_leakage</i>	Total leakage power
<i>power_leakage_clock</i>	Total leakage power in the clock network

Many applications: data for machine learning, CI/CD infrastructure for software quality, ...

A Comment on SP25 Classes

- **ECE 260C (Special Topics)**: focus is on RTL-to-GDS implementation with open-source enablement (EDA, PDK, Design)
 - See and modify what is inside the EDA tool; use OS lever to improve outcomes
- **CSE 241A (cross-listed with ECE 260B)**: “classic” 260B/241A with commercial EDA tools
 - “Job-readiness” in ASIC PD + what/how tools think inside, with more CSE flavor
- **CSE 291 H00 “Topics in ML for Chip Design”**: AI/ML in EDA and IC Design
 - Must use open source to explore AI/ML ! → similar tool, flow context as 260C

