Intro, Logistics, and OpenROAD

UCSD ECE 260C Spring 2026

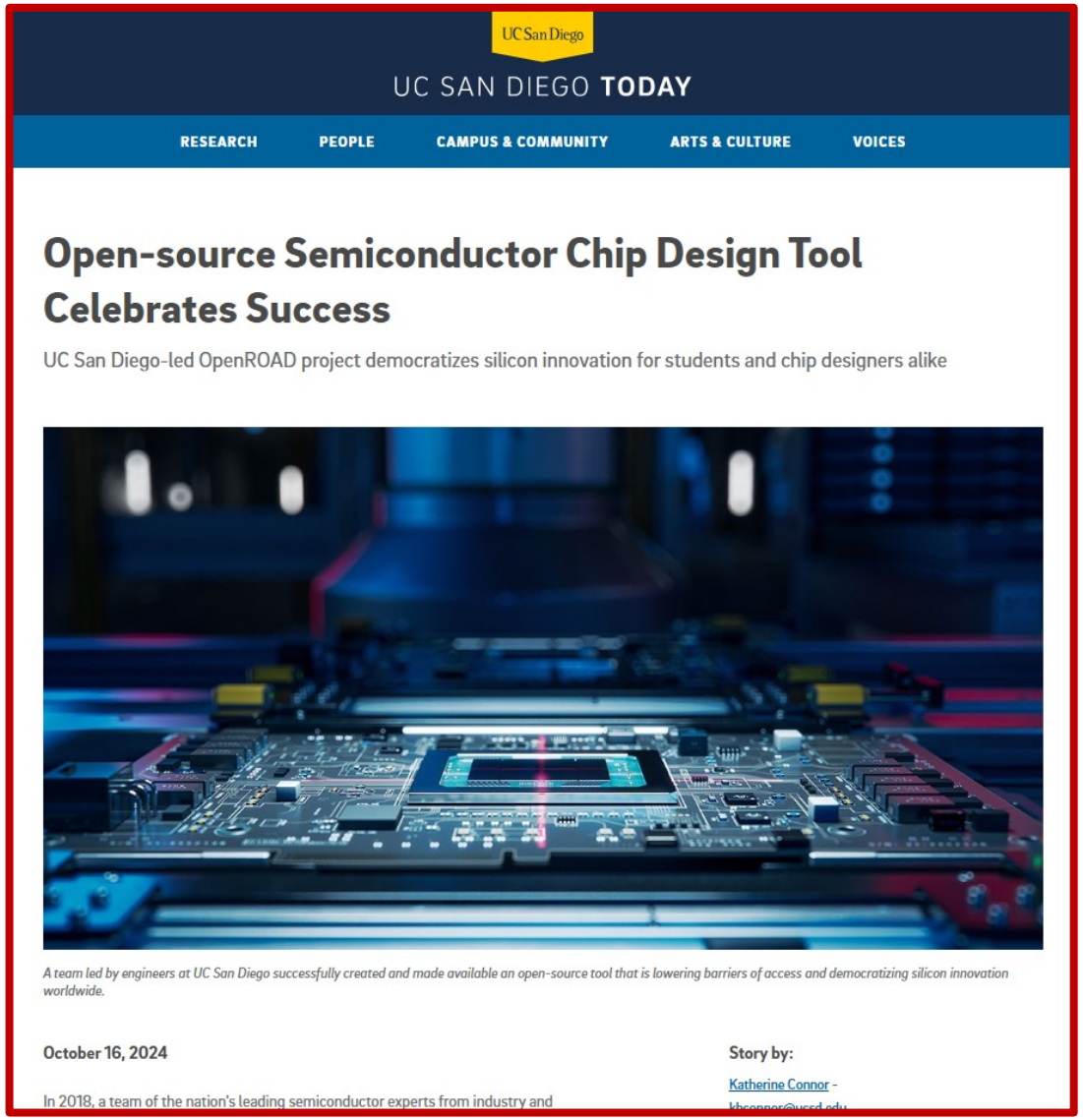
Prof. Andrew B. Kahng

abk@ucsd.edu

https://vlsicad.ucsd.edu

ECE 260C SP26: “VLSI Special Topics”

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



UC San Diego

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

Story by:

Katherine Connor -

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

Scope

- Understand the internals and algorithms that drive implementation
- Explore the expanding field of Open-Source EDA software
  - Learn how OpenROAD, Yosys and other tools provide alternatives or complements to commercial EDA (Cadence, Synopsys, Siemens) tools
  - Awesome List: https://github.com/aolofsson/awesome-opensource-hardware
- Learn how to apply scripting to achieve better results, bespoke goals
  - Open source enables deeper in-source changes
  - Industry demand for early design space exploration, pathfinding flows
- Apply learned methodologies in the context of System-on-Chip design

Logistics

- Lectures are Tues/Thurs from 5:00 pm to 6:20 pm in Center Hall 115
  - Lecture slides and associated reading links on the website

- There will be 5 labs (30% of your grade) and 4 homeworks (15%)
  - Lab 0 releases today
- There will be two mini-projects (30% of your grade)
  - More details when MP1 releases in Week 4
- We will hold an in-class midterm on May 12th (25% of your grade)

Course Website

- Course website https://abkcourses.github.io/ece260c-sp26/
  - Schedule, slides, labs, homeworks, miniprojects, and links can be found there
  - Piazza will be used for Q&A and Announcements
  - Gradescope will be used for grading
- Stay up to date on the Schedule on the site and announcements on Piazza / via email

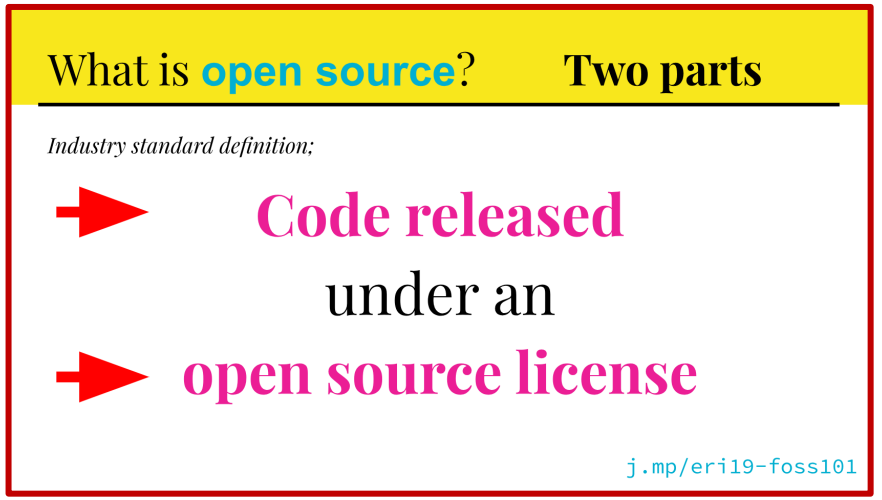
Go to the site
and sign up to Gradescope/Piazza now!

(Links can also be found on Canvas)

https://abkcourses.github.io/ece260c-sp26/

What is Open-Source EDA ?

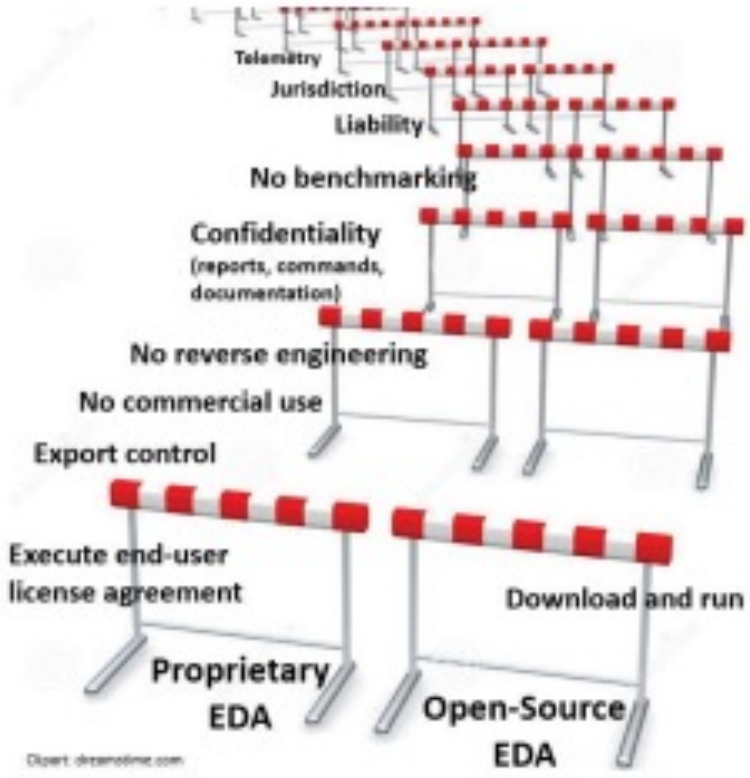
- EDA tools that satisfy the definition of open source
  - Tim Ansell, Open Source 101, 2019
- Freely usable, freely modifiable, and shareable
- License permits free redistribution, creation of derived works, and use by anyone for any purpose, in a technology-neutral manner
  - Transformative! → Linux,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



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

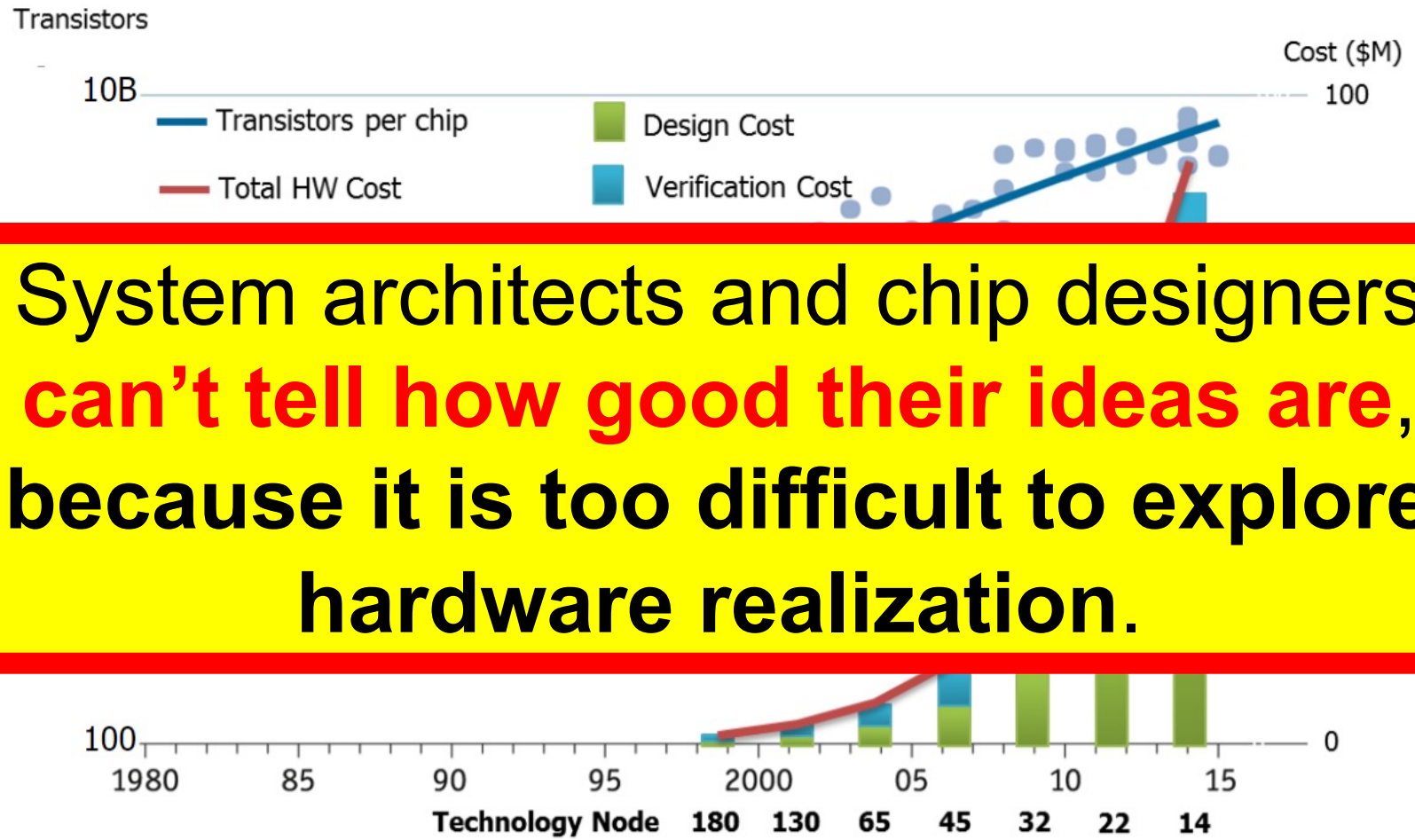
- Can you share your Tcl script with another user?
- Can you share code or write a tool that reads the same command syntax?
- Can you share the tool's output (gds, logfile, …)
- Can you compare it with another tool ("benchmarking")?
- Can you use the tool output to make a chip startup?
- Can you share 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)

Crisis of Design: Cost, Expertise, Risk



System architects and chip designers **can't tell how good their ideas are**, **because it is too difficult to explore hardware realization**.

Andreas Olofsson, keynote, Intl. Symp. on Physical Design, March 2018

Crisis of Design: Cost, Expertise, Risk

Open-source EDA
is our best hope to **fix**
the Crisis of Design

Andreas Olofsson, keynote, Intl. Symp. on Physical Design, March 2018

Open-source EDA: Innovation

- **Scaling** = *getting better results with less resources*
  - People, money, time, energy, area, …
  - *Moore's Law: 1 week = 1%*

- Scaling of **design innovation** demands scaling of EDA
  - More designers and designs → more EDA engineers
  - Differentiated design capability → bespoke EDA

Scale design, EDA and people

Open-source EDA: Technology

(DAC-2019 panel)

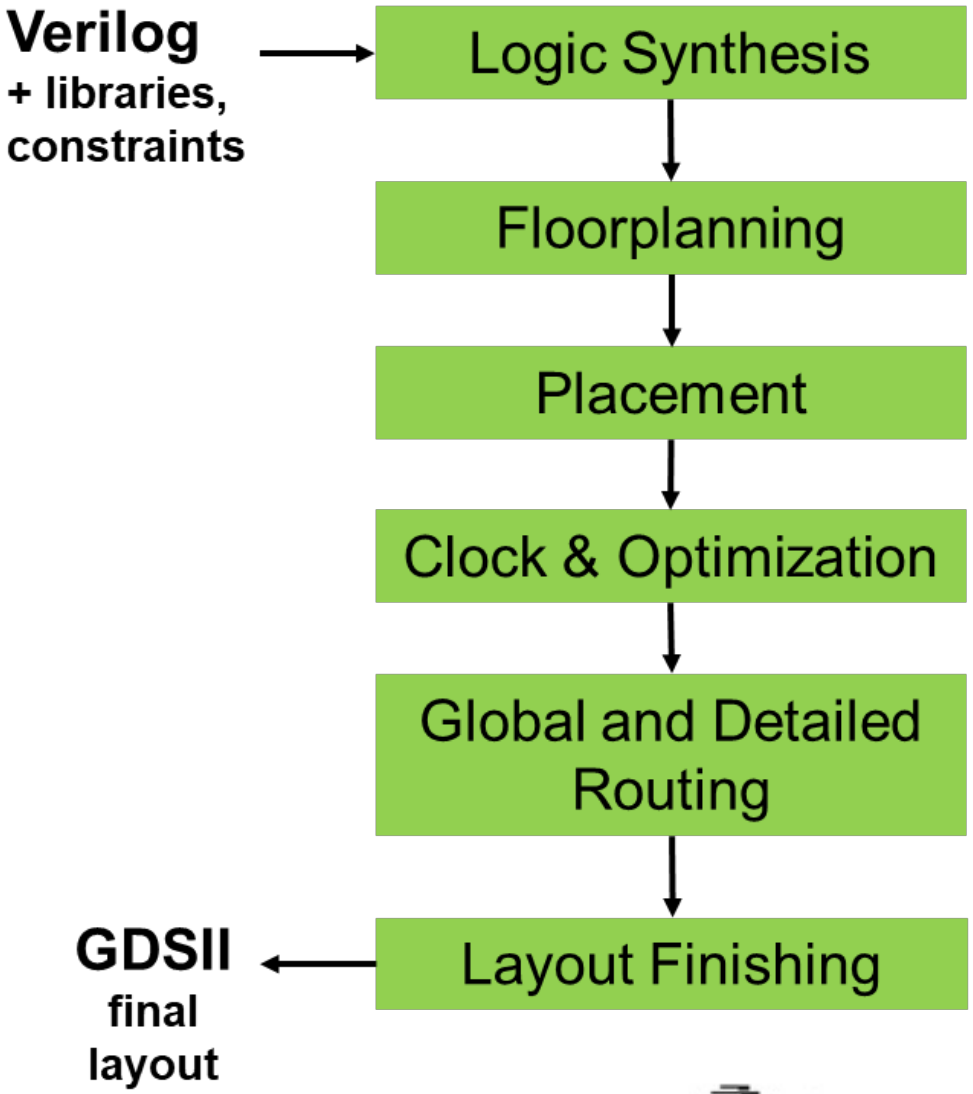
- **Clarity**
  - Leading edge and baselines become visible and well-defined
  - *Commercial EDA license terms: tools cannot be benchmarked*

- **Better science**
  - Advances are reproducible **and reusable**
  - *Avoid controversy → mature research culture*

- **Efficiency and velocity**
  - Less reinventing of wheels → field advances faster **and is more attractive**

Accelerate the advance of EDA technology

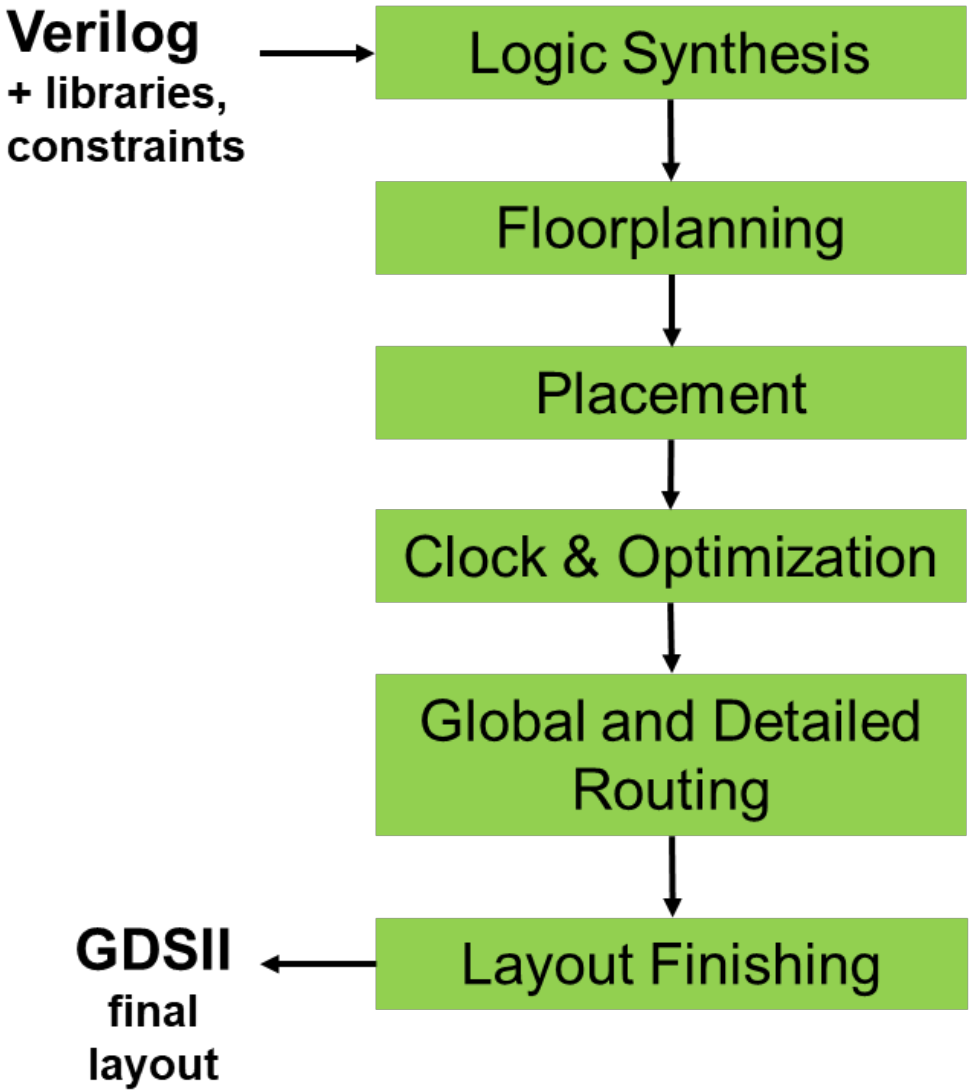
The OpenROAD Project, 2018-



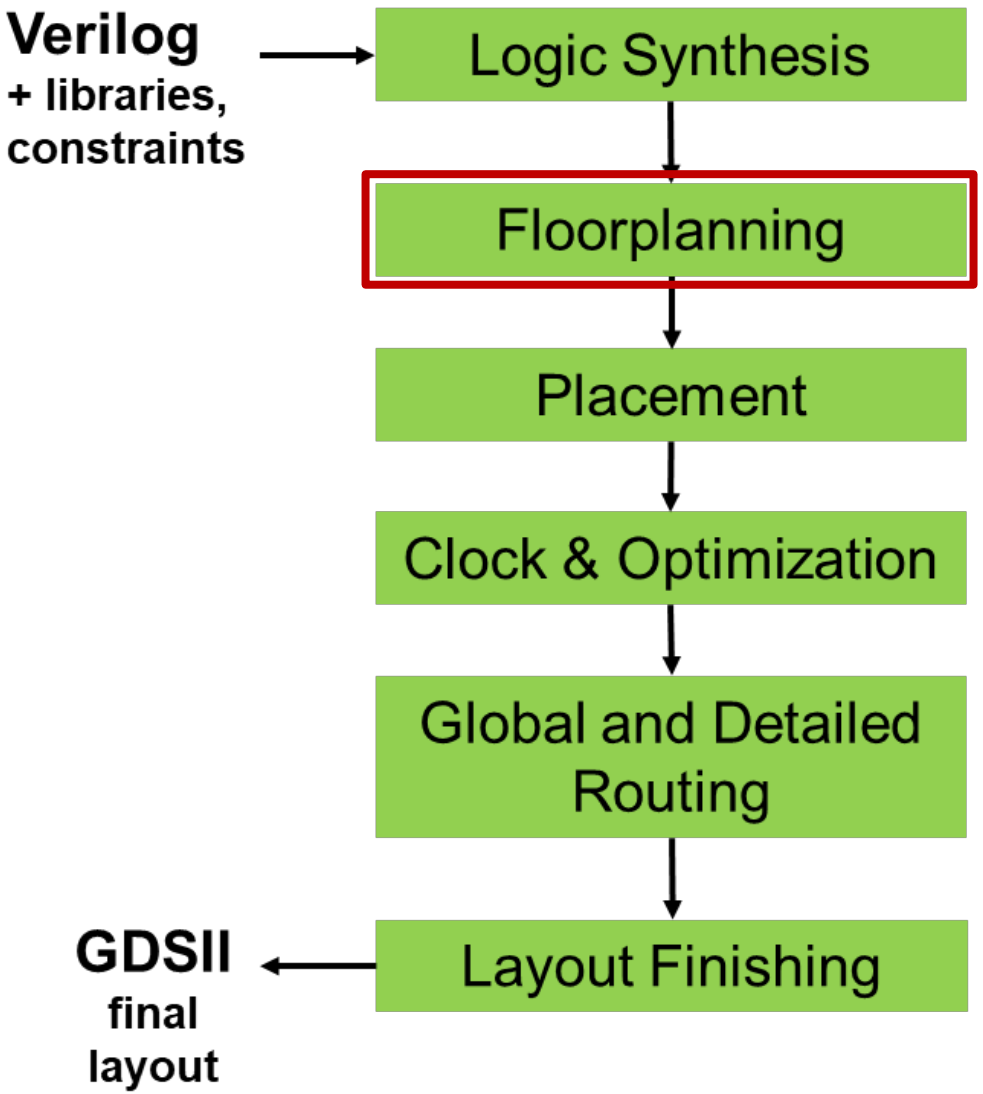
"**Foundations** and Realization of Open, Accessible Design"

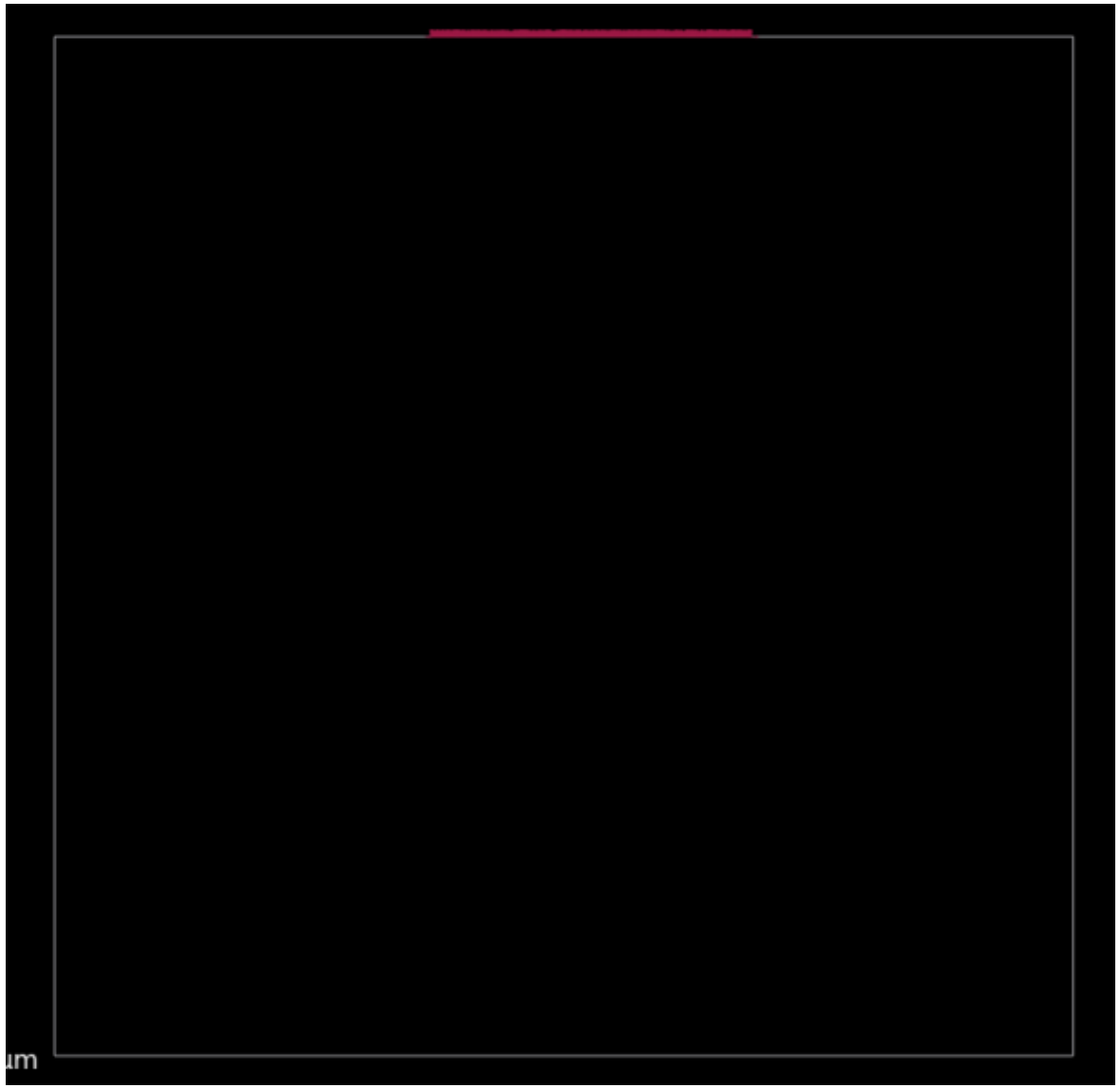
- U.S. DARPA project 2018-23
  - One of 12 projects in the "IDEA" program
- 24-hour, no-human-in-the-loop, tapeout-clean layout generation
  - In commercial FinFET nodes
  - **In permissive open source**

RTL-to-GDS Chip Implementation Flow

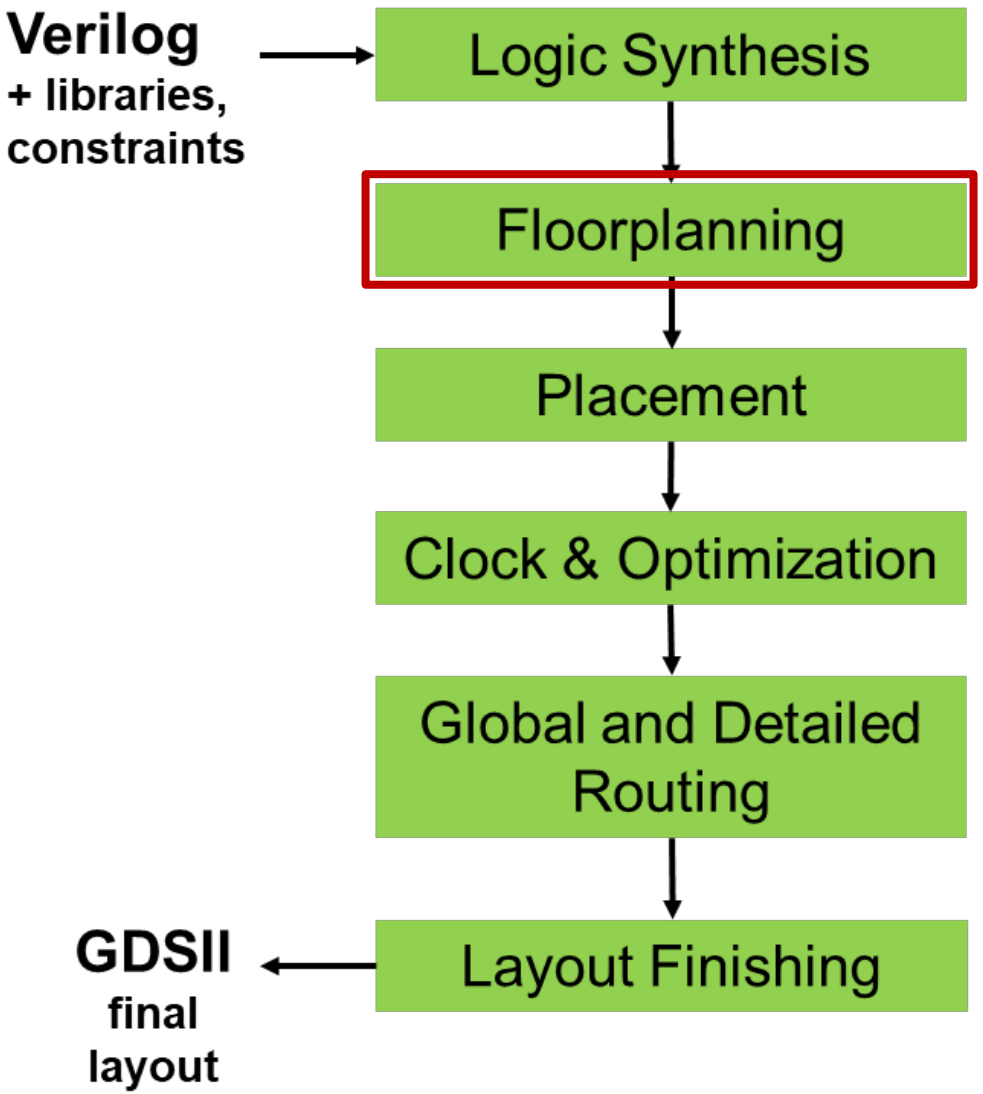


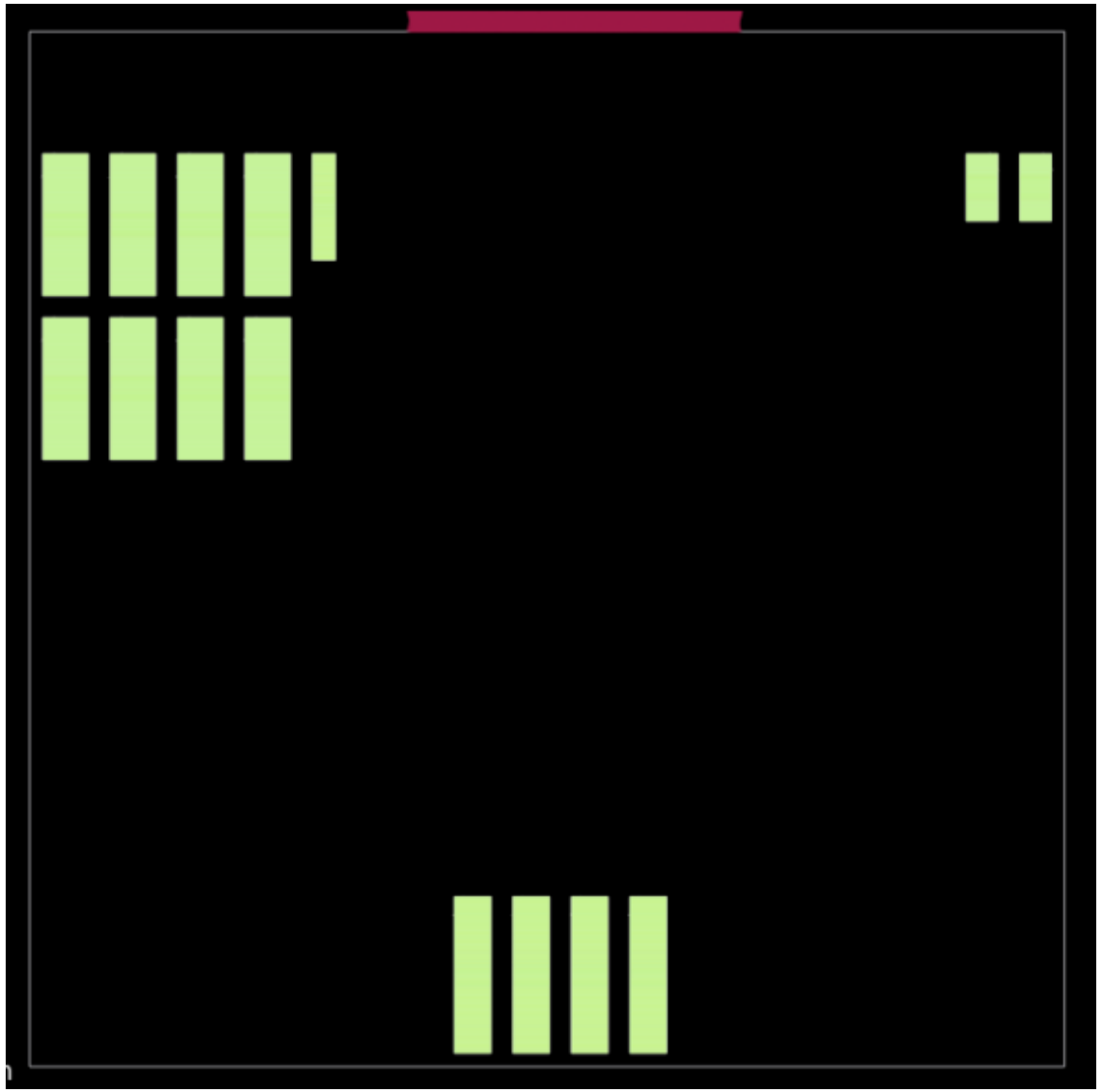
IO Placement (Example: foundry 12nm RISC core, “coyote”)



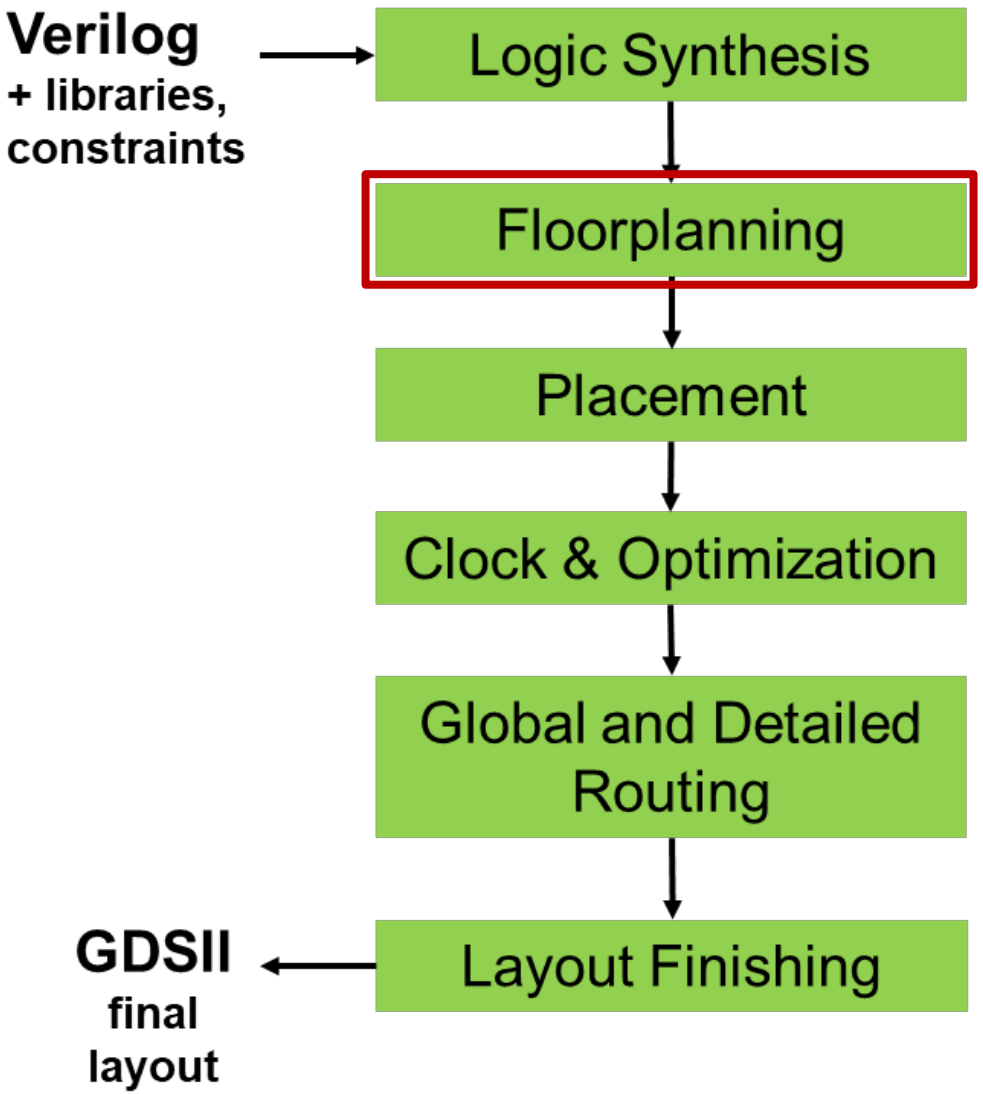


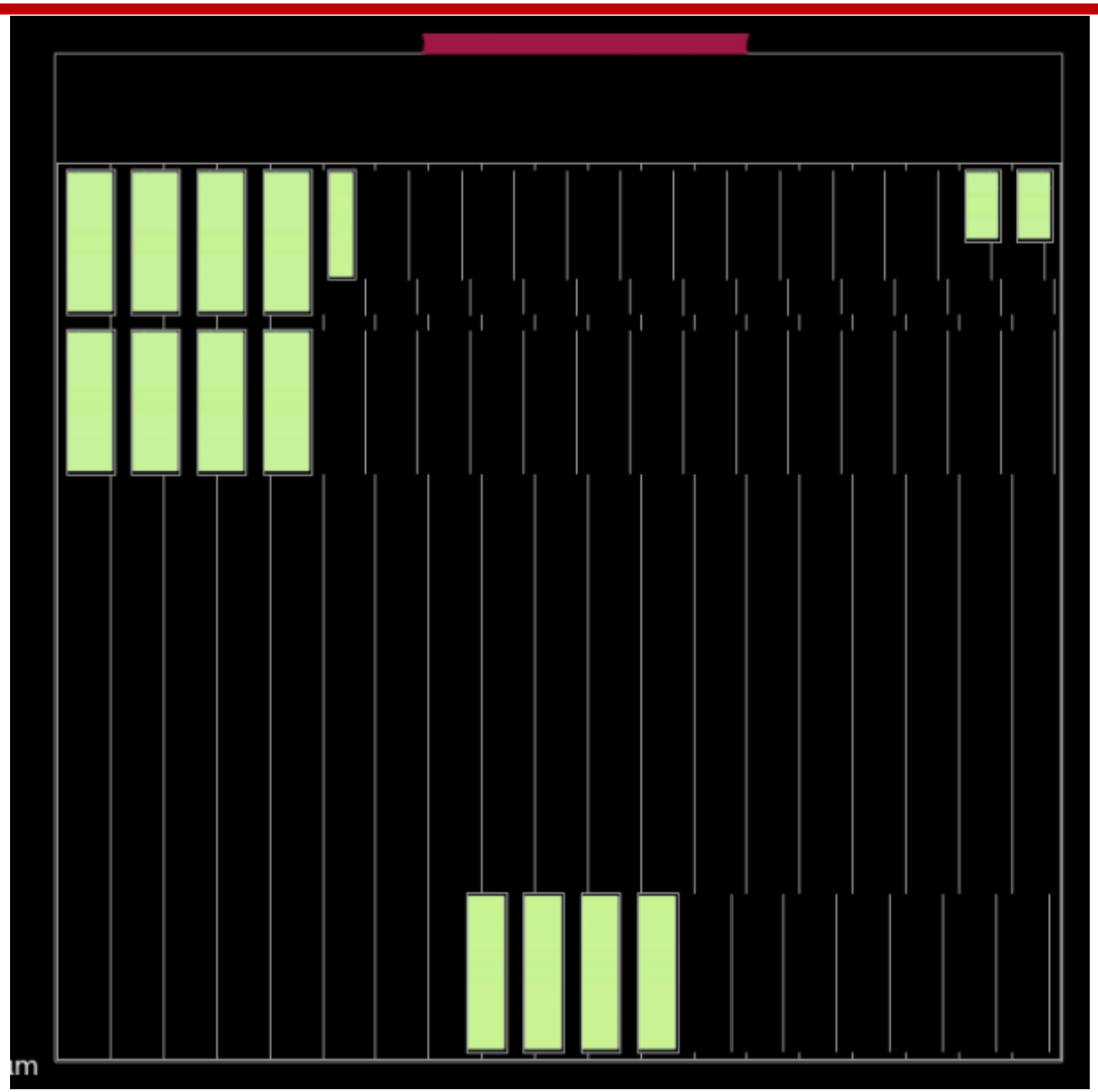
Macro Placement



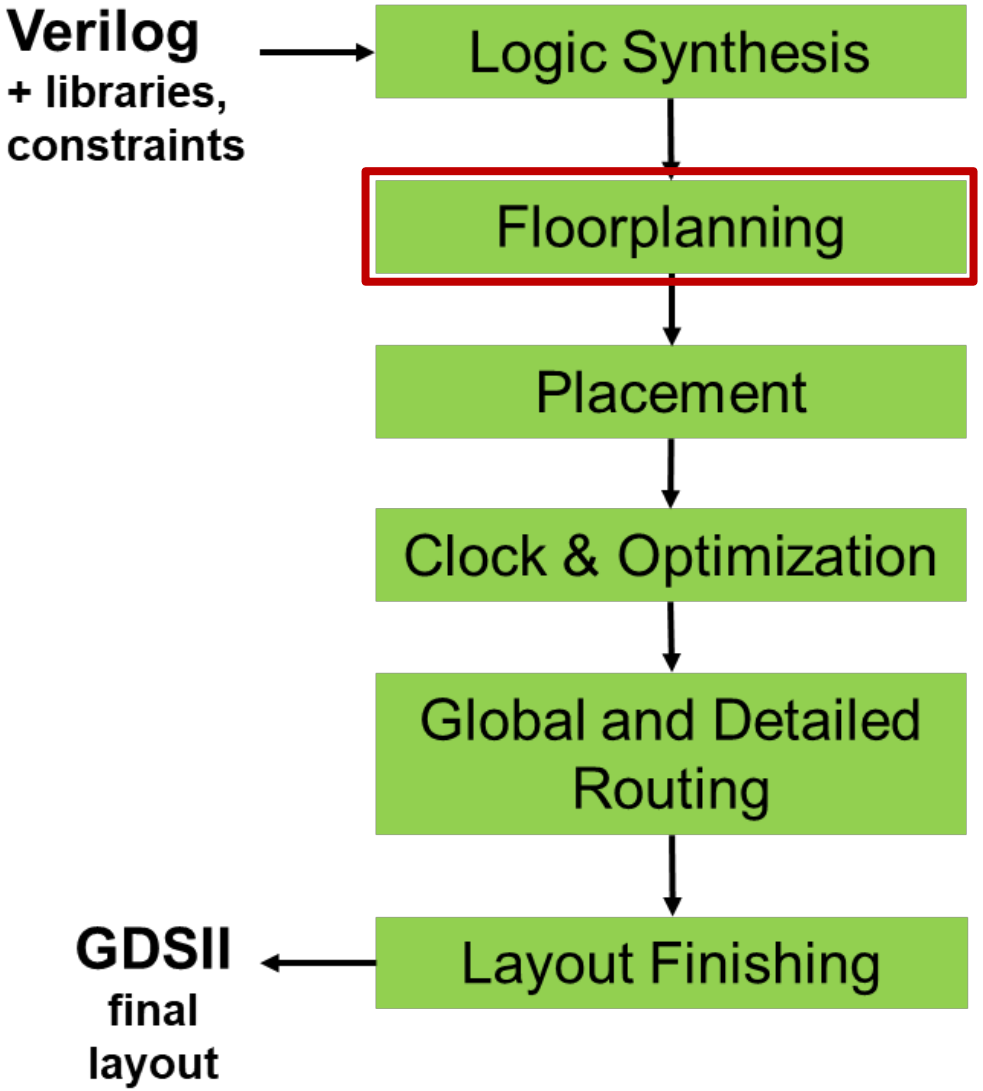


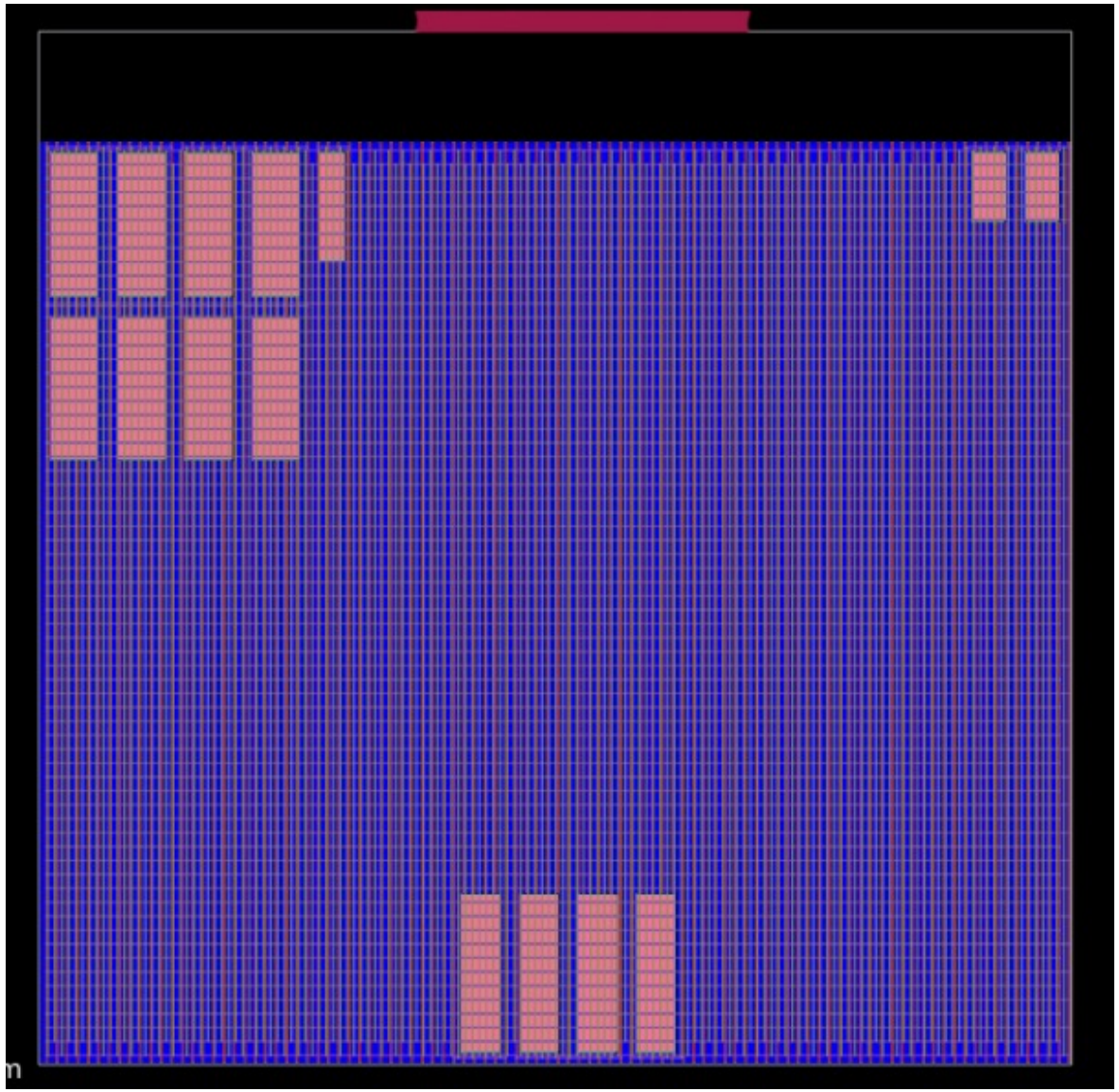
Tapcell (Well Tap) Insertion



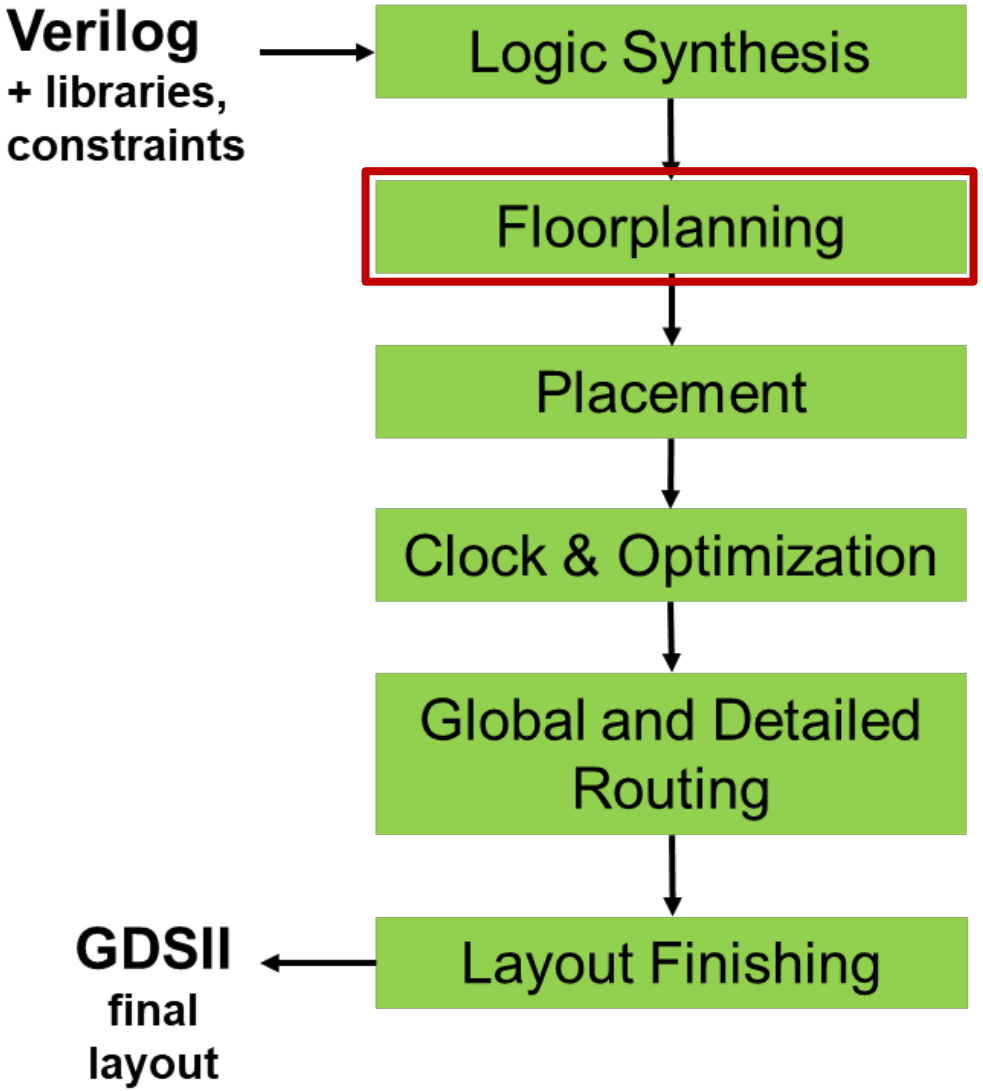


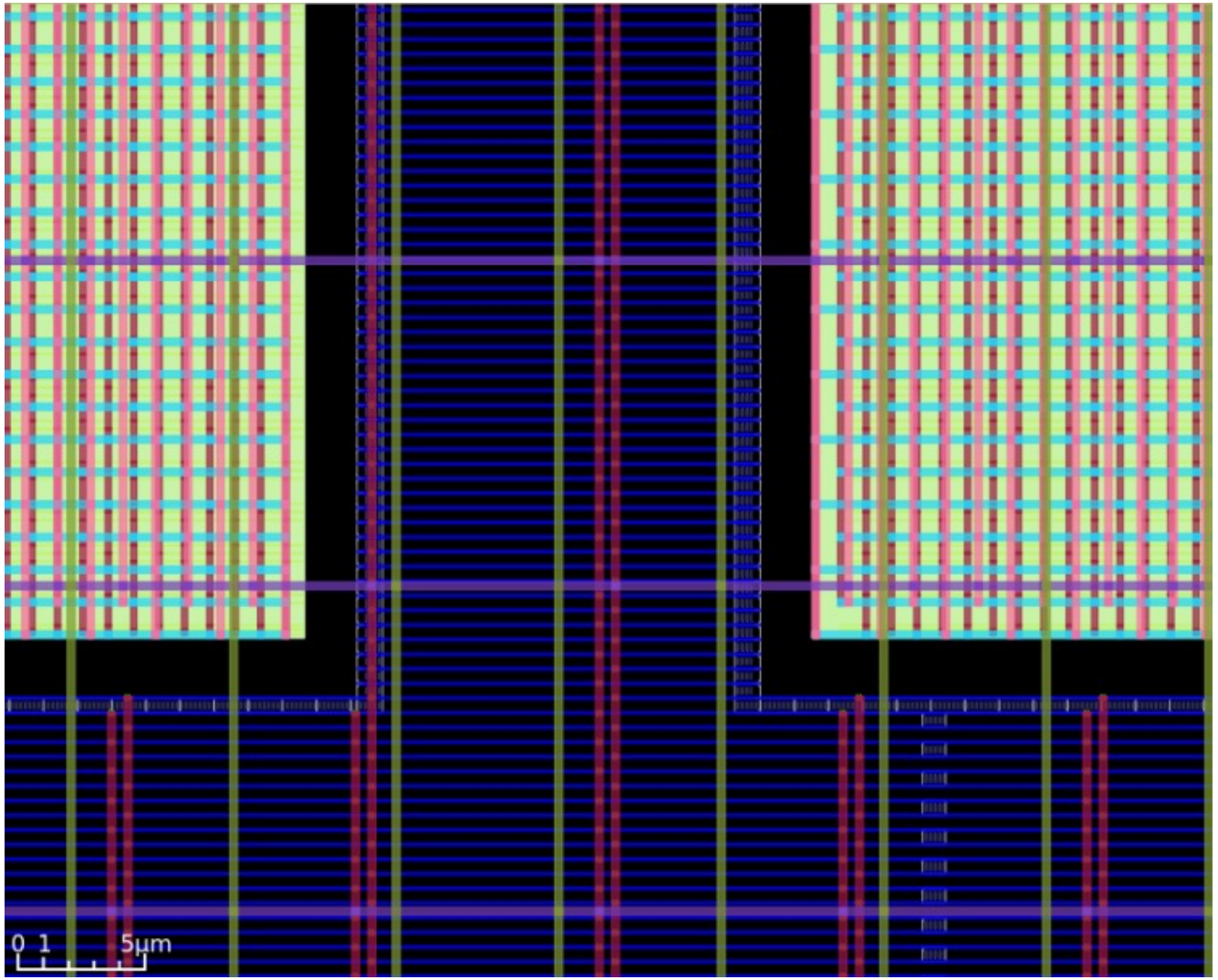
Power Delivery Network



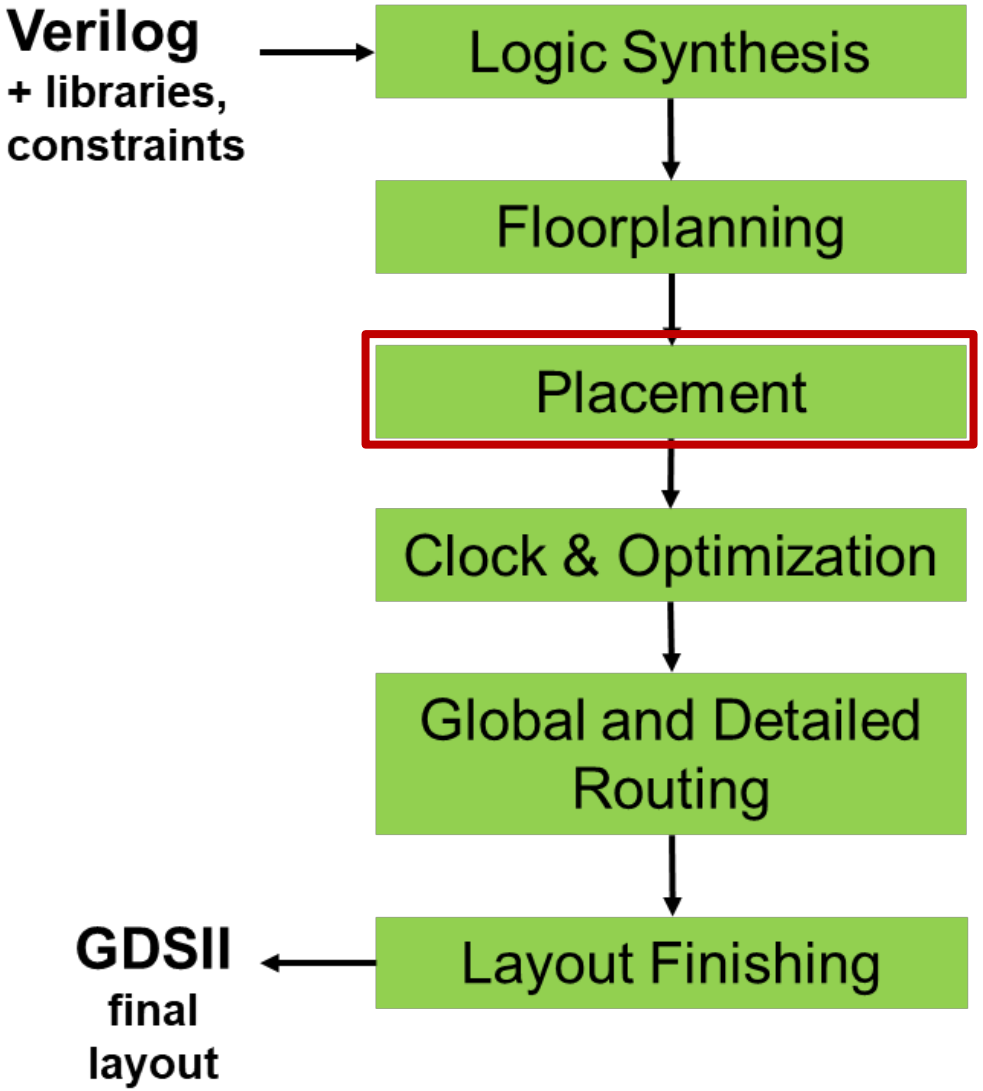


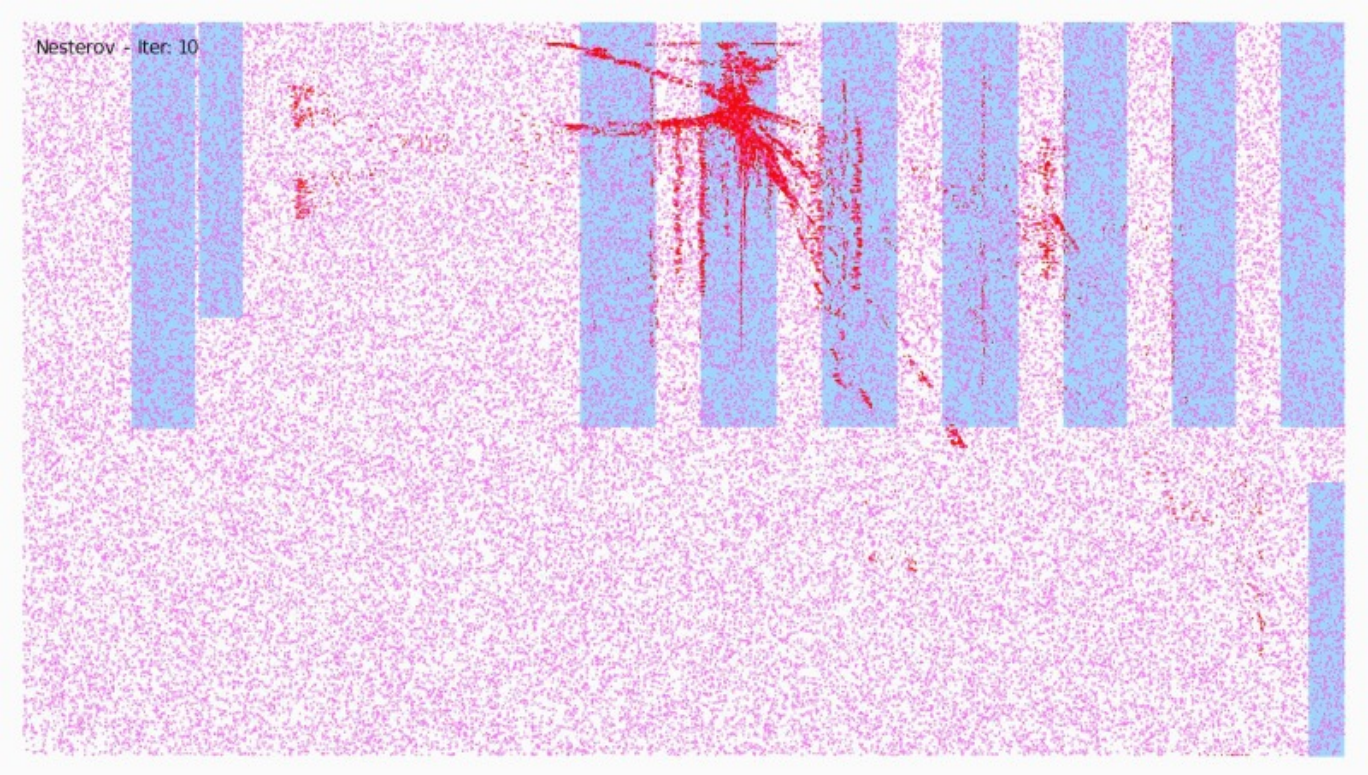
Power Delivery Network – Zoom-In



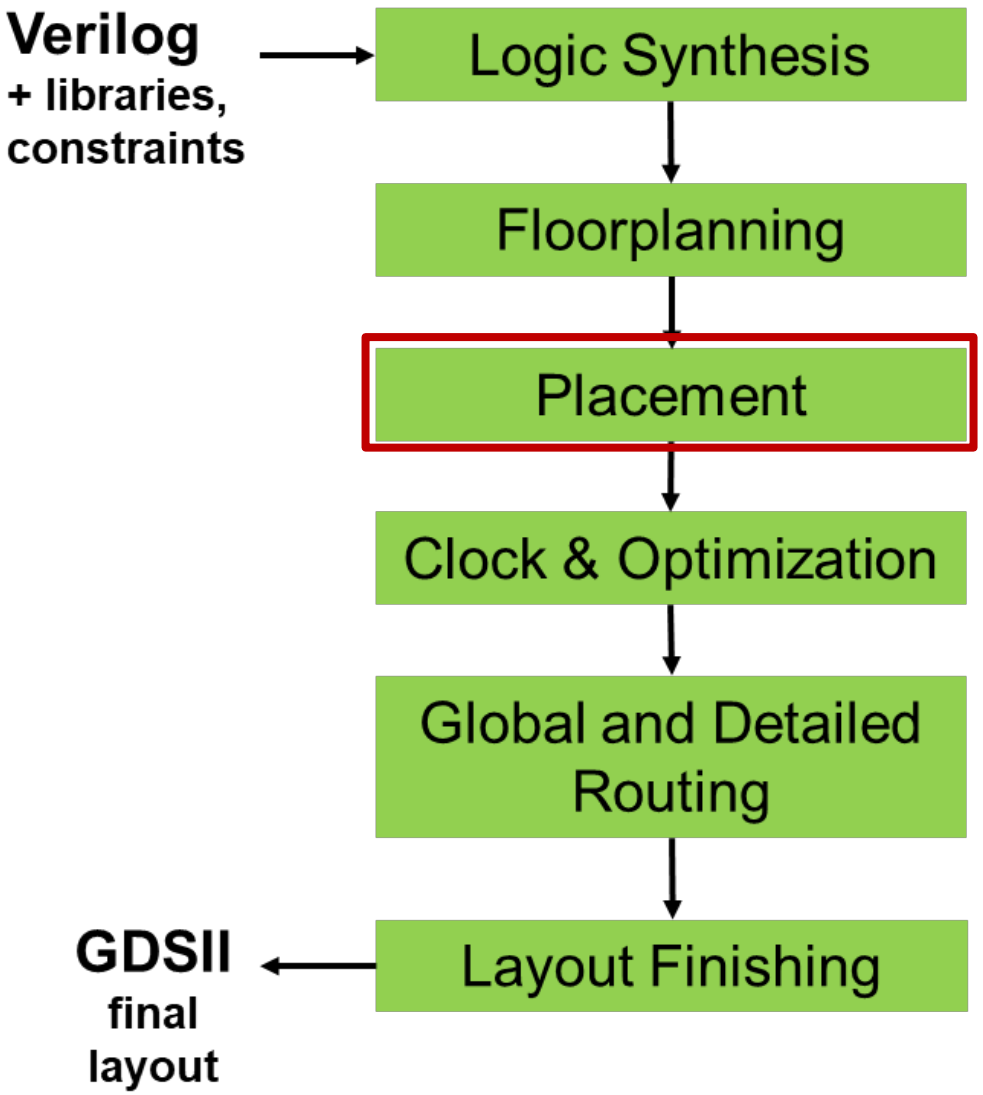


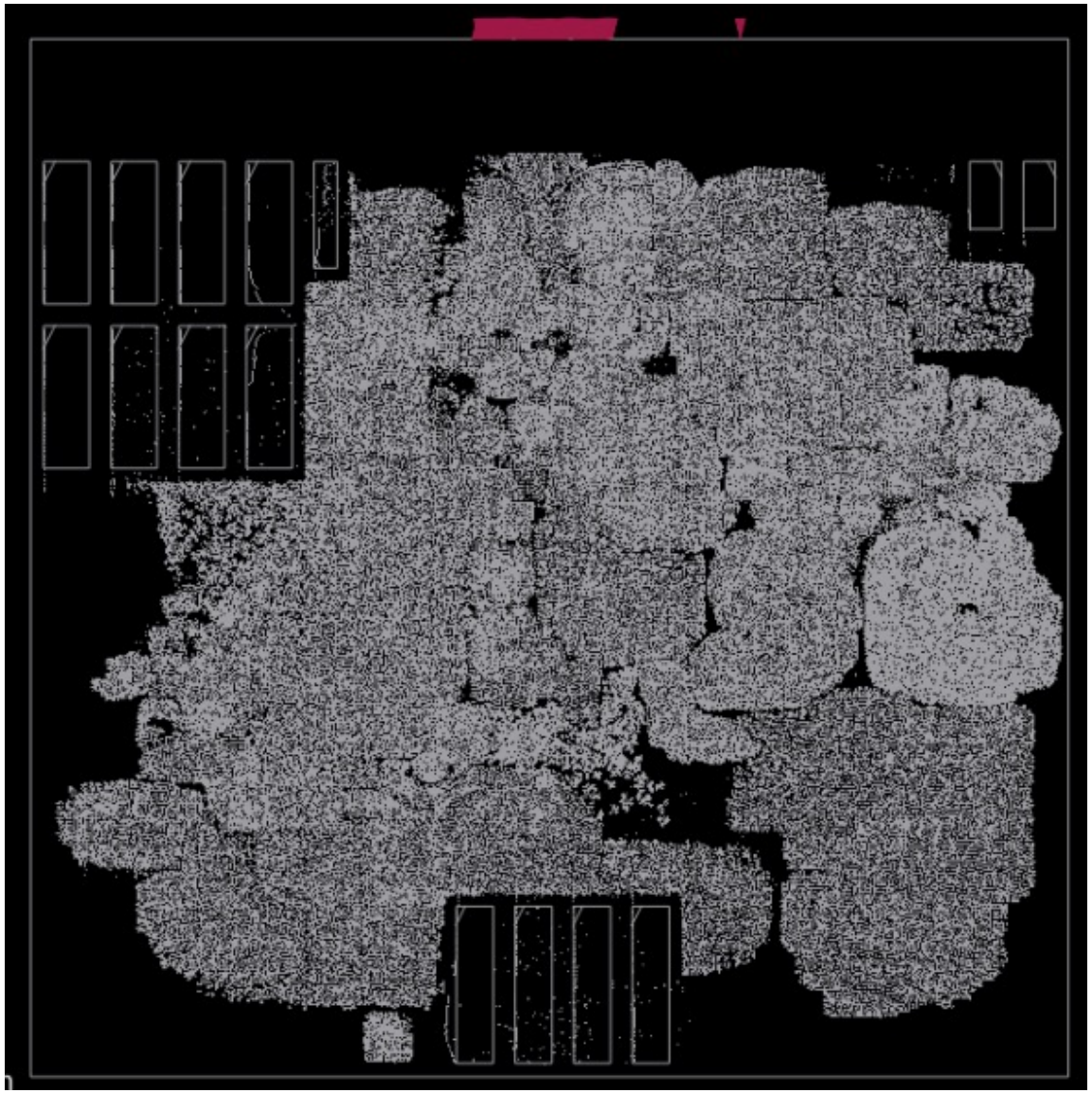
Global Placement



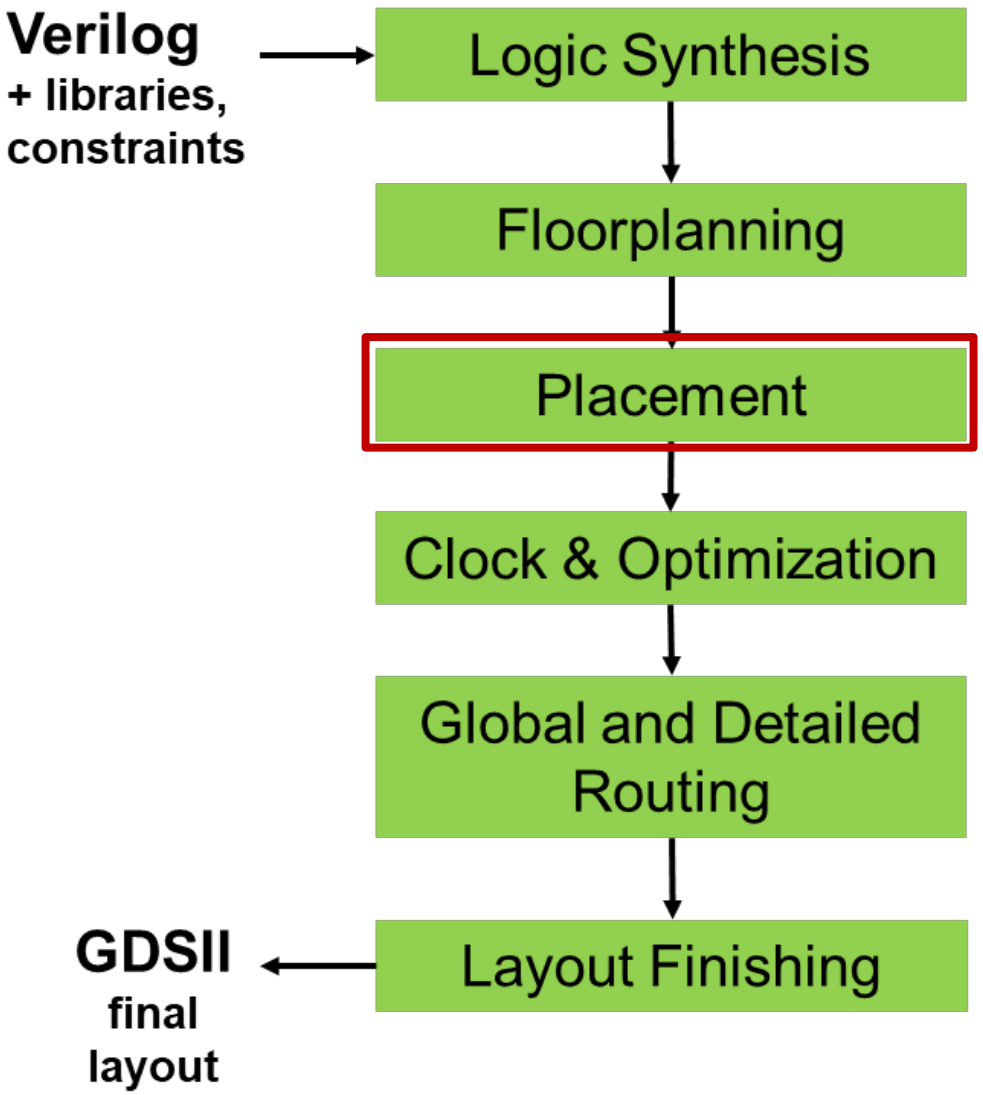


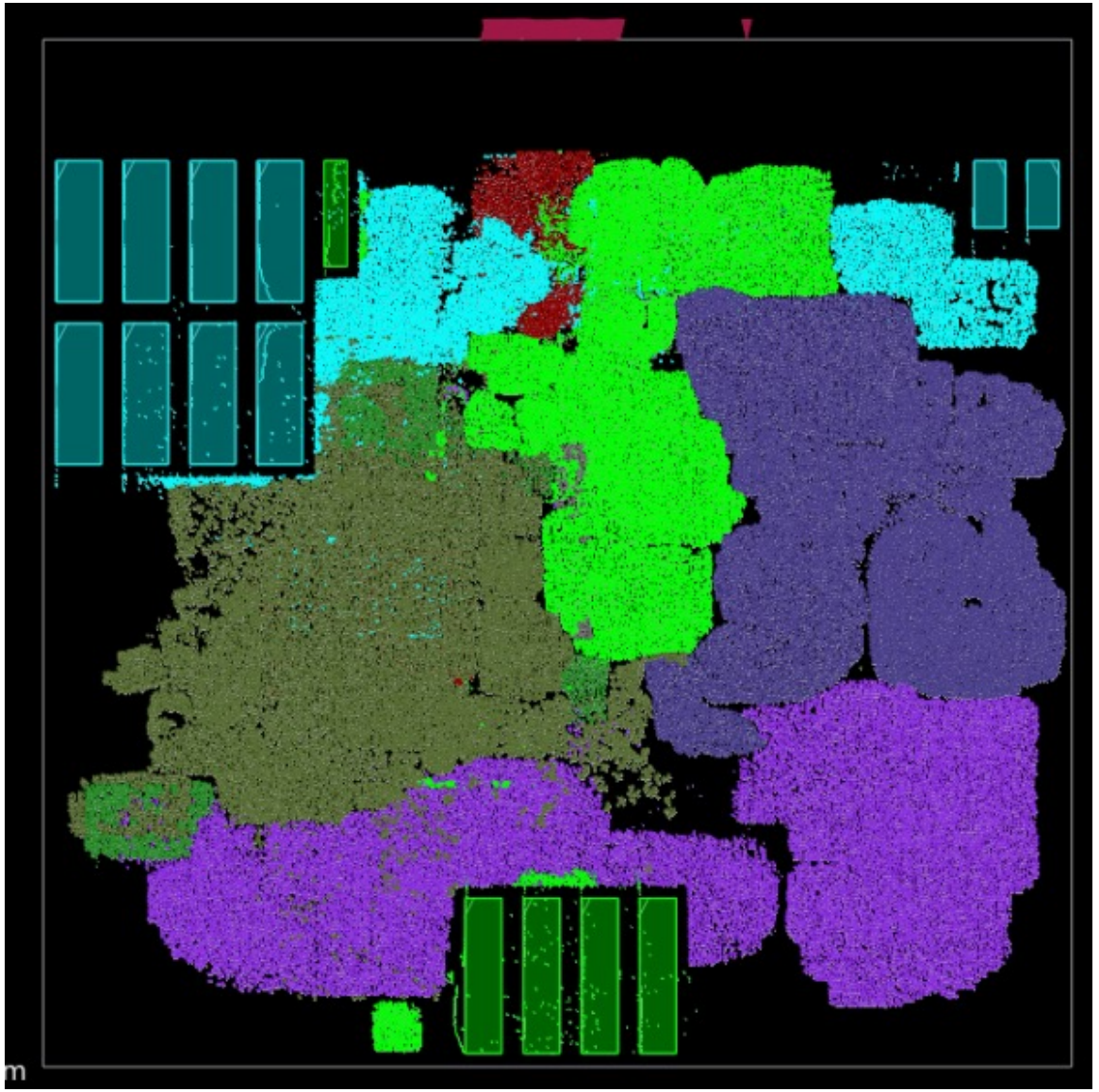
Global Placement



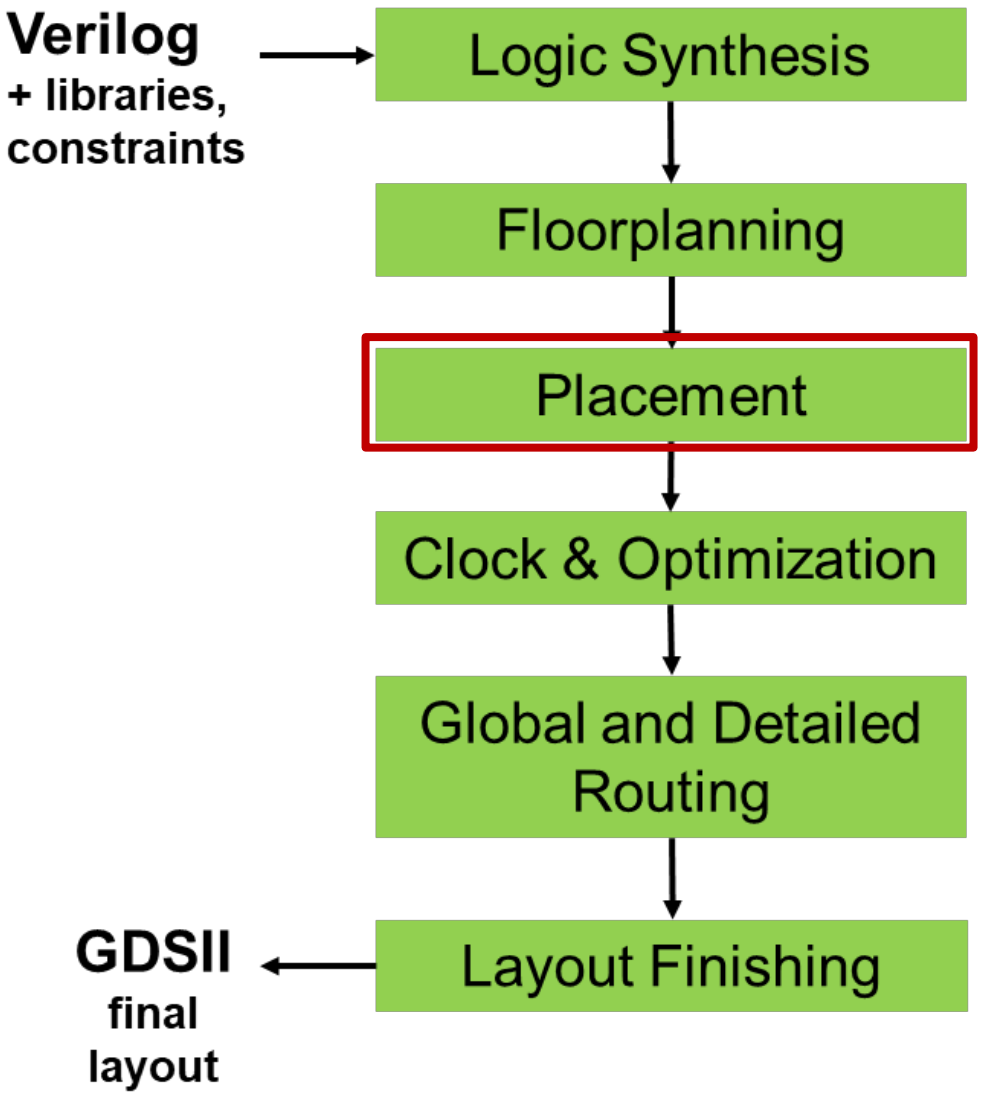


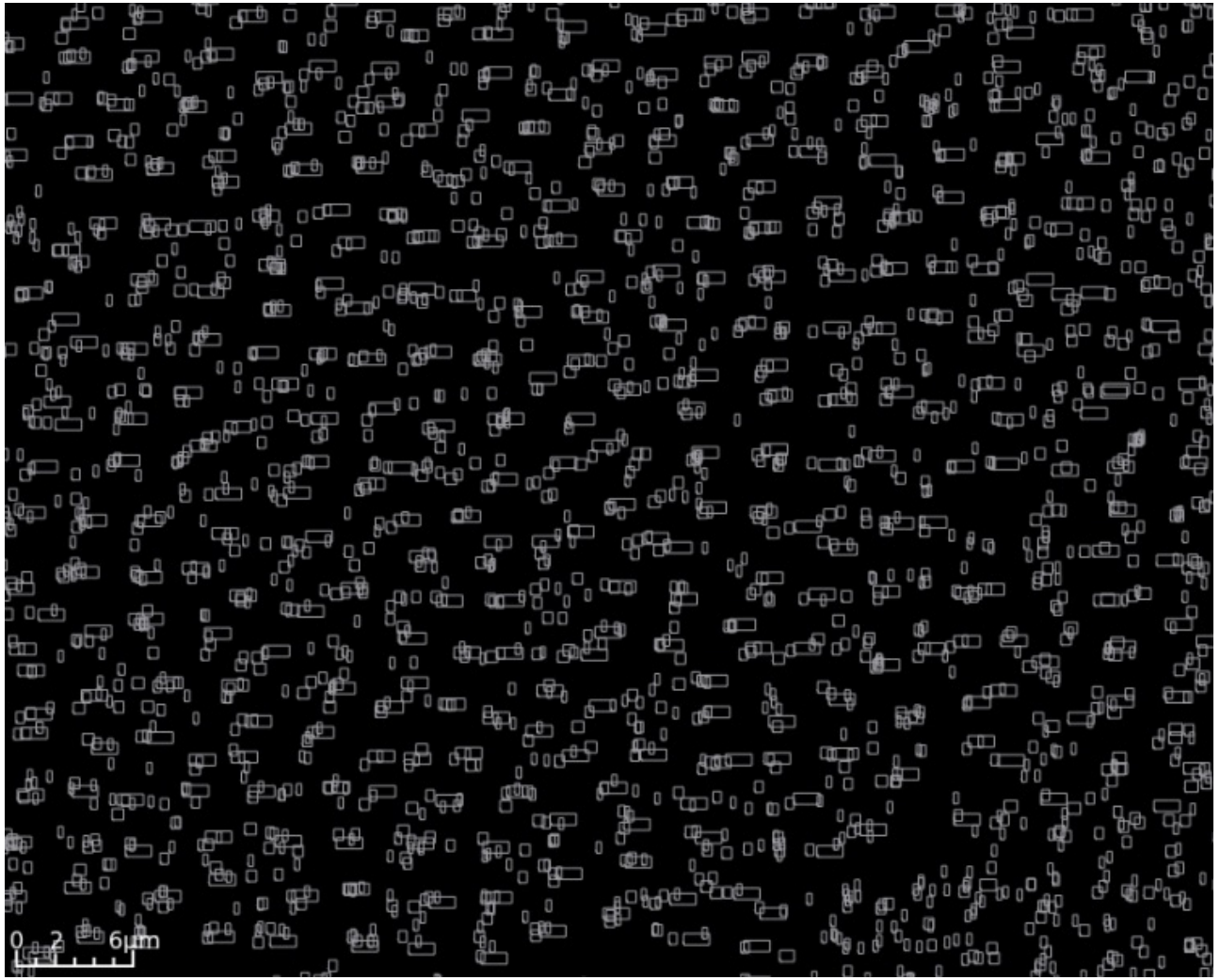
Global Placement w/Partially Expanded Hierarchy



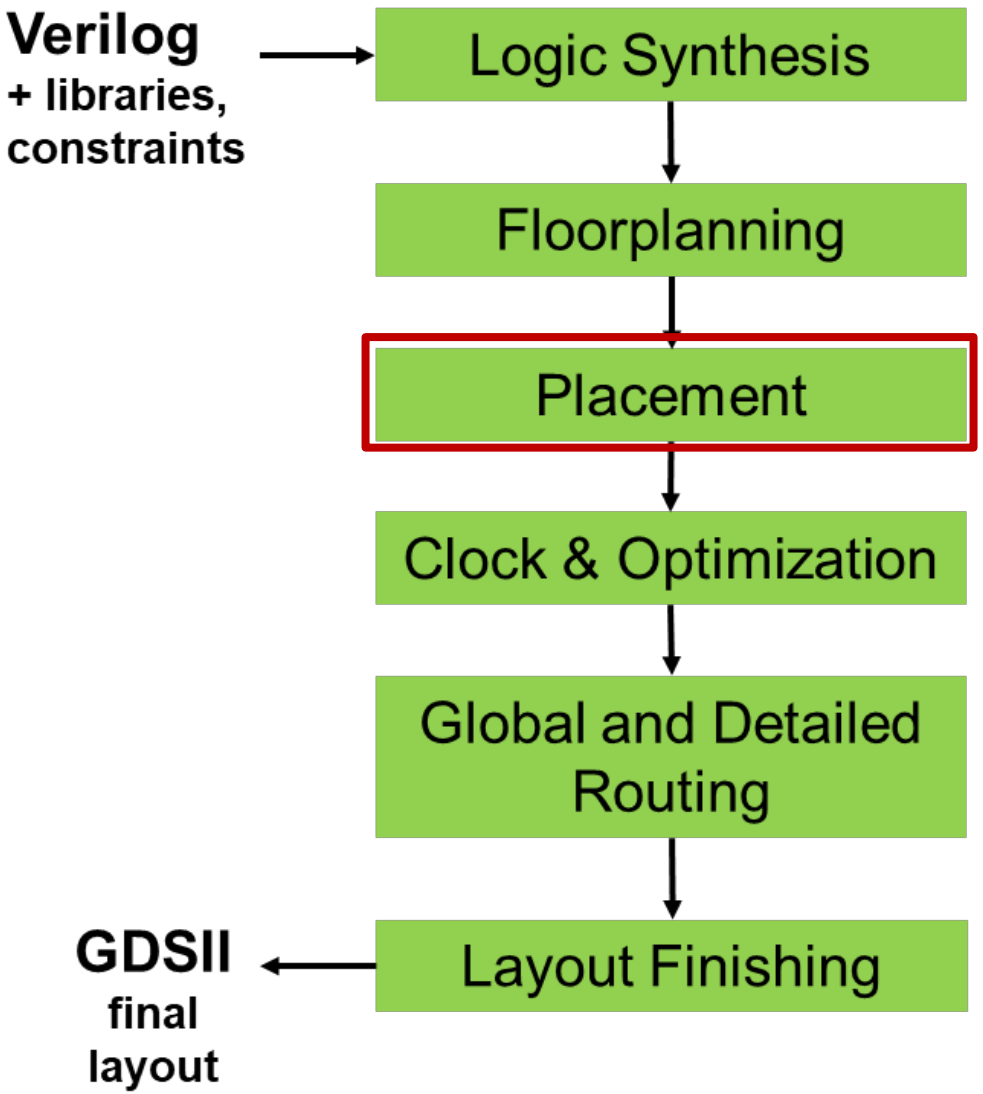


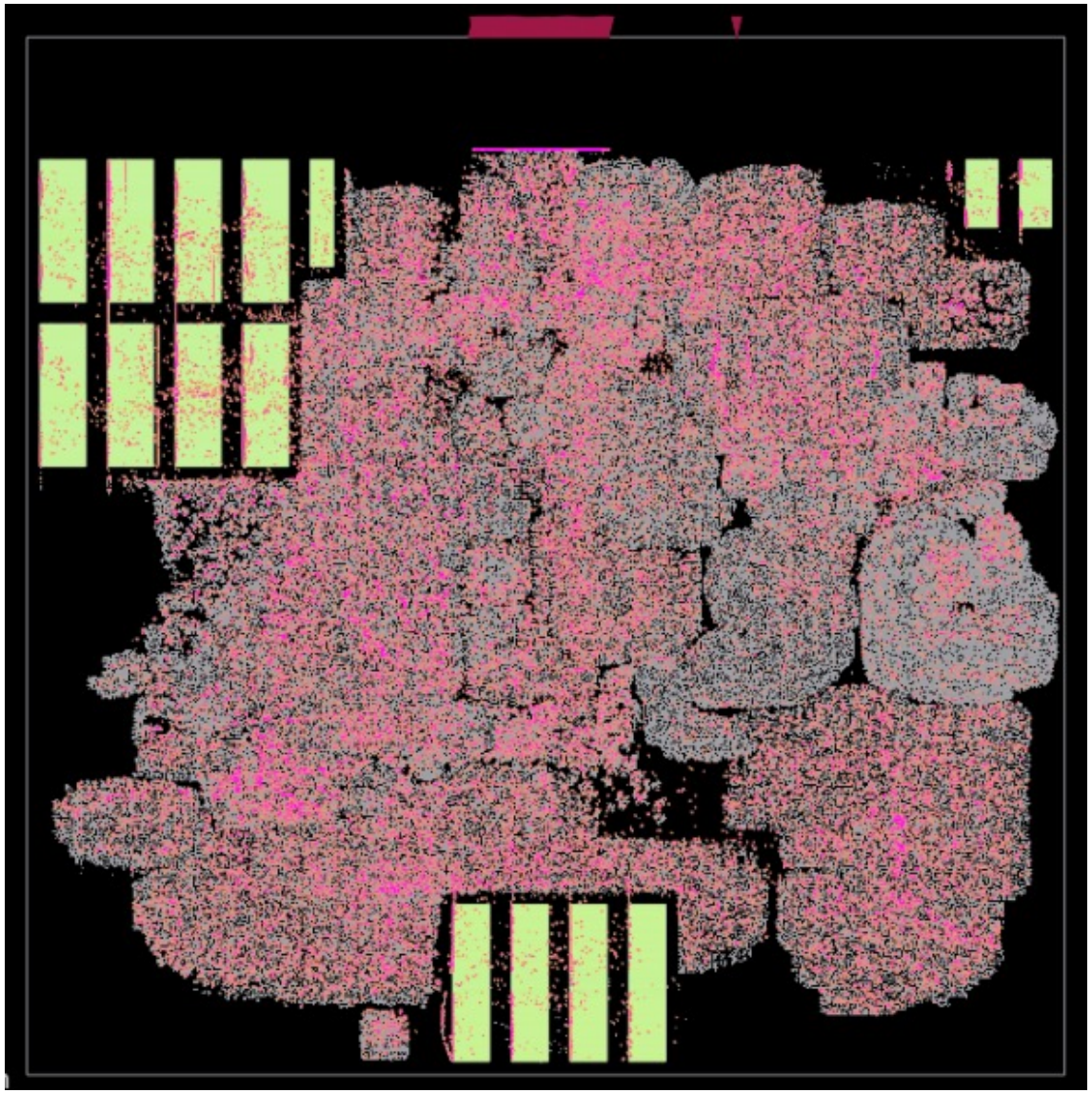
Global Placement Zoom-In



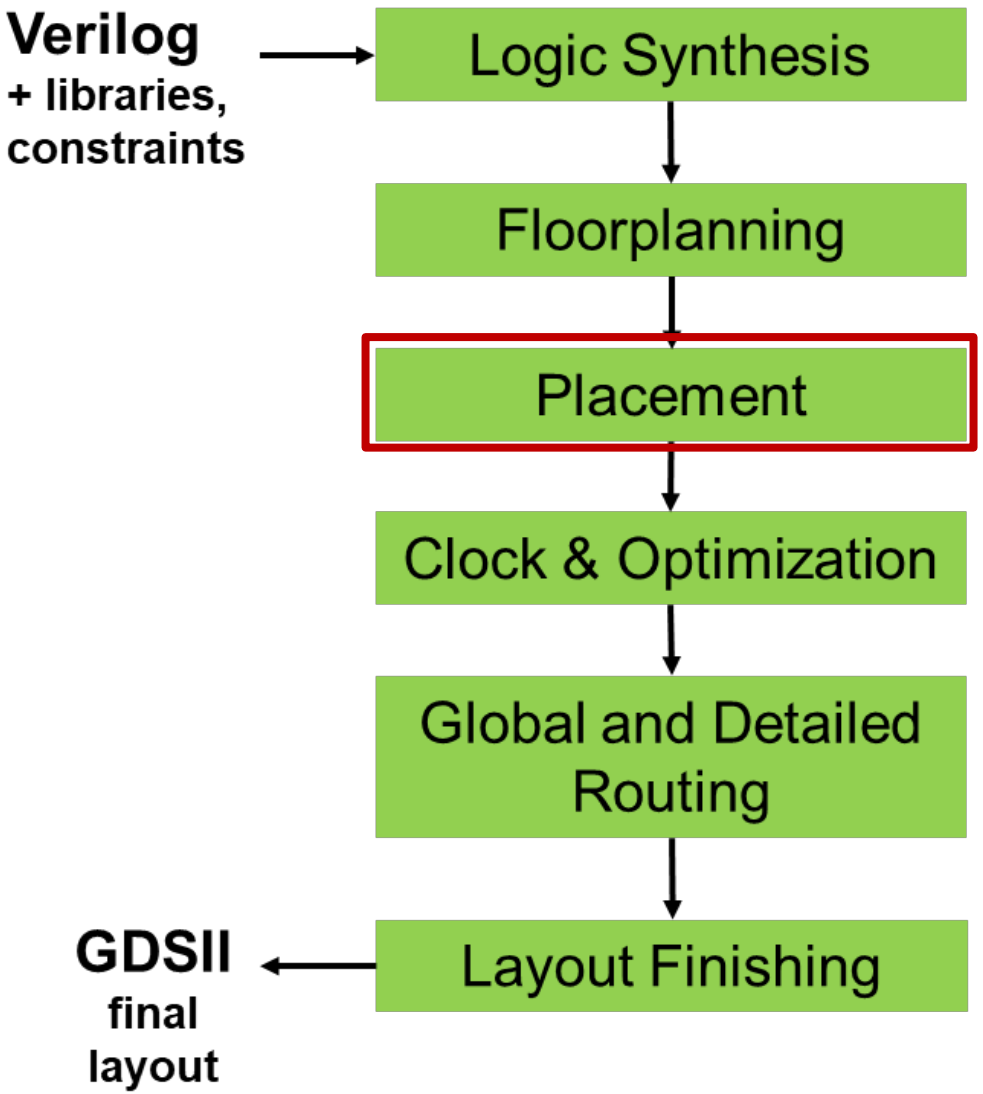


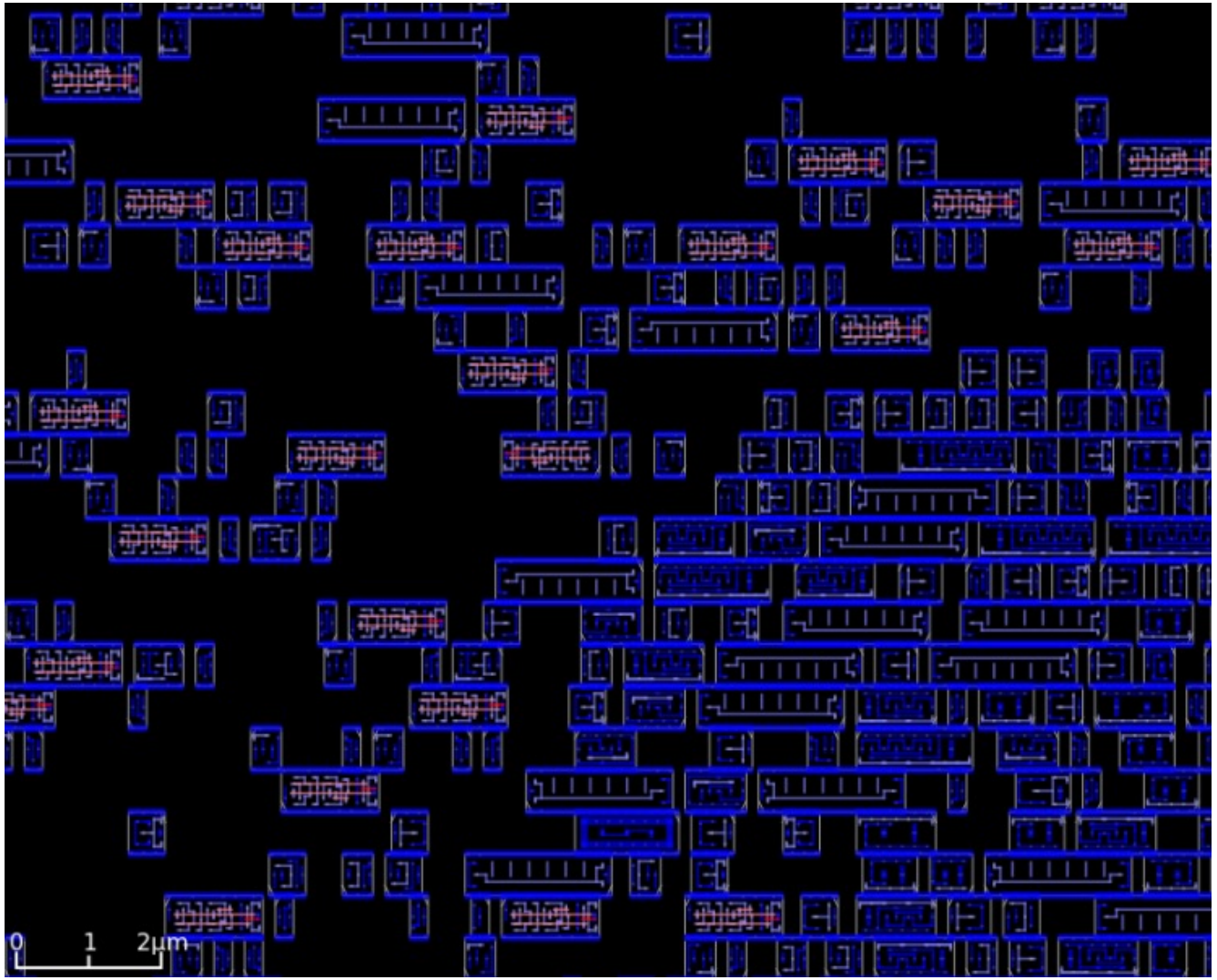
Sizing and Buffering (electrical rules)



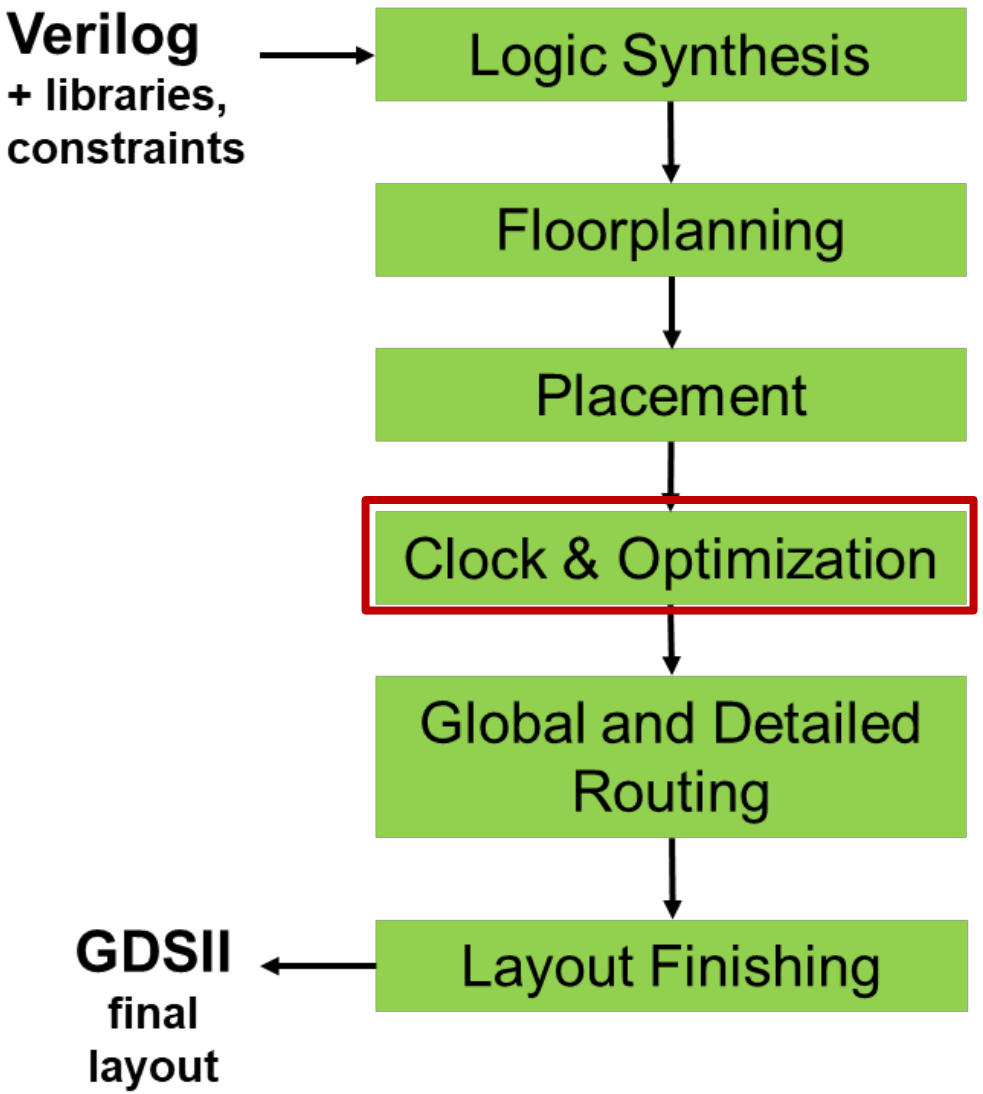


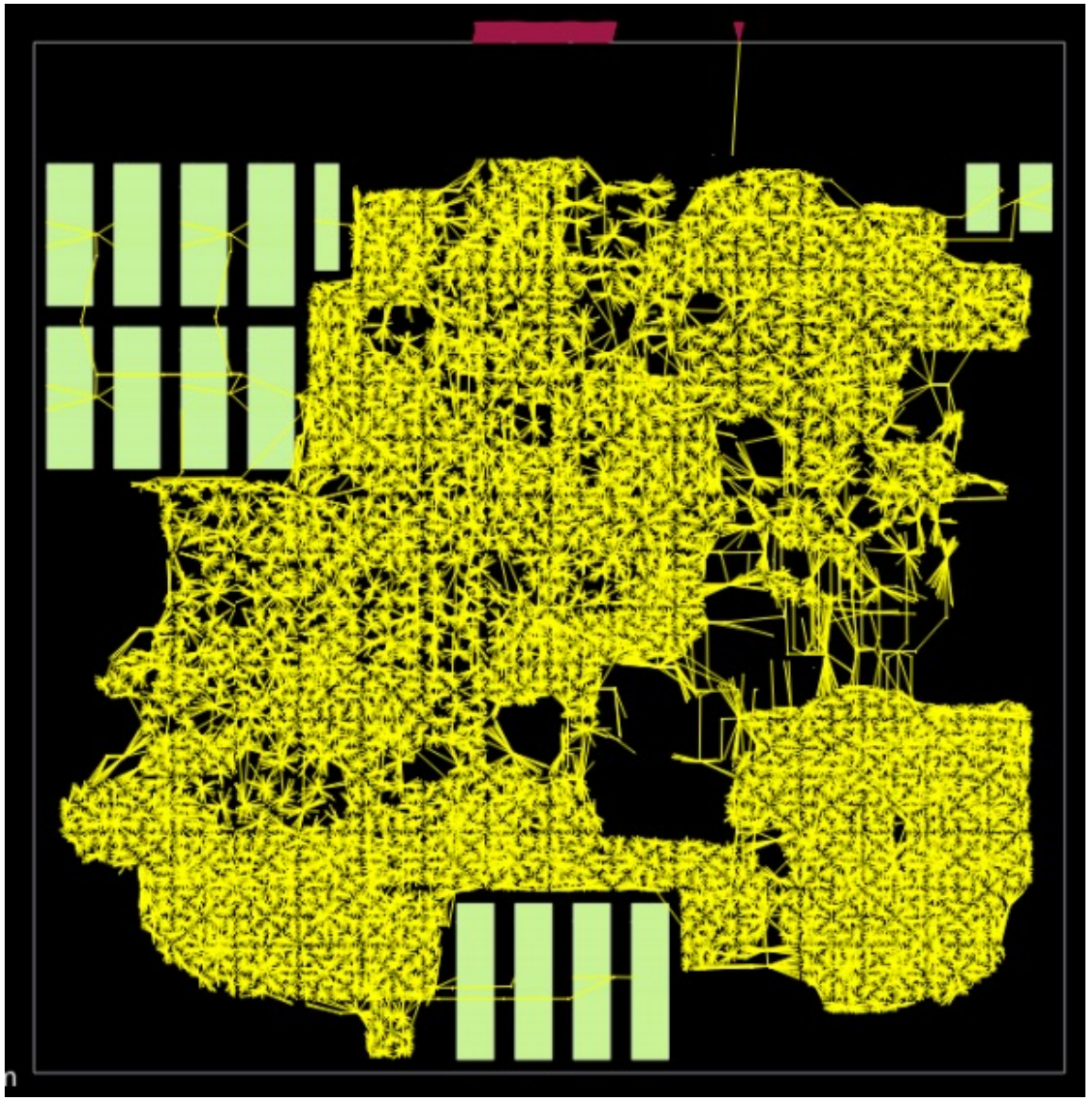
(Legalized) Detailed Placement Zoom-In



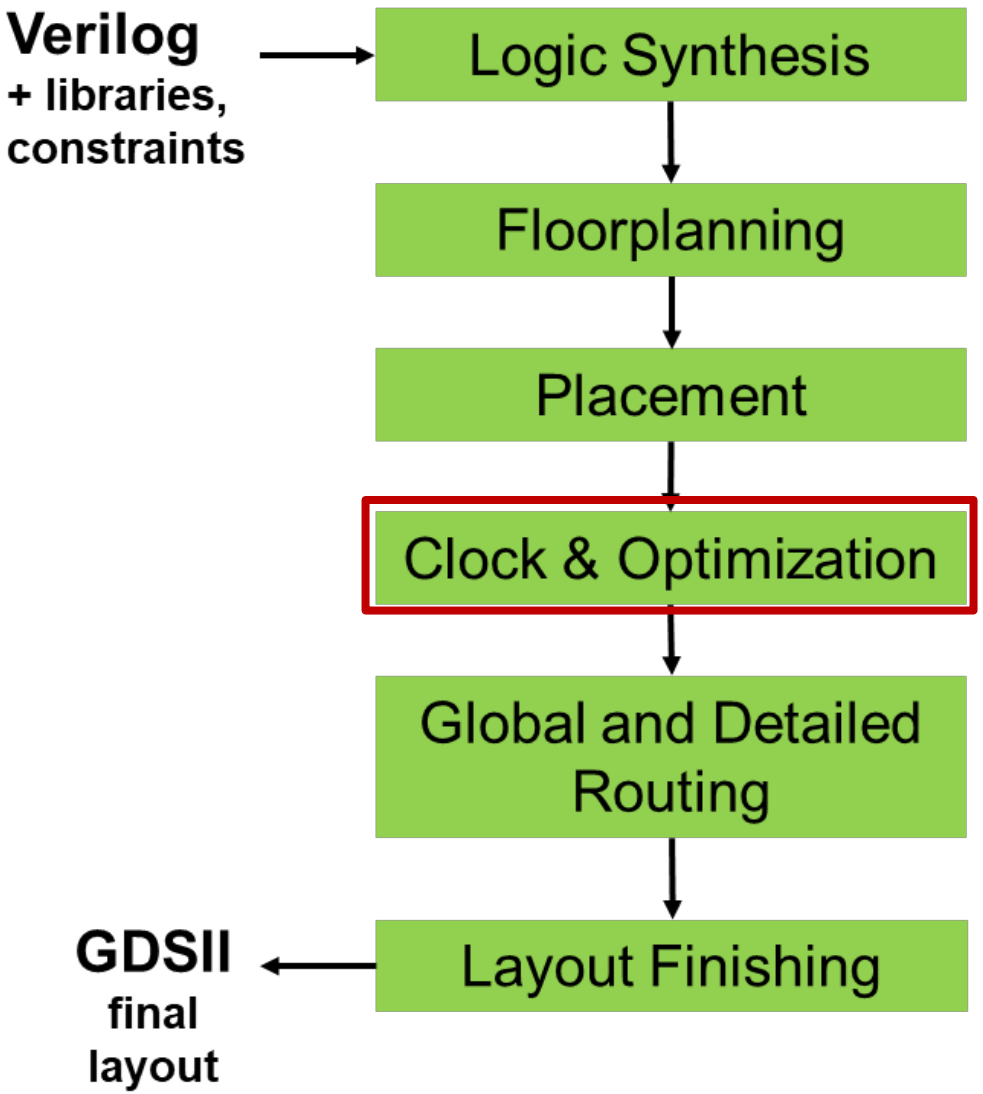


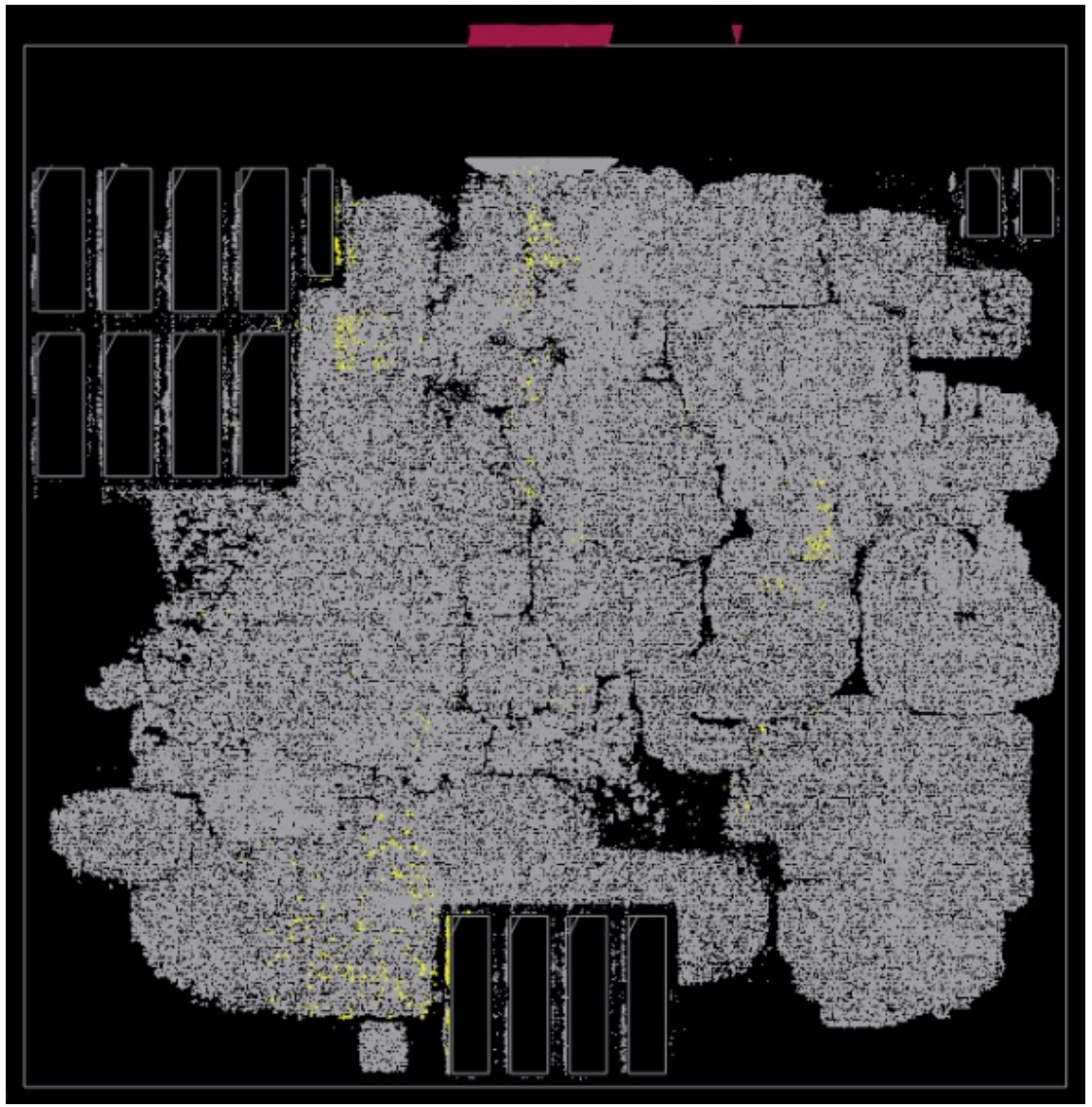
Clock Tree Synthesis





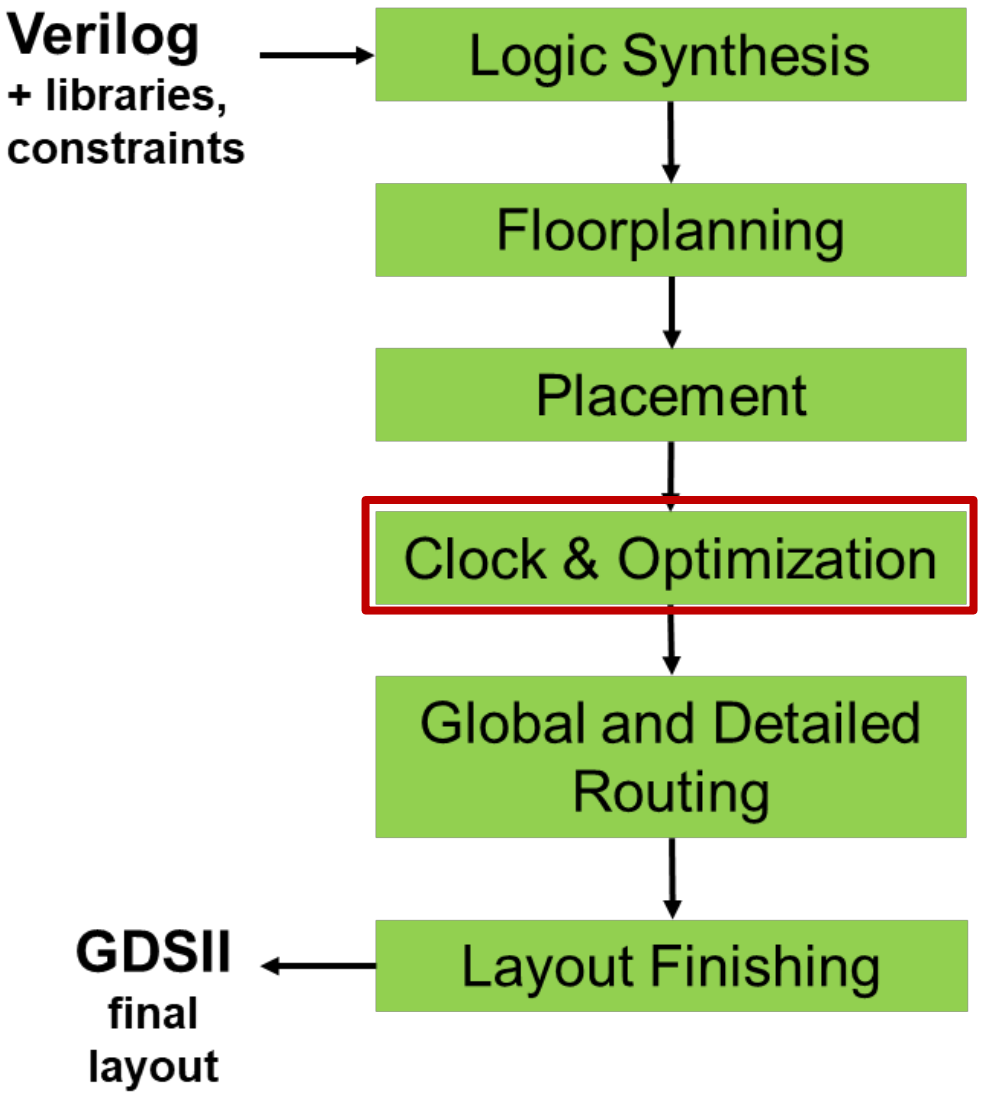
Hold Fix Buffers

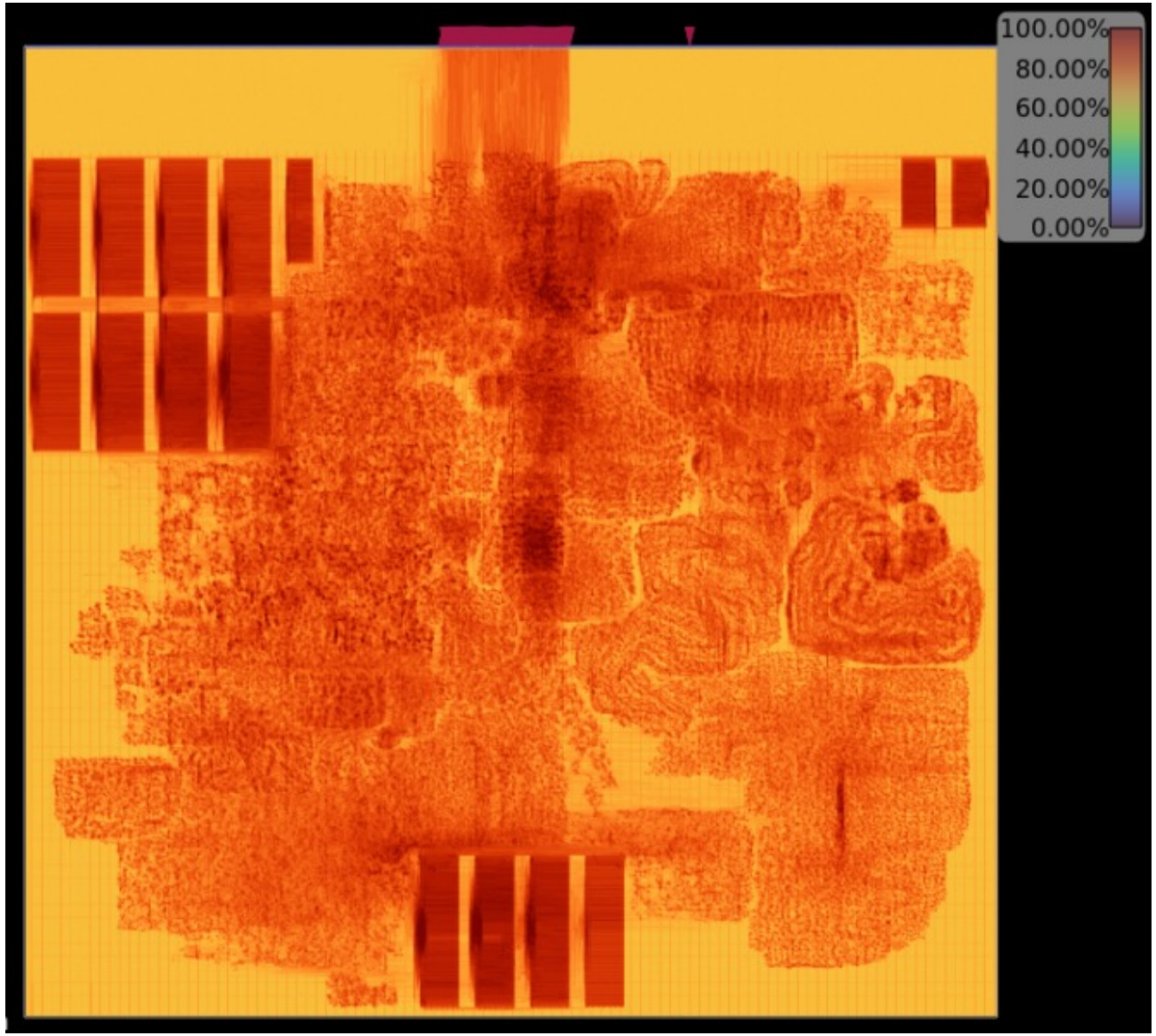




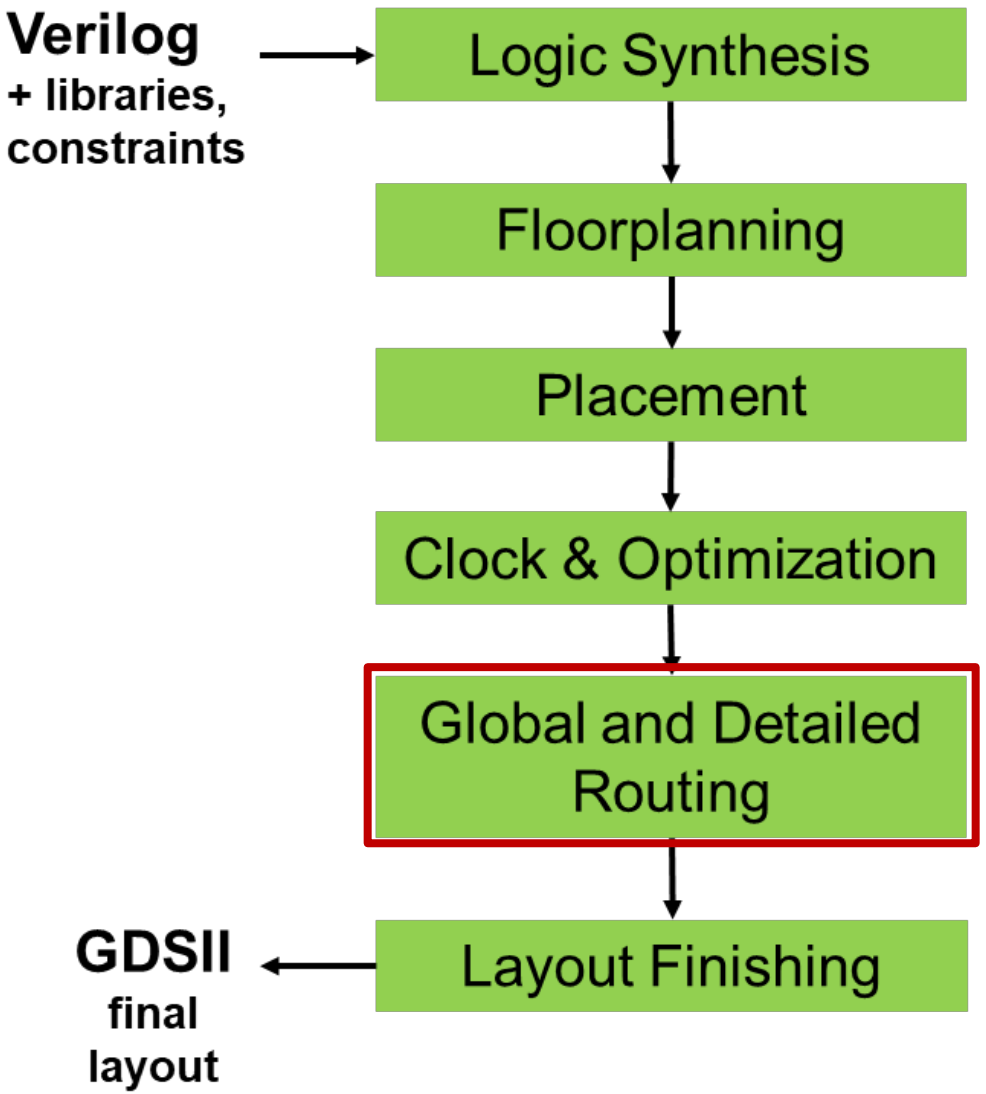
Congestion Map

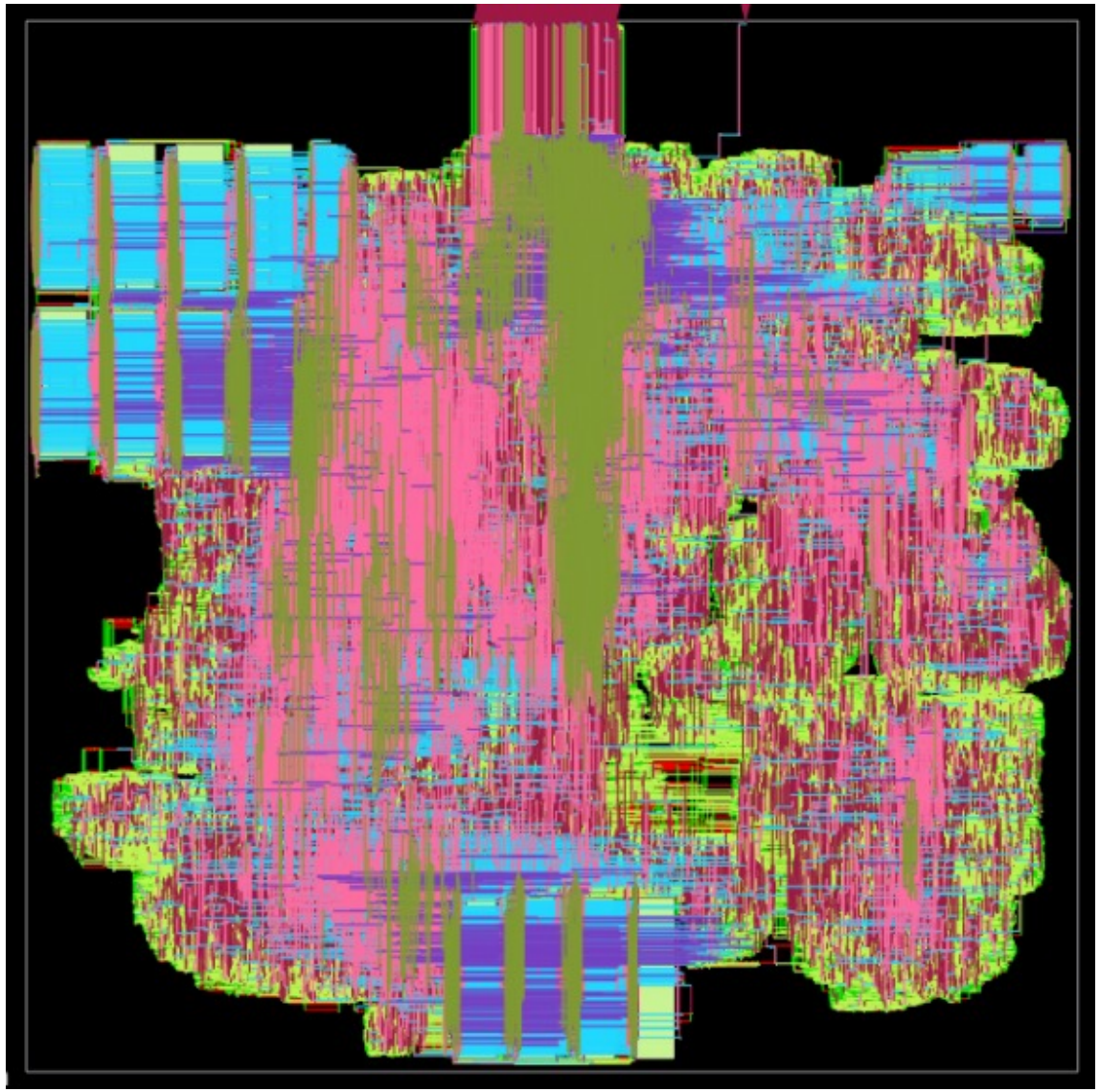
Is this final placement routable?



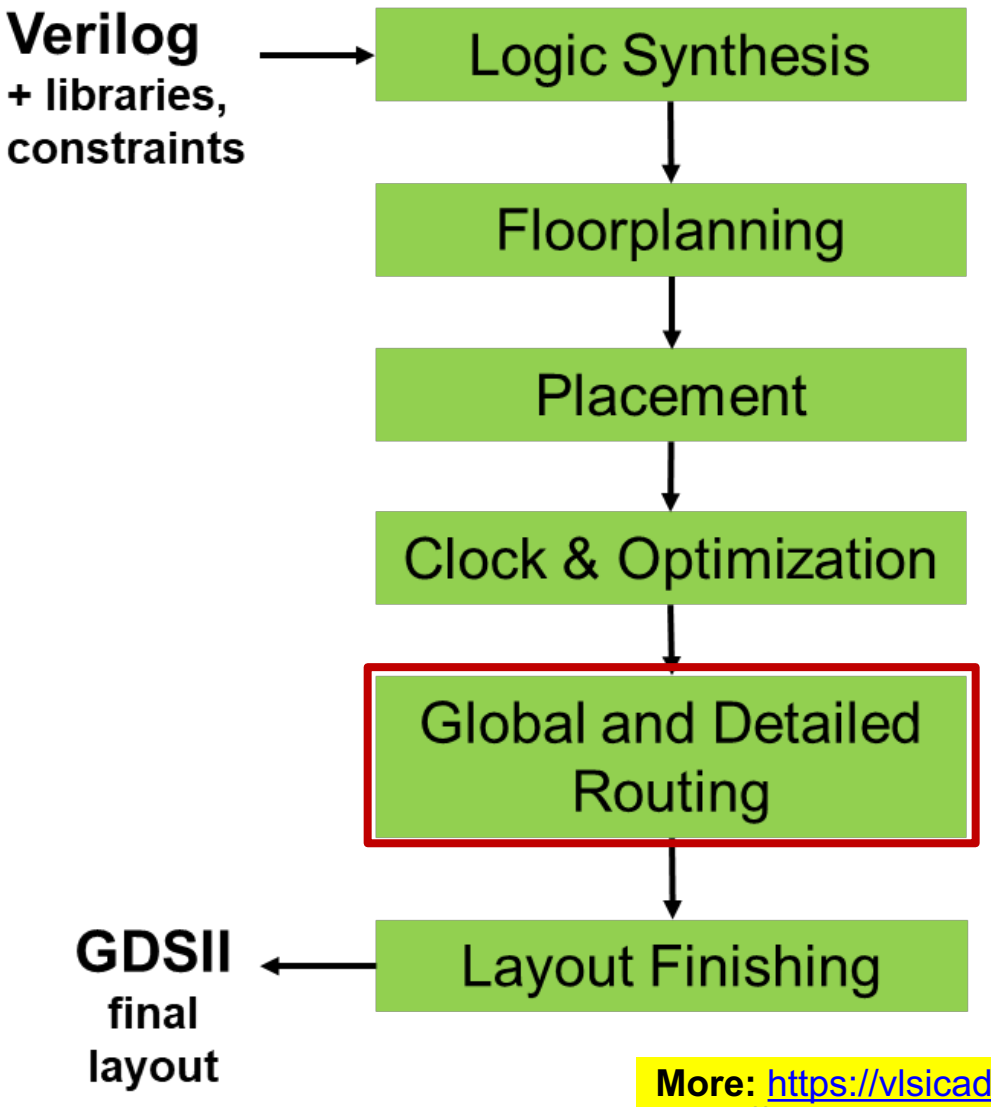


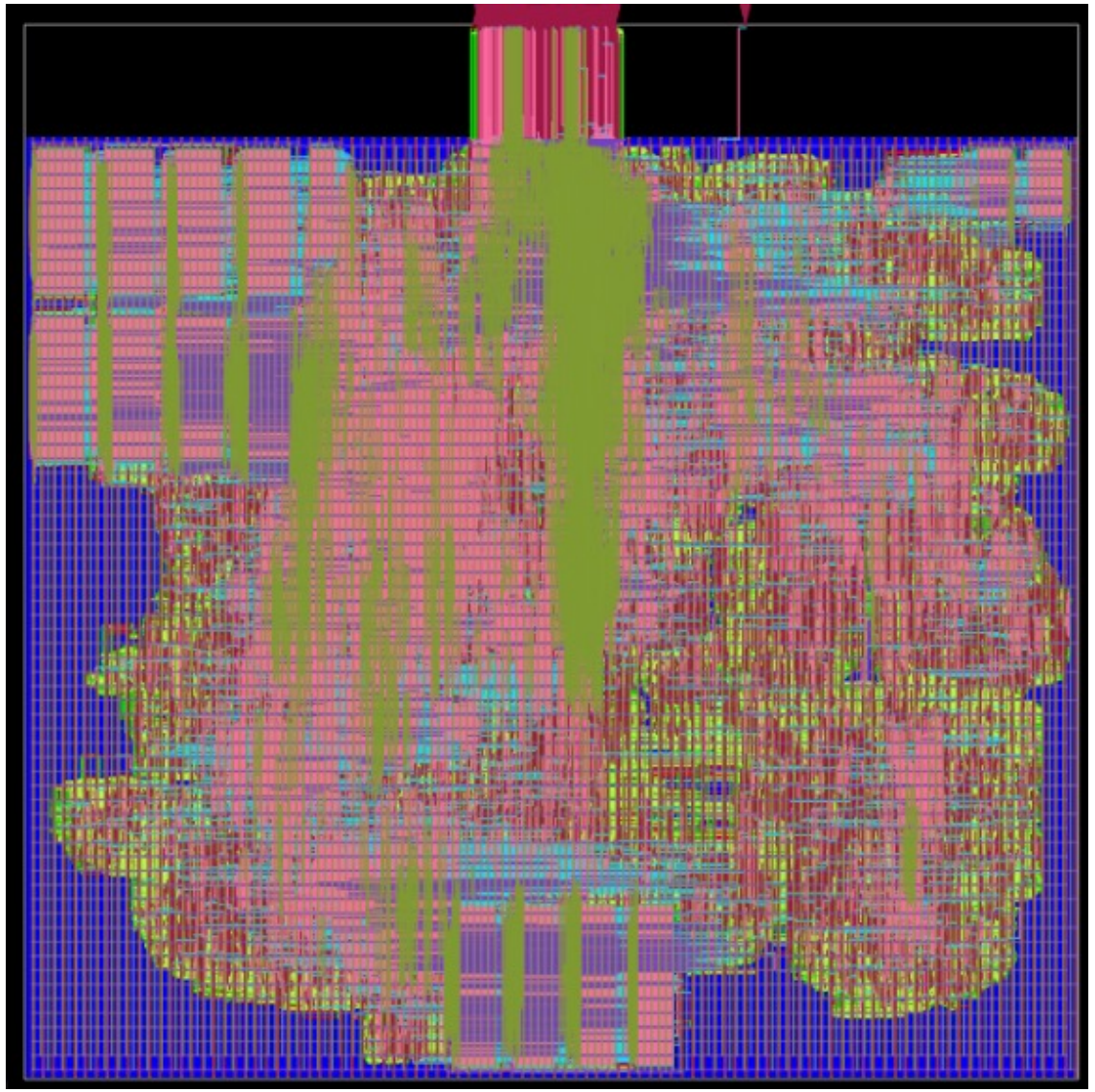
Final (Detailed) Routing





Final (Detailed) Routing, With PDN Shown

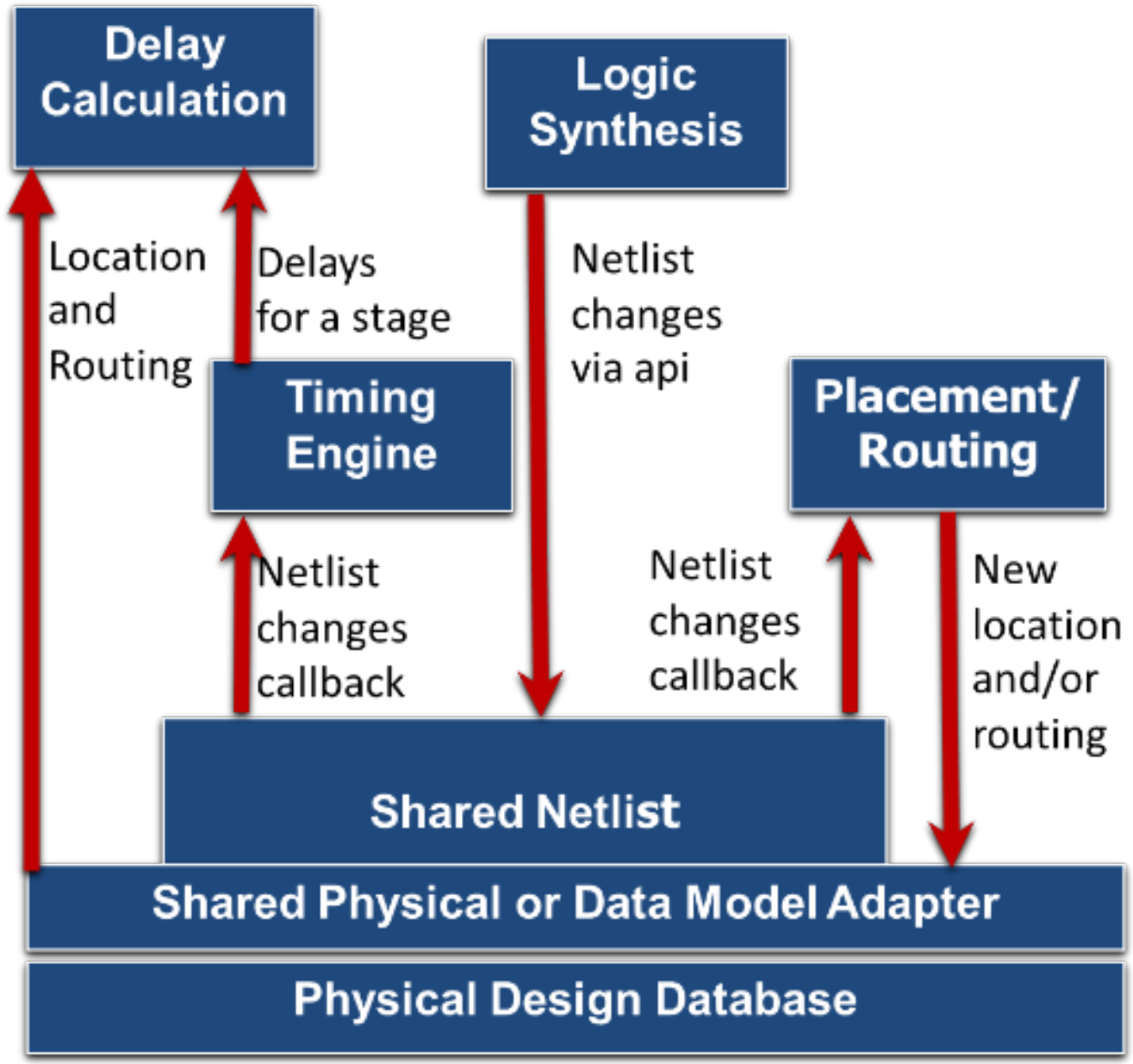




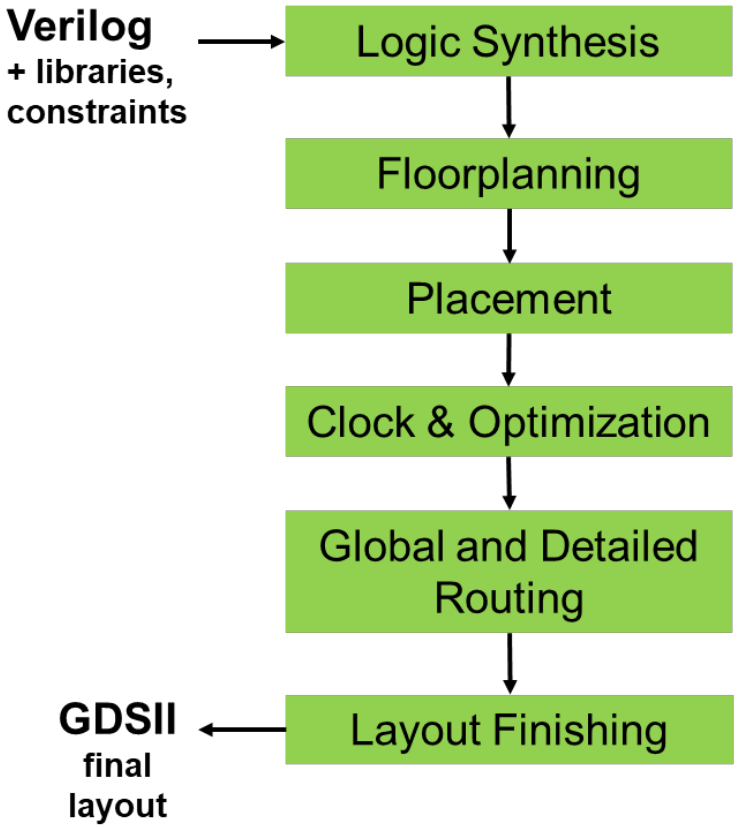
More: https://vlsicad.ucsd.edu
https://theopenroadproject.org

Industrial Strength Incremental Architecture: Built to Last

Further notes: link link link



OpenROAD, 2018-



- **Functionality:** numerous tapeouts at 130nm to 12nm foundry nodes
  - Used for RTL-to-GDS down to 12nm
  - Used for estimation down to 2nm
- **Developer community:** 38000+ commits from 176 contributors
- **User community:** researchers, small R&D teams, startups
- **Education/Workforce development**
  - High school to graduate level, extension
- **+ Robustness:** Multiple public and private/secure Jenkins pipelines
  - Every change runs through 80+ testcases

PDK	Node	# Designs
rapidus	2 nm	9
asap7	7 nm	19
gf12	12 nm	13
nangate45	45 nm	16
sky130hd	130 nm	7
sky130hs	130 nm	6
ihp-sg13g2	130 nm	7
gf180	180 nm	6

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
  - Matrix chat: https://element.fossi-chat.org/
  - OpenTapeout video: https://www.youtube.com/watch?v=wvPZREaP7E0&t=2652s

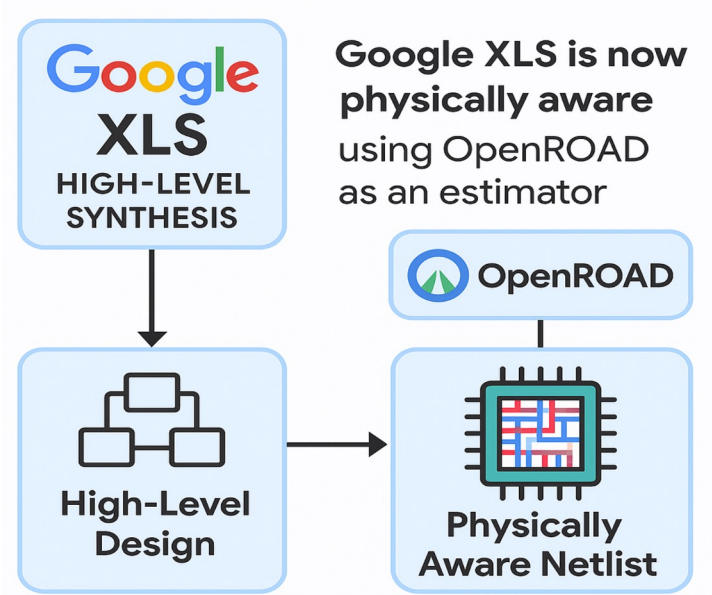
How is OpenROAD used ?

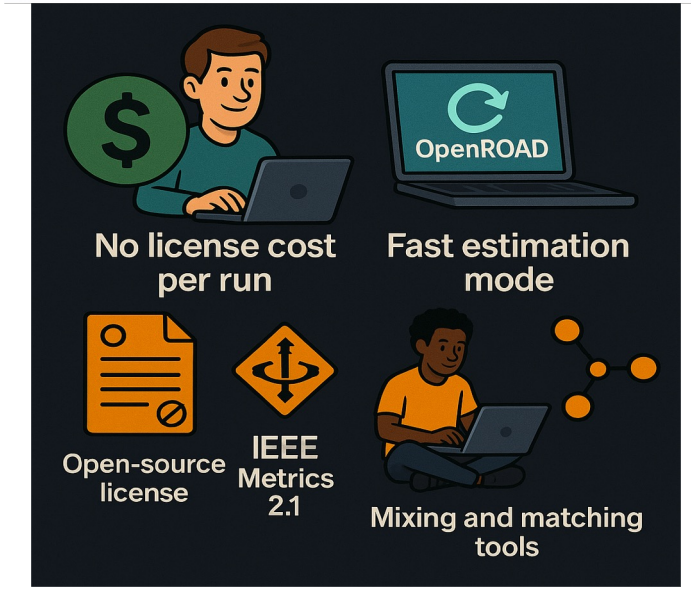
"What is OpenROAD actually good for?"

- **RTL-to-GDS (tapeout) for supported (mature) nodes**
- **Estimation, arch/RTL tuning for nodes down to 2nm**
  - *Anywhere that **estimates** of SP&R can provide useful feedback !*

Example: Usage in Google XLS HLS

- **Overview of important links**
  - High level overview of using OpenROAD for metrics gathering and revisit of HLS decisions
  - **https://google.github.io/xls/**
  - Code to gather metrics
  - **https://github.com/google/xls/blob/main/xls/synthesis/openroad/json_metrics_server_main.cc**
  - TCL code to pull timing metrics from OpenROAD's STA
  - **https://github.com/google/xls/blob/main/xls/synthesis/openroad/sta_by_stage.tcl**





How is OpenROAD used ?

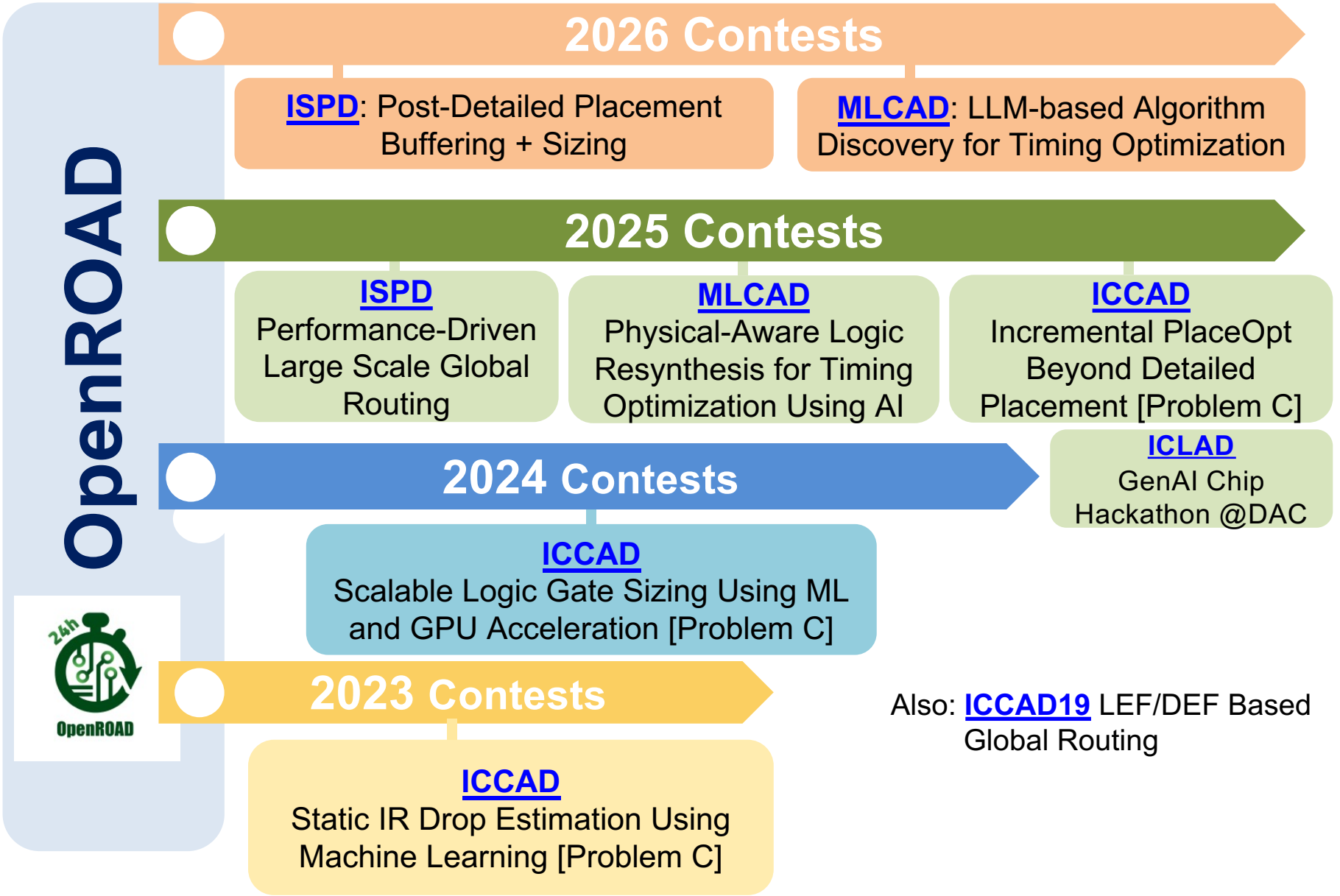
"What is OpenROAD actually good for?"

- **RTL-to-GDS (tapeout) for supported (mature) nodes**
- **Estimation, arch/RTL tuning for nodes down to 2nm**
  - *Anywhere that **estimates** of SP&R can provide useful feedback !*

- Massively parallel DSE and sampling *hyperscalers, chip startups*
- **(!) Smart hybrid open-commercial flows** *Zero ASIC, Rise DA*
- Analog/mixed-signal (big-A, little-D) *Ozark IC*
- Trusted microelectronics *auditable code, custom features*

- **Education and workforce development**
- **AI / ML** including agentic EDA R&D *EDA data that is usable for ML!*

Academic Contests Built on OpenROAD

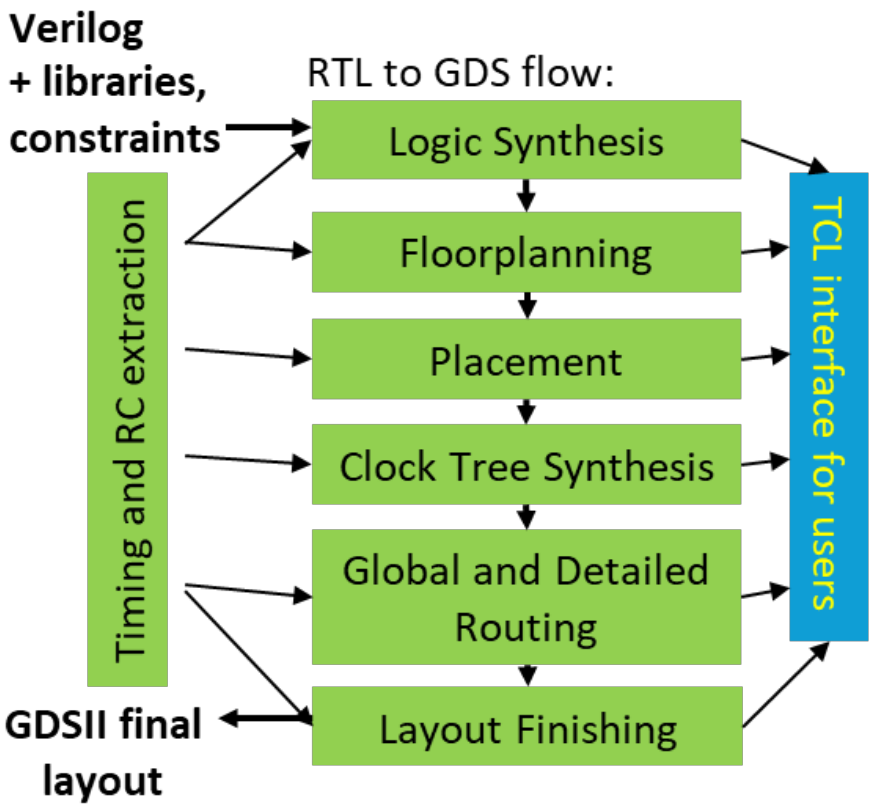


Also: ICCAD19 LEF/DEF Based Global Routing

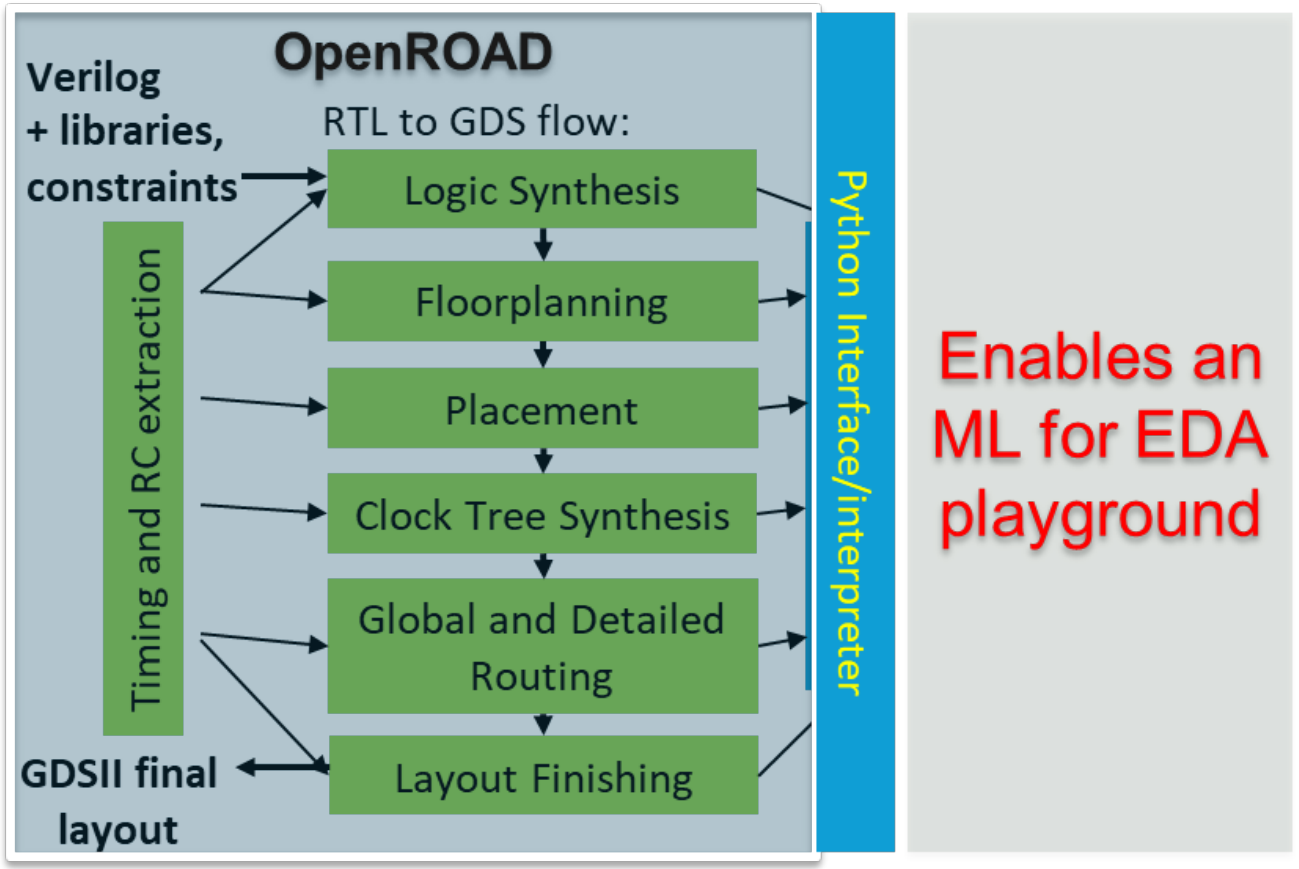
Snapshots / Vignettes: ML and GenAI

AI/ML for EDA, IC Design

What do designers/users work with?

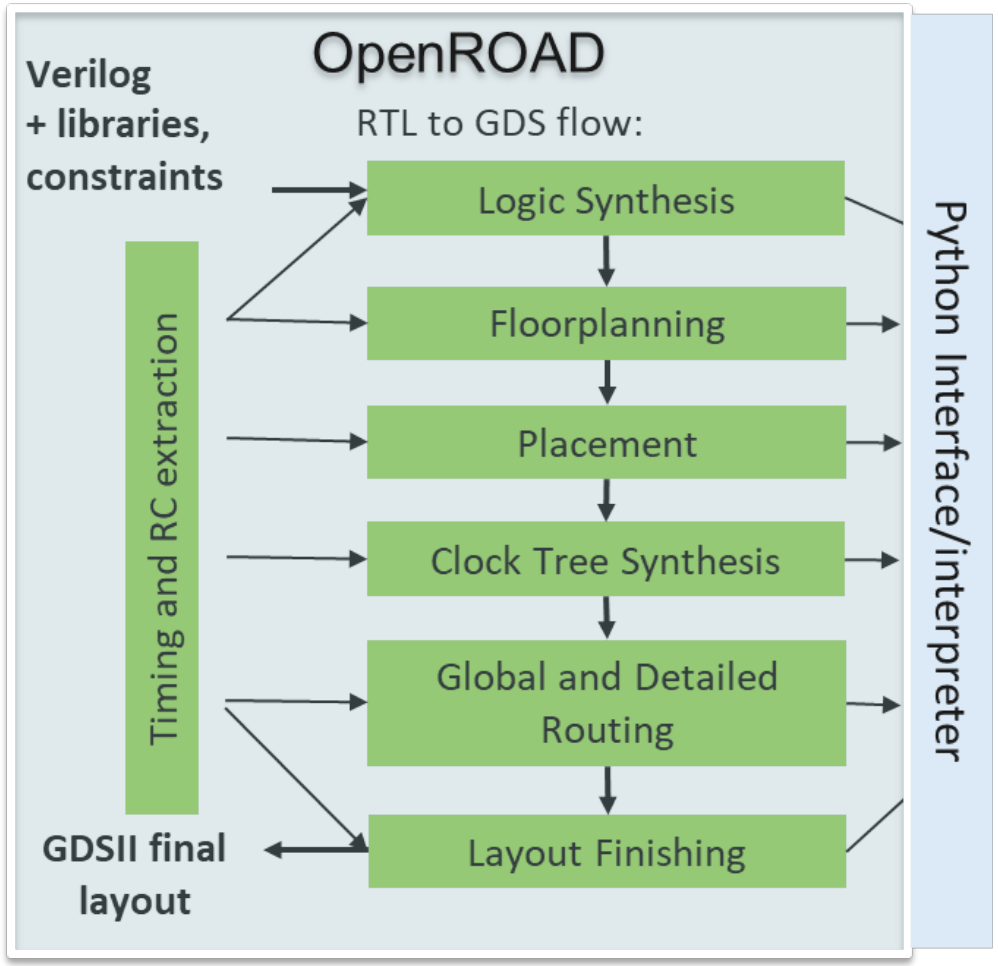


What can they <u>also</u> work with?



Thanks to: Prof. Vidya Chhabria, ASU

OpenROAD as an ML for EDA "Playground"



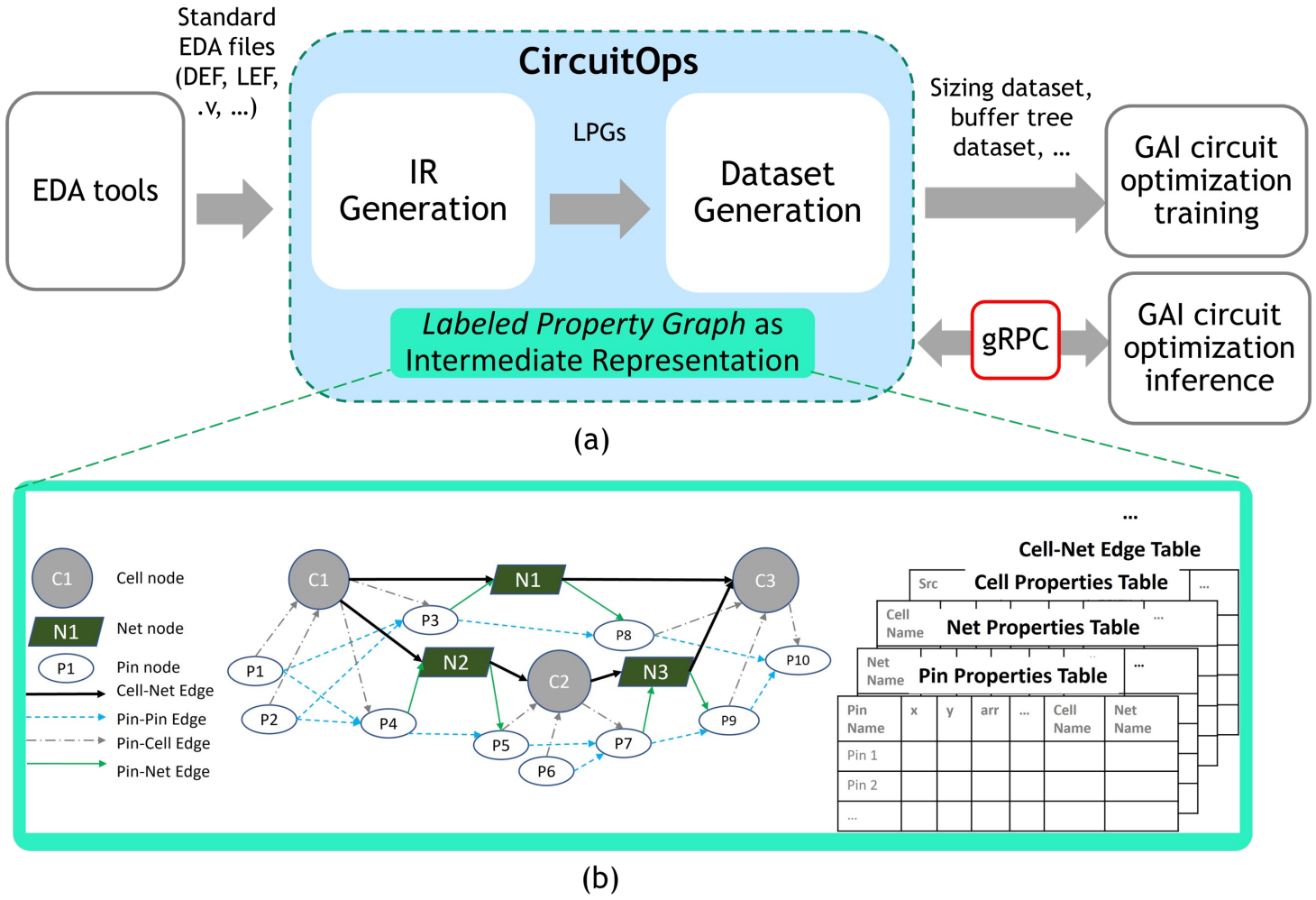
Enables an ML for EDA playground

- **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

Thanks to: Prof. Vidya Chhabria, ASU

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.

Thanks to: Prof. Vidya Chhabria, ASU

ML-centric APIs

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

   where ***all_slacks*** is a numpy array

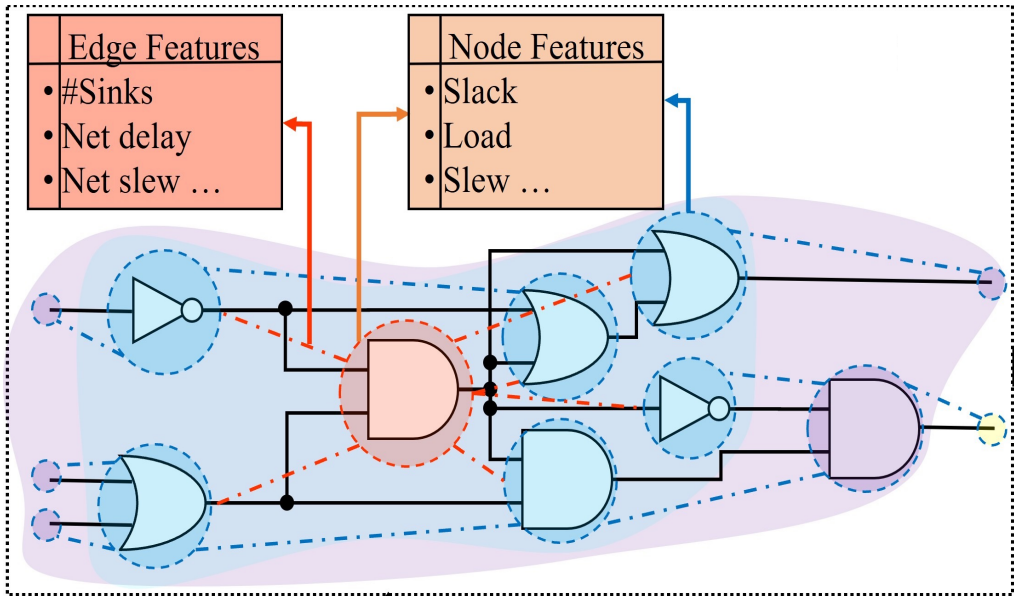
2) ***graph_design = ord.get_netlist(list_insts, properties)***

   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 graph-based data extraction: node, edge features

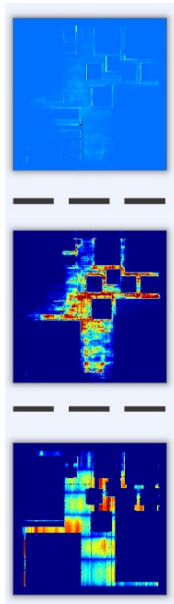


Examples of image-based ML data extraction

Congestion

DRC violations

IR drop



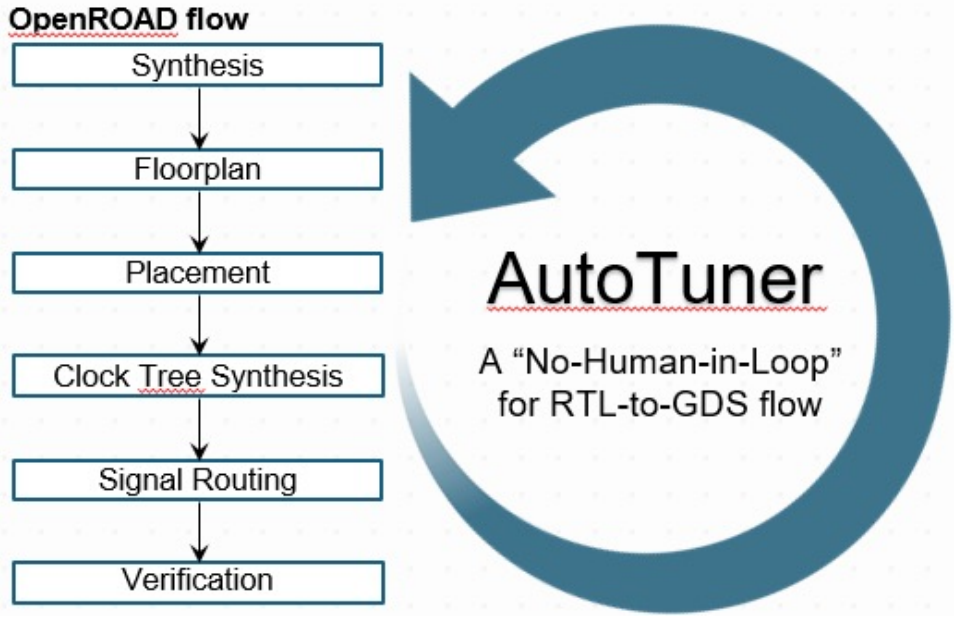
Thanks to: Prof. Vidya Chhabria, ASU

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

- Automatic, iterative tuning to improve user-defined score within a given hyperparameter range space

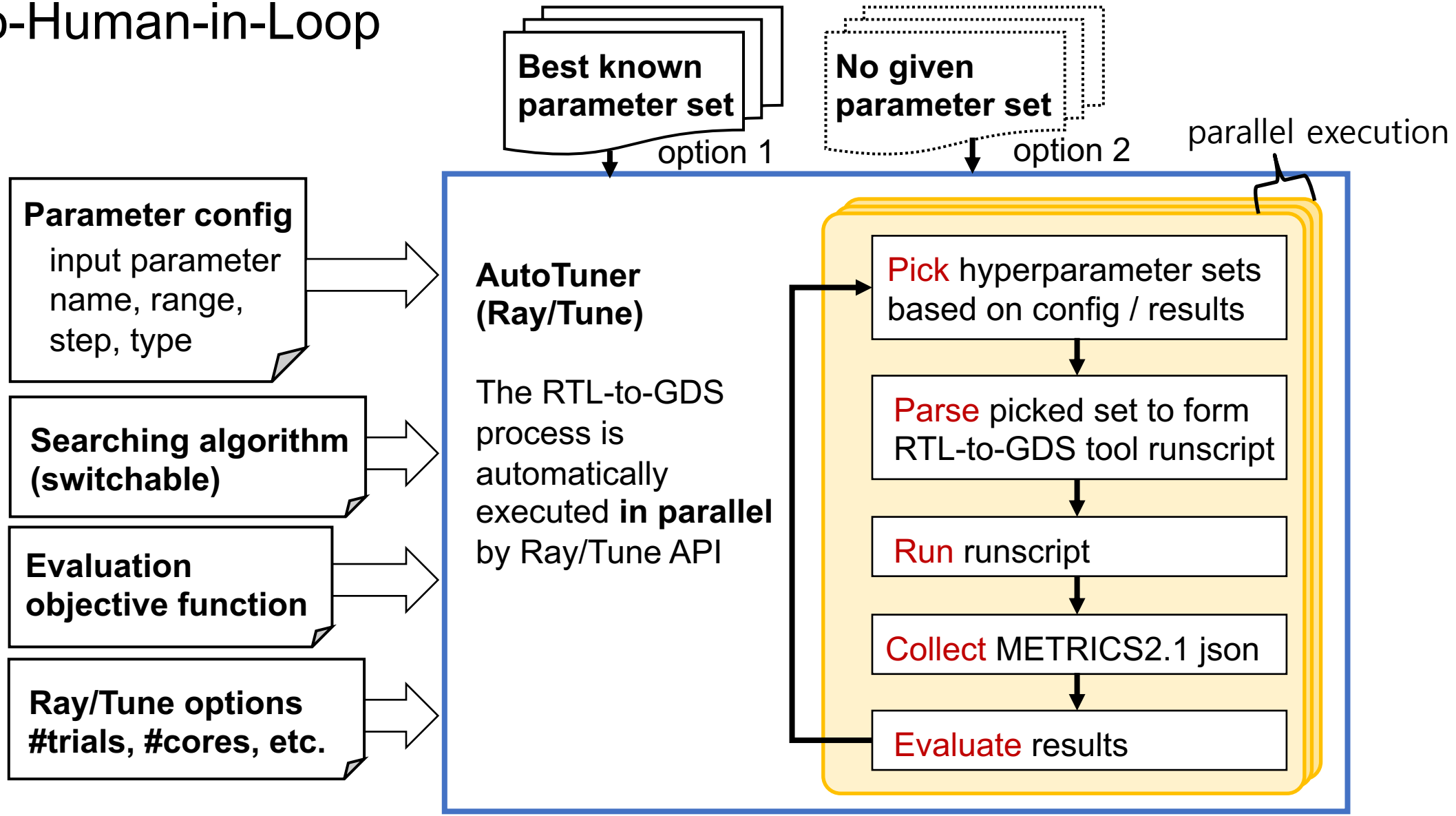
- Interface: Python packages Ray/Tune
- Algorithms: HyperOpt, PBT, Optuna, Nevergrad, Ax, random search

- 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>

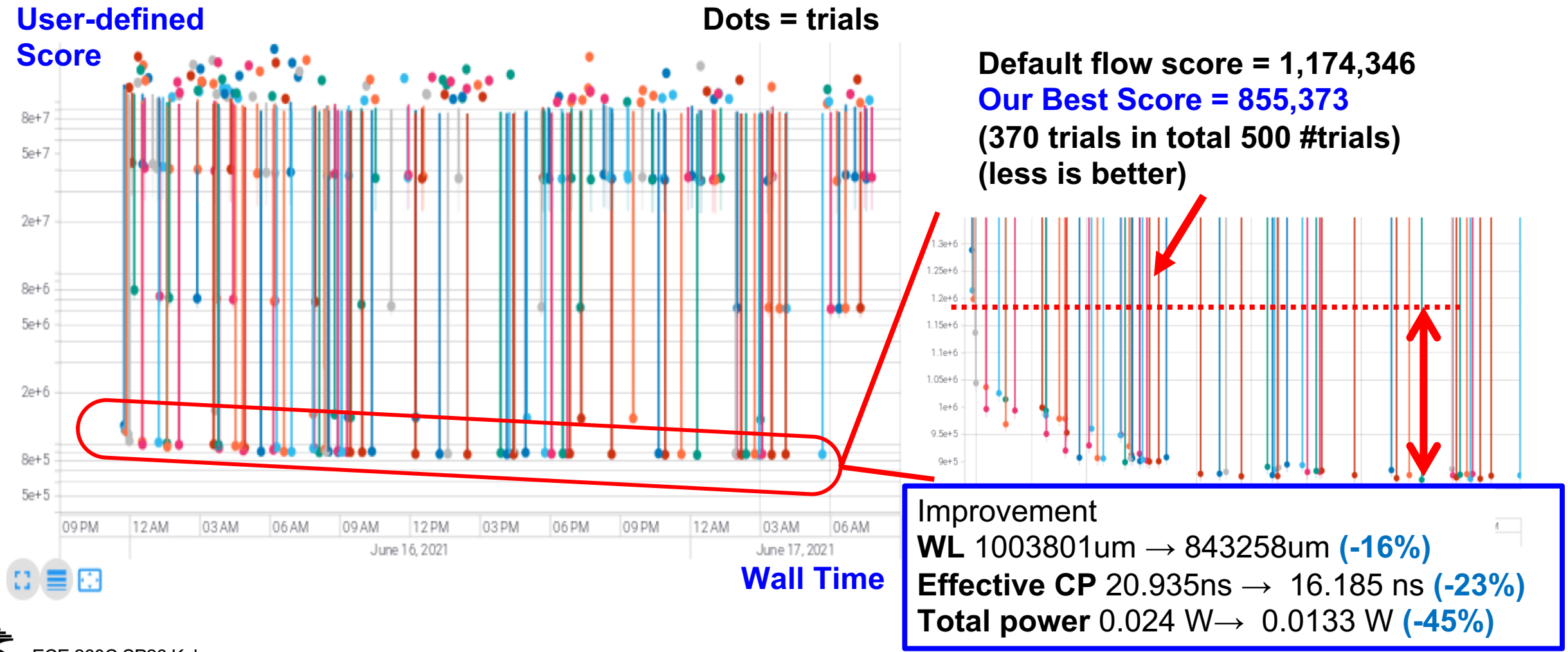
Assuming full factorial combinations,
1,058,298,150
= # possible combinations!

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

- 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 walltime



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

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
timing__setup__wns	Setup worst negative slack in the design
timing__setup__wns__clock:clk_a	Setup worst negative slack for clock “clk_a” in the design
timing__setup__wns__analysis_view:slow	Setup worst negative slack for analysis view “slow”
power_total	Total power consumption
power__leakage	Total leakage power
power__leakage__clock	Total leakage power in the clock network

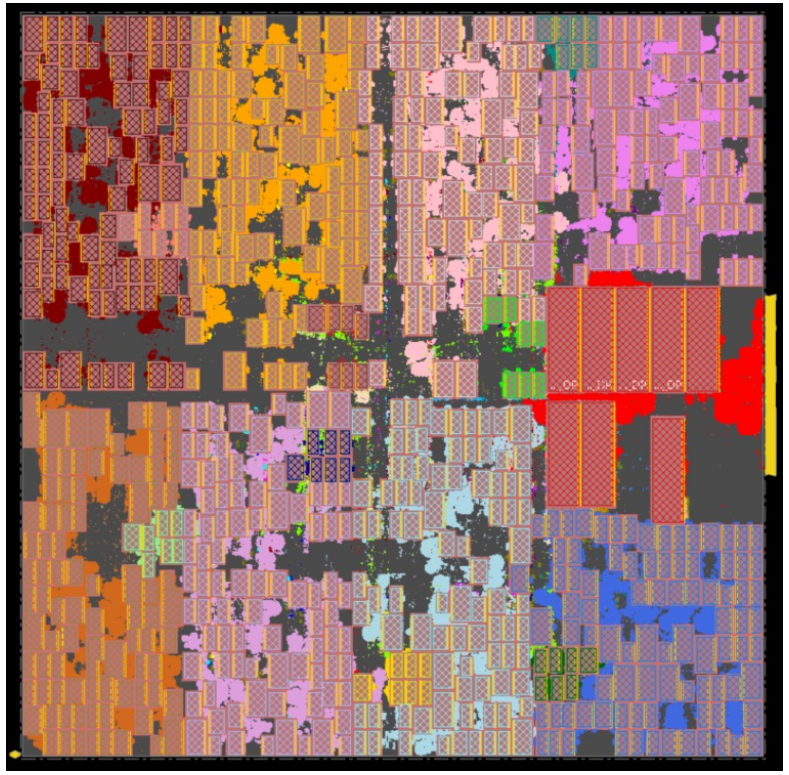
Many applications: data for machine learning, CI/CD infrastructure for software quality, …

Coming Fast: Agentic AI + Open-Source EDA

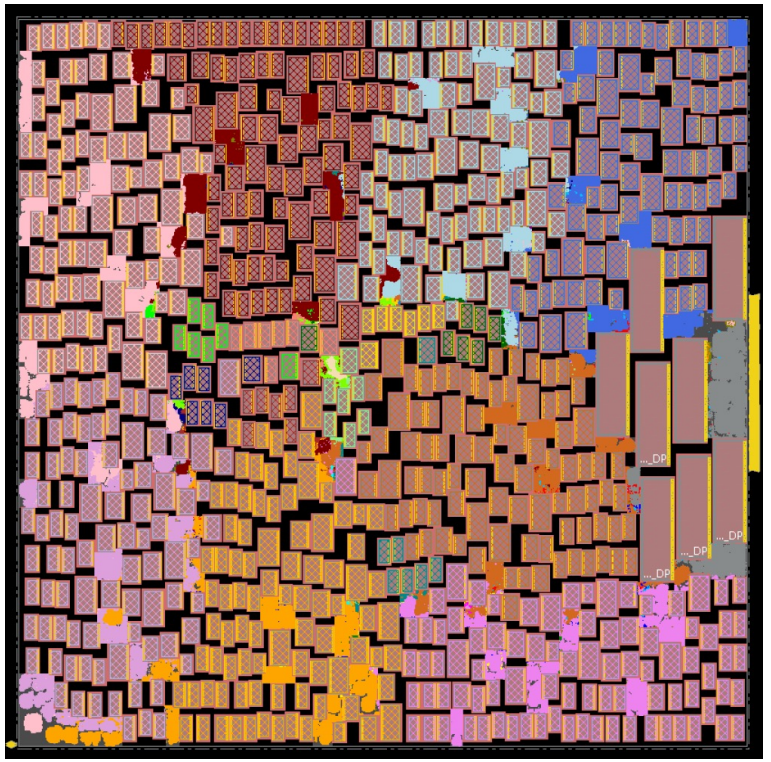
- **Open-source EDA**
  - Open source: “GitHub is a gold mine”
    - Historical record of development
    - Preference and temporal data: PRs merged/unmerged, issues opened/closed, commits
  - EDA: 70+ years of heuristics and literature = ripe for ML
  - **Vision: Agents become EDA R&D engineers !** *Agents as tool makers, not just tool users*
    - How We Will Make an Automated Digital IC Designer (.pptx)
    - Invited: Agentic AI for Physical Design R&D: Status and Prospects (.pdf) (.pptx)
    - Models for the Physical Universe of Semiconductor IC Design: What’s Different (and Difficult) (.pptx)
- **Docs are the basis**
  - EDA-scale repositories are massive
  - Human ramp-up times are too long
  - Docs compress 1M LoC into 3-5% of tokens → usable for MCP, LLMs, etc.
- **Many early proofpoints using OpenROAD (including from UCSD)**
  - ORFS-agent https://arxiv.org/abs/2506.08332
  - AuDoPEDA https://arxiv.org/abs/2601.06268

Snapshots / Vignettes: Agility and Extensibility

High-quality Engines: Hier-RTLMP Macro Placer



Hier-RTLMP (postRoute)



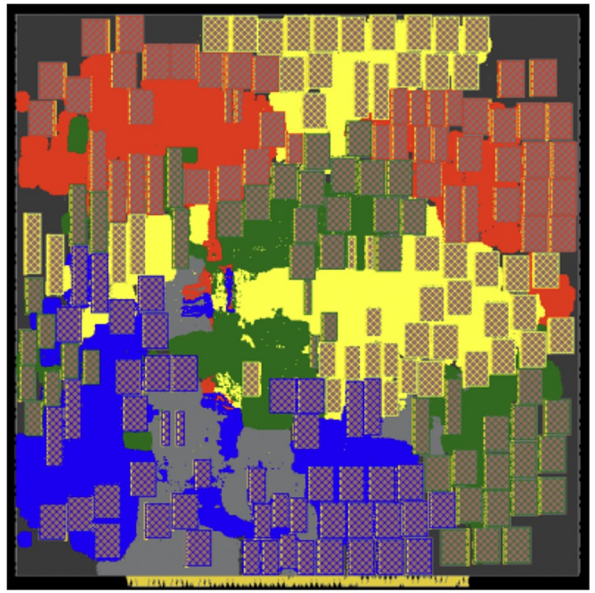
Comm Macro Placer (postRoute)

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

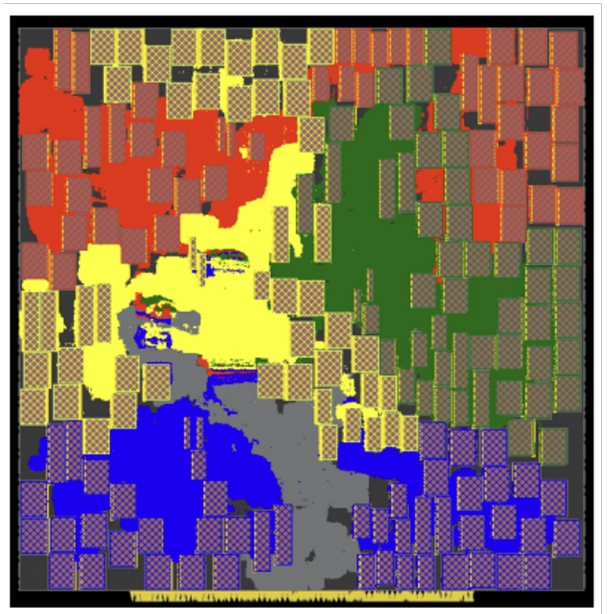
Macro Placer	Std Cell Area (mm^2)	Power (mW)	WNS (ns)	TNS (ns)
Hier-RTLMP	**0.160**	**640**	**-0.085**	**-0.417**
Comm	0.165	689	-0.370	-92.246

OpenROAD GitHub (/src/mpl2) , arXiv

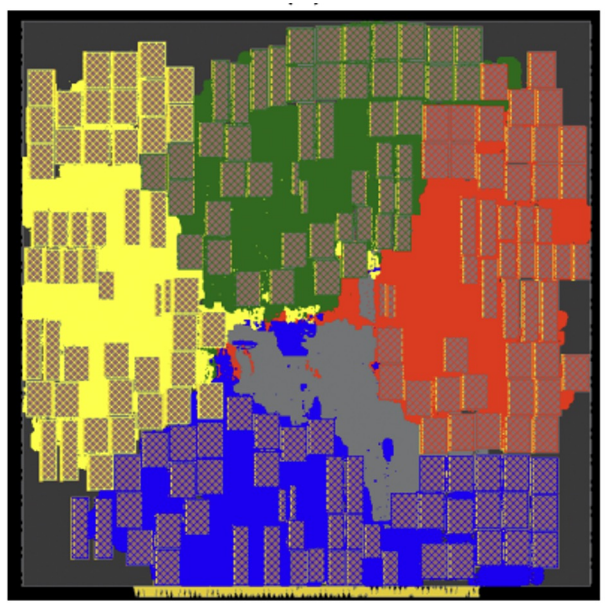
Dataflow-Aware GPU-Accelerated RePlAce arXiv



OpenROAD RePlAce



DREAMPlace

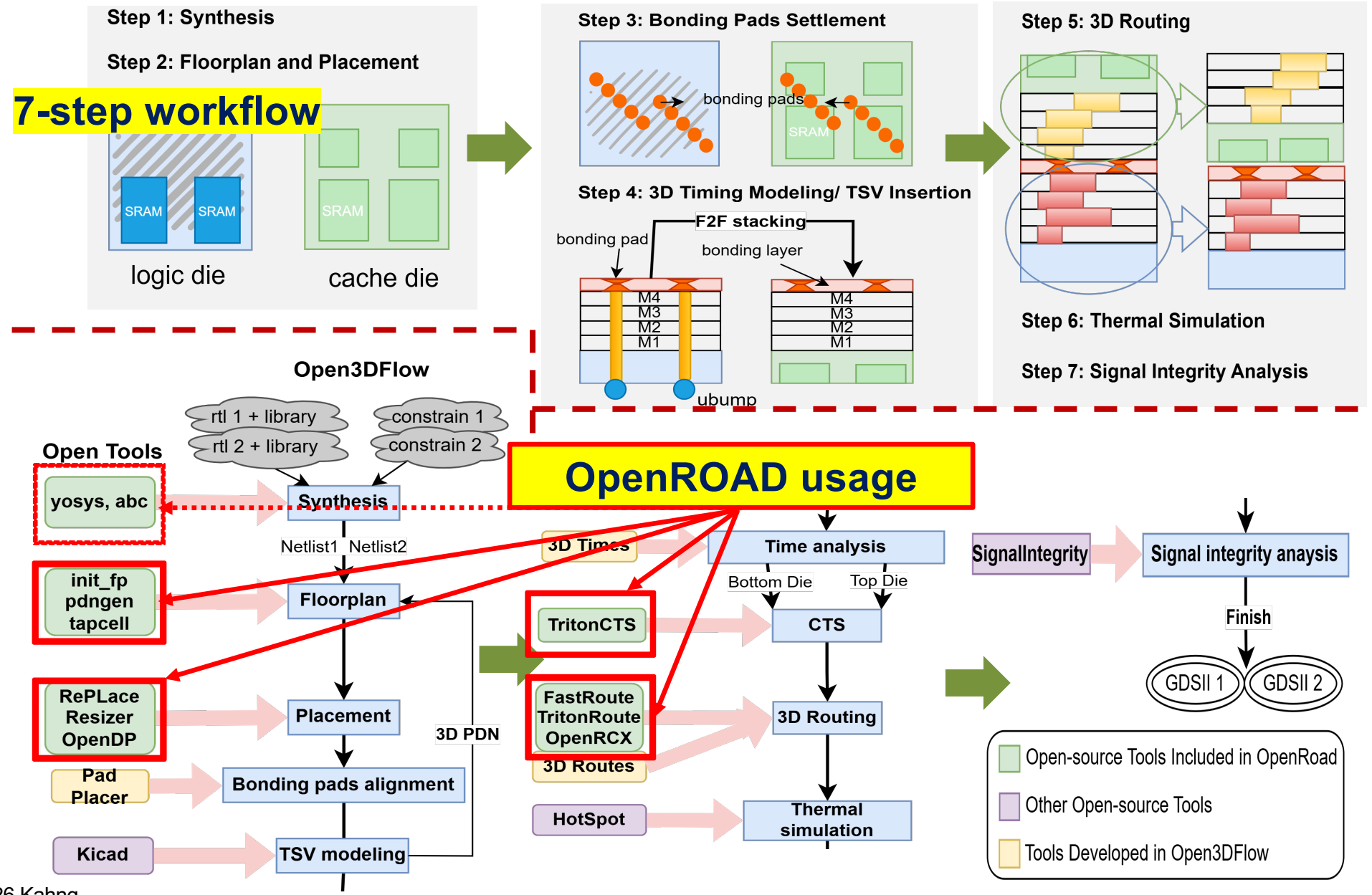


DG-RePlAce

Global Placer	WL	Power	WNS	TNS	GP (s)	TAT (s)
RePlAce	1.00	1.00	-0.123	-108.15	387	653
DREAMPlace	0.92	0.98	-0.023	-2.623	61	88
DG-RePlAce	0.90	0.97	-0.014	-0.078	32	200

Testcase: BlackParrot RISC-V (Quad-Core) (evaluator: INVS 21.1)
(827K stdcells, 196 macros in GF12LP)

Open3D Flow (Prof. Lei Ren, Tsinghua Univ. @DAC BOF)

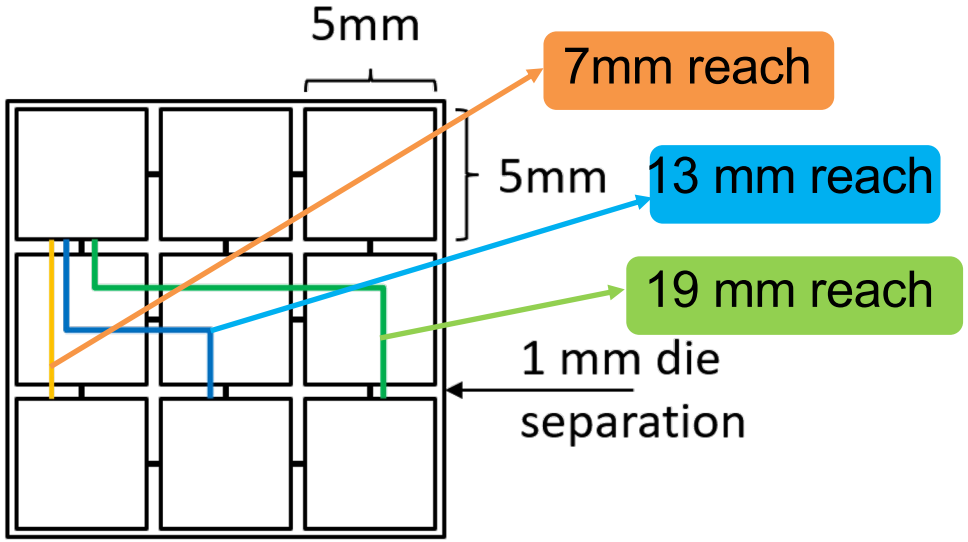


ChipletPart Partitioning

- **2.5D system partitioner** (GitHub)
  - Cost-aware (integration with CATCH chiplet cost model from UCLA)
  - Floorplan-aware (annealing-based chiplet floorplanning)
  - Technology-aware (chiplet technology assignments via genetic algorithm)
  - Core partitioner is based on OpenROAD's TritonPart
- **Up to 23% improvement** in chiplet cost with **heterogeneous** technology compared to **homogeneous** integration

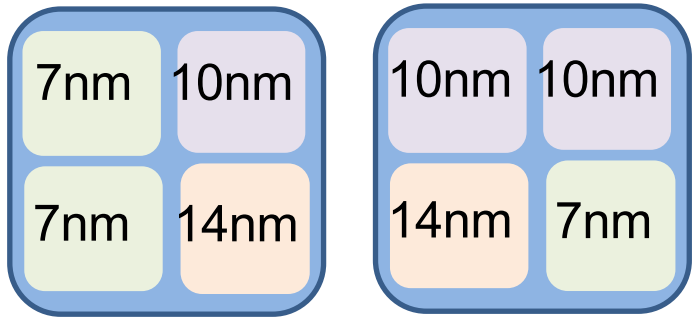
Floorplan awareness

Reach constraints → max. wirelength that can be driven by IO cells



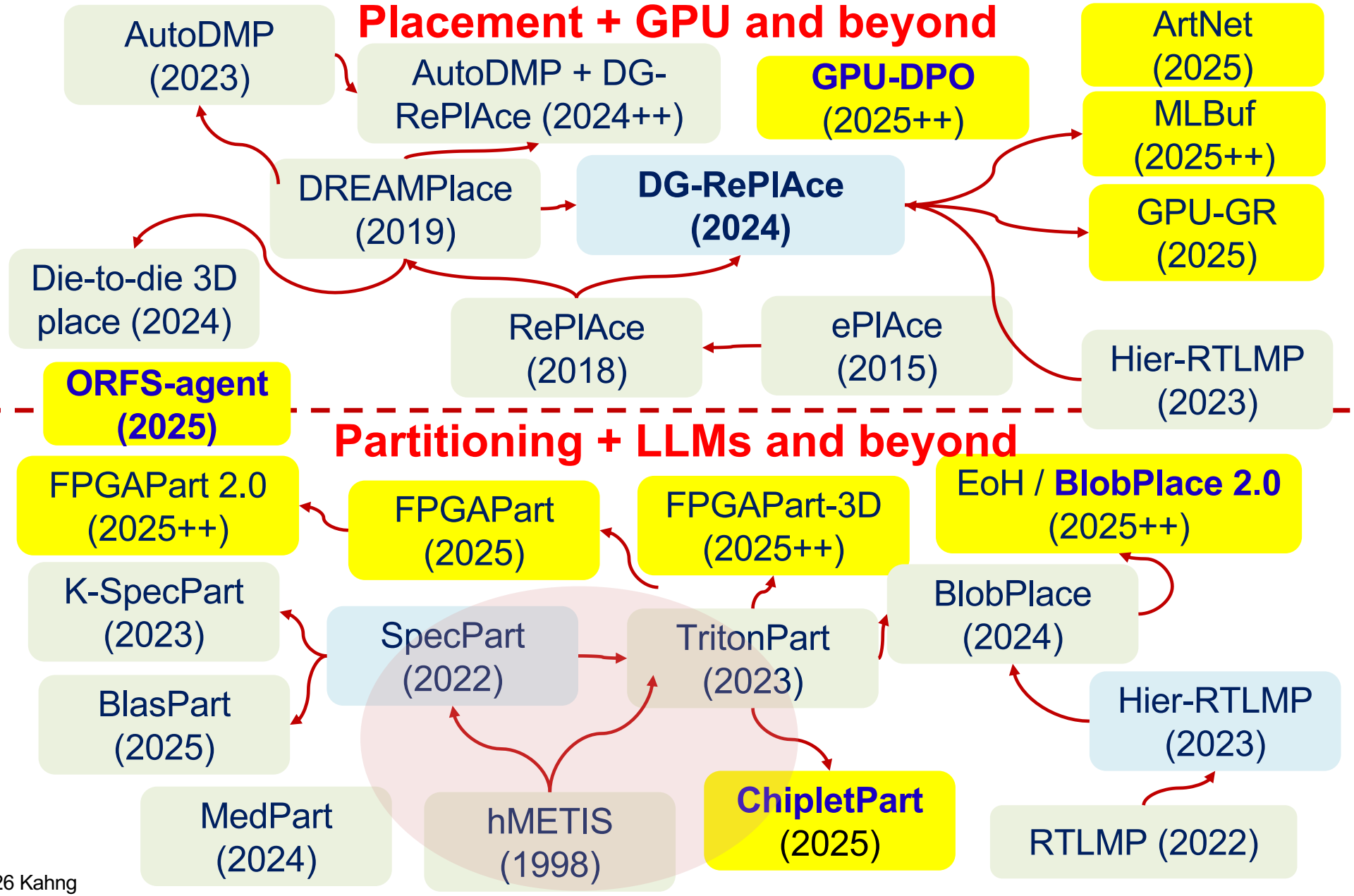
Technology awareness

Different chiplets implemented in different technology



Which partitioning solution is better?

Open-Source EDA Simply Moves Faster !!!



Next Time: OpenDB

BACKUP

Can agents improve EDA tools?

- RTL-to-GDS flows are **massive + poorly documented**
- Onboarding new engineers takes **considerable time**
- LLMs succeed at coding – but **struggle** at repo-scale reasoning
- **Hypothesis** – *"If EDA expertise is acquired by reading code + papers + flows then an agent can be onboarded the same way"*

- **Repository-grounded** coding agent
- **Literature-informed** planning
- **Closed-loop** QoR validation

AuDoPEDA

Outcomes:

- Up to: **5.9%** routed WL reduction, **10%** ECP reduction, **19.4%** power reduction

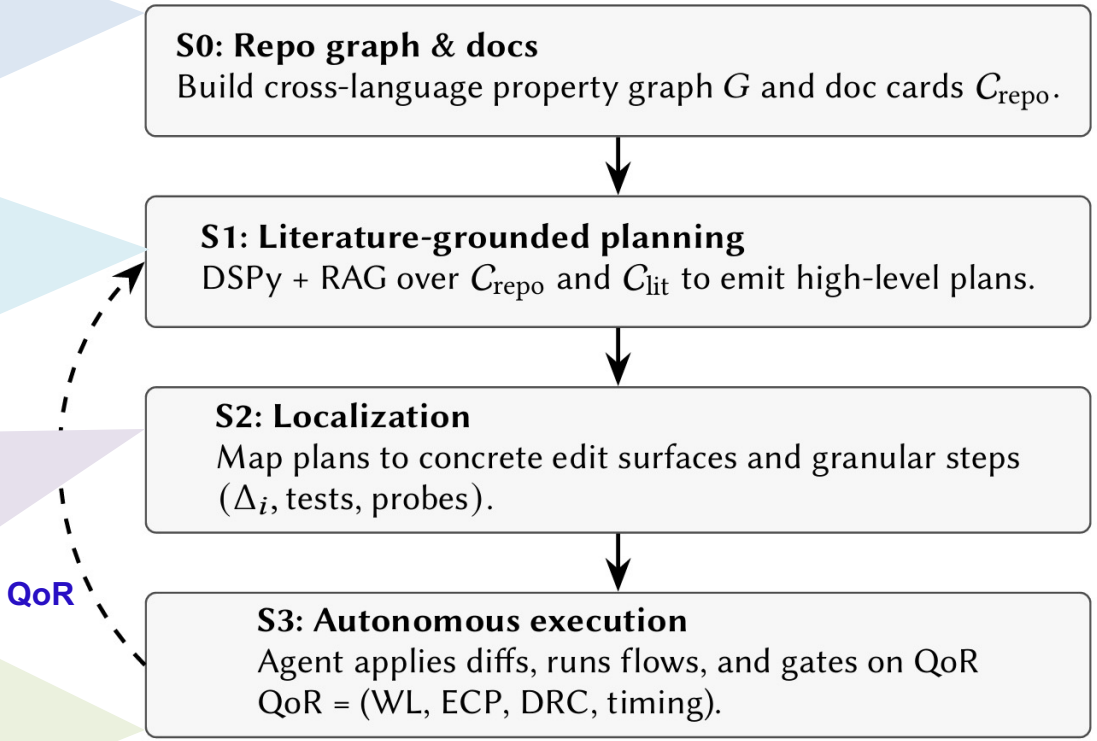
AuDoPEDA: *Documentation → Planning → Diff → QoR Loop*

Convert codebase to structured knowledge
- Parse OpenROAD via tree-sitter
- Build cross-language property graph
- Generate hierarchical doc cards
- Create machine-readable API/invariant summaries

Planning is typed and validated
- Retrieve from repo docs, EDA literature corpus
- DSPy-compiled planning program
- Outputs: research hypothesis, proposed interventions, QoR targets

Converts research intent into executable diffs
- Map plan → specific files/functions
- Generate granular diff plan: pre-checks, run config, QoR probes, rollback conditions

Accept only if QoR improves
- Codex-class agent
- Apply diffs
- Build
- Run flow
- Extract QoR
- Hill climb under safety gates



Some results

WL improvement diffs

- Modified detailed placement operators
- Added routability-aware cost
- Learned WL vs. congestion weights

Platform	Design	Base rWL	Our rWL	ΔrWL (%)
ASAP7	aes	64,640	62,710	−2.99
ASAP7	ibex	80,402	80,823	+0.52
ASAP7	jpeg	154,484	152,232	−1.46

ECP improvement diffs

- Modified gpl + rsz modules
- Added: slack-weight shaping, length-aware weighting, repair-loop convergence controls

Design	TCP	Base ECP	Our ECP	ΔECP
ariane133	3.4	3.59	3.43	-4.6%
ariane136	3.0	3.78	3.41	-10.0%
bp_fe	1.53	1.71	1.65	-3.4%
swerv_wrapper	1.7	2.19	2.16	-1.4%

Power improvement diffs

- Replaced path-based sizing
- Instance-level ranking
- Multi-pass: downsizing, buffer removal, VT swapping

Platform	Design	Base Pwr	Our Pwr	ΔPwr (%)	ΔECP (%)
ASAP7	aes	153.74	150.75	−1.947	9.836
ASAP7	ibex	58.10	47.24	−18.703	2.606
ASAP7	jpeg	119.70	117.48	−1.853	2.880

https://hub.docker.com/r/abkgroupucsd/audopeda

More details: https://arxiv.org/abs/2601.06268

Placement-EVO – main idea

- **Placement-EVO leverages LLMs + multi-agents**
- **Automatically discovers** + **improves** placement algorithms in the OpenROAD tool

- Natural language request → automatic target discovery → code evolution of target → improved algorithms
- **Rich corpus + agents + end-to-end automation**

Evolution loop

Optimize detailed placement in OpenROAD

Target discovery

Corpus query → improvePlacement, detailedPlacement, …

Knowledge loading

Literature cards, idea pool, pseudocodes of target functions

Baseline loading

Load baseline worktree, collect metrics, logs

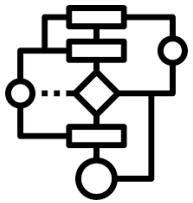
Initial population

Baseline + mutated codes, evaluate **fitness scores**

Generation loop

Pick operator (crossover vs. mutation), generate plan, Implement plan, collect fitness scores, run for generations

Final population



Algorithm with highest fitness score

Fitness score

$$w_1 \times \frac{HPWL_{baseline}}{HPWL_{candidate}} - w_2 \times \max\left(0, \frac{Runtime_{candidate}}{Runtime_{baseline}}\right)$$

Higher fitness score is better
w1, w2 are hyperparameters

Results with dpl's *improvePlacement*

- **Baseline:** default ***OpenROAD's*** dpl
- **Designs:** aes, jpeg, ibex, swerv, cva6, ariane37 w. ASAP7
- **User query:** "*Improve OpenROAD's improvePlacement for better HPWL*"
- Up to 3% reduction in HPWL and RWL compared to baseline

Design	Default-Detailed-Placer				Evolved-Detailed-Placer			
	HPWL (μm)	DP-rt (s)	RWL (μm)	RWL-rt (s)	HPWL (μm)	DP-rt (s)	RWL (μm)	RWL-rt (s)
aes	50507	24	62197	265	**49760**	**20**	**61349**	275
ibex	60746	22	83311	457	**59980**	**21**	**82924**	**416**
jpeg	128336	55	155633	718	**124319**	59	**151601**	757
cva6	540304	109	652133	2056	**534604**	138	**646855**	2267
swerv	892296	168	1102150	2453	**885881**	250	**1094216**	2825
ariane37	1243281	185	1445326	2852	**1202847**	219	**1402549**	2924