



# An Update on OpenROAD

---

**Zhiang Wang**  
Assistant Professor, Fudan University  
[zhiangwang@fudan.edu.cn](mailto:zhiangwang@fudan.edu.cn)

20251031

# Biography



- **Zhiang Wang (王志昂)**

- Assistant Professor at Fudan University (from August)
- PostDoc at UC San Diego (2024/03 – 2025/07)
- PhD at UC San Diego, Advisor: Andrew Kahng (2019/09 – 2024/03)
- BS at University of Science and Technology of China (USTC), 2019

- **Research area:**

- Digital physical design
- System Technology Co-Optimization
- Open-source EDA
- GPU-accelerated EDA

# Agenda



- **Research Overview**
- **Deep Dive of OpenROAD (My Contributions)**
  - Partitioning
  - Macro Placement
  - GPU-accelerated Global Placement
  - GPU-accelerated Detailed Placement
  - GPU-accelerated Routing
  - ORFS-agent: LLM-based Flow Tuning for OpenROAD
- **Future Directions and Ongoing Works**

# Research Overview

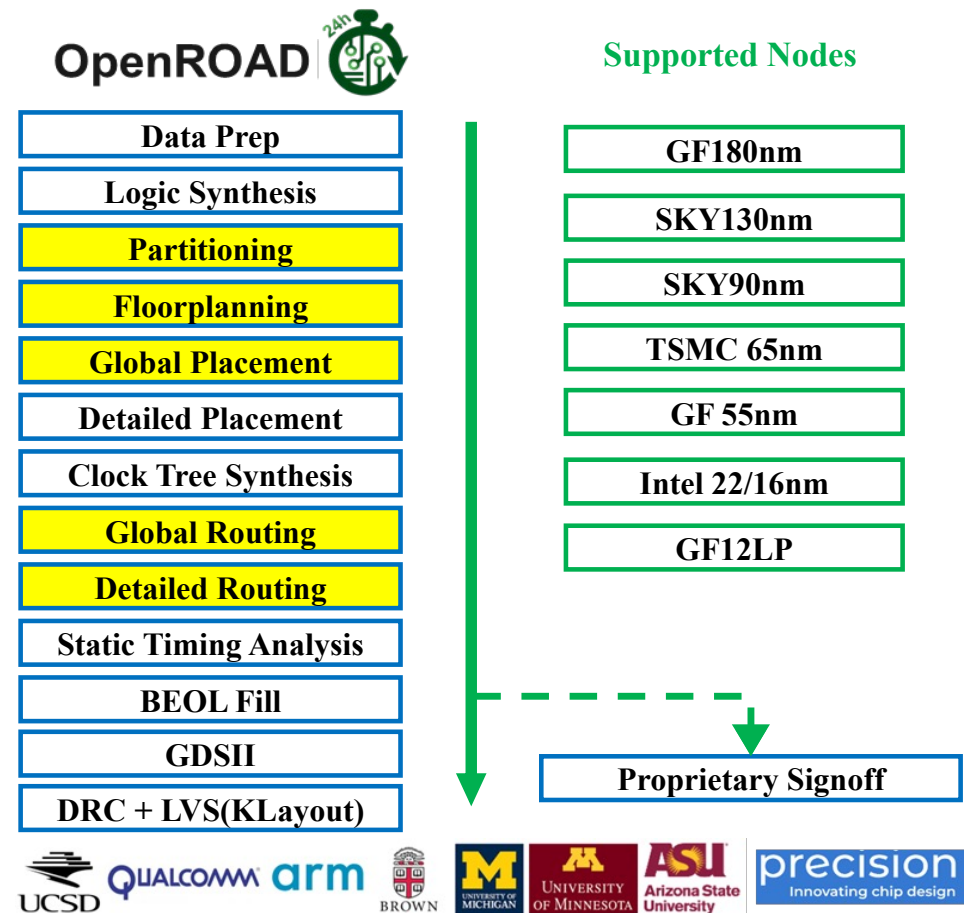


- **Open-Source EDA for Digital Electronics Designs**

- OpenROAD Infrastructure
- IEEE CEDA DATC Robust Design Flow

- **Optimization in VLSI CAD**

- Partitioning
- Floorplanning
- Placement (GPU-accelerated)
- Routing (GPU-accelerated)
- LLM-based autotuning



# Contests Built on OpenROAD



OpenROAD



## 2025 Contests

### ISPD

Performance-Driven  
Large Scale Global  
Routing

### MLCAD

ReSynthAI: Physical-Aware  
Logic Resynthesis for Timing  
Optimization Using AI

### ICCAD

Incremental Placement  
Optimization Beyond Detailed  
Placement [Problem C]

## 2024 Contests

### ICCAD

Scalable Logic Gate Sizing Using ML  
Techniques and GPU Acceleration  
[Problem C]

### ICLAD

GenAI Chip  
Hackathon @DAC

## 2023 Contests

### ICCAD

Static IR Drop Estimation Using Machine Learning  
[Problem C]

Also:

- **ICCAD19** LEF/DEF Based Global Routing
- **ISPD26/27** Buffering and Sizing

# Agenda



- Research Overview
- **Deep Dive of OpenROAD (My Contributions)**
  - Partitioning
  - Macro Placement
  - GPU-accelerated Global Placement
  - GPU-accelerated Detailed Placement
  - GPU-accelerated Routing
  - ORFS-agent: LLM-based Flow Tuning for OpenROAD
- Future Directions and Ongoing Works

# TritonPart: 21<sup>st</sup> Century Netlist Partitioner



- **Open-source replacement for hMETIS in all contexts**
  - Integrated with OpenROAD (src/par in OpenROAD)
  - Published at ICCAD 2023
- **Key features (constraints-driven partitioning engine)**
  - **Real-valued** multi-dimensional vertex weights e.g., multi-FPGA resources
  - **Multi-dimensional** balance constraints e.g., satisfy multi-FPGA balance
  - **Community constraints**: groups of vertices that stay together during partitioning e.g., keep macros and their direct fanins/fanouts together
  - **Multi-way** partitioning
  - **Embedding-aware** partitioning e.g., placement coordinates
  - **Timing-driven** partitioning e.g., minimize cuts on critical paths
- **Key results:**
  - Improvements over hMETIS up to ~20% on some benchmarks
  - ~21X reduction of cuts on timing-critical paths compared to hMETIS and KaHyPar



**source code**

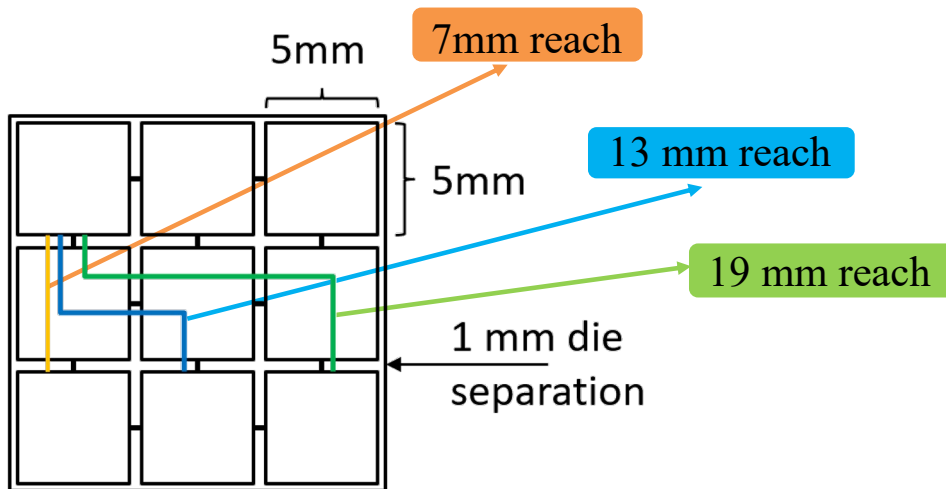
# Extension To Chiplet Partitioning: ChipletPart



- Cost-aware (integration with CATCH chiplet cost model from UCLA)
- Floorplan-aware (annealing-based chiplet floorplanning)
- Technology-aware (chiplet technology assignments via genetic algorithm)
- **Up to 23% improvement** in chiplet cost with **heterogeneous** technology compared to **homogeneous** integration

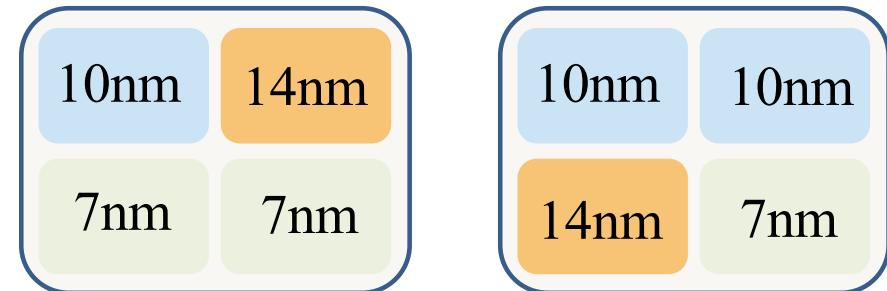
## Floorplan awareness

Reach constraints → max. wirelength driven by IO cells



## Technology awareness

Different chiplets implemented in different tech. nodes



Which partitioning solution is better?

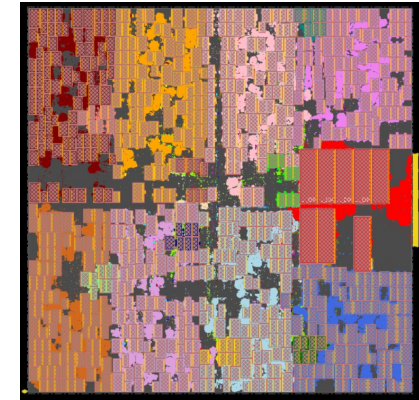


# Early Design Space Exploration (Arch, RTL)



- Can we better explore architecture, RTL, and SoC floorplan design spaces?
  - **Ideal: ultra-fast, yet match actual implementation**
- Hier-RTLMP (src/mpl in OpenROAD): RTL- and dataflow-driven, human expert-like results

## TILOS Macro Placement Benchmarks



Results for an AI Accelerator  
(GF12LP, 760 macros)

Macro Placement - benchmarks, evaluators, and reproducible results from leading methods in open source

📖 Readme  
📄 BSD-3-Clause license  
🔔 Activity  
📁 Custom properties  
★ 283 stars  
👁 9 watching  
🍴 49 forks  
Report repository

### Table of Contents

- [Recent Updates](#)
- [FAQs](#)
- [Testcases](#) contains open-source designs such as Ariane, MemPool and NVDLA.
- [Enablers](#) contains PDKs for open-source enablers such as NanGate45, ASAP7 and SKY130HD with FakeStack. Memories required by the designs are also included.
- [Flows](#) contains tool setups and runscripts for both proprietary and open-source SP&R tools such as Cadence Genus/Innovus and OpenROAD.
- [Code Elements](#) contains implementation of engines such as Clustering, Grouping, Gridding as well as Format translators required by Circuit Training flow.
- [Baseline for Circuit Training](#) provides a baseline for [Google Brain's Circuit Training](#).
- Links to Further Technical Details and Chronology
  - [Our Progress: A Chronology](#) provides latest updates.
  - [Proxy Cost](#) documentation gives implementation details to enable reproduction of the wirelength, density and congestion costs used by [Circuit Training](#).
- [Other Links](#)

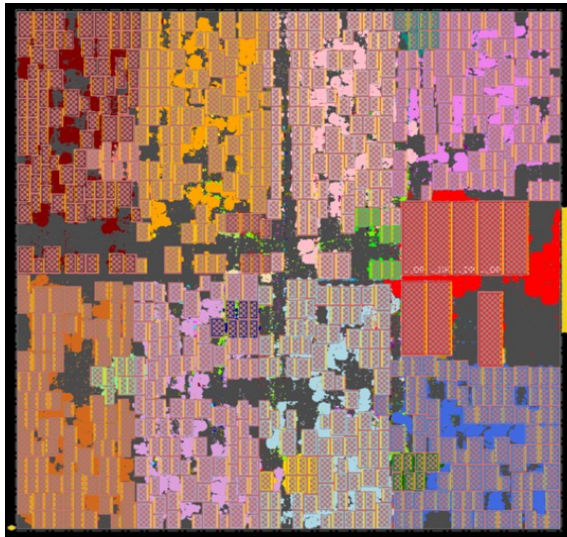


Scan me for TILOS  
benchmarks

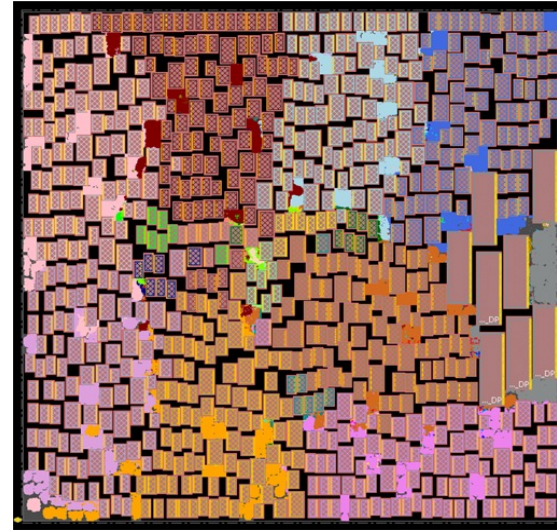
# Hier-RTLMP vs. Commercial Macro Placer



- Results for an AI Accelerator (GF12LP, 760 macros)



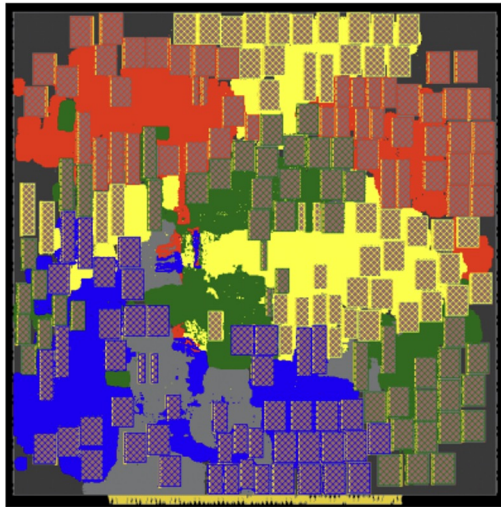
Hier-RTLMP (PostRoute)



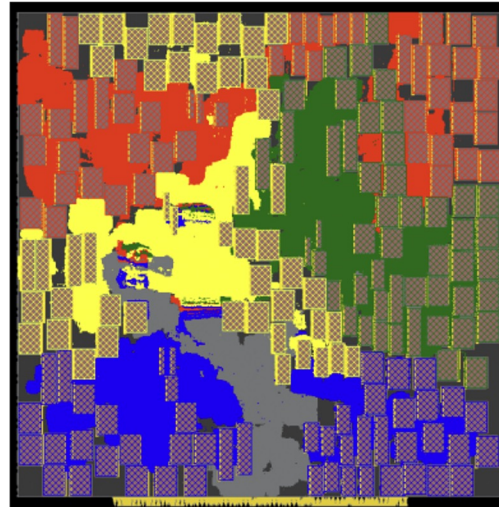
Commercial Macro Placer (PostRoute)

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

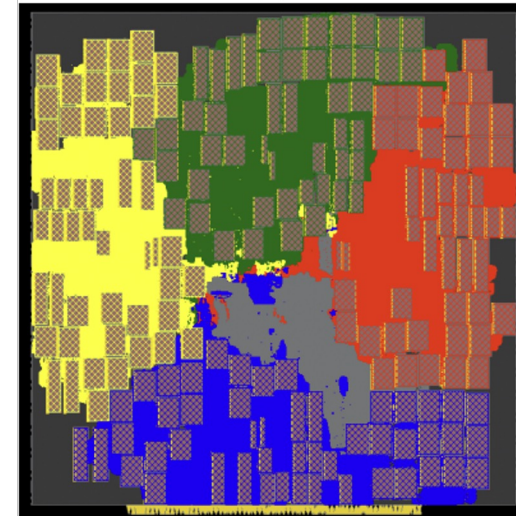
# Dataflow-Driven GPU-Accelerated RePlAce (DG-RePlAce)



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)



# Speed Enables Autotuning and Better Quality



## Step 1: Specify hyperparameters

### Hyperparameters (specified in configspace.json)

- coarsening\_ratio: range = [6, 20], type = int
- max\_num\_level: range = [1, 2], type = int
- virtual\_iter: range = [1, 8], type = int
- num\_hops: range = [1, 8], type = int
- halo\_width: range = [1.0, 3.0], type = float
- target\_density: range = [0.5, 0.8], type = float



```
A total of 29 unique configurations were sampled.
A total of 29 runs were executed.
The run took 10548.9 seconds to complete.
# Pareto-optimal points = 9
```

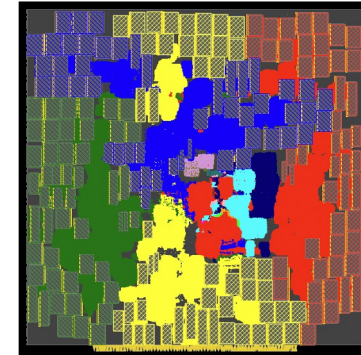
	rsmt	congestion	density
6	1.07373e+07	70.18	0.631724
11	1.10367e+07	66.1	0.503092
14	1.09998e+07	69.44	0.508124
17	1.08384e+07	70.97	0.554149
18	1.07772e+07	68.33	0.581476
22	1.08833e+07	69.77	0.558474
25	1.08008e+07	64.91	0.563338
26	1.07329e+07	68.42	0.68759
27	1.08633e+07	77.63	0.550306

Pareto candidates:

	rsmt	congestion	density
14	1.09998e+07	69.44	0.508124
17	1.08384e+07	70.97	0.554149
18	1.07772e+07	68.33	0.581476
26	1.07329e+07	68.42	0.68759
27	1.08633e+07	77.63	0.550306

## Step 2: Bayesian Opt. / NSGA-II tuner

**Early evaluation  
is done by  
global router  
in OpenROAD**



Post-route layout of RUN\_ID = 14



RUN_ID	WL	Power	WNS	TNS
default	0.90	0.972	-0.014	-0.078
14	0.86	0.967	-0.002	-0.007
17	0.85	0.971	-0.014	-1.048
18	0.86	0.968	-0.012	-0.216
26	0.85	0.969	-0.027	-1.794
27	0.86	0.970	-0.007	-0.139

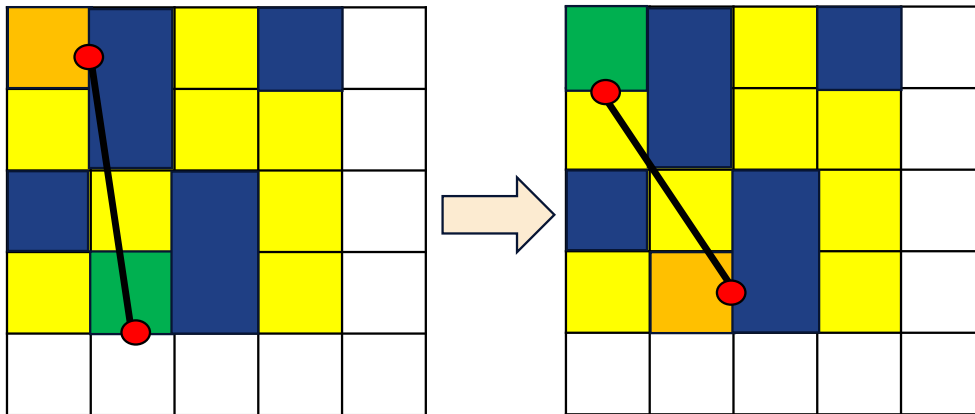
## Step 3: Run INVS P&R for Pareto candidates

# GPU-Accelerated Detailed Placement Optimizer

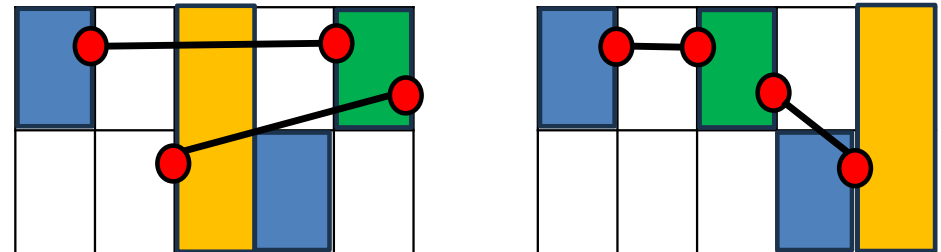


- First known implementation of GPU-accelerated detailed placement operators that move multi-height cells cf., e.g., ABCDPlace
  - GPU-accelerated global swap
  - GPU-accelerated local reordering
  - GPU-accelerated maximum Independent set matching
- Considers constraints that help maintain routability

**30X Speedup !!!**



Global Swap

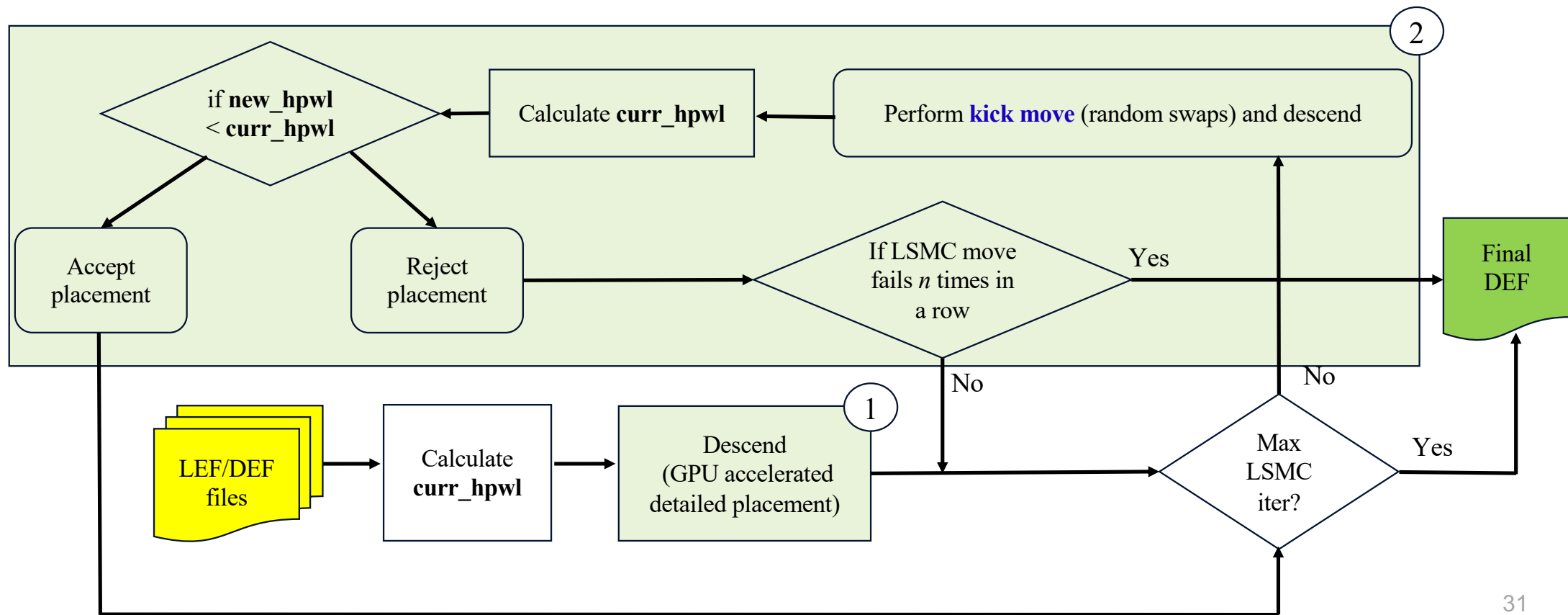


Local Reordering

# LSMC Framework for Better Quality (GPU-DPO)



- LSMC (Large-Step Markov Chain) metaheuristic enables better exploration of solution space efficiently on GPU, especially in high-density placements



# LSMC Framework for Better Quality



- GPU-DPO achieves 1.7% and 3.5% lower post-detailed placement HPWL compared to DPO and ABCDPlace

Testcase (Utilization)	Cells	Detailed Placer	HPWL (um)	DP Time (s)	TAT (s)
AES (0.91)	15K	DPO	44823	5	10
		ABCDPlace	45412	<b>1</b>	<b>4</b>
		GPU-DPO	<b>44226</b>	2	4
JPEG (0.72)	61K	DPO	98092	34	42
		ABCDPlace	101537	<b>3</b>	<b>10</b>
		GPU-DPO	<b>93665</b>	5	13
Mempool-Group (0.41)	2548K	DPO	25089409	1138	1375
		ABCDPlace	25102382	<b>24</b>	<b>77</b>
		GPU-DPO	<b>24963574</b>	35	164

Testcases with multi-height cells: AES, JPEG, MemPool-Group  
Platform: ASAP7 Evaluator: OpenROAD



# GPU-Accelerated Global Routing



- **Motivations:**

- FastRoute (2012, default in OpenROAD) suffer inefficiency on large-scale high-utilization testcases.
- Contest-driven CUGR-based routers only work on contest testcases.

**An open-source global router for large-scale high-utilization real testcases !**

- **Our Goals:**

- **Excellent scalability and superior speed:** 50M nets in half an hour
- **High quality:** achieve better performance on high-utilization real testcases.
- **Fully open-source:** integrated into OpenROAD
- **Easy-to-use:**
  - Real global router in routing stage → better wirelength, less #DRVs
  - Early global router in placement stage → congestion/timing estimation

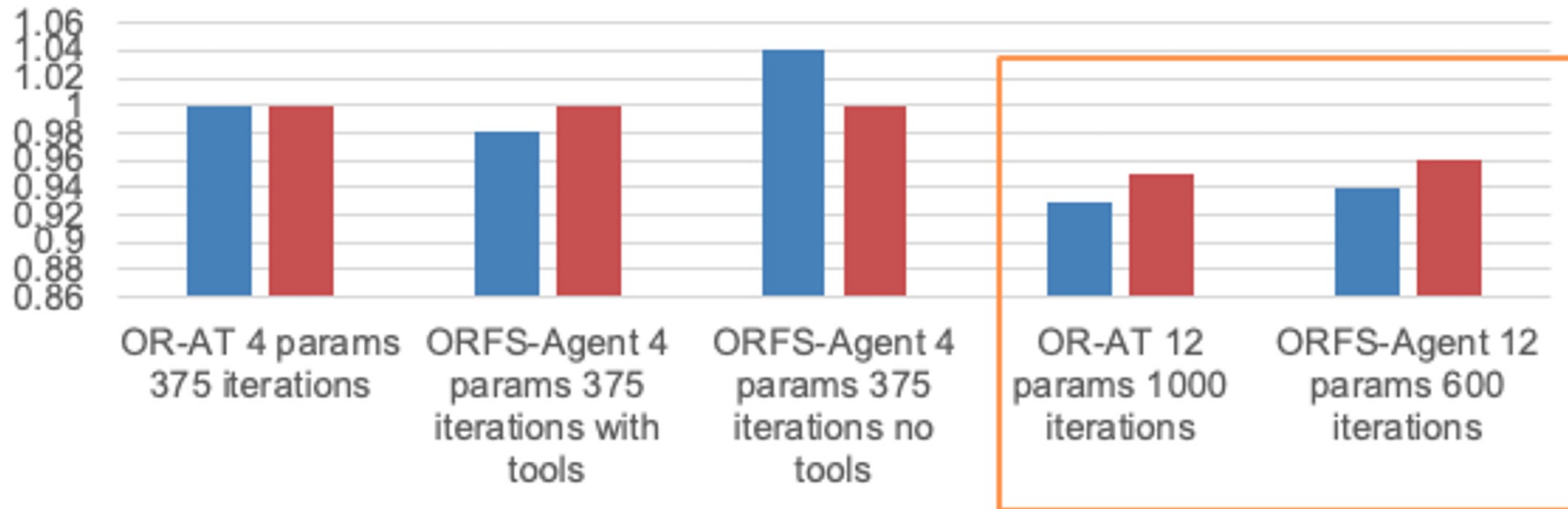


# ORFS-agent (MLCAD Best Paper Award 2025)



- **Autotunes OpenROAD flow using batch-based LLM exploration**
  - Built atop Claude-3.7 (Anthropic), now we are exploring DeepSeek
- Batch exploration: runs 25 OpenROAD jobs **in parallel**, each with different parameters
- Training data: generates <parameter set, quality metrics> tuples (e.g., rWL, ECP)
- Learning loop: LLM observes results and **proposes better configurations** over time
- Goal: discover “optimal” flow (tool) parameters
- Git: <https://github.com/ABKGroup/ORFS-Agent/tree/main>
- Paper: <https://vlsicad.ucsd.edu/Publications/Conferences/417/c417.pdf>
- Slides: [https://vlsicad.ucsd.edu/Publications/Conferences/417/c417\\_slides.pdf](https://vlsicad.ucsd.edu/Publications/Conferences/417/c417_slides.pdf)

# ORFS-agent vs. OR-AutoTuner



Comparison of ORFS-agent and OR-AutoTuner w.r.t. **wirelength** and **ECP**

Normalization: Results with OR-AT4 params and 375 iterations set as 1.0

- Baseline: OR-AT (4 vars, 375 iters)  $\equiv 1.0$
- ORFS-agent can achieve  $\approx 40\%$  **fewer iters iso-QOR**  $\approx 13\%$  gains in WL or ECP (single-objective) (details in the paper)

# Agenda

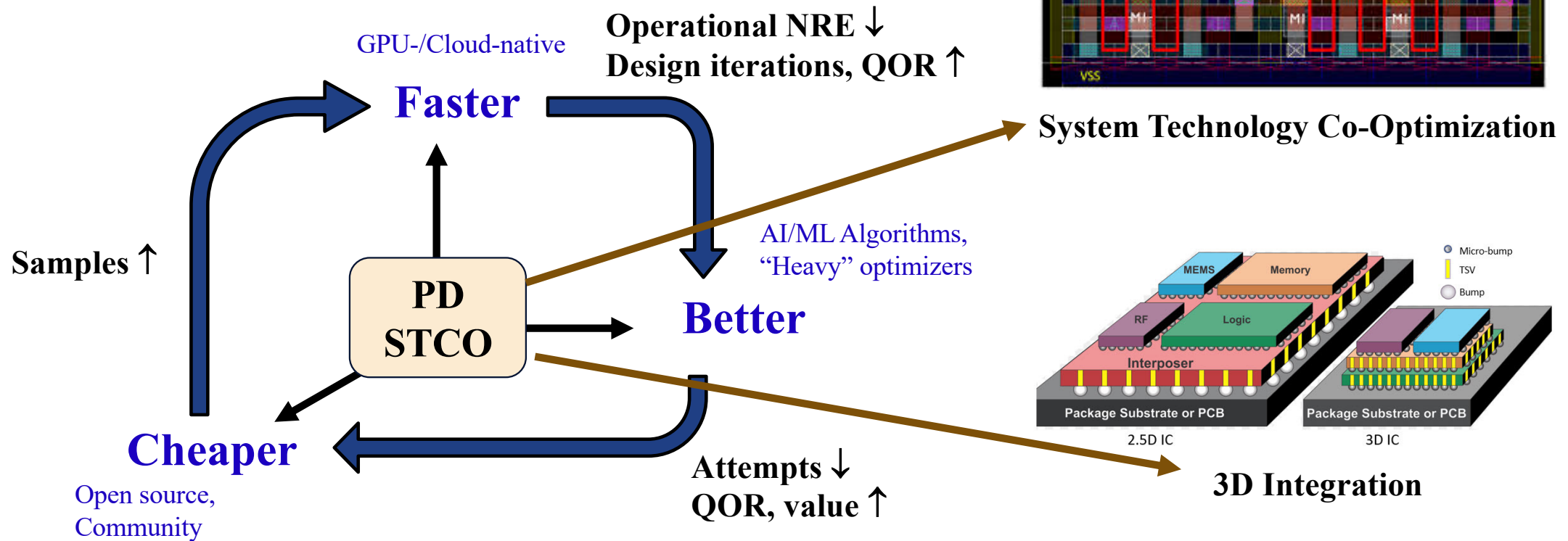


- **Research Overview**
- **Deep Dive of OpenROAD (My Contributions)**
  - **Partitioning**
  - **Macro Placement**
  - **GPU-accelerated Global Placement**
  - **GPU-accelerated Detailed Placement**
  - **GPU-accelerated Routing**
  - **ORFS-agent: LLM-based Flow Tuning for OpenROAD**
- **Future Directions and Ongoing Works**

# Goal: Faster, Better and Cheaper EDA



- “Faster, Better, Cheaper – pick any two” (it’s the law !)
- Question: Can open-source EDA give us all three at once?



# Physical Design for 3D IC (F2F)



## Current OpenROAD flow

### **TritonPart**

Multiple-constraints driven  
partitioning multi-tool

### **DG-RePlAce**

GPU-accelerated  
dataflow-driven global placer

### **TritonRoute-WXL**

State-of-the-art open-source  
global-detailed router

## 3D Extension

### **Co-DG-RePlAce**

Co-optimize tier partition and  
instance placement  
(GPU-accelerated,  
dataflow-driven)

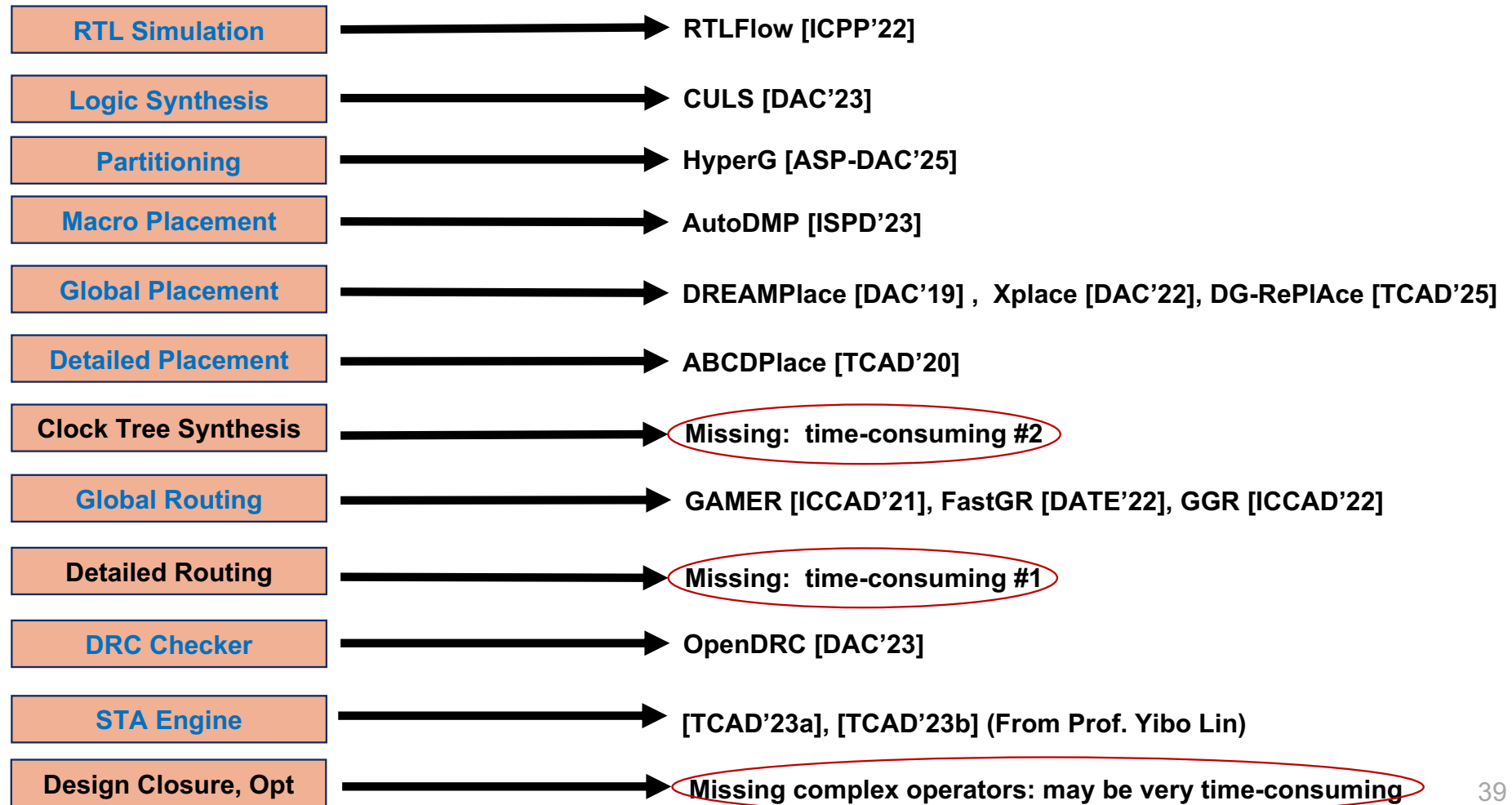
### **3D-TRoute (future work)**

# To Do: Fill in GPU-accelerated PD Flow



## Physical Design Flow

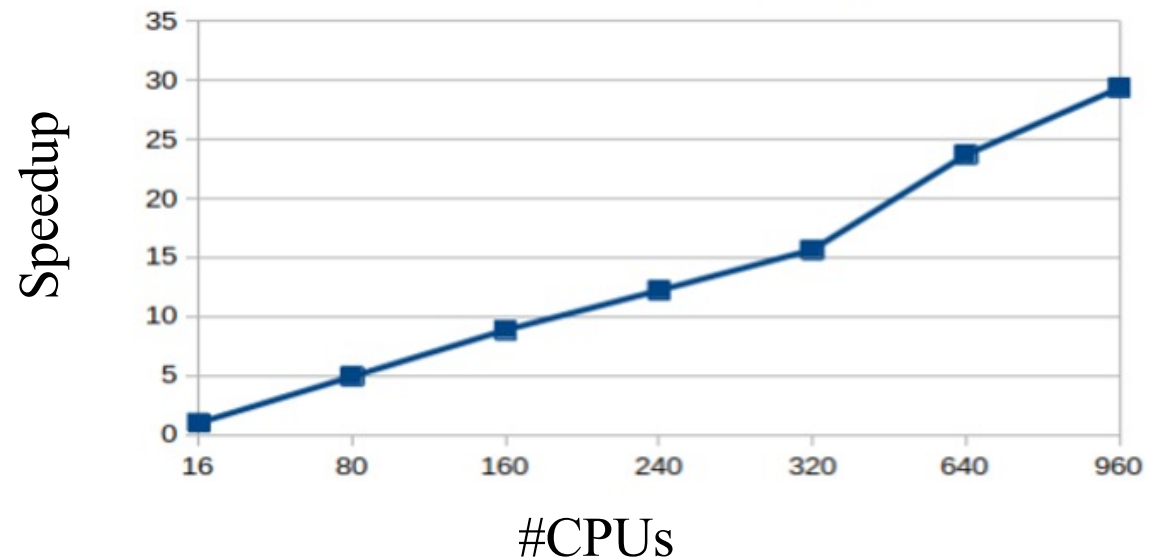
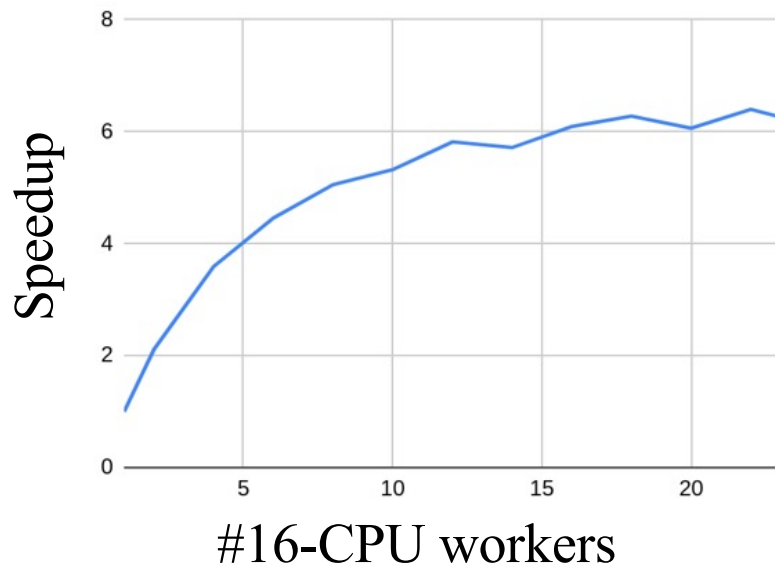
## Academic GPU-Accelerated Tools



# Distributed Computing for Physical Design

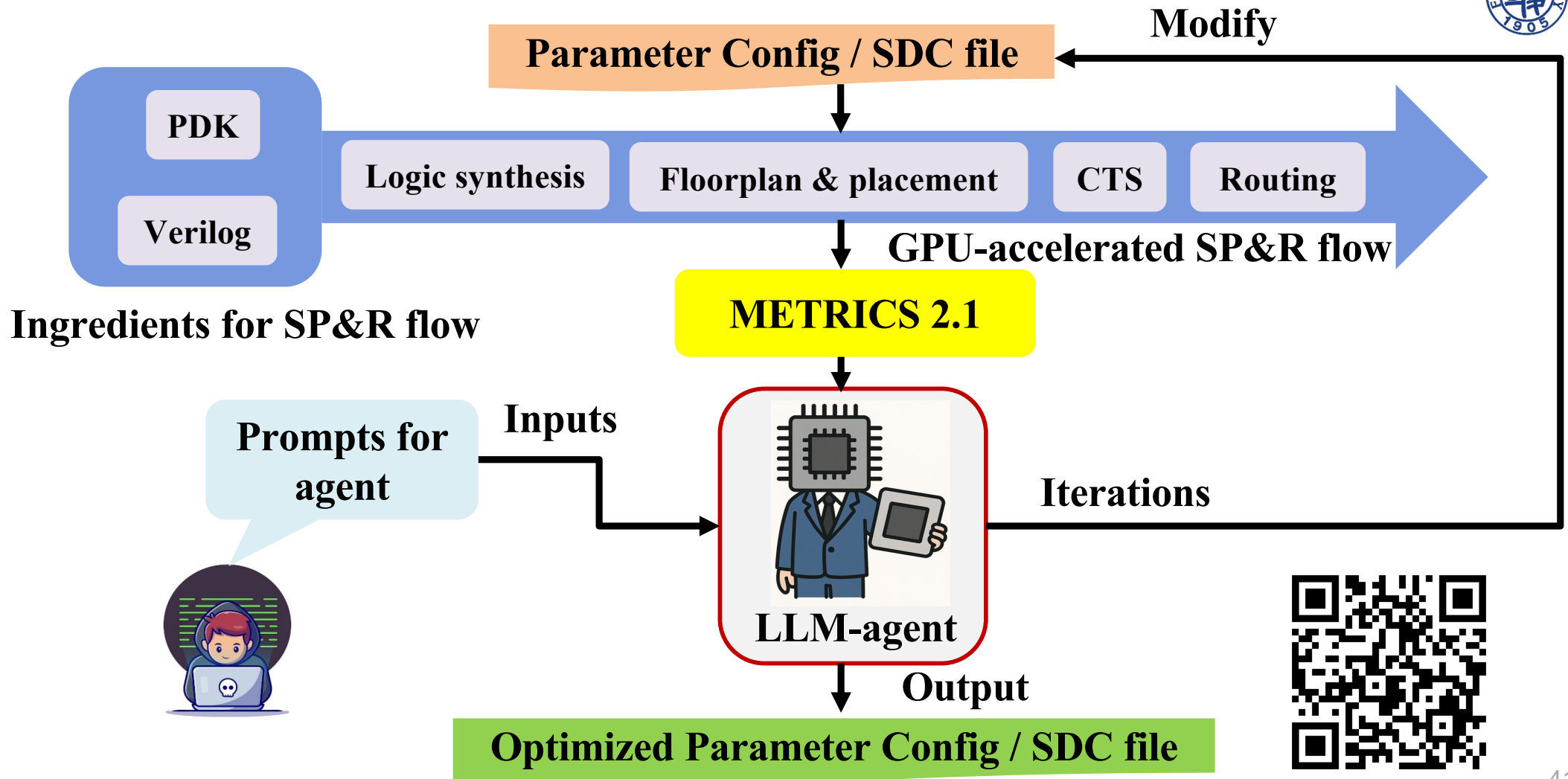


- Hardware limits the adoption of GPU-accelerated PD for extremely large designs
- Cloud deployment complements GPU acceleration
  - Cloud deployment + Open-source EDA (no license cost) → Faster and Better Solution
- Distributed incremental DR: ~100X speedup w/20 16-core workers
- Cloud-based pin access analysis: 30X speedup



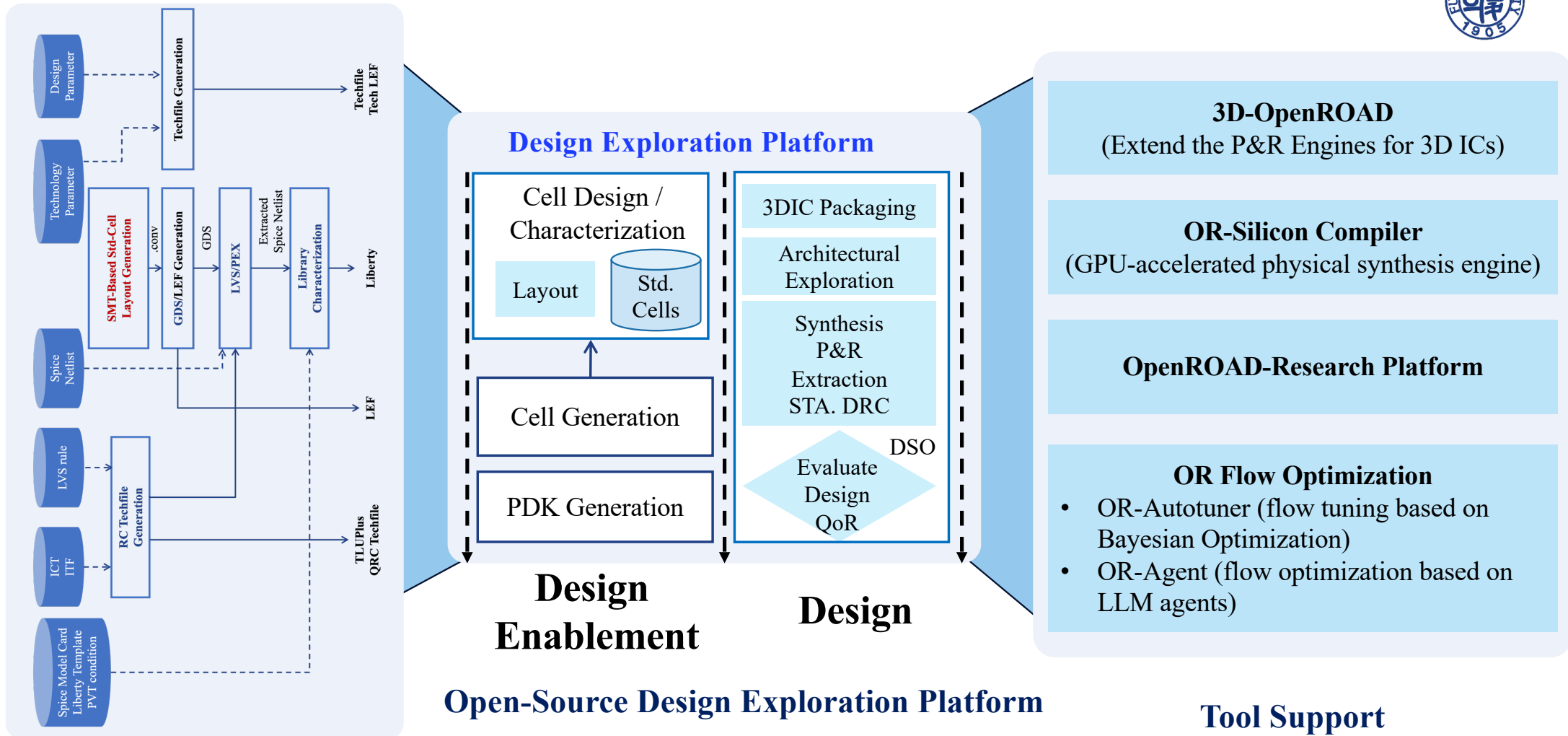
These data are from Professor Andrew B. Kahng's ISPD24 talk.

# LLMs Meet GPU-Accelerated Physical Design





# Support System-Technology Co-Optimization (STCO)



PROBE3.0 Framework

Open-Source Design Exploration Platform

Tool Support

# OpenROAD-Research: Accelerate Open-source Ecosystem



- Platform for developing and sharing for advanced P&R engines
  - **Open-source physical design for 2D/3D ICs**
  - **GPU-accelerated physical design or distributed computing for physical design**
  - **LLM/ML for physical design**
- Originated from OpenROAD (Developed at UCSD)
- Part of IEEE CEDDA DATC efforts (Please check our talk/paper at October 30)
- Fully open-source and free to use (BSD 3-Clause License)
- Copyrights are preserved through DCO authentication
- Led by Professor Zhiang Wang at Fudan University



Scan me for more  
information of  
OR-Research

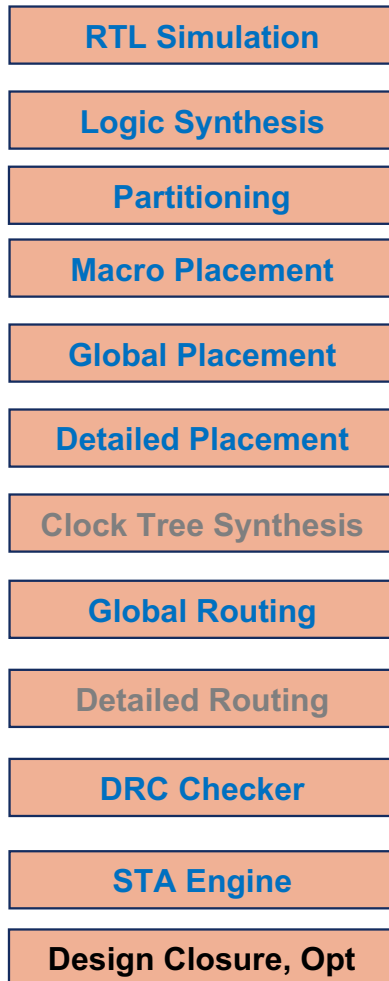


Scan me for more  
information of  
ORFS-Research

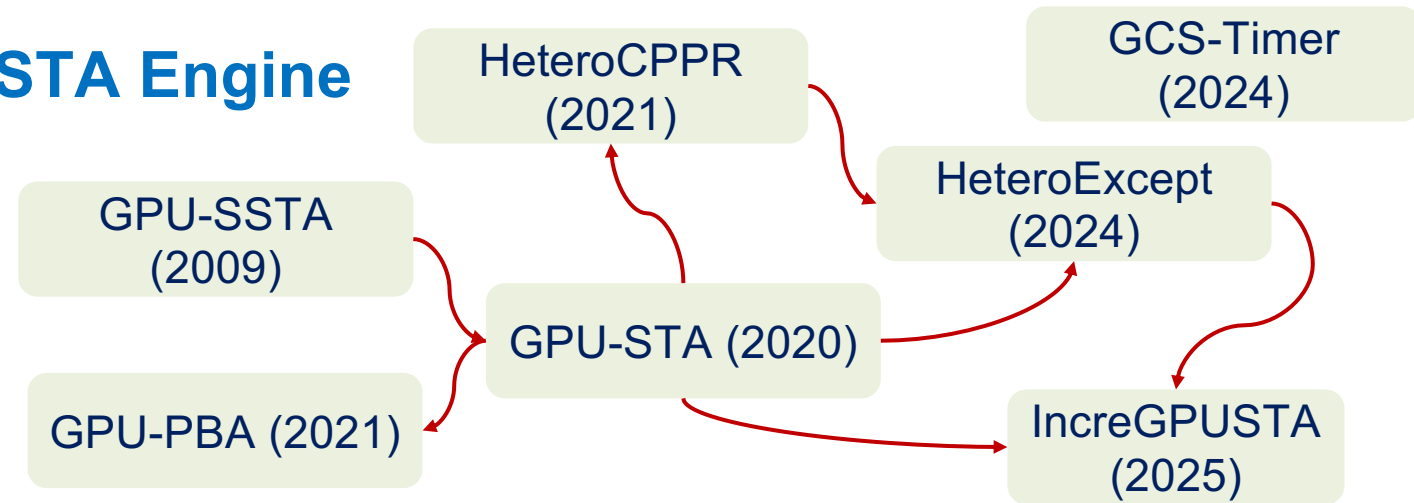
# Call for Participation: ISPD26/27 Contest



## Physical Design Flow



## STA Engine



## ISPD26 Contest: Post-Placement Buffering and Sizing

Organizing Team: UCSD, Fudan University, POSTECH

### Co-Chairs

Dr. Yiting Liu, UCSD ABKGroup [yil375@ucsd.edu]  
Prof. Zhiang Wang, Fudan University [zhiangwang@fudan.edu.cn]

### Table of Contents

- Contest description: [ISPD26\\_contest\\_description.pdf](#)
- Benchmarks: The first set of released benchmarks including [aes\\_cipher\\_top](#), [jpeg\\_encoder](#) and [ariane](#).
- Platform/ASAP7: Technology platform files and libraries for the ASAP7 PDK.
- Evaluation scripts: Evaluation scripts for [aes\\_cipher\\_top](#), [jpeg\\_encoder](#) and [ariane](#).
- Docker containers and submission formatting: Dockerfile and commands required to maintain a consistent evaluation and submission environment can be found at the [README](#).

## Organizing Team

UCSD (USA)  
Fudan University (China)  
POSTECH (Korea)



Join Now !!!



**Q&A**

---

**Thank You !**