GPU Acceleration in OpenROAD: An Update

06.25.2024

Zhiang Wang
UCSD ECE Department

zhw033@ucsd.edu

UCSD

GPU-Accelerated Global Placer: From DREAMPlace

DREAMPlace: Deep Learning Toolkit-Enabled GPU Acceleration for Modern VLSI Placement

Yibo Lin
ECE Department, UT Austin
yibolin@utexas.edu

Shounak Dhar
ECE Department, UT Austin
shounak.dhar@utexas.edu

Wuxi Li
ECE Department, UT Austin
wuxi.li@utexas.edu

Haoxing Ren
Nvidia, Inc., Austin
haoxingr@nvidia.com

Brucek Khailany
Nvidia, Inc., Austin
bkhailany@nvidia.com

David Z. Pan
ECE Department, UT Austin
dpan@ece.utexas.edu

$$\min \quad \underbrace{\left(\sum_{e\in E} WL(e;\mathbf{x},\mathbf{y})\right)}_{\text{Wirelength}} + \lambda \underbrace{D(\mathbf{x},\mathbf{y})}_{\text{Density}}$$

Wirelength OP

Smoothing HPWL by weighted average wirelength

$$\mathrm{WA}_e = \frac{\sum_{i\in e} x_i e^{\frac{x_i}{\gamma}}}{\sum_{i\in e} e^{\frac{x_i}{\gamma}}} - \frac{\sum_{i\in e} x_i e^{-\frac{x_i}{\gamma}}}{\sum_{i\in e} e^{-\frac{x_i}{\gamma}}}$$

Net-level Parallelization

30X speedup compared to RePlAce

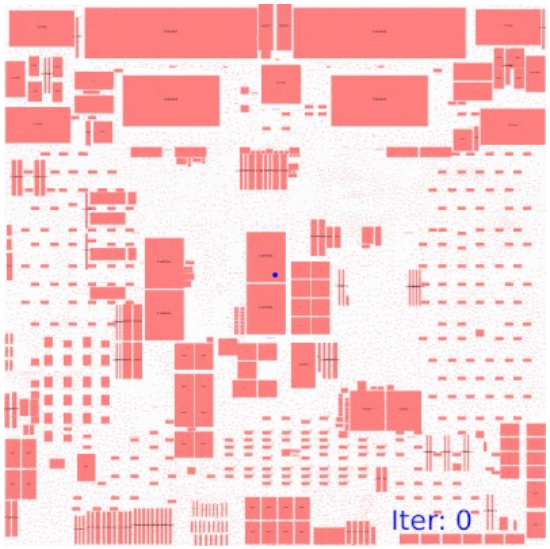
Density Penalty OP

Electrostatic system analogy

Cell instance Electric particle

$$a_{u,v} = \mathrm{DCT}(\mathrm{DCT}(\rho)^T)^T$$
$$\psi_{\mathrm{DCT}} = \mathrm{IDCT}(\mathrm{IDCT}(\{\frac{a_{u,v}}{w_u^2+w_v^2}\})^T)^T$$
$$\xi^X_{\mathrm{DSCT}} = \mathrm{IDXST}(\mathrm{IDCT}(\{\frac{a_{u,v}w_u}{w_u^2+w_v^2}\})^T)^T$$
$$\xi^Y_{\mathrm{DCST}} = \mathrm{IDCT}(\mathrm{IDXST}(\{\frac{a_{u,v}w_v}{w_u^2+w_v^2}\})^T)^T$$

GPU-Accelerated FFT



limbo018 / DREAMPlace

Code Issues 62 Pull requests

DREAMPlace Public

master 19 Branches 19 Tags

[From DREAMPlace repo]

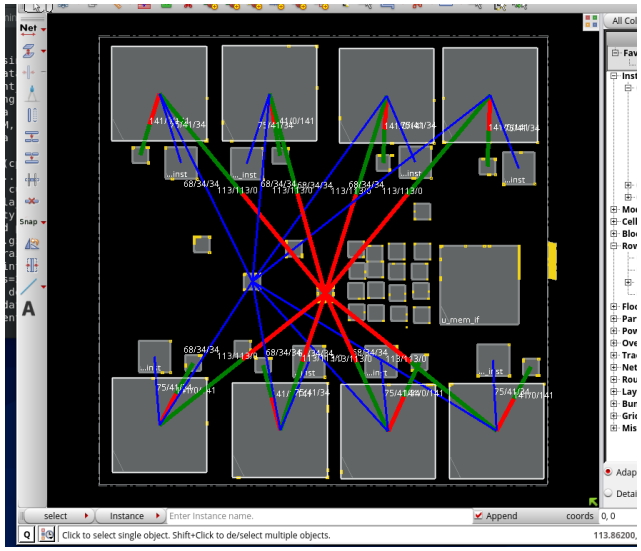
Faster is great but not enough !

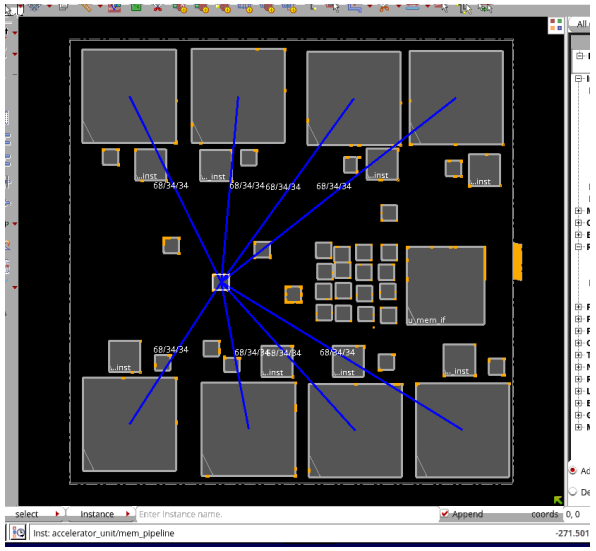
"Faster, Better, Cheaper – pick any two" (it's the law !)

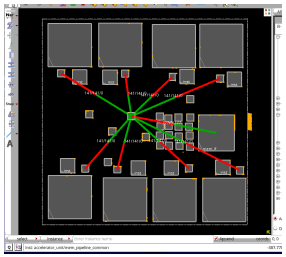
Hints from Macro Placement

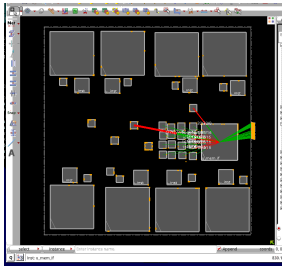
- Human experts usually need to consider multiple factors when they do macro placement
  - Dataflow
  - Connectivity between macros and input-output (IO) pins
  - Critical timing paths
  - ...

We also need to consider these factors when we do placement !
(Especially Mixed-Size Placement !!!)







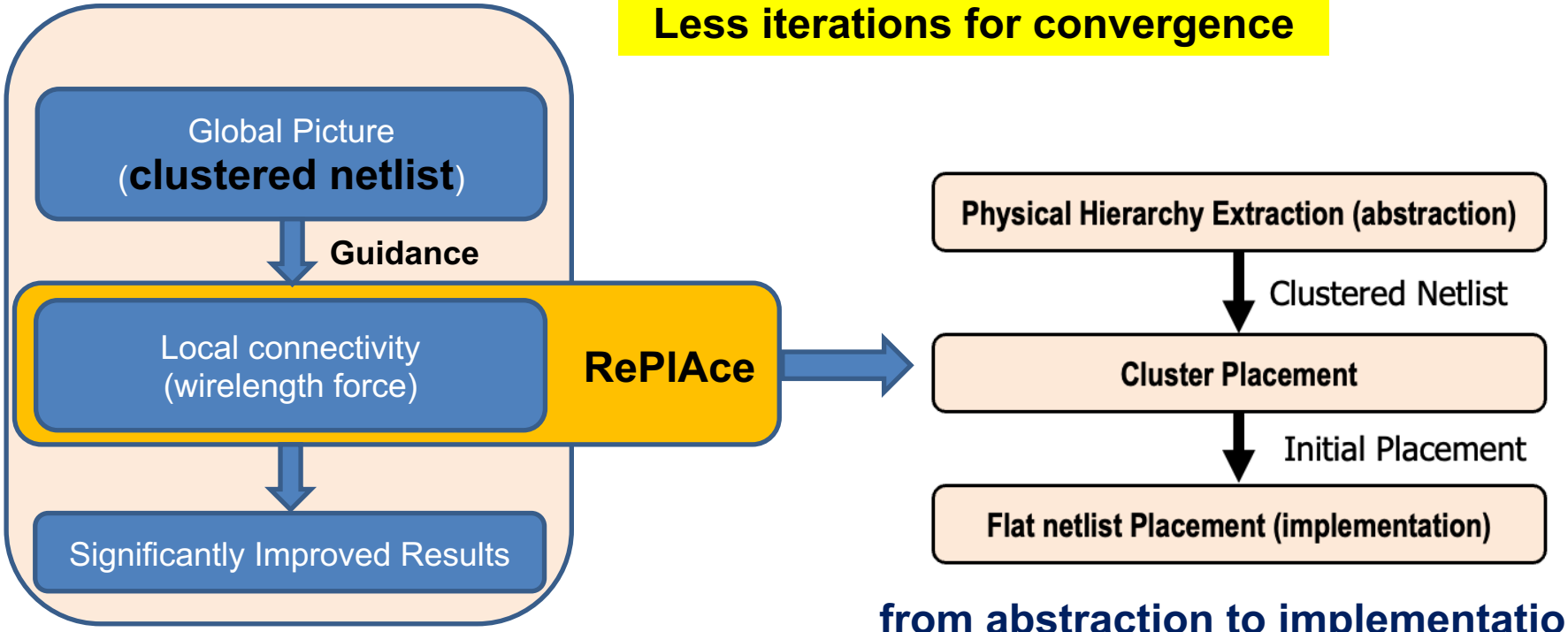


Example dataflow analysis of an AI accelerator

Clustering is another important lever !!!

- **Physical hierarchy aware placement**
  - Convert the logical hierarchy into physical hierarchy through merging and breaking logical modules (TCAD'24)
  - Inform the placer about clusters of standard cells that will stay together during placement **Better Placement**
  - Provide better initial locations for instances through cluster placement

Less iterations for convergence

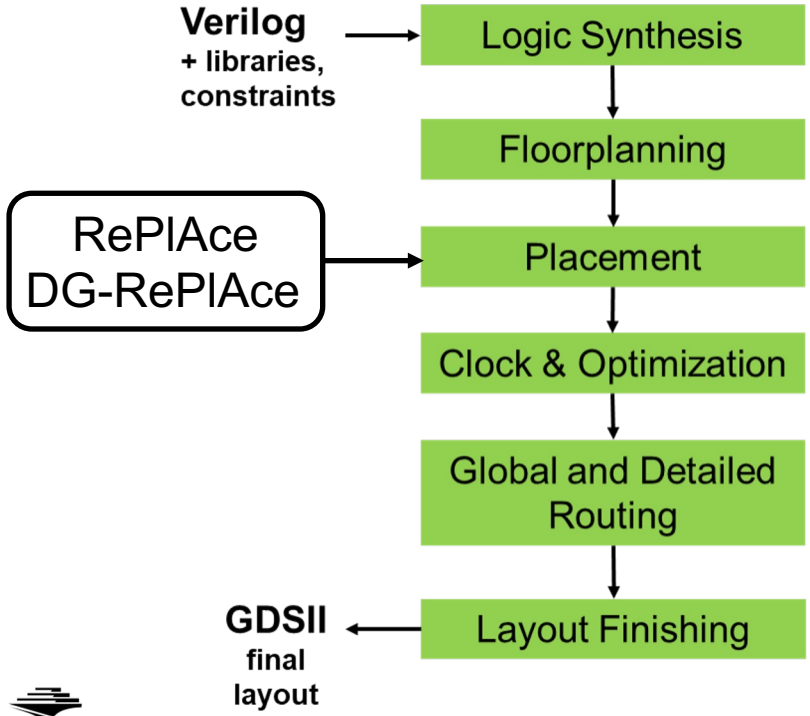


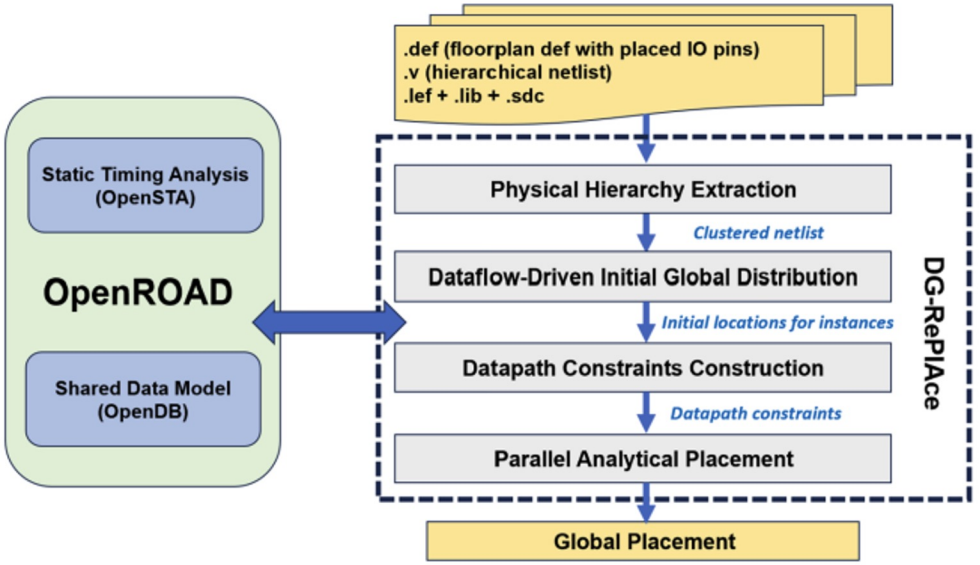
from abstraction to implementation

A. B. Kahng, R. Varadarajan and Z. Wang,
"Hier-RTLMP: A Hierarchical Automatic Macro Placer for Large-scale Complex IP Blocks", IEEE Trans. On CAD, 2024.

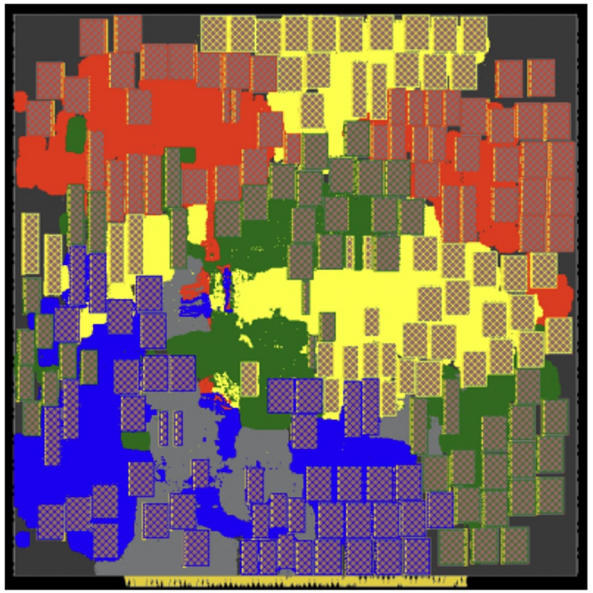
Dataflow-Driven GPU-Accelerated RePlAce

- **Global Placer for large-scale IP blocks (DG-RePlAce)** arXiv
  - **Excellent Scalability:** up to ~10M instances and ~1K macros
  - **Superior Speed**: more than 30X speedup compared to RePlAce
  - **High Quality:** dataflow-driven, physical hierarchy aware placement
  - **Accessibility:** fully open-source, integrated in OpenROAD src/gpl2
  - **Easy-to-use:** OpenROAD flow or plug into commercial productive flow

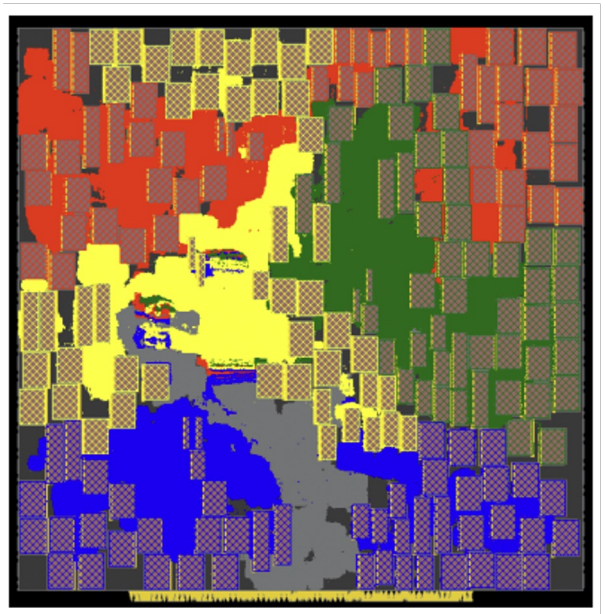




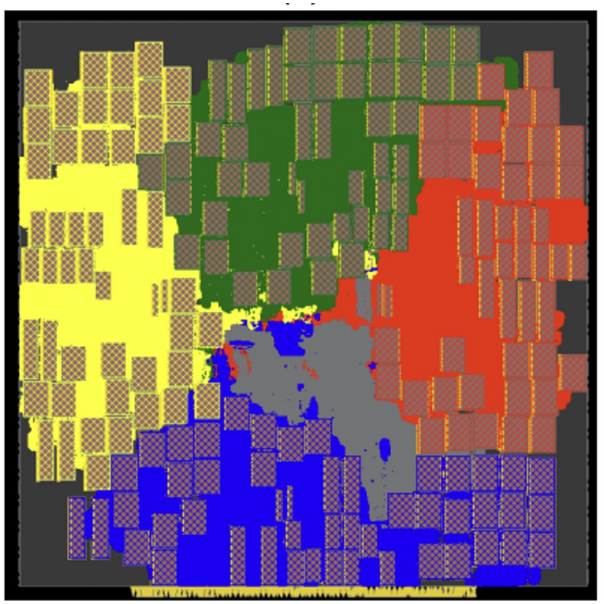
Dataflow-Aware GPU-Accelerated RePlAce arXiv



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

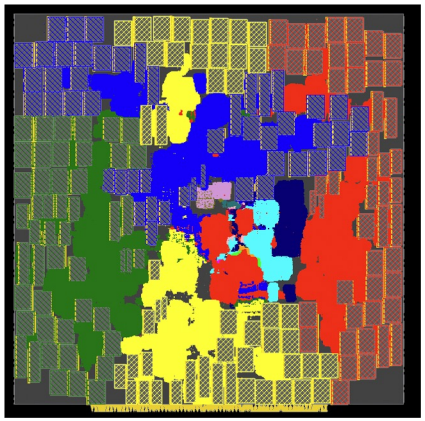
Testcase: BlackParrot RISC-V (Quad-Core) (evaluator: INVS 21.1)
(827K stdcells, 196 macros in GF12LP)

Speed Enables Autotuning (NVIDIA AutoDMP)

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



Demo: swerv_wrapper (NG45)

Post-route layout of RUN_ID = 14

```
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 |
```

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 2: MOTPE Bayesian Opt tuner

Step 3: Run INVS P&R for Pareto candidates

UCSD

GPU-Accelerated Router [Ongoing]

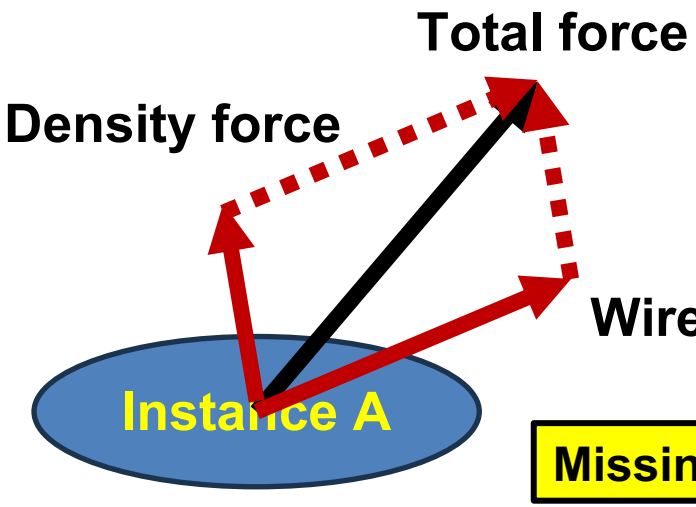
- **Router plays a central role in the physical design flow**
  - A chip will never be taped out if the router cannot achieve DRC-clean routing solutions.
  - Routing is the most timing-consuming step in the physical design flow.
  - GPU-accelerated router is still pending today. [Kahng24ISPD]
  - TritonRoute-WXL (**2021**) is the state-of-the-art academic detail-routability-driven unified global-detailed router. paper src/drt
- **A new GPU-accelerated global router for TritonRoute-WXL is on the way !**
- **Our goals:**
  - **Excellent scalability and superior speed**: handle designs with 50M nets in half an hour
  - **High quality**: achieve better performance as TritonRoute-WXL
  - **Fully open-source**: integrated into OpenROAD
  - **Easy-to-use:** used as real global router in the physical design flow or used as academic eGR for early evaluation (replace RUDY in AutoDMP)

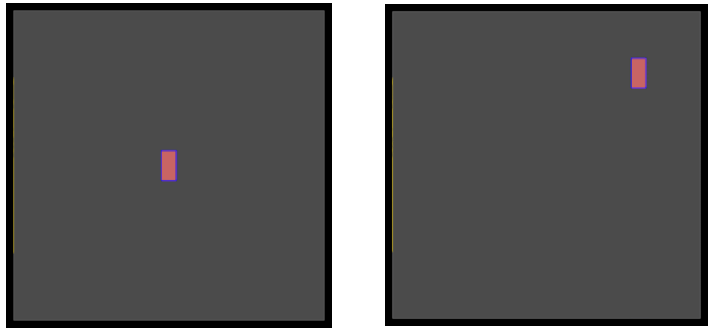
We want a better open-source GPU-accelerated EDA ecosystem !

THANK YOU !

Clustering is another important lever !!!

- **Clustering → better and faster placement (Get Both!)**
  - Better placement: telling the placer how a near-optimal placement looks like
  - Faster placement: **providing a better initial placement,** thereby reducing the number of iterations needed for convergence





Effect of initial placement [from AutoDMP]

Wirelength force (local connectivity)

Missing the global view of the topology of the netlist

Instances that should stay together may be far away from each other

Solution: tell the placer which instances have more chance to stay together