Performance-Driven Large Scale Global Routing Contest

ACM Reference Format:
. 2024. Performance-Driven Large Scale Global Routing Contest. In *Proceedings of ACM Conference (ISPD).* ACM, Austin, Texas, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Global routing is a critical component of the VLSI design process, exerting a substantial influence on circuit timing, power consumption, and overall routability. The efficiency of global routing is of paramount importance, as a swift and scalable approach can guide optimizations in early design stages like floor-planning and placement.

To encourage academic research in addressing the scalability challenges of global routing algorithms using GPU and machine learning (ML) technologies, Nvidia hosted the ISPD2024 contest on GPU/ML-enhanced large-scale global routing [3]. This contest introduced a set of large-scale benchmarks, encompassing up to 50 million cells. These benchmarks reflected industrial-level congestion, presenting challenging routing scenarios. The contest featured simplified input/output formats and evaluation metrics, framing the global routing challenges as mathematical optimization problems.

While the simplified input/output formats and evaluation metrics enhance the accessibility of the competition to participants from diverse backgrounds, they can introduce inaccuracies in performance modeling. The input files used in the ISPD2024 contest lack timing and power information. Metrics based solely on wirelength and routing overflow fail to accurately model timing performance and power consumption. For instance, minimizing total wirelength does not necessarily reduce delays on timing-critical paths. Wires on different metal layers exhibit varying resistance, resulting in different delays and power consumption. Additionally, the impact of vias on delays is difficult to model with simple metrics. Inter-wire coupling capacitance can also cause significant discrepancies between actual and nominal timing responses and power consumption.

In light of the above, we will host another GPU/ML-enhanced large-scale global routing to consider more accurate performance modeling, bringing the contest problem one step closer to the real-world routing challenges. Additionally, we encourage participants to release their code under a permissive open-source license. The main additions in this year's contest are as follows:

(1) For each testcase, this contest provides two sets of input files: a) industry-standard LEF, DEF, LIB, and SDC files, and b) simplified rerouting resource and net information files. The industry-standard files serve as the raw input, enabling contestants to perform the most accurate modeling of routing resources and performance. The simplified input files offer abstracted routing resource and net connection information, along with pre-routing timing estimates, allowing contestants to quickly engage with the contest.

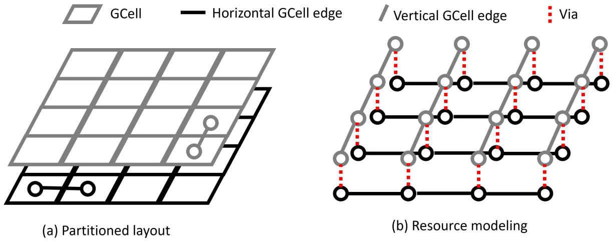
(2) This contest establishes a global routing performance evaluation flow utilizing OpenROAD [1], a leading open-source chip design toolset. The evaluation flow involves loading a global routing solution into OpenROAD, which then estimates parasitics, timing, power, and routing congestion for the global routing solutions.

(3) This contest encourages participants to release their code under a permissive open-source license, promoting contributions to the open-source community.

2 PROBLEM FORMULATION

In global routing, a 3D routing space is defined using global routing cells (GCells), created by a regular grid of horizontal and vertical lines, as illustrated in Figure 1. This configuration results in the formation of a grid graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where each GCell is treated as a vertex ($v \in \mathcal{V}$) and edges ($e \in \mathcal{E}$) connect adjacent GCells within the same layer (GCell edges) or between GCells in neighboring layers (via edges). The global router needs to establish a concrete path for each net within the grid graph and optimize the routability, timing and power.

For each testcase, the global router starts with a placed design, and generates a global routing solution. The global routing solution is evaluated by OpenROAD, which reports timing, power, and routing congestion. Additionally, the runtime and memory efficiency of the global router are critical factors.



(a) Partitioned layout

(b) Resource modeling

Figure 1: Illustration of a GCell grid graph.

2.1 Input Format

For each testcase, two sets of input files are provided: industry-standard files and simplified files.

The industry-standard files include DEF, LEF, LIB, and SDC files. The DEF file contains definitions for CORE, ROW, TRACKS, and GCELLGRID, along with placed COMPONENTS and unrouted NETS. Similar to the ICCAD2019 global routing contest [2], GCells are specified using the definition from the DEF GCELLGRID section. The LEF file includes MACRO definitions and technology information. The LIB files offer timing and power data for library cells, while the SDC files provide timing constraints. These files serve as

ISPD, Mar 2025, Austin, Texas, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

the raw input, allowing contestants to perform the most accurate routing resource and performance modeling.

For each circuit, we also provide a set of simplified input files, which include a routing resource file (with a `.cap` extension) and a net information file (with a `.net` extension). The routing resource file follows the same format as used in the ISPD2024 contest, while the net information file is an extended version of the one used in ISPD2024. The routing resource file offers a detailed representation of the GCell grid graph and its available routing resources. The net information file provides the access points for all the pins within each net, along with the pin names and pre-routing stage slack estimates. These slack estimates provide a rough timing view of the circuit and enable contestants to perform net-based timing optimization. For detailed formatting specifications of the routing resource file, please refer to Appendix A. Likewise, for the format specifications of the net information file, please consult Appendix B. The simplified input files enable contestants to quickly engage with the contest and facilitate framing global routing challenges as mathematical optimization problems.

2.2 Output Format

The output file conforms to the route segment file format compatible with the OpenROAD physical design flow. This format is similar to the output file format used in the ISPD2024 contest. The primary difference is that the ISPD2024 contest format describes route segments in the GCell coordinate system, whereas the route segment file format describes the route segments in the original layout coordinate system.

Here is an illustrative example of a global routing solution for a net:

```
# Net name
Net0
(
# $x_l$ $y_l$ $z_1$ $x_h$ $y_h$ $z_2$
270 13230 M4 270 13230 M5
270 13230 M5 270 13230 M6
270 13230 M4 810 13230 M4
810 13230 M4 810 13230 M3
810 13230 M2 1350 13230 M2
1350 13230 M2 1350 13230 M1
)
```

where each row (x_l y_l z_1 x_h y_h z_2) describes a segment spanning from (x_l, y_l, z_1) to (x_h, y_h, z_2). For example, "270 13230 M4 810 13230 M4" represents a horizontal line in the routing layer M4. And "270 13230 M4 270 13230 M5" represents a via going from M4 to M5. Notice that the via segments can go from lower layers to upper layers (like in "270 13230 M4 270 13230 M5") or from upper layers to lower layers (like in "810 13230 M4 810 13230 M3").

To be considered valid, a global routing solution for a net must ensure that its wires cover all pins of the net and that the wires collectively form a connected graph. In this graph representation, each wire corresponds to a vertex. An edge exists between two vertices (wires) if they satisfy one of the following conditions: (i) They touch each other on the same metal layer, or (ii) Vias connect them. The resulting graph must be a connected structure. For an overall global routing solution to be deemed valid, it must satisfy the validity criteria for all nets in the circuit.

3 BENCHMARKS

The organizers will provide a benchmark suite for the contest. This suite will encompass two sets of circuits, one publicly available and another unpublished blind set for final evaluation. Each set will include 10 placed circuits designed using the Nangate45 technology nodes. Notably, the unpublished blind benchmark suite shares the same netlists as the public suite but features distinct placement solutions. The largest circuit within these suites encompasses approximately 50 million cells. It is worth noting that some testcases may contain macros that restrict access to certain routing resources. For the sake of simplicity, these circuits do not incorporate power grid or clock tree routing.

4 EVALUATION AND RANKING

4.1 Evaluation Environment

Submitted global routers will run on a computing platform equipped with one A100 GPU, which boasts a memory capacity of 80GB. The utilization of up to 8 CPU threads will be supported. To facilitate standardized development environment, we will supply a Docker image preconfigured with CUDA and some popular machine learning libraries. We encourage participating teams to capitalize on the potential of GPU acceleration and explore the integration of machine learning techniques. However, the usage of the GPU is optional, and teams are free to choose their preferred approach, whether it involves leveraging the GPU or not.

4.2 Evaluation Metrics

The route segment file will be loaded into the OpenROAD flow to estimate parasitics, followed by conducting static timing analysis and power analysis.

The evaluation of a global routing solution encompasses two key aspects: the quality of the global routing solution and the runtime required to execute the global routing process. The quality of detailed routing result is measured by timing and power metrics. A solution with a smaller scaled score is considered as a better solution in this contest.

$$scaled\_score = original\_score * (1 + runtime\_factor)$$

The original score is measured by the weighted sum of the following metrics: total wire length, via utilization and routing overflow.

$$\begin{aligned} original\_score = {} & w_1 * (WNS - WNS_{ref}) \\ & + w_2 * \frac{TNS - TNS_{ref}}{N_{endpoint}} \\ & + w_3 * \frac{TotalPower - TotalPower_{ref}}{N_{net}} \\ & + w_4 * OverflowScore \end{aligned}$$

where WNS, TNS represent the worst negative slack and the total negative slack, respectively. $TotalPower$ represents total power consumption. $N_{endpoint}$ and N_{net} represent the number of timing path end points and the number of nets, respectively. WNS_{ref}, TNS_{ref}

and $TotalPower_{ref}$ represent the WNS, TNS, and total power values obtained from the global routing solution generated by the OpenROAD global router. $w_{1,2,3,4}$ are manually-defined weights.

The efficiency of global routing is also of great importance. Therefore, the runtime factor applies to the global router.

$$T = 0.02 * log2(\frac{GRouter\_Wall\_Time}{Median\_Wall\_Time})$$

$$runtime\_factor = min(0.2, max(-0.2, T))$$

The median wall time is the median runtime of all submitted global routers from contestants for the benchmark. The runtime penalty/benefit is limited within -0.2/0.2.

The scaled score is considered as infinity if one of the following condition happens:

1) The proposed router has segmentation fault/crash;

2) The proposal router changes placement, netlist or any design information in the solution file;

3) The runtime usage of the proposed router is over the given runtime limitation. The details about the runtime will be released later;

4) The memory usage of the proposed router is over 100GB in CPU or over the GPU memory limitation;

5) The routing solution file generated by the proposed router is invalid;

6) There are unconnected nets.

4.3 Ranking

Our ranking method aligns with the ISPD-2018/2019 contests. The ranking process follows these steps:

1) Rank each team for each benchmark. The team with a smaller scaled score will get a smaller ranking number, which means a better ranking;

2) Prune out the worst (i.e., biggest) ranking number, and then average the remaining rankings for each team. The team with the smallest averaged ranking number wins the contest;

3) If tight, the averaged ranking including the worst one will be considered.

5 USEFUL RESOURCES

The following resource might be useful to contestants:

1) FastRoute4.1: https://openroad.readthedocs.io/en/latest/main/src/grt/README.html

2) NCTU-GR2.0: https://people.cs.nctu.edu.tw/~whliu/NCTU-GR.htm

3) SPRoute2.0: https://github.com/asyncvlsi/SPRoute

4) CUGR: https://github.com/cuhk-eda/cu-gr

5) CUGR2: https://github.com/cuhk-eda/cu-gr-2

6) GGR: Superfast Full-Scale GPU-Accelerated Global Routing: https://github.com/cuhk-eda/Xplace/tree/main/cpp_to_py/gpugr

7) RouteNet for congestion estimation: https://github.com/circuitnet/CircuitNet/blob/main/models/routenet.py

8) InstantGR: https://github.com/cuhk-eda/InstantGR

REFERENCES

[1] Tutu Ajayi, Vidya A Chhabria, Mateus Fogaça, Soheil Hashemi, Abdelrahman Hosny, Andrew B Kahng, Minsoo Kim, Jeongsup Lee, Uday Mallappa, and Marina Neseem. 2019. Toward an open-source digital flow: First learnings from the openroad project. In *Proceedings of Design Automation Conference (DAC)*. 1–4.

[2] Sergei Dolgov, Alexander Volkov, Lutong Wang, and Bangqi Xu. 2019. 2019 cad contest: Lef/def based global routing. In *Proceedings of International Conference on Computer-Aided Design (ICCAD)*. IEEE, 1–4.

[3] Rongjian Liang, Anthony Agnesina, Wen-Hao Liu, and Haoxing Ren. 2024. GPU/ML-Enhanced Large Scale Global Routing Contest. In *Proceedings of International Symposium on Physical Design (ISPD)*. 269–274.

A FORMAT OF .CAP FILE

The routing resource file follows this format:

```
# Dimensions of GCell graph
nLayers xSize ySize
# Weights of performance metrics
UnitLengthWireCost UnitViaCost OFWeight[0] OFWeight[1] OFWeight[2] ···
# Lengths of horizontal GCell edges (edge count = xSize - 1)
HorizontalGCellEdgeLengths[0] HorizontalGCellEdgeLengths[1] HorizontalGCellEdgeLengths[2] ···
# Lengths of vertical GCell edges (edge count = ySize - 1)
VerticalGCellEdgeLengths[0] VerticalGCellEdgeLengths[1] VerticalGCellEdgeLengths[2] ···
# Information for the 0-th layer
## Layer name, prefered direction and minimum length of a wire at this metal layer. For direction, 0 represents horizontal, while 1 represents vertical.
layerName layerDirection layerMinLength
## Routing capacities of GCell edges at the 0-th layer
### Capacities of GCell at [x(0), y(0)], Capacities of GCell at [x(1), y(0)], ...
10 10 10 ···
### Capacities of GCell at [x(0), y(1)], Capacities of GCell at [x(1), y(1)], ...
10 10 10 ···
·········
## Information for the 1-th layer
·········
```

B FORMAT OF .NET FILE

The net information file follows this format:

```
# Net name
Net0
(
# Pin name, slack estimate, access point locations (layer, x, y) for pin 0. Selecting any one of these locations is sufficient for pin 0.
pin0, -2, [(location of access point 0), (location of access point 1), ··· ]
# Pin name, slack estimate, access point locations for pin 1
pin1, -1, [(location of access point 0), (location of access point 1), ··· ]
···
)
Net1
(
```

pin2, 2, [(location of access point 0), (location of access point 1), · · ·]

· · ·

)

pin3, 13, [(location of access point 0), (location of access point 1), · · ·]

· · · · · ·