ISPD 2024 Contest
GPU/ML-Enhanced Large Scale Global Routing

Rongjian Liang
NVIDIA
Austin, US
rliang@nvidia.com

Anthony Agnesina
NVIDIA
Austin, US
aagnesina@nvidia.com

Haoxing Ren
NVIDIA
Austin, US
haoxingr@nvidia.com

I. 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.

Over the past decade, GPU accelerated computing platforms have been evolving into highly versatile and programmable systems capable of delivering immense parallel computing power. Recent studies have successfully leveraged GPUs to achieve over a 10x acceleration in global routing without compromising performance [1]. Furthermore, machine learning (ML) techniques have been integrated into the global routing process, leading to enhanced routing solution quality [2].

The goal of this competition is to stimulate academic research aimed at developing a GPU/ML-enhanced global router tailored for industrial-level circuits. Notably, contemporary VLSI circuits often encompass tens of millions of cells, which is a significant departure from past global routing competitions that typically dealt with scenarios involving no more than 1 million cells. Due to the limitations of current routers, hierarchical or partitioning-based methods are commonly employed to manage large circuits, albeit at the risk of sacrificing a certain degree of optimality. It is of great importance to develop a scalable global router capable of handling circuits with tens of millions of cells, as it can greatly inform optimizations in the early design stages, such as floor-planning and placement. By fostering enthusiasm and innovation within the global routing research community, this competition aims to deliver substantial reductions in global routing runtime for these expansive industrial-grade circuits, harnessing the computational power of GPUs and the potential of machine learning techniques. Simultaneously, it strives to enhance the overall quality of routing results.

II. Input/Output Format

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 (a). 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), as depicted in Figure 1 (b). It's important to note that each layer has a preferred routing direction, which means GCell edges can be either horizontal or vertical. The essence of the global routing challenge is to establish concrete path for each net within the grid graph. This process ensures the interconnection of all pins without overflow while minimizing total wire length and the number of vias.

In each circuit, there are two essential input files: a routing resource file (with a `.cap` extension) and a net information file (with a `.net` extension). They contain all the necessary input information for global routing. Just for reference, we also release the LEF/DEF files of the circuits. The routing resource file offers a detailed representation of the GCell grid graph and the routing resources it encompasses. Meanwhile, the net information file shows the access points for all the pins within each net. For detailed formatting specifications of the routing resource file, please refer to "resource_file_format.txt." Likewise, for the format specifications of the net information file, consult "net_format.txt". Furthermore, we offer parsing scripts for the routing resource file and the net information file within the "evaluator.cpp" file.

The global routing solution is described in the GCell coordinate system. And the routing solution is defined on metal (routing) layers, from which via utilization can be inferred. To enhance routability and ensure

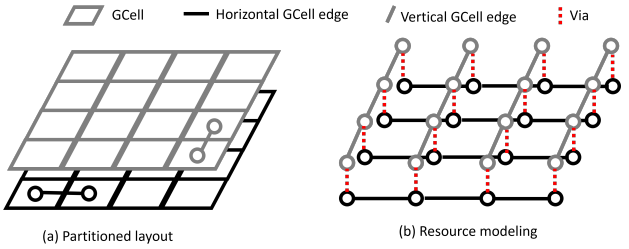

Fig. 1: Illustration of a GCell grid graph.

pin accessibility during the subsequent detailed routing process, we operate under the following assumptions:

- Metal 1 is not employed for net routing. To reach pins on Metal 1, vias must be utilized to establish connections from Metal 2.
- Regarding a via that links Metal layer i and Metal layer j, it is assumed that the via will either obstruct the utilization of specific routing resources on the associated GCell edge or result in increased overhead in routing resource usage. Detailed information on how vias impact routing resource usage can be found in the ”evaluator.cpp” file.

Here is an illustrative example of a global routing solution for a net (as depicted in Figure 2 (b)):

```
Net0
{
        0 0 1 1 metal1
        0 0 1 3 metal2
        0 2 4 3 metal3
        3 2 4 4 metal2
        3 3 4 4 metal1
}
```

In the above example, five wires are defined for Net0, each covers one or multiple contiguous GCells. And ”0 2 4 3 Metal3” represents a wire cover GCells (0,2), (1,2), (2,2) and (3,2) on metal 3. The total wire length of this routing solution is calculated by summing up the wire length on all metal layers:

WL_M1 = 0,
WL_M2 = 175 + 100 = 275,
WL_M3 = 550,
Total_WL = WL_M1 + WL_M2 + WL_M3 = 825.

The wire length on Metal 1 is always zero as Metal 1 is not utilized for net routing. The wire length on Metal 2 is determined by the total length of vertical GCell edges that are utilized on Metal 2. Likewise, the wire length on Metal 3 is equal to the total length of horizontal GCell edges that are employed on Metal 3. This routing solution necessitates the use of four vias, comprising two vias transitioning from metal 1 to metal 2, and an additional two vias from metal 2 to metal 3. It is important to mentioned that vias will lead to increased in routing resource usage though they do not contribute to the wire length. Comprehensive details on the computation of wire length and via count can be located in the “evaluator.cpp” file.

To be considered valid, a global routing solution for a net must ensure that all pins of the net are covered by its wires, and 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 either of the following conditions: (i) they touch each other on the same metal layer, or (ii) they reside on neighboring metal layers and have a non-zero overlapping area. 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. In addition, Metal 1 should not employed for net routing.

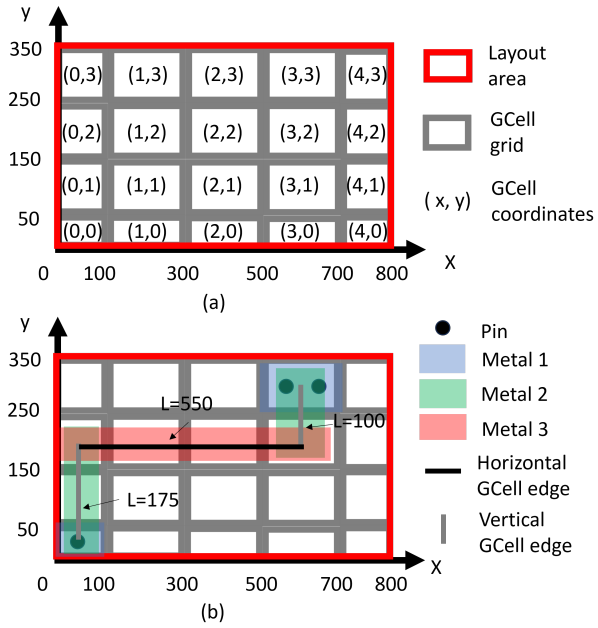

Fig. 2: Example of GCell definition and global routing solution. (a) shows an example of GCell definition; (b) depicts a global routing solution (Adapted from [3]).

III. 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 for 45nm and 28nm 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.

IV. Evaluation and Ranking

A. Evaluation Environment

Submitted global routers will run on a computing platform equipped with four 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.

B. Evaluation Metrics

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. Aligned with the ISPD-2018/2019 and the ICCAD-2019 contests, the quality of detailed routing result is measured by the following equation. A solution with a smaller scaled score is considered as a better solution in this contest.

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

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

$$origina\_socre = w_1 * TotalWL + w_2 * ViaCount + w_3 * OverflowScore$$

The computation of the routing overflow will consider factors such as pin accessibility and the impact of nets with multiple pins located within the same GCell. Comprehensive details regarding the original scoring methodology can be found in "evaluation.cpp".

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 an infinity number 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 are invalid;

6) There is unconnected net.

For simplicity, we intend to skip the nondeterministic penalty used in previous contests. We will run multiple times of the proposed router, and pick the median scaled score as the final score for a benchmark.

C. 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.

V. Useful Resource

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

REFERENCES

[1] S. Lin and M. D. Wong, "Superfast full-scale cpu-accelerated global routing," in *Proceedings of International Conference on Computer-Aided Design (ICCAD)*, 2022, pp. 1–8.

[2] Z. Zhou, Y. Pan, G. G. Lemieux, and A. Ivanov, "MEDUSA: A Multi-resolution Machine Learning Congestion Estimation Method for 2D and 3D Global Routing," *ACM Transactions on Design Automation of Electronic Systems*, 2023.

[3] S. Dolgov, A. Volkov, L. Wang, and B. Xu, "2019 cad contest: Lef/def based global routing," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2019, pp. 1–4.