2021 ICCAD CAD Contest Problem B: Routing with Cell Movement Advanced

Invited Paper

Kai-Shun Hu, Tao-Chun Yu, Ming-Jen Yang, and Chin-Fang Cindy Shen

Synopsys, Inc.

29F, No. 333, Section 1, Keelung Road, Taipei, Taiwan

{kaihu, taochun, mingjen, cfshen}@synopsys.com

Abstract

2021 ICCAD CAD Contest Problem B is an extended problem from 2020 ICCAD CAD Contest Problem B [1-2] for addressing more complex constraints. In the physical implementation, the common approach is to divide the problem into the placement and routing stage. By doing this divide-and-conquer approach, it may cause conservative margin reservation and miscorrelation.

In order to achieve multiple advanced objectives in terms of Power, timing Performance and Area, so called PPA, a certain amount of cell movement at the routing stage become a desired functionality in an EDA tool.

2021 ICCAD CAD Contest Problem B encourages the research in the techniques of routing with cell movement to achieve multiple objectives in the advanced process nodes (less than 7 nm).

We provided (*i*) a set of benchmarks and (*ii*) an evaluation metric of multiple objectives including power factor, the criticality of timing critical nets, the maximum number of moving cells, and the total routing length optimization that facilitate contestants to develop and test their new algorithms.

Keywords

Physical design; routing; placement

1 Introduction

Splitting the physical implementation problem to 2 sub-problems, placement and Routing, is a common approach to solve the modern complex physical implementation problem [3-5]. Inside these 2 major steps, the common approach is to further divide the placement & routing problem into more sub-problems. The purpose of this divide-and-conquer approach is to make sure this complex problem would be manageable and be converged in reasonable runtime.

With more and more sub-problems be defined, the correlation and the objective of each sub-problem might not be aligned with each other. Also, in certain sub-problem, it may preserve a margin on purpose for helping the convergence of upcoming steps. This kind of margins and/or misalignments would become a weakness point in this divide-and-conquer approach.

For example, in the placement stage, a placer commonly constrains the placement result by cell density or pin density. It is a very conservative way to preserve more margins for helping the convergence in routing stage. But cell density or pin density may not be well correlated to the real routing problem.

In the routing stage, there are more and more reasonings to cause the final routing DRC not converged without further cell movement in the advanced process nodes design. If the placement and routing are with independent engines, the long turn-around time of the placement & routing iterations could not be endured anymore.

Thus, in 2021 ICCAD CAD Contest Problem B [6], we try to explore the approach of having a routing engine which can also do cell movement. So, the weakness of the preserved margins and mis-alignments issues can be eliminated.

1.1 Contest Objective

The objective is to develop a global route routing engine to honor all the given routing constraints. And this global route routing engine can move the cells from one gGrid to another if all the given routing constraints can still be satisfied while the metric of routing objectives can be further reduced through the cell movements.

To simplify the problem, an initial routing result is given in the input file. Contestants can start from the given initial routing data and then do the incremental routing for re-connecting the broken parts caused by cell movements. Contestants can also discard all the given initial routing data and re-generate all the new routings while considering cell movements by their own.

We introduced a layer-based power factor for routing and the instance-cell-based voltage area constraints for placement in this contest. These two objectives are commonly used in the P&R EDA tools to do the power management and optimization.

From the timing perspective, we also added two objectives, net-based minimum routing layer and net-based weight for the net criticality on timing critical paths. These two objectives are widely used in the P&R EDA tools to do the timing optimization.

In the contest, the total number of moved cells among all cells are constrained. Having this cell movement constraint is because that we do not want the contestants to perform a whole new placement first to generate a totally different placement result. It is not the purpose of this contest. We also would like to avoid a big cell displacement with totally different timing result.

2 Problem Description

The input of this problem is a placed and routed design. The goal is to further minimize the routing metric of the design with given constraints.

2.1 Input Data

In the contest, we limit the number of cell movements. So, the maximum number of moved cells is given. However, the move distance of each moved cell is not limited.

Syntax

```
MaxCellMove <maxMoveCount>
```

Example

```
MaxCellMove 2
```

Figure 1: Syntax of max cell move and example

In the Fig. 1 example, two cell movements are allowed at most.

The number of rows & columns of the gGrids are given. gGrid is the minimum resolution of this problem. Cell instances and routes are all with the resolution of gGrid. All the routing layers have the same number of rows and columns.

Syntax

```
GGridBoundaryIdx <rowBeginIdx> <colBeginIdx> <rowEndIdx> <colEndIdx>
```

Example

```
GGridBoundaryIdx 1 1 5 5
```

Figure 2: Syntax of gGrid and example

In the Fig. 2 example, it means the design has 5 gGrid rows (from 1 to 5) and 5 gGrid columns (from 1 to 5).

The total number of routing layers, the routing direction of each layer, the default supply value of the gGrid on each layer, and the power factor of each layer are given. The routing direction will always be horizontal on M1 and routing direction of two adjacent layers are always different. In the contest, all H/V-direction routings must satisfy layer routing direction. For the simplification, vertical interconnect access (VIA) is simply modeled as z-direction routing. Its gGrid length and demand modeling are same as the routings in H/V direction.

The R/C characteristic is different on each layer. So, the routings on different layer would result to different power dissipation. We would like to model a layer-based factor `<powerFactor>` to represent the power dissipation of the routings on a layer.

Syntax

```
NumLayer <LayerCount>
Lay <layerName> <Idx> <RoutingDir> <defaultSupplyOfOneGGrid> <powerFactor>
```

Example

```
NumLayer 3
Lay M1 1 H 10 1.2
Lay M2 2 V 8 1.0
Lay M3 3 H 8 0.8
```

Figure 3: Syntax of Layer and example

In the Fig. 3 example, there are 3 routing layers in the design. The routing direction of M1 and M3 are horizontal and the routing direction of M2 is vertical. The default supply of one gGrid on M1 is 10. The default supply of one gGrid on M2 and M3 is 8. The power factor on M1, M2, and M3 are 1.2, 1.0, and 0.8, respectively.

There are gGrids with non-default supply. The increment or decrement value of the supply in certain gGrids is also given.

Syntax

```
NumNonDefaultSupplyGGrid <nonDefaultSupplyGGridCount>
<rowIdx> <colIdx> <LayIdx> <incrOrDecrValue>
```

Example

```
NumNonDefaultSupplyGGrid 2
2 2 1 +3
1 2 3 -2
```

Figure 4: Syntax of Non-default supply and example

In the Fig. 4 example, the supply of the (2,2,1) gGrid is the default supply of layer M1 plus 3. The supply of the (1,2,3) gGrid is the default supply of layer M3 minus 2.

The detailed information of the master cells which are used in the design is given. It includes the master cell name, number of pins in the master cell, the pin name & pin layer of each pin, number of blockages in the master cell, and the blockage name, blockage layer, & blockage demand of each blockage.

Syntax

```
NumMasterCell <masterCellCount>
MasterCell <masterCellName> <pinCount> <blockageCount>
Pin <pinName> <pinLayer>
Blkg <blockageName> <blockageLayer> <demand>
```

Example

```
NumMasterCell 2
MasterCell MC1 1 1
Pin P1 M1
Blkg B1 M1 2
MasterCell MC2 2 0
Pin P1 M2
Pin P2 M1
```

Figure 5: Syntax of master cell and example

In the Fig. 5 example, there are 2 master cells MC1 and MC2. MC1 has 1 pin P1 on M1 and 1 blockage B1 on M1. Blockage B1 would need 2 demands. While, MC2 has 2 pins, P1 is on M2 and P2 is on M1. There is no blockage in MC2.

The placed cell instances and the netlist are given. And the given nets may have minimum routing layer constraint. If a net does not have the minimum routing layer constraint, `<minRoutingLayConstraint>` is represent with a keyword `NoCstr`. There are cell instances which are not movable. This movable or fixed information will be described in `<movableCstr>` for every cell instance. If the cell instance is movable, `<movableCstr>` represents with a keyword `Movable`. If the cell instance is not movable, `<movableCstr>` represents with a keyword `Fixed`.

In the consideration of timing, we would like to model a net-based weight `<weight>` to represent the net criticality in terms of timing for the nets on critical paths. So, for the nets not on timing critical paths, the weight value of the nets will be 1.0. For the nets on timing critical paths, the weight value of these nets will be greater than 1.0. With this net-based weighted model, we would like to encourage the contestants to reduce the routing length for the nets on timing critical paths.

Syntax

```
NumCellInst <cellInstCount>
CellInst <instName> <masterCellName> <gGridRowIdx> <gGridColIdx> <movableCstr>
NumNets <netCount>
Net <netName> <numPins> <minRoutingLayConstraint> <weight>
Pin <instName>/<masterPinName>
```

Example

```
NumCellInst 2
CellInst C1 MC1 4 1 Fixed
CellInst C2 MC3 5 2 Movable
NumNets 1
Net N1 2 M2 1.5
Pin C1/P2
Pin C2/P1
```

Figure 6: Syntax of cell instance and example

In the Fig. 6 example, there are 2 cell instances, C1 and C2, which are placed in (4,1) and (5,2), respectively. There is 1 net, N1. Net N1 has 2 pins which are the P2 pin of instance C1 and P1 pin of instance C2. Furthermore, net N1 has minimum routing layer constraint that no horizontal and vertical routing is allowed below M2. And net N1 is on timing critical paths and its weight value is 1.5.

There are voltage area constraints. It has voltage area label for certain gGrid row & column pairs. Also, it has corresponding voltage area label for certain cell instances as well. For the cell instances having voltage area label, these cell instances can only be placed (moved) to the gGrid row & column pairs which has the same voltage area label. For the cell instances without voltage area label, all the gGrid row & column pairs in the given design can be the candidate for cell location.

Syntax

```
NumVoltageAreas <voltageAreaCount>
Name <voltageArea Name>
GGrids <gGridCount>
<gGridRowIdx> <gGridColIdx>
Instances <instanceCount>
<InstanceName>
```

Example

```
NumVoltageAreas 2
Name V1
GGrids 3
1 1
2 1
1 2
Instances 2
C1
C2
Name V2
GGrids 2
5 5
4 5
Instances 2
C5
C7
```

Figure 7: Syntax of voltage area constraints and example

In the Fig. 7 example, there are 2 voltage areas, V1 and V2. gGrid (row=1, col=1), (row=2, col=1), and (row=1, col=2) are in the voltage area V1. Cell instance C1 and C2 are associated to voltage area V1 as well. So, cell instance C1 and C2 can only be placed (moved) in these 3 gGrids. gGrid (row=5, col=5) and (row=4, col=5) are in voltage area V2. Cell instances C5 and C7 can only be placed (moved) inside voltage area V2.

An initial routing data is also given as the input. The provided initial routing data guarantees to be (*i*) all nets routed without net open and (*ii*) no overflow gGrid happened.

Syntax

```
NumRoutes <routeSegmentCount>
<sRowIdx> <sColIdx> <sLayIdx> <eRowIdx> <eColIdx> <eLayIdx> <netName>
```

Example

```
NumRoutes 2
4 1 1 4 1 3 N1
4 1 3 4 4 3 N1
```

Figure 8: Syntax of routes and example

In the Fig. 8 example, there are 2 routing segments for net N1. The 1st routing segment is a z-direction segment from layer 1 to layer 3 at (row 4, col 1). The 2nd routing segment is a horizontal segment from (row 4, col 1) to (row 4, col 4) on layer 3.

2.2 Supply and Demand

As described in **2.1**, the supply of each gGrid is given. The supply of a gGrid would be the default supply of its corresponding layer with an increment or a decrement of its non-default supply.

The demand of a gGrid is the summation of {the number of nets which has routing segment in this gGrid and all the blockage demands for the cell instances in this gGrid}. For one net, no matter how much routing segments crossing one gGrid, it only consumes one demand in that gGrid.

If the demand of a gGrid is larger than its supply, the gGrid is called an overflow gGrid. In the contest, any overflow gGrid is not allowed.

2.3 Connectivity Model

We can imagine a gGrid as a 3D block. A routing segment is considered as in the center line of the gGrid block. Two routing segments are connected if and only if they touch each other.

For each net, all pins are required to be connected by the routing segments. Floating or dangling routing segments are allowed but they are redundant and consumes routing lengths.

Since this is a simplified global routing problem, contestants do not need to worry about any net short.

2.4 Output Data

Following information is needed in the output file. (*i*) The moved cells instances and its new locations. (*ii*) The complete routing information of the result.

The syntax of moved cell instances result is shown in Fig. 9.

Syntax

```
NumMovedCellInst <movedCellInstCount>
CellInst <instName> <newRowIdx> <newColIdx>
```

Example

```
NumMovedCellInst 2
CellInst C1 3 3
CellInst C2 4 2
```

Figure 9: Syntax of the result of moved cells and example

No matter the given initial routing data is re-used or not, the complete routing information needs to be written out in the output file. The syntax of output routing data is the same as the input routing data described in section 2.1.

2.5 Multi-threading

It is encouraged to use all the techniques which can speed up the turn-around time for this place & route problem. Thus, multi-threading is allowed and encouraged in this contest as well. The maximum number of cores that can be used is 8-cores in the 2021 ICCAD CAD Contest Problem B.

3 Evaluation Methodology

All the given constraints need to be satisfied. (*i*) No overflow gGrid allowed. (*ii*) No open net allowed. All pins of a net must be in the same connected component by using output routes. (*iii*) Voltage area constraints must be satisfied. (*iv*) Max cell movement constraint must be satisfied. (*v*) Runtime limit is 1 hour for each case in the evaluation machine. The hidden cases would be in the same scale as public cases. If the program and the output result violate any of the above constraints, the score is 0 for this test case.

In addition, (*vi*) net minimum layer constraint and routing direction must be satisfied. The minimum layer constraint limits the H-direction and V-direction routing must be on or above the given minimum layer for the constrained net. The routing direction limits the H-direction routing must be on the odd layers and the V-direction routing must be on the even layers. The violated routing segment(s) of the (*vi*) constraint would be discarded during the evaluation. If this discarded segment(s) removal causes net open, the score is 0 for this test case as well.

Evaluation score =

$$\sum_{i=1}^{\#nets}\left(\sum_{j=1}^{\#layers}\left(net_i\ \text{gGrid length on}\ layer_j\right)\times \text{PowerFactor}(layer_j)\right)\times \text{Weight}(net_i)$$

The ranking of this contest is based on the summation of the score of each case including both the public and hidden cases.

3.1 Evaluator

We provided a public evaluator [6] in 2021 ICCAD CAD Contest Problem B. The provided public evaluator would help contestants quickly get the result of evaluation score by providing the input data file and the contestants generated output file. Also, the evaluator would indicate all the invalid scenarios in the giving output file to help contestants check what is the issue in the contestants generated output file.

4 Benchmarks

The benchmarks provided in 2021 ICCAD CAD Contest Problem B were generated by an in-house P&R design generator. The given cell placement and the given initial routing in both the public and the hidden cases were generated by an in-house placement and global route routing tool. This in-house placement and global route routing tool does consider the routing congestion, voltage area constraints, and the total routing length in both the placement and the routing stage.

The 2021 ICCAD CAD Contest Problem B benchmarks are available on the contest website [6] for download.

5 Conclusion

The 2021 ICCAD CAD Contest Problem B aims to encourage the research in the techniques of resolving the conservative margin reservation and the mis-correlation problem in the divide-and-conquer place & route approach. The evaluation metric and the benchmarks were provided to objectively quantify the performance of the extra gain of the routing length reduction by doing the cell movements from contestants. We look forward to receiving innovative algorithms from contestants in this 2021 ICCAD CAD Contest.

References

[1] 2020 ICCAD CAD Contest Problem B: http://iccad-contest.org/2020/problems.html

[2] K. -S. Hu, M. -J. Yang, T. -C. Yu and G. -C. Chen, "ICCAD-2020 CAD contest in Routing with Cell Movement: Invited Talk," *2020 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, 2020, pp. 1-4.

[3] X. He, T. Huang, L. Xiao, H. Tian, G. Cui and E. F. Y. Young, " Ripple: An effective routability-driven placer by iterative cell movement," *2011 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, San Jose, CA, 2011, pp. 74-79.

[4] Y. Xu, Y. Zhang and C. Chu, "FastRoute 4.0: Global router with efficient via minimization," *2009 Asia and South Pacific Design Automation Conference (ASP-DAC)*, Yokohama, 2009, pp. 576-581.

[5] W. Liu, W. Kao, Y. Li and K. Chao, "NCTU-GR 2.0: Multithreaded Collision-Aware Global Routing with Bounded-Length Maze Routing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 32, no. 5, pp. 709-722, May 2013.

[6] 2021 ICCAD CAD Contest Problem B: http://iccad-contest.org/2021/problems.html