TRADER: A Practical Track-Assignment-Based Detailed Router

Zhen Zhuang[*†], Genggeng Liu[*], Tsung-Yi Ho[†], Bei Yu[†], and Wenzhong Guo[*]
[*] Fuzhou University
[†] The Chinese University of Hong Kong

***Abstract*—As the last stage of VLSI routing, detailed routing should consider complicated design rules in order to meet the manufacturability of chips. With the continuous development of VLSI technology node, the design rules are changing and increasing which makes detailed routing a hard task. In this paper, we present a practical track-assignment-based detailed router to deal with the most representative design rules in modern designs. The proposed router consists of four major stages: (1) a graph-based track assignment algorithm is proposed to optimize the design rule violations of an entire die area; (2) an effective rip-up and reroute method is used to reduce the design rule violations in local regions; (3) a segment migration algorithm is proposed to reduce short violations; and (4) a stack via optimization technique is proposed to reduce minimum area violations. Practical benchmarks from 2019 ISPD contest are used to evaluate the proposed router. Compared with the state-of-the-art detailed router, Dr. CU 2.0, the number of violations can be reduced by up to 35.11% with an average reduction rate of 10.08%. The area of short can be reduced by up to 61.49% with an average reduction rate of 44.80%.**

I. Introduction

Shrinking feature sizes for very large scale integrated circuits (VLSI) make routing become more challenging [1]. In addition to the higher pin density and smaller pin geometry, complex design rules become a critical bottleneck [2]. In order to increase yield and avoid manufacture problems, the number of design rules provided by foundries extremely increases and is still rising as technology node advances [3]. Due to the complexity of design rules, routing is usually divided into two stages, global routing and detailed routing. In the global routing stage, the local regions, which are the routing guides for detailed routing, are selected based on coarse grid graphs [4]–[6]. Unlike global routing, exact interconnecting wires should be generated considering all design rules in the detailed routing stage. Detailed routing becomes one of the most complicated and time-consuming stages [7], [8].

During the past ten years, many detailed routers were proposed to deal with the impact of various manufacturing technologies, including self-aligned double patterning [9] and triple patterning [10]–[12]. However, many practical design rules released by industries, including parallel-run spacing, minimum area, and end-of-line spacing, are not considered to optimize the holistic performance of chips. Recently, ISPD holds two detailed routing contests in 2018 and 2019 [7], [8], which introduce several detailed routers considering practical design rules based on the circuits released by Cadence. Chen *et al*. [13], [14] and Li *et al*. [3] propose three maze-routing-based detailed routers. Kahng *et al*. [15] propose an ILP-based detailed router. Sun *et al*. [16] propose a track-assignment-based detailed router with the refinement of routing guides. Unlike the above works, we propose a detailed router based on a graph-based track assignment algorithm that can more effectively reduce design rule violations.

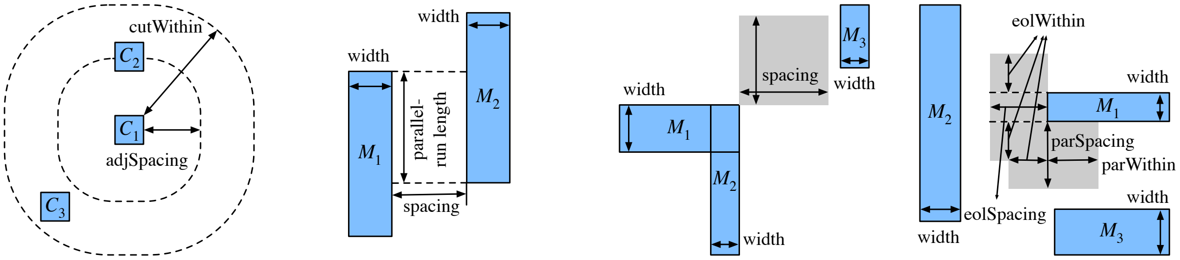
In the first stage of the proposed router, a graph-based track assignment algorithm is proposed to generate initial routing solutions considering practical design rules globally. Different objectives, such as crosstalk and performance, are optimized by previous work [17]. Recently, many routability-driven track assignment algorithms are proposed to bridge the gap between global routing and detailed routing. Wong *et al*. [18] propose a negotiation-based track assignment algorithm considering local nets. Shi *et al*. [19] propose a net-by-net track assignment algorithm considering local nets and the connectivity of nets. Liu *et al*. [20] propose a fast panel-by-panel track assignment algorithm to efficiently analyze the routability of a design. However, many practical design rules such as parallel-run spacing are not considered by the above works. Furthermore, the above algorithms can only finish track assignment tasks based on the exact global routing solutions without extra routing regions. Different from these algorithms, the proposed graph-based track assignment algorithm can effectively deal with practical design rules and flexible routing spaces. Then, a rip-up and reroute method, a segment migration algorithm, and a stack via optimization technique are used to reduce the violations of local regions.

In conclusion, a practical track-assignment-based detailed router, **TRADER**, is proposed to generate high-quality detailed routing solutions overcoming the disadvantages of traditional algorithms. The major contributions are shown below:

- A design-rule-driven track assignment is proposed to obtain high-quality initial solutions. The refinement of routing guides, the selection of via locations, and the pin access are integrated closely into the assignment process, which makes it flexible to obtain fewer violations.
- A segment migration algorithm is proposed to reduce the number of short violations and the area of short violations. This algorithm changes the metal layers of the segments with short violations to effectively utilize available routing resource.
- A stack via optimization is proposed to further reduce the minimum area violation number. This technique, collaborating with the segment migration algorithm, uses

TABLE I The violation weight of each routing metric

Routing Metric	Weight	Routing Metric	Weight	Routing Metric	Weight
Wirelength	0.5	Out-of-track Via Number	1	Parallel-run Spacing Violation Number	500
Via Number	4	Non-preferred Direction Wirelength	1	End-of-line Spacing Violation Number	500
Out-of-guide Wirelength	1	Short Violation Number	500	Cut Spacing Violation Number	500
Out-of-guide Via Number	1	The Area of Short	500	Adjacent Cut Spacing Violation Number	500
Out-of-track Wirelength	0.5	Minimum Area Violation Number	500	Corner-to-corner Spacing Violation Number	500



(a) Adjacent cut spacing (b) Parallel-run spacing (c) Corner-to-corner spacing (d) End-of-line spacing

Fig. 1 The illustration of design rules.

proper patches to reduce minimum area violations.

- TRADER can achieve high-quality detailed routing solutions. Compared with the state-of-the-art detailed router, Dr. CU 2.0 [3], the number of violations can be reduced by up to 35.11% with an average reduction rate of 10.08%. The area of short can be reduced by up to 61.49% with an average reduction rate of 44.80%.

II. Preliminaries

A. Connectivity Rules and Routing Preference Metrics

1) *Open* and *Short*: Open means any pin of a net is not connected. The routing solution with open nets is regarded as an invalid solution. Short is one of the most important metrics. If different metals have overlapped regions, they are regarded as shorted metals.

2) *Routing preference metrics*: Routing within guides, routing following the preferred direction and routing following tracks are the preference metrics.

B. Design Rules

The most representative design rules which should be handled by detailed routing are as follows [8].

1) *Cut spacing* (in Fig. 1(a)): For a cut, if the number of cuts in the `cutWithin` region guarantees the requirement of the adjacent cut spacing rule and there is a cut in the `adjSpacing` region, the adjacent cut spacing violation is registered. Furthermore, the cut spacing rule specifies the minimum spacing between two cuts.

2) *Parallel-run spacing* (in Fig. 1(b)): If the `parallel-run length` between two metals is more than zero, the `spacing` of the two metals should be more than a specified value. The greater the maximum `width` of the two metals, the greater the specified `spacing`. The greater the `parallel-run length`, the greater the specified `spacing`.

3) *Corner-to-corner spacing* (in Fig. 1(c)): For a metal, if there is a metal overlapped with the corner region (the gray region), the corner-to-corner spacing violation is registered. If the `parallel-run length` between two metals is more than zero, this violation is not registered.

4) *End-of-line spacing* (in Fig. 1(d)): For a metal (M_1), when *parallel end-of-line spacing* is defined and there is a metal overlapped with the triggered region of this design rule (the bottom gray region of M_1), the end-of-line spacing violation is registered if there is a metal overlapped with the end-of-line region (the left gray region of M_1). When *parallel end-of-line spacing* is not defined, there is no triggered region which should be considered. The end-of-line spacing violation is registered if there is a metal overlapped with the end-of-line region without considering parallel metals like M_3.

5) *Minimum area*: The minimum area rule specifies the minimum area of each metal.

C. Problem Formulation

The objective of TRADER is to minimize the most representative design rule violations and the connectivity violations which are listed above. Furthermore, TRADER can guarantee that there is no open net in each routing solution. The quality of each stage of TRADER is controlled based on the standard released by the 2019 ISPD contest [8] to balance different routing metrics. Each routing metric is given a weight for every unit as shown in TABLE I.

III. Detailed Design Flow

TRADER can be divided into two major stages as shown in Fig. 2. The first stage is track assignment. In this stage, a design-rule-driven track assignment algorithm, **DTA**, is proposed to obtain initial routing solutions. The second stage includes the rip-up and reroute step, the short optimization step, and the stack via optimization step.

A. Design-Rule-Driven Track Assignment

The input of track assignment is usually the global routing solution which is the set of net topologies. However, in order

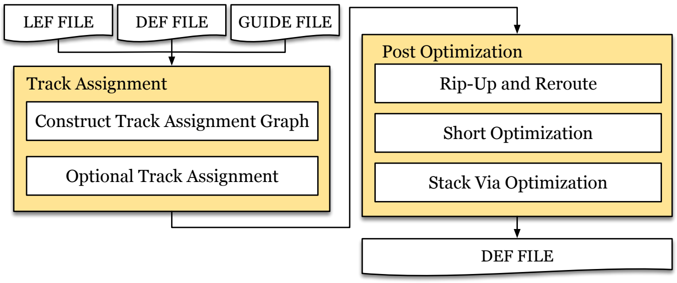
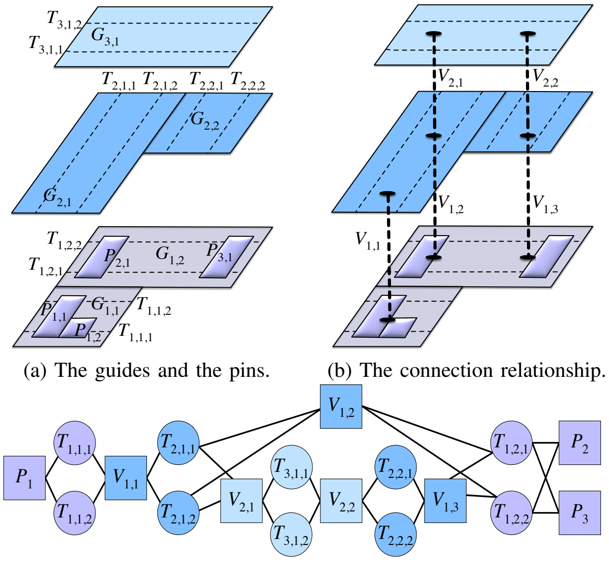


Fig. 2 The detailed design flow of TRADER.



(a) The guides and the pins.

(b) The connection relationship.

(c) The track assignment graph.

Fig. 3 The construction of track assignment graph.

to provide the proper guidance for detailed routing, the input of TRADER is flexible routing guides which are rectangles. Therefore, the topologies should be extracted from routing guides for traditional track assignment algorithms, like the work [16]. Since routing guides usually form a complex graph rather than a simple tree, the extraction of topologies which might remove many routing guides makes that the routing space is limited. As a result, an effective construction method for track assignment graphs is proposed to fully use the space of routing guides.

In order to construct an effective track assignment graph, the connection relationship of routing guides should be determined to guarantee the connectivity of nets at first. For two routing guides on the adjacent metal layers, if there is an intersecting region on their 2D projection plane, they are connected together by a via to determine the connection relationship with the consideration of routing metrics. Only when routing guides cannot be organized into a connected graph, the routing guides can be connected by non-preferred direction edges. Fig. 3(a) shows the guides and the pins of a net. $G_{i,j}$ represents the j-th routing guide on the i-th metal layer. $P_{i,j}$ represents the j-th rectangle of the i-th pin. The black dashed lines represent tracks. Fig. 3(b) shows the connection relationship of the net. $V_{i,j}$ represents the j-th via on the i-th cut layer. As $G_{1,1}$ and $G_{2,1}$, they are on the adjacent metal layers and there is an intersecting region on their 2D projection plane. Therefore, $G_{1,1}$ and $G_{2,1}$ are connected together by $V_{1,1}$.

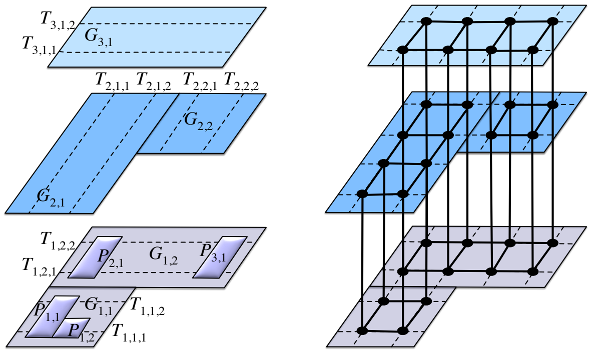
When the connection relationship of routing guides is determined, the track assignment graph can be constructed according to the relationship. There are three types of nodes, including track nodes, pin super nodes, and via super nodes. All nodes are connected together based on the connection relationship to construct the track assignment graph as shown in Fig. 3(c). P_i, $T_{i,j,k}$, and $V_{i,j}$ represent a pin super node, a track node, and a via super node, respectively. Since the track assignment graph in Fig. 3(c) is constructed based on the connection relationship of Fig. 3(b), P_i, $T_{i,j,k}$, and $V_{i,j}$ correspond to the i-th pin, the k-th track in $G_{i,j}$, and $V_{i,j}$ in Fig. 3(b), respectively.

Traditional track assignment algorithms aim to assign the segments extracted from global routing solutions to proper tracks. In this work, pin access and the selection of via locations are integrated into the assignment process by the searching of track assignment graphs to fully use the routing space. Therefore, good track assignment solutions in which the pins of each net are connected together can be obtained based on the construction of track assignment graphs. The location of each via super node is not unique when the segments connected by the via are assigned to different tracks. It is assumed that SG_1 and SG_2 are two routing guides and they are connected by a via super node SV_1. Let the number of tracks in SG_i is donated by sgn_i. If sgn_1 and sgn_2 are n and m, respectively, the number of possible via locations is $n \times m$. However, the via location is determined by the related tracks. As shown in Fig. 3(c), $V_{1,1}$ has 2×2 possible locations, but the location of $V_{1,1}$ can be determined if it is connected by the segments on $T_{1,1,1}$ and $T_{2,1,1}$. Therefore, all possible locations of $V_{i,j}$ are abstracted to a super node. Similarly, P_i is connected when the path to P_i is found although there are many access points of P_i. Therefore, all access points are abstracted to a super node. The access points are extracted from the intersecting locations of tracks to make it available for pins to be connected. For each pin, the intersecting points of the tracks of the adjacent metal layers in and around the pin with no violation are extracted as access points. It makes the number of access points is moderate.

Based on track assignment graphs, high-quality track assignment solutions can be obtained by effective maze routing algorithms. In this work, the design-rule-aware maze routing algorithm proposed by the work [14] is used to obtain track assignment solutions. In particular, any maze routing algorithms can be used in the proposed framework.

B. Post Optimization

Although high-quality initial solutions can be obtained by DTA, it is not easy for track assignment to reduce the design



(a) The guides and the pins. (b) The routing graph.

Fig. 4 The routing graph of maze routing.

Algorithm 1: Segment Migration Algorithm

```
   Input: net set $N$, current routing solution $RS$
   Output: updated routing solution $RS$
1  $SP \leftarrow \emptyset$;
2  for each $n \in N$ do
3      for each $s \in S_n$ do
4          for each $s^{'} \in RS$ do
5              if $s$ and $s^{'}$ have short violations then
6                  $SP \leftarrow (s, s^{'})$;
7  for each $sp \in SP$ do
8      $mc \leftarrow \infty$;
9      Get $ss$ in $sbb$;
10     for each $m_1 \in M$ do
11         for each $m_2 \in M$ do
12             Calculate $ec$ of $p^{'}$ according to $ss$;
13             if $ec < mc$ then
14                 Record $p^{'}$ and update $mc$;
15     Update $RS$;
```

rule violations in the local regions. Therefore, an iterative rip-up and reroute method proposed by the work [14] is used to reduce the regional violations at the first step of the post optimization stage.

For each iteration, the nets with violations are selected and the routing guides of each illegal net are widen one width in four directions to expand the search space. Then, the design-rule-aware maze routing algorithm is used to find better routing solutions based on the expanded routing guides. The intersecting locations of the tracks on the adjacent metal layers are extracted as the nodes of the routing graph for maze routing. Then, the nodes are connected together by vias and segments to generate routing graphs. Fig. 4 shows the routing graph of a net. $P_{i,j}$ and $G_{i,j}$ represent the j-th rectangle of the i-th pin and the j-th routing guide on the i-th metal layer, respectively. Based on routing graphs, violations can be effectively reduced by maze routing.

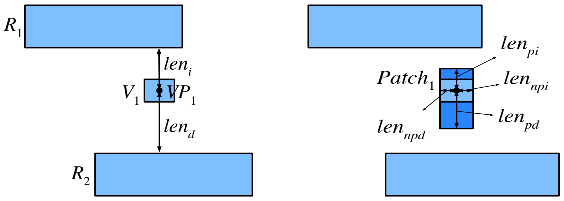
When there are no remaining illegal nets that can be optimized, the rip-up and reroute step is stopped. In order to balance the quality of optimization and the runtime, the expansion of routing guides is limited. After the rip-up and reroute step, many design rule violations, especially short violations, cannot be reduced within routing guides. Since short is a very important routing metric, a segment migration algorithm is proposed to reduce short violations in the short optimization step. At the region of short caused by two overlapped segments, they are migrated to different metal layers outside routing guides to reduce the number of short and the area of short. The pseudo code of the proposed segment migration algorithm is shown in Algorithm 1.

The input is the set of nets N and the current routing solution RS. The output is the updated routing solution RS. The rectangles of metals are organized to a balanced binary search tree (BBST) in RS. The shorted segment pairs are processed one by one in Algorithm 1. SP represents the set of the shorted segment pairs and it is initialized to be an empty set at first (line 1). Then, shorted segment pairs are found (lines 2–6). n represents a net in N. S_n represents the segment set of n. s represents a segment in S_n. For each segment s, the segments adjacent to s are found based on the BBST (lines 4–6). For each adjacent segment $s^{'}$, if s and $s^{'}$ have short violations, the shorted segment pair $(s, s^{'})$ is added to SP (line 6). Finally, each shorted segment pair is optimized (lines 7–15). The minimum cost mc of the best migration scheme is initialized to infinity before the selection of migration schemes (line 8). Based on the maximum spacing value which makes that causing violations is impossible, the bounding box sbb of sp is calculated. Then, the rectangles in all metal layers overlapped with sbb are found and added to ss by searching the BBST (line 9). Next, all combinations of segment migration are considered to find the best combination (lines 10–14). M represents the metal layer set. m_1 and m_2 represent the migrated metal layers of s and $s^{'}$, respectively. It is assumed that the number of metal layers is l, and thus the number of combinations is $l \times l$. Since the number of metal layers is not large, the consideration of all combinations can be accepted. The extra cost ec of each combination $p^{'}$ is calculated (line 12). If ec is less than mc, $p^{'}$ is recorded and mc is updated to be ec (line 14). Finally, the old migration scheme is removed and the new migration scheme is added to RS (line 15). The calculation of ec is shown below:

$$ec = \sum_{r \in R} W_r \times V_r, \tag{1}$$

where r and R represent a design rule and the set of design rules without the minimum area rule, respectively. W_r and V_r represent the weight of r and the violation value of r, respectively.

Since the routing space is usually big enough to make the segment migration successful, short violations can be effectively reduced by the proposed algorithm. However, if the metal layer number of a migrated segment varies greatly, the stack via which has many connected vias at the same location is generated. The via types with smaller enclosures are usually selected to avoid the increase of design rule violations. The enclosure area of the selected vias is usually less than the



(a) The stack via location (b) The optimized solution

Fig. 5 The minimum area optimization.

minimum legal area of the layer. In order to overcome the possible degeneration of the segment migration algorithm, a patch-adding method is proposed to reduce minimum area violations in the stack via optimization step.

In the stack via optimization step, each stack via is added patches. The size of a patch is calculated based on the preferred routing direction. Fig. 5(a) shows the patch-adding of a stack via. V_i represents the i-th stack via. R_i represents the i-th rectangle around V_1. VP_i represents the location of V_i. It is assumed that the preferred routing direction is vertical. As shown in Fig. 5(a), len_i represents the distance between V_1 and its nearest rectangle R_1 at the direction with the increase of coordinates. len_d represents the distance between V_1 and its nearest rectangle R_2 at the direction with the decrease of coordinates. The location of the patch is the same as that of the stack via. The size of the patch is calculated as below:

$$len_{npd} = len_{npi} = width_D \ / \ 2, \tag{2}$$

$$len_{pi} = \frac{len_i \times minarea}{width_D \times (len_i + len_d)}, \tag{3}$$

$$len_{pd} = \frac{len_d \times minarea}{width_D \times (len_i + len_d)}, \tag{4}$$

where len_{npi}, len_{npd}, len_{pi}, and len_{pd} represent the length at the non-preferred direction with the increase of coordinates, the non-preferred direction with the decrease of coordinates, the preferred direction with the increase of coordinates, and the preferred direction with the decrease of coordinates, respectively. $width_D$ represents the default width of the metal layer. $minarea$ represents the minimum legal area of the metal layer. The patch $Patch_1$ is shown in Fig. 5(b).

IV. Experimental Results

TRADER is implemented in C++ language on a Linux server with 64GB memory. The program used to evaluate routing solutions and the benchmarks are released by the 2019 ISPD contest [8]. The score and the violation values of each benchmark can be obtained by the evaluation program based on the Innovus of Cadence [21]. In our implementation, the version of Innovus is v18.11-s100_1.

A. The effectiveness of two optimization techniques

In order to show the effectiveness of the technique of short optimization, the data before and after the execution of this stage is listed in TABLE II. The number of short violations can be reduced by up to 96.45% with an average reduction rate of 87.36%. The area of short can be reduced by up to 98.47% with an average reduction rate of 88.22%. Since short is one of the most serious circuit problems, short optimization can effectively reduce the number of short violations and the area of short to improve the performance of the circuit. Due to stack via optimization, the number of minimum area violations can be reduced by up to 71.97% with an average reduction rate of 44.51%. Since the pages are limited, the data of stack via optimization is not listed here.

TABLE II The effectiveness of short optimization

Testcase	Before Short Optimization		After Short Optimization	
	#Short	ShortArea	#Short	ShortArea
1	535	590.88	23	10.66
2	12770	11814.58	453	212.18
3	800	659.51	34	10.07
4	21371	19316.92	1260	1298.30
5	1615	1486.42	1215	1271.12
6	31424	37679.22	1281	665.74
7	15932	7349.04	825	347.16
8	22149	14263.61	1510	479.35
9	36271	24071.28	3279	1238.58
10	35988	18940.69	2895	1027.66
Ratio	17.53	33.80	1.00	1.00

*#Short and ShortArea represent the number of short violations and the area of short, respectively.

B. Comparison between TRADER and Dr. CU 2.0

In order to show the performance of TRADER, we compare it with Dr. CU 2.0 [3] as listed in TABLE III. #Vio includes the number of short violations and the number of DRC violations. NP-WL includes non-preferred direction wirelength, out-of-track wirelength, and out-of-guide wirelength. #NP-Via includes the number of out-of-track vias and out-of-guide vias. The results of Dr. CU 2.0 are obtained from our server based on the open-source code [22]. Compared with Dr. CU 2.0, #Vio can be reduced by up to 35.11% with an average reduction rate of 10.08%. ShortArea can be reduced by up to 61.49% with an average reduction rate of 44.80%. For other routing metrics, the performances of TRADER and Dr. CU 2.0 have their merits and demerits.

C. Comparison between TRADER and ISPD Contest Winners

In this section, we compare many advanced detailed routers, including the top-2 winners of the 2019 ISPD contest and Dr. CU 2.0, to show the effectiveness of TRADER. In order to consider multiple routing metrics synthetically, we use the evaluation standard released by the 2019 ISPD contest to evaluate routing solutions. The scores of the routing solutions of each detailed router are shown in TABLE IV. The data of the top-2 winners is released by the 2019 ISPD contest. TRADER has a better ability to optimize practical routing metrics compared with the advanced detailed routers.

V. Conclusion

In this paper, a practical track-assignment-based detailed router, TRADER, is proposed to generate high-quality detailed routing solutions. Based on the proposed graph-based track assignment, the refinement of routing guides, the selection of via locations, and the pin access are integrated

TABLE III The comparison between TRADER and Dr. CU 2.0

Testcase	Dr. CU 2.0 [3]						
	#Vio	ShortArea	WL ($\times 10^5$)	NP-WL ($\times 10^5$)	#Via ($\times 10^5$)	#NP-Via ($\times 10^5$)	Runtime (min)
1	178	19.82	6.43	0.26	0.37	0.03	2.97
2	10381	547.08	249.61	6.28	8.11	0.58	34.34
3	589	22.87	8.42	0.30	0.66	0.03	1.22
4	2612	1783.13	304.91	7.06	10.31	0.45	36.15
5	1395	1279.76	47.80	0.31	1.54	0.03	3.32
6	8385	1558.64	660.67	12.43	19.98	0.79	60.86
7	32019	899.00	1225.58	15.35	48.34	1.26	162.60
8	20132	1070.81	1884.73	20.28	73.65	1.99	232.59
9	36596	1933.66	2853.91	32.21	122.49	3.37	313.82
10	36794	1984.04	2821.78	33.61	125.45	3.16	335.98
Ratio	1.13	1.97	1.00	1.01	0.98	0.89	0.90

Testcase	TRADER						
	#Vio	ShortArea	WL ($\times 10^5$)	NP-WL ($\times 10^5$)	#Via ($\times 10^5$)	#NP-Via ($\times 10^5$)	Runtime (min)
1	196	10.66	6.44	0.27	0.37	0.03	3.10
2	9501	214.36	249.68	6.52	8.21	0.61	36.27
3	531	10.07	8.43	0.27	0.67	0.03	1.35
4	1695	1300.00	305.35	6.43	10.73	0.50	36.68
5	1355	1271.70	47.86	0.24	1.60	0.04	3.56
6	6225	676.45	660.93	12.58	20.25	0.89	68.60
7	30558	346.24	1226.11	15.47	49.04	1.47	187.21
8	18382	479.35	1884.96	22.42	74.31	2.32	279.88
9	34038	1239.94	2854.26	36.84	123.63	3.95	367.59
10	33611	1029.56	2822.84	37.66	127.31	3.79	415.31
Ratio	1.00	1.00	1.00	1.00	1.00	1.00	1.00

*Since the Innovus versions of this paper and [3] are different, the data in this paper is different from that of [3].

*#Vio, ShortArea, WL, NP-WL, #Via, and #NP-Via represent the number of violations, the area of short, wirelength, non-preferred wirelength, the number of vias, and the number of non-preferred vias, respectively.

TABLE IV The scores of advanced detailed routers ($\times 10^5$)

Testcase	The 2nd Place	The 1st Place	Dr. CU 2.0	TRADER
1	62.51	6.23	5.95	6.04
2	1651.32	218.99	218.64	213.24
3	149.48	10.05	10.21	9.89
4	3046.01	227.56	223.13	217.44
5	516.92	43.79	43.75	43.75
6	4234.18	682.14	473.09	459.27
7	8491.09	991.29	987.13	980.34
8	–	1371.68	1364.90	1357.98
9	24304.11	2166.70	2144.53	2137.45
10	23560.67	2162.65	2142.70	2133.90
Ratio	11.04	1.07	1.01	1.00

*Since the Innovus versions of this paper and [3] are different, the data in this paper is different from that of [3].

into the assignment process of TRADER. Then, rip-up and reroute method, segment migration algorithm, and stack via optimization technique, are used to optimize the design rule violations of local regions. Compared with advanced routers, TRADER can achieve better routing performance.

Acknowledgment

This work was partially supported by the National Natural Science Foundation of China under Grants No. 61877010 and No. 11501114, State Key Laboratory of Computer Architecture (ICT, CAS) under Grant No. CARCHB202014, and Fujian Natural Science Funds under Grant No. 2019J01243. Genggeng Liu is the corresponding author of the article.

References

[1] ITRS Report 2015. http://www.itrs2.net/itrs-reports.html.

[2] D. Park, D. Lee, L. Kang, C. Holtz, S. Gao, B. Lin and C.-K. Cheng, “Grid-Based Framework for Routability Analysis and Diagnosis With Conditional Design Rules,” in *IEEE TCAD*, vol. 39, no. 12, pp. 5097–5110, 2020.

[3] H. Li, G. Chen, B. Jiang, J. Chen and E. F. Y. Young, “Dr. CU 2.0: A scalable detailed routing framework with correct-by-construction design rule satisfaction,” in *Proc. ICCAD*, pp. 1–7, 2019.

[4] Y. Xu, Y. Zhang and C. Chu, “FastRoute 4.0: Global router with efficient via minimization,” in *Proc. ASPDAC*, pp. 576–581, 2009.

[5] J. Liu, C.-W. Pui, F. Wang and E. F. Y. Young, “CUGR: detailed-routability-driven 3D global routing with probabilistic resource model,” in *Proc. DAC*, pp. 1–6, 2020.

[6] W. Liu, W. Kao, Y. Li and K. Chao, “NCTU-GR 2.0: Multithreaded collision-aware global routing with bounded-length maze routing,” in *IEEE TCAD*, vol. 32, no. 5, pp. 709–722, 2013.

[7] S. Mantik, G. Posser, W.-K. Chow, Y. Ding and W.-H. Liu, “ISPD 2018 initial detailed routing contest and benchmarks,” in *Proc. ISPD*, pp. 140—143, 2018.

[8] W.-H. Liu, S. Mantik, W.-K. Chow, Y. Ding, A. Farshidi and G. Posser, “ISPD 2019 initial detailed routing contest and benchmark with advanced routing rules,” in *Proc. ISPD*, pp. 147—151, 2019.

[9] H. Yu and Y. Chang, “DSA-friendly detailed routing considering double patterning and DSA template assignments,” in *Proc. DAC*, pp. 1–6, 2018.

[10] Q. Ma, H. Zhang and M. D. F. Wong, “Triple patterning aware routing and its comparison with double patterning aware routing in 14nm technology,” in *Proc. DAC*, pp. 591–596, 2012.

[11] Y. Lin, B. Yu, D. Z. Pan and Y. Li, “TRIAD: A triple patterning lithography aware detailed router,” in *Proc. ICCAD*, pp. 123–129, 2012.

[12] Z. Liu, C. Liu and E. F. Y. Young, “An effective triple patterning aware grid-based detailed routing approach,” in *Proc. DATE*, pp. 1641–1646, 2015.

[13] G. Chen, C.-W. Pui, H. Li, J. Chen, B. Jiang and E. F. Y. Young, “Detailed routing by sparse grid graph and minimum-area-captured path search,” in *Proc. ASPDAC*, pp. 754—760, 2019.

[14] G. Chen, C. Pui, H. Li and E. F. Y. Young, “Dr. CU: Detailed routing by sparse grid graph and minimum-area-captured path search,” in *IEEE TCAD*, vol. 39, no. 9, pp. 1902–1915, 2020.

[15] A. B. Kahng, L. Wang and B. Xu, “TritonRoute: An initial detailed router for advanced VLSI technologies,” in *Proc. ICCAD*, pp. 1–8, 2018.

[16] F. Sun, H. Chen, C. Chen, C. Hsu and Y. Chang, “A multithreaded initial detailed routing algorithm considering global routing guides,” in *Proc. ICCAD*, pp. 1–7, 2018.

[17] T.-Y. Ho, Y.-W. Chang, S.-J. Chen and D. T. Lee, “A fast crosstalk- and performance-driven multilevel routing system,” in *Proc. ICCAD*, pp. 382–387, 2003.

[18] M. Wong, W. Liu and T. Wang, “Negotiation-based track assignment considering local nets,” in *Proc. ASPDAC*, pp. 378–383, 2016.

[19] D. Shi and A. Davoodi, “TraPL: Track planning of local congestion for global routing,” in *Proc. DAC*, pp. 1–6, 2017.

[20] G. Liu, Z. Zhuang, W. Guo and T.-C. Wang, “RDTA: An efficient routability-driven track assignment algorithm,” in *Proc. GLSVLSI*, pp. 315—318, 2019.

[21] Cadence Innovus User Guide, http://www.cadence.com.

[22] Dr. CU 2.0. https://github.com/cuhk-eda/dr-cu.