Dr. CU 2.0: A Scalable Detailed Routing Framework with Correct-by-Construction Design Rule Satisfaction

Haocheng Li , Gengjie Chen , Bentian Jiang , Jingsong Chen , and Evangeline F. Y. Young
Department of Computer Science and Engineering, The Chinese University of Hong Kong
{hcli, gjchen, btjiang, jschen, fyyoung}@cse.cuhk.edu.hk

***Abstract*—Detailed routing becomes a crucial challenge in VLSI design with shrinking feature size and increasing design complexity. More complicated design rules were added to guarantee manufacturability, which made detailed routing an even harder task to achieve in the design flow. In this paper, we propose a detailed router that judiciously handles hard-to-access pins and new design rules including length-dependent parallel run length spacing, end-of-line spacing with parallel edges, and corner-to-corner spacing. Our experimental results show that our framework can effectively reduce the number of violations with comparable wirelength. Comparing our algorithm with the best score of each released designs in the ISPD'19 Contest, there is 2% score improvement. Compared with the state-of-the-art work [1], our algorithm achieves 69% better scores. The source code of Dr. CU 2.0 is available at https://github.com/cuhk-eda/dr-cu.**

***Index Terms*—detailed routing, design rule, pin access, VLSI routing, physical design**

I. Introduction

To handle complicated design rules and large solution space, routing is typically divided into global routing and detailed routing. Global routing partitions the whole routing region into global cells to estimate congestion and timing [2]. The detailed routing phase, which generates exact rectilinear wiring interconnects satisfying all design rules, can easily take days of turnaround time without success guaranty [3]. Unlike global routing, detailed routing needs to handle a large number of design rules on a huge and detailed routing grid graph. As feature size scales down, it becomes even more complex and has been a hard problem to resolve in the VLSI design flow, not only because of increased size and scale, but also due to the introduction of tedious design rules. To increase yield and to avoid manufacturing-unfriendly patterns, the number of manufacturability-driven rules provided by foundries increases to hundreds and thousands and is still rising as technology node advances [4]. On the other hand, the height of standard cells today can go down to five tracks with one fin to reduce area and save energy [5]. Consequently, pin access to standard cells [6] and lower layer routing [7] become more difficult and restrictive than ever due to increased pin density, limited access points and interference between pins.

Rule-based routing frameworks are friendly to conventional design flow and have been used for decades. Nieberg computed possible pin access path candidates and selected the shortest paths from pins to grid points not violating any design rules to perform correct pin access [8]. However, his approach cannot handle the case where there is no violation-free path from a pin to any grid point. RegularRoute applied regular routing patterns in a bottom-up layer-by-layer framework and formulated the global segment assignment problem inside each group of routing tracks as a maximum weighted independent set problem [9]. Xu *et al.* proposed a pin access-driven rip-up and reroute scheme under self-aligned double patterning constraints [10]. Li *et al.* presented a pin access-aware legalization algorithm for mixed-cell-height circuits [11]. There are some recent works due to ISPD'18 detailed routing contest [12]. Kahng *et al.* partitioned each layer into parallel panels and formulated the routing problem on each panel as an integer linear program [13]. Sun *et al.* modified the definition of hit points [10] and conduct track assignment considering via violations [14]. Dr. CU proposed an optimal path searching algorithm handling the minimum-area constraint [1]. The recent ISPD 2019 detailed routing contest introduced more design rules including length-dependent parallel run length spacing, EOL spacing with parallel edges and corner-to-corner spacing [15]. Most traditional detailed routers relies on post-processing to fix design rule violations. As design rules get tremendously more complex and abundant, post-processing fails more frequently.

In this paper, we propose a detailed routing algorithm handling pin access and design rules in advanced technology nodes. Our contributions can be summarized as follows.

- We design a pre-processing stage to compute the valid access points of each pin and create off-track vias if no same-layer access point is valid.
- We propose a design rule-aware maze routing to handle end-of-line spacing with parallel edges in a correct-by-construction manner and fix corner-to-corner spacing violations in post-processing.
- We develop a lookup-table-based via insertion method and select violation-free via types from the cell library.
- Our proposed method outperforms the state-of-the-art

The work described in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK14208917).

978-1-7281-2350-9/19/$31.00 ©2019 IEEE

works including the champion of the latest detailed routing contest.

The rest of the paper is organized as follows. Section II outlines the basic terminologies and new design rules handled in this work. In Section III, we describe the proposed algorithms to improve pin access and to perform design-rule-aware routing, whose effectiveness is evaluated in Section IV. Section V concludes the paper.

II. Preliminaries

In this section, we will describe some basic terminologies and concepts related to the routing grid graph, design rules, and constraints.

A. Routing Grid Graph

Better routability can be achieved when wires with different directions are separate to different metal layers. For each metal layer, cell libraries define a *preferred direction* with which the routing wires are preferred to be aligned. Preferred directions of adjacent layers are usually perpendicular with each other. *Tracks* on each metal layer are ideal routing positions along the preferred direction defined during floorplanning. Wires routed on tracks are less likely to incur violations since the tracks are well spaced according to the technology and constraints.

Because of the perpendicular nature between the preferred directions on adjacent layers, each track on a layer will intersect with the projections of the tracks on its adjacent layers at many points. All these intersection points are considered as grid points to form a 3-D grid graph on which maze routing will be performed. For the edges in the grid graph, besides the edges connecting neighboring grid points on the same track, there are also cross-layer edges that connect grid points vertically aligned on adjacent layers. For efficient query and memory usage, we adopted a two-level framework that consists of a global grid graph and local ones. The global grid graph maintains the information of all the routed wires and inserted vias. The local grid graph maintains the edge costs from the information on the global grid graph to perform maze routing for a net.

B. Design Rules and Constraints

More design rules are introduced in the advanced technology nodes. Three new design rules highlighted in the recent ISPD'19 detailed routing contest [15] are illustrated as follows.

In sub-90 nm technologies, the parallel run length (PRL) spacing requirements between two wires of different nets depend on both width and PRL of the two wires [16]. Considering the three cases of different relative positions between two metal rectangles (R1 and R2) shown in Figure 1, the effective width $\max(w_1, w_2)$ is the same among all the three cases but the PRLs are different. Longer PRL l_t will result in larger spacing s_t while shorter PRL l_b will result in smaller spacing s_b. Rectangles which do not overlap in the x- or the y- direction are considered as having negative PRL and their spacing in Euclidean distance s_m should not be less than the zero PRL spacing requirement.

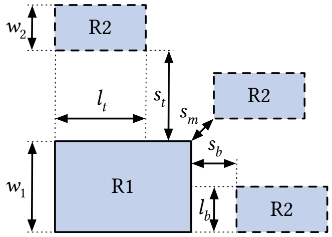


Fig. 1: Parallel run length spacing.

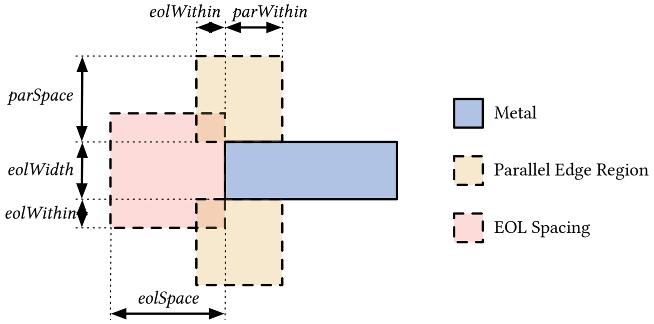


Fig. 2: End-of-line spacing with parallel edge.

Metal end-of-line (EOL) with parallel edges will cause additional stitches in double patterning lithography, that may lead to yield degradation or even cause native conflicts, which are undecomposable in double patterning [17]. A new kind of EOL spacing requirement that depends on the existence of other parallel edges is defined for the generation of lithography-friendly layouts. On both sides of the EOL, two yellow parallel edge regions are defined as shown in Figure 2. Their width is the parallel edge spacing $parSpace$ and their length is $eolWithin + parWithin$. This EOL spacing rule applies, i.e., no metal overlaps with the red region, if a parallel routing wire overlaps with the parallel edge regions.

Optical proximity correction (OPC) places corner stressing serifs in corners to prevent energy scarcity [18]. However, aggressive OPC will potentially cause metal shorts between close by corners. The corner-to-corner spacing rule requires a safe distance which increases monotonically with the maximum width of involved corners. An example is shown in Figure 3. If the widest rectangle of a metal corner is wider than a specified width $eolWidth$, and if there is another metal corner and their PRL is non-positive, i.e., they do not overlap in the x- or y-directions, then, their L_{inf} distance should not be less than the corner-to-corner spacing which can be looked up from a table according to the value of $\max(w_1, w_2, w_3)$ for this example.

III. Algorithms

The overall algorithmic flow of our router is illustrated in Figure 4. It consists of three stages. (1) Given a placement solution with special net structures and a global routing solution in the form of route guides for the detailed router, we first assign access points for each cell pin and I/O pin, which will be explained in Section III-A. Note that the special nets

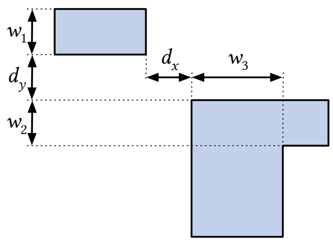


Fig. 3: Corner-to-corner spacing.

have been routed and their structure should not be changed. (2) Next, the nets are routed one by one, followed by rounds of rip-up and reroute. In each round of rip-up and reroute, the route guides are expanded, which will be explained in Section III-B. Maze routing of each net is distributed to a multi-threading scheme, followed by the via type selection for each net, which is also multi-threaded, as described in Section III-C. (3) Finally, a post-routing refinement stage is conducted to resolve corner-to-corner spacing violations. Some additional techniques to handle the new spacing constraints will be explained in Section III-D.

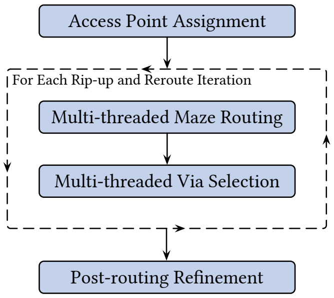
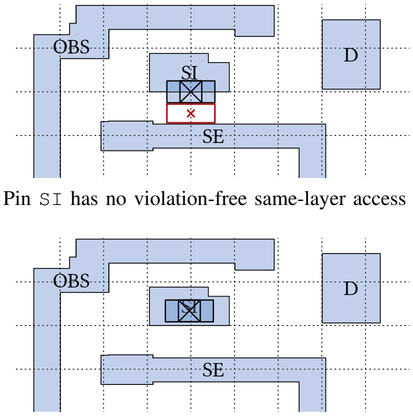

Access Point Assignment
For Each Rip-up and Reroute Iteration
Multi-threaded Maze Routing
Multi-threaded Via Selection
Post-routing Refinement

Fig. 4: Proposed Routing Flow.

A. *Pin Access*

Pin access is one of the most difficult problems in detailed routing. In order to simplify and tackle the pin access problem and to improve the routability, for each pin, we assign several same-layer grid points in its surrounding area as its *access points*. During the maze routing of the net connecting the pin, if one of the access points is reached by the routing path, the pin is considered as being connected. An L-shape metal will be inserted to finish the connection in case the access point reached is not sufficiently connected to the pin. We also penalize the grid points on the upper layer above the pin for other nets to guarantee that the pin can be reached and fade the penalty after the pin is successfully connected.

With the advancement of technology nodes, the number of routing tracks in a single placement row decreases. The limited number of in-cell tracks has brought high routing density and mutual blockages between accessing wires of neighboring pins. For some pins, there is even no violation-free same-layer grid point that can be assigned as an access point. More specifically, no via type can be used to connect to the pin directly or by a routing wire without causing any design rule violation. An example is shown in Figure 5(a), where pin SI is embraced by other pins and a via inserted at the grid point marked with a black cross will violate the PRL spacing rule with pin SE below. In fact, there is no grid point at which a via can be inserted to connect to SI, as either the selected via or the routing wire will cause violations.



(a) Pin SI has no violation-free same-layer access point.

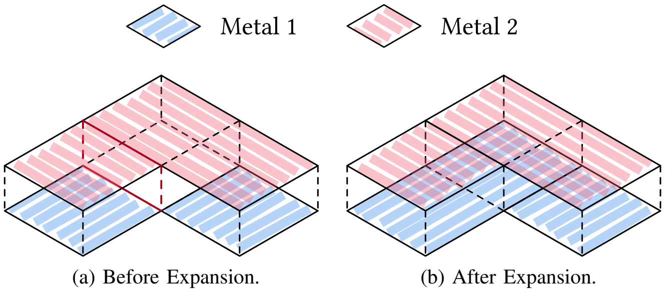
(b) Connect SI to upper-layer access points with an off-track via.

Fig. 5: Off-track Pin Access.

To handle this issue, we will also assign grid points on the upper layer as access points and use possibly off-track vias to connect pins with upper-layer access points. An example is shown in Figure 5(b), where the grid points surrounding the pin location on the upper layer are marked as access points of the pin. A proper off-track position (e.g., the center point of the pin shape) is selected to insert a violation-free via to reach the upper-layer access point. These off-track vias to resolve pin access problems are identified before maze routing and will not be moved during rip-up and reroute. Therefore, these off-track vias are marked as obstacles during routing to avoid violations with the wires and vias of other nets.

B. *Local Grid Graph Construction*

A local grid graph to perform maze routing for a net is generated based on the corresponding route guides before routing the net. For each iteration, the local grid graph is expanded in the x- and y-directions to provide larger solution space. If the number of violations in a part of local grid graph exceeds a certain threshold, it will even be extended to adjacent layers. Besides, it is possible that the original route guides of a net is not sufficient for making the connections by wires routing in the preferred direction only. As the example shown in Figure 6(a), the preferred directions of Metal 1 and Metal 2 are shown in blue and red strips, respectively. The red rectangle highlights the position where this group of route guides are disconnected, unless wrong-way wires are routed, which on the other hand will affect the routability. To guarantee a routable solution on the local grid graph, for each pair of route guides on adjacent layers with overlapping, we will expand the guides in their preferred directions to create a



(a) Before Expansion. (b) After Expansion.

Fig. 6: Unconnected Route Guides.

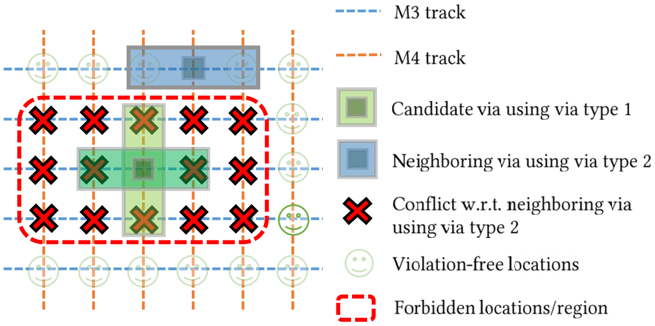

M3 track
M4 track
Candidate via using via type 1
Neighboring via using via type 2
Conflict w.r.t. neighboring via using via type 2
Violation-free locations
Forbidden locations/region

Fig. 7: Via-via LUT.

full connection. Figure 6(a) shows the expanded route guide of this example.

C. Via Type Selection

To enhance routability, multiple via types are provided for each cut layer. However, determining the via location and via type simultaneously is challenging due to the complicated metal shapes and cut patterns in the surrounding of the vias. To handle this efficiently, we will first construct a set of via conflict lookup tables (LUTs). Next, via insertion and via type selection will be performed for each net separately during routing. We will first determine the via locations and generate the routing topology for a net. Via type selection will then be performed to find the best via type locally. After routing all the nets, all via type and location information will be available, an additional round of via type selection will be performed in a post refinement stage to finally decide the best via types for the nets globally.

1) Construction of Conflict LUTs: The via insertion and via type selection processes require violation detection between the inserted via and the surrounding wires, vias, pins, and obstacles. These query operations happen so frequently that the run time of on-the-fly detection for every via insertion candidate position is not affordable. Fortunately, since we are working on a relatively regular grid graph, a set of light-weight LUTs can be constructed to accelerate the process. In general, via-related spacing violations can be divided into three categories:

1) Via-pin/obstacle conflicts: a via may have spacing violations with fixed pins or obstacles on the lower or the upper metal layers that it connects. The locations with this kind of violations will be marked before the routing stage and will be avoided during via insertion.
2) Via-wire conflicts: a via may have spacing violations with the wire on the lower or the upper metal layers that it connects.
3) Via-via conflicts: a via may have spacing violations with not only the vias on the same cut layer but also with vias on adjacent cut layers.

For each via type, the conflict LUTs with wires on the lower or the upper metal layer are built to provide immediate responses for queries about whether or not a neighboring edge conflicts with a given via type at a given location. Similarly, for each pair of via types on the same cut layer or on adjacent cut layers, a conflict LUT is built. For example, for the pair of via type 1 and via type 2 in Figure 7, a conflict LUT is built to mark the neighboring *forbidden regions* of via type 1 with respect to via type 2. When inserting a via of type 1 at the center, any routed via of type 2 inside the forbidden regions of via type 1 will trigger a conflict and we need to ensure that is not the case before inserting the type 1 via.

2) Via Type Selection with Pessimistic Query: When routing a net, we first assume that all the newly inserted vias are of the default via type, and determine their locations by minimizing the number of conflicts with the surrounding wires, vias, pins, and obstacles. After generating the routing topology of the net, via type selection is performed to determine the best type at the given via location. Meanwhile, a *global routed via map* is built and maintained to record the locations and types of all the settled vias.

Obviously, for a given via type at a given location, the efficiency of detecting the surrounding via-via violations will dominate the run time of the entire selection process. This is because the surrounding vias may be of different types and we need to look up different via-via conflict LUTs. A naive implementation is, for a given via type, all its via-via LUTs are checked one by one to count the accumulated violation number with respect to its surrounding vias. However, a lot of redundancy will be introduced with such an approach, since each location may be visited several times, and a long run time will be resulted.

In order to avoid repeated queries on the surrounding locations, an alternative approach is to identify all those neighboring vias that may conflict with the given via type using one single via-via LUT. Conceptually, this approach is conducted in a pessimistic manner, that is, a neighboring via (regardless of its via type) will first be marked as a suspicious conflicting via if it is inside **any** forbidden region of the given via type at the given location. After that, all suspicious conflict vias will be checked by looking up into their respective via-via LUTs to verify whether there are real conflicts.

Based on this idea, we construct a set of pre-computed *merged via-via LUTs* to find out all the neighbouring suspicious conflict vias of a given via type in a one-time-effort. As illustrated in Figure 8, for a specific via type (e.g. via type 1

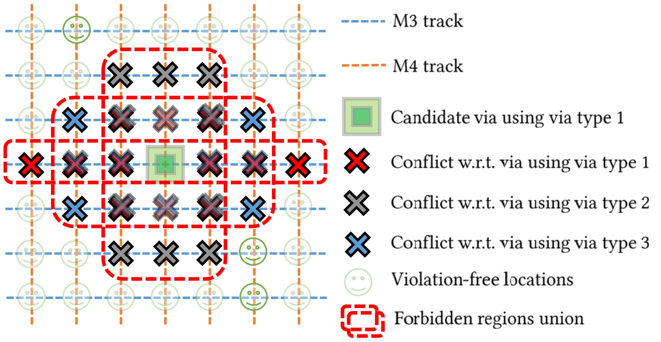

M3 track
M4 track
Candidate via using via type 1
Conflict w.r.t. via using via type 1
Conflict w.r.t. via using via type 2
Conflict w.r.t. via using via type 3
Violation-free locations
Forbidden regions union

Fig. 8: Merged Via-via LUT.

at the centre), the corresponding merged via-via LUT marks the union of the forbidden regions of via type 1 with respect to all other possible via types (e.g. via types 1, 2 and 3). Any neighbouring via (regardless of its via type) inside the forbidden region union of the merged LUT will be marked as a suspicious conflicting via. In the next verifying stage, if a neighbouring suspicious via has already had its type selected, its via type can immediately be looked up from the global routed via map (default via type is used otherwise). We can then identify if a neighbouring suspicious via is truly conflicting with the given via type at the given location by accessing the respective via-via LUTs. As a result, the best via type with the lowest violation count can be selected efficiently.

3) Multi-stage Selection and Multi-thread Scheduling: Via type selection is an order-sensitive process. Different orders of via type selection may lead to different results, and an optimal selection result may require simultaneous type changes for adjacent vias. Thus, with the information provided by the global routed via map, an additional round of via type selection is performed in the post refinement stage to reduce the order sensitivity of the selection results. We have implemented an efficient multi-thread scheduling scheme for via type selection, which can significantly accelerate the multi-stage via type selection process. Nets without routing topology overlap are assigned to different batches, and the via type selections of different batches will be executed in parallel.

D. More Techniques for New Design Rules

Techniques for handling EOL spacing with parallel edges and corner-to-corner spacing introduced in the ISPD'19 Contest are presented as follows.

1) EOL Spacing: There are two kinds of EOL spacing constraints on each metal layer: with or without parallel edges. On each layer, the $eolSpace$ (see Section II-B) of the EOL constraint with parallel edges (EWP) is larger than that without parallel edges (EWOP). When a metal violates only the EWP (not the EWOP) but the EWP does not apply since there is no parallel edge overlapping with the parallel edge regions, the track segments within the parallel edge regions should be forbidden to use in order not to trigger the EWP violation. In this case, some available tracks will be blocked in the middle, which can affect routability. To handle this issue, we only consider the EWOP but with the spacing requirement $eolSpace$ increased to that of the EWP. This pessimistic method is practical since, in most of the cases, the two $eolSpace$ values are close to each other and it is not beneficial to handle the complex conditions for the EWP.

2) Corner-to-corner Spacing: There are three sources for the corner-to-corner (C2C) spacing violations described in Section II-B, i.e., the violations can be caused by vias, wires, or pins/obstacles. We only handle the first two, vias and wires, as the majority of pins are on Metal 1 but there are rarely C2C rules on the bottom layer. For the violations caused by vias, we handle the C2C spacing rule using the LUTs described in Section III-C. For the violations caused by wires, they only exist at the corners of wrong-way wires because based on our observation, the $eolWidth$ for C2C spacing is wider than the wire width. Therefore, we will transform the corners of wrong-way wires to EOLs by adding small metal rectangles on track and handle the EOL spacing in maze routing.

IV. Experimental Results

We implement the proposed detailed routing algorithm in C++ programming language. Rsyn [19] is used as the parser of the LEF/DEF file format. We run all experiments on a 64-bit Linux machine with eight cores of Intel Xeon 2.2 GHz CPUs and 64 GB RAM.

In the first experiment, we compare our proposed algorithm with the best score of each design in ISPD'19 Initial Detailed Routing Contest. The basic characteristic of the designs and the experimental results are listed in Table I. Our proposed algorithm performs 2% better on average. The weights of the cost function used in the contest and the detailed results obtained by our algorithm is listed in Table II. The units used in the wirelength and short area calculation are the Metal 2 pitch and the square of the Metal 2 pitch, respectively. We can see from the table that only marginal percentages of the wires and vias are routed in a non-preferred way.

In the second experiment, we compare with the state-of-the-art work [1] and the results are listed in Table III. Note that the router in [1] does not handle the new design rules like the length-dependent PRL spacing, EOL spacing with parallel edges, and corner-to-corner spacing. Therefore, we compare our router with it using the designs of ISPD'18 Intial Detailed Routing Contest [12]. The column # Spc denotes the total number of spacing violations as the evaluator in the contest does not provide the detailed number of violations on each design rule. Our proposed algorithm significantly improves the quality score by 69% with 41% more memory consumption and comparable running time. The route guide is better honored by our detailed router as the out-of-guide wirelength is reduced by more than 3 times. At the same time, we resolved almost all of the short area. The number of spacing rule violations is reduced by more than 49 times.

In the third experiment, we verify the effectiveness of off-track pin access described in Section III-A and the local grid graph expansion to adjacent layers described in Section III-B. As shown in Figure 9, with upper-layer access points, the

TABLE I: Comparison with The Best Scores in ISPD'19 Contest.

Design	# Std. Cells	# Block Macros	# Nets	# I/O Pins	# Layers	Die Size (mm^2)	Tech. Node	ISPD'19 Quality Score		Time (s)	
								Best	Ours	Best	Ours
ispd19_test01	8879	0	3153	0	9	0.148 × 0.146	32 nm	626129	631347	150	260
ispd19_test02	72094	4	72410	1211	9	0.873 × 0.589	32 nm	21942353	21460183	1442	2491
ispd19_test03	8283	4	8953	57	9	0.195 × 0.195	32 nm	1044187	1060117	63	89
ispd19_test04	146442	7	151612	4802	5	1.604 × 1.554	65 nm	22755592	21370286	1622	2877
ispd19_test05	28920	6	29416	360	5	0.906 × 0.906	65 nm	3301288	3292562	157	269
ispd19_test06	179881	16	179863	1211	9	1.358 × 1.325	32 nm	47916252	46130808	2918	4108
ispd19_test07	359746	16	358720	2216	9	1.581 × 1.517	32 nm	99164297	98256678	7174	9458
ispd19_test08	539611	16	537577	3221	9	1.803 × 1.708	32 nm	137211940	136114789	10335	13102
ispd19_test09	899341	16	895253	3221	9	2.006 × 2.151	28 nm	216753037	212577797	14173	17987
ispd19_test10	899404	16	895253	3221	9	2.006 × 2.151	32 nm	216333270	212777964	14955	18673
Avg. Ratio								1.02	1.00	0.66	1.00

TABLE II: Detailed Metrics on ISPD'19 Designs.

Design	Basic Use		Non-preferred Use					Rule Violations							
			Out-of-guide		Off-track										
	WL‡	# V‡	WL	# V	WL	# V	Wrong-way WL	# Shorts	Short Area	# Min Area	# PRL	# EOL	# Cut	# Adj‡	# C2C‡
Weight	0.5	4.0	1.0	1.0	0.5	1.0	1.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0
ispd19_test01	642383	36889	12499	1617	767	741	10884	96	36	33	9	23	15	3	58
ispd19_test02	24943322	808060	363055	33552	21884	23633	207317	1323	450	556	870	831	198	88	5920
ispd19_test03	842146	65566	12259	1667	1962	661	16081	74	25	45	209	119	33	76	109
ispd19_test04	30484232	1033993	568497	48221	15201	2	122225	1481	670	36	260	44	0	0	0
ispd19_test05	4779913	153447	13530	2699	2090	18	15245	279	68	3	150	13	0	0	0
ispd19_test06	66019435	1991004	662101	64728	25138	12507	444711	2898	1008	656	1096	689	296	49	1229
ispd19_test07	122505150	4821351	855312	101530	48185	24324	608986	1915	601	1780	21580	1556	830	47	3900
ispd19_test08	188414168	7350652	1203856	162298	76715	36396	731245	2480	693	2585	4091	2649	1594	86	6488
ispd19_test09	285335800	12226079	2052346	276625	118145	60675	1215030	4989	1599	4913	6591	4696	2202	127	9567
ispd19_test10	282121356	12514819	2088315	256880	119526	60656	1418869	5273	1823	4705	6646	4374	1964	116	10646

‡ WL denotes wirelength. # V, # Adj, and # C2C denote the numbers of vias, adjacent cut and corner-to-corner violations.

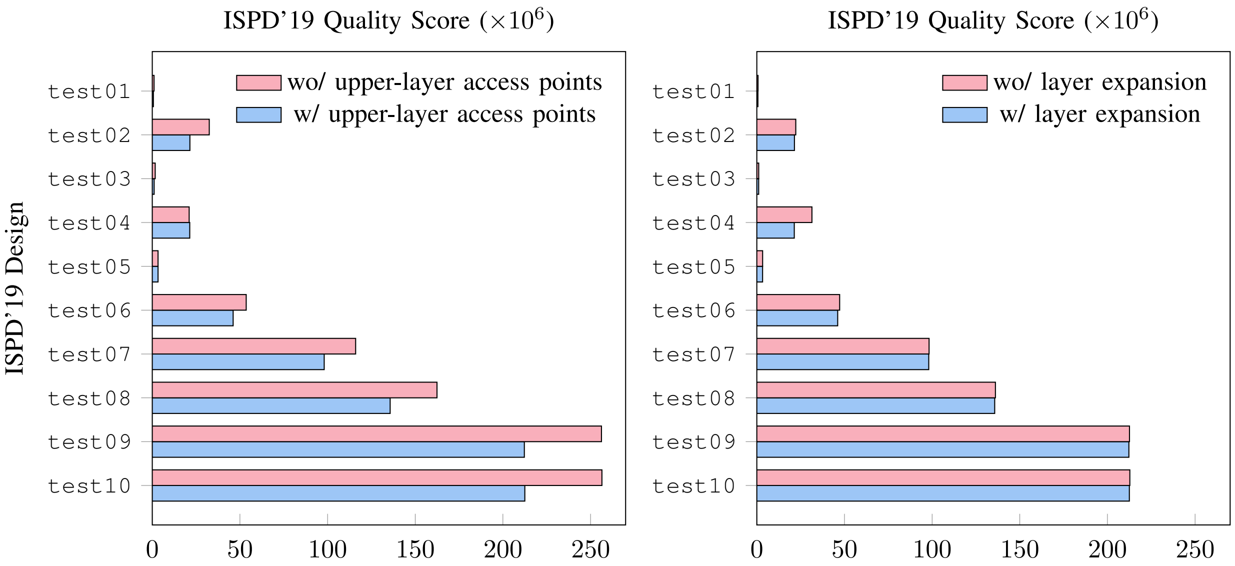
TABLE III: Comparison with the State-of-the-art Work on ISPD'18 Designs.

	Design	Basic Use		Non-preferred Use					Rule Violations			ISPD'18 Quality Score	Mem (GB)	Time (s)
				Out-of-guide		Off-track								
		WL	# V	WL	# V	WL	# V	Wrong-way WL	Short Area	# Min Area	# Spc			
	Weight	0.5	2.0	1.0	1.0	0.5	1.0	1.0	500.0	500.0	500.0	–	–	–
Ours	ispd18_test01	433538	32406	2187	442	414	0	5838	0	0	2	291307	0.41	18
	ispd18_test02	7832669	325618	40852	6534	5294	0	54151	0	0	57	4700255	1.86	141
	ispd18_test03	8718434	318353	68284	6673	6002	0	60028	379	0	97	5371855	3.84	377
	ispd18_test04	26410849	727732	347634	28839	17560	0	202113	86	93	588	15631883	12.36	1997
	ispd18_test05	27800973	965450	145035	21927	5841	3	76977	62	138	358	16357191	9.88	1109
	ispd18_test06	35702778	1480379	243634	36493	16548	16	118972	12	251	547	21624661	6.57	984
	ispd18_test07	65171367	2402251	395533	55611	33447	0	186904	145	364	186	38392522	14.40	2067
	ispd18_test08	65467807	2411866	393637	54455	33486	0	183358	156	390	177	38567127	15.12	1985
	ispd18_test09	54758969	2410569	357192	55216	26461	0	177411	10	513	113	33121862	11.23	1287
	ispd18_test10	68098610	2597471	1107400	101283	43250	0	235592	1206	595	736	41978776	11.91	3557
	Avg. ratio	1.00	1.00	1.00	1.00	1.00	–	1.00	–	–	1.00	1.00	1.00	1.00
[1]	ispd18_test01	434914	34443	4352	859	276	0	2363	15	0	122	362725	0.32	17
	ispd18_test02	7817285	339055	104720	11784	4353	0	22023	1330	0	1949	6366886	1.15	121
	ispd18_test03	8707641	331958	176736	10731	4344	0	22187	1982	0	2419	7430092	1.25	139
	ispd18_test04	26042785	701994	769265	31444	41791	0	89537	26329	0	11224	34112928	2.89	494
	ispd18_test05	27852167	942588	649224	43071	13390	0	63397	4722	0	7742	22805761	3.87	767
	ispd18_test06	35813473	1446807	976672	68656	20357	0	95811	12891	0	11023	33908653	5.16	1155
	ispd18_test07	65360688	2349580	2187794	101866	33105	0	170316	33041	0	14880	63816462	8.86	2071
	ispd18_test08	65668468	2360231	2288159	102982	33373	0	170583	22353	0	14384	58501486	8.92	2060
	ispd18_test09	54993356	2358857	1604576	115465	29620	0	168722	17316	0	14470	50010785	8.52	2016
	ispd18_test10	68282001	2532666	2826908	140343	32865	0	180586	150705	0	20837	128141527	8.98	2132
	Avg. ratio	1.00	1.00	3.62	1.75	1.20	–	0.68	–	–	49.86	1.69	0.59	0.85

average score improves by 26%. With the layer expansion of local grid graphs, the average score improves by 7%.

V. Conclusion

In this paper, we presented a detailed routing framework with correct-by-construction design rule satisfaction. First, we



(a) Upper-layer Access Points with Off-track Vias.

(b) Layer Expansion of Local Grid Graphs.

Fig. 9: ISPD'19 Quality Score with and without Upper-layer Access Point or Local Grid Graph Layer Expansion.

demonstrated the handling of pins which have no violation-free access points and a scheme of route guide expansion and shape refinement, which broaden the solution space in congested designs. We further illustrated a method of via insertion and via type selection. Our proposed framework was shown superiorly scalable on all ISPD'18 and ISPD'19 Initial Detailed Routing Contest designs. Compared with the best score of each released design in the ISPD'19 Contest, we achieved 2% better score on average. Compared with the state-of-the-art work [1], we produced much fewer spacing violations and achieve 69% better score. As part of the future work, we plan to extend our method to have a more accurate parallel run length spacing modeling.

References

[1] G. Chen, C.-W. Pui, H. Li, J. Chen, B. Jiang, and E. F. Young, "Detailed routing by sparse grid graph and minimum-area-captured path search," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, pp. 754–760, ACM, 2019.

[2] W.-H. Liu, W.-C. Kao, Y.-L. Li, and K.-Y. 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, 2013.

[3] D. Park, I. Kang, Y. Kim, S. Gao, B. Lin, and C.-K. Cheng, "ROAD: Routability analysis and diagnosis framework based on sat techniques.," in *ACM International Symposium on Physical Design (ISPD)*, 2019.

[4] D. Z. Pan, B. Yu, and J.-R. Gao, "Design for manufacturing with emerging nanolithography," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 32, no. 10, pp. 1453–1472, 2013.

[5] G. Hills, M. G. Bardon, G. Doornbos, D. Yakimets, P. Schuddinck, R. Baert, D. Jang, L. Mattii, S. M. Y. Sherazi, D. Rodopoulos, *et al.*, "Understanding energy efficiency benefits of carbon nanotube field-effect transistors for digital VLSI," *IEEE Transactions on Nanotechnology (TNANO)*, vol. 17, no. 6, pp. 1259–1269, 2018.

[6] M. Frederick, "Poly pitch and standard cell co-optimization below 28nm," in *IEEE International Electron Devices Meeting (IEDM)*, IEEE, 2014.

[7] H. Li, S. Patnaik, A. Sengupta, H. Yang, J. Knechtel, B. Yu, E. F. Young, and O. Sinanoglu, "Attacking split manufacturing from a deep learning perspective," in *ACM/IEEE Design Automation Conference (DAC)*, p. 135, ACM, 2019.

[8] T. Nieberg, "Gridless pin access in detailed routing," in *ACM/IEEE Design Automation Conference (DAC)*, pp. 170–175, ACM, 2011.

[9] Y. Zhang and C. Chu, "Regularroute: An efficient detailed router applying regular routing patterns," *IEEE Transactions on Very Large Scale Integration Systems (TVLSI)*, vol. 21, no. 9, pp. 1655–1668, 2013.

[10] X. Xu, B. Yu, J.-R. Gao, C.-L. Hsu, and D. Z. Pan, "PARR: Pin-access planning and regular routing for self-aligned double patterning," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 21, pp. 42:1–42:21, May 2016.

[11] H. Li, W.-K. Chow, G. Chen, E. F. Young, and B. Yu, "Routability-driven and fence-aware legalization for mixed-cell-height circuits," in *ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6, IEEE, 2018.

[12] S. Mantik, G. Posser, W.-K. Chow, Y. Ding, and W.-H. Liu, "Ispd 2018 initial detailed routing contest and benchmarks," in *ACM International Symposium on Physical Design (ISPD)*, pp. 140–143, ACM, 2018.

[13] A. B. Kahng, L. Wang, and B. Xu, "Tritonroute: an initial detailed router for advanced vlsi technologies," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–8, IEEE, 2018.

[14] F.-K. Sun, H. Chen, C.-Y. Chen, C.-H. Hsu, and Y.-W. Chang, "A multithreaded initial detailed routing algorithm considering global routing guides," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2018.

[15] 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 *ACM International Symposium on Physical Design (ISPD)*, pp. 147–151, ACM, 2019.

[16] Z.-D. Qi, Y.-C. Cai, and Q. Zhou, "Design-rule-aware congestion model with explicit modeling of vias and local pin access paths," *Journal of Computer Science and Technology*, vol. 30, no. 3, pp. 614–628, 2015.

[17] B. Yu, K. Yuan, D. Ding, and D. Z. Pan, "Layout decomposition for triple patterning lithography," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 34, pp. 433–446, March 2015.

[18] M. L. Côté, C. Pierrat, and P. Hurat, "Accelerated layout processing using opc pre-processing," Oct. 19 2004. US Patent 6,807,663.

[19] G. Flach, M. Fogaça, J. Monteiro, M. Johann, and R. Reis, "Rsyn: An extensible physical synthesis framework," in *ACM International Symposium on Physical Design (ISPD)*, pp. 33–40, ACM, 2017.