This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2021.3100296, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

Timing-Aware Layer Assignment for Advanced Process Technologies Considering Via Pillars

Genggeng Liu, Xinghai Zhang, Wenzhong Guo, Xing Huang, Wen-Hao Liu, Kai-Yuan Chao, and Ting-Chi Wang

***Abstract*—Interconnect delay is a key factor that affects the chip performance in layer assignment. Particularly in the advanced process technologies of 5nm and beyond, interconnect delay has grown significantly due to the increase of circuit scale. Moreover, coupling effect existed in wires reduces the accuracy of delay evaluation. On the other hand, the size of vias is often ignored in layer assignment, which enlarges the mismatch between global routing and detailed routing. To solve these problems, we propose *VPT*, a timing-aware layer assignment algorithm considering via pillars, which includes the following five key techniques: 1) via pillar structure combined with non-default-rule (NDR) wires is adopted to form a net delay optimization system for advanced process technologies; 2) a synthetical model that can adapt to varying types and sizes of both vias and wires is designed to evaluate overflow effectively; 3) a sorting strategy is devised to reduce uncertainty of layer assignment flow and improve stability of the proposed algorithm; 4) an awareness strategy based on multi-aspect congestion assessment is designed to reduce overflow significantly; 5) a net scalpel algorithm is devised to minimize the maximum delay of nets, so that the timing behaviors can be improved systematically. Experimental results on multiple benchmarks confirm that the proposed algorithm leads to lower delay and less overflow, while achieving the best solution quality among the existing algorithms with the shortest runtime.**

***Index Terms*—Layer Assignment, Via Pillar, Non-Default-Rule Wires, Delay, Congestion.**

I. Introduction

With the development of process technologies, the scale of integrated circuit increases significantly, which makes net delay greater. The increase of net delay has negative influence on chip performance. In the multilayer routing region, pins of each net are connected by wires and vias, and thus the net delay is mainly composed of wire delay and via delay. As an important part of physical design, layer assignment plays an important role in adjusting delay results due to the characteristics of routing region for advanced process technologies. As shown in Fig. 1(a) and 1(b), based on 2D routing solution, wires are assigned to different routing layers, and vias are generated to form a 3D routing solution in layer assignment. Furthermore, wire widths on different routing layers are inconsistent, which leads to the difference of wire resistance among different layers [1], [2]. Meanwhile, inconsistency also exists in the resistances of vias connecting different routing layers. Since delay calculation depends on

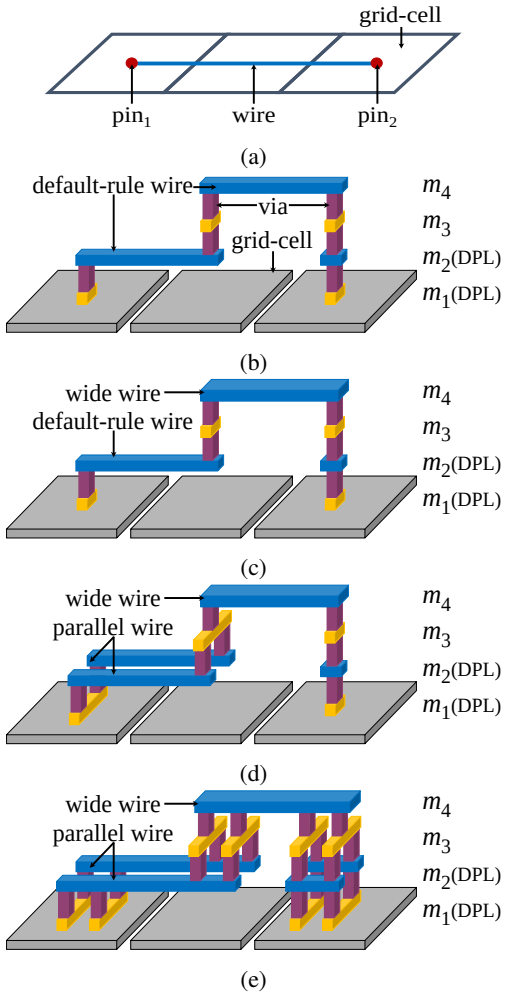


Fig. 1: (a) A 2D routing solution of a net. (b) The 3D routing solution corresponding to (a) with default-rule wires. (c) The updated 3D routing solution based on (b) with wide wire. (d) The updated 3D routing solution based on (c) with parallel wires. (e) The updated 3D routing solution based on (d) with via pillar structure.

A preliminary version of this work was published in the Proceedings of IEEE/ACM Design, Automation and Test in Europe Conference and Exhibition, 2020 [32]. (*Corresponding author: Xing Huang*)

G. Liu, X. Zhang, and W. Guo are with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, Fujian 350116, China (e-mail: liu_genggeng@126.com; zhang_xinghai@126.com; guowenzhong@fzu.edu.cn).

X. Huang is with the Chair of Electronic Design Automation, Technical University of Munich, Munich 80333, Germany (e-mail: xing.huang1010@gmail.com).

W.-H. Liu is with the Cadence Design Systems Inc., Austin, TX 78750, USA (e-mail: whliu@cadence.com).

K.-Y. Chao is with Skymizer, Taipei 100, Taiwan (e-mail: kychao@skymizer.com).

T.-C. Wang is with the Department of Computer Science, National Tsing Hua University, Hsinchu 30013, Taiwan (e-mail: tcwang@cs.nthu.edu.tw).

resistance and capacitance, the change of resistance directly affects delay. Therefore, layer assignment determining the locations of wires and vias has a significant impact on delay.

Besides making full use of low-delay routing resources, non-default-rule (NDR) wire technique is also an effective way to optimize wire delay and has been widely adopted in previous work [3]–[5]. NDR wires reduce resistance by enlarging wire width, so that the wire delay can be optimized. On the upper layers of multilayer routing region, NDR wires are implemented in the form of wide wires, as Fig. 1(c) shows. The width of a wide wire cannot be an arbitrary value due to the manufacturing limitations, and should be pre-defined and certified by foundries. As for the lower layers in sub-16nm designs, NDR wires have to be realized in parallel wires instead of wide wires due to strict manufacturing and design rules in double patterning lithography (DPL). Parallel wires expand wire width using two default width wires for a connection as Fig. 1(d) shows.

Apart from wire delay, via delay is also a crucial part of net delay. In advanced process technologies, via pillar technique has great potential in optimizing via delay, and becomes indispensable for high-performance physical design [6], [7]. In Fig. 1(e), a via pillar structure consists of closely spaced pairs of vias and wires, and each pair of wires is routed in the preferred direction of each layer. Unlike normal via structure shown in Fig. 1(d), each layer within a via pillar structure contains multiple vias and wires. Compared with the normal via structure in Fig. 1(d), via resistance is further reduced due to multiple vias in the via pillar structure in Fig. 1(e), so that via delay can be reduced. In spite of these benefits, via pillar structure needs more routing resources than traditional structure. Therefore, it is worth exploring how to use via pillar technique to optimize delay without hurting routability.

On the other hand, via size is often ignored in global routing, and thus the congestion problem caused by vias in the routing region cannot be considered. Moreover, since layer assignment is the intermediate process between 2D global routing and detailed routing, ignoring via size in layer assignment can lead to the mismatch between global routing and detailed routing. Therefore, it is necessary to take via size into consideration in delay-driven layer assignment procedure.

Based on the above analysis, this paper proposes *VPT*, a timing-aware layer assignment algorithm for advanced process technologies considering via pillars. The main contributions are outlined below.

- We propose an optimization method to combine via pillars and NDR wires to reduce via delay and wire delay respectively, so that net delay can be optimized based on advanced process technologies.
- We formulate a synthetical layer assignment model that considers varying types and sizes of both vias and wires to evaluate overflow properly.
- We design a total path length sorting strategy to reduce the uncertainty of layer assignment flow and improve the stability of the proposed algorithm.
- We propose a multi-aspect congestion awareness strategy to optimize congestion, and it can reduce delay simultaneously.
- We design a maximum-delay net scalpel algorithm to reduce the maximum delay, so that the timing performance of chips can be improved.

The rest of this paper is organized as follows. Section II analyses related work. Section III introduces the problem formulation. Section IV describes the consideration for coupling effect. Section V introduces the details of the proposed algorithm. Section VI shows the performance of the proposed algorithm and the conclusions are drawn in Section VII.

II. Related Work

To compare this work with related work in terms of considerations, we make a summary in Table I. Wire overflow and via count are important issues in layer assignment, and some studies focused on them have been carried out [8]–[12]. [8] took the total overflow and the maximum overflow as the congestion constraints and illustrated the rationality. [9] proposed a layer shifting method and a reassignment method, to get a solution with better routability. [10], [11] proposed two global routing algorithms and took congestion into consideration. [12] proposed an efficient layer assignment scheme to concurrently consider the wire segments of all nets. Besides, since via overflow is an important factor in affecting routability, [13]–[16] focused on via overflow minimization. Specifically, net order determination method, negotiation-based method, and linear programming method were adopted in [13], [15], and [16], respectively. Particularly in [16], a layer assignment model was proposed to capture the impact of local congestion caused by varying-size vias.

As very large scale integration (VLSI) technology enters the nanoscale, interconnect delay becomes a critical factor in circuit performance. [17] presented a statistical methodology based on extreme value theory for worst case delay estimation. [18]–[21] focused on delay testing and proposed different methods for various situations. [22] proposed a data-driven approach to reduce timing analysis effort for faster design convergence. [23] and [24] showed that the timing constrained minimum cost layer assignment problem is NP-complete, and proposed a fully polynomial time approximation solution. [25] proposed a systematic layer assignment method for double patterning with considering timing critical paths and coupling effect. [26] designed an efficient layer assignment scheme under a multilayer interconnect structure for minimizing delay and via count. [27]–[29] further took via overflow into consideration in timing-driven layer assignment and proposed a multiprocessing method to shorten runtime. Although NDR wires are able to optimize delay, [23]–[29] did not use NDR wires in layer assignment. [30] made up for this blank and took coupling effect into account to further optimize timing behaviors of layer assignment solution.

Although [16] and [30] proposed effective algorithms based on existing work, they did not consider some key factors in layer assignment. Specifically, [16] could not evaluate the impact of NDR wires and via pillars on congestion. Furthermore, [16] focused on the influence of via size on design rule constraints (DRC) violation, but did not take delay into consideration. On the other hand, [30] ignored the congestion

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2021.3100296, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

TABLE I: Comparison with related work

Work	Wire overflow	Via overflow	Delay	Coupling effect	Via count	Varying-size vias	NDR wires	Via pillars
[8]–[12]	✓				✓			
[13]–[15]	✓	✓			✓			
[16]	✓	✓			✓	✓		
[23], [24]			✓					
[25]			✓	✓	✓			
[26]	✓		✓		✓			
[27]–[29]	✓	✓	✓		✓			
[30], [32]	✓		✓	✓	✓		✓	
this work	✓	✓	✓	✓	✓	✓	✓	✓

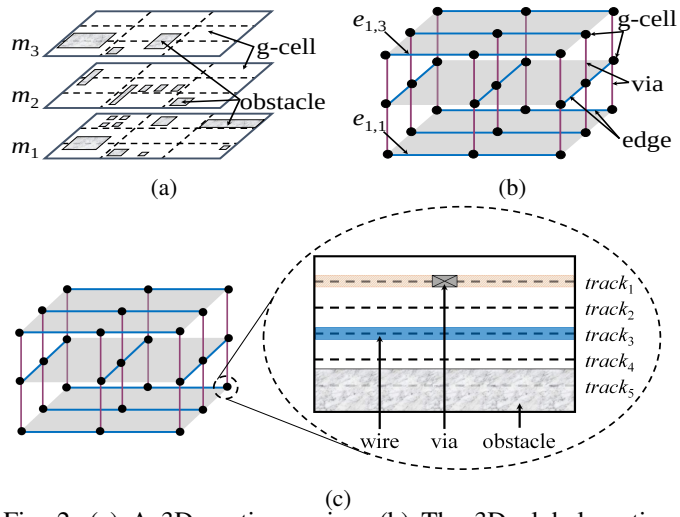


Fig. 2: (a) A 3D routing region. (b) The 3D global routing model corresponding to (a). (c) The internal details of a g-cell.

problem caused by vias. Besides, [30] did not combine NDR wires and via pillars to form a relatively complete optimization system for net delay. Based on the above analysis, we propose VPT to make up for the blank and overcome the shortcomings.

III. Problem Formulation

The proposed algorithm not only uses default-rule wire and normal via structure, but also introduces NDR wires and via pillar structure to optimize net delay. Besides, the size of grid cell (g-cell) is further restored to evaluate the congestion caused by vias and wires.

A. Routing Region

Multilayer routing region containing obstacles is usually divided into rectangles with the same size as shown in Fig. 2(a). In Fig. 2(b), each rectangle is defined as a g-cell that is abstracted into a point in global routing. The adjacent g-cells in the preferred direction of the same layer are connected by edges. Two adjacent layers are connected by vias.

To consider the congestion problem caused by vias, the size of g-cells is restored as shown in Fig. 2(c). In this way, both routing track resources of edges and routing area resources of g-cells can be calculated. In Fig. 2(c), both the g-cell area and the edge tracks could be occupied by wires, vias, and obstacles. In the proposed design flow, we adopt a relatively flexible congestion model based on our design experience in industry [31]. Specifically, an accurate congestion model considering too many details can usually decrease routability of detailed routing, delay the design convergence, and may even result in design failure. In contrast, adopting a flexible congestion model in the global routing and layer assignment stages helps to improve the success rate of detailed routing. Thus, in the proposed congestion model, $track_1$ can not be used by a wire any more since the via blocks this track.

In the proposed congestion model, a g-cell/an edge in the routing space can be occupied by nets/vias/obstacles. There are two cases that overflow may occur: (1) If the total area of a g-cell occupied by nets, vias, and obstacles is more than its area capacity (i.e., the maximum allowed area occupation), a g-cell overflow occurs. (2) If the total number of tracks in an edge occupied by nets, vias, and obstacles is more than the capacity of this edge (i.e., the maximum allowed tracks that pass through the edge), an edge overflow occurs. The overflow $of(g)$ of a g-cell g and the overflow $of(e)$ of an edge e are computed as follows:

$$of(g) = \begin{cases} dc(g) - tc(g) & \text{if } g \text{ has overflow} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$of(e) = \begin{cases} dc(e) - tc(e) & \text{if } e \text{ has overflow} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $dc\,(g)$ and $tc\,(g)$ represent the current usage and the total area capacity of g, respectively. $dc\,(e)$ and $tc\,(e)$ represent the current usage and the total capacity of e, respectively. Both g-cell area overflow and edge overflow can be attributed to via occupation and wire occupation. Since overflow makes routability worse, the wire overflow of our work obeys the following congestion constraints [8]:

$$TWO(S_k) = TWO(S) \quad (3)$$

$$MWO(S_k) = \lceil MWO(S) \times (2/k) \rceil \quad (4)$$

where TWO and MWO represent the total wire overflow and maximum wire overflow of all the nets, respectively. S represents the given 2D global routing solution, and S_k represents the 3D global routing solution of S in a k-layer structure routing region. In addition, overflows in Equations (3) and (4) refers to only wire overflow, without via overflow.

Equation (3) ensures the wire overflow in the 3D routing solution is equal to that in the 2D routing solution. Equation (4) ensures that the peak congestion of an edge in the 2D routing solution can be assigned to its corresponding edges in the 3D routing solution uniformly. Meanwhile, Equation (4)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2021.3100296, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

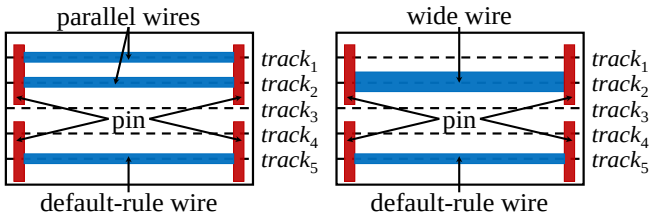


Fig. 3: Using different wires to connect two pins.

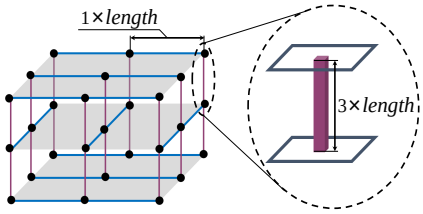


Fig. 4: The height of a via connecting two adjacent g-cells.

takes the preferred direction into consideration of each layer. In layer assignment process, if the congestion constraints are not satisfied, there are edges with unnecessary overflow in routing region. If a net passes through an edge which has unnecessary overflow, the net is called illegal net. In addition, if wire overflow exists in an edge in the 3D routing solution, but its corresponding edges in the 2D routing solution has no wire overflow, this type of wire overflow is referred to as unnecessary overflow, i.e., overflows that can be avoided.

B. Wire Types

Both default-rule wire and NDR wire are adopted in this work. On a certain metal layer, the width of a default-rule wire is called default width. The default width of an upper metal layer is usually greater than that of a lower metal layer. Compared with default-rule wires, NDR wires can reduce delay but occupy more routing resources due to greater wire widths. In Fig. 3(a) and 3(b), to connect two pins, a default-rule wire occupies one track, while parallel wires occupy two tracks, and a wide wire occupies three tracks. Since a wide wire requires more wire spacing than a default-rule wire, $track_1$ and $track_3$ cannot be used for routing in Fig. 3(b). Therefore, it is important to ensure good wire congestion when using NDR wires to reduce delay.

C. Via Types

Via size is considered in this work to evaluate congestion more detailedly. In Fig. 4, the height of a normal via is set as three units of wire length [33]. The horizontal projection of a normal via is a rectangle. Since the preferred direction between adjacent routing layers is orthogonal, the length and width of this rectangle depend on the widths of wires that are connected by this normal via, as shown in Fig. 5(a). If a normal via has no wire connection on a certain layer, the length or width of the horizontal projection is determined by the default wire width of this layer, as shown in Fig. 5(b). w_1 is the default width of layer m_1. Besides, via spacing is also considered and set as the corresponding wire spacing.

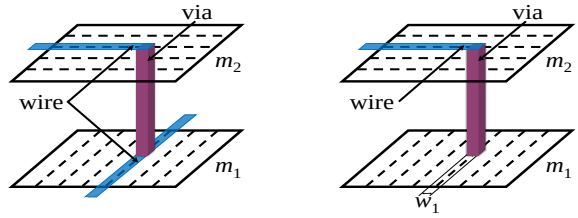


Fig. 5: The horizontal projection size of a via is affected by the condition of connected wires.

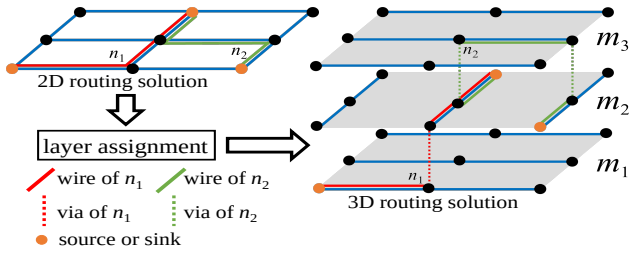


Fig. 6: A layer assignment example for nets n_1 and n_2.

D. Elmore Delay Model

Elmore delay model is adopted to compute net delay. Each net owns one source and multiple sinks. The delay $d(s)$ of segment s in a net is computed as follows:

$$d(s) = R(s) \times (C(s)/2 + C_{down}(s)) \tag{5}$$

where $R(s)$, $C(s)$, and $C_{down}(s)$ represent the resistance, capacitance, and downstream capacitance of segment s, respectively.

The total delay of all segments in path p from a sink to the source is represented by $d(p)$, which can be computed as follows:

$$d(p) = \sum\nolimits_{s \in S} d(s) \tag{6}$$

where S represents the set of all segments in the path p.

Net delay is the total weighted delay of all paths in a net. The delay of net n is represented by $d(n)$ and is computed as follows:

$$d(n) = \sum\nolimits_{p \in P} \alpha_p \times d(p) \tag{7}$$

where α_p and P represent the weight of path p and the set of all paths in net n, respectively. To make the delay of each path equally important, the weight of each path is set to $1/|P|$, where $|P|$ represents the number of paths in net n.

E. Problem Model of VPT

Let $G_k(V_k, E_k)$ be a k-layer routing region, where V_k and E_k represent the set of g-cells and the set of edges, respectively. The horizontal projections of G_k, V_k and E_k correspond to G, V and E, respectively. Thus the 2D model

TABLE II: Structure description for lookup table

Key			Value
Predictive wire density	Wire type	Routing layer	Capacitance and resistance

$G(V, E)$ corresponds to $G_k(V_k, E_k)$. Let S_k be a 3D global routing solution on G_k, and S be a 2D global routing solution on G. The VPT problem for timing-aware layer assignment algorithm considering via pillars can then be formulated as follows.

Given: S, G, and G_k.

Find: S_k without illegal nets, by assigning each segment of S to a corresponding edge of G_k subject to the congestion constraints.

Objective: 1) Minimize net delay of S_k. 2) Minimize the overflow of S_k.

Note: 1) The calculation of net delay is based on Section III-D. 2) The size of g-cells in G_k is restored as described in Section III-A. 3) The overflow can be caused by wires and vias. 4) Fig. 6 shows an example for layer assignment, where the types of wires and vias are described in Section III-B and III-C. In Fig. 6, based on the 2D routing solution of nets n_1 and n_2, our layer assignment procedure generates the 3D routing solution considering delay and overflow.

IV. Coupling Effect

Since coupling effect affects the accuracy of delay calculation, coupling effect is taken into consideration to evaluate delay more reasonably. In VLSI, due to coupling effect, wires that are closer can interact with each other, which makes some electrical characteristics of wires change, such as capacitance. Capacitance plays an important role in delay calculation. The exact wire coupling capacitance depends on the specific location of this wire, which is determined by track assignment. However, track assignment is not performed during global routing, thus we obtain the coupling capacitance by probability estimation based on a lookup table.

In the lookup table, the primary key is determined by layer, wire type and wire density. Each entry includes a capacitance value and a resistance value, as shown in Table II. Therefore, the wire capacitance and wire resistance of various wire types on different layers with different wire densities can be obtained in this table when calculating delay.

Each entry in the lookup table is calculated by following procedure. 1) A wire with a certain type is assigned to an edge on a layer. 2) Wires are randomly assigned to the corresponding edges on all possible layers such that the wire density on each layer is equal. 3) The coupling capacitance is extracted by fastCap [34] or FFTCAP [35]. Parallel wires are handled as [36] did. This process is repeated 100 times and the average values of coupling capacitances from the extraction are stored in the lookup table. The resistances of default-rule wires are set according to the ITRS roadmap. The resistances of NDR wires are set based on the roadmap with the assumption that each wire has the same lining thickness.

Since coupling effect is strong for the wires close to each other, the coupling effect of the wires assigned in a same 3D edge is relatively strong. On the other hand, coupling effect is sensitive to density. Thus, we introduce wire density and the corresponding definition is as follows:

$$wd(e_{i,j}) = \frac{tu(e_{i,j})}{ta(e_{i,j})} \tag{8}$$

where $wd(e_{i,j})$ represents the wire density of 3D edge $e_{i,j}$. $tu(e_{i,j})$ represents the number of tracks used by nets in $e_{i,j}$. $ta(e_{i,j})$ represents the number of tracks that can be used by nets in $e_{i,j}$.

Since nets are assigned one by one, the edges used by the assigning net may be used by other unassigned nets again. The wire densities of these edges are uncertain before all nets are assigned, which affects the calculation for delay of the assigning net negatively. To evaluate delay rigorously when assigning a net, average wire density $awd(e_i)$ of 2D edge e_i and predictive wire density $pwd(e_{i,j})$ of 3D edge $e_{i,j}$ are proposed and set as follows:

$$awd(e_i) = \frac{tu(e_i)}{ta(e_i)} \tag{9}$$

$$pwd(e_{i,j}) = \begin{cases} 1 & \text{if } e_{i,j} \text{ has overflow} \\ wd(e_{i,j}) & \text{else if } wd(e_{i,j}) \geq awd(e_i) \\ awd(e_i) & \text{otherwise} \end{cases} \tag{10}$$

where $tu(e_i)$ represents the number of tracks used by nets in e_i. $ta(e_i)$ represents the number of tracks that can be used by nets in e_i. In this way, the delay of an assigning net can be evaluated strictly. After all nets are assigned, the exact wire density of each edge can be obtained, and then each net delay is calculated again based on the exact wire density.

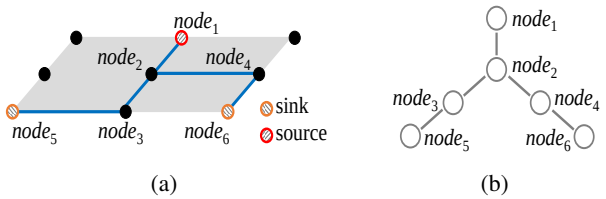
V. Details of the Proposed Algorithm

In this section, we introduce the details of VPT. Let DPLAA-MCAS represent a dynamic programming layer assignment algorithm (see Section V-A) combined with a multi-aspect congestion awareness strategy (see Section V-B). DPLAA-MCAS is the basic method in this work, and layer assignment process for each net is driven by this method. Based on DPLAA-MCAS, our framework is as follows. A sorting strategy (see Section V-C) is first proposed to determine a proper order for initial layer assignment. After an initial 3D routing solution is obtained, a negotiation-based method (see Section V-D) is presented to reassign illegal nets. Next a scalpel algorithm (see Section V-E) is proposed to optimize the maximum delay. Then a controlling strategy (see Section V-F) is presented to introduce NDR wires for wire delay optimization. Finally an optimization method (see Section V-G) is designed to further reduce net delay by combining via pillars and NDR wires.

A. Dynamic Programming Layer Assignment Algorithm

VPT is based on dynamic programming layer assignment (DPLA) algorithm to perform layer assignment for each net. DPLA algorithm regards the 2D routing solution of a net as a tree, and the source of the net is the root of the tree as shown in Fig. 7. The nodes and edges of the tree represent

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2021.3100296, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems



(a) (b)

Fig. 7: (a) A 2D routing solution of a net. (b) The routing tree corresponding to (a).

Algorithm 1: Layer Assignment Recursive (LAR) Function

Input: Net $n(V_n, E_n)$, tree node v_i, pin set P_n, 3D routing area G_k
Output: 3D routing solution

1 **Initialize:** The processing order of nets
2 **for** Each node v_j in $ch(v_i)$ **do**
3 **if** v_j is not visited **then**
4 $LAR(v_j, n, P_n)$;
5 **if** v_i is not a root node **then**
6 **for** Each layer L **do**
7 wt=$\emptyset$;
8 wt=$Rem(e_i, L)$;
9 **for** j= 0 to $|wt|$-1 **do**
10 **if** v_i is a leaf node **then**
11 $R(v_{i,L}, t_j) \leftarrow LeafSol(v_{i,L}, t_j, P_n)$;
12 **else**
13 $R(v_{i,L}, t_j) \leftarrow EnumSol(v_{i,L}, t_j, P_n)$;
14 $S(v_i)$=$S(v_i) \cup R(v_{i,L}, t_j)$;
15 **else**
16 $S(v_i) \leftarrow EnumSol(v_{i,L}, x, P_n)$;
17 $TopDownAssignment(S(v_i), G_k)$;

the involved g-cells and edges respectively in the 2D routing solution of the net. To consider delay, congestion and via count comprehensively, the cost function is defined as follows:

$$cost(n) = \alpha \times d(n) + \beta \times \sum\nolimits_{s \in n} cong(s) + \gamma \times vc(n) \quad (11)$$

where $cost(n)$ represents the cost for a 3D routing solution of net n. s represents a segment of n. $d(n)$ and $vc(n)$ represent the delay and the via count of this solution, respectively. $cong(s)$ represents the congestion cost of segment s. According to multiple experimental tests, the values of α, β, and γ are set to 0.3, 1, and 1, respectively.

The core of DPLA algorithm is layer assignment recursive (LAR) function and the pseudo code is shown in Algorithm 1. The notations used in Algorithm 1 are described in Table III. Given net $n(V_n, E_n)$, tree node v_i, pin set P_n, and 3D routing area G_k, DPLA algorithm calls LAR by setting node v_i to be the root of $n(V_n, E_n)$.

In Algorithm 1, there are four procedures, and the functions of them are described as follows.

- The procedure Rem is to determine the set of wire types on layer L, when connecting node v_i and its parent node.

TABLE III: Notations used in Algorithm 1

Notation	Definition
$n(V_n, E_n)$	A 2D routing tree corresponding to a net.
V_n	The tree node set of n.
E_n	The tree edge set of n.
P_n	The pin set of n.
G_k	A k-layer global routing area.
v_i	A tree node of n.
$ch(v_i)$	The set of child nodes belonging to node v_i.
wt	The set of wire types.
e_i	The edge connecting v_i and its parent node.
t_j	A wire type.
$R(v_{i,L}, t_j)$	The 3D routing solution for the subtree rooted at v_i when tree edge e_i is assigned to layer L with wire type t_j. It also stores the corresponding cost of each term in the cost function and the corresponding total capacitance.
$S(v_i)$	The set of 3D routing solutions for the subtree rooted at v_i.

- The procedures $LeafSol$ and $EnumSol$ generate 3D routing solutions for a leaf node and an internal node respectively, when the tree edge connecting v_i and its parent node is assigned to layer L with using wire type t_j and corresponding via type.
- The procedure $TopDownAssignment$ selects the solution with minimal cost to assign each tree edge to a 3D routing edge from the root node to each leaf node.

In Algorithm 1, before processing v_i, each child node of v_i is checked in lines 2-4. If child node v_j is not visited, LAR is called on v_j. Thereby, it can achieve the bottom-up manner from each leaf node to the root node. Lines 5-14 is the part for leaf nodes and internal nodes. When e_i is assigned to layer L with wire type t_j, the solutions of e_i is generated by $LeafSol$ or $EnumSol$ for a leaf node or an internal node, respectively. The solutions are collected in line 14. Lines 15-17 is the part for the root node. Since the root node has no parent node, the final solutions of the whole tree are generated by calling $EnumSol$ once, and then $TopDownAssignment$ bases on the solution with minimal cost to assign each tree edge of net n to a 3D routing edge from the root node to each leaf node.

B. Multi-Aspect Congestion Awareness Strategy

Many previous layer assignment algorithms only consider congestion cost in the event of wire overflow. However, when assigning a segment, even if there are two or more 3D routing solutions without overflow, these solutions have differences in influencing both local congestion and the flexibility of assigning subsequent nets. Besides, the overflow caused by vias is often ignored in congestion evaluation, and thus the interaction between routing resources and the objects occupying routing resources is not considered comprehensively. To consider all these factors synthetically, the congestion cost function is defined as follows:

$$cong(s) = cong(g_s) + cong(e_s) + of(e_s) \times h_e \quad (12)$$

$$cong(g_s) = \frac{dc(g) + gc(g)}{tc(g)} \quad (13)$$

$$cong(e_s) = \frac{dc(e) + gc(e)}{tc(e)} \quad (14)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2021.3100296, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

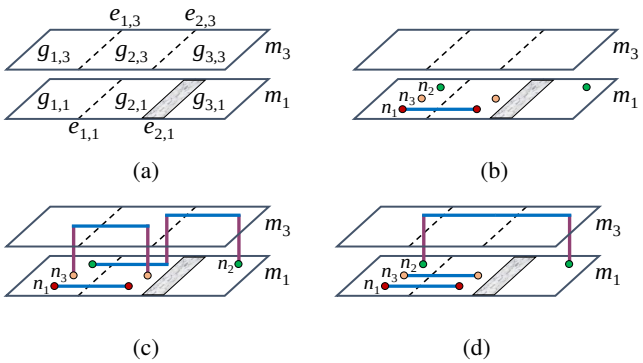


Fig. 8: (a) A 3D routing region. (b) A 3D routing solution of n_1. (c) A 3D routing solution of n_1, n_2 and n_3. (d) Another 3D routing solution of n_1, n_2 and n_3.

where s represents a wire segment or a via segment of a net. Segment congestion cost $cong\,(s)$ consists of g-cell congestion cost $cong\,(g_s)$, edge congestion cost $cong\,(e_s)$, and edge overflow cost $of(e_s) \times h_e$. g and e represent the g-cell and the edge that are passed by s, respectively. $tc\,(g)$ and $tc\,(e)$ represent the area capacity of g and the edge capacity of e, respectively. $dc(g)$ and $dc(e)$ represent the g-cell area and the number of edge tracks that have been used by obstacles and routed nets, respectively. $gc(g)$ and $gc(e)$ represent the g-cell area and the number of edge tracks that are required for s, respectively. $of\,(e_s)$ is the edge overflow defined in Equation (2). h_e is the history cost of edge e, and used for guiding the reassignment of illegal nets.

In Equation (13), if $dc(g)$ is large, much routing area of g has been used. Thus g is relatively congested and the routing resources of g are few for s, so that the congestion cost $cong(g_s)$ should be larger. Besides, if s requires relatively large area, $gc(g)$ is larger and thus the congestion cost increases. Equation (13) and (14) are designed for evaluating the congestion of g-cells and edges, respectively. The motivation of the settings for Equation (14) is similar to that for Equation (13). In this way, congestion can be evaluated based on edges and g-cells with considering wires, vias and obstacles.

The purpose of the congestion cost function is to fully consider the interaction between routing resources and the objects in the routing region. An example is used to illustrate this strategy. In Fig. 8(a), the routing region containing two metal layers has six g-cells that are $g_{1,1}$, $g_{2,1}$, $g_{3,1}$, $g_{1,3}$, $g_{2,3}$ and $g_{3,3}$. The four edges in Fig. 8(a) are represented by $e_{1,1}$, $e_{2,1}$, $e_{1,3}$ and $e_{2,3}$, respectively. The capacities of $e_{1,1}$, $e_{1,3}$ and $e_{2,3}$ are all 2, and the capacity of $e_{2,1}$ is 0 since all tracks of $e_{2,1}$ are occupied by the obstacle.

In Fig. 8, there are three nets, the order of layer assignment is n_1, n_2, n_3, and the colors of them are red, green and orange, respectively. In Fig. 8(b), only n_1 has been assigned. If congestion is considered based on wire overflow, the solution in Fig. 8(c) and the solution in Fig. 8(d) are equivalent for n_2. Besides, since each segment is assigned from the bottom layer to the top layer, the solution in Fig. 8(c) is selected eventually. And then the solution of n_3 is shown in Fig. 8(c). However, based on multi-aspect congestion awareness strategy, routing region with different occupation situation can be distinguished.

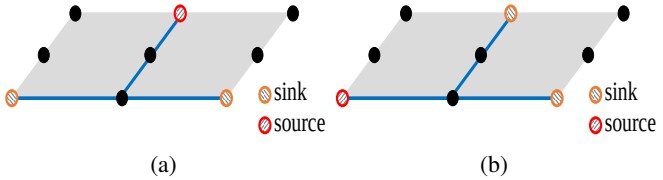


Fig. 9: 2D routing solutions of two nets for explaining total path length sorting strategy.

Specifically, routing in the region which has not been occupied by objects a lot pays lower cost. Thus, as for n_2, the congestion cost of the solution in Fig. 8(d) is lower than that in Fig. 8(c). Therefore, the solution in Fig. 8(d) is selected eventually.

Both the solutions in Fig. 8(c) and 8(d) have no illegal nets, However, Fig. 8(d) gives a 3D routing solution with fewer via count, better congestion and lower delay. Therefore, a better 3D routing solution can be selected based on multi-aspect congestion awareness strategy.

C. Total Path Length Sorting Strategy

In timing-aware layer assignment work, nets are usually sorted according to net delay. However, net delay is uncertain before an initial 3D routing solution is obtained. On the other hand, net delay is related to the total path length (TPL). If TPL of a net is greater, the net delay is usually greater. Thereby, sorting nets according to TPL in descending order can give great-delay nets priority to using routing resources.

It should be noted that TPL is different from the wire length in a net. In Fig. 9, suppose the unit length between adjacent g-cells is 1. TPL of the net in Fig. 9(a) is 6, and the wire length is 4. In addition, TPL of a net is closely related to the distribution of the source and sinks. Compared with Fig. 9(a) and 9(b), it can be seen that the wire lengths of the two nets are equal to 4, and the topological structures are similar. However, TPL of the net in Fig. 9(a) is 6, and that of the net in Fig. 9(b) is 5.

The proposed strategy sorts the nets according to TPL, so that great-delay nets have priority to choosing routing resources. This strategy reduces the uncertainty of the net order, which improves the efficiency of the proposed algorithm and the quality of the final solution.

D. Negotiation-Based Method

Since there may be illegal nets in initial 3D routing solution, negotiation-based method is adopted to reassign these nets. Specifically, if illegal nets exist, the cost of using an edge with overflow is increased in exponential form to reduce the probability of using this edge. This process is performed until the congestion constraints are satisfied. h_e in Equation (12) is used to guide this process and computed as follows:

$$h_e^{i+1} = \begin{cases} h_e^i + \rho \times 2^i & \text{if } e \text{ has overflow} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

where h_e^i and h_e^{i+1} are the history cost of e at the i-th and the $(i+1)$-th iteration, respectively. ρ is set to 0.05 according to experimental tests. By increasing h_e, Equation (15) reduces

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2021.3100296, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

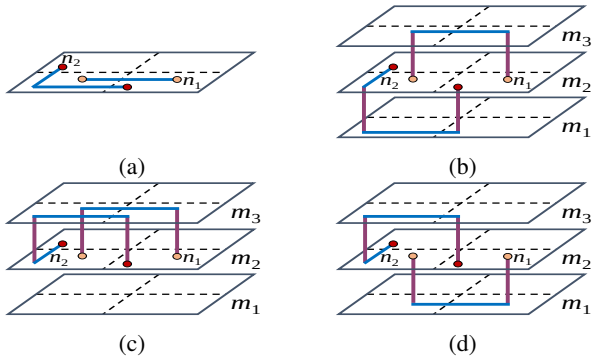


Fig. 10: (a) The 2D routing solution of n_1 and n_2. (b) The original 3D routing solution. (c) The 3D routing solution under adjustment. (d) The optimized 3D routing solution.

the probability of using the edge with overflow. Increasing h_e in exponential form is good for converging faster.

The negotiation-based method is adopted to solve the overflow problem by ripping up and reassigning some nets which pass through the congestion area. If NDR wires are used to route these nets, since the overflow in the congestion area will be further aggravated, this will undoubtedly reduce the efficiency of the algorithm. Therefore, unlike prior work, we remove the consideration of NDR wires from the negotiation-based method to effectively improve the efficiency of the proposed algorithm.

E. Maximum-Delay Net Scalpel Algorithm

After the 3D routing solution satisfying the congestion constraints is obtained, maximum-delay net scalpel algorithm is proposed to optimize the maximum delay. The basic idea can be expressed as follows: if there are timing non-critical nets that share routing area with the maximum-delay net, the routing area occupied by timing non-critical nets is released for the maximum-delay net to optimize the maximum delay.

In Fig. 10, n_1 is a timing non-critical net and n_2 is the maximum-delay net. The order of assignment is n_1, n_2, and the colors of them are orange and red, respectively. In Fig. 10(a), n_1 and n_2 compete for routing area. In the 3D routing area, the capacity of each edge is 1, and the wire resistance of an upper layer is less than that of a lower layer. To reduce the maximum delay, the routing area of an upper layer should be used preferentially by the maximum-delay net. Before using maximum-delay net scalpel algorithm, Fig. 10(b) shows the solution of n_1 and n_2. With maximum-delay net scalpel algorithm, the solution of n_2 is adjusted to the solution shown in Fig. 10(c) without considering the congestion constraints. To ensure good routability, the solution of n_1 is adjusted to the solution shown in Fig. 10(d) with considering the congestion constraints. Fig. 10(d) shows the final solution of n_1 and n_2. However, the actual circuit contains plenty of nets, only optimizing one net is not sufficient to fully reduce the maximum delay. Therefore, the proposed algorithm optimizes multiple nets with great delay in an adaptive way to minimize the maximum delay.

Algorithm 2: Maximum-Delay Net Scalpel Algorithm

```
   Input: 3D routing solution for net set N
   Output: The optimized 3D routing solution
1  Initialize: fail ← false
2  while fail is false do
3      Choose the net n_md with the maximum delay
       maxDelay;
4      Rip up and reassign n_md to get n_md2 without
       considering the congestion constraints;
5      if the delay of n_md2 >maxDelay then
6          Restore n_md;
7          fail ← true;
8      else
9          for each illegal net n_i ∈ N except n_md2 do
10             Rip up and reassign n_i to get n_i2 subject to
               the congestion constraints;
11             if the delay of n_i2 >maxDelay then
12                 Restore n_i;
13                 Rip up and reassign n_md2 to get n_md3
                   subject to the congestion constraints;
14                 if the delay of n_md3 >maxDelay then
15                     Restore n_md2;
16                     fail ← true;
17                     break;
18     if illegal nets exist then
19         Undo all operations of this loop;
20         break;
```

Algorithm 2 shows the pseudo code of maximum-delay net scalpel algorithm. Net n_{md} with the current maximum delay $maxDelay$ is got in line 3, and reassigned without considering the congestion constraints to get n_{md2} in line 4. In addition, n_{md} and n_{md2} represents the 3D routing solution of a net with the current maximum delay before and after performing the proposed net reassigning method, respectively. If the maximum delay is reduced, the nets sharing routing resources with n_{md2} are reassigned for satisfying the congestion constraints in lines 8-17. In this reassigning process, it is ensured that the delay of any net is less than $maxDelay$ in lines 11-17. Before the current loop ends, if illegal nets still exist, all operations in this loop are undone to satisfy the congestion constraints for routability in lines 18-20.

F. NDR Wire Controlling Strategy

This work adopts NDR wire technique to further reduce delay. Since NDR wires need more routing resources than default-rule wires, it is necessary to control the range of using NDR wires without making congestion worse. According to the cask effect, the limitation on timing performance of chips is mainly derived from the nets with great delay, and thus it is not significant to use NDR wires for the low-delay nets. Further, the wire segment close to the source usually has greater wire delay due to larger downstream capacitance. Fig. 11(a) shows

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2021.3100296, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

an example of the 2D routing solution of a net. The wire segment s_1 close to the source nd_1 has greater wire delay than s_2, s_3 and s_4, because the downstream capacitance of s_1 is larger than that of other segments.

However, not all wire segments close to the source have large downstream capacitance. Fig. 11(b) gives another example. In this 2D routing soliton, both s_1 and s_2 are connected with the source directly. s_1 has large downstream capacitance while s_2 does not, so that the wire delay of s_1 is greater but that of s_2 is low. Based on the analysis mentioned above, timing critical segments and timing non-critical segments should be distinguished in a net. Let regard the 2D routing solution of a net as a routing tree, and represent the characteristic value $cv(nd_i)$ of tree node nd_i as follows:

$$cv(nd_i) = \frac{dist(nd_i)}{dist(leafnd_{max-i})} \tag{16}$$

where $dist(nd_i)$ is the distance from nd_i to the source. For example, in Fig. 11(b), the distance from nd_3 to the source is 1, and the distance from nd_5 to the source is 2. In all paths from sinks to the source and through nd_i, the distance of the longest path is represented by $dist(leafnd_{max-i})$. For example, in Fig. 11(b), as for nd_2, there are two paths through this node, the distance from sink nd_5 to the source is 2, the distance from sink nd_6 to the source is 3, and thus $dist(leafnd_{max-2})$ is 3.

As for net n, the threshold value $limit(n)$ is introduced to distinguish timing critical segments from timing non-critical segments. If $cv(nd_i)$ is less than $limit(n)$, the segment connecting nd_i and its parent node is regarded as a timing critical segment, otherwise regarded as a timing non-critical segment. Since great-delay nets significantly affect timing performance of chips, the definition of $limit(n)$ is as follows:

$$limit(n) = -\frac{order(n)}{k} + b \tag{17}$$

where $order(n)$ is the position of net n in the descending sort order of all the nets according to net delay and parameter k is dependent on the total number of nets in real circuits. For instance, the order of the maximum-delay net n_{md} is 1. Parameters k and b are set as 50000 and 0.5 based on multiple sets of experimental tests. As for net n with longer delay, $order(n)$ is smaller and $limit(n)$ is greater, so that more segments in n can be regarded as timing critical segments, which has a positive impact on optimizing timing performance of the circuit. To ensure routability, the proposed algorithm allows the timing critical wire segments of top 5% nets use NDR wires, while other wire segments can only use default-rule wires.

NDR wire controlling strategy takes the timing criticality of segments into consideration according to the 2D routing solution of a net, and dynamically adjusts the threshold value of each network in an adaptive way. In this way, NDR wires can be used to optimize timing behaviors without making routability worse.

G. Via Pillar Optimization Method

Based on the latest 3D routing solution, nets are sorted according to net delay in descending order. Then all nets

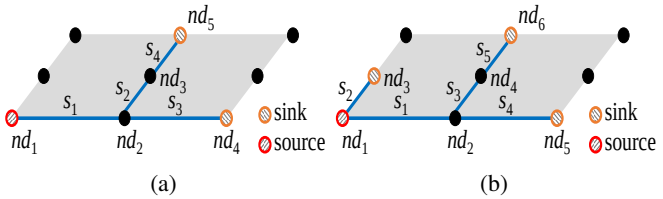


Fig. 11: 2D routing solutions of two nets for explaining NDR wire controlling strategy.

are ripped up and reassigned to further reduce delay subject to the congestion constraints, and via pillar structure is introduced to improve the timing behaviors of the final 3D routing solution. With taking full account of delay, congestion, varying types and sizes of both vias and wires, we propose an optimization method to adopt via pillars and NDR wires systematically. Since via pillars and NDR wires occupy more routing resources than normal vias and default-rule wires, the usage of via pillars and NDR wires should be controlled to avoid making routability worse. The following three aspects are taken into consideration to analyse the application scope of via pillars and NDR wires.

- Compared with the delays of other nets, the delays of timing critical nets have greater influence on the timing behaviors.
- In a timing critical net, only the segments closer to the source usually have great segment delays.
- If via pillars are used for downstream via segments in timing critical nets, the downstream capacitance of every upstream segment increases, which makes net delay greater.

Accordingly, via pillar optimization method only allows timing critical segments of timing critical nets to use via pillars and NDR wires. Specifically, in descending order of net delay, the top 5% nets are regarded as timing critical nets in this work. The timing critical segments are determined by Equation (16) and (17). In this way, the delays of timing critical nets can be reduced, and the usage of via pillars and NDR wires is controlled for routability.

To further guarantee routability, via pillar types and wire types should be taken into account when we combine via pillars and NDR wires. The type of a via pillar depends on the types of wires that are connected by this via pillar. Specifically, since parallel wires occupy two routing tracks on the lower layers and a wide wire occupies three routing tracks on the upper layers (the lower layers are layers 1 to 4, and the upper layers are layers 5 to 9 in this work), the basic via pillar types are set as Fig. 12 shows. In Fig. 12(a), a via pillar connecting parallel wires and a default-rule wire is 2×1-Type. In Fig. 12(b), a via pillar connecting two pairs of parallel wires is 2×2-Type. In Fig. 12(c), a via pillar connecting a wide wire and a default-rule wire is 3×1-Type. In Fig. 12(d), a via pillar connecting a wide wire and parallel wires is 3×2-Type. In Fig. 12(e), a via pillar connecting two wide wires is 3×3-Type. If a via pillar connects two default-rule wires, this via pillar is converted to a normal via as shown in Fig. 5(a).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2021.3100296, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

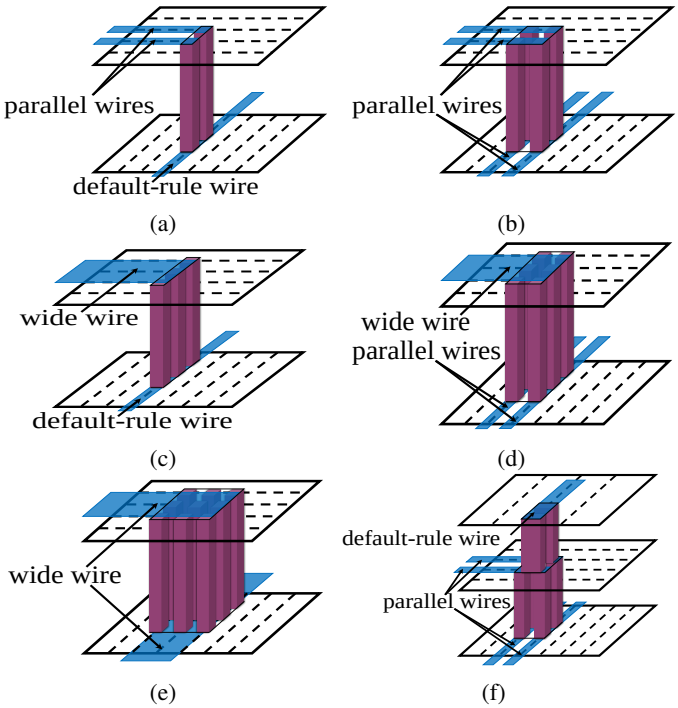


Fig. 12: (a) 2×1-Type via pillar. (b) 2×2-Type via pillar. (c) 3×1-Type via pillar. (d) 3×2-Type via pillar. (e) 3×3-Type via pillar. (f) An example of via pillar structure in multilayer structure.

According to the types of wires that are connected by via pillars, the implementation form of via pillars can adjust automatically based on the basic via pillar types mentioned above. The implementation form of via pillars is maneuverable for multilayer structure and Fig. 12(f) shows an example. In Fig. 12(f), after every wire type has been determined on each layer, a 2×1-Type via pillar is used for connecting a default-rule wire and parallel wires, and a 2×2-Type via pillar is used for connecting two pairs of parallel wires.

On all routing layers, the cost function as described in Equation (11) evaluates every possible combination type of via pillars and NDR wires for assigning a segment, and the solution with the minimal cost is selected eventually. In this way, the timing behaviors can be optimized with guaranteeing good routability.

VI. Experimental Results

The proposed VPT has been implemented in C++ language on a Linux workstation with 3.5 GHz Intel Xeon CPU and 128 GB memory. Experiments are performed on the DAC12 routability-driven placement benchmarks. The statistics of the benchmarks are summarized in Table IV. "Cir" represents circuit. The placement solutions of the benchmarks are obtained by NTUplace4 [37]. Based on the placement solutions, the 3D routing solutions which are generated by NCTU-GR 2.0 [38] are compressed to 2D routing solutions.

In the following content, we first validate the effectiveness of the proposed methods in Section VI-A, and then the performance of VPT in Section VI-B. In Fig. 13, Table VI and Table VII, "TD" and "MD" represent the total delay of all nets and the delay of the maximum-delay net, respectively. The average delays of the top 0.5%, 1%, and 5% timing-critical nets are represented by "0.5%AD", "1%AD", and "5%AD", respectively. The time unit of the delay results is picosecond (ps). "#OF" and "RT" are the number of g-cells with overflow and the runtime of the algorithms in seconds, respectively. "Ratio" is the result of a comparison algorithm to VPT in a certain indicator of a circuit, and "AR" is the average value of these ratio results of all circuits.

TABLE IV: Statistics of DAC12 benchmarks

Cir	#Grids	#Layers	#Nets
sp2	770×891	9	990899
sp3	800×415	9	898001
sp6	649×495	9	1006629
sp7	499×713	9	1340418
sp9	426×570	9	833808
sp11	631×878	9	935731
sp12	444×518	9	1293436
sp14	406×473	9	619815
sp16	465×404	9	697458
sp19	321×518	9	511685

TABLE V: Comparison results between the method with and without considering coupling effect

Cir	WCE_NW_P			WOCE_NW_P		
	TD	MD	RT	TD	MD	RT
sp2	2222970	504	1341	2425700	524	1339
sp3	775357	535	906	903792	518	903
sp6	654807	254	762	743034	246	753
sp7	624896	233	972	713584	240	989
sp9	413723	212	566	473686	222	586
sp11	750541	1602	606	811485	1475	605
sp12	542782	582	942	626970	620	1001
sp14	407862	205	475	459379	261	562
sp16	544414	186	606	635661	250	574
sp19	193844	305	304	216853	348	326
AR	1.00	1.00	1.00	1.13	1.08	1.03

In addition, the comparison results of coupling effect have been shown in Table V, where "WCE_NW_P" and "WOCE_NW_P" represent the layer assignment method with and without considering coupling effect, respectively. It can be seen that that coupling effect has a significant effect on the interconnect delay, and thus should not be ignored during layer assignment [30].

A. Validation of the Proposed Methods

In Fig. 13, "VPT" represents the complete algorithm. "VPT_MCA", "VPT_TPLS", "VPT_MNS", and "VPT_VPO" represent the complete algorithm without multi-aspect congestion awareness strategy, total path length sorting strategy, maximum-delay net scalpel algorithm, and via pillar optimization method, respectively. Fig. 13(a)-13(e) show that multi-aspect congestion awareness strategy has positive influence on reducing delay. Besides, Fig. 13(f) and 13(g) show that this strategy can significantly optimize overflow and runtime. Multi-aspect congestion awareness strategy can reduce the total delay, the maximum delay, the top 0.5% delay, the top 1% delay, the top 5% delay, overflow, and runtime by 2%, 11%, 3%, 3%, 3%, 22%, and 21.4%, respectively. Since routability

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2021.3100296, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

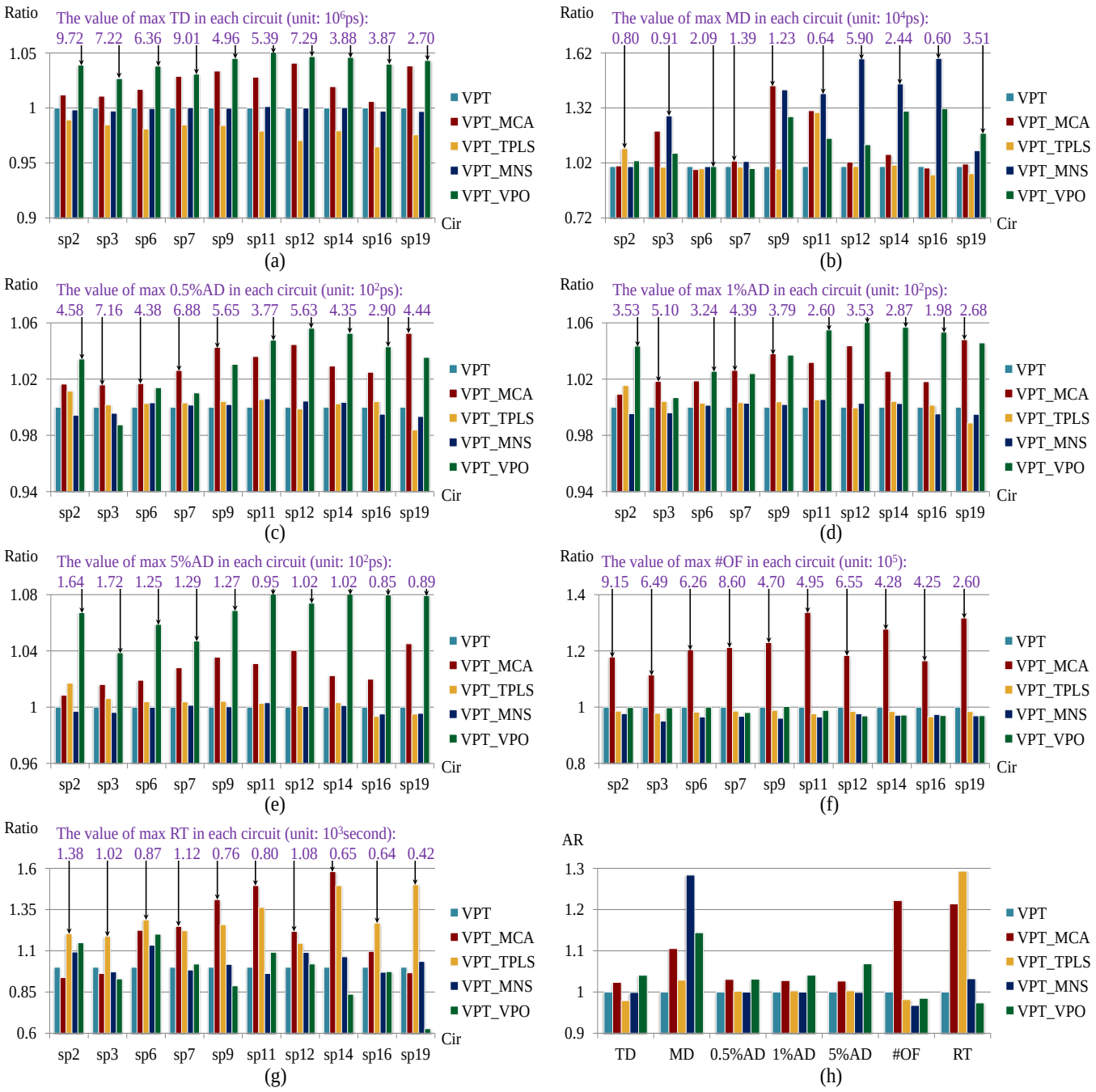


Fig. 13: (a)-(g) Experimental comparisons of VPT, VPT_MCA, VPT_TPLS, VPT_MNS, and VPT_VPO for each circuit in TD, MD, 0.5%AD, 1%AD, 5%AD, #OF, and RT, respectively. (h) Experimental comparisons of VPT, VPT_MCA, VPT_TPLS, VPT_MNS, and VPT_VPO for all circuits in AR of each target.

are positively correlated with timing behaviors to some extent, optimizing the congestion of routing solution will help to further reduce the corresponding timing delay. Thus, multi-aspect congestion awareness strategy can effectively optimize routability and timing performance simultaneously.

Fig. 13(g) shows that total path length sorting strategy can shorten runtime. Since the delay of a net is closely related to its total path length based on Elmore delay model, sorting all the nets in descending order according to their path lengths can give potential long-delay nets higher routing priority with more resources. In other words, the proposed total path length sorting strategy will lead to a more efficient processing order of nets in the initial stage of layer assignment, thereby reducing the workload of the subsequent rip-up and reassignment. As a result, the runtime of the proposed VPT can be reduced significantly. Fig. 13(b) shows that maximum-delay net scalpel algorithm significantly reduces maximum delay. Maximum-delay net scalpel algorithm can reduce the maximum delay by 28% and has a positive impact on delay optimization.

Fig. 13(a)-13(e) show that via pillar optimization method

can effectively optimize the total delay, the maximum delay, and the delays of timing-critical nets. Via pillar optimization method can reduce the total delay, the maximum delay, the top 0.5% delay, the top 1% delay, and the top 5% delay by 4%, 14%, 3%, 4%, and 7%, respectively. With Via pillar optimization method, via pillars and NDR wires are properly combined to reduce delay and overflow, thus optimizing timing and routability simultaneously. Fig. 13(h) shows that VPT including the methods mentioned above has good performance in delay, overflow and runtime.

B. Validation of VPT

To validate the performance of the proposed VPT, we compare VPT with VMD [39], DLA [30] and MND [32], respectively. Since the algorithms above do not take via size into consideration, for a fair comparison, the re-implemented these methods and apply the via model formulated in Section III-C into them. All the algorithms are run in the same environment and the corresponding comparison results are shown in Table VI and Table VII.

1) Comparison between VPT and VMD

The method VMD is generated by modifying the maximum-delay minimization algorithm in [39]. Since this method does not consider the implementation of NDR wires and via pillars, we re-implemented the method and further incorporate the consideration of NDR wires and via pillars. Compared with VMD, it can be seen from Table VI that the maximum delay is reduced by up to 22%.

2) Comparison between VPT and DLA

In Table VI and Table VII, it can be seen that VPT outperforms DLA in terms of delay, overflow and runtime in each circuit. Compared with DLA, the reduction rates are 6%, 35%, 14%, 14%, 15%, 36%, and 4% with respect to the total delay, the maximum delay, the top 0.5% delay, the top 1% delay, the top 5% delay, overflow, and rumtime, respectively. Compared with DLA, VPT adopts advanced via pillar structure to optimize the quality of 3D routing solution. Besides, total path length sorting strategy reorders nets properly and takes congestion into account when generating initial 3D routing solutions, which can release the burden of assigning illegal nets and save runtime. Furthermore, VPT proposes the maximum-delay net scalpel algorithm to optimize the maximum delay, and designs a multi-aspect congestion awareness strategy to reduce the overflow.

3) Comparison between VPT and MND

In Table VI and Table VII, it can be seen that VPT outperforms MND in terms of delay, overflow and runtime in each circuit. Compared with MND, the reduction rates are 3%, 35%, 9%, 9%, 10%, 3%, and 3% with respect to the total delay, the maximum delay, the top 0.5% delay, the top 1% delay, the top 5% delay, overflow, and rumtime, respectively.

Compared with MND, the proposed sorting strategy reorders nets according to total path length before generating initial 3D routing solution, which improves the efficiency of VPT. VPT enhances the congestion optimization strategy of MND. Specifically, VPT considers not only the track resources of edges, but also the area resources of g-cells. Further, the congestion caused by vias is taken into consideration, which has positive influence on reducing the mismatch between global routing and detailed routing. In addition, VPT improves the maximum delay optimization algorithm of MND. After the continuous optimization process for the maximum delay, if wire overflow problem exists, MND uses negotiation-based method to solve this problem, while VPT adopts a revocation mode to deal with this overflow situation. Compared with the negotiation-based method, the revocation mode can save the runtime and guarantee the optimization effect. Moreover, VPT combines via pillars and NDR wires, and limits the application scope of both techniques in a more reasonable way, so as to optimize the delay without damaging the routability. Table VIII shows the usages of NDR wires and via pillars in each circuit for VPT. "#NDRW" and "#Wire" represent the number of NDR wires and the number of all wires, respectively. "#VP" and "#Via" represent the number of via pillars and the number of all vias, respectively. "Pw" represent the proportion of "#NDRW" to "#Wire", and "Pv" represent the proportion of "#VP" to "#Via". As Table VIII shows, to avoid damaging routability, the usages of NDR wires and via pillars are limited in a low level in VPT.

TABLE VII: Experimental comparisons of VPT, VMD, DLA, and MND in #OF and RT

Cir	#OF				RT			
	VPT	VMD	DLA	MND	VPT	VMD	DLA	MND
sp2	776613	775006	962640	810017	1145	1140	1168	1147
sp3	582003	576268	695084	601244	858	891	899	923
sp6	520069	520550	672324	536231	675	669	681	687
sp7	709641	708750	958925	739438	901	992	939	909
sp9	382022	380663	550206	412683	536	545	558	581
sp11	370439	370064	597959	391706	534	504	545	541
sp12	553016	549456	727421	561345	890	836	895	909
sp14	335142	334045	462659	335422	415	401	427	415
sp16	364965	365828	450233	348728	506	490	571	516
sp19	197193	195316	299119	198581	282	265	301	298
AR	1.00	1.00	1.36	1.03	1.00	0.99	1.04	1.03

TABLE VIII: The usages of NDR wires and via pillars

Cir	#NDRW	#Wire	Pw(%)	#VP	#Via	Pv(%)
sp2	2227013	23870343	9.33	386044	6506051	5.93
sp3	1237718	16793262	7.37	425393	6409112	6.64
sp6	1315194	16238444	8.10	451558	6327067	7.14
sp7	1545380	21721530	7.11	973227	10013186	9.72
sp9	1147102	12575255	9.12	582026	5507211	10.57
sp11	1695193	15443695	10.98	550607	5707919	9.65
sp12	1606383	19146350	8.39	873158	9147428	9.55
sp14	861375	10681802	8.06	311094	4106868	7.57
sp16	799710	11603937	6.89	189472	3928648	4.82
sp19	550888	7570435	7.28	309154	3228071	9.58

VII. Conclusions

For advanced process technologies, this paper presents an efficient timing-aware layer assignment algorithm considering via pillars which includes an overflow evaluation model, two advanced techniques and three methods to optimize both delay and overflow considering coupling effect. Varying types and sizes of vias and wires are taken into account in the evaluation model to compute overflow properly. Since via delay and wire delay are main components of net delay, via pillars and NDR wires are combined to generate a synthetical delay optimization way for advanced process technologies.

TABLE VI: Experimental comparisons of VPT, VMD, DLA, and MND in TD, MD, 0.5%AD, 1%AD, and 5%AD

Cir	TD				MD				0.5%AD				1%AD				5%AD			
	VPT	VMD	DLA	MND	VPT	VMD	DLA	MND	VPT	VMD	DLA	MND	VPT	VMD	DLA	MND	VPT	VMD	DLA	MND
sp2	9356130	9314590	10050400	9588010	7281	7417	8821	8879	443	443	522	476	339	339	399	364	154	153	179	167
sp3	7036480	6981020	7334860	7211610	7157	8549	8599	8478	705	702	753	742	501	499	541	531	166	166	182	178
sp6	6126160	6130790	6407170	6243270	20919	20846	25255	25496	431	432	476	453	316	316	350	332	118	118	132	126
sp7	8743160	8719850	9244380	9017230	13533	14028	16274	16130	671	670	742	725	428	427	476	462	123	123	139	134
sp9	4745820	4732590	5104490	4960640	8532	12279	12905	12591	542	543	613	598	365	366	411	400	119	119	137	131
sp11	5125790	5128610	5462820	5310340	4569	6587	6425	6388	360	361	416	396	246	247	284	269	87	87	101	96
sp12	6961710	6859300	7389410	7191130	37129	52302	50466	50503	533	538	627	601	333	335	391	372	95	95	112	105
sp14	3708430	3697100	3924120	3809630	16838	17635	25457	25578	413	413	473	450	272	271	312	295	95	94	110	104
sp16	3720000	3717490	3841120	3737460	3748	5597	5391	5538	278	279	324	300	188	188	222	203	79	79	92	85
sp19	2584090	2546770	2753130	2695430	29689	33170	43089	42857	422	415	490	482	256	252	299	291	82	82	98	94
AR	1.00	1.00	1.06	1.03	1.00	1.22	1.35	1.35	1.00	1.00	1.14	1.09	1.00	1.00	1.14	1.09	1.00	1.00	1.15	1.10

Total path length sorting strategy gives proper processing order of nets to improve the rationality of layer assignment flow. Multi-aspect congestion awareness strategy can reduce overflow significantly and has a positive impact on timing. Maximum-delay net scalpel algorithm is designed to reduce the maximum delay, so that the timing behaviors can be further improved. Experimental results have confirmed that the proposed algorithm can optimize net delay, overflow and runtime simultaneously, and achieve the best solution quality among the existing algorithms with the shortest runtime.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grants No. 61877010 and No. 11501114, the State Key Laboratory of Computer Architecture (ICT, CAS) under Grant No. CARCHB202014, and the Fujian Natural Science Funds under Grants No. 2019J01243 and No. 2018J07005.

REFERENCES

[1] S. Mantik, G. Posser, W.-K. Chow, Y. Ding, and W.-H Liu, "ISPD 2018 initial detailed routing contest and benchmarks," in *Proceedings of International Symposium on Physical Design*, pp. 140-143, 2018.

[2] 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 *Proceedings of International Symposium on Physical Design*, pp. 147-151, 2019.

[3] C. C. N. Chu and D. F. Wong, "Greedy wire-sizing is linear time," in *Proceedings of International Symposium on Physical Design*, pp. 39-44, 1998.

[4] C. C. N. Chu and M. D. F. Wong, "An efficient and optimal algorithm for simultaneous buffer and wire sizing," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 9, pp. 1297-1304, 1999.

[5] J. Lilli, C.-K. Cheng, and T.-T. Y. Lin, "Optimal and efficient buffer insertion and wire sizing," in *Proceedings of Custom Integrated Circuits Conference*, pp. 259-262, 1995.

[6] L.-C. Lu, "Physical design challenges and innovations to meet power, speed, and area scaling trend," in *Proceedings of International Symposium on Physical Design*, pp. 63, 2017.

[7] Y. Zhong, T.-C. Yu, K.-C. Yang, and S.-Y. Fang, "Via pillar-aware detailed placement," in *Proceedings of International Symposium on Physical Design*, pp. 17-24, 2020.

[8] T.-H. Lee and T.-C. Wang, "Congestion-constrained layer assignment for via minimization in global routing," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 9, pp. 1643-1656, 2008.

[9] K.-R. Dai, W.-H. Liu, and Y.-L Li, "Efficient simulated evolution based rerouting and congestion-relaxed layer assignment on 3-D global routing," in *Proceedings of Asia and South Pacific Design Automation Conference*, pp. 570-575, 2009.

[10] Y.-J. Chang, Y.-T. Lee, J.-R. Gao, P.-C. Wu, and T.-C. Wang, "NTHU-Route 2.0: A robust global router for modern designs," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 12, pp. 1931-1944, 2010.

[11] K.-R. Dai, W.-H. Liu, and Y.-L. Li, "NCTU-GR: Efficient simulated evolution-based rerouting and congestion-relaxed layer assignment on 3-D global routing," in *IEEE Transactions on Very Large Scale Integration Systems*, vol. 20, no. 3, pp. 459-472, 2012.

[12] Y.-J. Jiang and S.-Y. Fang, "COALA: Concurrently assigning wire segments to layers for 2D global routing," in *Proceedings of International Conference On Computer Aided Design*, pp. 1-8, 2020.

[13] T.-H. Lee and T.-C. Wang, "Robust layer assignment for via optimization in multi-layer global routing," in *Proceedings of International Symposium on Physical Design*, pp. 159-166, 2009.

[14] C.-H. Hsu, H.-Y. Chen, and Y.-W. Chang, "Multilayer global routing with via and wire capacity considerations," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 5, pp. 685-696, 2010.

[15] W.-H. Liu and Y.-L. Li, "Negotiation-based layer assignment for via count and via overflow minimization," in *Proceedings of Asia and South Pacific Design Automation Conference*, pp. 539-544, 2011.

[16] D. Shi, E. Tashjian, and A. Davoodi, "Dynamic planning of local congestion from varying-size vias for global routing layer assignment," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 8, pp. 1301-1312, 2017.

[17] C. Antoniadis, D. Garyfallou, N. Evmorfopoulos, and G. Stamoulis, "EVT-based worst case delay estimation under process variation," in *Proceedings of Design, Automation and Test in Europe Conference and Exhibition*, pp. 1333-1338, 2018.

[18] M. Dalpasso, D. Bertozzi, and M. Favalli, "A boolean model for delay fault testing of emerging digital technologies based on ambipolar devices," in *Proceedings of Design, Automation and Test in Europe Conference and Exhibition*, pp. 297-300, 2018.

[19] F. A. Kuentzer, L. R. Juracy, and A. M. Amory, "On the reuse of timing resilient architecture for testing path delay faults in critical paths," in *Proceedings of Design, Automation and Test in Europe Conference and Exhibition*, pp. 379-384, 2018.

[20] H. A. Balef, K. Goossens, and J. P. d. Gyvez, "Chip health tracking using dynamic in-situ delay monitoring," in *Proceedings of Design, Automation and Test in Europe Conference and Exhibition*, pp. 304-307, 2019.

[21] G. D. Natale, E. I. Vatajelu, K. S. Kannan, and L. Anghel, "Hidden-delay-fault sensor for test, reliability and security," in *Proceedings of Design, Automation and Test in Europe Conference and Exhibition*, pp. 316-319, 2019.

[22] A. B. Kahng, U. Mallappa, L. Saul, and S. Tong, ""Unobserved Corner" Prediction: Reducing timing analysis effort for faster design convergence in advanced-node design," in *Proceedings of Design, Automation and Test in Europe Conference and Exhibition*, pp. 168-173, 2019.

[23] S. Hu, Z. Li, and C. J. Alpert, "A fully polynomial-time approximation scheme for timing-constrained minimum cost layer assignment," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 56, no. 7, pp. 580-584, 2009.

[24] S. Hu, Z. Li, and C. J. Alpert, "A faster approximation scheme for timing driven minimum cost layer assignment," in *Proceedings of International Symposium on Physical Design*, pp. 167-174, 2009.

[25] J. Sun, Y. Lu, H. Zhou, C. Yan, and X. Zeng, "Post-routing layer assignment for double patterning with timing critical paths consideration," in *Integration, the VLSI Journal*, vol. 46, no. 12, pp. 153-164, 2013.

[26] J. Ao, S. Dong, S. Chen, and S. Goto, "Delay-driven layer assignment in global routing under multi-tier interconnect structure," in *Proceedings of International Symposium on Physical Design*, pp. 101-107, 2013.

[27] D. Liu, B. Yu, S. Chowdhury, and D. Z. Pan, "Incremental layer assignment for critical path timing," in *Proceedings of Design Automation Conference*, pp. 1-6, 2016.

[28] D. Liu, B. Yu, S. Chowdhury, and D. Z. Pan, "Incremental layer

assignment for timing optimization," in *ACM Transactions on Design Automation of Electronic Systems*, vol. 22, no. 4, pp. 1-25, 2017.

[29] D. Liu, B. Yu, S. Chowdhury, and D. Z. Pan, "TILA-S: Timing-driven incremental layer assignment avoiding slew violations," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 1, pp. 231-244, 2017.

[30] S.-Y. Han, W.-H. Liu, R. Ewetz, C.-K. Koh, K.-Y. Chao, and T.-C. Wang, "Delay-driven layer assignment for advanced technology nodes," in *Proceedings of Asia and South Pacific Design Automation Conference*, pp. 456-462, 2017.

[31] W.-H. Liu, Y. Wei, C. Sze, C.-J. Alpert, Z. Li, Y.-L. Li, N. Viswanathan, "DRouting congestion estimation with real design constraints," in *Proceedings of Design Automation Conference*, pp.1-8, 2013.

[32] X. Zhang, Z. Zhuang, G. Liu, X. Huang, W.-H. Liu, W. Guo, and T.-C. Wang, "MiniDelay: Multi-strategy timing-aware layer assignment for advanced technology nodes," in *Proceedings of Design, Automation and Test in Europe Conference and Exhibition*, pp. 586-591, 2020.

[33] G.-J. Nam, M. Yildiz, D. Z. Pan, and P. H. Madden, "ISPD placement contest updates and ISPD 2007 global routing contest," in *Proceedings of International Symposium on Physical Design*, pp. 167, 2007.

[34] K. Nabors and J. White, "FastCap: A multipole accelerated 3-D capacitance extraction program," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 10, no. 11, pp. 1447-1459, 1991.

[35] J. Phillips and J. White, "A precorrected-FFT method for electrostatic analysis of complicated 3-D structures," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 10, pp. 1059-1072, 1997.

[36] R. Ewetz, C.-K. Koh, W.-H. Liu, T.-C. Wang, and K.-Y. Chao, "A study on the use of parallel wiring techniques for sub-20nm designs," in *Proceedings of Great Lakes Symposium on VLSI*, pp.129-134, 2014.

[37] M.-K. Hsu, Y.-F. Chen, C.-C. Huang, T.-C. Chen, and Y.-W. Chang, "Routability-driven placement for hierarchical mixed-size circuit designs," in *Proceedings of Design Automation Conference*, pp.1-6, 2013.

[38] 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," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 5, pp. 709-722, 2013.

[39] S. Dong, J. Ao, and F. Luo, "Delay-driven and antenna-aware layer assignment in global routing under multitier interconnect structure," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 5, pp. 740-752, 2015.

Genggeng Liu received the B.S. degree in Computer Science from Fuzhou University, Fuzhou, China, in 2009, and the Ph.D degree in Applied Mathematics from Fuzhou University in 2015. He is currently an associate professor with the College of Mathematics and Computer Science at Fuzhou University. His research interests include computational intelligence and very large scale integration physical design.

Xinghai Zhang is a master student at the College of Mathematics and Computer Science from Fuzhou University. His research interests include computational intelligence and very large scale integration physical design.

Wenzhong Guo received the B.S. and M.S degrees in Computer Science from Fuzhou University, Fuzhou, China, in 2000 and 2003, respectively, and the Ph.D degree in Communication and Information System from Fuzhou University in 2010. He is currently a full professor with the College of Mathematics and Computer Science at Fuzhou University. His research interests include computational intelligence and very large scale integration physical design.

Xing Huang received the Ph.D. degree in electronic science and technology from Fuzhou University, Fuzhou, China, in 2018. He is currently a Postdoctoral Research Fellow with the Chair of Electronic Design Automation, Technical University of Munich, Germany, sponsored by the Alexander von Humboldt Foundation. His current research interests include design automation for integrated circuits and microfluidic biochips.

Wen-Hao Liu received his Ph.D. degree in Computer Science from National Chiao Tung University, Taiwan in 2013. His research interests include routing, placement, clock synthesis, and logic synthesis. He has published more than 35 papers and 17 patents in these fields, and he has served on the technical program committee of DAC, ICCAD, ISPD, and ASPDAC. Currently, he works at Cadence as a software architect. He is the main developer of the next-generation routing engines used in multiple Cadences tools, and he has involved in the technology node enablement for 16nm, 10nm, 7nm, 5nm, 3nm, and 2nm.

Kai-Yuan (Kevin) Chao received the M.S.E. and Ph.D. degrees in electrical and computer engineering from the University of Texas at Austin, Austin, TX, in 1992 and 1995, respectively. He is currently working on 3DIC computing, cross-domain xTCO architectural optimization system, and 2D-material based ultra-low-power high-speed circuit application. Dr. Chao has coauthored more than 60 technical papers and two book chapters in the areas of VLSI/CAD, packaging, and radiology. He has worked in Intel for many leading CPUs since P6B to 14nm Broadwell as well as in Data Center Group Cloud Silicon Division, and has worked in Synopsys as the RD Group Director for Technology Pathfinding in Strategic Programs. Dr. Chao also advised few successful software, EDA, and thermal solution companies.

Ting-Chi Wang received the B.S. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, and the M.S. and Ph.D. degrees in computer sciences from the University of Texas at Austin, Austin, TX, USA. He is currently a Professor with the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan. His major research interest is in VLSI physical design automation. He received Best Paper Awards respectively from ASP-DAC 2006 and ISPD 2015, and supervised students to win the first place at ISPD 2008 Global Routing Contest. He was the General Co-Chair of VLSI-DAT 2019, the Technical Program Committee (TPC) Co-Chair of VLSI-DAT 2018, and a TPC member of ASP-DAC, DAC, DATE, ICCAD, ISLPED, and ISPD. He was an Associate Editor of ACM TODAES from 2013 to 2016, and the Chair of IEEE CEDA Taipei Chapter from 2014 to 2015.