MiniDelay: Multi-Strategy Timing-Aware Layer Assignment for Advanced Technology Nodes

Xinghai Zhang[†], Zhen Zhuang[†], Genggeng Liu[†], Xing Huang[ƒ], Wen-Hao Liu[‡], Wenzhong Guo[†], and Ting-Chi Wang[ƒ]

[†]College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China
[ƒ]Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
[‡]Block Implementation, ICD, Cadence Design Systems, Austin, TX, USA
Email: guowenzhong@fzu.edu.cn

***Abstract*—Layer assignment, a major step in global routing of integrated circuits, is usually performed to assign segments of nets to multiple layers. Besides the traditional optimization goals such as overflow and via count, interconnect delay plays an important role in determining chip performance and has been attracting much attention in recent years. Accordingly, in this paper, we propose *MiniDelay*, a timing-aware layer assignment algorithm to minimize delay for advanced technology nodes, taking both wire congestion and coupling effect into account. MiniDelay consists of the following three key techniques: 1) a non-default-rule routing technique is adopted to reduce the delay of timing critical nets, 2) an effective congestion assessment method is proposed to optimize delay of nets and via count simultaneously, and 3) a net scalpel technique is proposed to further reduce the maximum delay of nets, so that the chip performance can be improved in a global manner. Experimental results on multiple benchmarks confirm that the proposed algorithm leads to lower delay and few vias, while achieving the best solution quality among the existing algorithms with the shortest runtime.**

***Index Terms*—layer assignment, delay, non-default-rule wires, congestion, via**

I. Introduction

Layer assignment is an important phase in global routing. In this phase, segments of each net are assigned to certain metal layers. The results of layer assignment have great influence on interconnect delay, which is one of the important factors determining chip performance. During layer assignment, routing area is divided into several metal layers, particularly in advanced technology nodes, the wire width and wire spacing in upper layers are usually greater than that in lower layers, leading to a relatively small resistance of wires in upper layers [1], [2]. Accordingly, assigning timing critical nets to upper layers is good for improving the timing behaviors and thus improving the overall performance of chips.

On the other hand, the use of non-default rule (NDR) wires is an effective way to reduce interconnect delay and has been widely adopted in previous work [3]–[5]. NDR wires have two types: wide wires and parallel wires. The width of a wide wire is limited by manufacturing and can only be a predefined value. In the lower layer of advanced technology nodes, NDR wires can only be implemented in the form of parallel wires [6]. In Fig. 1, a black frame denotes the routing area of a certain layer, a red rectangle denotes a pin, a blue rectangle denotes a wire, and a broken line denotes a track. Using a default-width wire, a wide wire, and parallel wires to connect two pins are shown in Fig. 1b, Fig. 1c, and Fig. 1d, respectively. Although NDR wires require more routing area than the traditional default-width wires as Fig. 1 shows, NDR wires can reduce delay by reducing wire resistance.

Besides, routing area of each layer has its upper limit. If too many wires are assigned to upper layers, or NDR wires are used excessively to reduce delay, congestion becomes worse. As a result, routability is worse and the density of wires is increased. The high density of wires can lead to an increase in capacitance due to coupling effect, thereby affecting the timing behaviors negatively. Therefore, the use of routing area and NDR wires should be coordinated properly. Furthermore, since vias can also introduce extra delay to circuits, reducing the number of vias in the chip should also be considered carefully, and thus generating a layer assignment solution with low cost and high performance.

Vias and congestion are important issues in layer assignment, and some studies focused on them [7]–[11]. As VLSI technology enters the nanoscale, interconnect delay becomes a critical factor in circuit performance and is studied by [12]–[21]. Researches [12], [13] showed that the timing constrained minimum cost layer assignment problem is NP-complete, and proposed a fully polynomial time approximation solution. Research [14] developed post-routing layer assignment for double patterning and considered timing critical paths. Research [16] studied the more efficient layer assignment under a multilayer interconnect structure and focused on minimizing delay and via count. Research [18] proposed a timing model for critical paths that are allowed to propagate through several sequential stages. Research [19] proposed an incremental layer assignment framework, which aimed to optimize delay for timing critical nets. Research [20] proposed a timing-driven incremental layer assignment tool for reassigning layers among routing segments of nets. Although the use of parallel wires is a key technique for advanced technology nodes in the industry, [12]–[20] did not use parallel wires for layer assignment. Research [21] made up for this blank and proposed a delay-driven layer assignment (DLA) algorithm which consists of three stages. Although this recent work can effectively optimize delay with considering NDR wires and coupling effect, it has some shortcomings. 1) Much effort is wasted. Routability is neglected to obtain a low-delay solution at the first stage, but delay optimization is sacrificed for maintaining routability at the next stage. 2) It is inappropriate to use NDR wires in the process of reassigning nets with unnecessary overflow. The routing area occupied by these nets is congested and these nets are not necessarily timing-critical. 3) The assessment for congestion is limited. The congestion assessment method of DLA is not able to distinguish the differences of layer assignment solutions without

978-3-9819263-4-7/DATE20/©2020 EDAA

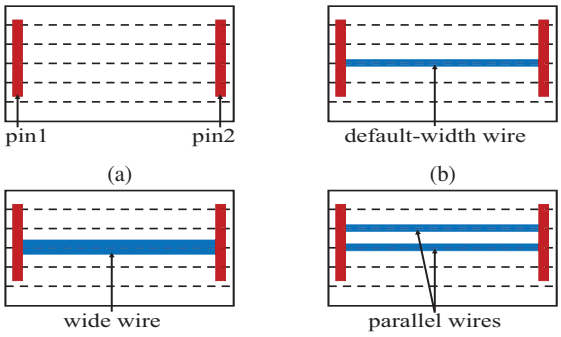

Fig. 1: (a) Two pins in a routing area. (b) Using a default-width wire to connect two pins. (c) Using a wide wire to connect two pins. (d) Using parallel wires to connect two pins.

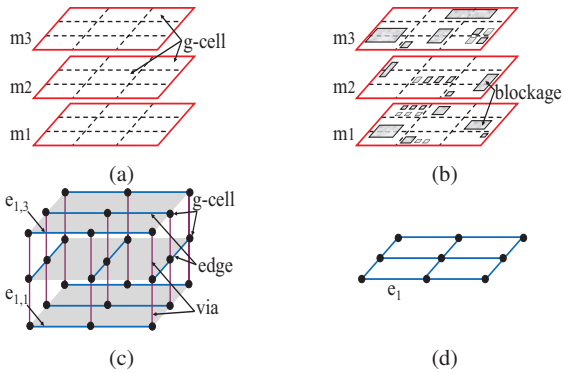

Fig. 2: (a) G-cells of a 3D routing area. (b) Blockages of a 3D routing area. (c) The 3D global routing model. (d) The 2D global routing model of (c).

unnecessary overflow. 4) No specific strategy is proposed to optimize the maximum delay which is a key factor in limiting chip performance. 5) DLA is not conducive to optimizing the delay of timing critical nets especially the maximum delay. In the negotiation-based process, timing critical nets compete with timing non-critical nets for routing area without any privilege.

Accordingly, we propose *MiniDelay*, a multi-strategy timing-aware layer assignment algorithm considering congestion and coupling effect. This work can improve the quality of layer assignment solution by using NDR wires effectively. The contributions are outlined below.

- We adopt multiple strategies to optimize delay, including congestion awareness strategy (CAS), maximum-delay net scalpel (MNS) algorithm, negotiation-based method, segments distinguishing method, probabilistic estimation method and dynamic programming method.
- We propose a congestion awareness strategy to more fully consider congestion, which can reduce delay and via count.
- We propose a maximum-delay net scalpel algorithm to reduce the maximum delay which is an important factor affecting chip performance.
- We properly use NDR wires to reduce the delay of timing critical nets considering congestion and coupling effect. Currently, the use of NDR wires is semi-automatic rather than fully automatic in the industry, which is not good enough. But we realize using NDR wires fully automatically in this work.
- Experimental results show that when compared with [21], layer assignment solutions generated by our algorithm have been greatly improved in many aspects, among which the maximum delay is optimized for 14.8%, the total delay is optimized for 9.1%, the via count is optimized for 3.6% and the runtime is optimized for 1.6%.

The rest of this paper is organized as follows. Section II introduces the problem formulation. Section III introduces MiniDelay. Section IV shows the effectiveness of MiniDelay and the conclusions are drawn in Section V.

II. PROBLEM FORMULATION

The routing area contains multiple metal layers, and each layer is divided into several rectangles with the same size. Each rectangle is defined as a g-cell. In Fig. 2a and Fig. 2b, $m1$, $m2$, and $m3$ denote three metal layers, respectively. Each layer is divided into many g-cells with the same size as shown in Fig. 2a. The gray blocks denote blockages occupying routing area as shown in Fig. 2b.

In global routing, each g-cell is abstracted into a point without geometric dimensions, and routing area can be described as the model shown in Fig. 2c. Adjacent g-cells in the preferred direction of the same layer are connected by edges. Two adjacent layers are connected by vias. The edge capacity is the number of available routing tracks of this edge. If the number of tracks occupied by objects such as nets and blockages is more than the capacity of an edge, an overflow occurs. The overflow of edge e is computed as follows:

$$of(e) = \begin{cases} u(e) - c(e) & \text{if } e \text{ has overflow} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $u(e)$ and $c(e)$ denote the current usage and the capacity of e, respectively.

Layer assignment is expressed as follows. In Fig. 2c, the set of g-cells is denoted by V_k, the set of edges is denoted by E_k, and a k-layer structure routing area is denoted by $G_k(V_k, E_k)$. The horizontal projections of G_k, V_k and E_k are denoted by G, V and E, respectively. The 2D model $G(V, E)$ from $G_k(V_k, E_k)$ is shown in Fig. 2d. Let S denote a 2D global routing solution on G, and S_k denote a 3D global routing solution on G_k. The task of layer assignment is to assign each segment of S to a corresponding edge of G_k to get S_k. e_i is a 2D edge of G and $e_{i,j}$ is the corresponding 3D edge of G_k, where j denotes the j-th layer. For example, assigning e_1 in Fig. 2d to one of $e_{1,1}$ and $e_{1,3}$ in Fig. 2c is a simple instance for layer assignment. The layer assignment problem considered in this paper is to optimize both delay considering coupling effect and via count, subject to the congestion constraints. More details are introduced in this section.

A. Delay Model

Elmore delay model is used to calculate the delay of each net. Each net has multiple sinks and one source. Each sink has a loading capacitance and the source has a driving resistance. The delay $d(s)$ of a segment s in a net is computed as follows:

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

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

The delay $d(p)$ is the total delay of all segments on a certain path p from a sink to the source. $d(p)$ is computed as follows:

$$d(p) = \sum\nolimits_{s \in S} d(s) \tag{3}$$

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

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

$$d(n) = \sum\nolimits_{p \in P} \alpha_p \times d(p) \tag{4}$$

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

Since coupling effect has influence on capacitance, coupling effect is considered in calculating delay. We adopt a probabilistic estimation method to obtain the average value of coupling capacitance to consider coupling effect [21].

B. Non-default-rule Wires

The default widths of different metal layers could be different. The default width of an upper metal layer is usually greater than that of a lower metal layer. In this paper, it is assumed that routing area consists of nine metal layers, the default widths are 1W, 2W, and 4W of the layers 1 to 4, 5 to 7, and 8 to 9, respectively. Furthermore, for timing critical nets, NDR wires can be used for them to reduce delay. NDR wires are parallel wires on layers 1 to 4 and wide wires on the other layers. Note that our layer assignment algorithm is not limited to the layer structure mentioned above, but can handle any layer structure.

Compared with default-width wires, NDR wires can reduce delay, but occupy more routing area. For example, to connect two pins, a default-width wire occupies one track while parallel wires occupy two tracks on the third layer. A default-width wire occupies one track while a wide wire occupies three tracks on the fifth layer. A wide wire whose width is greater than default width requires more wire spacing during the manufacturing process. Since wide wires and parallel wires occupy more routing area, overusing them may make congestion worse. Therefore, NDR wires should be used properly.

C. Congestion Constraints

To ensure good routability, NDR wires should be used properly in layer assignment, and the routing area of each edge should not be over-occupied. Therefore, our work obeys the following congestion constraints:

$$TWO(S_k) = TWO(S) \tag{5}$$

$$MWO(S_k) = \lceil MWO(S) \times (2/k) \rceil \tag{6}$$

MWO and TWO denote the maximum wire overflow and total wire overflow of all nets, respectively. S denotes the given 2D global routing solution, and S_k denotes the layer assignment solution of S on a k-layer structure routing area.

Equation (5) ensures that the wire overflow of the 3D layer assignment solution does not exceed that of the 2D global routing solution. Equation (6) ensures that the peak congestion of an edge in the 2D global routing solution can be assigned to its corresponding edges in the 3D layer assignment solution uniformly. Meanwhile, (6) takes into consideration the preferred direction. Furthermore, the overflow in our layer assignment algorithm is the unnecessary overflow relative to the 2D global routing solution. If a net passes through a grid edge that has unnecessary overflow, the net is called an illegal net.

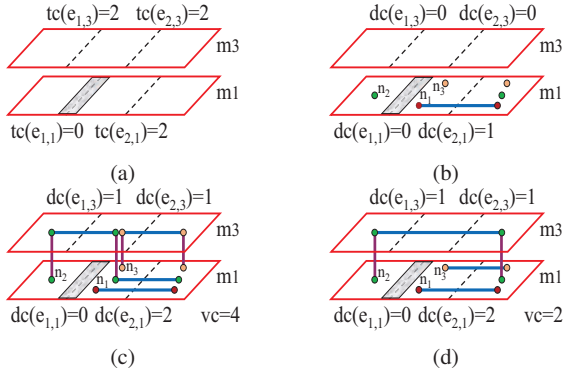

Fig. 3: (a) A routing area model. (b) A layer assignment solution of n_1. (c) A layer assignment solution of n_1, n_2 and n_3. (d) Another layer assignment solution of n_1, n_2 and n_3.

III. MiniDelay: Multi-strategy Timing-aware Layer Assignment Algorithm

A. Congestion Awareness Strategy

To generate a better layer assignment solution, we propose a congestion awareness strategy. When considering congestion, many previous layer assignment algorithms only generate congestion cost in the event of overflow. However, when assigning a segment, even if there are two or more layer assignment solutions without overflow, these solutions are different in influencing both local congestion and the flexibility of assigning subsequent nets. To more fully consider congestion, the congestion cost is defined as follows:

$$cong(s_e) = \frac{dc(e)}{tc(e)} + \frac{gc(e)}{tc(e)} - \frac{tc(e)}{mc(e)} + ofc(e) \tag{7}$$

where $cong(s_e)$ denotes the congestion cost of assigning segment s to 3D edge e. $dc(e)$ denotes the number of tracks that have been used by nets in e. $gc(e)$ denotes the number of tracks that are to be used by currently assigned net in e. $mc(e)$ denotes the total number of tracks in e, including those that cannot be used by nets, such as those occupied by blockages. $tc(e)$ denotes the number of tracks that can be used by nets in e.

In (7), if $tc(e)/mc(e)$ is large, routing area occupied by blockages is little. Routing area that can be used by nets is relatively adequate. Therefore, the cost of using edge e should be reduced. Thus, the sign of the third item is negative. As for the other three items, if their values are large, edge e is relatively congested. Therefore, the cost of using edge e should be increased. Thus, the signs of these three terms are positive.

The first three items generate congestion cost no matter whether overflow exists or not, while the last item generates congestion cost if and only if overflow exists. To avoid using the edge with overflow, $ofc(e)$ is used to increase congestion cost greatly. $ofc(e)$ is computed as follows:

$$ofc(e) = of(e) \times h_e \tag{8}$$

where h_e is history cost. h_e is set to 1 at initial stage, and the setting of h_e at other stages is introduced in Section III-C.

An example is used to explain this strategy. As shown in Fig. 3a, the routing area has six g-cells. Three of the g-cells are located in the first layer, and the other three are located in the third layer. The preferred direction of these two layers are the same. $e_{i,j}$ denotes a 3D edge connecting adjacent g-cells in the preferred direction of the same layer. j denotes the j-th layer, and i denotes the i-th edge. The four edges in Fig. 3 are denoted as $e_{1,1}$, $e_{2,1}$, $e_{1,3}$ and $e_{2,3}$. $mc\,(e_{1,1})$, the total number of tracks in $e_{1,1}$, is 2. Blockages occupy all the tracks of $e_{1,1}$, thus $tc\,(e_{1,1})$ is 0. In this example, $mc\,(e_{2,1})$, $mc\,(e_{1,3})$, $mc\,(e_{2,3})$, $tc\,(e_{2,1})$, $tc\,(e_{1,3})$ and $tc\,(e_{2,3})$ are 2. $gc\,(e_{i,j})$ is 1 for each edge.

In Fig. 3, there are three nets, the order of assignment is n_1, n_2, n_3, and the color of them are red, green and orange, respectively. In Fig. 3b, n_1 has been assigned but n_2 and n_3 have not. If congestion is assessed based on (8), the solution in Fig. 3c and the solution in Fig. 3d are equivalent for n_2. Besides, since each segment is assigned from the bottom layer to the top layer, the priority of the solution in Fig. 3c is higher than that in Fig. 3d for n_2. Therefore, the solution in Fig. 3c is selected eventually. And then the solution of n_3 is shown in Fig. 3c. However, based on (7), as for n_2, the value of $cong(s_{e_2})$ is the calculation result of the four items whose value are 0.5, 0.5, 1.0, and 0.0, respectively, for the solution of Fig. 3c while that are 0.0, 0.5, 1.0, and 0.0, respectively, for the solution of Fig. 3d. Therefore, the cost of the solution in Fig. 3c is greater than that of the solution in Fig. 3d for n_2. Thus, the solution in Fig. 3d is selected eventually. And then the solution of n_3 is shown in Fig. 3d.

Both the solution in Fig. 3c and the solution in Fig. 3d satisfy the congestion constraints, but delay and via count of the solution in Fig. 3d are smaller. Therefore, based on congestion awareness strategy, a better layer assignment solution can be selected.

B. Maximum-delay Net Scalpel Algorithm

The maximum delay is a key factor affecting chip performance. Accordingly, 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.

An example is used to explain this algorithm. In Fig. 4, n_1 is a timing non-critical net and n_2 is the maximum-delay net. The color of n_1 and n_2 are orange and red, respectively. The order of assignment is n_1, n_2. Fig. 4a shows the 2D routing solution of n_1 and n_2 that compete for routing area. In 3D routing area, the capacity of each edge is 1. 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. 4b shows the layer assignment solution of n_1 and n_2. With maximum-delay net scalpel algorithm, the layer assignment solution of n_2 is adjusted to the solution shown in Fig. 4c without considering the congestion constraints. To ensure good routability, the layer assignment solution of n_1 is adjusted

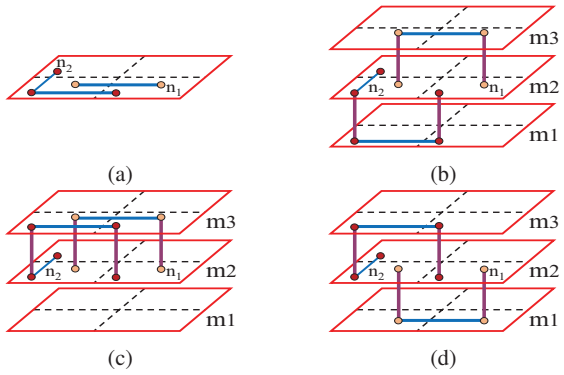

Fig. 4: (a) A 2D routing solution of n_1 and n_2. (b) Layer assignment solution of n_1 and n_2 before using MNS algorithm. (c) Layer assignment solution of n_1 and n_2 when using MNS algorithm. (d) Layer assignment solution of n_1 and n_2 after using MNS algorithm.

to the solution shown in Fig. 4d with considering the congestion constraints. Fig. 4d shows the final layer assignment solution of n_1 and n_2. Therefore, based on maximum-delay net scalpel algorithm, the solution of the maximum-delay net can be adjusted for better timing behaviors without making routability worse.

C. Algorithm Flow

The proposed layer assignment algorithm is to optimize delay with considering coupling effect, subject to the congestion constraints based on (5) and (6). NDR wires are used properly to reduce delay without making routability worse. The proposed algorithm consists of four stages: initial stage, repair stage, optimization stage and refinement stage. Based on dynamic programming, nets are assigned one by one at each stage. To consider delay, via count and congestion comprehensively, the cost function is computed as follows:

$$\alpha \times d(n) + \beta \times vc(n) + \gamma \times \sum\nolimits_{s_e \in n} cong(s_e) \tag{9}$$

where $d\,(n)$ denotes the delay of net n and $vc\,(n)$ denotes the via count of net n. According to several experimental tests, the values of α, β and γ are set as follows. At the first three stages, α, β and γ are set to 10, 1 and 1, respectively. At the last stage, α, β and γ are set to 10, 1.5 and 1, respectively.

Initial stage generates a primary layer assignment solution for subsequent stages. Nets are assigned according to the default order in the benchmark. To avoid wasting much effort, attention is not only focused on delay but also focused on congestion. Congestion awareness strategy is adopted to assess congestion in these four stages. Besides, via count should be considered as well. The via delay of a net is an important component of the total delay, and reducing via count can optimize the assignment cost. Therefore, to generate a good primary solution, delay, congestion and via count are all considered. Furthermore, although it does not ensure that the primary solution satisfies the congestion constraints, the burden of the next stage can be reduced since the congestion is considered at this stage. It is mainly reflected in saving runtime and reducing delay.

The main task of repair stage is to reassign illegal nets until the congestion constraints are satisfied. The negotiation-based

TABLE I: Experimental results with and without maximum-delay net scalpel algorithm

Circuit	TD		MD		#vc	
	NO_MNS	MNS	NO_MNS	MNS	NO_MNS	MNS
sp2	2222970	2217160	503.897	475.502	7129655	7119848
sp3	775357	772222	535.101	428.923	6382992	6382773
sp6	654807	651665	253.668	212.033	6202333	6165813
sp7	624896	621734	233.242	175.786	9335272	9317876
sp9	413723	414315	211.535	154.011	5120143	5125955
sp11	750541	743459	1602.300	1393.020	5592279	5564283
sp12	542782	539915	581.533	579.734	8500960	8435546
sp14	407862	404983	204.972	201.838	4038943	4030948
sp16	544414	544598	186.320	158.814	4122687	4086418
sp19	193844	195194	304.509	307.009	3038941	3024003
AVG	713120	710525	461.708	408.667	5946421	5925346
ratio	1.000	0.997	1.000	0.877	1.000	0.996

method is adopted to guide layer assignment. Specifically, if a segment is assigned to an edge whose all routing area is occupied by other segments or blockages, the congestion cost of using this edge is increased to avoid assigning a segment to this edge. The expression of h_e in (8) is computed as follows:

$$h_e^{i+1} = \begin{cases} h_e^i + \rho \times 2^i & \text{if } e \text{ has overflow} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

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 a parameter set to 0.05 according to experimental tests.

NDR wires are prohibited at repair stage. The nets that are reassigned at this stage are illegal nets. The routing area occupied by these nets is relatively congested and these nets are not necessarily timing-critical. Thus, it is inappropriate to use NDR wires for these nets.

At optimization stage, maximum-delay net scalpel algorithm is adopted to optimize the maximum delay. This stage reassigns timing critical nets. Therefore, NDR wires are allowed to be used for the timing critical segments of timing critical nets. Timing critical segments are the segments close to the source of a net. To distinguish timing critical segments from timing non-critical segments in a net, segments distinguishing method is adopted. With segments distinguishing method, segments closer to the source of a net are given higher scores while segments farther from the source are given lower scores. In this way, the maximum delay can be reduced, and the use of NDR wires is strictly limited to avoid making routability worse.

All nets are sorted according to delay of each net at refinement stage. Based on this order, subject to the congestion constraints, all nets are ripped up and reassigned once for delay optimization. Although delay and via count are two different goals, to some extent, they are positively correlated. Therefore, attention to via count is increased at this stage, which is reflected in setting the value of β in (9). We tried several values in the range of 0.5-15 with a gradient of 0.25. In multi-group parameter experiments, setting β to 1.5 has the best effect. Besides, the top 5% of nets are allowed to use NDR wires, and other nets are only allowed to use the default-width wires.

IV. Experimental Results

The proposed MiniDelay has been implemented in C++ language on a Linux workstation with 3.5 GHz Intel Xeon CPU and 128 GB memory. To compare the experimental results fairly, we run the program of DLA in the same experimental environment and use the same DAC12 routability-driven placement benchmarks. Each benchmark has a multilayer structure. Since the overflow of the 2D global routing solution is 0 in each benchmark, the layer assignment solutions generated by our algorithm and DLA have no overflow subject to the congestion constraints. Therefore, no data about overflow is shown in experimental results.

A. Validation of Maximum-delay Net Scalpel Algorithm

To illustrate the effectiveness of maximum-delay net scalpel algorithm, we conduct two experiments to compare the results in Table I. "MNS" and "NO_MNS" denote the algorithm with and without maximum-delay net scalpel algorithm, respectively. "MD" and "TD" denote the delay of the maximum-delay net and total delay of all nets, respectively. "#vc" denotes the via count of all nets. "AVG" is the average value of all benchmarks. "ratio" is the average ratio of the MNS values to corresponding NO_MNS values in each benchmark. The time unit for the delay results is picosecond. In Table I, maximum-delay net scalpel algorithm can reduce the maximum delay by 12.3%, and have a positive impact on total delay and via count. In summary, maximum-delay net scalpel algorithm can effectively optimize delay, especially the maximum delay.

B. Validation of Congestion Awareness Strategy

To illustrate the effectiveness of congestion awareness strategy, we conduct two experiments to compare the results in Table II. "CAS" and "NO_CAS" denote the algorithm with and without congestion awareness strategy, respectively. The average delays of the top 0.5%, 1%, and 5% timing-critical nets are respectively shown in the "0.5%", "1%", and "5%" columns of Table II. As shown in Table II, with congestion awareness strategy, the reduction rates are 4.6%, 6.9%, 4.0%, 4.4%, 5.4%, and 1.0% for the total delay, the maximum delay, the top 0.5% delay, the top 1% delay, the top 5% delay, and via count, respectively.

C. Final Comparison: Validation of MiniDelay

To illustrate the advantage of our algorithm, we conduct two experiments to compare the results in Table III. MiniDelay denotes the full version of our algorithm and DLA denotes the algorithm of [21]. As shown in Table III, our delay values and via count are better than DLA, and the average runtime whose unit is second is also better than DLA. The reduction rates are 9.1%, 14.8%, 9.8%, 10.9%, 12.1%, 3.6%, and 1.6% for the total delay, the maximum delay, the top 0.5% delay, the top 1% delay, the top 5% delay, via count, and runtime, respectively.

Specially, congestion awareness strategy has a positive impact on reducing delay and via count of all nets. Maximum-delay net scalpel algorithm can effectively reduce the maximum delay. Besides, considering congestion at initial stage also helps to reduce the total delay and via count. This approach also reduces the workload of repair stage and the runtime is saved. NDR wires are used for the timing critical nets at optimization and refinement stages. In this way, the purpose of not excessively occupying routing area is achieved by limiting the use of NDR wires, and NDR wires are used in a targeted manner for the timing critical segments of timing critical nets to improve delay.

V. Conclusions

In this paper, we have proposed a four-stage MiniDelay which includes congestion awareness strategy, maximum-delay net scalpel algorithm, negotiation-based method, segments distinguishing method, probabilistic estimation method and dynamic

TABLE II: Experimental results with and without congestion awareness strategy

Circuit	TD		MD		0.5%		1%		5%		#vc	
	NO_CAS	CAS	NO_CAS	CAS	NO_CAS	CAS	NO_CAS	CAS	NO_CAS	CAS	NO_CAS	CAS
sp2	2222970	2178310	503.897	521.397	145.040	143.274	116.200	114.932	51.476	50.632	7129655	7016786
sp3	775357	747977	535.101	515.606	89.119	86.364	66.155	64.189	23.212	22.280	6382992	6310625
sp6	654807	627400	253.668	217.987	57.257	54.891	43.950	42.213	16.932	16.099	6202333	6144733
sp7	624896	588785	233.242	195.569	43.106	40.682	31.445	29.393	10.871	10.081	9335272	9254774
sp9	413723	393268	211.535	197.886	46.031	44.445	34.157	32.634	13.069	12.304	5120143	5067387
sp11	750541	714755	1602.300	1505.300	109.711	106.685	68.188	65.594	19.770	18.659	5592279	5546025
sp12	542782	513512	581.533	532.895	40.209	38.443	29.912	28.326	10.118	9.514	8500960	8455465
sp14	407862	386961	204.972	196.754	56.046	54.041	41.876	40.005	15.245	14.303	4038943	3997575
sp16	544414	525423	186.320	176.858	54.418	52.775	42.922	41.830	18.872	18.189	4122687	4070159
sp19	193844	181226	304.509	279.549	24.561	22.326	18.973	17.284	8.657	7.883	3038941	3015080
AVG	713120	685762	461.708	433.980	66.550	64.393	49.378	47.640	18.822	17.994	5946421	5887861
ratio	1.000	0.954	1.000	0.931	1.000	0.960	1.000	0.956	1.000	0.946	1.000	0.990

TABLE III: Comparison between DLA and MiniDelay

Circuit	TD		MD		0.5%		1%		5%		#vc		Runtime	
	DLA	MiniDelay	DLA	MiniDelay	DLA	MiniDelay	DLA	MiniDelay	DLA	MiniDelay	DLA	MiniDelay	DLA	MiniDelay
sp2	2222970	2139820	503.897	475.170	145.040	138.327	116.200	111.922	51.476	49.717	7129655	6857552	1025.885	926.435
sp3	775357	718602	535.101	367.956	89.119	83.200	66.155	62.225	23.212	21.137	6382992	6186023	671.553	653.051
sp6	654807	592284	253.668	203.883	57.257	50.906	43.950	38.881	16.932	14.853	6202333	5945350	572.153	491.159
sp7	624896	558499	233.242	187.797	43.106	38.578	31.445	27.589	10.871	9.241	9335272	9134247	725.330	649.707
sp9	413723	376370	211.535	140.718	46.031	42.820	34.157	30.903	13.069	11.515	5120143	4990615	426.852	395.648
sp11	750541	695731	1602.300	1595.970	109.711	104.118	68.188	63.012	19.770	17.776	5592279	5388497	462.180	589.800
sp12	542782	487128	581.533	536.043	40.209	36.217	29.912	26.043	10.118	8.847	8500960	8283491	678.570	581.177
sp14	407862	363108	204.972	192.094	56.046	49.510	41.876	36.303	15.245	13.054	4038943	3907452	348.640	309.337
sp16	544414	485113	186.320	155.425	54.418	46.744	42.922	36.160	18.872	15.980	4122687	3796930	448.206	647.247
sp19	193844	172812	304.509	280.957	24.561	20.353	18.973	15.789	8.657	7.141	3038941	2951412	233.365	190.854
AVG	713120	658947	461.708	413.601	66.550	61.077	49.378	44.883	18.822	16.926	5946421	5744157	559.273	543.442
ratio	1.000	0.909	1.000	0.852	1.000	0.902	1.000	0.891	1.000	0.879	1.000	0.964	1.000	0.984

programming method. Congestion awareness strategy is proposed to assess the congestion for guiding layer assignment. Maximum-delay net scalpel algorithm is adopted to reduce the maximum delay which is an important factor in affecting chip performance. Both congestion and coupling effect are considered for delay calculation and routability. NDR wires are used properly to reduce the delay of timing critical nets. As the experimental results show, our layer assignment algorithm can optimize the maximum delay, the total delay, timing critical nets delay, via count and runtime well, and finally achieve the best solution quality among the existing algorithms with shortest runtime.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grants No. 61877010 and No. 11501114, and the Fujian Natural Science Funds under Grant No. 2019J01243. Wenzhong Guo is the corresponding author of the article.

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 M. 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] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] B. Li, N. Chen, Y. Xu, and U. Schlichtmann, "On timing model extraction and hierarchical statistical timing analysis," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 3, pp. 367–380, 2013.

[16] 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.

[17] G. L. Zhang, B. Li, and U. Schlichtmann, "EffiTest: Efficient delay test and statistical prediction for configuring post-silicon tunable buffers," in *Proceedings of Design Automation Conference*, pp. 60-65, 2016.

[18] G. L. Zhang, B. Li, M. Hashimoto, and U. Schlichtmann, "VirtualSync: Timing optimization by synchronizing logic waves with sequential and combinational components as delay units," in *Proceedings of Design Automation Conference*, pp. 1-6, 2018.

[19] 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.

[20] 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.

[21] 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.